Competitive Routing over Time *

Martin Hoefer † Vahab S. Mirrokni ‡ Heiko Röglin § Shang-Hua Teng ¶

June 7, 2011

Abstract

Congestion games are a fundamental and widely studied model for selfish allocation problems like routing and load balancing. An intrinsic property of these games is that players allocate resources simultaneously and instantly. This is particularly unrealistic for many network routing scenarios, which are one of the prominent application scenarios of congestion games. In many networks, load travels along routes over time and allocation of edges happens sequentially. In this paper we consider two frameworks that enhance network congestion games with a notion of time. We introduce *temporal network congestion games* that are based on coordination mechanisms — local policies that allow to sequentialize traffic on the edges. In addition, we consider *congestion games with time-dependent costs*, in which travel times are fixed but quality of service of transmission varies with load over time. We study existence and complexity properties of pure Nash equilibria and best-response strategies in both frameworks for the special case of linear latency functions. In some cases our results can be used to characterize convergence properties of various improvement dynamics, by which the population of players can reach equilibrium in a distributed fashion.

*An extended abstract of this paper has appeared in the proceedings of the 5th Workshop on Internet and Network Economics (WINE 2009) [33].

†Department of Computer Science, RWTH Aachen University, Germany, mhoefer@cs.rwth-aachen.de. Supported by DFG through UMIC Research Centre at RWTH Aachen University and grant Ho 3831/3-1.

‡Google Research New York, USA, mirrokni@google.com

§Department of Computer Science, University of Bonn, Germany, heiko@roeglin.org. Supported by a fellowship within the Postdoc-Program of the German Academic Exchange Service (DAAD).

¶Computer Science Department, University of Southern California, USA, shanghua.teng@gmail.com.

1 Introduction

As an intuitive game-theoretic model for competitive resource usage, *network congestion games* have recently attracted a great deal of attention [2, 24, 48]. These games are central in modeling routing and scheduling tasks with distributed control [49]. Such games can be described by a routing network and a set of players who each have a source and a target node in the network and choose a path connecting these two nodes. The quality of a player's choice is evaluated in terms of the total delay or latency of the chosen path. For this, every edge e has a latency function that increases with the number of players whose paths include edge e. Ignoring the *inherent delay* in transmitting packets in networks or routing cars in road networks, this model implicitly assumes that players use all edges on their paths instantaneously and simultaneously.

Depending on the application, it might not be reasonable to assume that a player instantaneously allocates all edges on his chosen path. Consider for instance a road traffic network, in which players route cars to their destinations. Clearly, a traffic jam that delays people at rush hour might be harmless to a long distance traveler who reaches the same road segment hours later. In this case, it is more natural to assume that edges are allocated consecutively, and players take some time to pass an edge before they reach the next edge on their path. In addition, for connections in computer networks the system may use a *local queuing policy* to schedule the players traversing this edge.

In this paper, we study two different models that extend the standard model of network congestion games by a temporal component. In our first model, we incorporate the assumption that on each edge, the traffic over the edge must be sequentialized which in turn results in a *local scheduling problem with release times* on each edge, and requires a formal description of the local scheduling or queuing policy on each edge. To model these local scheduling policies, we use the idea of *coordination mechanisms* [7, 14, 20, 36] that have been introduced and studied mainly in the context of machine scheduling and selfish load balancing [55]. In selfish load balancing, each player has a task and has to assign it to one of several machines in order to minimize his completion time. A coordination mechanism is a set of local scheduling policies that run locally on machines. Given an assignment of tasks to machines, the coordination mechanism run on a machine e gets as input the set of tasks assigned to e and their processing times on e. Based on this information, it decides on a preemptive or non-preemptive schedule of the tasks on e. The local scheduling policies of the coordination mechanism do not have access to any global information, like, e.g., the set of all tasks and their current allocation.

Applying the idea of coordination mechanisms to network congestion games results in the definition of *temporal congestion games*, which are studied in Section 3. We assume that each edge in a network congestion game is a machine equipped with a local scheduling policy, and each player has a task and chooses a path. Starting from their source, tasks travel along their path from one edge to another until they reach the target. They become available on the next edge of their path only after they have been processed completely on the previous edges. The player incurs as latency the total travel time that his task needs to reach the target. Each player then strives to pick a path that minimizes his travel time.

In our second model, which we term *congestion games with time-dependent costs* and study in Section 4, we assume that the travel time along each edge is a constant independent of the number of players using that edge. This model captures the property that increased traffic yields decreased quality of service for transmitting packets. For instance, in wireless networks increased congestion on a link can increase the failure probability of transmissions and packets get lost. Similarly, in road networks increased traffic can increase the probability that a car is involved in a serious accident. One could try to incorporate this aspect via an adjusted travel time. However, the travel time of a lost packet or a car involved in a serious accident is usually extremely large compared to normal travel times. In addition, there are other obvious disadvantages in having an accident than just increased travel time. Thus, combining these fundamentally different aspects into one function is an unsuitable approach. Instead, we here use a separate time-dependent cost function to capture such risks. We assume time is discretized into units (e.g., seconds), and the cost of an edge during a second depends on the number of players currently traveling on the edge. Each player now strives to pick a path that minimizes the total time-dependent costs during the travel time along the edges.

Our games extend the model of atomic congestion games, which were initially considered by Rosenthal [48]. They are a vivid research area in (algorithmic) game theory and have attracted much research interest, especially over the last decade. A variety of issues have been addressed, most prominently existence and computational complexity of equilibrium concepts such as pure Nash equilibria [2, 24, 48], approximate equilibria [1, 8, 15, 18, 26, 51], strong equilibria [34, 35], or states with no-regret property [11, 37]. Another important direction is bounding the inefficiency of equilibrium states, which has been done extensively, e.g., for pure Nash equilibria [6, 19, 50], approximate Nash equilibria [20], Pareto-optimal Nash and strong equilibria [3, 17], or states with no-regret property [12]. For an overview and introduction to the topic we refer to the recent expositions by Roughgarden [49] and Vöcking [55].

Addressing algorithmic aspects of congestion games with different notions of time has only been started very recently in a number of papers [4, 25, 39]. Koch and Skutella [39] consider a general model for flows over time known in the traffic literature as deterministic queuing model. In their model traveling times are constant but the time spent in FIFO queues at the nodes may vary depending on traffic over time. They prove existence of equilibria, provide a structural characterization and efficient algorithms for computation as well as bounds on the inefficiency of equilibria. Recently, Macko et al. [40] provide further insights to characterize Braess paradox in flows over time and stronger lower bounds on the inefficiency of equilibria. Bhaskar et al. [9] further bound the price of anarchy for different social cost functions and show how to successfully apply Stackelberg strategies in this scenario. For a similar model of flows over time, Anshelevich and Ukkusuri [4] derive a number of related results. These papers relate to the classic Wardrop model of static selfish flows [56]. More generally, they relate to a significant amount of work in the literature on flows over time. While most related work addresses flows over time with respect to global optimization [16, 27, 30, 47, 52], there are also a variety of papers that address competitive situations and equilibria [42, 46, 53, 54, 57]. However, due to the complex dependencies in these models and their analysis, there are many open problems with respect to characterization and computation of equilibria. For a deeper discussion of related work in this area, see, e.g., [39].

Let us point out that there are several differences between our model and flows over time. First, in contrast to our work, all the above surveyed literature addresses non-atomic congestion games, in which players are infinitesimally small flow particles and thus do not have different weights or induce different transmission delays. In fact, in many cases it is assumed that transmission time on an edge is constant. Strategic issues arise only from different waiting times to enter the next edge, which depend on the queued amount of earlier flow. Second, in flows over time as studied in [9, 39], we have a common source that emits a rate of flow, that is, players enter the game consecutively at the same source node over time and decide upon arrival on a route through the network to the (common) destination. Intermediate edges and nodes are assumed to forward traffic according to a

FIFO strategy. In our work, we assume that all players are present initially at potentially different sources in the network and want to route to player-specific destinations. Intermediate nodes can have different queuing policies to forward traffic. However, our model is quite related to [39] for games with common source and sink, unweighted players and the FIFO policy. In this case, our existence result in Theorem 1 is similar to their main existence theorem, but the slightly different and discrete nature of our problem allows a much simpler argumentation. Despite some differences, tools from the area of flows over time can be of use also for the analysis of our models, e.g., for congestion games with time-dependent costs we use time-expanded networks in Section 4.

Finally, Farzad et al. [25] consider a priority-based scheme for both, non-atomic and atomic games. In their model players have priorities, and a resource yields different latencies depending on the priority of players allocating it. This includes an approach of Harks et al. [31] as a special case. While there can be different latencies for different players, this model does not include a more realistic "dynamic" effect that players delay other players only for a certain period of time. This is the case in our paper, as well as in [4, 39] for the non-atomic case.

