A finite time combinatorial algorithm for instantaneous dynamic equilibrium flows

Lukas Graf, Tobias Harks

Angaben zur Veröffentlichung / Publication details:

Graf, Lukas, and Tobias Harks. 2021. “A finite time combinatorial algorithm for instantaneous dynamic equilibrium flows.” *Integer Programming and Combinatorial Optimization: 22nd International Conference, IPCO 2021, Atlanta, GA, USA, May 19–21, 2021, Proceedings*, 104–18. https://doi.org/10.1007/978-3-030-73879-2_8.

A Finite Time Combinatorial Algorithm for Instantaneous Dynamic Equilibrium Flows

Lukas Graf(✉) and Tobias Harks

Augsburg University, Institute of Mathematics, Augsburg 86135, Germany
{lukas.graf,tobias.harks}@math.uni-augsburg.de

Abstract. Instantaneous dynamic equilibrium (IDE) is a standard game-theoretic concept in dynamic traffic assignment in which individual flow particles myopically select en route currently shortest paths towards their destination. We analyze IDE within the Vickrey bottleneck model, where current travel times along a path consist of the physical travel times plus the sum of waiting times in all the queues along a path. Although IDE have been studied for decades, no exact finite time algorithm for equilibrium computation is known to date. As our main result we show that a natural extension algorithm needs only finitely many phases to converge leading to the first finite time combinatorial algorithm computing an IDE. We complement this result by several hardness results showing that computing IDE with natural properties is NP-hard.

1 Introduction

Flows over time or dynamic flows are an important mathematical concept in network flow problems with many real world applications such as dynamic traffic assignment, production systems and communication networks (e.g., the Internet). In such applications, flow particles that are sent over an edge require a certain amount of time to travel through each edge and when routing decisions are being made, the dynamic flow propagation leads to later effects in other parts of the network. A key characteristic of such applications, especially in traffic assignment, is that the network edges have a limited flow capacity which, when exceeded, leads to congestion. This phenomenon can be captured by the *fluid queueing model* due to Vickrey [23]. The model is based on a directed graph $G = (V, E)$, where every edge e has an associated physical transit time $\tau_e \in \mathbb{Q}_{>0}$ and a maximal rate capacity $\nu_e \in \mathbb{Q}_{>0}$. If flow enters an edge with higher rate than its capacity, the excess particles start to form a queue at the edge's tail, where they wait until they can be forwarded onto the edge. Thus, the total travel time experienced by a single particle traversing an edge e is the sum of the time spent waiting in the queue of e and the physical transit time τ_e.

This physical flow model then needs to be enhanced with a *behavioral model* prescribing the actions of flow particles. There are two main standard behavioral models in the traffic assignment literature known as *dynamic equilibrium (DE)*

© Springer Nature Switzerland AG 2021
M. Singh and D. P. Williamson (Eds.): IPCO 2021, LNCS 12707, pp. 104–118, 2021.
https://doi.org/10.1007/978-3-030-73879-2_8

(cf. Ran and Boyce [19, Sect. V–VI]) and *instantaneous dynamic equilibrium (IDE)* ([19, Sect. VII–IX]). Under DE, flow particles have complete information on the state of the network for all points in time (including the future evolution of all flow particles) and based on this information travel along a shortest path. The full information assumption is usually justified by assuming that the game is played repeatedly and a DE is then an attractor of a learning process. In an IDE, at every point in time and at every node of the graph, flow particles only enter those edges that lie on a currently shortest path towards their respective sink. The behavioral model of IDE is based on the concept that drivers are informed in real-time about the current traffic situation and, if beneficial, reroute instantaneously no matter how good or bad that route will be in hindsight. IDE has been proposed already in the late 80's (cf. Boyce, Ran and LeBlanc [1,20] and Friesz et al. [7]).

A line of recent works starting with Koch and Skutella [17] and Cominetti, Correa and Larré [3] derived a complementarity description of DE flows via so-called *thin flows with resetting* which leads to an α-extension property stating that for any equilibrium up to time θ, there exists $\alpha > 0$ so that the equilibrium can be extended to time $\theta + \alpha$. An extension that is maximal with respect to α is called a *phase* in the construction of an equilibrium and the existence of equilibria on the whole $\mathbb{R}_{\geq 0}$ then follows by a limit argument over phases using Zorn's lemma. In the same spirit, Graf, Harks and Sering [10] established a similar characterization for IDE flows and also derived an α-extension property.

For both models (DE or IDE), it is an open question whether for constant inflow rates and a finite time horizon, a *finite number of phases* suffices to construct an equilibrium, see [3,10,17]. Proving finiteness of the number of phases would imply an exact finite time combinatorial algorithm. Such an algorithm is not known to date neither for DE nor for IDE.[1] More generally, the computational complexity of equilibrium computation is widely open.

1.1 Our Contribution and Proof Techniques

In this paper, we study IDE flows and derive algorithmic and computational complexity results. As our main result we settle the key question regarding finiteness of the α-extension algorithm.

Theorem 1: For single-sink networks with piecewise constant inflow rates with bounded support, there is an α-extension algorithm computing an IDE

[1] Algorithms for DE or IDE computation used in the transportation science literature are *numerical*, that is, only approximate equilibrium flows are computed given a certain numerical precision using a discretized model, see for example [1,6,11]. While a recent computational study [24] showed some positive results in regards to convergence for DE, Otsubo and Rapoport [18] also reported "significant discrepancies" between the continuous and a discretized solution for the Vickrey model.

after finitely many extension phases. This implies the first finite time combinatorial exact algorithm computing IDE within the Vickrey model.

