Flows over Time: Towards a more Realistic and Computationally Tractable Model*

Alexander Hall† Heiko Schilling‡

Abstract

We introduce a novel model for "flows over time" which captures the behavior of cars traveling through a road network better than previous models. We show that computing an optimal solution in the new model is $\mathcal{NP}$-hard and present an LP-based algorithm which we evaluate with several experiments on real world data of road networks and generated requests. Among other things we compare the quality of the solutions with solutions generated by an FPTAS for a related but considerably less realistic model.

1 Introduction

The recent years have witnessed a revival of the *flows over time* model (also referred to as dynamic flows) which was already introduced by Ford and Fulkerson in the late 1950s. A host of papers considering diverse facettes and extensions of the problem have appeared. Indeed, flow variation over time is an important feature in network flow problems arising in numerous applications such as road or air traffic control, evacuation problems, production systems, communication networks (e.g. the Internet), and financial flows. The survey articles by Aronson [2] and Powell et al. [23] as well as the book published by Ran and Boyce [25] contain further examples and detailed descriptions of possible applications.

In the model introduced by Ford and Fulkerson [9, 10] the individual edges of a network have associated constant transit times, determined by the speed at which flow traverses them. The flow rates into the edges may vary over time and are bounded by given capacities. In [8] an FPTAS for the *multicommodity quickest flow* problem—where the objective is to send given demands from their sources to their sinks as quickly as possible—was proposed. The problem was later shown to be $\mathcal{NP}$-hard [13].

When considering road traffic (and many other settings as well, for that matter) it is apparent that the assumption of having constant transit times is quite unrealistic. The speed at which traffic travels heavily depends on the current situation, e.g. during rush hour it is much slower than at four o'clock at night. There have been a few attempts to incorporate such flow-dependent transit times into the flows over time model, see e.g. [4, 19, 14]. They have in common though that the assumptions made are unrealistic in one way or another. We will give details in the next section, where the so called inflow-dependent and load-dependent settings will be described.

There are common approaches to study traffic problems other than the flows over time model, such as traffic simulation (see e.g. [21, 3]) and models based on fluid dynamics [24] and variational inequalities [7]. While simulation is a powerful tool to evaluate traffic scenarios, it misses the optimization potential—i.e. the objective to send flow as quickly as possible through the network. On the other hand, fluid models and other models based on differential equations capture very well the dynamical behaviour of traffic as a continuous quantity, but currently cannot handle large networks.

Contributions, Outline of the Paper. The main contribution of this paper is the proposition of a new approach to model the dependence of travel speed on the current flow situation and a heuristic algorithm for this setting. The idea of the *rate-dependent* model is quite straightforward: the maximal possible speed at any position on an edge, at any point in time always directly depends on the current flow rate. Put simply, the higher the current flow rate, the slower the flow can move. After a detailed description of the flows over time problem and the two aforementioned variations in the next section we introduce the rate-dependent model in Section 3, which finishes with a qualitative comparison of the different models. In Section 4 we then prove

*First author partially supported by the European Commission in the 6th Framework Programme under contract 001907 (DELIS), with funding provided by the Swiss Federal Office for Education and Science; additional support by DICS-Project No. 1838 by the Hasler Foundation. Second author supported by the Deutsche Forschungsgemeinschaft (DFG) as part of the Research Cluster "Algorithms on Large and Complex Networks" (1126) and by the European Graduate Program "Combinatorics, Geometry, and Computation".

†ETH Zürich, Switzerland, alex.hall@gmail.com

‡TU Berlin, Germany, heiko.schilling@tu-berlin.de

$\mathcal{NP}$-hardness of the rate-dependent model. We proceed to introduce a heuristic for the quickest flow problem in Section 5. Finally, the paper concludes with an experimental comparison of the heuristic algorithm with an FPTAS presented in [14] for the inflow-dependent model.

2 Preliminaries

We consider network flow problems in a directed graph $G = (V, E)$ with $n := |V|$ nodes and $m := |E|$ edges. For an edge $e = (v, w)$ we write $\text{tail}(e) := v$ and $\text{head}(e) := w$. For a node $v \in V$ we denote by $\delta^+(v)$ and $\delta^-(v)$ the outgoing edges of v (i.e. $\text{tail}(e) = v$) and the incoming edges of v (i.e. $\text{head}(e) = v$) respectively.

We start with giving some notation for classical *static* flow problems, then we move on to the *flows over time* setting with constant transit times, as introduced by Ford and Fulkerson. Finally, we briefly describe two generalizations where the transit times depend on the current flow situation: inflow-dependent and load-dependent transit times.

2.1 Static Flows. We are given edge capacities $u_e \in \mathbb{R}^+$, for $e \in E$, and a set of commodities $K = \{1, \ldots, k\}$. For each commodity $i \in K$ there is a source $s_i \in V$, a sink $t_i \in V$, and a demand $d_i \in \mathbb{R}^+$. A *static multicommodity flow* x in G assigns every edge–commodity pair $e \in E$, $i \in K$ a flow value $x_{e,i} \in \mathbb{R}^+$ such that *flow conservation* holds: $\sum_{e \in \delta^+(v)} x_{e,i} - \sum_{e \in \delta^-(v)} x_{e,i} = 0$, for all $v \in V \setminus \{s_i, t_i\}$. If additionally $\sum_{e \in \delta^-(t_i)} x_{e,i} - \sum_{e \in \delta^+(t_i)} x_{e,i} = d_i$ holds for all $i \in K$, the flow x *satisfies* the demands. Finally, x is said to be *feasible* if it obeys the capacity constraints $x_e := \sum_{i \in K} x_{e,i} \leq u_e$, for all $e \in E$.

2.2 Flows over Time

2.2.1 Constant Transit Times. As in static flows we are given a directed graph $G = (V, E)$ with edge capacities and a set of commodities $K = \{1, \ldots, k\}$ with sources, sinks, and demands. New in the flows over time setting is that each edge $e \in E$ is associated with a (constant) *transit time* $\tau_e \in \mathbb{R}^+$.

A *multicommodity flow over time* f in G with *time horizon* T is given by Lebesgue-measurable functions $f_{e,i} : [0, T) \to \mathbb{R}^+$, for $e \in E$ and $i \in K$. The value $f_{e,i}(\theta)$ gives the rate of flow (per time unit) of commodity i entering e at time θ. This rate reaches $\text{head}(e)$ at time $\theta + \tau_e$. The edges have to be empty from time T on, that is we require $f_{e,i}(\theta) = 0$, for $\theta \in [T - \tau_e, T)$. For ease of exposition, we sometimes use $f_{e,i}(\theta)$ for $\theta \notin [0, T)$. In such cases we assume $f_{e,i}(\theta) = 0$.

When generalizing the notion of *flow conservation* we distinguish two cases: either *storage* of flow at intermediate nodes is allowed, or it is not. Intuitively, we could describe the problem as follows: in the first case flow can only leave the node if it has previously entered the node, i.e. it can wait for an abritrary period of time at the node before it flows on. In the second case the total amount of flow that has left a node up to a given point in time must be equal to the total amount that has entered the node until that time. To formalize this we define

$$\begin{aligned} &D^-_{v,i}(\xi) := \textstyle\sum_{e \in \delta^-(v)} \int_{\tau_e}^{\xi} f_{e,i}(\theta - \tau_e)\, d\theta \qquad \text{and} \\ &D^+_{v,i}(\xi) := \textstyle\sum_{e \in \delta^+(v)} \int_0^{\xi} f_{e,i}(\theta)\, d\theta \end{aligned} \tag{2.1}$$

to be the total inflow (outflow) of commodity $i \in K$ into (out of) node $v \in V$ until time $\xi \in [0, T]$. The generalized flow conservation constraints then amount to:

$$\begin{aligned} &\text{With storage} && \to \quad D^+_{v,i}(\xi) - D^-_{v,i}(\xi) \leq 0 \\ &\text{Without storage,} && \\ &\text{or if } \xi = T && \to \quad D^+_{v,i}(\xi) - D^-_{v,i}(\xi) = 0, \end{aligned} \tag{2.2}$$

for all $\xi \in [0, T], i \in K$, and $v \in V \backslash \{s_i, t_i\}$. Generally, flow must not remain in any node other than the sinks at time T. Therefore, we also require that equality holds at time $\xi = T$ for the case with storage. We assume w.l.o.g. that sources have only outgoing and sinks only incoming edges, implying $D^-_{s_i,i}(\xi) = D^+_{t_i,i}(\xi) = 0$, for $\xi \in [0, T), i \in K$. The flow over time f *satisfies* the multicommodity demands if

$$D^-_{t_i,i}(T) \left[\overset{(2.2)}{=} D^+_{s_i,i}(T) \right] = d_i$$

for every commodity $i \in K$. Finally, a flow over time f is called *feasible* if the rate of flow entering an edge e is upper bounded by the capacity u_e, i.e. $f_e(\theta) := \sum_{i \in K} f_{e,i}(\theta) \leq u_e$, at any point in time $\theta \in [0, T)$. This is in contrast to the static case where the capacities bound the actual flow amount.

