Algorithmica (2007) 47: 299–321
DOI: 10.1007/s00453-006-0196-3

Algorithmica

An FPTAS for Quickest Multicommodity Flows with Inflow-Dependent Transit Times[1]

Alex Hall,[2] Katharina Langkau,[3] and Martin Skutella[4]

Abstract. Given a network with capacities and transit times on the arcs, the quickest flow problem asks for a "flow over time" that satisfies given demands within minimal time. In the setting of flows over time, flow on arcs may vary over time and the transit time of an arc is the time it takes for flow to travel through this arc. In most real-world applications (such as, e.g., road traffic, communication networks, production systems, etc.), transit times are not fixed but depend on the current flow situation in the network. We consider the model where the transit time of an arc is given as a non-decreasing function of the rate of inflow into the arc. We prove that the quickest s–t-flow problem is NP-hard in this setting and give various approximation results, including a fully polynomial time approximation scheme (FPTAS) for the quickest multicommodity flow problem with bounded cost.

Key Words. Network flows over time, Approximation, Transit times.

1. Introduction. Flows over time were introduced more than 40 years ago by Ford and Fulkerson [7], [8]. Given a directed graph with capacities and transit times on the arcs, a source node s, a sink node t, and a time horizon T, they consider the problem of sending the maximum possible amount of flow from s to t within T time units. A flow over time specifies a flow rate for each arc at each point in time. The capacity of an arc is an upper bound on this flow rate, i.e., on the amount of flow that can be sent into the arc during each unit of time. Flow on an arc progresses at a constant speed which is determined by its transit time.

Known results for flows over time with constant transit times. Ford and Fulkerson show that the maximum s–t-flow over time problem can be solved by essentially one static min-cost flow computation in the given network, where transit times are interpreted as

[1] An extended abstract of this work appeared in *Proceedings of the* 6th *International Workshop on Approximation Algorithms for Combinatorial Optimization Problems* [11]. This work was supported in part by EU Thematic Network APPOL II, Approximation and Online Algorithms, IST-2001-30012, and by DFG Focus Program 1126, "Algorithmic Aspects of Large and Complex Networks", Grant No. SK 58/4-1 and SK 58/5-3. It was also supported by the joint Berlin/Zürich graduate program Combinatorics, Geometry, and Computation (CGC) financed by ETH Zürich and the German Science Foundation Grant GRK 588/2.

[2] Institut für Theoretische Informatik, ETH Zentrum, CAB H 39.2, 8092 Zürich, Switzerland. alex.hall@gmail.com.

[3] Institut für Mathematik, TU Berlin, Straße des 17 Juni 136, 10623 Berlin, Germany. langkau@math.tu-berlin.de.

[4] Fachbereich Mathematik, Universität Dortmund, 44221 Dortmund, Germany. martin.skutella@uni-dortmund.de.

Received September 2003; revised October 2005. Communicated by A. Panconesi.
Online publication February 28, 2007.

costs. An arbitrary path decomposition of such a static min-cost flow can be turned into a flow over time by sending flow at the given flow rate into each path as long as there is enough time left for the flow on a path to arrive at the sink before time T. A flow featuring this structure is called “temporally repeated.”

A problem closely related to the maximum s–t-flow over time problem is the quickest s–t-flow problem. Here, the flow value (or “demand”) is fixed and the task is to find a flow over time with minimal time horizon T. Clearly, this problem can be solved in polynomial time by incorporating the algorithm of Ford and Fulkerson into a binary search framework. Burkard et al. [2] give a strongly polynomial algorithm for the quickest s–t-flow problem which is based on the parametric search method of Megiddo [22]. Hoppe and Tardos [15], [16] study the quickest trans-shipment problem which, given supplies and demands at the nodes, asks for a flow over time that zeros all supplies and demands within minimal time. They give a polynomial time algorithm which is, however, based on a submodular function minimization routine.

The latter fact already indicates that flow over time problems are, in general, considerably harder than their static counterparts in classical network flow theory. The best evidence for this allegation is maybe provided by a surprising result of Klinz and Woeginger [18]. They show that computing a quickest s–t-flow of minimum cost in a network with cost coefficients on the arcs is already NP-hard in series–parallel networks. Moreover, it is even strongly NP-hard to find a quickest temporally repeated s–t-flow of minimum cost.

Only recently, Hall et al. [10] have shown that computing quickest multicommodity flows is NP-hard, even on series–parallel networks. They also take a closer look at the common distinction between the settings where storage of flow at intermediate nodes is allowed and where storage is not allowed. The NP-hardness result holds for both settings. For the case where storage at intermediate nodes is not possible they give an entirely different reduction which proves that this case is even strongly NP-hard and that no fully polynomial time approximation scheme (FPTAS) exists, unless $\mathrm{P} = \mathrm{NP}$.

On the other hand, Ford and Fulkerson [7], [8] introduce the concept of time-expanded networks which allows us to solve many flow over time problems in pseudopolynomial time. The node set of a time-expanded network consists of several copies of the node set of the underlying graph building a “time layer”. The number of time layers is equal to the integral time horizon T and thus pseudopolynomial in the input size. Copies of an arc of the underlying graph join copies of its endnodes in time layers whose distances equal the transit time of that arc. Ford and Fulkerson observe that a flow over time in the given graph corresponds to a static flow in the time-expanded network, and vice versa. Thus, many flow over time problems can be solved by static flow computations in the time-expanded network.

Fleischer and Skutella [4], [6] come up with so-called “condensed” time-expanded networks which are of polynomial size and can be used to compute provably good multicommodity flows over time with costs in polynomial time. In particular, they present an FPTAS for the quickest multicommodity flow problem with bounded cost [4]–[6]. Using completely different techniques, they also show that 2-approximate temporally repeated flows can be obtained from a static, length-bounded flow computation in the given graph [4], [6]. The advantage of the latter solutions is that they have a very simple structure and also do not use storage of flow at intermediate nodes.

Flow-dependent transit times. So far we have considered the setting of flows over time where transit times of arcs are fixed. In many practical applications, however, the latter assumption is not realistic since transit times vary with the flow situation on an arc. We refer to [1], [23], and [24] for an overview and further references. Usually, the correlation of the transit time and the flow situation on an arc is highly complex. It is a major challenge to come up with a mathematical model that, on the one hand, captures the real behavior as realistically as possible and, on the other hand, can be solved efficiently even on large networks.

Köhler and Skutella [20] consider a model where, at any moment in time, the actual speed of flow on an arc depends on the current amount of flow on the arc. Under this assumption, they give a 2-approximation algorithm for the quickest s–t-flow problem and show that no polynomial time approximation scheme (PTAS) exists, unless P = NP. A simpler model is studied by Carey and Subrahmanian [3]. They assume that the transit time on an arc only depends on the current rate of inflow into the arc and propose a time-expanded network whose arcs somehow reflect this behavior. Köhler et al. [19] give a 2-approximation algorithm for the quickest s–t-flow problem in the setting of inflow-dependent transit times. The algorithm uses the algorithm of Ford and Fulkerson [7], [8] on a so-called "bow graph" with fixed transit times on the arcs. In the bow graph, every arc of the original graph is replaced by a bunch of parallel arcs corresponding to different transit times. The quickest flow problem in the bow graph is a relaxation of the quickest flow problem with inflow-dependent transit times. Note that this connection was already mentioned in the early 1960s in a graduate report by Jorgensen [17, page 9].

Hall and Schilling [12] suggest the rate-dependent setting which attempts to capture more realistically the behavior of flow in road networks. Aspects such as restricting the intermediate storage of flow and modeling congestion effects are considered. They present a purely heuristic approach to compute rate-dependent flows, which is based on the FPTAS presented in this paper for the inflow-dependent setting. The results of an experimental comparison of inflow-dependent and rate-dependent flows are given.

Contribution of this paper. While, for the special case of constant transit times, quickest s–t-flows can be computed in polynomial time [2], [7], [8], we show in Section 6 that the problem becomes NP-hard if we allow inflow-dependent transit times. In Section 4 we generalize the 2-approximation result given in [19] to the setting with costs and multiple commodities. Our approach is based on a new and stronger relaxation of the quickest flow problem, which we introduce in Section 3. This relaxation is defined in a bow graph similar to the one introduced in [19], but it uses additional "coupling constraints" between flow values on different copies of one arc in the original graph. In particular, this relaxation can no longer be solved by standard network flow algorithms but requires general linear programming techniques. Nevertheless, as shown in Section 4, the approximation technique based on length-bounded static flows presented in [4] and [6] can be generalized to yield provably good solutions to our bow graph relaxation. Moreover, we prove that such a solution to the relaxation can be turned into a feasible multicommodity flow over time with inflow-dependent transit times and bounded cost.

The main contribution of this paper is an FPTAS for the quickest multicommodity flow problem with bounded cost and inflow-dependent transit times (see Section 5). It again uses the new bow graph relaxation introduced in Section 3 and generalizes the approach based on condensed time-expanded networks from [5] and [6]. Interestingly, the time-expanded version of our bow graph relaxation essentially coincides with the modified time-expanded graph considered by Carey and Subrahmanian [3].

Both the 2-approximation algorithm and the FPTAS are for the setting where flow can be stored temporarily at intermediate nodes. It follows from [10] that, unless P = NP, the latter result cannot be achieved if storage of flow at intermediate nodes is forbidden.

While approximation results and, in particular, approximation schemes are often considered to be of purely theoretical interest, the situation is quite different here. Flow-dependent transit times represent a crucial phenomenon inherent in many real-world applications of network flows. Nevertheless, there are hardly any models and algorithmic techniques known which are capable of providing reasonable solutions even for networks of rather modest size. The FPTAS for inflow-dependent transit times presented in this paper is based on rather simple and efficient flow computations in condensed time-expanded networks. It therefore reveals a promising direction and raises hope for the development of efficient and flexible tools that can deal with reasonably sized real-world networks.