The proof of our result is based on the following ideas. We first consider the case of *acyclic* networks and use a topological order of nodes in order to schedule the extension phases in the algorithm. The key argument for the finiteness of the number of extension phases is that for a single node v and any interval with linearly changing distance labels of nodes closer to the sink and constant inflow rate into v, this flow can be redistributed to the outgoing edges in a finite number of phases of constant outflow rates from v. We show this using the properties (derivatives) of suitable edge label functions for the outgoing edges. The overall finiteness of the algorithm follows by induction over the nodes and time. We then generalize to arbitrary single-sink networks by considering *dynamically* changing topological orders depending on the current set of active edges.

We then turn to the computational complexity of IDE flows and show that several natural decision problems about the existence of IDE with certain properties are NP-hard.

Theorem 2: The following decision problems are all NP-hard:

– Given a specific edge: Is there an IDE using/not using this edge?
– Given some time horizon T: Is there an IDE that terminates before T?
– Given some $k \in \mathbb{N}$: Is there an IDE with at most k phases?

1.2 Related Work

The concept of flows over time was studied by Ford and Fulkerson [5]. Shortly after, Vickrey [23] introduced a game-theoretic variant using a deterministic queueing model. Since then, dynamic equilibria have been studied extensively in the transportation science literature, see Friesz et al. [7]. New interest in this model was raised after Koch and Skutella [17] gave a novel characterization of dynamic equilibria in terms of a family of static flows (thin flows). Cominetti et al. [3] refined this characterization and Sering and Vargas-Koch [22] incorporated spillbacks in the fluid queuing model. In a very recent work, Kaiser [16] showed that the thin flows needed for the extension step in computing dynamic equilibria can be determined in polynomial time for series-parallel networks. The papers [3,16] explicitly mention the problem of possible non-finiteness of the extension steps. For further results regarding a discrete packet routing model, we refer to Cao et al. [2], Ismaili [14,15], Scarsini et al. [21], Harks et al. [12] and Hoefer et al. [13].

2 Model and the Extension-Algorithm

In this paper we consider networks $\mathcal{N} = (G, (\nu_e)_{e\in E}, (\tau_e)_{e\in E}, (u_v)_{v\in V\setminus\{t\}}, t)$ given by a directed graph $G = (V, E)$, edge capacities $\nu_e \in \mathbb{Q}_{>0}$, edge travel times $\tau_e \in \mathbb{Q}_{>0}$, and a single sink node $t \in V$ which is reachable from anywhere in the graph. Each node $v \in V \setminus \{t\}$ has a corresponding (network) inflow rate $u_v : \mathbb{R}_{\geq 0} \to \mathbb{Q}_{\geq 0}$ indicating for every time $\theta \in \mathbb{R}_{\geq 0}$ the rate $u_v(\theta)$ at which the infinitesimal small agents enter the network at node v and start traveling through the graph until they leave the network at the common sink node t. We will assume that these network inflow rates are right-constant step functions with bounded support and finitely many, rational jump points.

A *flow over time* in $\mathcal{N}$ is a tuple $f = (f^+, f^-)$ where $f^+, f^- : E \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ are integrable functions. For any edge $e \in E$ and time $\theta \in \mathbb{R}_{\geq 0}$ the value $f_e^+(\theta)$ describes the *(edge) inflow rate* into e at time θ and $f_e^-(\theta)$ is the *(edge) outflow rate* from e at time θ. For any such flow over time f we define the *cumulative (edge) in- and outflow rates* F^+ and F^- by $F_e^+(\theta) := \int_0^\theta f_e^+(\zeta)d\zeta$ and $F_e^-(\theta) := \int_0^\theta f_e^-(\zeta)d\zeta$, respectively. The queue length of edge e at time θ is then defined as $q_e(\theta) := F_e^+(\theta) - F_e^-(\theta + \tau_e)$.

Such a flow f is called a *feasible flow* in $\mathcal{N}$, if it satisfies the following constraints (1) to (4). The *flow conservation constraints* are modeled for all nodes $v \neq t$ as

$$\sum_{e\in\delta_v^+} f_e^+(\theta) - \sum_{e\in\delta_v^-} f_e^-(\theta) = u_v(\theta) \quad \text{for all } \theta \in \mathbb{R}_{\geq 0}, \tag{1}$$

where $\delta_v^+ := \{vu \in E\}$ and $\delta_v^- := \{uv \in E\}$ are the sets of outgoing edges from v and incoming edges into v, respectively. For the sink node t we require

$$\sum_{e\in\delta_t^+} f_e^+(\theta) - \sum_{e\in\delta_t^-} f_e^-(\theta) \leq 0 \tag{2}$$

and for all edges $e \in E$ we always assume

$$f_e^-(\theta) = 0 \text{ for all } \theta < \tau_e. \tag{3}$$

Finally we assume that the queues operate at capacity which can be modeled by

$$f_e^-(\theta + \tau_e) = \begin{cases} \nu_e, & \text{if } q_e(\theta) > 0 \\ \min\{f_e^+(\theta), \nu_e\}, & \text{if } q_e(\theta) \leq 0 \end{cases} \quad \text{for all } e \in E, \theta \in \mathbb{R}_{\geq 0}. \tag{4}$$

Following the definition in [10] we call a feasible flow an IDE (flow) if whenever a particle arrives at a node $v \neq t$, it can only ever enter an edge that is the first edge on a currently shortest v-t path. In order to formally describe this property we first define the *current* or *instantaneous travel time* of an edge e at θ by

$$c_e(\theta) := \tau_e + \frac{q_e(\theta)}{\nu_e}. \tag{5}$$

We then define time dependent node labels $\ell_v(\theta)$ corresponding to current shortest path distances from v to the sink t. For $v \in V$ and $\theta \in \mathbb{R}_{\geq 0}$, define

$$\ell_v(\theta) := \begin{cases} 0, & \text{for } v = t \\ \min\limits_{e=vw \in E} \{\ell_w(\theta) + c_e(\theta)\}, & \text{else.} \end{cases} \tag{6}$$

We say that an edge $e = vw$ is *active* at time θ, if $\ell_v(\theta) = \ell_w(\theta) + c_e(\theta)$, denote the set of active edges by $E_\theta \subseteq E$ and call the subgraph induced by E_θ the *active subgraph at time* θ.