1.1 Our Contribution

For temporal congestion games, we study four different (classes of) coordination mechanisms:

1. *FIFO*, in which tasks are processed non-preemptively in order of arrival, see Section 3.1,
2. *Non-preemptive global ranking*, in which there is a global ranking among the tasks that determines in which order tasks are processed non-preemptively (e.g., Shortest-First or Longest-First), see Section 3.2,
3. *Preemptive global ranking*, in which there is a global ranking that determines in which order tasks are processed and higher ranked tasks can preempt lower ranked tasks, see Section 3.3,
4. *Fair Time-Sharing*, in which all tasks currently located at an edge get processed simultaneously and each of them gets the same share of processing time, see Section 3.4.

Our interest is to characterize algorithmic properties of equilibria in these games. In particular, we are interested in existence of pure Nash equilibria, i.e., states that are resilient against unilateral player deviations. Pure Nash equilibria are the standard solution concept in static congestion games and have a natural and intuitive appeal. In addition to existence, an important aspect of equilibria is their computational complexity. If computing an equilibrium is hard, it is in general unreasonable to assume that an equilibrium can be obtained by the players. More importantly, we strive to obtain natural and simple strategy updating procedures that allow players to reach equilibria in a distributed and decentralized fashion. Our results on these issues are as follows.

For the FIFO policy (in unweighted single-source games) and the Shortest-First policy (in weighted single-source games) we show an interesting contrast of positive and negative results: even though computing a best response strategy for a player is NP-hard, there always exists a pure Nash equilibrium, which can be computed in polynomial time. It turns out that this is also a strong equilibrium [5], which is resilient to deviations of coalitions of players. In addition, there are a large number of natural improvement dynamics, using which the population of agents is able to find this strong equilibrium quickly even without solving computationally hard problems.

We then proceed to show that Shortest-First is the only global ranking that guarantees the existence of a pure Nash equilibrium in the non-preemptive setting. That is, for any other global ranking (e.g., Longest-First) there exist temporal congestion games without pure Nash equilibria. In contrast to this, we show that preemptive games are potential games for every global ranking and that uncoordinated agents can reach a pure Nash equilibrium quickly using improvement dynamics. Again, this pure Nash equilibrium is a strong equilibrium and therefore resilient against any coalitional deviation. Finally, we show that even though Fair Time-Sharing is an appealing coordination mechanism, it does not guarantee the existence of pure Nash equilibria, not even for identical tasks and networks with a common source and a common sink.

For the second model, congestion games with time-dependent costs, we prove that these games can be reduced to standard congestion games. Hence, they are potential games [41], and they have pure Nash equilibria and the finite improvement property. In addition, the known results on the price of anarchy for congestion games with corresponding delay functions carry over. We prove that computing a best response strategy in these games is NP-hard in general. Additionally, we show that even for a very restricted class of games with polynomially bounded delays and acyclic networks computing a pure Nash equilibrium is PLS-complete.

2 Notation

A *network congestion game* is described by a directed graph $G = (V, E)$, a set $\mathcal{N} = \{1, \dots, n\}$ of *players* with *source nodes* $s_1, \dots, s_n \in V$ and *target nodes* $t_1, \dots, t_n \in V$, and a non-decreasing *latency function* $\ell_e \colon [n] \to \mathbb{R}_{\geq 0}$ for each edge e. We will only consider linear latency functions of the form $\ell_e(x) = a_e x$ in this paper. For such functions, we call a_e the *speed* of edge e.[1] The *strategy space* Σ_i of a player $i \in \mathcal{N}$ is the set of all simple paths in G from s_i to t_i. We call a network congestion game *weighted* if additionally every player i has a weight $w_i \geq 1$, and *unweighted* if $w_1 = \ldots = w_n = 1$. Given a *state* $P = (P_1, \dots, P_n) \in \Sigma = \Sigma_1 \times \cdots \times \Sigma_n$ of a network congestion game, we denote by $n_e(P) = \sum_{i:e \in P_i} w_i$ the *congestion* of edge $e \in E$. The *individual latency* that a player i incurs is $\ell_i(P) = \sum_{e \in P_i} \ell_e(n_e(P))$, and every player is interested in choosing a path of minimum individual latency. We call a congestion game a *single-source game* if every player has the same source node s. If all players have the same source and target nodes, their strategy spaces are the same and we call the game an *s-t-network* game.[2] If not explicitly mentioned otherwise, we consider general unweighted network congestion games.

We incorporate time into the standard model in two different ways. Formally, this alters the individual latency functions ℓ_i. The specific definitions will be given in the sections below. For our altered games we are interested in stable states, which are pure strategy Nash equilibria of the games. Such an equilibrium is given by the condition that each player plays a best response and has no unilateral incentive to deviate, i.e., P is a *(pure) Nash equilibrium* if for every player i and every state Q that is obtained from P by replacing i's path by some other path, it holds $\ell_i(P) \leq \ell_i(Q)$, where ℓ_i denotes the (altered) latency function of player i. More generally, a state P has an *improving move* for a coalition of players C if there is a state Q obtained by replacing the path of some of the players in C by different paths, such that $\ell_i(P) > \ell_i(Q)$ for every player $i \in C$. A state P is a *strong equilibrium* if it has no improving move for any arbitrary coalition C.

[1] It would be more accurate to call a_e the *inverse of the speed*. However, to shorten terminology we call a_e just the *speed* of edge e.

[2] Usually, such games are called *symmetric network games*. In our temporal adjustment, however, s-t-network games will not be symmetric games because of different task weights and queuing priorities. Therefore, we resort to a different name here.

Note that we will not consider mixed Nash equilibria in this paper, and the term Nash equilibrium will refer to the pure version throughout.

3 Coordination Mechanisms

In this section we consider *temporal network congestion games.* These games are described by the same parameters as standard weighted network congestion games with linear latency functions. However, instead of assuming that a player allocates all edges on his chosen path instantaneously, we consider a scenario in which players consecutively allocate the edges on their paths. We assume that each player has a weighted task that needs to be processed by the edges on his chosen path.

Formally, at each point in time $\tau \in \mathbb{R}_{\geq 0}$, every task i is located at one edge $e_i(\tau)$ of its chosen path, and a certain fraction $f_i(\tau) \in [0,1]$ of it is yet unprocessed on that edge. The coordination mechanism run on edge e has to decide in each moment of time which task to process. If it decides to work on transmitting task i for $\Delta\tau$ time units starting at time τ, then the unprocessed fraction $f_i(\tau + \Delta\tau)$ of task i at time $\tau + \Delta\tau$ is $\max(0, f_i(\tau) - \Delta\tau/(a_e w_i))$. In total, task i needs $a_e w_i$ time units to finish on edge e. Once $f_i(\tau) = 0$, task i arrives at the next edge on its path and becomes available for processing. The coordination mechanism can base the decision on which task to process next and for how long only on local information available at the edge — such as the weights and arrival times of those tasks that have already arrived at the edge. The individual latency $\ell_i(P)$ of player i in state P is the time at which task i is completely finished on the last edge of P_i.

3.1 The FIFO Policy

One of the most natural coordination mechanisms is the FIFO policy. If several tasks are currently located at the same edge, then the one that has arrived first is executed non-preemptively until it finishes. In the case of ties, there may be an arbitrary tie-breaking that is consistent among the edges.

3.1.1 Unweighted and Single-Source Games

In this section we treat unweighted temporal network congestion games with a single source. For these games we obtain an interesting contrast of positive and negative results: even though computing a best response is NP-hard, there always exists a Nash equilibrium, which can be computed in polynomial time. Moreover, the equilibrium is not only efficiently computable, but uncoordinated agents are able to find it quickly even without solving computationally hard problems.

In order to prove that a Nash equilibrium can be computed efficiently, we will use the notion of *greedy best responses.* A greedy best response for player i is a path $s, v_1, \ldots, v_k = t$ from s to t such that for every intermediate node $v_{k'}$ the subpath $s, v_1, \ldots, v_{k'}$ is a shortest path from s to $v_{k'}$. To be more precise, given the current strategies of the other players, there is no possibility for player i to reach node $v_{k'}$ earlier than following the subpath $s, v_1, \ldots, v_{k'}$.