2.2.2 Inflow-Dependent Transit Times. In this more general setting introduced by Carey and Subrahmanian [4], the transit time of edge $e \in E$ may depend on the rate of flow entering e, i.e. it is given by a left-continuous Lebesgue-measurable function $\tau_e : [0, u_e] \to \mathbb{R}^+$. Then the flow at rate $f_e(\theta)$ entering e at time θ arrives at $\text{head}(e)$ at time $\theta + \tau_e(f_e(\theta))$. To adapt the definitions from the constant transit times case, only the expression given for $D^-_{v,i}(\xi)$ in (2.1) needs to be adjusted accordingly. An FPTAS and a proof of $\mathcal{NP}$-hardness for this setting were peresented in [14].

Remark. The speed at which units of flow traverse the edge (i.e. their transit time) is fixed when they enter the edge. Therefore flow entering at low rate might overtake flow which previously entered the edge at higher rate. Since the edge capacities are only ensured when entering the edge ($f_e(\theta) \leq u_e$), the edge capacities may be exceeded arbitrarily when units of flow overtake each other.

2.2.3 Load-Dependent Transit Times.

Köhler and Skutella [19] propose a variation of the problem where the transit time of edge $e \in E$ depends on the current load $l_e(\theta)$ of that edge. The load $l_e(\theta)$ is the cumulative inflow into the edge e until $\theta \in [0,T)$ (i.e. $\int_0^\theta f_e(\xi)\,d\xi$) minus the cumulative outflow out of e until θ.

At any point in time $\theta \in [0,T)$ all flow on the edge travels with the same speed, namely the inverse of the 'current transit time' $\hat{\tau}_e(l_e(\theta))$. In this case it is more appropriate to speak of the (again Lebesgue-measurable) functions $\hat{\tau}_e : \mathbb{R}^+ \to \mathbb{R}^+$ as giving the current *pace* of the flow, rather than its *transit time*. In [19] a 2-approximation algorithm is presented for the problem with only one commodity.

Remark. In this setting units of flow cannot overtake each other, since at any point in time all units of flow presently on an edge travel at the same pace. Thus the abovementioned problem of exceeded capacities does not occur here. Units of flow already on an edge may be slowed down though by additional units of flow entering the edge later (and thereby increasing the load). This is quite an unrealistic peculiarity when considering traffic flows.

2.2.4 Problem Definitions.

For constant or flow-dependent transit times one can consider the following problem variations: In the *multicommodity flow over time* problem one seeks to find a feasible flow over time f with time horizon T that satisfies all demands. The *quickest multicommodity flow* problem is to additionally minimize the time horizon T.

Moreover, costs $c_{e,i}$ can be associated with each edge–commodity pair giving the cost of sending one unit of flow of commodity $i \in K$ via edge $e \in E$. Now, a further objective could be to consider only flows whose costs are within a given *budget* C. Note that for simplicity we did not include costs in the definition of the rate-dependent setting below. Both the model and the subsequently proposed heuristic can be modified in a straightforward way though to incoporate costs $c_{e,i}$ per unit of flow, for $i \in K$ and $e \in E$.

3 The Rate-Dependent Model

As motivated above, now we introduce a model which we believe reflects behavior of real world traffic better than both the load-dependent and the inflow-dependent models described in the previous section. As a first step we refine the notion of a flow over time by defining it not only via inflow rate functions over time into the edges (i.e. $f_{e,i}(\theta)$) but via flow rate functions over time and for any point on an edge. To be more precise: a *multicommodity flow over time* f in G with time horizon T is given by Lebesgue-measurable functions $f_{e,i} : [0,T) \times [0,1] \to \mathbb{R}^+$, for $e \in E$ and $i \in K$; again, we sometimes use $f_{e,i}(\theta,\cdot)$ for $\theta \notin [0,T)$ in which case we assume a value of 0. The second parameter gives the relative position on the edge. For instance, $f_{e,i}(\theta,0)$ would be the inflow rate $f_{e,i}(\theta)$ as defined in the previous section, the other extreme—$f_{e,i}(\theta,1)$—denotes the outflow rate. It is obvious that both the load-dependent and the inflow-dependent model lend themselves to natural definitions of $f_{e,i}(\theta,p)$, for $\theta \in [0,T)$ and $p \in [0,1]$, assuming that the functions $f_{e,i}(\theta)$ are given, for $\theta \in [0,T)$, $e \in E$, and $i \in K$: simply let the flow rates "move across the edge" in a way that is compatible with the corresponding assumptions concerning the transit times, see also Figure 2.

Extending the definition of a flow over time in such a way will make it possible to introduce a certain coupling constraint between the flow rate and the pace at which flow travels for all positions on the edge. But first, we need a few preliminary definitions. Let

$$D_{e,i}(\xi,p) := \int_0^\xi f_{e,i}(\theta,p)\,d\theta \tag{3.3}$$

denote the total amount of flow arriving at position $p \in [0,1]$ of $e \in K$ until time $\xi \in [0,T)$, where $i \in K$. Then

$$\begin{aligned} D^-_{v,i}(\xi) &:= \textstyle\sum_{e\in\delta^-(v)} D_{e,i}(\xi,1) \qquad \text{and} \\ D^+_{v,i}(\xi) &:= \textstyle\sum_{e\in\delta^+(v)} D_{e,i}(\xi,0) \end{aligned} \tag{3.4}$$

are the total inflow and outflow respectively until time $\xi \in [0,T)$ of commodity $i \in K$ at node $v \in V$. These modified definitions allow us to adopt the flow conservation constraint for the nodes as in (2.2). Since the flow f is now defined for all points on the edge (and could be given by arbitrary functions), we have to make sure that flow conservation also holds on the edges.

Flow Conservation for Edges. First we demand $D_{e,i}(T,p) = D_{e,i}(T,0)$ to hold for $p \in [0,1]$ and $i \in K$, i.e. no flow traverses the network from time T on. We now take a closer look at a cumulative flow amount $D \in [0, D_{e,i}(T,0)]$ of commodity $i \in K$ and consider when this amount D reaches a position $p \in [0,1]$ on e:

$$t_{e,i}(D,p) := \min\{\xi | D_{e,i}(\xi,p) = D\}. \tag{3.5}$$

Note that this is well defined since $D_{e,i}(\xi,p)$ is continuous and weakly increasing (non-decreasing) in ξ, for fixed $p \in [0,1]$. This follows from the definition of $D_{e,i}(\xi,p)$ and from $f(\cdot,\cdot) \geq 0$. Moreover, since $D_{e,i}(0,p) = 0$ and $D_{e,i}(T,p) = D_{e,i}(T,0)$ the value D is reached at some point in time ξ.

For flow conservation to hold for a flow f, $t_{e,i}(D,p)$ must be weakly increasing and once differentiable in p, for fixed D.