Definition 1. A feasible flow over time f is an *instantaneous dynamic equilibrium (IDE)*, if for all $\theta \in \mathbb{R}_{\geq 0}$ and $e \in E$ it satisfies

$$f_e^+(\theta) > 0 \Rightarrow e \in E_\theta. \tag{7}$$

During the computation of an IDE we also temporarily need the concept of a *partial IDE up to some time* $\hat{\theta}$. This is a flow f such that constraints (1) to (4) as well as constraint (7) only hold for all $\theta \in [0, \hat{\theta})$, while $f_e^+(\theta) = f_e^-(\theta + \tau_e) = 0$ for all $\theta \geq \hat{\theta}$. For any such flow, we then define the *gross node inflow rates* b_v^- by setting $b_v^-(\theta) := \sum_{e \in \delta_v^-} f_e^-(\theta) + u_v(\theta)$ for all $v \in V \setminus \{ t \}$ and $\theta \in [\hat{\theta}, \hat{\theta} + \tau_{\min})$, where $\tau_{\min} := \min \{ \tau_e \mid e \in E \} > 0$.

As shown in [10, Sect. 3] such a partial IDE can always be extended for some additional proper[2] time interval on a node by node basis using constant edge inflow rates. The existence of IDE for the whole $\mathbb{R}_{\geq 0}$ then follows by applying Zorn's lemma. This also leads to the following natural algorithm for computing IDE in single-sink networks:

1. Start with the zero-flow f – a partial IDE up to time 0.
2. While f is not an IDE for all times, extend f for some additional interval.

In the extension step, we first determine a topological order of the nodes in the active subgraph (e.g. sort the nodes w.r.t. to their current node labels ℓ_v). Then we go through the nodes in this order (beginning with the sink node t) and at each node determine a constant distribution of the current gross node inflow rate to the outgoing active edges in such a way that the used edges remain active for some proper time interval into the future. Finally, we take the smallest of these intervals and extend the whole partial IDE over it. For the extension at a single node v at some time θ, we can use a solution to the following convex optimization problem, which can be determined in polynomial time using a simple water filling procedure (see [10, Algorithm 1 (electronic supplementary material)]):

$$\begin{aligned} \min \quad & \sum_{e=vw \in \delta_v^+ \cap E_\theta} \int_0^{x_e} \frac{g_e(z)}{\nu_e} + \partial_+ \ell_w(\theta) dz \qquad (\text{OPT-}b_v^-(\theta)) \\ \text{s.t.} \quad & \sum_{e \in \delta_v^+ \cap E_\theta} x_e = b_v^-(\theta), \quad x_e \geq 0 \text{ for all } e \in \delta_v^+ \cap E_\theta, \end{aligned}$$

[2] We call an interval $[a, b)$ *proper* if $a < b$.

where ∂_+ denotes the right side derivative and $g_e(z) := z - \nu_e$, if $q_e(\theta) > 0$, and $g_e(z) := \max\{\, z - \nu_e, 0 \,\}$, otherwise. Any solution to (OPT-$b_v^-(\theta)$) corresponds to a flow distribution to active edges so that for every edge $e = vw \in \delta_v^+ \cap E_\theta$ the following condition is satisfied (see [10, Lemma 3.1])

$$\begin{aligned} f_e^+(\theta) > 0 \quad &\implies \quad \partial_+\ell_v(\theta) = \partial_+c_e(\theta) + \partial_+\ell_w(\theta) \\ f_e^+(\theta) = 0 \quad &\implies \quad \partial_+\ell_v(\theta) \leq \partial_+c_e(\theta) + \partial_+\ell_w(\theta). \end{aligned} \tag{8}$$

Because the network inflow rates as well as all already constructed edge inflow rates are piecewise constant and the node label functions as well as the queue length functions are continuous, the used edges will remain active for some proper time interval. As IDE flows in single-sink networks always have a finite termination time ([10, Theorem 4.6]) it suffices to extend the flow for some finite time horizon (in [9] we even provide a way to explicitly compute such a time horizon). Thus, the only possible obstruction for the extension algorithm to terminate within finite time is some Zeno-type behavior of the lengths of the extension phases, e.g. some sequence of extension phases of lengths $\alpha_1, \alpha_2, \dots$ such that $\sum_{i=1}^{\infty} \alpha_i$ converges to some point strictly before the IDE's termination time. In fact, in the full version of this paper [8], we provide an example of a rather simple network wherein extension phases may indeed become arbitrarily small, provided a long enough lasting network inflow rate.[3] However, this is not a counter example to the finiteness of the extension algorithm, as the shrinking of the extension phases is slow enough to still allow for a finite number of phases to span any fixed time horizon. In the following section we will show that this is in fact true for *all* single-sink networks, i.e. that we can reach any given time horizon within a finite number of phases.

3 Finite IDE-Construction Algorithm

For the proof of our main theorem we will employ two reductions: First, we observe that for acyclic networks, it suffices to consider a single node with constant gross node inflow rate for a given interval and linear node label functions at all the nodes reachable via a single edge from this node. For this situation, we show that the incoming flow can be distributed over active edges using a finite number of phases. Second, we argue that for general networks, we can group the extension phases into finitely many larger intervals such that during each such interval the extension algorithm only has to consider a certain fixed acyclic subgraph (reducing to the first case).