Let us remark that greedy best responses are the discrete analog of subpath-optimal flows introduced by Cole et al. [21]. The basic idea in the proof of Theorem 1 below is that if greedy best responses are played according to some player ordering, a Nash equilibrium will evolve. This approach has been used before in weighted network congestion games on parallel links [28] or classes of series-parallel graphs [29]. Before we turn to the proof, however, we note that, in general, greedy

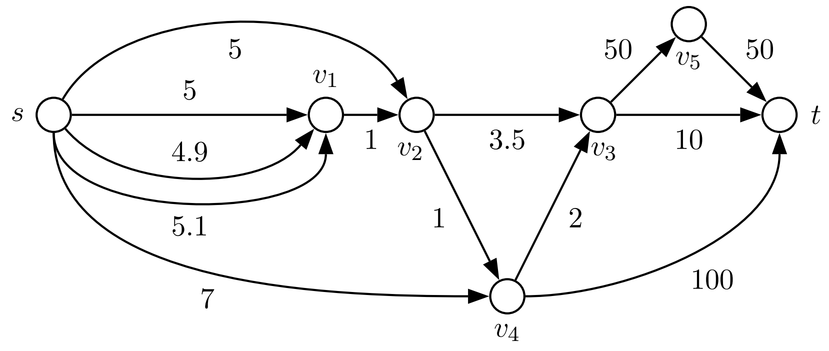

Figure 1: This example shows that not every best response is greedy and that greedy best responses do not always exist. It uses multi-edges, which can easily be substituted by normal edges if one adds additional nodes.

best responses are a strict subclass of best responses and do not always exist. Let us consider an example to illustrate this point.

Example 1. Consider the network depicted in Figure 1. Assume there are four unweighted players. The highest ranked player has chosen the path s, v_2, v_4, t. The second highest ranked player has chosen the path $s, v_1, v_2, v_4, v_3, v_5, t$, where he uses edge (s, v_1) of speed 5. The third highest ranked player has chosen the path s, v_4, v_3, t. Let us consider a best response of the fourth and lowest ranked player. If he chooses the path s, v_1, v_2, v_3, t with the edge (s, v_1) of speed 5.1, then he reaches node t at time 20.5. If he chooses the same path with (s, v_1) of speed 4.9, then he reaches node t only at time 29 because he is delayed at node v_3 by the third player. One can check that all other paths are even worse for the fourth player. Hence, choosing the aforementioned path with the edge of speed 5.1 is the only best response. It is, however, not greedy as the fourth player does not arrive at v_1 at the earliest possible time.

Theorem 1. *For unweighted single-source temporal network congestion games with the FIFO policy a Nash equilibrium always exists. Moreover, a Nash equilibrium can be computed efficiently.*

Proof. Let us assume without loss of generality that players are numbered according to their rank in tie-breaking, i.e. 1 is the highest ranked player, and n is the lowest ranked player. We claim that we obtain an equilibrium from an arbitrary state $P = (P_1, \ldots, P_n)$ if we let the players $1, 2, \ldots, n$ play each one greedy best response in this order. Let $\tilde{P}_1, \ldots, \tilde{P}_n$ denote the paths chosen by the players in these greedy best responses. We prove the following invariant: in each intermediate state $(\tilde{P}_1, \ldots, \tilde{P}_i, P_{i+1}, \ldots, P_n)$ and for each player $j \in \{1, \ldots, i\}$ the current path $\tilde{P}_j$ is a best response and none of these players can be delayed at any node by a lower ranked player $k > i$. Both these properties remain true even if all lower ranked players $k > i$ are allowed to change their paths arbitrarily.

For $i = 0$ this invariant is trivially true. For $i > 0$ we construct a distance function $d\colon V \to \mathbb{R}_{\geq 0}$ for the network $G = (V, E)$, which eventually tells us for every node how long it takes player i to get there. The construction of this distance function follows roughly Dijkstra's algorithm: Let $I \subseteq V$ denote the set of nodes that have already an assigned distance. We start with $I = \{s\}$ and

$d(s) = 0$. For extending the set I, we crucially use the fact that the players $1, \dots, i-1$ cannot be delayed by other players, which means that every edge $e \in E$ has a fixed schedule saying when it is used by the players $1, \dots, i-1$ and when it is available for player i. These fixed schedules imply in particular that for every node $v \in V$ there exists a shortest path $s, v_1, \dots, v_k = v$ for player i from s to v such that every subpath $s, v_1, \dots, v_{k'}$ is a shortest path from s to $v_{k'}$. Hence, taking into account the fixed schedules and the possible delays that they induce on player i, we can extend the set I as in Dijkstra's algorithm, that is, we insert a node $v \in V \setminus I$ into I that minimizes $\min_{u \in I} d(u) + \ell(u, v)$, where $\ell(u, v)$ denotes the time it takes player i to get from u to v if he arrives at node u at time $d(u)$. The distance $d(v)$ assigned to node v is $\min_{u \in I} d(u) + \ell(u, v)$. This algorithm implicitly constructs a path from s to any other node.

For any node u the distance $d(u)$ is by construction the earliest time at which player i can reach node u taking into account the strategies of the higher ranked players. Hence any path from s to the destination t_i of player i that can be constructed by this algorithm (the degree of freedom is the tie-breaking) is a greedy best response for player i. On any such path player i cannot be delayed at any node by a lower ranked player. Assume for contradiction that there is a node v and a player $j > i$ such that j arrives earlier at node v than i. This contradicts the construction of the path as it implies that there is a faster way to get from s to v. Again this argument crucially uses the property that the players $1, \dots, i-1$ cannot be delayed by lower ranked players. As player i cannot be delayed by lower ranked players, he reaches node t_i at the earliest possible time $d(t_i)$ if he follows the path computed by the algorithm regardless of the strategies of the lower ranked players. This proves that choosing such a path is a best response against all other players even if all lower ranked players are allowed to change their paths arbitrarily. This proves the correctness of the invariant.

The theorem follows from the correctness of the invariant and the efficient algorithm for computing a greedy best response for player i when players $1, \dots, i-1$ play already greedy best responses. □

The previous result can easily be extended to show that the derived Nash equilibrium is also a strong equilibrium. Suppose the Nash equilibrium allows an improving move for some coalition C. Consider the highest ranked player $i^* \in C$. Any greedy best response is a "dominant" strategy no matter what lower ranked players do. Thus, there is no way in which a strategy switch of lower ranked players can lead to a strict improvement in the delay of i^*. This contradicts that the move is improving for $i^* \in C$ and shows that the Nash equilibrium is really a strong equilibrium.

Corollary 1. *For unweighted single-source temporal network congestion games with the FIFO policy a strong equilibrium always exists. Moreover, a strong equilibrium can be computed efficiently.*

In addition to existence, the previous proof also shows that players reach the strong equilibrium in a distributed fashion using different forms of dynamics. Consider the following Nash dynamics among the players. At each point in time, one player is picked and allowed to change his strategy. We show below that in general it is NP-hard for this player to decide whether he can decrease his latency by changing his path. In that case, the player might stick to his current path or make an arbitrary strategy change, following some heuristic. However, at each point in time there is one player who can easily find a (greedy) best response, namely the highest ranked player $i + 1$ that does not play a greedy best response, but the players $1, \dots, i$ do. We assume that this player changes to a greedy best response when he becomes activated. We also assume that a player who is already playing a greedy best response does not change his strategy when he becomes activated.

A *round* is a sequence of activations in which every player gets at least once the chance to change his strategy. From the proof of Theorem 1 it follows easily that a Nash equilibrium is reached after at most n rounds. We are interested in particular in the *random greedy best response dynamics*, in which in each iteration the activated player is picked uniformly at random, and the *concurrent best response dynamics*, in which in each iteration all players are simultaneously allowed to change their strategy, each one with some constant probability $0 < p_i \leq 1$. In both these dynamics, rounds are polynomially long with high probability. In the random greedy best response dynamics the highest ranked player who does not yet play a greedy best response is picked with probability $1/n$. Hence, the expected number of rounds is $O(n^2)$. For the concurrent best response dynamics the expected number of rounds until player i is allowed to change his strategy is $1/p_i$. Hence, the expected number of rounds is $O(\sum_{i=1}^{n} 1/p_i)$. Summarizing, we obtain the following corollary.

Corollary 2. *In every unweighted single-source temporal network congestion game with the FIFO policy it takes at most n rounds to reach a strong equilibrium. In particular, the random and concurrent greedy best response dynamics reach a strong equilibrium in expected polynomial time.*

Finally, we turn to the hardness result.

