

How to find Nash equilibria with extreme total latency in network congestion games?

Heike Sperber*

University of Kaiserslautern, Department of Mathematics, P.O.Box 3049, 67653 Kaiserslautern, Germany.

sperber@mathematik.uni-kl.de

Abstract. We study the complexity of finding extreme pure Nash equilibria in symmetric network congestion games and analyse how it depends on the graph topology and the number of users. In our context best and worst equilibria are those with minimum respectively maximum total latency.

We establish that both problems can be solved by a Greedy algorithm with a suitable tie breaking rule on parallel links. On series-parallel graphs finding a worst Nash equilibrium is NP-hard for two or more users while finding a best one is solvable in polynomial time for two users and NP-hard for three or more. Additionally we establish NP-hardness in the strong sense for the problem of finding a worst Nash equilibrium on a general acyclic graph.

Keywords: network congestion game, total latency, extreme equilibria, complexity

1 Introduction

Nash equilibria are one of the most common concepts in non-cooperative game theory. The classic questions concerning these stable states of a game, in which no selfish user is unsatisfied and wants to change to a different strategy, are those of existence and uniqueness.

Modern algorithmic game theory brings up additional questions such as computability of equilibria and the overall performance of the system under selfish behaviour. Pigou [13] gave a first negative answer by stating that in general selfish non-cooperative behaviour does not lead to social optimal outcome. Papadimitriou [12] introduced the coordination ratio as the quotient of the social cost of a worst Nash equilibrium and the minimal social cost. It is often called “Price of Anarchy” as it reflects the degradation in performance due to missing central regulation.

Koutsoupias and Papadimitriou [10] established a model (KP-Model) in which users of different size travel on parallel links with linear latency functions analogously to uniform/related machines in scheduling. The price of anarchy of this game and various similar models was studied extensively [2, 11, 5, 6].

The problems of finding extreme (best and worst) Nash equilibria concerning makespan social cost for this model was studied by Fotakis et al. [7], who established them to be NP-hard in the strong sense.

These hardness proofs rely on the different sizes of users and the corresponding scheduling and bin-packing problems are easy to solve for unit-sized users. Additionally Epstein et al. [3] show all Nash equilibria for unit-sized users on parallel links to have the same makespan.

Unit-sized users traveling through more complex graphs are modeled by network congestion games. Rosenthal [14] established the existence of pure Nash equilibria in these games. Fabrikant et al. [4] gave a polynomial time algorithm to compute an

* Supported by the Rhineland-Palatinate Cluster of Excellence Dependable Adaptive Systems and Mathematical Modeling.

arbitrary Nash equilibrium for a symmetric (single-commodity) network congestion game by the use of a certain min-cost flow instance. On series-parallel graphs this min-cost flow instance can be solved by the Greedy algorithm GBR of Fotakis et al. [8].

Recently Gassner et al. [9] analysed extreme Nash equilibria in network congestion games for makespan social cost, measuring the latency of the longest path chosen by a user, and showed that finding a worst equilibrium is “easier” in the sense that a worst equilibrium can be found in polynomial time on series-parallel networks while establishing a best one is NP-hard on this topology.

In contrast Awerbuch et al. [1] used the total latency as measure of social cost to establish bounds on the price of anarchy for network congestion games.

Contribution We give a complete characterization of the complexity of finding Nash equilibria with minimum or maximum *total latency* in network congestion games with non-decreasing latency functions on edges.

On parallel links both problems can be solved by a Greedy algorithm with tie breaking according to the increase in cost. But the problem of finding a best Nash equilibrium is slightly harder as this approach fails for non-increasing latencies.

The situation is more involved for series-parallel graphs: Unfortunately the problem of finding a worst Nash equilibrium is NP-hard even for two users. Here finding the best equilibrium is somehow easier, as we can find a best equilibrium for two users in polynomial time but the problem is NP-hard for three or more users. Our reductions so far imply only NP-hardness in the weak sense. For finding a worst Nash equilibrium we additionally establish NP-hardness in the strong sense on general acyclic networks.

These results are summarized in the following chart:

<i>Find a ... on ...</i>	Nash equilibrium with minimum total latency	Nash equilibrium with maximum total latency
parallel links	polynomially solvable for increasing latencies by Greedy ^{min}	polynomially solvable for non-decreasing latencies by Greedy ^{max}
series-parallel graphs	polynomially solvable for two users by Greedy ^{min} but NP-hard for three or more	NP-hard for two or more users (NP-hard in the strong sense on general acyclic graphs)

Road Map We start by introducing notation and preliminary results in Section 2 and then establish our results on finding a best Nash equilibrium in Section 3 and in Section 4 for a worst Nash equilibrium, respectively.