[3] This example also shows that the number of (distinct!) extension phases can be exponential in the encoding size of the given instance and that networks with forever lasting network inflow rates may require IDE flows which never reach a stable state. An exponential number of extension phases has also been observed in DE flows while stable states are always reached there (see [4]).

Acyclic Networks. Due to our first reduction, which we will justify afterwards, the proof for acyclic networks essentially rests on the following key lemma.

Lemma 1. *Let $\mathcal{N}$ be a single-sink network on an acyclic graph with some fixed topological order on the nodes, v some node in $\mathcal{N}$ and $\theta_1 < \theta_2 \leq \theta_1 + \tau_{\min}$ two times. If f is a flow over time in $\mathcal{N}$ such that*

- *f is a partial IDE up to time θ_1 for nodes at least as far away from t as v,*
- *f is a partial IDE up to time θ_2 for nodes closer to t than v,*
- *during $[\theta_1, \theta_2)$ the gross node inflow rate into v is constant and*
- *the label functions at the nodes reachable via direct edges from v are linear on this interval,*

then we can extend f to a partial IDE up to θ_2 at v in a finite number of phases.

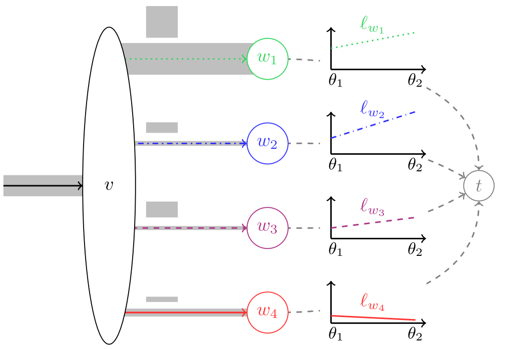

Fig. 1. The situation in Lemma 1: We have an acyclic graph and a partial IDE up to some time θ_2 for all nodes closer to the sink t than v and up to some earlier time θ_1 for v and all nodes further away than v from t. Additionally, over the interval $[\theta_1, \theta_2)$ the edges leading into v have a constant outflow rate (and a physical travel time of at least $\theta_2 - \theta_1$) and the nodes w_i all have affine label functions ℓ_{w_i}. The edges vw_i start with some current queue lengths $q_{vw_i}(\theta_1) \geq 0$.

Proof. Let f be the flow after an, a priori, infinite number of maximal extension steps getting us to a partial IDE up to some $\hat{\theta} \in (\theta_1, \theta_2]$ at node v. Furthermore, let $\delta_v^+ = \{ vw_1, \ldots, vw_p \}$ be the set of outgoing edges from v. Then for every such edge vw_i we can define a function $h_i : [\theta_1, \hat{\theta}) \to \mathbb{R}_{\geq 0}, \theta \mapsto c_{vw_i}(\theta) + \ell_{w_i}(\theta)$, denoting for every time $\theta \in [\theta_1, \hat{\theta})$ the shortest current travel time to the sink t for a particle entering edge vw_i at that time. Consequently, during this interval we have $vw_i \in E_\theta$ if and only if $h_i(\theta) = \min \{ h_j(\theta) \mid j \in [p] \} = \ell_v(\theta)$. We start by stating several important observations and then proceed by showing two key-properties of the functions h_i and ℓ_v, which are also visualized in Figs. 2 and 3:

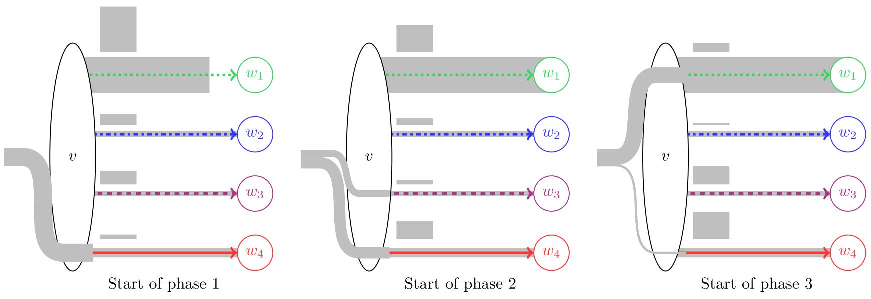

Fig. 2. The first three phases of a possible flow distribution from the node v for the situation depicted in Fig. 1. The corresponding functions h_i are depicted in Fig. 3

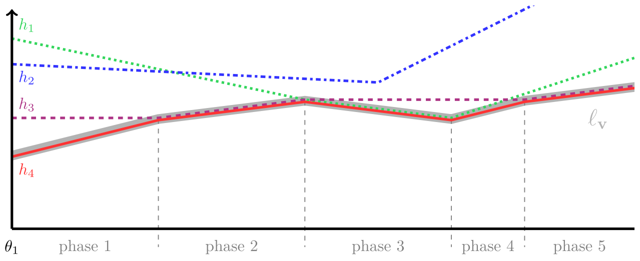

Fig. 3. The functions h_i corresponding to the flow distribution for the situation depicted in Fig. 1 and depicted in Fig. 2 for the first three phases. The second, third and fifth phase start because an edge becomes newly active (edges vw_3, vw_1 and vw_3 again, respectively). The fourth phase starts because the queue on the active edge vw_1 runs empty. These are the only two possible events which can trigger the beginning of a new phase. Edge vw_2 is inactive for the whole time interval and, thus, has a convex graph. The bold gray line marks the graph of the function ℓ_v.