2. Preliminaries. We consider network flow problems in a directed graph $G = (V, E)$ with $n := |V|$ nodes and $m := |E|$ arcs. Each arc $e \in E$ has associated with it a positive capacity u_e and a non-negative, non-decreasing transit time function τ_e: $[0, u_e] \to \mathbb{R}^+$. There is a set of commodities $K = \{1, \ldots, k\}$; every commodity $i \in K$ is defined by a source–sink pair[5] $(s_i, t_i) \in V \times V$. The objective is to send a prespecified amount of flow $d_i > 0$, called the demand, from s_i to t_i. Finally, each arc e has associated cost coefficients $c_{e,i}$, for $i \in K$, where $c_{e,i}$ is interpreted as the cost (per flow unit) for sending flow of commodity i through the arc. For an arc $e = (v, w) \in E$, we use the notation head$(e) := w$ and tail$(e) := v$.

2.1. *Static Flows*. A *static* (*multicommodity*) *flow* x in G assigns every arc e and commodity i a non-negative flow value $x_{e,i}$ such that *flow conservation* holds:

$$\sum_{e \in \delta^-(v)} x_{e,i} - \sum_{e \in \delta^+(v)} x_{e,i} = 0 \qquad \text{for all} \quad v \in V \setminus \{s_i, t_i\} \quad \text{and} \quad i \in K.$$

Here, $\delta^+(v)$ and $\delta^-(v)$ denote the set of arcs leaving and entering node v, respectively. The static flow x satisfies all demands if

$$\sum_{e \in \delta^-(t_i)} x_{e,i} - \sum_{e \in \delta^+(t_i)} x_{e,i} = d_i \qquad \text{for all} \quad i \in K.$$

It is called *feasible* if it obeys the capacity constraints $x_e := \sum_{i \in K} x_{e,i} \le u_e$, for all $e \in E$. The cost of a static flow is defined as $c(x) := \sum_{e \in E} \sum_{i \in K} c_{e,i}\, x_{e,i}$.

[5] To simplify notation, we restrict to the case of only one source and one sink for each commodity. However, our results can be directly generalized to the case of several sources and sinks with given supplies and demands for each commodity.

2.2. *Flows over Time with Constant Transit Times.* A (*multicommodity*) *flow over time* f in G with time horizon T is given by Lebesgue-measurable functions $f_{e,i}\colon [0, T) \to \mathbb{R}^+$, where $f_{e,i}(\theta)$ is the rate of flow (per time unit) of commodity i entering arc e at time θ. In order to simplify notation, we sometimes use $f_{e,i}(\theta)$ for $\theta \notin [0, T)$, implicitly assuming that $f_{e,i}(\theta) = 0$ in this case. The capacity u_e is an upper bound on the rate of flow entering arc e at any moment of time, i.e., $f_e(\theta) \le u_e$ for all $\theta \in [0, T)$ and $e \in E$. Here, $f_e(\theta) := \sum_{i\in K} f_{e,i}(\theta)$ is the total rate at which flow is entering arc e at time θ.

In the original setting of flows over time, the transit time function τ_e of arc e is assumed to be constant. Then the flow $f_{e,i}(\theta)$ of commodity i entering arc e at time θ arrives at head(e) at time $\theta + \tau_e$. All arcs must be empty from time T on, i.e., $f_{e,i}(\theta) = 0$ for $\theta \ge T - \tau_e$. To generalize the notion of *flow conservation*, we define

$$D^-_{v,i}(\xi) := \sum_{e\in\delta^-(v)} \int_{\tau_e}^{\xi} f_{e,i}(\theta - \tau_e)\, d\theta$$

to be the total inflow of commodity $i \in K$ into node v until time $\xi \in [0, T]$. Similarly,

$$D^+_{v,i}(\xi) := \sum_{e\in\delta^+(v)} \int_{0}^{\xi} f_{e,i}(\theta)\, d\theta \tag{1}$$

is the corresponding outflow. We consider the model with storage of flow at intermediate nodes. That is, flow entering a node can be held back for some time before it is sent onward. To rule out a deficit at any node, we require that the total inflow must upper bound the total outflow for any point in time and any node other than the source:

$$D^-_{v,i}(\xi) - D^+_{v,i}(\xi) \ge 0 \qquad \text{for all} \quad \xi \in [0, T), i \in K, \quad \text{and} \quad v \in V\backslash\{s_i\}. \tag{2}$$

Moreover, flow must not remain in any node other than the sinks at time T. Therefore, we require that equality holds in (2) for every $i \in K$, $v \in V\backslash\{s_i, t_i\}$, at time $\xi = T$. The flow over time f satisfies the multicommodity demands if

$$D^-_{t_i,i}(T) - D^+_{t_i,i}(T) = d_i \qquad \text{for any commodity} \quad i \in K. \tag{3}$$

The cost of a flow over time f is defined as $c(f) := \sum_{e\in E} \sum_{i\in K} c_{e,i} \int_0^T f_{e,i}(\theta)\, d\theta$.

2.3. *Time-Expanded Graphs.* Many flow over time problems can be solved by static flow algorithms in time-expanded graphs [7], [8]. Given a graph $G = (V, E)$ with integral transit times on the arcs and an integral time horizon T, the T-*time-expanded graph* of G, denoted G^T, is obtained by creating T copies of V, labeled V_0 through V_{T-1}, with the θth copy of node v denoted $v(\theta)$, $\theta = 0, \ldots, T - 1$. For every arc $e = (v, w) \in E$ and $\theta = 0, \ldots, T - 1 - \tau_e$, there is an arc $e(\theta)$ from $v(\theta)$ to $w(\theta + \tau_e)$ with the same capacity and costs as arc e. In addition, there is an infinite capacity *holdover arc* from $v(\theta)$ to $v(\theta + 1)$, for all $v \in V$ and $\theta = 0, \ldots, T - 2$, which models the possibility to hold flow at node v during the time interval $[\theta, \theta + 1)$.

Any static flow in this time-expanded network corresponds to a flow over time of equal cost: interpret the flow on arc $e(\theta)$ as the flow through arc $e = (v, w)$ that starts at node v in the time interval $[\theta, \theta + 1)$. Similarly, any flow over time completing by time

T corresponds to a static flow in G^T of the same value and cost obtained by mapping the total flow starting on e in time interval $[\theta, \theta+1)$ to flow on arc $e(\theta)$. Thus, we may solve a flow over time problem by solving the corresponding static flow problem in the time-expanded network.

One drawback of this approach is that the size of G^T depends linearly on T, so that if T is not bounded by a polynomial in the input size, this is not a polynomial-time method. However, the following useful observation can be found in [4] and [6]: If all transit times are multiples of some large number $\Delta > 0$, then instead of using the T-time-expanded graph, we may rescale time and use a Δ-*condensed time-expanded graph* that contains only $\lceil T/\Delta \rceil$ copies of V. Since in this setting every arc corresponds to a time interval of length Δ, capacities are multiplied by Δ. For more details we refer to [4] and [6].

2.4. *Flows over Time with Inflow-Dependent Transit Times.* In the original setting of flows over time discussed above, it is assumed that transit times are fixed throughout, so that flow on arc e progresses at a uniform speed. In the following we will consider the more general model of *inflow-dependent* transit times. Here, the transit time of an arc may vary with the current amount of flow using this arc. Each arc e has an associated non-negative transit time function τ_e which determines the time it takes for the flow to traverse arc e. In order to define a (*multicommodity*) *flow with inflow-dependent transit times* $(\tau_e)_{e\in E}$ and time horizon T, we generalize the requirements for the flows over time as defined above: the flow of commodity i entering arc e at time θ at rate $f_{e,i}(\theta)$ arrives at head(e) at time $\theta + \tau_e(f_e(\theta))$. In order to obey the time horizon T, we require for all $e \in E$ and $\theta \in [0, T)$ that $f_e(\theta) > 0$ must imply $\theta + \tau_e(f_e(\theta)) < T$. The total inflow of commodity $i \in K$ into node v until time $\xi \in [0, T]$ is now given by

$$D^-_{v,i}(\xi) := \sum_{e\in\delta^-(v)} \int\limits_{\substack{\theta\geq 0:\\ \theta+\tau_e(f_e(\theta))\leq\xi}} f_{e,i}(\theta)\, d\theta.$$

The definition of the total outflow (1) remains unchanged. With these slight adjustments, the flow conservation constraints (2) and the demand constraints (3) can be directly adopted.

We will later need the following simple observation which follows from the fact that flow can be stored at intermediate nodes.

OBSERVATION 1. *For every arc $e \in E$, let $\tau_e\colon [0, u_e] \to \mathbb{R}^+$ and $\tau'_e\colon [0, u_e] \to \mathbb{R}^+$ be transit time functions on arc e such that $\tau'_e(x) \leq \tau_e(x)$ for all $x \in [0, u_e]$. Then a flow over time with inflow-dependent transit times $(\tau_e)_{e\in E}$ and time horizon T also yields a flow over time with inflow-dependent transit times $(\tau'_e)_{e\in E}$ and time horizon T.*

The *quickest* (*multicommodity*) *flow problem with costs* is to find a (multicommodity) flow over time with inflow-dependent transit times in G that satisfies the demands within minimal time T at a cost which is bounded from above by a given *budget* C.

3. The Bow Graph. In [19] a so-called *bow graph* is introduced in order to attack inflow-dependent transit times. The bow graph is an expansion of the original graph according to the given transit time functions. Every arc with inflow-dependent transit time is replaced by a bunch of arcs with fixed transit times. On the one hand, it is shown in [19] that a flow over time with inflow-dependent transit times in the original graph can be regarded as a flow over time (with constant transit times) in the bow graph. Hence, the bow graph serves as a relaxation of inflow-dependent transit times. On the other hand, it is shown that a quickest s–t-flow (with constant transit times) in the bow graph can be turned into an s–t-flow over time with inflow-dependent transit times in the original graph while losing only a factor of 2 on the optimal time horizon of a quickest s–t-flow in G.