Theorem 2. *Computing best responses is* NP*-hard in unweighted temporal s-t-network congestion games with the FIFO policy.*

Proof. We show how to reduce instances of 3-SAT to temporal network games with unweighted players and a single source and sink. Let an arbitrary instance for 3-SAT with variables $x_1, \ldots, x_n$ and clauses $C_1, \ldots, C_m$ be given, and assume that $C_j = l_{j1} \vee l_{j2} \vee l_{j3}$, where every literal l_{jk} is either x_i or $\overline{x_i}$ for one i. We assume that the literals are ordered such that l_{j1} belongs to a variable x_i and l_{j2} belongs to a variable $x_{i'}$ with $i < i'$. We assume the same monotonicity for l_{j2} and l_{j3}. The temporal congestion game that we construct has $1 + 7m$ players, one player p^D, who we call the decider and who is supposed to play a best response, one player p_j^C for every clause C_j, and the players p_{ij}^0 and $\tilde{p}_{ij}^0$ if clause C_j contains the literal $\overline{x_i}$ or the players p_{ij}^1 and $\tilde{p}_{ij}^1$ if clause C_j contains the literal x_i. For the construction it is only important that all players have higher priorities than the decider p^D, and that the players p_{ij}^0 and p_{ij}^1 have higher priorities than the clause players.

Figure 2 depicts the network that we construct. It is composed of the following parts:

- There are two rows of nodes, and both rows are subdivided into n blocks of $m+1$ nodes each. At the end of a block, there is the possibility to switch from the upper to the lower row or vice versa. Intuitively, each block corresponds to one variable x_i and the decider either uses the upper row in the block, corresponding to $x_i = 0$, or he uses the lower row, corresponding to $x_i = 1$. Both rows lead to a vertex t' from which there is a direct edge to the target t. All edges in the two rows (including the edges from s to the first nodes in the rows) have a speed of 1. All edges between the two rows also have a speed of 1. The speed of the edges from the last nodes in the rows to t' is $5nm$. The speed of the edge from t' to t is 1.

- If literal x_i occurs in clause C_j, then there is a direct edge from s to the j-th node in the i-th block in the upper row. If literal $\overline{x_i}$ occurs in clause C_j, then there is a direct edge from s to the j-th node in the i-th block in the lower row. In any case, we denote the speed of the edge by L_{ij}. It is chosen such that taking the direct edge is slightly slower than following a path along the rows (assuming no delays occur). To be more precise, we set $L_{ij} = (m+1)(i-1) + j + \varepsilon$, for a small $\varepsilon > 0$.

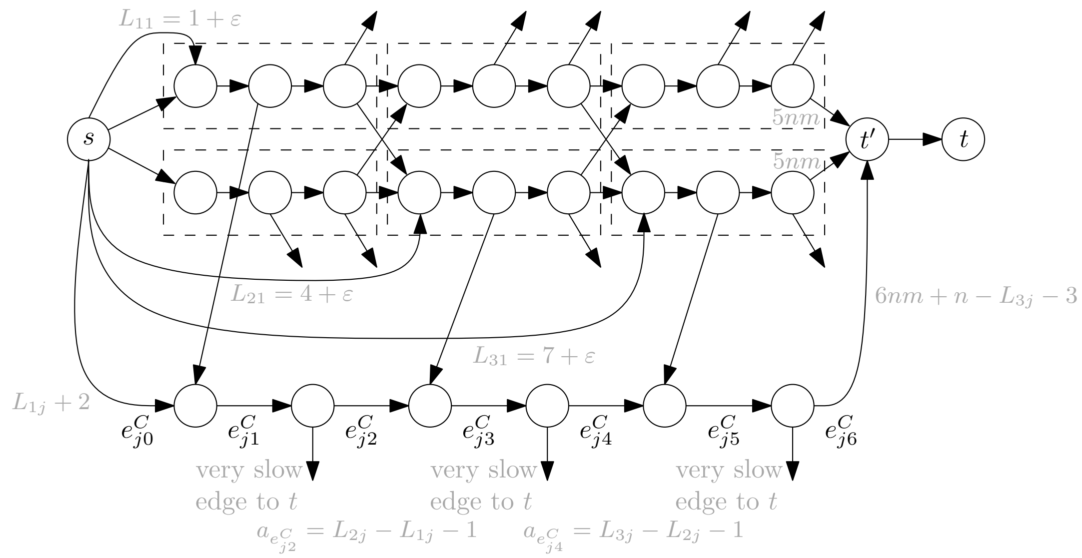

Figure 2: Construction in the proof of Theorem 2. Gray labels indicate speeds, black labels are the names of the edges. In this example, $n = 3$, $m = 2$, and the shown clause is $C_1 = x_1 \vee \overline{x_2} \vee \overline{x_3}$.

- In addition to the direct edges described above, there are additional direct edges. If literal x_i occurs in clause C_j, then there is a direct edge from s to node $(j+1)$ in the i-th block in the upper row. If literal $\overline{x_i}$ occurs in clause C_j, then there is a direct edge from s to node $(j+1)$ in the i-th block in the lower row. In any case, the speed of the edge is again L_{ij}. (These edges are not depicted in Figure 2.)
- For each clause $C_j = l_{j1} \vee l_{j2} \vee l_{j3}$, there is a path with seven edges $e^C_{j0}, \ldots, e^C_{j6}$ from the source node s to the node t'. Let the literals in clause C_j correspond to the variables i_1, i_2, and i_3, in this order. Then the speeds of the first, third, and fifth edge of the path are $L_{i_1 j} + 2$, $L_{i_2 j} - L_{i_1 j} - 1$, and $L_{i_3 j} - L_{i_2 j} - 1$, respectively. Due to the monotonicity among the literals, all these speeds are non-negative. The speeds of the second, fourth, and sixth edge are 1, and the speed of the seventh edge is $6nm + n - L_{i_3 j} - 3$, which is also non-negative as $L_{i_3 j} \leq n(m+1)$.
- Consider the j-th node in the i-th block for $j \in \{1, \ldots, m\}$ and $i \in \{1, \ldots, n\}$. If the k-th literal of clause C_j is x_i, then there is an edge of speed 1 from the $(j+1)$-th node in the i-th block of the upper row to the starting node of the edge $e^C_{jk'}$ for $k' = 2(k-1)+1$. If the k-th literal of clause C_j is $\overline{x_i}$, then there is an edge of speed 1 from the $(j+1)$-th node in the i-th block of the lower row to the starting node of the edge $e^C_{jk'}$ for $k' = 2(k-1)+1$. In both cases there is a very slow edge (say, with speed $10nm$) from the end node of $e^C_{jk'}$ to t.

Now we describe the current strategies of all players except for the decider, for whom we want to compute a best response.

- For each clause C_j the corresponding clause player p^C_j uses the path with the seven edges $e^C_{j0}, \ldots, e^C_{j6}$ from the source node s to the node t', from which he uses the direct edge to t.
- Let clause C_j be an arbitrary clause and let the k-th literal of C_j be x_i Then there are the two players p^1_{ij} and $\tilde{p}^1_{ij}$. Player p^1_{ij} uses the direct edge with speed L_{ij} from the source node

s to the j-th node in the i-th block of the upper row, then he follows the edge in this row to node $j+1$ in the i-th block from which goes to the starting node of the edge $e^C_{jk'}$ for $k' = 2(k-1)+1$. He then follows the edge $e^C_{jk'}$ and subsequently uses the slow direct edge to t. The player $\tilde{p}^1_{ij}$ is defined analogously with the only difference that he uses the edge with speed L_{ij} from s to node $(j+1)$ in the i-th block. From there he follows directly the outgoing edge to the starting node of $e^C_{jk'}$.

The players p^0_{ij} and $\tilde{p}^0_{ij}$ are defined analogously with the only exception that they use the lower row.

Now the s-t-network congestion game is completely defined and we claim that there exists a best response for the decider with latency at most $6nm+n+1$ if and only if the 3-SAT formula is satisfiable. This follows from the following observations about our construction.

- If the decider p^D sticks to the edges in the two rows to reach t', then he reaches each edge on his path first and is thus never delayed on his way to t'. This means, he reaches t' at time $6nm+n$.
- If the clause player p^C_j for some clause C is not delayed on his way to t', then he reaches t' also at time $6nm+n$.
- As the decider has the lowest priority of all players, we can draw the first conclusion: If the decider sticks to the edges in the two rows to reach t', then he can only have a delay of at most $6nm+n+1$ if all clause players are delayed.
- If the decider sticks to the edges of the two rows, then the players $\tilde{p}^0_{ij}$ and $\tilde{p}^1_{ij}$ do not interfere with any other player because when p^0_{ij} and p^1_{ij} arrive at the edges connecting the rows with the clause paths, the players $\tilde{p}^0_{ij}$ and $\tilde{p}^1_{ij}$ are already finished there.
- The decider can delay clause players by making the right choices between the upper and lower row. Let us consider a clause C_j and the corresponding path of seven edges. The speeds of these edges are chosen such that the clause player p^C_j, if not delayed, reaches the starting nodes of the edges e^C_{j1}, e^C_{j3}, and e^C_{j5} exactly at the same time as the corresponding players p^0_{ij} and p^1_{ij} if they are not delayed. As those players have higher priorities than the clause player p^C_j, they will delay him and make him reach t' only after the decider, unless they are delayed themselves before. To be more precise, for every clause C_j there are three corresponding players of the form p^0_{ij} or p^1_{ij} that overlap with the path of p^C_j. If and only if at least one of them is not delayed, then p^C_j will reach t' later than $6nm+n$.