(i) The functions h_i are continuous and piece-wise linear. In particular they are differentiable almost everywhere and their left and right side derivatives $\partial_- h_i$ and $\partial_+ h_i$, respectively, exist everywhere. The same holds for the function ℓ_v.
(ii) A new phase begins at a time $\theta \in [\theta_1, \hat{\theta})$ if and only if at least one of the following two events occurs at time θ: An edge vw_i becomes newly active or the queue of an active edge vw_i runs empty.
(iii) There are uniquely defined numbers $\ell_{I,J}$ for all subsets $J \subseteq I \subseteq [p]$ such that $\ell'_v(\theta) = \ell_{I,J}$ within all phases, where $\{ vw_i \mid i \in I \}$ is the set of active edges in δ_v^+ and $\{ vw_i \mid i \in J \}$ is the subset of such active edges that also have a non-zero queue during this phase.

Claim 1. *If an edge vw_i is inactive during some interval $(a, b) \subseteq [\theta_1, \hat{\theta})$, the graph of h_i is convex on this interval.*

Claim 2. *For any time θ define $I(\theta) := \{\, i \in [p] \mid h_i(\theta) = \ell_v(\theta) \,\}$. Then, we have*

$$\min\{\, \partial_- h_i(\theta) \mid i \in I(\theta) \,\} \leq \partial_+ \ell_v(\theta). \tag{9}$$

If no edge becomes newly active at time θ, we also have $\partial_- \ell_v(\theta) \leq \partial_+ \ell_v(\theta)$.

Proof of Claim 1. By the lemma's assumption ℓ_{w_i} is linear on the whole interval. For an inactive edge vw_i its queue length function consists of at most two linear sections: One where the queue depletes at a constant rate of $-\nu_e$ and one where it remains constant 0. Thus, h_i is convex as sum of two convex functions. ■

Proof of Claim 2. To show (9), let I' be the set of indices of edges active immediately after θ, i.e. $I' := \{\, i \in I(\theta) \mid \partial_+ h_i(\theta) = \partial_+ \ell_v(\theta) \,\}$. Since the total outflow from node v is constant during $[\theta_1, \hat{\theta})$ and flow may only enter edges vw_i with $i \in I'$ after θ, there exists some $j \in I'$, where the inflow rate into vw_j after θ is the same or larger than before. But then we have $\partial_+ h_j(\theta) \geq \partial_- h_j(\theta)$ and, thus,

$$\min\{\, \partial_- h_i(\theta) \mid i \in I(\theta) \,\} \leq \partial_- h_j(\theta) \leq \partial_+ h_j(\theta) = \partial_+ \ell_v(\theta).$$

If, additionally, no edge becomes newly active at time θ, all edges vw_i with $i \in I'$ have been active directly before θ as well implying

$$\partial_- \ell_v(\theta) = \min\{\, \partial_- h_i(\theta) \mid i \in I(\theta) \,\} \overset{(9)}{\leq} \partial_+ \ell_v(\theta).$$

■

Using these properties we can now first show a claim which implies that the derivative of ℓ_v can attain the smallest $\ell_{I,J}$ only for a finite number of intervals. Inductively the same then holds for all of the finitely many $\ell_{I,J}$, which by observation (iii) are the only values ℓ'_v can attain. The proof of the lemma finally concludes by observing that an interval with constant derivative of ℓ_v can contain only finitely many phases.

Claim 3. *Let $(a_1, b_1), (a_2, b_2) \subseteq [\theta_1, \hat{\theta})$ be two disjoint maximal non-empty intervals with constant $\ell'_v(\theta) =: c$. If $b_1 < a_2$ and $\ell'_v(\theta) \geq c$ for all $\theta \in (b_1, a_2)$ where the derivative exists, then there exists an edge vw_i such that*

1. *the first phase of (a_2, b_2) begins because vw_i becomes newly active and*
2. *this edge is not active between a_1 and a_2.*

In particular, the first phase of (a_1, b_2) is not triggered by vw_i becoming active.

Claim 4. *Let $(a, b) \subseteq [\theta_1, \hat{\theta})$ be an interval during which ℓ'_v is constant. Then (a, b) contains at most $2p$ phases.*

Proof of Claim 3. Since we have $\partial_+ \ell_v(a_2) = c$, Claim 2 implies that there exists some edge vw_i with $h_i(a_2) = \ell_v(a_2)$ and $\partial_- h_i(a_2) \leq c$. As (a_2, b_2) was chosen to be maximal and $\ell'_v(\theta) \geq c$ holds almost everywhere between b_1 and a_2, we have $\partial_- \ell_v(a_2) > c$. Thus, vw_i was inactive before a_2.

Now let $\tilde{\theta} < a_2$ be the last time before a_2, where vw_i was active. By Claim 1 we know then that $h_i'(\theta) \leq c$ holds almost everywhere on $[\tilde{\theta}, a_2]$. At the same time we have $\ell_v'(\theta) \geq c$ almost everywhere on $[a_1, a_2]$ and $\ell_v'(\theta) > c$ for at least some proper subinterval of $[b_1, a_2]$, since the intervals (a_1, b_1) and (a_2, b_2) were chosen to be maximal. Combining these two facts with $\ell_v(\theta) = h_i(\theta)$ implies $\ell_v(\theta) < h_i(\theta)$ for all $\theta \in [\tilde{\theta}, a_2) \cap [a_1, a_2)$. As both functions are continuous we must have $\tilde{\theta} < a_1$. Thus, vw_i is inactive for all of $[a_1, a_2)$. ■

Proof of Claim 4. By Claim 1 an edge that changes from active to inactive during the interval (a, b) will remain inactive for the rest of this interval. Thus, at most p phases can start because an edge becomes newly active. By Claim 3 if a phase begins because the queue on an active edge vw_i runs empty at time θ, we have $\partial_+ h_i(\theta) > \partial_- h_i(\theta) = \partial_- \ell_v(\theta) = \partial_+ \ell_v(\theta)$ meaning that this edge will become inactive. Thus, at most p phases start because the queue of an active edge runs empty. Since by observation (ii) these are the only ways to start a new phase, we conclude that there can be no more than $2p$ phases during (a, b). ■

