Competitive Routing over Time

Martin Hoefer[1]*, Vahab S. Mirrokni[2], Heiko Röglin[3]**, and Shang-Hua Teng[4]

[1] Department of Computer Science, RWTH Aachen University
mhoefer@cs.rwth-aachen.de
[2] Google Research New York
mirrokni@google.com
[3] Department of Quantitative Economics, Maastricht University
heiko@roeglin.org
[4] Computer Science Department, University of Southern California
shanghua.teng@gmail.com

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 propose *temporal network congestion games* that use 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. In some cases our results can be used to characterize convergence for various distributed dynamics.

1 Introduction

As an intuitive game-theoretic model for competitive resource usage, *network congestion games* have recently attracted a great deal of attention [1–3]. These games are central in modeling routing and scheduling tasks with distributed control [4]. 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

* Supported by DFG through UMIC Research Centre at RWTH Aachen University.
** Supported by a fellowship within the Postdoc-Program of the German Academic Exchange Service (DAAD).

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 street 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 particular, each edge may have a *local queueing 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 queueing policy on each edge. To model these local scheduling policies, we use the idea of *coordination mechanisms* [5–8] that have been introduced in the context of machine scheduling and selfish load balancing [9]. 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. We model this via a time-dependent cost function. 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 atomic congestion games, which were initially considered by Rosenthal [3]. 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 complexity of computing equilibria [1–3] and bounding their inefficiency [10–12]. For an overview and introduction to the topic we refer to the recent expositions by Roughgarden [4] and Vöcking [9]. Addressing the notion of time in congestion games has only been started very recently in a number of papers [13–15]. Koch and Skutella [15] present a general model for flows over time using queueing models. Similarly, Anshelevich and Ukkusuri [13] derive a number of related results for a similar model of flows over time. In contrast to our work both papers address non-atomic congestion games, in which players are infinitesimally small flow particles. Farzad et al. [14] 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. [16] 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 [13, 15] 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. (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). (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. (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.

For the FIFO policy (in unweighted symmetric games) and the Shortest-First policy (in weighted symmetric games) we show an interesting contrast of positive and negative results: even though computing a best response is NP-hard, there always exists an equilibrium, which can be computed in polynomial time. Moreover, the equilibrium is not only efficiently computable, but we present natural dynamics in which uncoordinated agents are able to find an equilibrium quickly even without solving computationally hard problems. We then show that Shortest-First is the only global ranking that guarantees the existence of Nash equilibria in the non-preemptive setting. That is, for any other global ranking (e.g., Longest-First) there exist temporal congestion games without equilibria. In contrast to this, we show that preemptive games are potential games for every global ranking and that uncoordinated agents reach an equilibrium quickly. Finally, we show that even though Fair Time-Sharing sounds like an appealing coordination mechanism it does not guarantee the existence of equilibria, not even for unweighted symmetric games.

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, and in addition the known results on the price of anarchy carry over. We prove that computing a best response in these games is NP-hard in general. Even for a very restricted class of games with polynomially bounded delays and acyclic networks computing an equilibrium is PLS-complete. Due to space limitations, some proofs are deferred to the full version of this paper.

2 Notation

A *network congestion game* is described by a directed graph $G = (V, E)$, a set $\mathcal{N} = \{1, \ldots, n\}$ of *players* with *source nodes* $s_1, \ldots, s_n \in V$ and *target nodes* $t_1, \ldots, t_n \in V$, 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. 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* $P = (P_1, \ldots, P_n) \in \Sigma = \Sigma_1 \times \cdots \times \Sigma_n$ 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 *symmetric* if every player has the same source node and every player has the same target node. If not explicitly mentioned otherwise, we consider unweighted asymmetric 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. 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 $f_i(\tau + \Delta t)$ of task i at time $\tau + \Delta t$ is $\max(0, f_i(t) - \Delta\tau/(a_e w_i))$. 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 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.

Unweighted and Symmetric Games In this section we treat unweighted symmetric temporal network congestion games. 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.