In this section we will define a bow graph that is very similar to the one defined in [19]. However, it turns out that, in order to approximate the quickest (inflow-dependent) *multicommodity* flow in G, one must not allow arbitrary flows over time in the bow graph as a relaxation to inflow-dependent transit times. Instead, we will restrict to a certain subclass of flows over time in the bow graph in order to get a stronger relaxation of the problem under consideration.

Let us for the moment assume that all transit time functions are piecewise constant, non-decreasing, and left-continuous. This transit time function of arc e is denoted by τ_e^s. It is given by breakpoints $0 = x_0 < x_1 < \ldots < x_\ell$ and corresponding transit times $\tau_1 < \ldots < \tau_\ell$. Flow entering arc e at rate $x \in (x_{i-1}, x_i]$[6] needs τ_i time to traverse arc e. Later we will use the fact that general transit time functions can be approximated by such step functions within arbitrary precision.

The bow graph, denoted $G^{\mathrm{B}} = (V^{\mathrm{B}}, E^{\mathrm{B}})$, is defined on the same node set as G, i.e., $V^{\mathrm{B}} := V$, and is obtained by creating several copies of an arc, one for every possible transit time on this arc. Thus, arc e is replaced by ℓ parallel *bow arcs* $a_1, \ldots, a_\ell$. The transit time of bow arc a_i is τ_i and its capacity is x_i, $i = 1, \ldots, \ell$. We will denote the set of bow arcs corresponding to arc $e \in E$ by E_e^{B}, and refer to E_e^{B} as the *expansion* of arc e. The cost coefficients of every arc $a \in E_e^{\mathrm{B}}$ are identical to those of e, i.e., $c_{a,i} := c_{e,i}$, for $i \in K$. For every arc $a \in E_e^{\mathrm{B}}$, let $e(a)$ denote the corresponding original arc e.

3.1. *A Relaxation of Inflow-Dependent Transit Times.* We will now discuss the relationship between flows over time with inflow-dependent transit times in G and flows over time in the bow graph G^{B}. Any flow over time f in G with inflow-dependent transit times $(\tau_e^s)_{e\in E}$ and time horizon T can be interpreted as a flow over time f^{B} in G^{B} (with constant transit times) with the same time horizon T: if flow is entering arc $e \in E$ at time θ with flow rate $f_e(\theta)$, then, in the bow graph, this flow is sent onto the bow arc $a \in E_e^{\mathrm{B}}$ representing the transit time $\tau_e^s(f_e(\theta))$.

Unfortunately, an arbitrary flow over time f^{B} in G^{B} does not correspond to a flow over time f with inflow-dependent transit times $(\tau_e^s)_{e\in E}$ in G. In addition, we have to require the following property: for every original arc $e \in E$ and at every point in time θ, the flow f^{B} sends flow into at most one bow arc $a \in E_e^{\mathrm{B}}$. This property ensures that flow units entering arc e at the same time θ travel through e at the same pace. A flow over time in G^{B} fulfilling this property is called *inflow-preserving*.

[6] Note that the interval is left exclusive and right inclusive due to τ_e^s being left-continuous.

OBSERVATION 2. *Every inflow-preserving flow over time f^{B} in G^{B} with time horizon T corresponds to a flow over time f in G with inflow-dependent transit times $(\tau_e^s)_{e\in E}$ and time horizon T, and vice versa.*

Notice that the set of inflow-preserving flows over time is not convex. In particular, it is difficult to compute inflow-preserving flows directly. Therefore, we also consider a relaxed notion which can be interpreted as a convexification of inflow-preserving flows: For any arc $a \in E^{\mathrm{B}}$, let $\lambda_a(\theta) := f_a^{\mathrm{B}}(\theta)/u_a$ denote the *per capacity inflow rate* into arc a at time θ. Then a flow over time f^{B} in G^{B} with time horizon T is called *weakly inflow-preserving* if $\sum_{a\in E_e^{\mathrm{B}}} \lambda_a(\theta) \le 1$ for all $e \in E$ and $\theta \in [0, T)$. Since every inflow-preserving flow over time is also weakly inflow-preserving, it follows from Observations 1 and 2 that weakly inflow-preserving flows over time in G^{B} constitute a relaxation of flows over time with inflow-dependent transit times in G:

OBSERVATION 3. *For every arc $e \in E$, let $\tau_e^s : [0, u_e] \to \mathbb{R}^+$ and $\tau_e : [0, u_e] \to \mathbb{R}^+$ be transit time functions on arc e such that τ_e^s is a step function with $\tau_e^s(x) \le \tau_e(x)$ for all $x \in [0, u_e]$. Then every flow over time with inflow-dependent transit times $(\tau_e)_{e\in E}$ and time horizon T in G yields a* (*weakly*) *inflow-preserving flow over time with time horizon T in G^{B}.*

The basic idea of the approximation algorithms presented in this paper is to compute weakly inflow-preserving flows over time in an appropriate bow graph and turn these into flows over time in G with inflow-dependent transit times. The following lemma and its corollary make this approach work. Consider the expansion of a single arc $e \in E$ to bow arcs $E_e^{\mathrm{B}} = \{a_1, \dots, a_\ell\}$.

LEMMA 1. *Let f^{B} be a weakly inflow-preserving flow over time with time horizon T in E_e^{B} and $\delta > 0$. Then f^{B} can be turned into an inflow-preserving flow over time $\hat{f}^{\mathrm{B}}$ in E_e^{B} such that every* (*infinitesimal*) *unit of flow in $\hat{f}^{\mathrm{B}}$ reaches head*(*e*) *at most δ time units later than it does in f^{B}.*

PROOF. For every bow arc a_i, $i = 1, \dots, \ell$, we set up a buffer b_i in tail(e) for temporary storage of flow. The buffer b_i is collecting all flow in f^{B} which is about to be shipped through bow arc a_i. It can output this flow in a first-in-first-out manner, i.e., flow units must enter and leave the buffer in the same order. Buffer b_i has only two output modes. Either it is *closed* and no flow is leaving the buffer, or it is *open* and flow is leaving the buffer at constant rate u_{a_i}, immediately entering arc a_i. In our modified solution $\hat{f}^{\mathrm{B}}$, at every point in time at most one of the buffers b_i, $i = 1, \dots, \ell$, will be open. This guarantees that $\hat{f}^{\mathrm{B}}$ is inflow-preserving.

As above, let $\lambda_a(\theta) := f_a^{\mathrm{B}}(\theta)/u_a$ be the per capacity inflow rate of f^{B} on arc $a \in E_e^{\mathrm{B}}$ at time θ. We partition the time horizon into intervals of length $\tilde{\delta}$, where $\tilde{\delta} := \delta/2$. Let $\lambda_{a,j}$ be the average per capacity inflow rate on arc $a \in E_e^{\mathrm{B}}$ during time interval $[(j-1)\tilde{\delta}, j\tilde{\delta})$, i.e.,

$$\lambda_{a,j} := \frac{1}{\tilde{\delta}} \int_{(j-1)\tilde{\delta}}^{j\tilde{\delta}} \lambda_a(\theta)\, d\theta, \qquad \text{for} \quad j = 1, \dots, \lceil T/\tilde{\delta} \rceil.$$

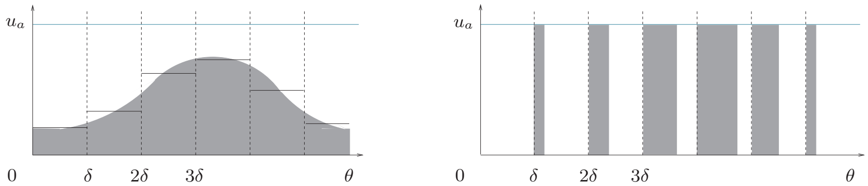


Fig. 1. Original flow rate on bow arc a (on the left) and modified flow rate produced by buffering in tail(a) (on the right).

We define the modified flow $\hat{f}^{\mathrm{B}}$ as follows: During the first $\tilde{\delta}$-round, all buffers are closed. During each following $\tilde{\delta}$-round, we open the buffers in a "round robin" fashion. More precisely, during time interval $[j\tilde{\delta}, (j+1)\tilde{\delta})$, we first open buffer b_1 for $\lambda_{a_1,j}\tilde{\delta}$ time, then buffer b_2 for $\lambda_{a_2,j}\tilde{\delta}$ time, and so on. Since f^{B} is weakly inflow-preserving, $\sum_{i=1}^{\ell} \lambda_{a_i,j} \leq 1$ holds and the last buffer is closed again before the end of this $\tilde{\delta}$-round. Figure 1 illustrates how the buffer changes the original inflow rate of a single bow arc a.

We show that the buffers are never empty while they are open. Consider bow arc a_i. During the interval $[(j-1)\tilde{\delta}, j\tilde{\delta})$, the flow f^{B} sends $\tilde{\delta}\lambda_{a_i,j}u_{a_i}$ units of flow into bow arc a_i. This is exactly the amount of flow that the corresponding buffer b_i is sending out during the succeeding interval $[j\tilde{\delta}, (j+1)\tilde{\delta})$. Hence buffer b_i is never emptied and, in particular, every unit of flow is delayed for at most $2\tilde{\delta} = \delta$ time. Note that throughout these modifications no flow is rerouted. We only make use of storage in nodes. Therefore, the cost of f^{B} remains unchanged. □

For $\delta > 0$, a flow over time f^{B} in G^{B} is called *δ-resting* if, for every node $v \in V \backslash \{s_1, \ldots, s_k\}$, all flow arriving at v is stored there for at least δ time units before it moves on. A weakly inflow-preserving flow over time f^{B} in G^{B} which is δ-resting can be easily interpreted as an inflow-preserving flow over time $\hat{f}^{\mathrm{B}}$: Consider a single arc $e \in E$ and its expansion E_e^{B}. Applying Lemma 1, the flow over time f^{B} restricted to E_e^{B} can be modified to an inflow-preserving flow over time such that every unit of flow is delayed by at most δ. The resting property of f^{B} makes up for this delay and ensures that every such flow unit can continue its way on time. Applying Observation 2, the flow $\hat{f}^{\mathrm{B}}$ can then be interpreted as a flow over time f in G with inflow-dependent transit times $(\tau_e^s)_{e \in E}$.