Assume that the formula is satisfiable and let $x_1,\dots,x_n$ denote a satisfying assignment. If $x_i = 0$ in this assignment, then the decider p^D chooses the upper row of block i. Then he does not delay players p^0_{ij}, but he delays all players p^1_{ij}. This means that the clause players p^C_j of all clauses C_j that contain the literal $\overline{x_i}$ will reach t' only after the decider. If $x_i = 1$, then he chooses the lower row of block i and does not delay p^1_{ij}. This means that the clause players p^C_j of all clauses C_j that contain the literal x_i will reach t' only after the decider. As the assignment satisfies each clause, none of the clause players will reach t' at time $6nm+n$. This implies that the decider is not delayed and reaches t at time $6nm+n+1$, as desired.

Assume on the other hand that the formula is not satisfiable. By the above reasoning we obtain that the decider cannot delay all clause players if he sticks to the edges in the rows. That implies that always at least one clause player will reach node t' at the same time as the decider, which in turn implies that the decider reaches the node t later than at time $6nm + n + 1$, as desired.

The only remaining step in the proof is to show that the decider cannot benefit from using edges not within the rows. As he has a lower priority than the players p_{ij}^0, p_{ij}^1, $\tilde{p}_{ij}^0$, and $\tilde{p}_{ij}^1$ he cannot benefit from using the direct edges from s to nodes in the two rows. So the only possibility left for the decider is to follow the edges in the rows up to some node and to use the edge to one of the clause players' paths from there. Without the players $\tilde{p}_{ij}^0$ and $\tilde{p}_{ij}^1$, this might indeed be beneficial as the decider reaches a node on the path before the clause player. However, the players $\tilde{p}_{ij}^0$ and $\tilde{p}_{ij}^1$ cause the decider to wait ε time units before the edge to the path of the clause player becomes available for him. This means he reaches the node on the path of the clause player at exactly the same time as the clause player. As the decider has the lowest priority he will thus be delayed. Hence, the players $\tilde{p}_{ij}^0$ and $\tilde{p}_{ij}^1$ ensure that the decider sticks to the edges in the two rows, which concludes the proof. □

3.1.2 Weights and General Networks

Now we show that any relaxation of the restrictions in the previous sections leads to games without Nash equilibria.

Theorem 3. *There exist temporal congestion games with the FIFO policy and without Nash equilibria that (1) are weighted and s-t-network games, or (2) are unweighted.*

Proof. The example for the first case is simple; it consists of three edges: there are three nodes s, v, and t and two parallel edges from s to v (if multi edges are not allowed, they can be split up into two edges each by inserting intermediate nodes) and one edge from v to t. All edges have speed 1. Assume that there are two players with weights 2 and 3, and assume that the player with weight 3 has higher priority. If both players use the same edge from s to v, then the player with weight 2 has an incentive to switch to the free edge. If they use different edges, the player with weight 3 has an incentive to use the same edge as the other player.

Now let us turn to the second case. We consider the instance shown in Figure 3 (a). In this game there are three unweighted players, and each player i has two possible strategies: the vertical three edges (denoted by A_i) and another path (denoted by B_i). The following sequence of moves constitutes a cycle in the best response dynamics: $(A_1, A_2, A_3) \to (B_1, A_2, A_3) \to (B_1, B_2, A_3) \to (B_1, B_2, B_3) \to (A_1, B_2, B_3) \to (A_1, A_2, B_3) \to (A_1, A_2, A_3)$. It is easy to verify that the remaining configurations (A_1, B_2, A_3) and (B_1, A_2, B_3) are no Nash equilibria either. □

3.2 Non-preemptive Global Ranking

Another natural approach to queuing is to assume that there is a global ranking $\pi\colon [n] \to [n]$ on the set of tasks with $\pi(1)$ being the task with the highest priority and so on. In this case, tasks are scheduled non-preemptively according to this ranking. When an edge e becomes available, the highest ranked task i that is currently located at the edge is processed non-preemptively. It exclusively uses e for $a_e w_i$ time units. After that, task i moves to the next edge on its path, and e selects the next task if any. In this section, we consider mainly weighted games and assume without loss of generality that $w_1 \le w_2 \le \cdots \le w_n$.

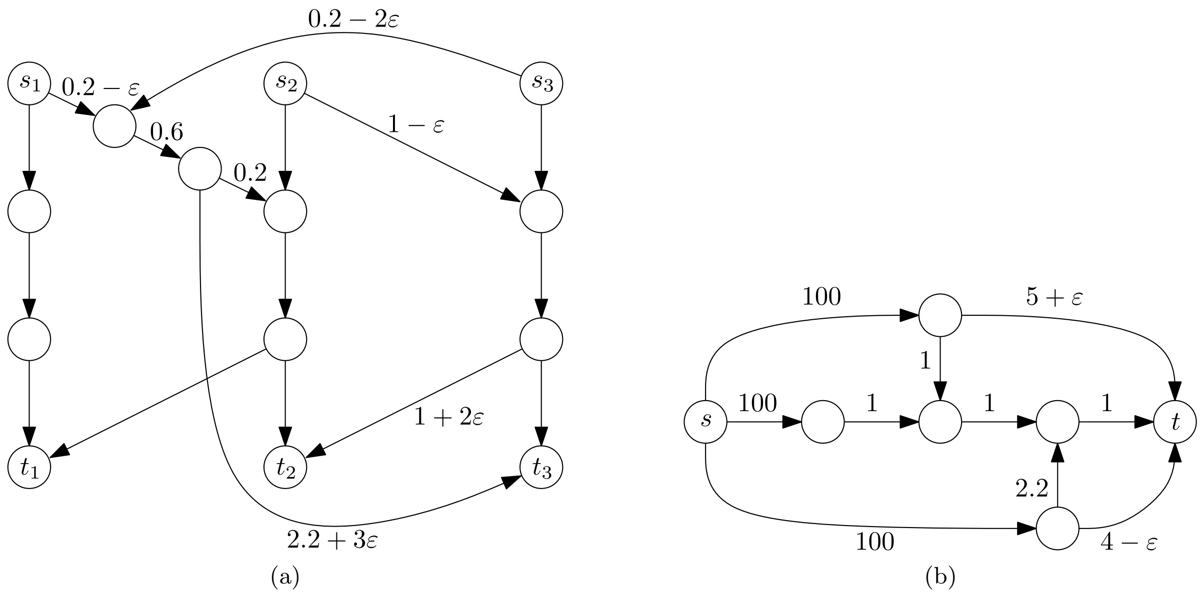

Figure 3: (a) General network game without Nash equilibrium for FIFO. Edge labels indicate the speeds a_e. For all unlabeled edges e, we have $a_e = 1$. (b) Unweighted s-t-network game without Nash equilibrium for Time-Sharing.

3.2.1 Shortest-First Policy

In this section we consider the identity ranking $\pi(i) = i$, which corresponds to the *Shortest-First policy*. It is easy to see that Theorem 1, Corollary 1 and Corollary 2 carry over to this case. The proof for FIFO was essentially based on the observation that once all players $1, \ldots, i$ play a (greedy) best response, they cannot be affected by the lower ranked players. This is even more true for the Shortest-First policy as the lower ranked players now face the additional disadvantage of having a longer processing time.

Theorem 4. *In every weighted single-source temporal network congestion game with the Shortest-First policy a strong equilibrium exists. It can be computed efficiently, as it takes at most n rounds to reach a strong equilibrium. In particular, the random and concurrent greedy best response dynamics reach a strong equilibrium in expected polynomial time.*

Also the hardness result in Theorem 2 carries over easily. We just need to set all weights to 1 and embed the same tie-breaking as in the proof of Theorem 2 in the ranking π. In the construction only the tie-breaking was important; the FIFO policy was never used, that is, it never happens that at a busy edge two players arrive one after another.

Theorem 5. *In (unweighted) temporal s-t-network congestion games with the Shortest-First policy computing a best response is* NP*-hard.*

Although the previous arguments guarantee existence and convergence to a Nash equilibrium for the Shortest-First policy, such games are not necessarily potential games.