Theorem 1. *For unweighted symmetric 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. Assume that we start in an arbitrary state of the game in which the players have chosen arbitrary paths. Below we define a subclass of best responses, which we call *greedy best responses*. We claim that we obtain an equilibrium if we let the players $1, 2, \ldots, n$ play each one greedy best response in this order. To prove this, assume that the players $1, \ldots, i$ are already playing greedy best responses, and now let player $i+1$ also change his strategy to a greedy best response. We show that after this strategy change the players $1, \ldots, i$ are still playing greedy best responses, which proves by induction that a Nash equilibrium is reached once every player has played a greedy best response. We prove the following invariant: if the players $1, \ldots, i$ play greedy best responses, then none of them can be delayed at any node by a lower ranked player $j > i$. Furthermore, the current paths of the players $1, \ldots, i$ are (greedy) best responses no matter which paths the other players $j > i$ choose. For the first player, every best response is defined to be a greedy best response. Given this definition, we argue that the aforementioned claim is true for $i = 1$: We consider the network $G = (V, E)$ as a weighted graph in which every edge $e \in E$ has weight a_e. Let P_1 denote a shortest path in this weighted graph from the source s to the target t and let a^* denote its length. If the highest ranked player chooses path P_1, then he cannot be delayed at any node v by any other player j, as otherwise, j would have found a shorter path from s to v, contradicting the choice of P_1 as shortest path from s to t. Hence, when player 1 chooses path P_1 his total latency is a^* no matter which paths the other players choose. Clearly, the length a^* is also a lower bound on the time it takes any player to reach the target, and hence, choosing P_1 is a (greedy) best response for player 1. Moreover, any (greedy) best response of player 1 corresponds to a shortest path P_1 in the aforementioned weighted graph. Now let us recursively define what a greedy best response is for player $i + 1 > 1$. For this, assume that the players $1, \ldots, i$ play already greedy best responses. Based on the paths chosen by these players, 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 $i + 1$ 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 $1, \ldots, i$ 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 $1, \ldots, i$ and when it is available for player $i+1$. These fixed schedules imply in particular that for every node $v \in V$ there exists a shortest path $s, v_1, \ldots, v_k = v$ for player $i+1$ from s to v such that every subpath $s, v_1, \ldots, v_{k'}$ 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 $i+1$, we can extend the set I as in Dijkstra's algorithm, that is, we insert the 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 $i + 1$ 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 constructs implicitly a path from s to any other node. Any path from s to t that can be constructed by this algorithm (the degree of freedom is the tie-breaking) is called a greedy best response of player $i + 1$.

It is easy to see that any such greedy best response is really a best response for player $i + 1$ if only the players $1, \ldots, i + 1$ are present, because there is no quicker way to reach the target t from the source s given the current paths of the players $1, \ldots, i$. To complete the proof we need to argue that player $i + 1$ cannot be delayed by players $j > i + 1$. Assume there is a node v and a player $j > i + 1$ such that j arrives earlier at node v than $i + 1$. This contradicts the construction of the path as it implies that there is a faster way to get from the source s to the node v. Again this argument crucially uses the property that the players $1, \ldots, i$ cannot be delayed by lower ranked players. $\square$

Let us remark that greedy best responses defined in the previous proof are the discrete analogon of subpath-optimal flows introduced by Cole et al. [17]. Basically, a path $s, v_1, \ldots, v_k, t$ is a greedy best response for player i if any subpath $s, v_1, \ldots, v_{k'}$ is a shortest path from s to $v_{k'}$. Note that this is not the case in arbitrary best responses: it could, for example, be the case that player i has to wait at some node $v_{k'}$ because he is blocked by a player with a higher rank. Then, the subpath from s to $v_{k'}$ is not necessarily the shortest possible path in every best response. However, we believe that the restriction to greedy best responses is a natural assumption on the players' behavior.