COROLLARY 1. *Let f^{B} be a weakly inflow-preserving flow over time in G^{B} with time horizon T which is δ-resting. Then f^{B} can be turned into a flow over time f in G with inflow-dependent transit times $(\tau_e^s)_{e \in E}$ and with the same time horizon and the same cost as f^{B}. Moreover, the flow over time f is given by piecewise constant functions $(f_e)_{e \in E}$ such that the number of breakpoints of f_e is bounded by $2|E_e^{\mathrm{B}}|\lceil T/\delta \rceil$.*

4. A $(2+\varepsilon)$-Approximation Algorithm for Quickest Flows. In this section we present a fairly simple $(2+\varepsilon)$-approximation algorithm for the quickest multicommodity flow problem with inflow-dependent transit times. The algorithm consists of the following three main steps. First, the original transit times $(\tau_e)_{e\in E}$ are replaced by lower step functions $(\tau_e^s)_{e\in E}$ and the corresponding bow graph G^{B} is constructed. Then an appropriately modified version of the $(2+\varepsilon)$-approximation algorithm presented in [4] and [6] is applied yielding a weakly inflow-preserving flow over time in G^{B}. Finally, the output is turned into a feasible solution to the original problem.

The bow graph G^{B} is defined in the first step according to step functions fulfilling the requirements stated in the following observation. We will later specify the parameters $\delta, \eta > 0$ such that the size of the resulting bow graph is polynomial in the input size and $1/\varepsilon$.

OBSERVATION 4. *Let $\delta, \eta > 0$. For every non-negative, non-decreasing, and left-continuous function τ: $[0, u] \to \mathbb{R}^+$, there exists a step function τ^s: $[0, u] \to \mathbb{R}^+$, with*

(i) $\tau^s(x) \le \tau(x) \le (1+\eta)\tau^s(x) + \delta$ *for every* $x \in [0, u]$, *and*
(ii) *the number of breakpoints of* τ^s *is bounded by* $\lceil \log_{1+\eta}(\tau(u)/\delta)\rceil + 1$.

Note that the non-decreasing function τ is left-continuous if and only if $\sup\{\tau(x')|x' < x\} = \tau(x)$. Left-continuity of τ is important, since the step function τ^s needs to be left-continuous in order to construct bow graphs as described at the beginning of Section 3.

4.1. $(2+\varepsilon)$-*Approximate Quickest Weakly Inflow-Preserving Flows*. Fleischer and Skutella [4], [6] propose a $(2+\varepsilon)$-approximation algorithm for the quickest multicommodity flow problem with bounded cost and constant transit times. The method is based on an approximate length-bounded static flow computation. The same approach can be applied to the problem of finding a quickest weakly inflow-preserving multicommodity flow over time with bounded cost in the bow graph.

Let f^{B} be an optimal solution to this problem with minimal time horizon T. As suggested in [4] and [6], we consider the static multicommodity flow x^{B} in G^{B} which results from averaging the flow f^{B} over the time interval $[0, T)$, i.e.,

$$x^{\mathrm{B}}_{a,i} := \frac{1}{T}\int_0^T f^{\mathrm{B}}_{a,i}(\theta)\,d\theta \qquad \text{for all} \quad a \in E^{\mathrm{B}} \quad \text{and} \quad i \in K.$$

As proven in [4] and [6], this static flow

(i) satisfies a fraction of $1/T$ of the demands covered by the flow over time f^{B},
(ii) has cost $c(x^{\mathrm{B}}) = c(f^{\mathrm{B}})/T$, and
(iii) is T-length-bounded.

The latter property means that the flow of every commodity $i \in K$ can be decomposed into a sum of flows on s_i–t_i-paths such that the length $\tau(P) := \sum_{a\in P}\tau_a$ of any such path P is at most T. Since f^{B} is weakly inflow-preserving, so is x^{B}, i.e., its *per capacity flow values* $\lambda_a := x^{\mathrm{B}}_a/u_a$, $a \in E^{\mathrm{B}}$, satisfy

(iv) $\sum_{a\in E_e^{\mathrm{B}}} \lambda_a \le 1$ for every arc $e \in E$.

Any static flow x in G^{B} meeting requirements (i)–(iv) can be turned into a weakly inflow-preserving flow over time g in G^{B} meeting the same demands at the same cost as f^{B}

within time $2T$: Send flow into every s_i–t_i-path P given by the length-bounded path decomposition of x at the corresponding flow rate $x_{P,i}$ for exactly T time units; wait for at most another T time units until all flow has arrived at its destination. Since $g_a(\theta)/u_a$ is always upper-bounded by x_a/u_a, it follows from property (iv) that g is weakly inflow preserving. Thus, g is a 2-approximate solution to the problem under consideration.

Unfortunately, computing the T-length-bounded flow x is NP-hard, even for the special case of a single commodity [13]. Yet, as discussed in [4] and [6], the T-length-bounded multicommodity flow problem can be approximated within arbitrary precision in polynomial time by slightly relaxing the length bound T. We generalize this observation to length-bounded, weakly inflow-preserving flows.

Let $\mathcal{P}_i^T$ be the set of all s_i–t_i-paths in G^{B} whose transit times are bounded from above by T. Finding a static flow satisfying (i)–(iv) is equivalent to solving the linear program shown in Figure 2.

Notice that the separation problem for the dual can be formulated as a length-bounded shortest path problem: find a shortest s_i–t_i-path P with respect to the arc weights $p_{e(a)}/u_a + c_{a,i}$ whose length $\tau(P)$ is at most T, i.e., $P \in \mathcal{P}_i^T$. Using exactly the same argument as in [4] and [6] the next lemma follows from the fact that the length-bounded shortest path problem can be approximated with arbitrary precision; see, e.g., [14] and [21].

The primal LP:

$$\begin{array}{lll}
\min & \displaystyle\sum_{i \in K} \sum_{P \in \mathcal{P}_i^T} c_i(P) x_{P,i} & \\
\text{s.t.} & \displaystyle\sum_{P \in \mathcal{P}_i^T} x_{P,i} \geq \frac{d_i}{T} & \text{for all} \quad i \in K, \\
& \displaystyle\sum_{a \in E_e^B} \frac{1}{u_a} \left(\sum_{i \in K} \sum_{\substack{P \in \mathcal{P}_i^T: \\ a \in P}} x_{P,i} \right) \leq 1 & \text{for all} \quad e \in E, \\
& x_{P,i} \geq 0 & \text{for all} \quad i \in K,\, P \in \mathcal{P}_i^T.
\end{array}$$

The dual LP:

$$\begin{array}{lll}
\max & \displaystyle\sum_{i \in K} \left(\frac{d_i}{T} \right) z_i - \sum_{e \in E} p_e & \\
\text{s.t.} & \displaystyle\sum_{a \in P} \left(\frac{p_{e(a)}}{u_a + c_{a,i}} \right) \geq z_i & \text{for all} \quad i \in K,\, P \in \mathcal{P}_i^T, \\
& z_i,\, p_e \geq 0 & \text{for all} \quad i \in K, e \in E.
\end{array}$$

Fig. 2. An LP formulation for the problem of finding a weakly inflow-preserving, T-length-bounded static flow.

LEMMA 2. *Assume that there exists a weakly inflow-preserving multicommodity flow over time with time horizon T and cost at most C. Then, for every $\varepsilon > 0$, a weakly inflow-preserving multicommodity flow over time with time horizon at most $(2+\varepsilon)T$ and cost at most C can be computed in time polynomial in the input size and $1/\varepsilon$.*

Lower bounds. If all transit time functions τ_e are constant, the $(2+\varepsilon)$-approximation algorithm in Lemma 2 and the one presented in [4] and [6] basically coincide. In [4] and [6] an example is given which shows that the performance guarantee of both algorithms is not better than 2. More precisely, for every $k \in \mathbb{N}$, a k-commodity problem is defined for which the algorithm has a performance ratio of $(2k-1)/k$.

The following instance shows that even in the single source, single sink case, the approximation ratio of the discussed algorithm cannot be better than 4/3. The example consists of a single arc $e = (s, t)$. The transit time of e is 0 if flow is entering at rate $x_e \leq 1$, and it is 1 if flow is entering at rate $x_e \in (1, 2]$. We want to send 2 units of flow from s to t as quickly as possible. A quickest weakly inflow-preserving flow finishes within $T = \frac{3}{2}$ simply by sending flow at rate 2 during the time interval $[0, \frac{1}{2})$, and at rate 1 during the time interval $[\frac{1}{2}, \frac{3}{2})$. Note that this flow is even inflow-preserving.

A weakly inflow-preserving flow over time f^{B} which is generated from a path decomposition of a static flow as described above needs at least 2 time units. To see this, consider the corresponding bow graph G^{B} consisting of two parallel arcs a_1 and a_2, where a_1 has transit time 0 and capacity 1, and a_2 has transit time 1 and capacity 2. Let λ_i be the per capacity flow rate of f^{B} on a_i. Then, for $T \geq 1$, the flow f^{B} manages to send $\lambda_1 T + 2\lambda_2(T-1)$ flow units from s to t within time T. It is easily checked that f^{B} needs at least time $T = 2$ to satisfy the demand.

4.2. $(2+\varepsilon)$-*Approximate Quickest Flows with Inflow-Dependent Transit Times.* So far, we have presented an algorithm to compute a $(2+\varepsilon)$-approximate solution to the quickest multicommodity flow problem in the relaxed model of weakly inflow-preserving flows over time. Such a solution has a simple structure, namely it is generated from a path decomposition of a static flow in the bow graph. We will use this property to turn such a flow into a solution to the original problem. Throughout this modification we will make sure that the time horizon only increases by a small factor.

