Spillback Changes the Long-Term Behavior of Dynamic Equilibria in Fluid Queuing Networks

Theresa Ziemke[1]
Combinatorial Optimization and Graph Algorithms, Technische Universität Berlin, Germany
Transport Systems Planning and Transport Telematics, Technische Universität Berlin, Germany

Leon Sering
Institute for Operations Research, ETH Zürich, Switzerland

Kai Nagel
Transport Systems Planning and Transport Telematics, Technische Universität Berlin, Germany

Abstract

We study the long-term behavior of dynamic traffic equilibria and find that it heavily depends on whether spillback is captured in the traffic model or not. We give an example where no steady state is reached. Although the example consists of a single-commodity instance with constant inflow rate, the Nash flow over time consists of infinitely many phases. This is in contrast to what has been proven for Nash flows over time without spillback [3, 7].

Additionally, we show that similar phase oscillations as in the Nash flow over time with spillback can be observed in the co-evolutionary transport simulation MATSim. This reaffirms the robustness of the findings as the simulation does (in contrast to Nash flows over time) not lead to exact user equilibra and, moreover, models discrete time steps and vehicles.

2012 ACM Subject Classification Computing methodologies → Agent / discrete models; Mathematics of computing → Network flows; Applied computing → Transportation

Keywords and phrases flows over time, transport simulation, Nash flow, dynamic equilibrium, long-term behavior, steady state, oscillation, spillback, MATSim

Digital Object Identifier 10.4230/OASIcs.ATMOS.2023.11

Funding *Theresa Ziemke*: Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – The Berlin Mathematics Research Center MATH+ (EXC-2046/1, project ID: 390685689).

Acknowledgements We thank Max Zimmer for technical work on the Nash flow over time computation tool.

1 Introduction

Reliable transport models are a central instrument to design efficient transport systems that provide accessibilities to locations of interest for people and goods and at the same time reduce the transport system's negative effects like environmental pollution or its significant contribution to climate warming.

Arguably, static (i.e., time-independent) models are still the mainstay of transport modeling, despite their shortcomings in particular with respect to temporal effects [1]. An important reason is that they are less complex than dynamic (i.e., time-dependent) models, well-studied, and usually have unique solutions under relatively light conditions [16, 10, 14]. Often, static models are motivated with the argument that they are at least able to model a stable long-term situation of their dynamic counterpart. This stable long-term situation –

[1] corresponding author

23rd Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2023).
Editors: Daniele Frigioni and Philine Schiewe; Article No. 11; pp. 11:1–11:14
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

called *steady state* – is in this context defined as a situation which is reached after an initial warm-up phase where congestion builds up on the links of the network and then remains constant (as long as the overall demand remains stable). But does such a steady state exist?

For the dynamic flows over time model (in the single sink-source setting), Cominetti et al. [3] proved the existence of a steady state which is reached in finite time under the necessary condition that the minimal *s*-*t* cut is larger or equal to the constant network inflow rate. Olver et al. [7] generalized this result by proving that even without this condition, a Nash flow over time will reach a final *stable phase* in finite time (whereby in a final stable phase, queue length do not need to be constant but at least only change linearly, forever into the future).

That result, however, assumes that no *spillback* occurs, i.e., traffic jams are never longer than a link. This can be achieved by making the particles infinitely small or by giving links unlimited storage capacity. As the occurrence of a spillback determines whether a local over-saturation of a link affects other (upstream) links – and maybe even blocks intersections or road segments – it is critical for a realistic model to be able to capture spillback effects. Recently, the aforementioned flows over time model has been extended to also capture spillback effects [13]. It is well-known that spillback can have a significant impact on the price of anarchy, i.e., on the efficiency of the equilibrium [5, 11, 15]. Our study complements this related work by identifying another effect that changes when spillback effects are modeled.

Our contribution

We study the long-term behavior of Nash flows over time capturing spillback effects and show that they do not necessarily reach a steady state and not even a stable phase. This is of high relevance as it changes two fundamental properties that have been proven for the long-term behavior of Nash flows over time without spillback: On the one hand, we show that – in contrast to what Cominetti et al. [3] have proven for the case without spillback – there are instances with spillback where Nash flow queue lengths do not become stable, even if the minimal *s*-*t* cut exceeds the constant network inflow rate. Additionally, the same example shows that – in contrast to what Olver et al. [7] have proven for the case without spillback – Nash flows over time with spillback even do not necessarily reach a situation where queues increase linearly forever (i.e., a stable phase).

With that, this study shows that it is essential to include spillback effects into dynamic network models to realistically model effects over time.

Moreover, we show that the Nash flow over time with spillback can consist of an infinite number of phases – even for a single commodity with constant inflow rate. This shows that there are instances where no algorithm exists to calculate Nash flows over time with spillback that runs in polynomial time dependent on the input size, as doubling the value of the considered time horizon doubles the number of phases. In the studied example, the Nash flow phases behave periodically, though.

In general, this study highlights the importance of dynamic transport models: There are effects over time that do not vanish after an initial warm-up phase. Although the importance of dynamic transport models has been apparent for real-world applications with high time dependency, it is still a common motivation for the use of static models that they model the situation beyond the initial warm-up phase, i.e., the steady state. However, we show that such a steady state does not necessarily exist, even for very simple instances.

The given example is not limited to the flows-over-time model. We also study its outcome in the multi-agent transport simulation MATSim [4]. It can be seen that also this simulation does not necessarily lead to a steady state and similar phase oscillations are visible. This shows that the non-existence of a steady state with spillback is not limited to continuous models (as the simulation models discrete time steps and discrete vehicles).

As MATSim is based on a co-evolutionary process which iteratively improves the users' choice, it does not result in exact user equilibria. The fact that the Nash flow phase oscillations can still be observed supports the robustness of this finding. However, the results of the simulation also show that the phase oscillations become rarer the more randomness (or error) is included in the co-evolutionary process. Moreover, phase oscillations even become indistinct and vanish in the long term when we average out the deviations resulting from the specific random seeds in the simulation.