Proposition 1. *There is a temporal s-t-network congestion game with the Shortest-First policy that is no potential game.*

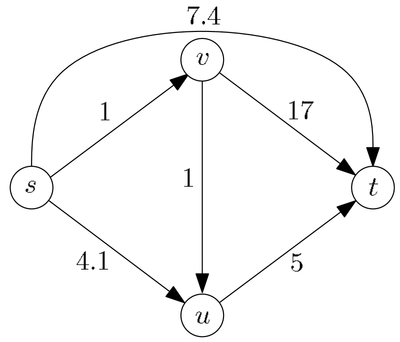

Figure 4: For two players with weights $w_1 = 1$ and $w_2 = 2$, this temporal network congestion game with Shortest-First policy is not a potential game.

Proof. The game is depicted in Figure 4. For $w_1 = 1$ and $w_2 = 2$ the following cycle can be repeated infinitely by better-response dynamics:

$$((s,u,t),(s,v,u,t)) \to ((s,v,t),(s,v,u,t)) \to ((s,v,t),(s,t))$$
$$\to ((s,u,t),(s,t)) \to ((s,u,t),(s,v,u,t)) \ .$$

□

3.2.2 Other Global Rankings or General Networks

In this section we consider the case of more general rankings. For technical reasons we need to slightly adjust a ranking in the presence of player tasks with equal weights. In particular, for a set of task weights we consider a numbering such that $w_1 \leq \ldots \leq w_n$. Consider a ranking π of the indices and a distinct weight w. The set of tasks with weight w corresponds to a consecutive interval of task numbers $\mathcal{N}_w = \{x, x+1, \ldots, y\}$. These tasks occupy a set of positions $\mathcal{P}_w = \{j_i \mid \pi(j_i) = i, i = x, \ldots, y\}$ in π. A ranking is called *normalized* if the tasks of $\mathcal{N}_w$ appear in strictly increasing order of their numbering at positions of $\mathcal{P}_w$, for any distinct weight w. Note that for every ranking π there is a unique corresponding normalized ranking. We can normalize a ranking π with respect to weight w by setting $j_x = \min \mathcal{P}_w$, $j_{x+1} = \min\{\mathcal{P}_w - \{j_x\}\}$ until $j_y = \max \mathcal{P}_w$, and then assign $\pi(j_i) = i$ for all $i = x, \ldots, y$. If we apply this ordering step for every weight, we arrive at the corresponding normalized ranking. As an example, if all weights are the same, there is only one normalized ranking, which is the identity $\pi(i) = i$.

When we use general normalized rankings other than the identity for queuing of player tasks at intermediate nodes, there always exists a game without a Nash equilibrium.

Theorem 6. *For any given set of players with task weights $w_1 \leq \cdots \leq w_n$ and any normalized ranking π other than the identity, there exist a graph and latency functions such that the resulting temporal s-t-network congestion game does not have a Nash equilibrium.*

Proof. Let j denote the index with the property that for player $i \in \{1, \ldots, j-1\}$ task w_i has the i-th highest priority, but player j with weight w_j does not have the j-th highest priority. Let w_k with $k > j$ be the weight of the player with the j-th highest priority. The network we construct consists of two levels of parallel links. On the first level there are n edges with speed 1. On the second level there are $k-1$ slow edges with speed a, where a is sufficiently large.

Now consider an arbitrary state of this game and assume that the players $w_1, \dots, w_{j-1}$ have chosen disjoint paths, which must be true in every Nash equilibrium. If one of the players $j, \dots, k-1$ has to share his edge on the first level with another player with a higher priority, then he will change to an unused edge on the first level. This edge is guaranteed to exist because there are n parallel links. On the other hand, if none of the players $j, \dots, k-1$ shares his edge with another task of higher priority, then players $1, \dots, k-1$ are the first ones that arrive at the intermediate node. Hence, for a sufficiently large, k has to wait for a long time until he can pass the second level. Hence, k has an incentive to change to an edge on the first level that is used by player l with $l \in \{j, \dots, k-1\}$. Since k has a higher priority, he will be able to arrive before l and he does not have to wait at the intermediate node. □

The same result holds for general games with the Shortest-First policy. We can simply add a separate source for each player and connect it via a single edge to the original source. By appropriately adjusting the delays a_e on these edges, we can ensure that smaller tasks are suitably delayed before arriving at the original source. This results in the same incentives and the absence of Nash equilibrium.

Corollary 3. *For any given set of players with task weights $w_1 \leq \cdots \leq w_n$ and the Shortest-First policy, there exist a graph and latency functions such that the resulting temporal network congestion game does not have a Nash equilibrium.*

3.3 Preemptive Global Ranking

When we assume a global ranking and allow preemptive execution, it is possible to adapt the arguments of Theorem 1 to general weighted temporal network congestion games. Indeed, all arguments in this section work for a very general class of preemptive games with unrelated edges. That is, every player i has its own processing time p_{ie} for every edge e. These processing times may even depend on the time at which player i reaches edge e. The only assumption we need to make is that the processing times are monotone in the sense that if task i reaches edge e at time t, then it does not finish later than when it reaches edge e at time $t' > t$.

Theorem 7. *Every temporal network congestion game with preemptive policy π is a potential game. A strong equilibrium exists and can be computed in polynomial time. For any state and any player, a best response can be computed in polynomial time.*

Proof. The main observation here is that no task $\pi(i)$ can influence the travel time of any task $\pi(j)$ with $j < i$, because it will be preempted whenever it is scheduled simultaneously with any of these tasks on an edge. This means that the vector $(\ell_{\pi(1)}(P), \dots, \ell_{\pi(n)}(P))$ decreases strictly lexicographically whenever a player changes his path and decreases his individual latency. This proves that any such game is a potential game, which contrasts e.g. Proposition 1. Note that the lexicographic decrease implies that the lexicographic minimum is also a strong equilibrium (c.f. [32]).

For efficient computation of a strong equilibrium, we consider iterative entry dynamics according to the ranking with best-response computation of players. By previous arguments this process outputs a strong equilibrium. For efficient computation of a best response strategy, we use the same variant of Dijkstra's algorithm that we have already used in the proof of Theorem 1. This time, however, a lower ranked task $\pi(i)$ can arrive at a node before a higher ranked task $j < i$ if it has a different source node. Then as soon as $\pi(j)$ arrives, $\pi(i)$ is preempted and blocked until it becomes the unfinished task of highest rank at the edge. Hence, the correctness of the algorithm is not affected by this. Finally, note that the previous algorithm does not rely on the fact that higher ranked players play a best response. The difference to Theorem 1 is that higher ranked players can never be delayed by lower ranked players even if they do not play best responses. Hence, the algorithm can be used in general to compute a best response for any player and any state. □

Note that our lexicographical potential function argument is similar to [10, Theorem 2], where standard congestion games are considered with a directed acyclic social knowledge graph restricting the latency dependencies among the players. While our scenario with a global ranking can be formulated in terms of a directed acyclic social knowledge graph, our games are somewhat different because we consider coordination mechanisms over time with preemption.

Similarly, we can adapt the previous observations in Corollary 2 and show that various improvement dynamics converge in polynomial time.

Corollary 4. *In every temporal network congestion game with any preemptive policy π, it takes at most n rounds to reach a strong equilibrium. The expected number of iterations to reach a strong equilibrium for random and concurrent best response dynamics is bounded by a polynomial.*

3.4 Fair Time-Sharing

In this section we consider fair time-sharing, a natural coordination mechanism based on the classical idea of fair queuing [43] and uniform processor sharing [38]. When multiple player tasks are present at an edge e, they are all processed simultaneously, and each one of them gets the same share of bandwidth or processing time. As in generalized processor sharing [45] we assume round-robin processing with infinitesimal time slots. Even though such a fairness property is desirable, the following theorem shows that Nash equilibria are not even guaranteed to exist for unweighted s-t-network games. This obviously remains true for extensions, where bandwidth is allocated using player priority weights (that might be different from the task weights), which are used e.g. in weighted fair queuing [22].

Theorem 8. *There is an unweighted temporal s-t-network congestion game with the Time-Sharing policy without a Nash equilibrium.*

Proof. The instance shown in Figure 3 (b) has three players. As the three edges leaving the source s are very slow, in any Nash equilibrium all three players will use different edges leaving the source. We assume without loss of generality that the first player chooses the upper edge, the second player chooses the middle edge, and the third player chooses the lower edge. Then players 1 and 3 still have two alternatives on how to continue, whereas the path of player 2 is already determined. The speeds of the edges are chosen such that player 1 wants to use the edge with speed $5+\varepsilon$ if and only if player 3 does not use the edge with speed $4-\varepsilon$. On the other hand, player 3 wants to use the edge with speed $4-\varepsilon$ if and only if player 1 uses the edge with speed $5+\varepsilon$, which completes the proof. □

Dürr and Nguyen [23] show that Time-Sharing on parallel links always yields a potential game, even for unrelated machines (edges). That is, for parallel links Nash equilibria always exist. Their potential function can be rewritten as the sum of the completion times (individual latencies) of the players. It is known [13] that a schedule minimizing this sum can be computed in polynomial time.