Let f^{B} be a weakly inflow-preserving multicommodity flow over time with time horizon T^{B} in G^{B}, which is generated from a static flow x^{B} as described in the last section. In particular, x^{B} is weakly inflow-preserving and has a length-bounded path decomposition. Let $\mathcal{P}_i$ denote the set of s_i–t_i-paths from the length-bounded path decomposition of x^{B} and $\mathcal{P} := \bigcup_{i=1}^{k} \mathcal{P}_i$.

LEMMA 3. *The flow over time f^{B} can be turned into a flow over time f in G with inflow-dependent transit times $(\tau_e)_{e\in E}$ and time horizon T, where T is bounded from above by $(1+\eta)T^{\mathrm{B}} + 2n\delta$.*

PROOF. We increase transit times in G^{B} in order to emulate the original transit times $(\tau_e)_{e\in E}$. For every arc $a \in E^{\mathrm{B}}$, let $\tilde{\tau}_a := (1+\eta)\tau_a + \delta$ be the new transit time along a. Note that this corresponds to constructing the bow graph according to step functions $(\tilde{\tau}_e^s)_{e\in E}$, where $\tilde{\tau}^s_e(x) := (1+\eta)\tau^s_e(x)+\delta$ for every $x \in [0, u_e]$. Consider a path $P \in \mathcal{P}$. The flow f^{B} sends flow at constant rate x_P into P for a certain time period. Before increasing transit times, flow traveling along P needed $\tau(P) := \sum_{a \in P} \tau_a$ time to reach its destination. After the increase, this time goes up to $\tilde{\tau}(P) := \sum_{a \in P} \tilde{\tau}_a \le (1+\eta)\tau(P) + n\delta$. Since $\tau(P)$ is bounded from above by T^{B}, the transit time of every unit of flow increases by at most $\eta T^{\mathrm{B}} + n\delta$.

We repeat this procedure, but this time we increase the transit time of every arc $a \in E^{\mathrm{B}}$ by another additive factor of δ. This way, we obtain a weakly inflow-preserving flow over time $\hat{f}^{\mathrm{B}}$ in the bow graph constructed with respect to transit times $(\tilde{\tau}^s_e)_{e \in E}$ which is δ-resting and whose time horizon is bounded by $(1+\eta)T^{\mathrm{B}} + 2n\delta$. Notice that throughout these modifications no flow is rerouted. We only make use of storage in nodes. Therefore, the cost of f^{B} remains unchanged. Applying Corollary 1, this yields a flow over time f with inflow-dependent transit times $(\tilde{\tau}^s_e)_{e \in E}$ in G. Observation 1 implies that f can be interpreted as a flow over time with inflow-dependent transit times $(\tau_e)_{e \in E}$ in G which concludes the proof. □

We are now ready to state the main result of this section.

THEOREM 1. *For the quickest multicommodity flow problem with inflow-dependent transit times and bounded cost, there exists a polynomial time algorithm that, for any* $\varepsilon > 0$, *finds a solution of the same cost as optimal with time horizon at most* $2 + \varepsilon$ *times the optimal time horizon* T^*.

PROOF. We can compute in polynomial time a lower bound L on T^* such that $L \le T^* \le p(n)kL$, for some polynomial p. Namely, it is proven in [19] that for every commodity i, a lower bound L_i on the optimal time horizon T_i for sending commodity i such that $L_i \le T_i \le p(n)L_i$ can be computed in polynomial time. Setting $L := \max_i L_i$ yields the desired bound.

We fix η to $\varepsilon/8$ and δ to $\varepsilon L/(12n)$. For every arc $e \in E$, we pick lower step functions according to Observation 4(i). As already observed in [19], the number of breakpoints of τ^s_e is then in $O(\log(n/\varepsilon)/\varepsilon)$ and thus polynomially bounded. The latter is a direct consequence of Observation 4(ii) and the fact that without loss of generality we can set the capacity of every arc $e \in E$ to $u'_e := \max\{x \in [0, u_e] | \tau_e(x) \le p(n)kL\}$. We then construct the bow graph G^{B} with respect to these step functions. Because of the relaxation property of G^{B} (see Observation 3), the time horizon T^{B} of a quickest weakly inflow-preserving flow in G^{B} is a lower bound on T^*. If $T^{\mathrm{B}} \le L$, then using Lemma 2 with $T = L$ we can compute a weakly inflow-preserving multicommodity flow over time with time horizon at most $(2 + \varepsilon/4)L \le (2 + \varepsilon/4)T^*$. Otherwise $L \le T^{\mathrm{B}} \le p(n)kL$ holds. Using geometric mean binary search together with Lemma 2, we can compute a weakly inflow-preserving multicommodity flow over time with time horizon T such that $T \le (2 + \varepsilon/4)T^{\mathrm{B}} \le (2 + \varepsilon/4)T^*$.

Applying Lemma 3, this flow over time can be turned into a flow over time f with inflow-dependent transit times in G. Its time horizon is bounded by $(1+\eta)(2+\varepsilon/4)T^* + 2n\delta = (1 + \varepsilon/8)(2 + \varepsilon/4)T^* + \varepsilon/6L \le (2 + \varepsilon)T^*$. Recall that f is given by piecewise constant functions $(f_e)_{e \in E}$. Corollary 1 implies that the number of breakpoints of each such function is indeed polynomial in the input size and $1/\varepsilon$. □

We have shown in this section that a $(2+\varepsilon)$-approximation algorithm for the quickest multicommodity flow problem can be derived using length-bounded static flows in a suitable bow graph. The examples at the end of Section 4.1 show that the performance ratio of the described algorithm is not better than 2 in the multicommodity case, and not better than 4/3 in the single commodity case.

5. An FPTAS for Quickest Flows. In this section we present an FPTAS for the quickest multicommodity flow problem with inflow-dependent transit times and bounded cost. We use ideas similar to the ones employed in [5] and [6] for the problem with fixed transit times. The FPTAS is based on a static weakly inflow-preserving flow computation in a condensed time-expanded bow graph.

5.1. *Preliminary Definitions.* To state our algorithm and prove its correctness, we define the following three bow graphs, which are derived from $G = (V, E)$ with transit time functions $(\tau_e)_{e\in E}$, given a time horizon T and a small constant $\varepsilon > 0$.

- $G^{\downarrow}$: the *lower bow graph* is constructed from the lower step functions $\tau_e^{\downarrow}(x) := \lfloor \tau_e(x)/\Delta \rfloor \Delta$, for $e \in E$, $x \in [0, u_e]$, where $\Delta := \varepsilon^2 T/n$ (we always assume that n/ε^2 is integral such that T is a multiple of Δ). That is, $\tau_e(x)$ is rounded down to the nearest multiple of Δ. By the choice of Δ, the size of $G^{\downarrow}$ is polynomially bounded since we can delete all arcs with transit times greater than T.
- $G^{\uparrow}$: the *upper bow graph* is constructed from $G^{\downarrow}$ by lengthening the transit time of each arc by Δ. The corresponding transit time step functions are given by $\tau_e^{\uparrow}(x) := \tau_e^{\downarrow}(x) + \Delta$, for $e \in E$, $x \in [0, u_e]$.
- $G^{\Uparrow}$: the 2Δ*-lengthened bow graph* is constructed analogously to $G^{\uparrow}$ except that the transit times are lengthened by 2Δ, i.e., $\tau_e^{\Uparrow}(x) := \tau_e^{\downarrow}(x) + 2\Delta$, for $e \in E$, $x \in [0, u_e]$.

Let the *fan graph* $G^{\mathrm{F}} = (V^F, E^F)$ be the Δ-condensed time-expansion of $G^{\Uparrow}$ for time horizon T (see Section 2.3). Each arc $e = (v, w) \in E$ is represented in the bow graph $G^{\Uparrow}$ by its expansion $E_e^{\Uparrow}$. Thus, the fan graph contains, for each time $\theta \in S := \{0, \Delta, \dots, T - \Delta\}$, a "*fan*" of arcs

$$E_e^F(\theta) := \{a(\theta) \mid a \in E_e^{\Uparrow},\ \theta + \tau_a^{\Uparrow} \in S\},$$

where $a(\theta) = (v(\theta), w(\theta + \tau_a^{\Uparrow}))$. Furthermore, there are *holdover arcs* $(v(\theta), v(\theta+\Delta))$ of infinite capacity to simulate intermediate storage at nodes, for all $v \in V$.

For a static flow x in G^{F} and $\theta \in S$, we define $\lambda_a(\theta) := x_{a(\theta)}/u_{a(\theta)}$ to be the per capacity inflow value on arc $a(\theta) \in E^F$. With these definitions, the concept of weakly inflow-preserving flows directly carries over to static flows x in G^{F}, i.e., x is weakly inflow-preserving if

$$(4) \qquad \sum_{a \in E_e^{\Uparrow}} \lambda_a(\theta) \le 1 \qquad \text{for all} \quad e \in E \quad \text{and} \quad \theta \in S.$$

Moreover, the problem of computing a weakly inflow-preserving static flow in G^{F} can easily be formulated as a linear program. Take a standard network flow formulation and add extra constraints (4). In particular, such a flow can be computed in polynomial time. Note that any weakly inflow-preserving *static flow* in G^{F} corresponds to a weakly inflow-preserving *flow over time* in $G^{\Uparrow}$, and vice versa (see Section 2.3).

5.2. *The Algorithm and Its Running Time.* Let T^* denote the time horizon of a quickest flow with inflow-dependent transit times in G. We can now give an overview of our algorithm which yields a flow over time with time horizon at most $(1 + O(\varepsilon))T^*$, for any given $\varepsilon > 0$. In the following, let $\bar{T} := (1 + 6\varepsilon)T^*$.

FPTAS

(1) Guess T such that $\bar{T} \leq T \leq (1 + \varepsilon)\bar{T}$ holds (binary search).
(2) Construct the fan graph G^{F} with respect to T. Compute a weakly inflow-preserving *static* multicommodity flow in G^{F} satisfying all demands at cost bounded by the budget C.
(3) Interpret this static flow as a weakly inflow-preserving *flow over time* in $G^{\Uparrow}$. Modify this flow to make it inflow-preserving in $G^{\uparrow}$ and, from this, derive a flow over time in G with inflow-dependent transit times and time horizon at most T.