The remainder of this paper is structured as follows. The next section concentrates on the long-term behavior of Nash flows over time with spillback. We describe the model, present the considered example and analyze the oscillating long-term behavior of the resulting phases of the Nash flow over time. In Section 3 we transfer the example to the co-evolutionary transport simulation MATSim. Section 4 draws a conclusion based on the findings.

2 Long-term behavior of dynamic equilibira in fluid queuing networks with spillback

2.1 Flows over time with spillback

We consider a network consisting of a directed graph $G = (V, E)$ with a source node s and and sink node t. Each link $e \in E$ is equipped with free-flow transit time $\tau_e \geq 0$, an in- and outflow capacity rate $\nu_e^+ > 0$ and $\nu_e^- > 0$, and most importantly, a storage capacity $\sigma_e > 0$.

From time 0 on, infinitesimally small flow particles are released with constant network inflow rate of $R > 0$ at the source s. The particles, each seen as a single agents, travel into the network with the goal to reach the destination as fast as possible. After entering a link e, the particles first traverse this link, which takes τ_e seconds. Arriving at the end of the link, they may have to wait in a queue, which forms if the outflow rate of the link exceeds the outflow capacity, or if spillback occurs. These queues always operate at the maximum rate, so either with outflow capacity rate (in the case of no spillback) or with a throttled rate which might happen if a downstream link is full. After the waiting time particles reach the next node and decide which outgoing link they want to take to continue their journey.

Spillback occurs if the total volume of flow that is on a link e from v to w (so either traversing or waiting in the queue) reaches the storage capacity σ_e. In this case, the inflow capacity rate of e is immediately reduced to the outflow capacity rate (or in the case that the outflow capacity is already throttled to this throttled value). This ensures that links never become overfull. As flow cannot wait on nodes, this might lead to a reduction of the outflow capacity rate of upstream links (i.e. incoming links at v). If there is more than one incoming link, the reduction of the capacity rate is done proportionally, i.e., the spillback factor c_v at v is the maximum value of $(0, 1]$ such that – if all outflow capacities of incoming links of v are multiplied by this value – flow conservation at v is possible (and throttled inflow capacity rates are respected). For more details on these flow dynamics refer to [13] and [11].

A *Nash flow over time* (also called *dynamic equilibrium*) in this setting is a feasible flow over time in which each particle travels from s to t on a shortest path. They form a Nash equilibrium in the following sense: Each agent considers the chosen paths of all other agents. This determines the given flow over time f. (Note that a single infinitesimally small agent does not have any impact on the flow over time so it does not matter if the agent considers the flow over time with or without themself.) We identify agents by the time they enter the network, so let $\theta \geq 0$ be the agent that starts their journey at time θ. With the given flow over time it is possible to determine the travel time of θ for any s-t path in the network. We call f a Nash flow over time if every agent travels along a path that has the shortest travel time over all possible path choices. For a mathematically precise definition, see [13] and [11].

It is far from trivial to see if these flows over time exist at all. Fortunately, it has been proven by Sering and Vargas Koch [13] that Nash flows over time always exist in the spillback setting (under some natural condition on the network). Thereby, the path choice of an agent θ does only depend on the path choices of all agents entering the network before time θ. That is, we have a network-wide FIFO (first in first out) principle: Agents cannot be overtaken by following agents and, therefore, are not impacted by them by any means. Even better, intervals of agents always choose the same path (more precisely the same convex combination of paths). This means that a Nash flow over time can be decomposed into phases corresponding to the choices of an interval of agents. For example in Figure 2, all agents entering the network within $[0, 14)$ choose the middle path; agents entering within $[21, 50)$ split up between the middle and the lower path, more precisely a rate of two takes the middle path and a rate of four takes the lower path.

These phases are called *thin flow phases* and the flow split is given by very specific static flows called *spillback thin flows*; see [13]. These flows can be computed with the help of a mixed integer program, which leads to a constructive algorithm for a Nash flow over time: Start with the empty network (1), compute a spillback thin flow for the current configuration (2), determine for how long this phase will be valid (3), and compute the Nash flow over time for this phase by simulating it with the flow split given by the spillback thin flow. Step (2) to (4) can be repeated to extend the Nash flow over time until a final phase is found (i.e., a phase that is valid forever). For Nash flows over time in the model without spillback this final phase is always reached in finite time [7] but for the spillback model we show that no such final phase exists and, therefore, such an algorithm may not terminate.

2.2 Periodic long-term behavior of Nash flows over time with spillback

The oscillating long-term behavior of Nash flows over time can be observed in the example given in Figure 1. All demand (six flow units per second; in the following denoted as $^{vol}/_{s}$) travels from s to t through the network. There are three possible paths: The upper path, the middle path and the lower path. Free-flow travel times (in s) per link are given in black, outflow capacities (in $^{vol}/_{s}$) in red ovals, and storage capacities (in vol) in green boxes.

Because of the specific choice of the travel times, outflow-, and storage capacities, an interesting Nash flow pattern arises: There is no point in time when a steady state is reached. Path inflow rates change periodically over time and no stable queue lengths are reached (neither constant nor linearly increasing). This special pattern will be described in the remainder of this section.

Note that the network resembles Braess' network [2] with adapted travel times and capacities. This network is well-known to be hard for selfish routing network flows. Similar to the original Braess' network, the present example contains a Braess' paradox: If one removes the middle link v_3v_4, the equilibrium travel time is much lower (and queue length are constant). Accordingly, one can show that the price of anarchy in the present example is unbounded, which is in line with previous research on the impact of spillback on the price of anarchy [5, 11, 15].