Pace. Let $f_e(\theta,p) = \sum_{i\in K} f_{e,i}(\theta,p)$ and correspondingly $D_e(\xi,p) = \sum_{i\in K} D_{e,i}(\xi,p)$, for $p \in [0,1]$ and $\theta, \xi \in [0,T)$. Analoguous to above, given a cumulative flow amount $D \in [0, D_e(T,0)]$, let

$$t_e(D,p) := \min\{\xi | D_e(\xi,p) = D\}$$

be the first point in time when the amount D (of all commodities) reaches position $p \in [0,1]$ on e. Again, we focus on flows over time f for which $t_e(D,p)$ is weakly increasing and once differentiable in p, for fixed D. Given $\theta \in [0,T)$ and $p \in [0,1]$ we define the current *pace*

$$\tau_e(\theta,p) := \begin{cases} \partial\, t_e(D,p)/\partial\, p & \text{if } \exists D \text{ s.t. } t_e(D,p) = \theta, \\ \infty & \text{otherwise.} \end{cases}$$

Note: the pace is "time divided by distance", i.e. the inverse of the velocity. Example: the pace $\tau_e(\theta,0)$ describes the time it would take an infinitesimal unit of flow entering e at time θ to reach the other end of e, if the pace stays constant during the traversal of the edge.

Feasibility. A flow over time f is *feasible*, if

$$(f_e(\theta,p), \tau_e(\theta,p)) \in F_e,$$

for all $\theta \in [0,T)$, $p \in [0,1]$, and $e \in E$. Where F_e is a certain closed *feasibility region* giving which combinations of pace and flow rate can occur. This is the aforementioned coupling constraint. Figure 1 shows an example of such a region adapted from a model of pace–flow rate interrelation suggested by Greenshields [12]. Intuitively speaking, at first an increase in pace (i.e. less speed) allows cars to move closer to each other and thus the flow rate can increase. From a certain point on though, when cars move closer and closer the decrease in distance cannot compensate for the increase in pace anymore: the flow rate starts decreasing again. In the extreme, the cars are bumper to bumper and do not move at all, i.e. the flow rate is 0 and the pace ∞. In other words Greenshields model takes into account traffic jam behavior occurring in the real world.

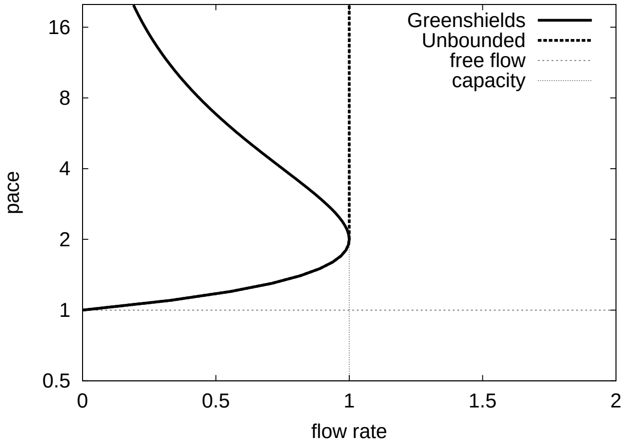

Figure 1: The relationship between pace and flow rate as given by Greenshields. As feasibility region we propose the area to the left of the curve (between ordinate and curve). The additionally depicted unbounded region permits arbitrary storage of flow. The free flow pace is the pace at which an infinitesimal unit of flow can travel, if it is not disturbed by any other units of flow. The capacity is an upper bound for the flow rate $f_e(\cdot,\cdot)$. Both are normalized to 1.

Allowing pace–flow rate combinations from a whole region (e.g. left of the curve depicted in Figure 1) basically amounts to permitting flow to travel slower than the maximum possible speed but also not too slow, considering the current flow rate. This is novel compared to the models described in the previous section, where the transit times are fixed (constant, or depending on the inflow-rate or load). This more flexible approach could potentially be helpful when minimizing the time horizon. In practice, imagine situations where a route guidance system or changable speed limit signs advise drivers to slow down in order to avoid a potential traffic jam. Of course this flexibility could be simulated by allowing intermediate storage of flow at nodes. This situation is however not very realistic in road networks, as it assumes an infinite storage capacity at each node (road crossing).

Figure 1 also shows a possible unbounded feasibility region. In this case flow can be stored arbitrarily on an edge, which is analogous to allowing storage of flow at nodes in the other settings. Other well known pace–flow rate relations were for instance suggested by the U.S. Bureau of Public Roads or Davidson, see e.g. [5] for details. For our heuristic algorithm below we will make the natural restriction that the feasibility region must be convex for flow rates > 0.

In principal, we treat the feasibility region F_e of each edge $e \in E$ as part of the input. In practice, it makes sense to consider only one normalized region as shown in Figure 1. For different edge lengths, this region could simply be scaled individually for each edge e in such a way that the free flow pace equals a given transit time τ_e. Similarly, the region could be scaled to meet a given capacity u_e.

3.1 Comparison of Models. Figure 2 shows a comparison of how a certain flow given as an edge inflow-rate function could traverse the edge in the four models described. In the load-dependent setting one can see that the first units of flow entering the edge start at higher speed than in the constant transit times case, but then are slowed down as more flow arrives on the edge. In the inflow-dependent case the first two blocks of flow entering at low rates can move away. Similarly, the last two blocks of flow (low rates) overtake the large block of flow in the center that entered at a high rate. In the second and third snapshot this might lead to a violation of the edge capacities, as mentioned before. In the example shown for the rate-dependent setting again the first two blocks of flow move away, since their low flow rates permit smaller paces. But here the two last blocks of flow do not overtake the central high rate block. This would lead to infeasible flow rate and pace combinations. In this example the two last blocks simply slow down and merge with the block of higher rate flow.

Note: Another way of defining a more realistic model similar to the rate-dependent model would be to consider either the load-dependent or the inflow-dependent models and to subdivide each edge into many small edges, possibly investigating the limit.

4 $\mathcal{NP}$-Hardness

The following theorem can be shown by a reduction of the flows over time problem with constant transit times to the rate-dependent setting. The former has been proven to be $\mathcal{NP}$-hard already for two commodities or series-parallel graphs in [13].

Theorem 4.1. *The multicommodity flow over time problem with rate-dependent transit times is $\mathcal{NP}$-hard for two or more commodities, or alternatively for series-parallel graphs and an arbitrary number of commodities.*

Proof. Given an instance for the setting with constant transit times—i.e. a graph $G = (V, E)$, capacities u_e, transit times τ_e, for $e \in E$, and commodities K, assuming that intermediate storage of flow at nodes is allowed—construct an instance of the rate-dependent model as follows. Copy the graph and the commodities $\overline{G} = (\overline{V}, \overline{E}) := (V, E)$. For the rest of the proof we fix $e \in E$ as an arbitrary edge (with $\overline{e} \in \overline{E}$ as its copy) and $i \in K$ as an arbitrary commodity. Let $\overline{e}$'s feasibility region be given by:

$$F_{\overline{e}} := \{(u, \tau) | u \le u_e, \tau \ge \tau_e\}.$$

Since these regions are unbounded we may assume for simplicity that intermediate storage of flow at nodes is also permitted in the rate-dependent setting. Otherwise it could be simulated on the outgoing edges of the individual nodes.

Let f $(\overline{f})$ denote a flow over time in G $(\overline{G})$ with constant transit times (rate-dependent transit times). For clarity we write $\overline{D}_{\overline{e},i}(\cdot,\cdot)$ and $\overline{D}_{\overline{v}}^{-/+}(\cdot)$ when referring to the expressions given in (3.3) and (3.4) respectively. Now we prove that given a flow f in G, a flow $\overline{f}$ in $\overline{G}$ with the same time horizon can be derived and vice versa.

Given a flow f with time horizon T let

$$\overline{f}_{\overline{e},i}(\theta, p) := f_{e,i}(\theta - p \cdot \tau_e), \quad \text{for } \theta \in [0, T),\ p \in [0, 1).$$

It is easy to see that by construction $t_{\overline{e},i}(D, p) = t_{\overline{e},i}(D, 0) + p \cdot \tau_e$, which is continuous and strictly increasing in $p \in [0, 1]$ for fixed $D \in [0, \overline{D}_{\overline{e},i}(T, 0)]$. Hence, flow conservation holds on the edges. Now we have to show that flow conservation also holds at each node $\overline{v} \in \overline{V}$ and each point in time $\xi \in [0, T)$: from $\overline{f}_{\overline{e},i}(\theta, 1) = f_{e,i}(\theta - \tau_e)$ and (2.1), (3.4) it follows directly that $\overline{D}^-_{\overline{v},i}(\xi) = D^-_{v,i}(\xi)$. Obviously $\overline{D}^+_{\overline{v},i}(\xi) = D^+_{v,i}(\xi)$ holds as well and f is a feasible flow over time, thus flow conservation (2.2) also holds for $\overline{f}$.