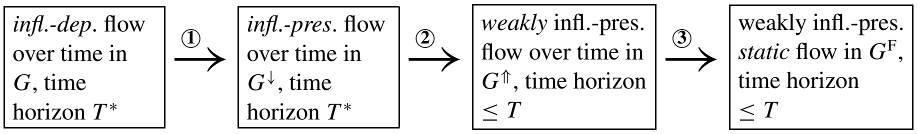
We proceed as follows: First we discuss issues related to the running time of the algorithm and detail how step 3 is implemented. Then, in Section 5.3, we prove that a static flow in G^{F} with the properties claimed in step 2 actually exists.

Using the $(2 + \varepsilon)$-approximation from Section 4, one can obtain upper and lower bounds on $\bar{T} = (1 + 6\varepsilon)T^*$ within a constant factor of each other. Thus, the estimate T can be found within $\mathcal{O}(\log(1/\varepsilon))$ geometric mean binary search steps. The fan graph G^{F} constructed in step 2 contains n/ε^2 time layers and thus n^2/ε^2 nodes and $\mathcal{O}(mn^2/\varepsilon^4)$ arcs.[7] Therefore, the static flow in G^{F} can be computed in time polynomial in n, m, and $1/\varepsilon$. We now go into the details of step 3. As mentioned before, interpreting the static flow in G^{F} as a weakly inflow-preserving flow over time in $G^{\Uparrow}$ is done in the canonical way, as described in Section 2.3. If we now shorten all arcs of $G^{\Uparrow}$ by Δ (which gives bow graph $G^{\uparrow}$), we obtain a weakly inflow-preserving flow over time in $G^{\uparrow}$which is Δ-resting. Applying Corollary 1, we derive an inflow-preserving flow over time in $G^{\uparrow}$. Finally, by Observation 1, we get a flow over time in G with inflow-dependent transit times $(\tau_e)_{e \in E}$ with time horizon at most $T = (1 + O(\varepsilon))T^*$. Clearly, step 3 can also be done in time polynomial in n, m, and $1/\varepsilon$. This concludes the discussion of the algorithm's running time.

5.3. *Analysis of the Algorithm.* In this section we prove that our algorithm actually is an FPTAS by showing that a feasible flow as claimed in step 2 exists. Without loss of generality, we can assume in the remainder of this section that $T = \bar{T} = (1 + 6\varepsilon)T^*$; notice that the topology of the fan graph G^{F} does not depend on the exact choice of T and that its arc capacities only increase with increasing T. Moreover, for technical reasons we assume that $\varepsilon \leq \frac{1}{6}$ in the following.

[7] Note that each fan contains at most n/ε^2 arcs, potentially one for each layer of G^{F}, and each of the m arcs in E induces n/ε^2 fans.

We transform a quickest flow in G with inflow-dependent transit times into a weakly inflow-preserving static flow in G^{F} without increasing its cost. This transformation is done in several steps which are illustrated in the following diagram. The cost of the flow remains unchanged in all three steps.


infl.-dep. flow over time in G, time horizon T^*
①
infl.-pres. flow over time in $G^{\downarrow}$, time horizon T^*
②
weakly infl.-pres. flow over time in $G^{\Uparrow}$, time horizon $\leq T$
③
weakly infl.-pres. *static* flow in G^{F}, time horizon $\leq T$

With Observation 3, step ① is easy to see. For step ③, the flow over time in $G^{\Uparrow}$ is interpreted as a static flow in G^{F} as described in Section 2.3. As discussed above, since the flow over time in $G^{\Uparrow}$ is weakly inflow-preserving, the static flow in G^{F} is weakly inflow-preserving, too.

Step ② is the most interesting but also the most intricate one. By lengthening the arcs, the flow might not remain feasible. This problem is overcome similarly to [5] and [6] by carefully averaging flow to derive an "almost feasible" flow, then subsequently sending less to obtain a feasible flow and finally increasing the time horizon to meet the demands. However, in contrast to [5] and [6], our flows must have the additional property of being weakly inflow-preserving. The proof of Lemma 5 provides a more detailed discussion of transformation ②, in which we stress why this property is preserved.

We first state a slight generalization of a lemma in [5] and [6]. It permits us to modify a feasible inflow-preserving flow such that the total amount of flow sent through the network is increased by a factor of $1+\delta$. Hereby, the time horizon and cost are increased by the same factor.

LEMMA 4. *Let $\delta > 0$. Given a (weakly) inflow-preserving flow over time f in a bow graph G^B with time horizon T satisfying demands d_i with a budget C, there exists a (weakly) inflow-preserving flow over time f' in G^B which satisfies demands $d_i' := (1+\delta)d_i$ within time $T' := (1+\delta)T$, and at cost $C' := (1+\delta)C$.*

PROOF. Simply scale time by a factor $1+\delta$, i.e., define a flow $f'(\theta) := f(\theta/(1+\delta))$. It is easy to see that f' has the desired properties. □

Lemma 5 provides details for transformation ②.

LEMMA 5. *A weakly inflow-preserving flow over time f in $G^{\downarrow}$ with time horizon T^* can be transformed into a* weakly *inflow-preserving flow over time in $G^{\Uparrow}$ with time horizon at most $T = (1+6\varepsilon)T^*$ and the same cost as f.*

PROOF. In f, every infinitesimal unit of flow describes a simple path P in $G^{\downarrow}$. Notice that cycles can be avoided by storing flow at intermediate nodes. The fact that we may assume paths to be simple will be helpful below, when bounding the total increase of the length of such a path P after lengthening all arcs.

An infinitesimal unit of flow additionally describes a "delay configuration" at the nodes of its path P. That is, if P is given by nodes $(v_0, v_1, \ldots, v_q)$, then a vector of non-negative delays $\delta = (\delta_1, \ldots, \delta_{q-1})$ specifies the amount of time δ_j for which the infinitesimal unit of flow is stored at node v_j before it continues towards node v_{j+1}. Let P^δ denote the path P with delay vector δ. Then the flow over time f can be decomposed into flows over time f_{P^δ} on paths P^δ. As suggested in [5] and [6], we average the flow along each path P^δ and thereby define a new flow over time $\hat{f}$:

$$(5) \qquad \hat{f}_{P^\delta}(\theta) := \frac{1}{\varepsilon T} \int_{\theta - \varepsilon T}^{\theta} f_{P^\delta}(\xi)\, d\xi \qquad \text{for} \quad \theta \in [0, T^* + \varepsilon T).$$

The new flow $\hat{f}$ has time horizon at most $T^* + \varepsilon T \le (1 + 2\varepsilon)T^*$. It is easy to check that $\hat{f}$ satisfies all demands, has the same cost as f, and obeys capacity and flow conservation constraints in $G^\downarrow$. We now interpret the flow $\hat{f}$ as a flow over time in the 2Δ-lengthened bow graph $G^\Uparrow$. Notice that the definition of $\hat{f}$ in (5)—as flows into paths P^δ—implies that flow conservation still holds. However, with the new transit times $\tau_a^\Uparrow = \tau_a^\downarrow + 2\Delta$ for $a \in E^\downarrow$ $(= E^\Uparrow)$, the flow $\hat{f}$ is not necessarily weakly inflow-preserving anymore. In fact, it might even violate capacity constraints. The reason is that flow on different paths using the same arc will, in general, experience different delays such that congestion might occur on that arc. Nevertheless, we will show that a simple rescaling of $\hat{f}$ results in a feasible weakly inflow-preserving flow over time in $G^\Uparrow$ that satisfies all demands at the same cost as f.

Note that every path P^δ is simple, and therefore contains at most $n - 1$ arcs. By definition of Δ, every path P^δ is lengthened by at most $n\,2\Delta \le 2\varepsilon^2 T$ in $G^\Uparrow$ compared with $G^\downarrow$. Thus the time horizon of $\hat{f}$ in $G^\Uparrow$ is bounded by $(1+2\varepsilon)T^* + 2\varepsilon^2 T \le (1+3\varepsilon)T^*$.

Consider a path P^δ of the path decomposition of f, i.e., P^δ is given by a path $P = (v_0, v_1, \ldots, v_q)$ in $G^\downarrow$ and a delay vector δ. For $a := (v_\ell, v_{\ell+1}) \in P$, we define

$$\tau^\downarrow(P^\delta, a) := \sum_{j=1}^{\ell} (\tau^\downarrow_{(v_{j-1}, v_j)} + \delta_j)$$

to be the transit time with delay δ of the subpath $(v_0, v_1, \ldots, v_\ell)$ in $G^\downarrow$. Similarly, $\tau^\Uparrow(P^\delta, a)$ denotes the transit time with delay δ of this subpath in $G^\Uparrow$. The flow in $\hat{f}$ entering arc a in $G^\Uparrow$ at time θ can be computed as follows:

$$(6) \qquad \hat{f}_a(\theta) = \sum_{P^\delta :\, a \in P} \hat{f}_{P^\delta}(\theta - \tau^\Uparrow(P^\delta, a)) \qquad \text{for all} \quad a \in E^\Uparrow.$$