Phase description of the Nash flow over time

Figure 2 shows the resulting phases of the Nash flow over time. A phase consists of agents θ departing in a specific time interval and experiencing similar network conditions. The phase illustrations are, therefore, given from the perspective of the agents.

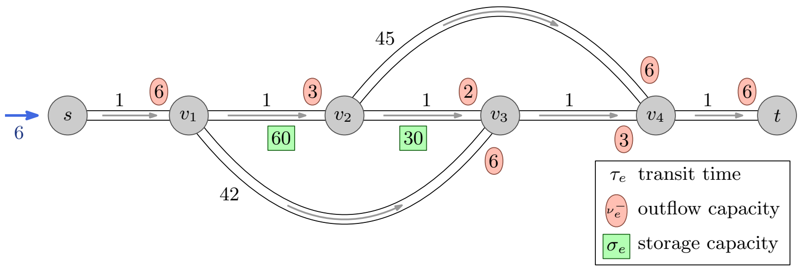

Figure 1 The network used in this study. Free-flow travel times [s] per link are stated as black numbers, outflow capacities [vol/s] in red ovals, and storage capacities [vol] in green boxes. Links without a label for storage capacities have unlimited storage. The network inflow rate constitutes six flow units per second traveling from s to t through the network.

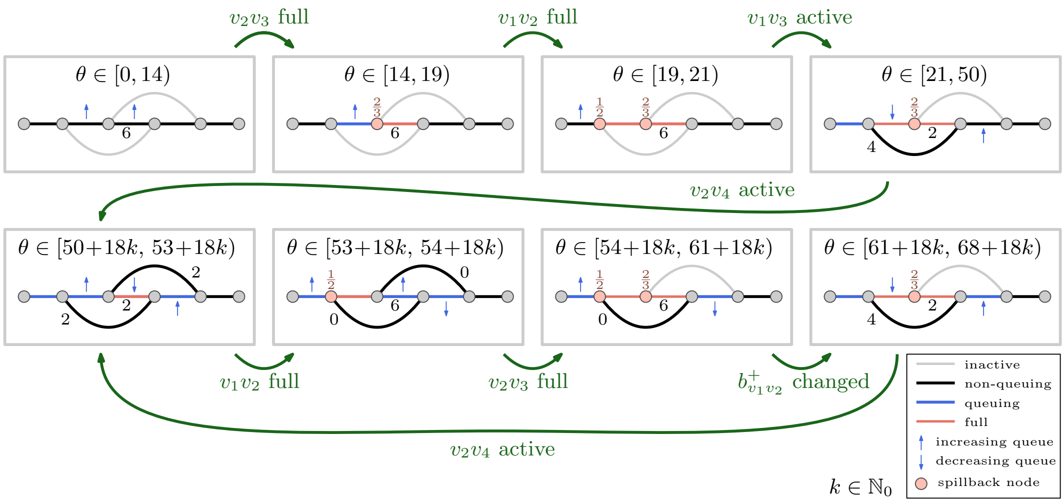

Figure 2 Consecutive thin flow phases of the Nash flow over time in the present example. There are infinitely many of them, as phase five to eight repeat periodically every 18 seconds. θ denotes the network entrance time of the agents forming the phases. The small numbers close to the links (in black) denote the path inflow rates during the phase. The small numbers on top of the spillback nodes (in red) denote the spillback factor – the outflow capacity rates of all incoming links are reduced by this factor. Between the phases the event that triggered the previous phase to end is indicated. Note that the events between the phases are not in chronological order, but from the perspective of the agents; e.g. all agents of the second phase (i.e., all agents entering the network within $[14, 19)$) experience a non-full link v_1v_2 but a full link v_2v_3 even though time-wise v_1v_2 becomes full before v_2v_3. The links are colored by their category: *Inactive* links do not belong to any fastest s-t path in the corresponding phase; on *non-queuing* links, the first agent of the interval is not delayed; on *queuing* links (also called *resetting* links in the literature), all agents of the interval have to wait in a queue; and *full* links are full when the first agent of the interval reaches its tail.

It is important to understand that this *agent view* is different from a *snapshot view*. The latter shows the dynamics of the network over time as one is used to from visualizations of simulations. The former shows the view of the decision-making agent. For this, the state of the links as they *will be* when the agent *will be there* is important. Note that with the agent view events between the phases do not appear in chronological order, but in the order the agents are impacted by them.

To understand this perspective, let us first consider the network filling up (corresponding to the phases in the first line of Figure 2). In the first three phases (consisting of all agents entering the network before time 21), all flow takes the shortest middle path.

For link v_1v_2 this means that from time 1 onward (i.e., after the first agent has passed link sv_1), a flow rate of 6 $^{vol}/_{s}$ enters the link. Due to the bottleneck given by the outflow capacity, only a rate of 3 $^{vol}/_{s}$ leaves the link after the free-flow travel time, i.e., from time 2 onward. Hence, at time 20, there is a flow volume of $19 \cdot 6 - 18 \cdot 3 = 60$ on this link, which means that it becomes full at this moment. This is the time when agent $\theta = 19$ arrives at v_1.

Let us consider link v_2v_3 now: From time 2 on, a flow rate of 3 $^{vol}/_{s}$ enters this link, but due to the even smaller bottleneck, only a rate of 2 $^{vol}/_{s}$ leaves it, starting at time 3. Hence, at time 30 a total flow volume of $28 \cdot 3 = 84$ has entered the link and a flow volume of $27 \cdot 2 = 54$ has left, i.e., at that time the flow volume reaches the storage capacity of 30 and the link becomes full. The agent departing at $\theta = 14$ reaches this link at time 30, thus, from the perspective of departing agents, v_2v_3 is full from $\theta = 14$ on, thus, earlier than v_1v_2 (see the phase illustration in Figure 2).