Analogously we obtain $t_{\overline{e}}(D, p) = t_{\overline{e}}(D, 0) + p \cdot \tau_e$, for $D \in [0, \overline{D}_{\overline{e}}(T, 0)]$ and $p \in [0, 1]$. Thus the pace $\tau_{\overline{e}}(\theta, p) \in \{\tau_e, \infty\}$ at any time $\theta \in [0, T)$ and for any relative position $p \in [0, 1]$.[1] Feasibility follows from $\overline{f}_{\overline{e},i}(\theta, p) = f_{e,i}(\theta - p \cdot \tau_e) \le u_e$, for $\theta \in [0, T)$ and $p \in [0, 1]$. Because $\overline{f}$ clearly has the same time horizon T as f this completes the first direction of the proof.

For the other direction, let $\overline{f}$ be a feasible rate-dependent flow with time horizon T. We set f to

$$f_{e,i}(\theta) := \overline{f}_{\overline{e},i}(\theta, 0).$$

Below we will argue that

$$t_{\overline{e},i}(D, 1) \ge t_{\overline{e},i}(D, 0) + \tau_e, \tag{4.6}$$

for $D \in [0, \overline{D}_{\overline{e},i}(T, 0))$. This is helpful for proving that flow conservation (2.2) holds for f, since it implies a

[1] By definition the pace is τ_e, if flow is “passing” p at time θ (i.e. there is D such that $t_{\overline{e}}(D, p) = \theta$). Otherwise the pace is ∞.

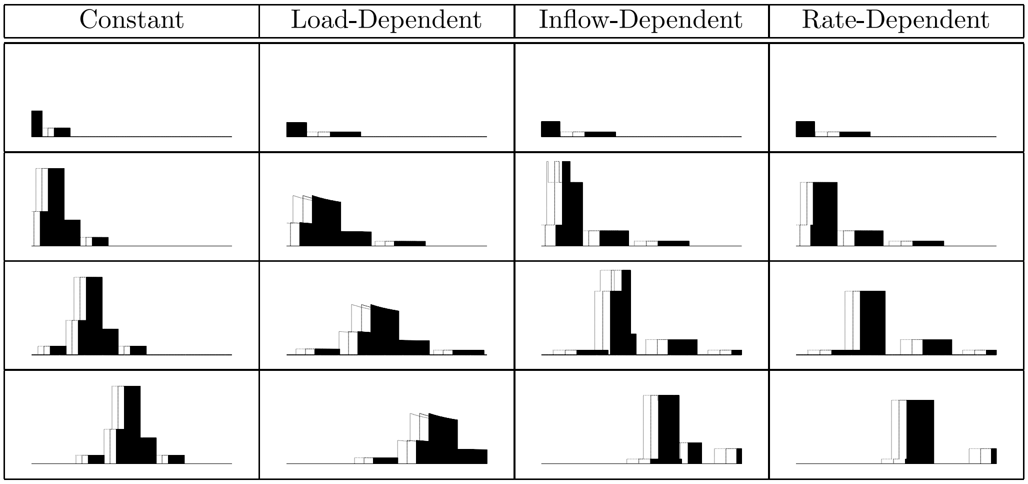

Figure 2: A comparison of different models incorporating flow-dependent transit times. Time is increased stepwise from one row to the next. The black region in each cell shows $f_{e,i}(\theta, p)$ for a fixed θ, for all $p \in [0, 1]$, and a specific model. The dotted lines indicate the flow rate functions for two recent time-steps. The incoming flow rates are the same for all four models. Note that the flow shown for the rate-dependent model is only one of infinitely many possible solutions.

transit time of τ_e for every infinitesimal unit of flow in $\overline{f}$ on edge $\overline{e}$. Formally, by f's construction and (4.6)

$$\text{(4.7)} \qquad \int_0^\xi f_{e,i}(\theta)\,d\theta = \overline{D}_{\overline{e},i}(\xi, 0) \geq \overline{D}_{\overline{e},i}(\xi + \tau_e, 1),$$

for $\xi \in [0, T)$. To see the inequality, consider that by (4.6) any fixed amount of flow $D \in [0, \overline{D}_{\overline{e},i}(T, 0)]$ reaches head(e) at least τ_e units of time later than it arrived at tail(e).

Consider a node $v \in V$ and a point in time $\xi \in [0, T)$. Plugging (4.7) into (2.1) we obtain $D^-_{v,i}(\xi) \geq \overline{D}^-_{\overline{v},i}(\xi)$. With $D^+_{v,i}(\xi) = \overline{D}^+_{\overline{v},i}(\xi)$ flow conservation (2.2) for f follows, since flow conservation holds for $\overline{f}$.

From the definition of the feasibility region $F_{\overline{e}}$ it follows directly that $f_{e,i}(\theta) = \overline{f}_{\overline{e},i}(\theta, 0) \leq u_e$, for $\theta \in [0, T)$. Therefore, f is feasible.

To conclude the proof it remains to show that (4.6) holds. We need a special property of the problem instances used in the $\mathcal{NP}$-hardness proofs in [13]: the edges with non-zero transit times (for the others (4.6) is trivial) all transport flow of only one distinct commodity during the entire time horizon. Let e be such an edge. Since $t_{\overline{e},i}(D, p) = t_{\overline{e}}(D, p)$, its first derivative with respect to p and for fixed D is at least τ_e (by construction of the feasibility region $F_{\overline{e}}$). Inequality (4.6) follows immediately, which concludes the proof of the theorem. $\diamondsuit$

5 Computation of Rate-Dependent Flows over Time

We start this section by describing the useful notion of time-expanded graphs for the constant transit times setting, then we make a suggestion for deriving a graph with constant transit times from a rate-dependent instance. Time-expansion of this graph and some subsequently presented modifications will result in a "diamond graph". We then move on to show how a rate-dependent flow over time in the original instance can be obtained from a static flow—which adheres to certain coupling constraints—in the diamond graph. We conclude this section by presenting our heuristic algorithm for the quickest flow problem for the rate-dependent setting.

5.1 Time-Expanded Graphs. Many flow over time problems can be solved by static flow algorithms in time-expanded graphs, which were introduced in [9, 10]. Given a graph $G = (V, E)$ with integral edge transit times 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 $v(\theta)$, $\theta = 0, \dots, T-1$. For every edge $e = (v, w) \in E$ and $\theta = 0, \dots, T-1-\tau_e$, there is an edge $e(\theta)$ from $v(\theta)$ to $w(\theta + \tau_e)$ with the same capacity and costs as edge e. Additionally, if intermediate storage of flow is allowed, there is an infinite capacity *holdover edge* from $v(\theta)$ to $v(\theta + 1)$, for all $v \in V$ and $\theta = 0, \dots, T-2$, which models the possibility of holding flow at node v during the time interval $[\theta, \theta+1)$.

Any static flow x in this time-expanded network corresponds to a flow over time f: interpret the static flow on edge $e(\theta)$ as the flow over time through edge $e = (v, w)$ that starts at node v in the time interval $[\theta, \theta+1)$. Formally, we set $f_{e,i}(\xi) := x_{e(\theta),i}$ for $i \in K$, $\xi \in [\theta, \theta+1)$, and $\theta \in \{0, \ldots, T-1\}$. Similarly, any flow over time completed by time T corresponds to a static flow in G^T of the same value, obtained by mapping the total flow over time entering e in the time interval $[\theta, \theta+1)$ to static flow on edge $e(\theta)$. Thus, we may solve a flow over time problem by solving the corresponding static flow problem in the time-expanded network. See Figure 3 for an example of the correspondence of flow over time in G and static flow in G^T.