With this expression for the inflow rates, we now aim at bounding the extent by which the weakly inflow preserving property is violated, i.e., how large the sum of the per capacity inflow values $\hat{\lambda}_a(\theta)$ for $a \in E_e^\Uparrow$ actually is. Applying $\tau^\downarrow(P^\delta, a) \le \tau^\Uparrow(P^\delta, a) \le \tau^\downarrow(P^\delta, a) + 2\varepsilon^2 T$ (every path is lengthened by at most $n2\Delta \le 2\varepsilon^2 T$), we obtain the following bound:

$$\begin{aligned} \sum_{a \in E_e^\Uparrow} \hat{\lambda}_a(\theta) \quad &= \quad \sum_{a \in E_e^\Uparrow} \frac{\hat{f}_a(\theta)}{u_a} \\ &\overset{(6)}{=} \quad \sum_{a \in E_e^\Uparrow} \sum_{P^\delta :\, a \in P} \frac{\hat{f}_{P^\delta}(\theta - \tau^\Uparrow(P^\delta, a))}{u_a} \end{aligned}$$

$$
\begin{aligned}
&\overset{(5)}{=} \quad \frac{1}{\varepsilon T} \sum_{a \in E_e^{\Uparrow}} \sum_{P^\delta :\, a \in P} \int_{\theta - \tau^{\Uparrow}(P^\delta, a) - \varepsilon T}^{\theta - \tau^{\Uparrow}(P^\delta, a)} \frac{f_{P^\delta}(\xi)}{u_a}\, d\xi \\
&\leq \quad \frac{1}{\varepsilon T} \sum_{a \in E_e^{\Uparrow}} \sum_{P^\delta :\, a \in P} \int_{\theta - \tau^{\downarrow}(P^\delta, a) - 2\varepsilon^2 T - \varepsilon T}^{\theta - \tau^{\downarrow}(P^\delta, a)} \frac{f_{P^\delta}(\xi)}{u_a}\, d\xi \\
&= \quad \frac{1}{\varepsilon T} \int_{\theta - 2\varepsilon^2 T - \varepsilon T}^{\theta} \sum_{a \in E_e^{\Uparrow}} \sum_{P^\delta :\, a \in P} \frac{f_{P^\delta}(\xi - \tau^{\downarrow}(P^\delta, a))}{u_a}\, d\xi \\
&\overset{\text{as in (6)}}{=} \quad \frac{1}{\varepsilon T} \int_{\theta - 2\varepsilon^2 T - \varepsilon T}^{\theta} \sum_{a \in E_e^{\Uparrow}} \frac{f_a(\xi)}{u_a}\, d\xi \\
&\leq \quad \frac{1}{\varepsilon T} \int_{\theta - 2\varepsilon^2 T - \varepsilon T}^{\theta} 1\, d\xi = 1 + 2\varepsilon.
\end{aligned}
$$

Dividing $\hat{f}$ by $(1 + 2\varepsilon)$ thus yields a feasible weakly inflow-preserving flow over time in $G^{\Uparrow}$ that satisfies a fraction of $1/(1 + 2\varepsilon)$ of all demands at cost at most $C/(1 + 2\varepsilon)$. By Lemma 4, a weakly inflow-preserving flow over time satisfying all demands can be obtained by increasing the time horizon by a factor of $(1 + 2\varepsilon)$, yielding a time horizon of at most $(1 + 2\varepsilon) \cdot (1 + 3\varepsilon)T^* \leq (1 + 6\varepsilon)T^*$. The cost of this flow is equal to the cost of f. □

The following theorem comprises the main result of this section.

THEOREM 2. *There is an FPTAS for the quickest multicommodity flow problem with inflow-dependent transit times and bounded cost.*

6. Complexity. In this section we prove the following result on the complexity of the quickest s–t-flow problem with inflow-dependent transit times.

THEOREM 3. *The quickest s–t-flow problem with inflow-dependent transit times, with or without storage of flow at intermediate nodes, is* NP-*hard in the strong sense.*

The proof uses a reduction from the well-known NP-complete problem 3-PARTITION, see [9].

3-PARTITION

Given: A set of $3n$ items, $n \in \mathbb{N}$, with associated sizes $b_1, \ldots, b_{3n} \in \mathbb{N}$, a bound $B \in \mathbb{N}$, such that each b_i satisfies $B/4 < b_i < B/2$ and such that $\sum_{i=1}^{3n} b_i = nB$.
Question: Can $\{1, \ldots, 3n\}$ be partitioned into n disjoint sets $I_1, \ldots, I_n$ such that, for $j \in \{1, \ldots, n\}$, $\sum_{i \in I_j} b_i = B$?

Given an instance of 3-PARTITION, we construct a network with inflow-dependent transit times as shown in Figure 3.

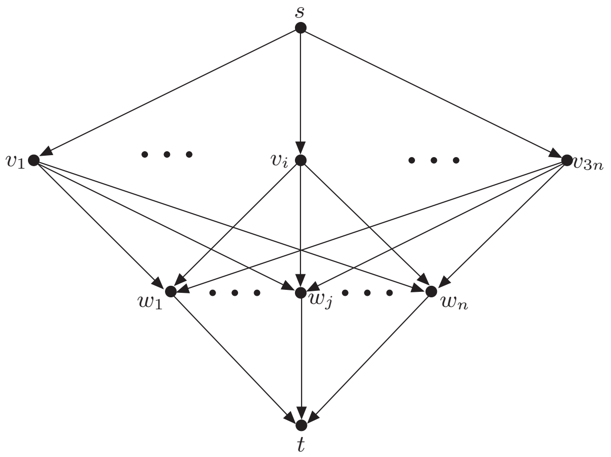


Fig. 3. Reduction of the problem 3-PARTITION to an s–t-flow over time problem with inflow-dependent transit times.

Each item b_i is represented by a node v_i, each index set I_j is represented by a node w_j. The capacities are defined as follows:

$$u((s, v_i)) := nb_i + 1, \qquad u((v_i, w_j)) := b_i + 1, \qquad u((w_j, t)) := (n+1)B + 3.$$

We define inflow-dependent transit times on (v_i, w_j) as

$$\tau_{(v_i,w_j)}(x) := \begin{cases} 0 & \text{if} \quad x \le b_i, \\ 1 & \text{else.} \end{cases}$$

All other arcs in the network have transit time zero. The task is to send $D := 2n^2B + 3n$ units of flow from s to t.

LEMMA 6. *If the underlying instance of* 3-*PARTITION is a "yes"-instance, then there exists an s–t-flow over time with inflow-dependent transit times which sends* $2n^2B + 3n$ *units of flow from s to t in time $T := 2$ without using storage of flow at intermediate nodes.*

PROOF. Given a partition $I_1, \ldots, I_n$ of $\{1, \ldots, 3n\}$ such that, for $j \in \{1, \ldots, n\}$, $\sum_{i \in I_j} b_i = B$, we define a flow over time with inflow-dependent transit times as follows. During the time interval [0, 1) we send flow at constant rate $nb_i + 1$ into arc (s, v_i), for every $i \in \{1, \ldots, 3n\}$. This flow is sent to the nodes $w_1, \ldots, w_n$ according to the following rule. We set the flow rate of arc (v_i, w_j) to $b_i + 1$, if $i \in I_j$, and to b_i, otherwise. During the time interval [1, 2) we send flow at constant rate nb_i into arc (s, v_i), for every $i \in \{1, \ldots, 3n\}$, and define the flow rate into arc (v_i, w_j), $j = 1, \ldots, n$, to be b_i. With these definitions, it is easy to see that at every point in time flow is entering a node w_j, $j = 1, \ldots, n$, at rate bounded by $(n+1)B + 3$. Thus it can be sent immediately to t using the arc (w_j, t). Obviously, flow conservation holds at every point in time and no storage in intermediate nodes is used. Moreover, all $2n^2B + 3n$ units of flow arrive in t before time 2. □

It remains to show that the existence of a flow over time f with inflow-dependent transit times of value D with time horizon at most 2 yields a feasible solution to the underlying instance of 3-PARTITION. To do this, we need to make the following reasonable assumption on f: all flow rate functions are essentially continuous, i.e., on every arc a of the given network we require that the flow rate function f_a has at most finitely many discontinuities. In the following we call a flow over time *reasonable* if it satisfies this assumption.

In f, we color every (infinitesimal) unit of flow either red or green. If it enters the network before time 1, it is colored red, else it is colored green. We denote the corresponding flows by f^{r} and f^{g}.

CLAIM 1. *For every* $\theta \in [0, 1)$, *the following properties hold*:

$$\int_0^\theta f^{\mathrm{r}}_{(s,v_i)}(\tau)\,d\tau = \theta(nb_i + 1) \qquad \textit{for all} \quad i \in \{1, \dots, 3n\}, \tag{7}$$

$$\int_1^{1+\theta} f^{\mathrm{g}}_{(v_i,w_j)}(\tau)\,d\tau = \theta b_i \qquad \textit{for all} \quad i \in \{1, \dots, 3n\}, \quad j \in \{1, \dots, n\}. \tag{8}$$

PROOF. After time 1 flow can enter an arc (v_i, w_j) at rate at most b_i since, otherwise, it cannot reach t before time 2. Thus, in total, at most n^2B units of green flow can be sent to t. Then, in order to satisfy the demand $2n^2B + 3n$, at least $n^2B + 3n$ units of red flow must leave s. Since the capacity of arc (s, v_i) is bounded by $nb_i + 1$, for $i \in \{1, \dots 3n\}$, at most $\sum_{i=1}^{3n}(nb_i + 1) = n^2B + 3n$ units of red flow can be sent in total. Hence, exactly $n^2B + 3n$ units of red flow and exactly n^2B units of green flow must travel from s to t. As a consequence, (7) and (8) must hold. □

Consider a node v_i, $i \in \{1, \dots, 3n\}$. If flow is entering an arc (v_i, w_j), $j \in \{1, \dots, n\}$, at rate at most b_i, this flow arrives in w_j instantaneously. Otherwise, this flow needs one unit of time to reach w_j and is therefore *delayed*. We now investigate how much red flow is delayed for each node v_i.