For brevity, we are not going through all details of the Nash flow phases, here. Exemplarily, let us examine the phase shift between phase three and four in more detail in the following: Consider an (infinitesimally small) agent departing at time 20. From her perspective, both links v_1v_2 and v_2v_3 are full. Due to spillback across node v_1 from snapshot time 20 on, the outflow capacity of link sv_1 reduces to 3 $^{vol}/_{s}$. Note that it is not 2 $^{vol}/_{s}$, because at this snapshot time, the downstream link v_2v_3 is not full yet. So, agent $\theta = 20$ has 3 flow volumes in the queue on link sv_1 ahead of her, meaning that she needs one second for traversing the link plus one second for waiting and reaches node v_1 at time 22. Consider the agents ahead of her: The first one needs 3 seconds to reach v_3, and from then on v_3 discharges agents with a rate of 2 $^{vol}/_{s}$. As agent $\theta = 20$ has $20 \cdot 6 = 120$ agents ahead of her, this needs another 60 seconds, i.e., she can finally leave v_2v_3 at time 63, and, with that, 41 seconds after arriving at node v_1. Thus, the bypass with 42 seconds travel time is still slightly longer.

This, however, changes for agent $\theta = 21$. He reaches v_1 at time 24 (transit time of 1 plus a waiting time of 2). $21 \cdot 6 = 126$ agents are ahead of him, i.e., he can leave v_2v_3 at time $3 + 63 = 66$, i.e., 42 seconds after arriving at node v_1. If he takes the lower path instead, he would experience the same travel time. For that reason, the lower path or, in particular, v_1v_3 becomes active for agent $\theta = 21$.

The following agents split $2 : 4$ between the middle and the lower path such that travel times on both paths stay balanced during the whole phase (for all agents $\theta \in [21, 50)$; see Figure 2). At snapshot time 24, when agent $\theta = 21$ enters link v_1v_2, agents still leave the link with the full outflow capacity of 3 $^{vol}/_{s}$ as the downstream link v_2v_3 only becomes full at snapshot time 30. With that, the queue on v_1v_2 decreases again (by 1 $^{vol}/_{s}$ for 6 time steps, and then stays constant) such that spillback dissipates from node v_1. Nevertheless, the queue on sv_1 does not decrease but stays constant. On the other hand, link v_3v_4 starts growing a queue (from the perspective of the agents).

The Nash flow phases continue to process as depicted in Figure 2. Let us concentrate on the main aspects in the following. For more details and examples on how Nash flows over time with spillback evolve in general, the reader is referred to previous studies [11, 13].

The fourth phase ends with the moment when travel times on all three paths become equal, and, with that, the upper path becomes active and used. This happens at departure time $\theta = 50$. With phase five, the warm-up phase has ended and from now on phase five to eight repeat cyclically every 18 seconds.

The increased inflow to link v_1v_2 causes the queue on that link to grow again. Simultaneously (so for the same agents) only $1/3$ of the flow uses the link v_2v_3 (which was full at the beginning) causing it to become non-full. For agents entering only 3 seconds later, v_1v_2 gets full again, which means that the queue on this link can no longer grow. As a consequence, all flow travels through the middle path, which leads to spillback across v_1. For agents entering only one second later, v_2v_3 becomes full yet again, but this does not change the flow behaviour. For agents entering seven seconds later, the outflow rate of link v_1v_2, which was 3 vol/s before, is reduced to 2 vol/s due to spillback arriving at v_2 (from the perspective of the particle). For that reason, the lower path (which was active all the time but not used by any flow) is used again causing link v_1v_2 to become non-full. Finally, for agents entering seven seconds later, the upper path becomes active, which results in the same situation as at the beginning of the loop (phase five) with the only difference that the queue on the first link sv_1 has increased. Since the storage capacity on this link is unbounded and all the flow has to traverse this link in any case, the loop repeats indefinitely.

Flow values, travel times, and queue length of the Nash flow over time

Figure 3 shows the cumulative flow values of flow particles in the Nash flow over time for the three different routes in the present example for the first 200 seconds. When the distribution of flow particles on the three routes changes between the phases of the Nash flow over time, this can be observed in the plot by changes in line slopes.

Figure 5 and Figure 6 illustrate the link travel time and volume of queuing flow particles in the Nash flow over time for the two most interesting links sv_1 and v_3v_4. The plots verify that the queue (and, with that, the travel time) on the first link keeps growing (in phases $6 + 4k$ and $7 + 4k$, $k \in \mathbb{N}_0$, respectively; see Figure 2) and never decreases. Hence, also overall travel time in the network (which is depicted in Figure 4) increases over time. This shows that the network throughput is strictly smaller than the inflow rate, although the minimal s-t cut is larger, and, thus, it would be possible to send all flow through the network without delay. Such an inefficient Nash flow is typical for a Braess-like instance, though. Note that Nash flow travel times of all flow particles with the same departure time are equal, independent of their route, as all flow particles share the same origin and destination.

The steps in the link travel time plot for the first link (see Figure 5) exactly correspond to the time intervals of the phases from Figure 2: Link travel time increases when spillback occurs, i.e., when the next link is full. Interestingly, the corresponding steps on the queue volume plot in Figure 6 happen more seldom. This is because queue volumes increase on link sv_1 as long as flow particles experience a situation with larger in- than outflow (in this case due to spillback from v_1v_2). Note that this might be longer than the departure time interval of the agents corresponding to that phase. Consider for example the spillback phases six and seven. The first agent of that phases, $\theta = 53$, arrives at node v_1 at time $53 + 1 + 2 = 56$. The last agent $\theta = 61$ arrives 24 seconds later at v_1, i.e., the queue volume increase holds on for these 24 seconds, although the two phases together only have a length of 8 seconds (in terms of departure time at s). Afterwards, the queue volume stays constant for 10 seconds, which corresponds to the length of the departure time interval of phases eight and nine, because travel time on sv_1 stays constant in that time period. Together, one Nash flow phase cycle of 18 seconds corresponds to a queue volume period of 34 seconds, and the queue volume periods repeat periodically, similar to the Nash flow phases.