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 [8]: if all transit times and T 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 T/Δ copies of V. Since in this setting every edge corresponds to a time interval of length Δ, capacities are multiplied by Δ.

With the help of Δ-condensed time-expansion it is possible to devise FPTASs for the constant transit times and even the inflow-dependent transit times settings, presented in [8] and [14] respectively. To obtain an inflow-dependent flow the latter replaces each edge by a "bow" of edges, each having a constant transit time. By adding certain coupling constraints on flow traversing the various edges of a bow it is possible to deduce an inflow-dependent flow over time from a flow over time in this reformulation with constant transit time bow edges. We use a similar technique for our rate-dependent heuristic.

5.2 Bow Graph. Given a rate-dependent instance $G = (V, E)$ with feasibility regions F_e, for $e \in E$, we construct a *bow graph* $G^B = (V^B, E^B)$ with constant transit times: let $V^B = V$ and for each original edge $e \in E$ add a *bow* of edges $E_e^B = \{e_0, e_1, \ldots\}$ "simulating" the original edge. The transit time of e_i is given by $\tau_{e_i} = i \cdot \Delta$, where Δ is chosen in such a way that $L := T/\Delta$, the number of *levels*, is polynomial in the input size. Since we are looking for a flow over time with time horizon T, we only consider L bow edges, i.e. $|E_e^B| = L$. The capacity of e_i is given by $u_{e_i} = \Delta \cdot \max\{u | (u, i \cdot \Delta) \in F_e\}$.

5.3 Diamond Graph. The *diamond graph* $G^D = (V^D, E^D)$ is obtained by first doing a Δ-condensed time-expansion of the bow graph G^B and then adding *crossing nodes* to V^D. Consider an edge $e = (v, w) \in E$, its bow E_e^B, and the time-expansion of the bow, say E_e^T. Imagine the depiction of E_e^T in the plane, $v(\theta)$ at position $(\theta, 0)$ and $w(\theta)$ at position $(\theta, 1)$, where $\theta \in \{0, \Delta, \ldots, T-\Delta\}$. Each edge $e' \in E_e^T$ translates into a straight line in the plane (Figure 3 shows such a depiction for several edges combined). For each pair of crossing edges $e', e'' \in E_e^T$ we now add a node to V^D that represents this crossing (and modify E^D correspondingly). I.e. if $e' = (v(\theta'), w(\theta' + \tau_{e'}))$ and $e'' = (v(\theta''), w(\theta'' + \tau_{e''}))$, this crossing node represents the point in time and space (θ, p) for which $\theta = \theta' + p \cdot \tau_{e'} = \theta'' + p \cdot \tau_{e''}$. It basically enables flow to change its pace from $\tau_{e'}$ to $\tau_{e''}$ or vice versa at position p on the original edge.

Now our aim is to derive a rate-dependent flow over time in G from a static flow x in G^D (computed by using an LP formulation). To ensure feasibility we add a coupling of flow values on edges in E_e^D (= all edges in E^D corresponding to an edge $e \in E$), if these edges represent overlapping areas in time and space.

5.3.1 Coupling Constraints, LP Formulation. Let $e = (v(\theta_1), w(\theta_2)) \in E^D$ denote an edge in the diamond graph, which corresponds to a pace τ_e. Let $p_1 \in [0,1]$ ($p_2 \in [0,1]$) be the position of $v(\theta_1)$ ($w(\theta_2)$) on the original edge. Note that $v(\theta_1)$ ($w(\theta_2)$) could be a crossing node. The edge e represents a diamond shaped area in time and space

$$\begin{aligned} A_e \quad := \quad & \{(\theta, p) | p \in [p_1, p_2], \theta \in [\xi, \xi + \Delta), \\ & \text{with } \xi := \theta_1 + (p - p_1) \cdot \tau_e\}. \end{aligned}$$

For $\overline{e} \in E^D$ let $U_{\overline{e}} := \{e \in E^D | e, \overline{e}$ resulted from the same bow, $A_e \cap A_{\overline{e}} \neq \emptyset\}$ denote the set of edges whose areas overlap the area of $\overline{e}$. We can now state our additional coupling constraints posed on a static flow x in G^D:

$$(5.8) \qquad \sum_{e \in U_{\overline{e}}} \lambda_e := \sum_{e \in U_{\overline{e}}} x_e / u_e \leq 1, \qquad \text{for } \overline{e} \in E^D.$$

In other words, the sum of the *per capacity flow rates* λ_e of overlapping edges must not exceed one. Below we prove that this ensures feasibility under a certain assumption concerning the feasibility regions.

It is simple to compute a flow which adheres to these constraints. Just add them to a standard LP formulation for static flows in G^D. Note that the objective of this static flow problem is to satisfy all demands, i.e. to send d_i units of flow from any $s_i(\theta)$

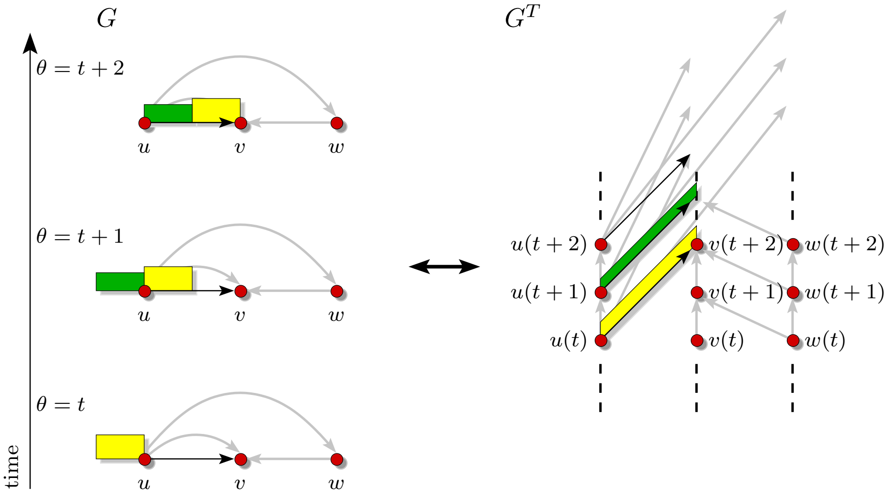

Figure 3: Simple example of how a flow over time corresponds to a static flow in the time-expanded graph. The two "boxes" entering edge (u, v) are flow amounts arriving over time. The respective heights of the boxes give the rate of flow. The corresponding boxes in G^T represent the same flow amounts, here as static flow.

to any $t_i(\theta)$, $\theta \in \{0, \Delta, \ldots, T - \Delta\}$, for each commodity $i \in K$.

5.3.2 Deriving a Rate-Dependent Flow over Time. Given a static flow x in G^D adhering to (5.8) the corresponding flow over time f is easy to state. Consider an edge $e' \in E$ in the original graph and a point in time and space $(\theta, p) \in [0, T) \times [0, 1]$. Let

$$\overline{E} := \{e \in E^D_{e'} | (\theta, p) \in A_e\} \tag{5.9}$$

denote all edges in the diamond graph whose area contains this point. The flow rate of commodity $i \in K$ is then given by

$$f_{e',i}(\theta, p) := \sum_{e \in \overline{E}} x_{e,i}/\Delta.$$

It is easy to see that flow conservation at nodes and on edges holds by construction.[2] For f to be feasible we need the following assumption.

Convexity Assumption. The intersection $F_e \cap \{(u, \tau)|u > 0, \tau >= 0\}$ is convex for all $e \in E$. This holds e.g. for the unbounded region shown in Figure 1. The Greenshields region could be modified to be convex by cutting of the top part at a tangent without losing too much of the region and, more importantly, thereby restricting to flows which are also feasible for the original region.