Combining Claims 3 and 4 we see that $[\theta_1, \hat{\theta})$ only contains a finite number of phases and, thus, we achieve $\hat{\theta} = \theta_2$ with finitely many extensions. □

For acyclic networks we can now fix some topological order of the nodes w.r.t. the whole graph at the beginning of the algorithm and then always do the node-wise extensions in this order. Since in a partial IDE up to time $\hat{\theta}$ the gross node inflow rates are already completely determined for the interval $[\hat{\theta}, \hat{\theta} + \tau_{\min})$ we can – for the purpose of the following analysis – slightly rearrange the extension steps, without changing the outcome of the algorithm, by directly extending the partial IDE at each node for this whole interval (using multiple phases). It then suffices to show that these extensions (of constant length $\tau_{\min} > 0$) at a single node only need a finite amount of phases, which follows by repeatedly applying Lemma 1 and using the fact that, by induction, the gross node inflow rate at the current node as well as the label functions ℓ_{w_i} at all nodes closer to the sink are piece-wise constant and piece-wise linear with finitely many breakpoints, respectively.

General Networks. In order to extend this result to general networks, we introduce the concept of a *lazy set of active edges*, which is a time dependent subset of edges $\tilde{E}(\theta)$ satisfying the following properties:

– At every time θ the set $\tilde{E}(\theta)$ contains all currently active edges but no cycle.
– There are (flow independent) constants $C, D > 0$ such that during any time interval of length at most C the set $\tilde{E}(\theta)$ changes at most D times.

This allows us to subdivide the whole time into a finite number of intervals during which $\tilde{E}(\theta)$ does not change and, during those, we can restrict ourselves to considering only edges in the fixed acyclic subgraph induced by the edges in $\tilde{E}(\theta)$. To obtain such a lazy set of active edges we add edges, whenever they become active, but only remove edges if, otherwise, $\tilde{E}(\theta)$ would contain a cycle. Additionally, in this case we always only remove the "most inactive" edge of the cycle. This leads to the final variant of the extension algorithm which is formalized in Algorithm 1 and for which we will now show our main theorem.

Algorithm 1: IDE-Construction Algorithm for single-sink networks

```
Input: A network $\mathcal{N}$ with piecewise constant network inflow rates
Output: An IDE flow $f$ in $\mathcal{N}$
1  Choose $T$ large enough such that all IDE flows in $\mathcal{N}$ terminate before $T$
2  Let $f$ be the zero flow, set $\theta \leftarrow 0$ and $\tilde{E} \leftarrow E_0$
3  Determine a top. order $t = v_1 < v_2 < \cdots < v_n$ w.r.t. the edges in $\tilde{E}$
4  while $\theta < T$ do
5      Choose the largest $\alpha_0 > 0$ s.th. all $b_v^-$ are constant over $(\theta, \theta + \alpha_0)$
6      for $i = 1, \dots, n$ do
7          Compute a constant distribution to the outgoing active edges from $v_i$
           satisfying (8)
8          Determine the largest $\alpha_i \leq \alpha_{i-1}$ such that the set of active edges does
           not change during $(\theta, \theta + \alpha_i)$
9      end for
10     Extend the flow $f$ up to time $\theta + \alpha_n$ and set $\theta \leftarrow \theta + \alpha_n$
11     if $E_\theta \setminus \tilde{E} \neq \emptyset$ then
12         Define $\tilde{E} \leftarrow \tilde{E} \cup E_\theta$.
13         while there exists a cycle $C$ in $\tilde{E}$ do
14             Remove an edge $e = xy \in C$ with maximal $\ell_y(\theta) - \ell_x(\theta)$
15         end while
16         Find a top. order $t = v_1 < v_2 < \cdots < v_n$ w.r.t. the edges in $\tilde{E}$
17     end if
18 end while
```

Theorem 1. *For any single-sink network with piecewise constant network-inflow rates an IDE flow can be constructed in finite time using the natural extension algorithm (Algorithm 1).*

Proof. As we already have the above theorem for acyclic networks and Algorithm 1 only uses a fixed acyclic subgraph of the whole network as long as the set $\tilde{E}$ used in Algorithm 1 remains unchanged, it suffices to show that this set is indeed a lazy set of active edges. The first property is obvious from the way $\tilde{E}$ is obtained in the algorithm, for the second one we need the following two claims:

Claim 5. *Any edge xy removed from $\tilde{E}$ in line 14 satisfies $\ell_x(\theta) < \ell_y(\theta)$.*

Claim 6. *For any given network there exists some constant $L > 0$ such that for all flows, all nodes v and all times θ we have $|\ell_v'(\theta)| \leq L$.*

Proof of Claim 5. Let $C \subseteq \tilde{E}$ be a cycle containing the removed edge xy. Since $\tilde{E}$ was acyclic before we added the newly active edges in line 12, this cycle also has to contain some edge vw which is currently active, therefore satisfying $\ell_v(\theta) > \ell_w(\theta)$. Thus, summing the differences of the label functions at the two ends of every edge over all edges in C yields the existence of at least one edge $uz \in C$ with $\ell_z(\theta) - \ell_u(\theta) > 0$. This then, in particular, also holds for edge xy. ■