Such a global minimum of the potential function must obviously be a pure Nash equilibrium for the Time-Sharing policy, yielding the following corollary.

Corollary 5. *For games on parallel links with unrelated tasks and the Time-Sharing policy a Nash equilibrium can be computed efficiently.*

4 Constant Travel Times and Quality of Service

Now let us consider *network congestion games with time-dependent costs*. Again, players consecutively allocate the edges on their paths. However, the travel time along an edge e in the network is fixed to a constant delay d_e. If a player chooses a path along the edges $e_1, e_2, \ldots$, then he arrives at e_2 at time d_1 and at e_3 at time $d_1 + d_2$, and so on. This travel time through the network is independent of how many other players allocate any of the edges. We here consider the general case of asymmetric network games. For the strategic part we assume that each edge generates a separate usage cost c_e per time unit. This could, for instance, measure the quality of service that is enjoyed by the players during transmission. The cost depends on the number of players allocating the edge at a given point in time. In particular, edge e has a cost function $c_e \colon [n] \to \mathbb{N}$ that describes the cost for allocating it for one unit of time (measured, e.g., in milliseconds) in terms of the current number of players. If for a state P an edge e is shared at time τ by $n_e(\tau, P)$ players, all these players get charged cost $c_e(n_e(\tau, P))$. The cost incurred by player i on a path $P_i = (e_1, \ldots, e_l)$ is then $\ell_i(P) = \sum_{j=1}^{l} \sum_{\tau=\tau_j}^{\tau_j + d_{e_j} - 1} c_{e_j}(n_{e_j}(\tau, P))$, where $\tau_1 = 0$ and $\tau_j = \sum_{k=1}^{j-1} d_{e_k}$.

It turns out that this model is a subclass of standard congestion games. For each edge and each time unit we introduce a resource $r_{e,\tau}$ and modify the strategy spaces as follows: For a path $P = (e_1, \ldots, e_l)$ the new strategy includes all resources $r_{e_j,\tau}$ for $\tau = \tau_j, \ldots, \tau_j + d_{e_j} - 1$ and $j = 1, \ldots, l$. This is a standard congestion game with latencies given by the time costs. Hence, results on the existence of Nash equilibria and the price of anarchy carry over.

Corollary 6. *Network congestion games with time-dependent costs are equivalent to a class of standard congestion games. In particular, there is a pure Nash equilibrium in every game, and any better-response dynamics converges to Nash equilibrium.*

However, the standard congestion game obtained by this reduction might have a large number of resources. In addition, the standard game is not necessarily a network congestion game. Hence, the complexity results known for standard network congestion games do not carry over.

Theorem 9. *Computing a best response in network congestion games with time-dependent costs is* NP*-hard. For games with polynomially bounded delays and acyclic networks, best responses can be computed efficiently, but computing a Nash equilibrium is* PLS*-complete.*

Proof. The NP-hardness of computing a best response follows easily with a reduction from the partition problem. The input to this problem consists of n integers $w_1, \ldots, w_n$. One has to decide if there exists a subset of these numbers that add up to exactly $W/2$, where $W = \sum_{i=1}^{n} w_i$. We construct a graph with vertices $s = v_0, v_1, \ldots, v_n, t$ and consider the best response of a player whose source node is s and whose target node is t. Between each pair of nodes (v_i, v_{i+1}) there are two parallel edges with delays w_{i+1} and 0, respectively, and costs 0. In addition to this, there is one edge e from v_n to t with delay $W/2$ and cost function $c_e(n_e) = n_e$. We assume that we have two additional players, one of which arrives at edge e at time 0 and one of which arrives at edge e at time W. That is, only if the player manages to arrive at node v_n exactly at time $W/2$, then his costs on e will be $W/2$. Otherwise, it will be at least $W/2 + 1$. Any path from s to v_n corresponds to a subset of the weights w_i, and hence, the player has a strategy with costs $W/2$ if and only if the partition instance is solvable. This proves the NP-hardness.

Now we turn to acyclic networks with polynomially bounded delays. For this restricted case best responses can be computed efficiently by standard dynamic programming on time-expanded graphs. We store for each of the polynomially many time points τ and every node v the least expensive path that arrives at v exactly at time τ. First, we fill this table, taking into account only paths of length at most 1. From this, we can easily compute another table taking into account paths of length at most 2, and so on. This approach uses the fact that the network is acyclic, and it proves that the problem of computing a Nash equilibrium belongs to PLS.

For the completeness, we use the reduction in [2] for asymmetric network congestion games. This reduction has the property that it generates only acyclic networks. We will argue that there is a generic way to transform a standard network congestion game with acyclic network G into an acyclic network congestion game with time-dependent costs and polynomially bounded delays. For this, we take the network G and compute a topological ordering of the nodes. Let us assume without loss of generality that this ordering is $v_1, v_2, \ldots, v_k$, where v_1 has no incoming and v_k has no outgoing edges. If the source node of a player is v_i, then we introduce a new source node s_i for that player, which is connected by an edge with delay i and costs 0 to node v_i. This allows us to choose polynomially bounded delays for all edges such that every player whose path includes a node v_i arrives at this node at exactly time i. Hence, if we can keep the cost functions, this congestion game with time-dependent costs behaves exactly as the standard congestion games as players are now synchronized. □

5 Conclusion

In this paper we study atomic network congestion games involving a notion of time, which is an important aspect of routing that is neglected by standard congestion games. Our results reveal an interesting contrast of efficient computation and convergence on the one hand and hardness of computing best responses and/or equilibria on the other hand. An obvious open problem is to derive a realistic non-preemptive coordination mechanism that always admits pure Nash or even strong equilibria for all temporal network congestion games. More generally, we have not addressed the inefficiency of equilibria in our case. It would be interesting to see in which way results and characterizations for (the cost of) Nash equilibria for coordination mechanisms on parallel links can be helpful here. Finally, there is always the challenge to improve existing models by incorporating other important and challenging aspects of realistic routing scenarios.

References

[1] Heiner Ackermann, Petra Berenbrink, Simon Fischer, and Martin Hoefer. Concurrent imitation dynamics in congestion games. In *Proc. 28th Symp. Principles of Distributed Computing (PODC)*, pages 63–72, 2009.

[2] Heiner Ackermann, Heiko Röglin, and Berthold Vöcking. On the impact of combinatorial structure on congestion games. *J. ACM*, 55(6), 2008.

[3] Nir Andelman, Michal Feldman, and Yishay Mansour. Strong price of anarchy. *Games Econom. Behav.*, 65(2):289–317, 2009.

[4] Elliot Anshelevich and Satish Ukkusuri. Equilibria in dynamic selfish routing. In *Proc. 2nd Intl. Symp. Algorithmic Game Theory (SAGT)*, pages 171–182, 2009.

[5] Robert Aumann. Acceptable points in general cooperative n-person games. In *Contributions to the Theory of Games IV*, volume 40 of *Annals of Mathematics Study*, pages 287–324. Princeton University Press, 1959.

[6] Baruch Awerbuch, Yossi Azar, and Amir Epstein. The price of routing unsplittable flow. In *Proc. 37th Symp. Theory of Computing (STOC)*, pages 331–337, 2005.

[7] Yossi Azar, Kamal Jain, and Vahab Mirrokni. (Almost) optimal coordination mechanisms for unrelated machine scheduling. In *Proc. 19th Symp. Discrete Algorithms (SODA)*, pages 323–332, 2008.

[8] Anand Bhalgat, Tanmoy Chakraborty, and Sanjeev Khanna. Nash dynamics in congestion games with similar resources. In *Proc. 5th Intl. Workshop Internet & Network Economics (WINE)*, pages 362–373, 2009.

[9] Umang Bhaskar, Lisa Fleischer, and Elliot Anshelevich. A Stackelberg strategy for routing flow over time. In *Proc. 22nd Symp. Discrete Algorithms (SODA)*, pages 192–201, 2011.

[10] Vittorio Bilò, Angelo Fanelli, Michele Flammini, and Luca Moscardelli. Graphical congestion games. In *Proc. 4th Intl. Workshop Internet & Network Economics (WINE)*, pages 660–671, 2008.

[11] Avrim Blum, Eyal Even-Dar, and Katrina Ligett. Routing without regret: On convergence to Nash equilibria of regret-minimizing algorithms in routing games. *Theory of Computing*, 6(1):179–199, 2010.

[12] Avrim Blum, MohammadTaghi Hajiaghayi, Katrina Ligett, and Aaron Roth. Regret minimization and the price of total anarchy. In *Proc. 40th Symp. Theory of Computing (STOC)*, pages 373–382, 2008.