Theorem 5.1. *The flow f is feasible under the convexity assumption*

Proof. Consider an edge $e' \in E$ and a point in time and space $(\theta, p) \in [0, T) \times [0, 1]$. Let $\overline{E}$ be defined as in (5.9). Note that $\overline{E} \subseteq U_{\overline{e}}$, for any $\overline{e} \in \overline{E}$. With this and (5.8) we have $\sum_{e \in \overline{E}} \lambda_e \leq 1$. We will prove below that the following inequality results for the flow rate on e' at (θ, p):

$$\begin{aligned} f_{e'}(\theta, p) &= \sum_{e \in \overline{E}} x_e/\Delta = \sum_{e \in \overline{E}} \lambda_e \cdot (u_e/\Delta) \\ &\leq \sum_{e \in \overline{E}} \frac{x_e}{\sum_{\overline{e} \in \overline{E}} x_{\overline{e}}} \cdot (u_e/\Delta) =: u. \end{aligned} \tag{5.10}$$

The pace is given as the weighted sum of the paces on the individual overlapping edges.

$$\tau_{e'}(\theta, p) = \sum_{e \in \overline{E}} \frac{x_e}{\sum_{\overline{e} \in \overline{E}} x_{\overline{e}}} \cdot \tau_e =: \tau.$$

The point (u, τ) is clearly a linear combination of points in the feasibility region $F_{e'}$, see the definition of the bow $E^B_{e'}$ above. Hence, $(u, \tau) \in F_{e'}$ and because of the inequality for u also $(f_{e'}(\theta, p), \tau_{e'}(\theta, p) \in F_{e'}$.

It remains to prove inequality (5.10). As a first step we can assume w.l.o.g. that $\sum_{e \in \overline{E}} \lambda_e = 1$. If this is not the case, we can multiply the λ_e by a corresponding

[2] It helps our intuition if we consider the depiction of G^D in the plane with the flow values of x associated to the corresponding, partially overlapping edge areas. This image directly represents the flow f traveling through the network. Flow conservation at the (crossing and original) nodes follows from x being a valid static flow. For flow conservation on the edges it needs to be checked that the conditions on the expression given in (3.5) hold (strictly increasing, continuous, and once differentiable). This is straightforward calculus.

factor; this would only increase the right hand side of the first line in (5.10). We now want to show that

$$\begin{aligned}\sum_{e\in\overline{E}} \lambda_e \cdot u_e \quad &\le \quad \sum_{e\in\overline{E}} \frac{x_e}{\sum_{\overline{e}\in\overline{E}} x_{\overline{e}}} \cdot u_e \\ \sum_{e\in\overline{E}} \lambda_e \cdot u_e \quad &\le \quad \sum_{e\in\overline{E}} \frac{\lambda_e \cdot u_e}{\sum_{\overline{e}\in\overline{E}} \lambda_{\overline{e}} \cdot u_{\overline{e}}} \cdot u_e \\ \Big(\sum_{e\in\overline{E}} \lambda_e \cdot u_e\Big)^2 \quad &\le \quad \sum_{e\in\overline{E}} \lambda_e \cdot u_e^2\end{aligned}$$

Let $y := \sum_{e\in\overline{E}} \lambda_e \cdot u_e^2 - \big(\sum_{e\in\overline{E}} \lambda_e \cdot u_e\big)^2$. In order to show that $y \geq 0$ we consider the partial derivatives with respect to u_e:

$$\frac{\partial\, y}{\partial\, u_e} = 2\lambda_e \cdot u_e - 2\Big(\sum_{\overline{e}\in\overline{E}} \lambda_{\overline{e}} \cdot u_{\overline{e}}\Big) \cdot \lambda_e,$$

for $e \in \overline{E}$. Setting all partial derivatives to zero we obtain $u_e = \sum_{\overline{e}\in\overline{E}} \lambda_{\overline{e}} \cdot u_{\overline{e}}$, for $e \in \overline{E}$. This holds iff u_e is set to a constant c, for $e \in \overline{E}$, since $\sum_{e\in\overline{E}} \lambda_e = 1$. At these extremal points $u_e = c$, for $e \in \overline{E}$, we get $y = 0$. Since $\partial\, y/\partial^2\, u_e = 2\lambda_e - 2\lambda_e^2 \geq 0$ (note that $\lambda_e \leq 1$) we know that these extremal points are minima and that for all other points $y \geq 0$ holds. This completes the proof. ♢

5.4 Quickest Flow Heuristic. To summarise, we now plug together the methods described in the previous section and present our heuristic for the quickest flow problem in the rate-dependent setting. The algorithm is given the number of levels L as input parameter, telling it how fine granular the time horizon should be subdivided.

HEURISTIC: RATE-DEPENDENT QUICKEST FLOW

1. Minimize T with geometric mean binary search: repeat the following steps for the different—guessed—values of T until the change of T is within given bounds.
2. Set $\Delta := T/L$. Construct the diamond graph G^D from G with respect to T and Δ.
3. Compute a static flow x in G^D adhering to (5.8) and satisfying all demands. If it exists, derive a rate-dependent flow over time f in G as described above.

6 Implementation and Experiments

We implemented both our heuristic for the rate-dependent case and the FPTAS presented in [14] for the inflow-dependent case where intermediate storage of flow at nodes is allowed.[3] For both a column generation approach was used to solve the corresponding LP formulations. Also in the case of our heuristic a new column can still be found by one shortest path computation even though the additional constraints (5.8) are present. In our experiments the column generation performed very well. It has the further advantage that additional constraints can be posed on the flow carrying paths in a simple manner. Such a constraint could be, for instance, to restrict flow paths to a limited length: e.g. for commodity $i \in K$ paths cannot be longer than c times the shortest path between s_i and t_i. Such a restriction would be well received in practice, e.g. in route guidance systems, where it would ensure that individuals are not forced to make long detours only for the good of others. Incorporating this amounts to solving a constraint shortest path problem each time a new column is generated. This problem is described as being NP-complete in Garey and Johnson already [11], but fully polynomial approximation schemes have also been suggested in [22, 15, 26]. A standard algorithm for constrained shortest paths which we found to work well in practice is a generalized or labeling Dijkstra algorithm [1, 6, 18].

To yield results that we can compare to other models, we chose the unbounded setting shown in Figure 1 for both the feasibility regions and the inflow-dependent transit time functions of our heuristic and the FPTAS respectively. As instances we are given road networks with edge lengths. We use these lengths to scale the region / transit time function in Figure 1 appropriately (the free flow pace then equals the given length). For the rate-dependent setting as a preprocessing step we subdivide long edges into several short ones with free flow pace less or equal to Δ. This turned out to be helpful since hereby the first few edges created for each bow E_e^B have greater differences in capacity and transit time.

Here presented algorithms were implemented in C++ using using Intel c++ compiler version 8.0 [17] on a Linux 2.6 system (SuSE 9.1). All computations were done on a 32bit Intel Pentium IV processor, 2.80GHz, 2GB memory. CPLEX version 9.0 [16] was used for solving the LPs.

6.1 Setup of Experiments. Most experiments were conducted on a small graph with 166 nodes and 238 edges, respresenting the center of Berlin. For each data point we did seven runs of randomly generated requests

[3] For the load-dependent model so far only an algorithm for the single commodity case has been proposed.

and show the mean and standard deviation of these runs, unless stated otherwise.

A Run. In a single run we generate a given number of commodities at random, i.e. we select a random source and a random sink for each. Two measures are taken in order to assure that no single commodity dominates the time horizon: first, we ensure that the shortest paths of all commodities are about the same length (i.e. to be within certain bounds). Second, the demand of a commodity is assessed according to the bottleneck capacity along its shortest path.

For each run a flow over time is computed by the heuristic and the FPTAS algorithm. The time horizons of both are divided by a lower bound computed for the inflow-dependent setting (this can be done with a slight modification of the FPTAS) in order to normalize the data, unless stated otherwise. For all three computations—heuristic, FPTAS, and lower bound—the same number of levels L is chosen.

6.2 Results and Discussion. In this section we describe each of the experiments conducted and discuss the results. The constraint shortest path factor c is set to a very large number except for the experiments concerning the variation of c. I.e. the flow of commodity $i \in K$ can take arbitrary paths and is not restricted to paths shorter than c times the length of the shortest s_i, t_i path.

6.2.1 Number of Bow Edges. Early experiments revealed that the number of edges created for each bow E_e^B (which are then time-expanded in G^D) can be kept small. Even restricting to the two edges of E_e^B with the shortest transit times yields very good results, as hinted in Figure 4. This is mainly due to the subdivision of edges in the preprocessing step of the rate-dependent heuristic.