2 Preliminaries

We consider a symmetric network congestion game, namely N unit-sized users each choosing a path from the source s to the sink t in the directed graph $G = (V, E)$. The strategy set \mathcal{P} of all users are thus all simple s - t -paths in G . We denote by n the number of vertices and m the number of edges of G . The edges are equipped with *non-decreasing latency functions* $\ell_e : \mathbb{N}_0 \rightarrow \mathbb{R}_+^0$ for all $e \in E$ modeling the congestion effects. An instance of the game is thus given by $[G = (V, E), (\ell_e)_{e \in E}, s \in V, t \in V, N]$.

In our context a *flow* is a function $f : \mathcal{P} \rightarrow \mathbb{N}_0$ that assigns integer values to paths in the network. The latency on a path is the sum of the latencies on its edges that depends on the total flow on the edge:

$$\ell_P(f) := \sum_{e \in P} \ell_e \left(\sum_{P' \in \mathcal{P}: e \in P'} f_{P'} \right)$$

We denote by $f_e := \sum_{P \in \mathcal{P}: e \in P} f_P$ the flow on edge e uniquely induced by the flow f defined on paths.

Note that there may be different so-called flow-decompositions or flows on paths that correspond to the same flow on edges. Even more, Gassner et al. [9] give an example that the property of a feasible flow to be a Nash equilibrium might depend on the flow decomposition.

A stable state of the system is a choice of paths such that no user can benefit by deviating from her choice given those of the other users:

Definition 1 (Nash Equilibrium, Nash Flow). *A flow $f = (f_P)_{P \in \mathcal{P}}$ is at Nash equilibrium, if and only if for all paths P_1, P_2 with $f_{P_1} > 0$ we have*

$$\ell_{P_1}(f) \leq \ell_{P_2}(\tilde{f}) \text{ with } \tilde{f}_P = \begin{cases} f_P - 1 & \text{if } P = P_1 \\ f_P + 1 & \text{if } P = P_2 \\ f_P & \text{otherwise} \end{cases}.$$

Rosenthal [14] established that every instance of a network congestion game possesses a least one pure strategy Nash equilibrium. We want to analyse Nash equilibria with respect to an additional measure of quality:

Definition 2 (Total latency social cost). *The total latency $C(f)$ of a flow f in a network $G = (V, E)$ with edge latency functions ℓ_e is defined as*

$$C(f) = \sum_{e \in E} \ell_e(f_e) f_e$$

We denote the highest latency experienced by a user as the makespan $C_{\max}(f)$ of a flow f :

$$C_{\max}(f) := \max_{P \in \mathcal{P}: f_P > 0} \ell_P(f)$$

Example 1 (Nash equilibria with different social cost). Consider the graph G in Figure 1 consisting of just two nodes s, t and two parallel edges e_1, e_2 between them with latency $\ell_{e_1}(x) = x$ and $\ell_{e_2}(x) = \frac{1}{2}x$. We want to send two users from s to t .

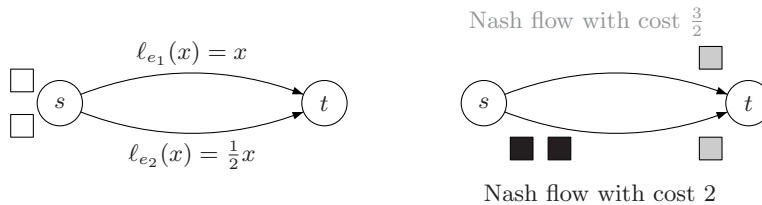


Fig. 1. Two Nash equilibria on parallel links need not have the same total latency. (Example 1).

In this setting there are two Nash equilibria: One sending all flow on edge e_2 for costs of 2 and the second one sending one user on each edge e_1 and e_2 resulting in lower costs of $\frac{3}{2}$.

Even more, in general no Nash equilibrium is optimal concerning total latency even on parallel links:

Example 2 (No optimal Nash equilibrium). Consider the graph G in Figure 2 with two parallel links and latencies $\ell_{e_1}(x) = x$, $\ell_{e_2}(x) = 2x + \epsilon$ for $0 < \epsilon < \frac{1}{2}$. We send two users from s to t .