[13] J. Bruno, E. Coffman Jr., and R. Sethi. Scheduling independent tasks to reduce mean finishing time. *Comm. ACM*, 17(7):382–387, 1974.

[14] Ioannis Caragiannis. Efficient coordination mechanisms for unrelated machine scheduling. In *Proc. 20th Symp. Discrete Algorithms (SODA)*, pages 815–824, 2009.

[15] Tanmoy Chakraborty and Sanjeev Khanna. Nash dynamics in constant player and bounded jump congestion games. In *Proc. 2nd Intl. Symp. Algorithmic Game Theory (SAGT)*, pages 196–207, 2009.

[16] Huey-Kuo Chen. *Dynamic Travel Choice Models: A Variational Inequality Approach*. Springer Verlag, 1999.

[17] Steve Chien and Alistair Sinclair. Strong and Pareto price of anarchy in congestion games. In *Proc. 36th Intl. Coll. Automata, Languages and Programming (ICALP)*, pages 279–291, 2009.

[18] Steve Chien and Alistair Sinclair. Convergence to approximate Nash equilibria in congestion games. *Games Econom. Behav.*, 71(2):315–327, 2011.

[19] George Christodoulou and Elias Koutsoupias. The price of anarchy in finite congestion games. In *Proc. 37th Symp. Theory of Computing (STOC)*, pages 67–73, 2005.

[20] George Christodoulou, Elias Koutsoupias, and Akash Nanavati. Coordination mechanisms. *Theoret. Comput. Sci.*, 410(36):3327–3336, 2009.

[21] Richard Cole, Yevgeniy Dodis, and Tim Roughgarden. Bottleneck links, variable demand, and the tragedy of the commons. In *Proc. 17th Symp. Discrete Algorithms (SODA)*, pages 668–677, 2006.

[22] Alan Demers, Srinivasan Keshav, and Scott Shenker. Analysis and simulation of a fair queueing algorithm. In *Proc. Symp. Communications Architectures and Protocols (SIGCOMM)*, pages 1–12, 1989.

[23] Christoph Dürr and Nguyen Kim Thang. Non-clairvoyant scheduling games. In *Proc. 2nd Intl. Symp. Algorithmic Game Theory (SAGT)*, pages 135–146, 2009.

[24] Alex Fabrikant, Christos Papadimitriou, and Kunal Talwar. The complexity of pure Nash equilibria. In *Proc. 36th Symp. Theory of Computing (STOC)*, pages 604–612, 2004.

[25] Babak Farzad, Neil Olver, and Adrian Vetta. A priority-based model of routing. *Chicago J. Theoret. Comput. Sci.*, 2008.

[26] Simon Fischer, Harald Räcke, and Berthold Vöcking. Fast convergence to Wardrop equilibria by adaptive sampling mehtods. *SIAM J. Comput.*, 39(8):3700–3735, 2010.

[27] Lisa Fleischer and Éva Tardos. Efficient continuous-time dynamic network flow algorithms. *Oper. Res. Lett.*, 23(3–5):71–80, 1998.

[28] Dimitris Fotakis, Spyros Kontogiannis, Elias Koutsoupias, Marios Mavronicolas, and Paul Spirakis. The structure and complexity of Nash equilibria for a selfish routing game. *Theoret. Comput. Sci.*, 410(36):3305–3326, 2009.

[29] Dimitris Fotakis, Spyros Kontogiannis, and Paul Spirakis. Symmetry in network congestion games: Pure equilibria and anarchy cost. In *Proc. 3rd Intl. Workshop Approximation and Online Algorithms (WAOA)*, pages 161–175, 2005.

[30] Horst Hamacher and Stevanus Tjandra. Mathematical modelling of evacuation problems: A state of the art. In M. Schreckenberg and S. Sharma, editors, *Pedestrian and Evacuation Dynamics*, pages 227–266. Springer Verlag, 2002.

[31] Tobias Harks, Stefan Heinz, and Marc Pfetsch. Competitive online multicommodity routing. *Theory Comput. Syst.*, 45(3):533–554, 2009.

[32] Tobias Harks, Max Klimm, and Rolf Möhring. Strong Nash equilibria in games with the lexicographical improvement property. In *Proc. 5th Intl. Workshop Internet & Network Economics (WINE)*, pages 463–470, 2009.

[33] Martin Hoefer, Vahab Mirrokni, Heiko Röglin, and Shanghua Teng. Competitive routing over time. In *Proc. 5th Intl. Workshop Internet & Network Economics (WINE)*, pages 18–29, 2009.

[34] Martin Hoefer and Alexander Skopalik. On the complexity of Pareto-optimal Nash and strong equilibria. In *Proc. 3rd Intl. Symp. Algorithmic Game Theory (SAGT)*, pages 312–322, 2010.

[35] Ron Holzman and Nissan Law-Yone. Strong equilibrium in congestion games. *Games Econom. Behav.*, 21(1-2):85–101, 1997.

[36] Nicole Immorlica, Li Li, Vahab Mirrokni, and Andreas Schulz. Coordination mechanisms for selfish scheduling. *Theoret. Comput. Sci.*, 410(17):1589–1598, 2009.

[37] Robert Kleinberg, Georgios Piliouras, and Éva Tardos. Multiplicative updates outperform generic no-regret learning in congestion games. In *Proc. 41st Symp. Theory of Computing (STOC)*, pages 533–542, 2009.

[38] Leonard Kleinrock. *Queueing Systems Vol. 2: Compurer Applications*. Wiley, 1976.

[39] Ronald Koch and Martin Skutella. Nash equilibria and the price of anarchy for flows over time. In *Proc. 2nd Intl. Symp. Algorithmic Game Theory (SAGT)*, pages 323–334, 2009.

[40] Martin Macko, Kate Larson, and Luboš Steskal. Braess's paradox for flows over time. In *Proc. 3rd Intl. Symp. Algorithmic Game Theory (SAGT)*, pages 262–275, 2010.

[41] Dov Monderer and Lloyd Shapley. Potential games. *Games Econom. Behav.*, 14:1124–1143, 1996.

[42] Richard Mounce. Convergence in a continuous dynamic queuing model for traffic networks. *Transportation Res. Part B*, 40(9):779–791, 2006.

[43] John Nagle. On packet switches with infinite storage. *IEEE Trans. Comm.*, 35(4):435–438, 1987.

[44] Noam Nisan, Éva Tardos, Tim Roughgarden, and Vijay Vazirani, editors. *Algorithmic Game Theory*. Cambridge University Press, 2007.

[45] Abhay Parekh and Robert Gallager. A generalized processor sharing approach to flow control in integrated services networks: The single-node case. *IEEE/ACM Trans. Netw.*, 1(3):344–357, 1993.

[46] Srinivas Peeta and Athanasios Ziliaskopoulos. Foundations of dynamic traffic assignment: The past, the present and the future. *Networks and Spatial Economics*, 1(3–4):233–265, 2001.

[47] Bin Ran and David Boyce. *Modelling Dynamic Transportation Networks*. Springer Verlag, 1996.

[48] Robert Rosenthal. A class of games possessing pure-strategy Nash equilibria. *Intl. J. Game Theory*, 2:65–67, 1973.

[49] Tim Roughgarden. Routing games. In Nisan et al. [44], chapter 18.

[50] Tim Roughgarden. Intrinsic robustness of the price of anarchy. In *Proc. 41st Symp. Theory of Computing (STOC)*, pages 513–522, 2009.

[51] Alexander Skopalik and Berthold Vöcking. Inapproximability of pure Nash equilibria. In *Proc. 40th Symp. Theory of Computing (STOC)*, pages 355–364, 2008.

[52] Martin Skutella. An introduction to network flows over time. In W. Cook, L. Lovácz, and J. Vygen, editors, *Research Trends in Combinatorial Optimization*, chapter 21, pages 451–482. SV, 2009.

[53] Mike Smith. A new dynamic traffic model and the existence and calculation of dynamic user equilibria on congested capacity-constrained road networks. *Transportation Res. Part B*, 27(1):49–63, 1993.

[54] William Vickrey. Congestion theory and transport investment. *Amer. Econ. Rev.*, 59(2):251–260, 1969.

[55] Berthold Vöcking. Selfish load balancing. In Nisan et al. [44], chapter 20.

[56] J.G. Wardrop. Some theoretical aspects of road traffic research. In *Proceedings of the Institute of Civil Engineers, Pt. II*, volume 1, pages 325–378, 1952.

[57] Samuel Yagar. Dynamic traffic assignment by individual path minimization and queuing. *Transportation Res.*, 5(3):179–196, 1971.