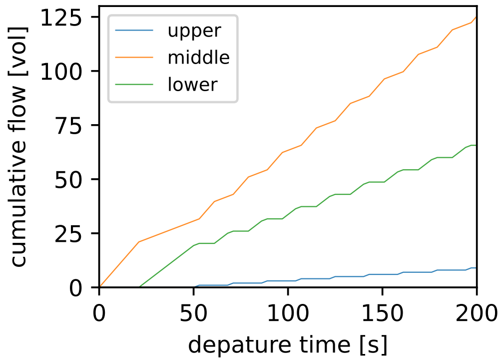

Figure 3 Cumulative flow values in the Nash flow over time for the three different routes.

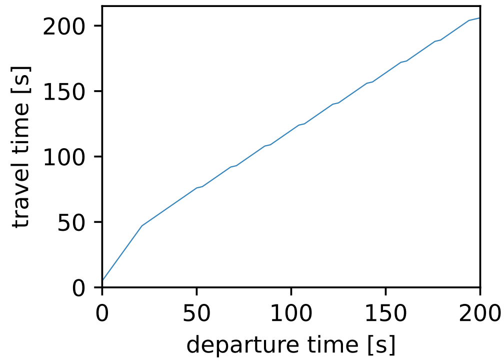

Figure 4 Travel time per departure time in the Nash flow over time.

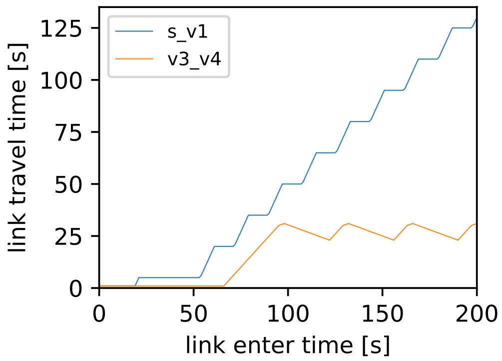

Figure 5 Link travel time of links sv_1 and v_3v_4 in the Nash flow over time dependent on the link enter time.

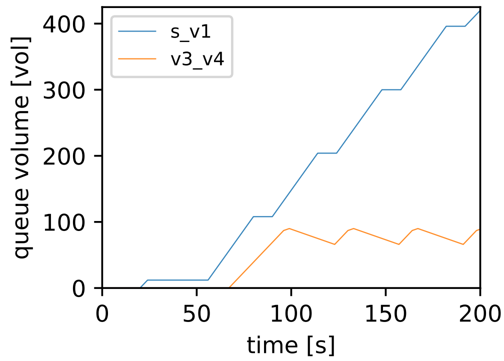

Figure 6 Volume of queuing flow particles on the links sv_1 and v_3v_4 in the Nash flow over time.

For link v_3v_4, the periods for the travel time and the queue volume in Figure 5 and Figure 6 have the same length, because the travel time does not increase every period but oscillates around a stable value dependent on the Nash flow phase oscillation. Both measures have a period length of 34 seconds.

In sum, the analysis of phases and queue length of the Nash flow over time in this instance has shown that there is no point in time when a steady or stable state is reached, and there exist infinitely many oscillating phases in the Nash flow over time. To be precise, the oscillating pattern of the lines in Figures 3–6 repeats indefinitely.

3 Long-term behavior of equilibria in a discrete, co-evolutionary transport simulation

This section shows that it is not only a theoretical finding that the phases of a dynamic equilibrium oscillate infinitely and no steady state is reached when spillback effects are modeled. When we apply the co-evolutionary transport simulation MATSim to the same instance as in Section 2, similar phase oscillations can be observed. This is interesting as the simulation does (in contrast to Nash flows over time) not lead to an exact user equilibrium and, moreover, it models discrete time steps and vehicles (whereas Nash flows over time are continuous). For a detailed model comparison the reader is referred to our previous study

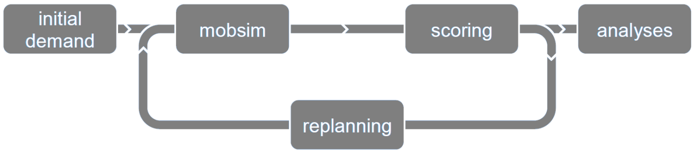

Figure 7 Iterative, co-evolutionary cycle of MATSim [4].

[17]. Despite the different perspectives, both models behave very similar. Our previous experiments indicate that Nash flows over time are the limit of the convergence processes when decreasing the vehicle size and time step length in the simulation coherently [17]. Accordingly, we were able to mathematically prove the convergence of the flow models [12], and, even further, that Nash flows over time converge to competitive packet routing games (similar to MATSim) with decreasing refinement level [8].

3.1 The multi-agent transport simulation MATSim

In MATSim, the road network is represented by a directed graph. Each link is defined by a free-flow travel time, a flow capacity and a storage capacity. The storage capacity determines the number of vehicles which fit on a link spatially. An exceeded storage capacity effects that vehicles have to remain on the upstream link and, as such, the model allows to model spillback effects. MATSim's traffic simulation handles each link as a first-in-first-out (FIFO) queue. A vehicle that enters a link is immediately put into the FIFO queue and a so-called earliest exit time is set as the entrance time plus the link's free-flow travel time. In each time step, MATSim's traffic simulation checks the following conditions to determine whether a vehicle can leave the queue of a given link: (1) The vehicle is at the head of the queue, (2) the link's earliest exit time has passed since the vehicle entered the link, (3) the flow capacity of the link is sufficient, and (4) the next link has sufficient space left, i.e., its storage capacity is not exceeded. When a vehicle leaves a link its flow volume is subtracted from the remaining flow capacity for this time step. If a sufficient flow capacity for the flow volume of the next vehicle remains, this other vehicle is allowed to leave the link. Otherwise, a next vehicle can only leave once sufficient flow capacity has accumulated over the following time step(s). When no vehicle wants to leave the link for some time, the flow capacity does not accumulate more than its value per time step, i.e., flow capacity cannot be saved for the future.