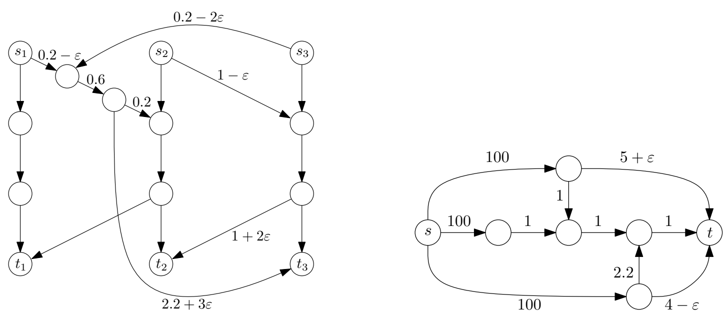
The previous result shows not only that a Nash equilibrium always exists, but it also shows that players reach it in a distributed fashion using different forms of dynamics. Consider the following Nash dynamics among the players. At each point in time, one player is picked and allowed to change his strategy. We show below that in general it is NP-hard for this player to decide whether it 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 $1, \ldots, i$ 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. Summarizing, we obtain the following corollary.

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

Finally, we turn to the hardness result.

Theorem 3. *Computing best responses is NP-hard in unweighted symmetric temporal network congestion games with the FIFO policy.*

Weights and Asymmetric Players Now we show that any relaxation of the restrictions in the previous sections leads to games without equilibria.



(a) (b)

Fig. 1: (a) Asymmetric temporal network congestion 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 symmetric game without Nash equilibrium for Time-Sharing.

Theorem 4. *There exist temporal congestion games with the FIFO policy that do not possess Nash equilibria and (1) are weighted and symmetric, or (2) are unweighted and asymmetric.*

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 1 (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. $\square$

3.2 Non-preemptive Global Ranking

Another natural approach 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 possible. In this section, we consider mainly weighted games and assume without loss of generality that $w_1 \leq w_2 \leq \cdots \leq w_n$.

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 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 5. *In every weighted symmetric temporal network congestion game with the Shortest-First policy a Nash equilibrium exists. Moreover, a Nash equilibrium can be computed efficiently, and it takes at most n rounds to reach a Nash equilibrium. In particular, the random and concurrent greedy best response dynamics reach a Nash equilibrium in expected polynomial time.*

Also the hardness result in Theorem 3 carries over easily.

Theorem 6. *In (unweighted) symmetric temporal 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 7. *There is a symmetric temporal network congestion game with the Shortest-First policy that is no potential game.*

Other Global Rankings or Asymmetric Players Now we consider the case of more general rankings.

Theorem 8. *For any given set of player task weights $w_1 \leq \cdots \leq w_n$ and any ranking π other than the identity, there exist a graph and latency functions such that the resulting symmetric temporal congestion game does not possess a Nash equilibrium.*

The proof is given in the full version of the paper. It relies on the fact that for rankings other than the identity a larger task can delay smaller tasks near the source due to the ranking, but smaller tasks can delay larger tasks near the sink due to faster travel time. The same result holds for asymmetric 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.

Corollary 9. *For any given set of task weights $w_1, \ldots, w_n$ and the Shortest-First policy, there exist a graph and latency functions such that the resulting asymmetric temporal 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 weighted asymmetric 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 10. *Every asymmetric temporal congestion game with preemptive policy π is a potential game. A Nash equilibrium exists and can be computed in polynomial time. For any state and any player, a best response can be computed in polynomial time.*

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

Corollary 11. *In every asymmetric temporal network congestion game with any preemptive policy π, it takes at most n rounds to reach a Nash equilibrium. The expected number of iterations to reach a Nash 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 coodination mechanism based on the classical idea of uniform processor sharing [18]. When multiple 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 [19] 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 symmetric unweighted games.

Theorem 12. *There is an unweighted symmetric temporal network congestion game with the Time-Sharing policy that does not have a Nash equilibrium.*

Proof. The instance shown in Figure 1 (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 have still two alternatives 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 [20] 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 player. It is known [21] 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 13. *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. In this section, we only consider asymmetric 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 second 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 $\tau_j = \sum_{k=1}^{j-1} d_{e_j}$.