Proof of Claim 6. For any node v we can bound the maximal inflow rate into this node by the constant $L_v := \sum_{e\in\delta_v^-} \nu_e + \max\{\, u_v(\theta) \mid \theta \in \mathbb{R}_{\geq 0} \,\}$ using constraint (4). Together with flow conservation (1) this, in turn, allows us to upper bound the inflow rates into all edges $e \in \delta_v^+$ and, thus, the rate at which the queue length and the current travel time on these edges can change by $L_e := \frac{L_v}{\nu_e}$. Since this rate of change is also lower bounded by -1 setting $L := \sum_{e\in E} \max\{\, 1, L_e \,\}$ proves the claim, as for all nodes v and times θ, we then have $|\ell_v'(\theta)| \leq \sum_{e\in E} |c_e'(\theta)| \leq \sum_{e\in E} \max\{\, 1, L_e \,\} = L$. ■

Combining these two claims with the fact that an edge xy is only added to $\tilde{E}$ if it becomes active, i.e. $\ell_x(\theta) = \ell_y(\theta) + c_{xy}(\theta) \geq \ell_y(\theta) + \tau_{xy}$, shows that no edge can enter $\tilde{E}$ twice during any sufficiently small time interval, which implies the second property concluding the proof of the theorem. □

Remark 1. A closer inspection of the proofs allows us to also derive the following rough but explicit bound on the number of phases needed assuming that all edge travel times and capacities are integers (which can always be achieved by rescaling the network):

$$\mathcal{O}\left(P\Big(2(\varDelta+1)^{4^{\varDelta}+1}\Big)^{D/C\cdot T\cdot|V|}\right).$$

Hereby, $\varDelta := \max\{\, |\delta_v^+| \mid v \in V \,\}$ is the maximum out-degree of any edge, T the termination time of the IDE and P is the number of intervals with constant network inflow rates. A formal deduction of this bound can be found in the full version of this paper [8].

4 Computational Complexity of IDE

While Theorem 1 shows that IDE can be constructed in finite time, the derived explicit bound is clearly superpolynomial. In this section we complement this result by showing that many natural decision problems about IDE are NP-hard.

Theorem 2. *The following decision problems are NP-hard:*

(i) Given a network and a specific edge: Is there an IDE not using this edge?
(ii) Given a network and a specific edge: Is there an IDE using this edge?
(iii) Given a network and a time T: Is there an IDE that terminates before T?
(iv) Given a network and some $k \in \mathbb{N}$: Is there an IDE with at most k phases?

This theorem can be shown by a reduction from the NP-complete problem `3SAT` to the above problems. The main idea of the reduction is as follows: For any given instance of `3SAT` we construct a network which contains a source node for each clause with three outgoing edges corresponding to the three literals of the clause. Any satisfying interpretation of the 3SAT-formula translates to a distribution of the network inflow to the literal edges (where sending at least 1/3 of the flow along an edge corresponds to the respective literal being true), which leads to an IDE flow that passes through the whole network in a straightforward manner. If, on the other hand, the formula is unsatisfiable, every IDE flow will cause a specific type of congestion which will divert a certain amount of flow into a different part of the graph. This part of the graph may contain an otherwise unused edge (for (i)), a gadget blocking access to an otherwise used edge (for (ii)), a gadget which results in a long travel time (for (iii)) or one which produces many phases (for (iv)). The congestion occurs because the flow corresponding to a variable being false forms a queue while the flow corresponding to the same variable being true is delayed. Thus, when the latter finally arrives, the former has already blocked the direct path and diverts the latter away from it. On the other hand, in a flow corresponding to a satisfying interpretation of the formula this does not happen, as only one of the two types of flow is present for every variable. The detailed construction of the described gadgets as well as the formal proof of the reduction's correctness can be found in the full version of this paper [8]. For an illustration of the reduction see Fig. 4.

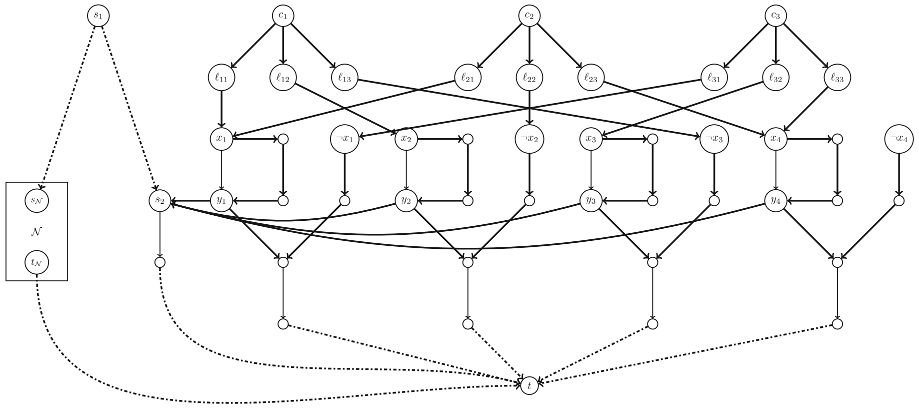

Fig. 4. The whole network for the 3SAT-formula $(x_1 \vee x_2 \vee \neg x_3) \wedge (x_1 \vee \neg x_2 \vee x_4) \wedge (\neg x_1 \vee x_3 \vee x_4)$. The bold edges have infinite capacity, while all other edges have capacity 1. The solid edges have a travel time of 1, the dashdotted edges may have variable travel time (depending on the subnetwork $\mathcal{N}$). The network inflow rates are 12 over the interval $[0, 1]$ at all nodes c_i and 0 everywhere else.