Because of the good performance for two edges per bow and the considerably smaller size of G^D in the following we concentrate on the setting with two edges.

6.2.2 Number of Levels L. Obviously, it is an interesting question to ask how coarse a discretization of time can be while still providing good results. In Figure 5 on the left the number of levels L is varied from 12 to 40. For this setting from about 30 levels on the improvements are quite small. As L tends to infinity the normalized time horizon given by the FPTAS ("Inflow-dep. t.h. / l.b." in the figure) tends to 1, since the lower and upper bound come arbitrarly close. It seems very promising that the time horizon of the rate-dependent flow over time computed by our heuristic is only about 1.5 times larger than the one computed by the FPTAS for the considerably less realistic model. The runtimes of the two algorithms are on what appear to be parallel lines in a logarithmic scale. This would imply a constant factor between the two, which can be expected, and an exponential runtime. The runtime is not asymptotical however, at least if a standard LP formulation (without column generation and constraint shortest path computations) is used to solve the instances.

6.2.3 Number of Commodities k. Another natural question is to ask how much the (normalized) time-horizon computed by the two algorithms (and the corresponding runtime) varies with the number of commodities sent through the network. Figure 5 on the right shows some experimental results where the demand was increased from 1 to 100 commodities with the number of levels L set to 30. The time horizon normalized by the lower bound (otherwise it would be of course increasing) looks quite stable from 50 commodities on, this is probably the point when the network becomes congested. Interestingly again, also for higher number of commodities the ratio of the rate-dependent flow's time horizon to the inflow-dependent flow's time horizon is about 1.5. As observered already when varying the levels, the runtimes of the two algorithms appear to be a constant factor apart.

In an attempt to assess the quality and the degree of realism of the two different types of flows over time, we also measured the "smoothness" of the flows and how much the edge capacities are potentially violated; both also in dependence of the number of commodities. Smoothness we measure as the average of the absolute change of flow rates over time on an edge. By construction in the inflow-dependent flows over time computed by the FPTAS the flow values on an edge change frequently. This is confirmed by the experimental results shown in Figure 6 on the left. The per capacity flow rates of the inflow-dependent flows change a lot more than the rates of the rate-dependent flows. In applications concerning road traffic it might be more desireable that a flow is rather smoothly spread out instead of having rapid oscillations between flow peaks and no or little traffic.

An obviously large drawback of the inflow-dependent setting is that theoretically the flow rates along the edge could arbitrarily exceed its capacity, as pointed out in Section 2. In Figure 6 on the right it can be seen that this actually happens in practice, when computing flows with the FPTAS. For more than 10 commodities the maximum factor by which a capacity is violated is larger than 4, which is rather high. It is also very interesting to note that the average flow rates in the rate-dependent setting are considerably less than the ones in the

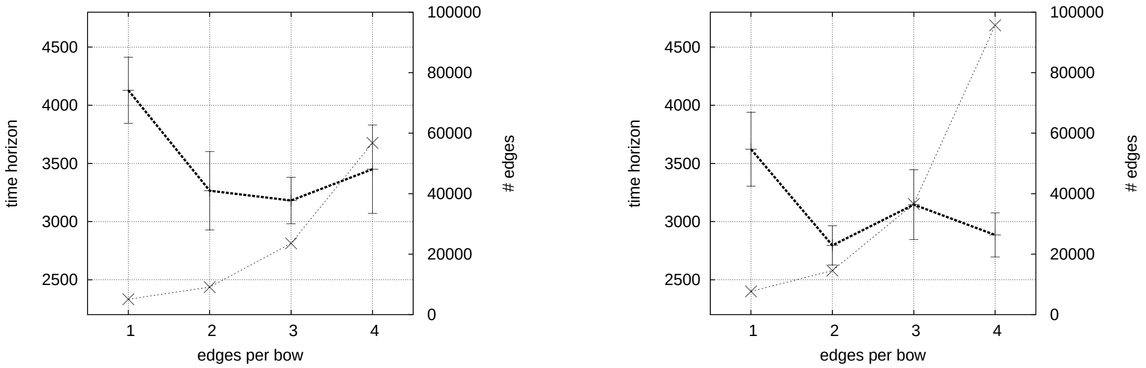

Figure 4: Variation of the number of edges created for each bow E_e^B. On the left for $L = 15$ levels and on the right for $L = 20$ levels. The number of generated commodities is set to 50. The thick lines show the (absolute) time horizon and the thin lines the number of edges in G^D which increase drastically with $|E_e^B|$.

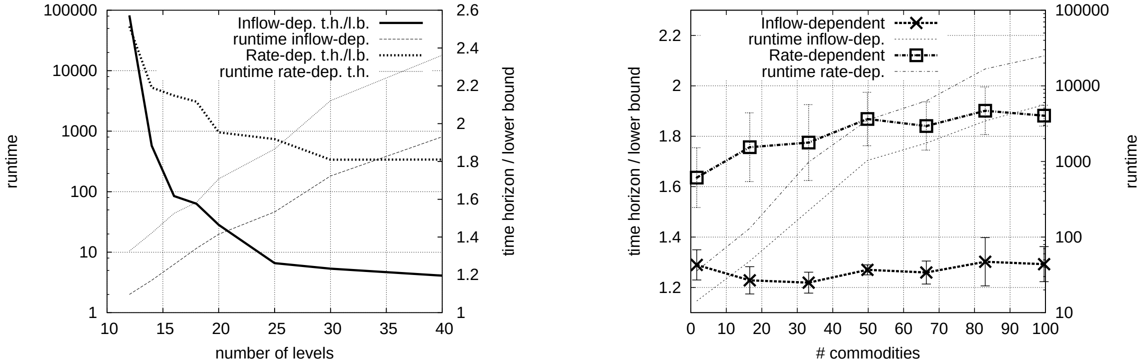

Figure 5: Left: Variation of the number of levels L, with 50 commodities. The time horizons of the inflow-dependent flows over time (FPTAS) and the rate-dependent flows over time (heuristic) are shown, both normalized by the lower bound. Additionally the runtime in seconds of both algorithms is given. Right: Variation of the number of commodities, for $L = 30$ levels. The time horizons shown are normalized by the lower bound.

inflow-dependent setting. This means that in the flows computed by the heuristic the risk of congestion and traffic jams will be a lot less, if the flow routes were to be used for a route guidance system.

6.2.4 Constrained Shortest Paths. For Figure 7, left, the factor was varied by which the length of a flow path for commodity $i \in K$ can exceed the length of the shortest s_i, t_i path. It is quite interesting that if commodities are only allowed to take detours of length at most 1.6 times their shortest path, the resulting flows over time have about the same time horizons as when the paths are not constrained. This is very promising concerning the degree of acceptance of potential route guidance systems based on such approaches.

6.2.5 Large Instances. To see how our heuristic performs on larger instances we ran it on eight different graphs representing increasingly bigger portions of Berlin; their sizes (the number of nodes in the second line, the number of edges in the third):

1	2	3	4	5	6	7	8
56	166	320	538	749	1095	3500	4000
72	238	472	867	1224	1801	8745	8745

The resulting time horizons are shown in Figure 7 on the right. The runtime increases drastically, but note that e.g. the diamond graph created for graph number 8 has 90 million edges and the total computation time for the flow is 17 hours, which seems quite acceptable for instances this large. Since there is still plenty of room for improvements and optimizations concerning the code we consider these results very promising.