It turns out that this model is equivalent to a regular congestion game. 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 regular 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 14. *Network congestion games with time-dependent costs are equivalent to a class of regular congestion games. In particular, there is a pure Nash equilibrium in every game, and any better-response dynamics converges.*

However, as the standard congestion game obtained by this reduction might have a large number of resources and as it is not necessarily a network congestion game, complexity results do not carry over.

Theorem 15. *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.*

References

1. Ackermann, H., Röglin, H., Vöcking, B.: On the impact of combinatorial structure on congestion games. Journal of the ACM **55** (2008)
2. Fabrikant, A., Papadimitriou, C.H., Talwar, K.: The complexity of pure Nash equilibria. In: Proc. 36th Symp. Theory of Computing (STOC). (2004) 604–612
3. Rosenthal, R.W.: A class of games possessing pure-strategy Nash equilibria. International Journal of Game Theory **2** (1973) 65–67
4. Roughgarden, T.: Routing Games. In: Algorithmic Game Theory. Cambridge University Press (2007) 461–486
5. Azar, Y., Jain, K., Mirrokni, V.S.: (Almost) optimal coordination mechanisms for unrelated machine scheduling. In: Proc. 19th Symp. Discrete Algorithms (SODA). (2008) 323–332
6. Caragiannis, I.: Efficient coordination mechanisms for unrelated machine scheduling. In: Proc. 20th Symp. Discrete Algorithms (SODA). (2009) 815–824
7. Christodoulou, G., Koutsoupias, E., Nanavati, A.: Coordination mechanisms. In: Proc. 31st Intl. Coll. Automata, Languages and Programming (ICALP). (2004) 345–357
8. Immorlica, N., Li, L., Mirrokni, V.S., Schulz, A.S.: Coordination mechanisms for selfish scheduling. In: Proc. 1st Intl. Workshop on Internet and Network Economics (WINE). (2005) 55–69
9. Vöcking, B.: Selfish Load Balancing. In: Algorithmic Game Theory. Cambridge University Press (2007) 517–542
10. Christodoulou, G., Koutsoupias, E.: The price of anarchy of finite congestion games. In: Proc. 37th Symp. Theory of Computing (STOC). (2005) 67–73
11. Awerbuch, B., Azar, Y., Epstein, A.: The price of routing unsplittable flow. In: Proc. 37th Symp. Theory of Computing (STOC). (2005) 57–66
12. Roughgarden, T.: Intrinsic robustness of the price of anarchy. In: Proc. 41st Symp. Theory of Computing (STOC). (2009)
13. Anshelevich, E., Ukkusuri, S.: Equilibria in dynamic selfish routing. In: Proc. 2nd Intl. Symp. Algorithmic Game Theory (SAGT). (2009)
14. Farzad, B., Olver, N., Vetta, A.: A priority-based model of routing. Chicago Journal of Theoretical Computer Science (2008) Article no. 1.
15. Koch, R., Skutella, M.: Nash equilibria and the price of anarchy for flows over time. In: Proc. 2nd Intl. Symp. Algorithmic Game Theory (SAGT). (2009)
16. Harks, T., Heinz, S., Pfetsch, M.: Competitive online multicommodity routing. In: Proc. 4th Intl. Workshop on Approximation and Online Algorithms (WAOA). (2006) 240–252
17. Cole, R., Dodis, Y., Roughgarden, T.: Bottleneck links, variable demand, and the tragedy of the commons. In: Proc. 17th Symp. Discrete Algorithms (SODA). (2006) 668–677
18. Kleinrock, L.: Queueing Systems Vol. 2: Compurer Applications. Wiley (1976)
19. Parekh, A., Gallager, R.: A generalized processor sharing approach to flow control in integrated services networks: The single-node case. IEEE/ACM Transactions on Networking **1** (1993) 344–357
20. Dürr, C., Thang, N.K.: Non-clairvoyant scheduling games. In: Proc. 2nd Intl. Symp. Algorithmic Game Theory (SAGT). (2009)
21. Bruno, J., E. G. Coffman, J., Sethi, R.: Scheduling independent tasks to reduce mean finishing-time (extended abstract). SIGOPS Oper. Syst. Rev. **7** (1973) 102–103