CLAIM 2. *For all* $i \in \{1, \dots, 3n\}$, *the following properties hold*:

(1) *At least* $b_i + 1$ *units of red flow passing through* v_i *are delayed*.
(2) *If exactly* $b_i + 1$ *units of red flow passing through* v_i *are delayed*, *then at almost every*[8] *point* $\theta \in [0, 1)$,

$$f^{\mathrm{r}}_{(v_i,w_j)}(\theta) \in \{b_i, b_i + 1\} \qquad \textit{for all} \quad j \in \{1, \dots, n\}.$$

PROOF. Fix $i \in \{1, \dots, 3n\}$. At most nb_i units of red flow can be sent out of v_i instantaneously during $[0, 1)$ by setting the flow rate of every arc (v_i, w_j), $j = 1, \dots, n$,

[8] The subset of $[0, 1)$ where the property fails has Lebesgue-measure zero.

to the threshold value b_i during time interval [0, 1). By (7), at least one additional unit of red "excess" flow has to be sent out of v_i by exceeding this threshold value on some of the arcs (v_i, w_j), $j = 1, \dots, n$. For every arc (v_i, w_j), $j = 1, \dots, n$, let

$$\delta_j(\theta) := \begin{cases} f^{\mathrm{r}}_{(v_i,w_j)}(\theta) - b_i & \text{if} \quad f^{\mathrm{r}}_{(v_i,w_j)}(\theta) > b_i, \\ 0 & \text{otherwise} \end{cases}$$

denote the *excess rate* of arc (v_i, w_j), then $\sum_{j=1}^{n} \int_0^1 \delta_j(\theta)\, d\theta \geq 1$ must hold. Since $0 \leq \delta_j(\theta) \leq 1$ at every point in time $\theta \in [0, 1)$,

$$b_i + \delta_j(\theta) \geq (b_i + 1)\delta_j(\theta). \tag{9}$$

Whenever the excess rate $\delta_j(\theta)$ is strictly greater than zero, not only the excess flow is delayed, but all flow entering the arc (v_i, w_j) at time θ. We conclude that the total amount of delayed flow can be lower-bounded as follows:

$$\sum_{j=1}^{n} \int_{\theta:\delta_j(\theta)>0} (b_i + \delta_j(\theta))\, d\theta \overset{(9)}{\geq} (b_i + 1) \sum_{j=1}^{n} \int_0^1 \delta_j(\theta)\, d\theta \geq b_i + 1. \tag{10}$$

This proves the first statement of the claim.

To prove the second statement, assume that exactly $b_i + 1$ units of red flow are delayed. In a first step we prove that at almost every point $\theta \in [0, 1)$, the excess rate $\delta_j(\theta)$ is either 0 or 1. By contradiction, assume that there exists $j \in \{1, \dots, n\}$ for which the property fails; let $\Theta := \{\theta \in [0, 1) \colon 0 < \delta_j(\theta) < 1\}$. For all $\theta \in \Theta$, the inequality in (9) is strict. Since $f_{(v_i,w_j)}$ has only a finite number of discontinuities, so does δ_j. Hence Θ contains a small interval where δ_j is continuous and so the first inequality in (10) must be strict, too. Thus, more than $b_i + 1$ units of flow are delayed leading to a contradiction. We conclude that at almost every point in time, for all $j \in \{1, \dots, n\}$, either $f^{\mathrm{r}}_{(v_i,w_j)}(\theta) \leq b_i$ (if $\delta_j(\theta) = 0$) or $f^{\mathrm{r}}_{(v_i,w_j)}(\theta) = b_i + 1$ (if $\delta_j(\theta) = 1$). Next assume that there exists $j \in \{1, \dots, n\}$ for which there is a set with Lebesgue-measure greater than zero where $f^{\mathrm{r}}_{(v_i,w_j)}$ is strictly less than b_i. Then the excess flow $\sum_{j=1}^{n} \int_0^1 \delta_j(\theta)\, d\theta$ has to be strictly greater than 1, implying that the second inequality in (10) is strict. Again, more than $b_i + 1$ units of flow are delayed leading to a contradiction. This proves the second statement. □

CLAIM 3. *For all $i \in \{1, \dots, 3n\}$, exactly $b_i + 1$ units of red flow passing through v_i are delayed. Moreover, all arcs (w_j, t), $j = 1, \dots, n$, are completely filled with green flow and delayed red flow during* $[1, 2)$.

PROOF. It follows from (8) that nB units of green flow must travel via the arcs (w_j, t), $j \in \{1, \dots, n\}$. Thus, due to capacity constraints, during the interval $[1, 2)$ at most another $nB + 3n$ units of delayed red flow can pass through all of the arcs (w_j, t), $j = 1, \dots, n$. It then follows from the first statement in Claim 2 that, for all $i \in \{1, \dots, 3n\}$, exactly $b_i + 1$ units of red flow passing through v_i are delayed and all arcs (w_j, t), $j = 1, \dots, n$, are completely filled with green flow and delayed red flow $[1, 2)$. □

LEMMA 7. *If a reasonable s–t-flow over time f with inflow-dependent transit times exists which sends $D = 2n^2B + 3n$ units of flow from s to t in time $T := 2$, then the underlying instance of* 3*-PARTITION is a "yes"-instance.*

PROOF. Pick a non-empty interval $(0, \mu)$ during which all flow rate functions $f_{(v_i,w_j)}$ are continuous. By Claim 3 and the second part of Claim 2, each of these flow rates must be constant, either b_i or $b_i + 1$. Claim (7) together with flow conservation implies that, for each $i \in \{1, \dots, 3n\}$, at most one arc leaving v_i has a flow rate of $b_i + 1$ during $(0, \mu)$. We define partition sets as follows: for all $j \in \{1, \dots, n\}$ let I_j be the set of items b_i for which $f_{(v_i,w_j)}(\theta) = b_i + 1$ during $(0, \mu)$. Notice that no item is contained in more than one partition set. We claim that each partition set I_j satisfies $\sum_{i \in I_j} b_i = B$. If not, there exists $j \in \{1, \dots, n\}$ such that $\sum_{i \in I_j} b_i < B$. Then less than $\mu(B + 3)$ units of delayed red flow arrive in w_j during $(1, 1 + \mu)$. Again, by (8), at most another $\mu n B$ units of green flow arrive in w_j during $(1, 1 + \mu)$, contradicting Claim 3. This concludes the proof of Lemma 7 and Theorem 3. □

Acknowledgments. We thank two anonymous referees for many helpful comments that lead to an improved presentation of this paper.

References

[1] J. E. Aronson. A survey of dynamic network flows. *Annals of Operations Research*, 20:1–66, 1989.

[2] R. E. Burkard, K. Dlaska, and B. Klinz. The quickest flow problem. *ZOR—Methods and Models of Operations Research*, 37:31–58, 1993.

[3] M. Carey and E. Subrahmanian. An approach to modelling time-varying flows on congested networks. *Transportation Research B*, 34:157–183, 2000.

[4] L. Fleischer and M. Skutella. The quickest multicommodity flow problem. In W. J. Cook and A. S. Schulz, editors, *Integer Programming and Combinatorial Optimization*, pages 36–53. Volume 2337 of Lecture Notes in Computer Science. Springer, Berlin, 2002.

[5] L. Fleischer and M. Skutella. Minimum cost flows over time without intermediate storage. In *Proceedings of the* 14*th Annual ACM–SIAM Symposium on Discrete Algorithms*, pages 66–75, Baltimore, MD, 2003.

[6] L. Fleischer and M. Skutella. Quickest flows over time. *SIAM Journal on Computing*. To appear.

[7] L. R. Ford and D. R. Fulkerson. Constructing maximal dynamic flows from static flows. *Operations Research*, 6:419–433, 1958.

[8] L. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, NJ, 1962.

[9] M. R. Garey and D. S. Johnson. *Computers and Intractability*: *A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, CA, 1979.

[10] A. Hall, S. Hippler, and M. Skutella. Multicommodity flows over time: efficient algorithms and complexity. *Theoretical Computer Science*. To appear.

[11] A. Hall, K. Langkau, and M. Skutella. An FPTAS for quickest multicommodity flows with inflow-dependent transit times. In S. Arora, K. Jansen, J. D. P. Rolim, and A. Sahai, editors, *Approximation*, *Randomization*, *and Combinatorial Optimization*, pages 71–82. Volume 2764 of Lecture Notes in Computer Science. Springer, Berlin, 2003.

[12] A. Hall and H. Schilling. Flows over time: towards a more realistic and computationally tractable model. In *Proceedings of the* 7*th Workshop on Algorithm Engineering and Experiments* (*ALENEX*), pages 55–67. ACM–SIAM, New York/Philadelphia, PA, 2005.

[13] G. Handler and I. Zang. A dual algorithm for the constrained shortest path problem. *Networks*, 10:293–310, 1980.

[14] R. Hassin. Approximation schemes for the restricted shortest path problem. *Mathematics of Operations Research*, 17:36–42, 1992.

[15] B. Hoppe. Efficient dynamic network flow algorithms. Ph.D. thesis, Cornell University, 1995.

[16] B. Hoppe and É. Tardos. The quickest transshipment problem. *Mathematics of Operations Research*, 25:36–62, 2000.

[17] N. Jorgensen. Some aspects of the urban traffic assignment problem. Graduate Report, Institute of Transportation and Traffic Engineering, UC Berkeley, 1963.

[18] B. Klinz and G. J. Woeginger. Minimum-cost dynamic flows: the series–parallel case. *Networks*, 43:153–162, 2004.

[19] E. Köhler, K. Langkau, and M. Skutella. Time-expanded graphs for flow-dependent transit times. In *Proceedings of the* 10*th Annual European Symposium on Algorithms* (*ESA*), pages 599–611. Volume 2461 of Lecture Notes in Computer Science. Springer, Berlin, 2002.

[20] E. Köhler and M. Skutella. Flows over time with load-dependent transit times. *SIAM Journal on Optimization*, 15:1185–1202, 2005.

[21] D. H. Lorenz and D. Raz. A simple efficient approximation scheme for the restricted shortest path problem. *Operations Research Letters*, 28:213–219, 2001.

[22] N. Megiddo. Combinatorial optimization with rational objective functions. *Mathematics of Operations Research*, 4:414–424, 1979.

[23] 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*, chapter 3, pages 141–295. Volume 8 of Handbooks in Operations Research and Management Science. North-Holland, Amsterdam, 1995.

[24] B. Ran and D. E. Boyce. *Modelling Dynamic Transportation Networks*. Springer, Berlin, 1996.