7 Conclusion

We introduced the rate-dependent model for flows over time which more realistically captures the behavior of road traffic than previous approaches. In several

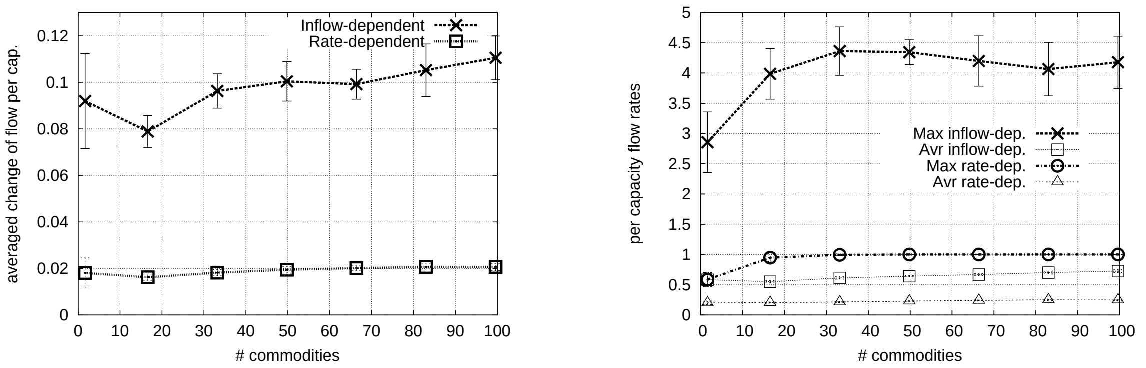

Figure 6: On the left the smoothness of the flows in the two different models is shown for increasing number of commodities. The average (over time) of the absolute values of changes in flow rates is depicted. On the right the maximum and average per capacity flow rates are shown for both models. A value greater then one corresponds to a violation of the capacities. The averages are taken only over non zero flow rates. L is set to 30 in both cases.

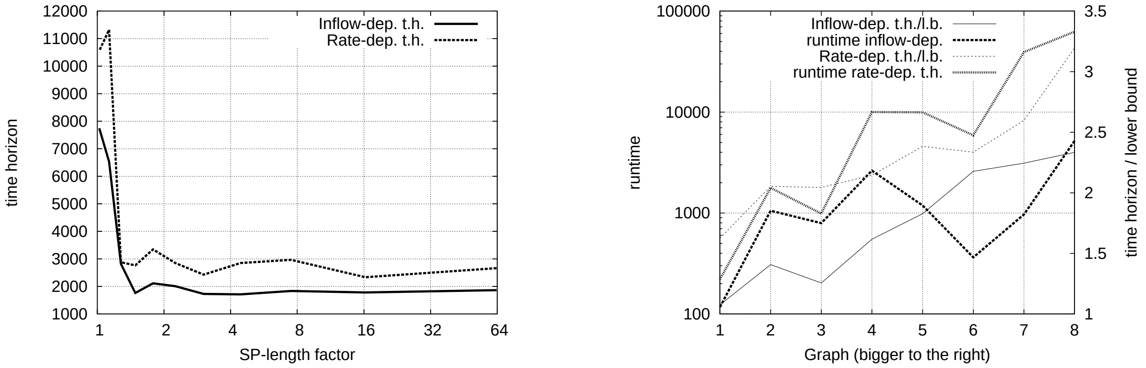

Figure 7: In the diagram on the left the factor by which the length of a shortest path can be exceeded is varied, for $L = 30$ levels. On the right the (normalized) time horizon and the runtime are given for 8 different instance sizes (only one run for each), $L = 20$, and 100 commodities, see text for details.

experiments we compare a heuristic we proposed with an FPTAS for the considerably less realistic inflow-dependent setting (among other things, it turns out that the FPTAS actually computes flows which violate edge capacities). The flows derived by the heuristic are "smoother"—i.e. the flow is spread out more evenly—which might lead to less congested traffic flows, when applied in practice. Nevertheless, the time horizons of the flows computed by the heuristic come close to lower bounds computed for the inflow-dependent model and already now, relatively large instances are possible. Optimizations as for instance not creating the whole diamond graph explicitly (but rather "on demand" when flow is to be routed along a path) could further increase the performance significantly.

For future work it is planned to cooperate with the group of Kai Nagel (ETH Zurich/ TU Berlin) with the goal of feeding flow paths obtained by the heuristic into their large scale traffic simulator MATSIM [20]. From a theoretical point of view it would be of great interest to find approximation algorithms for the quickest flow problem in the rate-dependent model and to investigate whether the model could possibly be enhanced to incorporate the "first in first out" characteristic of single lane roads. Currently flow of different commodities can overtake each other (which is also called the freeway model).

Acknowledgements. The authors would like to thank Thomas Erlebach for many fruitful discussions on the topic of flows over time.

References

[1] Y. P. Aneja, V. Aggarwal, and K. P. K. Nair, *Shortest chain subject to side constraints*, Networks, 13 (1983), pp. 295–302.

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

[3] M. Ben-Akiva, H. Koutsopoulos, R. Mishalani, and Q. Yang, *Simulation laboratory for evaluating dynamic traffic managment systems*, ASCE Journal of Transportation Engineering, (1997), pp. 283–289.

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

[5] C. F. Daganzo, *Fundamentals of transportation and traffic operations*, Pergamon, Oxford, 1997.

[6] M. Desrochers and F. Soumis, *A generalized permanent labeling algorithm for the shortest path problem with time windows*, INFORMS, 26 (1988), pp. 191–212.

[7] S. P. Dirkse and M. C. Ferris, *Traffic modeling and variational inequalities using gams*, in Operations Research and Decision Aid Methodologies in Traffic and Transportation Managment, P. L. Toint, M. Labbe, K. Tanczos, and G. Laporte, eds., vol. 166 of NATO ASI Series, Springer, 1998, pp. 136–163.

[8] L. Fleischer and M. Skutella, *The quickest multicommodity flow problem*, in Integer Programming and Combinatorial Optimization, W. J. Cook and A. S. Schulz, eds., vol. 2337 of LNCS, Springer-Verlag, 2002, pp. 36–53.

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

[10] ———, *Flows in Networks*, Princeton University Press, Princeton, New Jersey, 1962.

[11] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, New York, 1979.

[12] B. D. Greenshields, *A study in highway capacity*, Proceedings of the Highway Research Board, 14 (1935), pp. 458–.

[13] A. Hall, S. Hippler, and M. Skutella, *Multicommodity flows over time: Efficient algorithms and complexity*, in Proceedings of the 30th International Colloquium on Automata, Languages and Programming (ICALP), vol. 2719 of LNCS, Springer-Verlag, 2003, pp. 397–409.

[14] A. Hall, K. Langkau, and M. Skutella, *An FPTAS for quickest multicommodity flows with inflow-dependent transit times*, in Proceedings of the 6th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX), vol. 2764 of LNCS, Springer-Verlag, 2003, pp. 71–82.

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

[16] ILOG, *CPLEX optimizer, version 9.0. http://www.ilog.com/products/cplex.*

[17] Intel, *C++ compiler, version 8.0. http://www.intel.com/software/products/compilers/linux.*

[18] I. Ioachim, S. Gélinas, F. Soumis, and J. Desrosiers, *A dynamic programming algorithm for the shortest path problem with time windows and linear node costs*, Networks, 31 (1998), pp. 193–204.

[19] E. Köhler and M. Skutella, *Flows over time with load-dependent transit times*, in Proceedings of the 13th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA), San Francisco, California, January 2002, pp. 174–183.

[20] MATSIM, *MATSIM: multi-agent traffic simulations, http://www.matsim.org.*

[21] K. Nagel, J. Esser, and M. Rickert, *Large-scale traffic simulations for transportation planning.*, in Annual Review of Computational Physics, D. Stauffer, ed., vol. VII, World Scientific Publishing Company, 2000.

[22] C. A. Phillips, *The network inhibition problem*, in Proceedings of the Twenty-Fifth Annual ACM Symposium on the Theory of Computing, 1993, pp. 776–785.

[23] W. B. Powell, P. Jaillet, and A. Odoni, *Stochastic and dynamic networks and routing*, in Network Routing, M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser, eds., vol. 8 of Handbooks in Operations Research and Management Science, North–Holland, Amsterdam, The Netherlands, 1995, ch. 3, pp. 141–295.

[24] I. Prigogine and R. Herman, *Kinetic Theory of Vehicular Traffic*, Elsevier Science B. V., Amsterdam, 1971.

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

[26] A. Warburton, *Approximation of pareto optima in multiple-objective, shortest-path problems*, Operations Research, 35 (1987), pp. 70–79.