MATSim is based on a co-evolutionary algorithm, i.e., an iterative process where in each iteration a fraction of agents is allowed to change their plans by choosing from a set of good responses with the goal to improve their (individual) score. This procedure leads to a state where most of the agents do not have any incentive to deviate, but this does not necessarily correspond to an exact user equilibrium. The co-evolutionary algorithm consists of the three steps *mobsim*, *scoring* and *replanning* and is illustrated in Figure 7. The flow model described above corresponds to the *mobsim* module, where plans of agents are executed on the network. Next, all executed plans are evaluated by the *scoring* module (in this study, scores are only based on the experienced travel times). Based on these scores, agents either change their plans within their current plan choice set or generate completely new plans during *replanning*. In this study, agents are only allowed to change their routes, whereas in general changes along other choice dimensions (e.g., departure time or mode choice) can be represented in MATSim [4]. During re-routing, agents use the knowledge of all travel times in the network of the last iteration and, based on those, choose the shortest possible route based on the last iteration.

Simulation setup for the present study

For the present study, we use a simulation time step size of $1/16$ seconds, and, in line with our previous study [17], the square of this as the vehicle size. MATSim's co-evolutionary algorithm is run for 1000 iterations. At the beginning, all agents of the simulation are equipped with the three possible routes and aim to find their best option within that plan choice set over the iterations. In the first 800 iterations, $1/3$ of the agents choose the plan with the best score (i.e., lowest travel time in this case), $1/3$ stay with their last choice, and the other $1/3$ of the agents apply a logit model to choose a plan from their choice set, i.e., a plan is chosen with a certain probability based on the score. From iteration 800 on, the logit-model-based strategy is switched off and the probability of the other two strategies becomes $50:50$. Additionally, a method of successive averaging is applied on the score of the plans from iteration 800 onwards. With that, the plan scores become more stable between iterations which supports convergence towards a stable choice.

Some parts of MATSim depend on random values. In this setup this mainly applies to the logit model used to choose plans from the choice set of the agents. One can influence the randomness by choosing the initial random seed of the simulation. Based on this initial value, the simulation will then set the random seed to a different value in each iteration. With the same initial seed, two different simulation runs will underlie the same random values, though. To be able to analyze the deviations depending on the specific random values, we, therefore, repeat the simulation with 20 different initial random seeds.

We analyze the present example for an inflow time interval of $[0, 500]$, i.e. in each simulation run agents depart within the first 500 seconds of simulation time (with the aforementioned, constant inflow rate). This keeps the run time within reasonable limits and still shows the relevant pattern of phase oscillations.

3.2 Long-term behavior of equilibria in MATSim

The oscillating phases of the Nash flow over time (see Section 2.2) can also be observed in MATSim: Figure 8 illustrates the queue volumes over time in the simulation for the two links sv_1 and v_3v_4 with the most interesting behavior. The plots show the full period of queue increase and decrease for the simulation time of 500 seconds. (All following figures are zoomed into the first 300 seconds of the simulation to better see the phase structures.) For all following figures, the right plot shows the values for a specific random seed run; the left plots show the average value over all random seed runs.

First of all, we can see a lot of phase switches where path inflow rates change (identified by changes in line slopes). They are particularly distinct in the plots on the right that show a selected random seed run. Interestingly, the phases in MATSim become the longer the more time has passed. Probably, this is because the simulation does never result in exact best solutions, but includes some randomness in agents' route choice. This causes small errors that accumulate over time and, again, increase the inaccuracy of the route choice of the following agents. In the present example, this leads to slightly fewer agents using the current best path in the simulation than in the Nash flow over time. Accordingly, queues on the best path in the simulation built up a bit slower. Therefore, the balancing of route travel times, which is necessary for the phase shifts, happens later than in the Nash flow. Hence, phases expand.

The expansion of phases can also be seen in the right part of Figure 9, which depicts link travel times on the two links sv_1 and v_3v_4: In particular, the link travel times on the first link show the aforementioned delayed increase for later phases (by a decreasing slope).

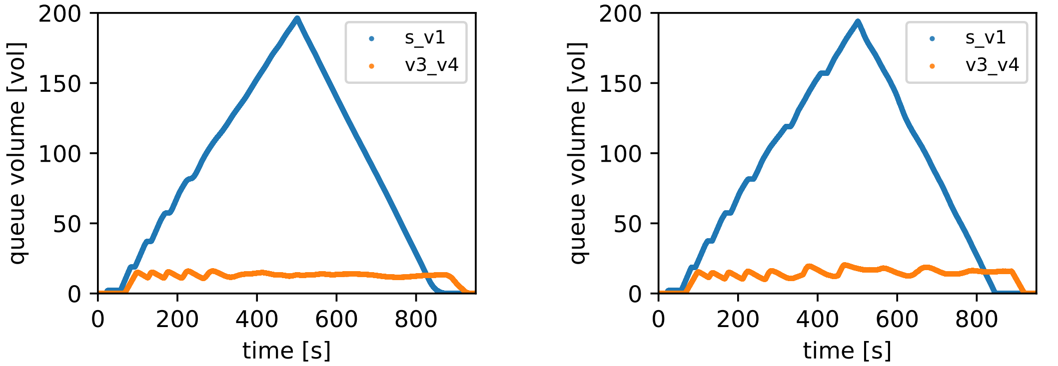

Figure 8 Flow volume of delayed vehicles in MATSim for the two links sv_1 and v_3v_4 – on the left, averaged over all random seed runs; on the right, corresponding to a selected random seed run.