5 Conclusion

We showed that Instantaneous Dynamic Equilibria can be computed in finite time for single-sink networks by applying the natural extension algorithm. We complemented this result by showing that several natural decision problems involving IDE flows are NP-hard by describing a reduction from `3SAT`. One common observation that can be drawn from many proofs involving IDE flows (in this paper as well as in [9,10]) is that they often allow for some kind of *local analysis* of their structure – something which seems out of reach for DE flows. This was a key aspect of the positive result about the finiteness of the extension algorithm where it allowed us to use inductive reasoning over the single nodes of the given network. At the same time, such local argumentation allows us to analyse the behavior of IDE flows in the rather complex instance resulting from the reduction in the hardness-proof by looking at the local behavior inside the much simpler gadgets from which the larger instance is constructed. We think that this local approach to the analysis of IDE flows might also help to answer further open questions about IDE flows in the future. One such topic might be a further investigation of the computational complexity of IDE flows. While both our upper bound on the number of extension steps as well as our lower bound for the worst case computational complexity are superpolynomial bounds, the latter is at least still polynomial in the termination time of the constructed flow, which is not the case for the former. Thus, there might still be room for improvement on either bound.

Acknowledgments. We are grateful to the anonymous reviewers for their valuable feedback on this paper. Additionally, we thank the Deutsche Forschungsgemeinschaft (DFG) for their financial support. Finally, we want to thank the organizers and participants of the 2020 Dagstuhl seminar on "Mathematical Foundations of Dynamic Nash Flows", where we had many helpful and inspiring discussions on the topic of this paper.

Funding. The research of the authors was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - HA 8041/1-2

References

1. Boyce, D.E., Ran, B., LeBlanc, L.J.: Solving an instantaneous dynamic user-optimal route choice model. Transp. Sci. **29**(2), 128–142 (1995)
2. Cao, Z., Chen, B., Chen, X., Wang, C.: A network game of dynamic traffic. In: Daskalakis, C., Babaioff, M., Moulin, H. (eds.) Proceedings of the 2017 ACM Conference on Economics and Computation, EC 2017, Cambridge, MA, USA, 26–30 June 2017, pp. 695–696. ACM (2017)
3. Cominetti, R., Correa, J., Larré, O.: Dynamic equilibria in fluid queueing networks. Oper. Res. **63**(1), 21–34 (2015)
4. Cominetti, R., Correa, J., Olver, N.: Long term behavior of dynamic equilibria in fluid queuing networks. Oper. Res. (2020, to appear). https://doi.org/10.1287/opre.2020.2081
5. Ford, L.R., Fulkerson, D.R.: Flows in Networks. Princeton University Press, Princeton (1962)
6. Friesz, T.L., Han, K.: The mathematical foundations of dynamic user equilibrium. Transp. Res. Part B Methodol. **126**, 309–328 (2019)

7. Friesz, T.L., Luque, J., Tobin, R.L., Wie, B.-W.: Dynamic network traffic assignment considered as a continuous time optimal control problem. Oper. Res. **37**(6), 893–901 (1989)
8. Graf, L., Harks, T.: A finite time combinatorial algorithm for instantaneous dynamic equilibrium flows. https://arxiv.org/abs/2007.07808 (2020)
9. Graf, L., Harks, T.: The price of anarchy for instantaneous dynamic equilibria. In: Chen, X., Gravin, N., Hoefer, M., Mehta, R. (eds.) WINE 2020. LNCS, vol. 12495, pp. 237–251. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64946-3_17
10. Graf, L., Harks, T., Sering, L.: Dynamic flows with adaptive route choice. Math. Program. **183**(1), 309–335 (2020). https://doi.org/10.1007/s10107-020-01504-2
11. Han, K., Friesz, T.L., Yao, T.: A partial differential equation formulation of Vickrey's bottleneck model, part ii: numerical analysis and computation. Transp. Res. Part B Methodol. **49**, 75–93 (2013)
12. Harks, T., Peis, B., Schmand, D., Tauer, B., Vargas-Koch, L.: Competitive packet routing with priority lists. ACM Trans. Econ. Comput. **6**(1), 4:1–4:26 (2018)
13. Hoefer, M., Mirrokni, V.S., Röglin, H., Teng, S.-H.: Competitive routing over time. Theor. Comput. Sci. **412**(39), 5420–5432 (2011)
14. Ismaili, A.: Routing games over time with FIFO policy. In: Devanur, N.R., Lu, P. (eds.) WINE 2017. LNCS, vol. 10660, pp. 266–280. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-71924-5_19
15. Ismaili, A.: The complexity of sequential routing games. CoRR, abs/1808.01080 (2018)
16. Kaiser, M.: Computation of dynamic equilibria in series-parallel networks. Math. Oper. Res. (2020, forthcoming)
17. Koch, R., Skutella, M.: Nash equilibria and the price of anarchy for flows over time. Theory Comput. Syst. **49**(1), 71–97 (2011)
18. Otsubo, H., Rapoport, A.: Vickrey's model of traffic congestion discretized. Transp. Res. Part B Methodol. **42**(10), 873–889 (2008)
19. Ran, B., Boyce, D.E.: Dynamic Urban Transportation Network Models: Theory and Implications for Intelligent Vehicle-Highway Systems. Lecture Notes in Economics and Mathematical Systems. Springer, New York (1996). https://doi.org/10.1007/978-3-662-00773-0
20. Ran, B., Boyce, D.E., LeBlanc, L.J.: A new class of instantaneous dynamic user-optimal traffic assignment models. Oper. Res. **41**(1), 192–202 (1993)
21. Scarsini, M., Schröder, M., Tomala, T.: Dynamic atomic congestion games with seasonal flows. Oper. Res. **66**(2), 327–339 (2018)
22. Sering, L., Vargas-Koch, L.: Nash flows over time with spillback. In: Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms. ACM (2019)
23. Vickrey, W.S.: Congestion theory and transport investment. Am. Econ. Rev. **59**(2), 251–60 (1969)
24. Ziemke, T., Sering, L., Vargas-Koch, L., Zimmer, M., Nagel, K., Skutella, M.: Flows over time as continuous limits of packet-based network simulations. In: Transportation Research Procedia, vol. 52, pp. 123–130 (2021). https://doi.org/10.1016/j.trpro.2021.01.014