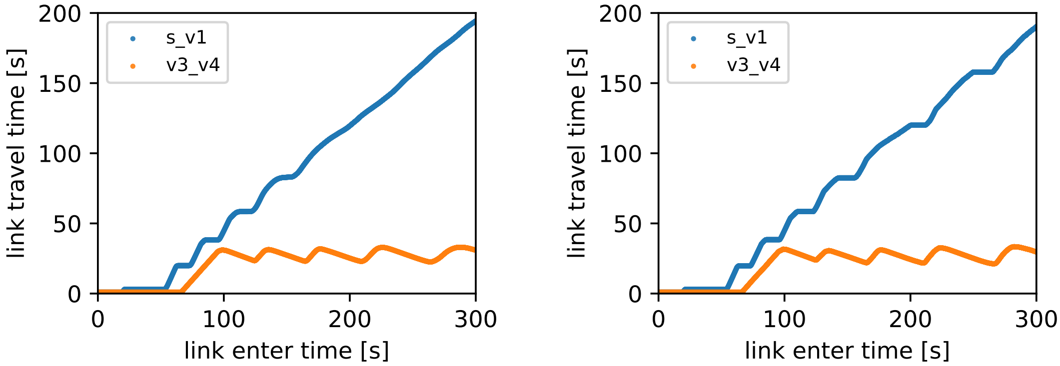

Figure 9 Link travel time on sv_1 and v_3v_4 in MATSim dependent on the link enter time – on the left, averaged over all random seed runs; on the right, corresponding to a selected random seed run.

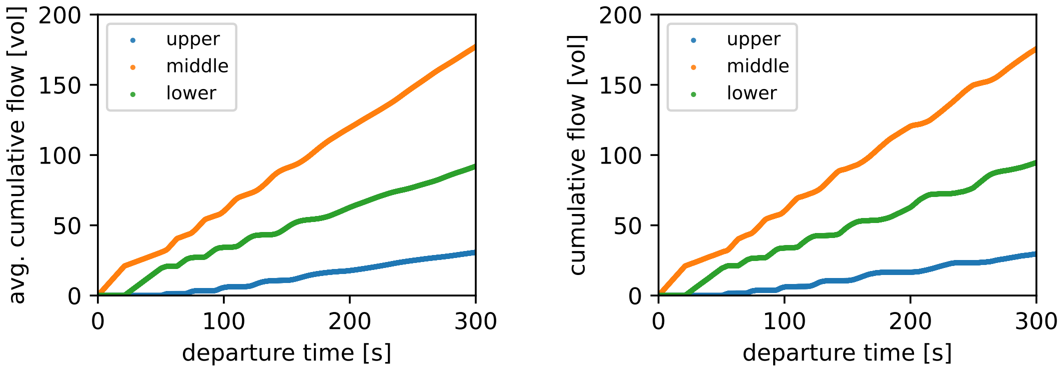

Figure 10 Cumulative flow values of vehicles in MATSim for the three different routes – on the left, averaged over all random seed runs; on the right, corresponding to a selected random seed run.

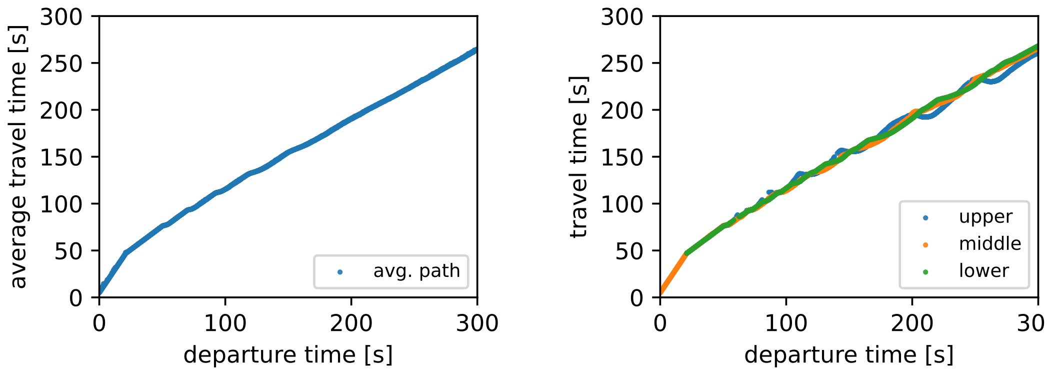

Figure 11 Travel time of vehicles in MATSim dependent on their departure time. The left plot shows the average travel time over all routes and random seed runs, the right plot the average travel time for one specific random seed run and each route separately.

Figure 10 shows the cumulative flow values on the three different routes in the simulation. Again, the oscillating route distribution depending on the phases and their expansion over time can be seen, especially for the selected random seed run on the right.

The effect that most of the phase oscillation vanishes in the long run, when the results are averaged over multiple random seed runs (see all left plots in Figures 8–10), is an interesting side finding of this study. The reason for this effect is a natural consequence of the aforementioned error in the simulation that accumulates over time. As this error is heavily dependent on the random values that are used in the simulation (i.e., the initial random seed), the deviations from the exact user equilibrium are dependent on the random values as well. With that, the results of the different random seed runs diverge more and more, the more time has passed, i.e., the more error has accumulated. Clearly, averaging over multiple random seed runs, therefore, averages out the long-term oscillations and results in stable, average line slopes. This is important as it might result in significantly different results. While modeling transport realistically, one usually is not interested in average network travel times or queue length, but wants to know where and when congestion occurs and which effects exist over time.

Because fewer agents use the middle route in the simulation than in the perfect Nash flow due to the aforementioned deviations over time, another interesting side effect occurs: The overall travel time (see Figure 11) is slightly lower compared to the Nash flow over time (the higher the departure time, the lower the slope of the plot). This means that the simulation leads to a slightly better overall situation – not because of intelligent measures, but because of randomness and inaccuracy.

However, the right plot of Figure 11 also shows that the travel time in the simulation is not fully converged: Some agents travel longer than other agents with the same (or later) departure time and can, therefore, improve by unilaterally changing their route. In consequence, we assume that route travel times, and, with that, cumulative flow values would approximate further to the Nash flow over time values if the co-evolutionary learning approach of the simulation was run for even more iterations and, thus, would result in a situation that is closer to the exact user equilibrium. Alternatively, one could force the agents in the simulation to choose their routes sequentially, one after the other, depending on their departure time. Because our example constitutes a single-commodity instance, this would result in the perfect user equilibrium [6]. However, real-world traffic will never be so perfectly distributed. Instead, the observed deviations that the co-evolutionary approach results in, might even rather align with how real-world travelers make their decisions and are, therefore, also relevant to study on its one.

4 Conclusion and outlook

This study has shown that dynamic Nash flows capturing spillback effects do not necessarily reach a steady state, i.e., a situation with constant queue lengths. Moreover, there are (indeed very simple) instances where the phases of the Nash flow over time oscillate infinitely. As a consequence, the long-term behavior of dynamic equilibria heavily depends on the fact whether spillback is captured in the model or not. These findings also highlight the importance of dynamic transport models in general. Additionally, we have shown that similar phase oscillations as in the Nash flows over time model can be observed in the multi-agent transport simulation MATSim. This supports robustness of the findings as the simulation does (in contrast to Nash flows over time) not lead to exact user equilibria and, moreover, uses discrete time steps and vehicles.

However, we also observed a significant deviation in the results when more randomness is added to the co-evolutionary process of finding a stable state in the simulation. This is because there is a strong dependence between following vehicles in the example considered here. A route change of a preceding vehicle influences the travel time of all following vehicles. Thus, deviations from the user equilibrium accumulate and persist over time and cause even further deviations/errors. Still, the simulation outcome always had a structure similar to the Nash flow over time. It would be an interesting follow-up question whether there exists an example where these vehicle dependencies and accumulated errors result in a structurally different solution, or, whether the co-evolutionary algorithm of the simulation might even result in a chaotic solution, e.g. a grid lock, whereas the Nash flow over time does not.

Related to this is the question regarding continuity of dynamic equilibria, i.e., whether small perturbations to the instance can lead to structurally different equilibrium solutions. Although continuity of Nash flows over time has been proven recently for the case without spillback [7], simulation studies have shown, that the co-evolutionary approach of MATSim can lead to situations where a small change in the agent behavior can lead to huge changes in the congestion pattern [9]. We assume, this discrepancy also stems from the presence/absence of spillback effects. It would be interesting to investigate this further.

In the present example, the Nash flow over time consists of an infinite number of phases, but the same three phases repeat periodically. A naturally next question is whether there exists an instance with finite input and pure infinite output. This would finally rule out the efficient calculation of Nash flows over time with spillback.

References

1. M. Balmer, N. Cetin, K. Nagel, and B. Raney. Towards truly agent-based traffic and mobility simulations. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 60–67, 2004.
2. D. Braess. Über ein Paradoxon aus der Verkehrsplanung. *Unternehmensforschung*, 12:258–268, 1968.
3. R. Cominetti, J. Correa, and N. Olver. Long-term behavior of dynamic equilibria in fluid queuing networks. *Operations Research*, published online, 2021.
4. A. Horni, K. Nagel, and K. Axhausen, editors. *Multi-agent transport simulation MATSim*. Ubiquity Press, London, UK, 2016.
5. J. Israel and L. Sering. The impact of spillback on the price of anarchy for flows over time. In *International Symposium on Algorithmic Game Theory (SAGT)*, pages 114–129, Augsburg, Germany, 2020. Springer, Springer.
6. R. Koch and M. Skutella. Nash equilibria and the price of anarchy for flows over time. *Theory of Computing Systems*, 49(1):71–97, 2011.

7 N. Olver, L. Sering, and L. Vargas Koch. Continuity, uniqueness and long-term behavior of Nash flows over time. *IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 851–860, 2021.

8 N. Olver, L. Sering, and L. Vargas Koch. Convergence of approximate and packet routing equilibria to Nash flows over time. *IEEE 64nd Annual Symposium on Foundations of Computer Science (FOCS)*, 2023.

9 M. Rieser and K. Nagel. Network breakdown "at the edge of chaos" in multi-agent traffic simulations. *European Journal of Physics*, 63(3):321–327, 2008.

10 R.W. Rosenthal. The network equilibrium problem in integers. *Networks*, 3(1):53–59, 1973.

11 L. Sering. *Nash flows over time.* PhD thesis, Technische Universität Berlin (Germany), 2020.

12 L. Sering, L. Vargas Koch, and T. Ziemke. Convergence of a Packet Routing Model to Flows Over Time. *Mathematics of Operations Research*, 2022.

13 L. Sering and L. Vargas Koch. Nash flows over time with spillback. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 935–945, San Diego, California, USA, 2019. SIAM.

14 Y. Sheffi. *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods.* Prentice-Hall, Englewood Cliffs, NJ, USA, 1985.

15 T. Thunig and K. Nagel. Braess's paradox in an agent-based transport model. *Procedia Computer Science*, 83:946–951, 2016.

16 J.G. Wardrop. Some theoretical aspects of road traffic research. *Proceedings of the Institute of Civil Engineers*, 1(3):325–378, 1952.

17 T. Ziemke, L. Sering, L. Vargas Koch, M. Zimmer, K. Nagel, and M. Skutella. Flows over time as continuous limit of packet-based network simulations. *Transportation Research Procedia*, 52:123–130, 2021.