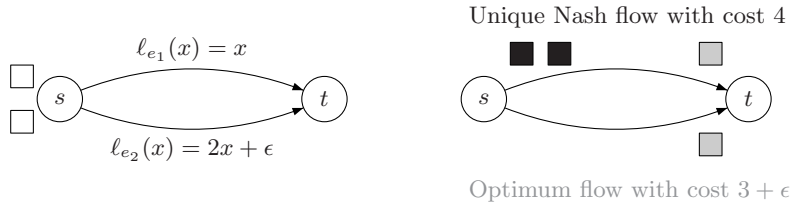


Fig. 2. Nash equilibria on parallel links need not be optimal concerning total latency. (Example 2).

The unique Nash equilibrium sends both users on edge e_1 for costs of 4, while the optimal solution splits the flow resulting in cost of $3 + \epsilon$.

Examples 1 and 2 motivate to study the following two problems:

Definition 3 (Best Nash Equilibrium Problem (BNash)).

Given: Network congestion game $[G = (V, E), (\ell_e)_{e \in E}, s \in V, t \in V, N]$

Output: Nash equilibrium f with minimum total latency

Definition 4 (Worst Nash Equilibrium Problem (WNash)).

Given: Network congestion game $[G = (V, E), (\ell_e)_{e \in E}, s \in V, t \in V, N]$

Output: Nash equilibrium f with maximum total latency

We are going to examine the dependence of the complexity of both problems on the topology of the underlying network. Thereby, we look at parallel links, arbitrary (acyclic) graphs and series-parallel graphs. The latter are defined inductively as follows:

Definition 5 (Series-parallel graph). A single edge $e = (s, t)$ is series-parallel with start terminal s and end-terminal t by definition. Let G_i be series-parallel with start-terminal s_i and end-terminal t_i ($i = 1, 2$). Then the graph $S(G_1, G_2)$ obtained by identifying t_1 as s_2 is a series-parallel graph, with s_1 and t_2 as its terminals (series composition). The graph $P(G_1, G_2)$ obtained by identifying s_1 as s_2 and also t_1 as t_2 is a series-parallel graph (parallel composition) with $s_1 (= s_2)$ and $t_1 (= t_2)$ as its terminals.

For our positive results we modify the algorithm GBR introduced by Fotakis et al. [8] which works as follows: The users are iteratively assigned to a path minimizing the latency induced by the users already assigned. To be more precise denote by f_i the result of GBR in the i th iteration, f_0 the constant zero flow on all edges and

$$L^+(f_i) := \min_{P \in \mathcal{P}} \sum_{e \in P} \ell_e(f_{i,e} + 1) \quad (1)$$

the minimum latency for a new $(i + 1)$ st user. Thus GBR chooses a path P_{i+1} of user $(i + 1)$ such that the latency on P_{i+1} is $L^+(f_i)$ after the assignment. Fotakis et al. [8] establish that this algorithm yields a Nash equilibrium on series-parallel graphs.

This path P_{i+1} is in general not uniquely determined by (1) but there is a set $\mathcal{P}^+(f_i)$ of paths with minimal latency for an additional $(i+1)$ st user. We add tie breaking rules to chose a specific path from this set:

Definition 6 (Greedy^{min} and Greedy^{max}). *In the following we denote by Greedy^{min} the algorithm GBR that chooses among candidate paths $\mathcal{P}^+(f_i)$ one with minimal cost increase:*

$$P_{i+1} := \operatorname{argmin}_{P \in \mathcal{P}^+(f_i)} \sum_{e \in P} \Delta c_e(f_{i,e} + 1)$$

The cost increase $\Delta c_e : \mathbb{N} \rightarrow \mathbb{R}_+^0$ for all edges $e \in E$ is defined as

$$\Delta c_e(1) = \ell_e(1) \text{ and } \Delta c_e(n) = n \cdot \ell_e(n) - (n-1) \cdot \ell_e(n-1) \text{ for all } n \geq 2.$$

Greedy^{max} denotes the analog algorithm which chooses a candidate path with maximal cost increase.

Observe that the running time of Greedy^{min} [Greedy^{max}] is polynomial in the input size of the network congestion game on a series-parallel graph as in the $(i+1)$ th iteration we just have to find a lexicographic shortest s - t -path for the fixed edge labels $(\ell_e(f_{i,e} + 1), \Delta c_e(f_{i,e} + 1))$ $[(\ell_e(f_{i,e} + 1), -\Delta c_e(f_{i,e} + 1))]$, which can be done in linear time on these acyclic graphs with the help of a topological sorting of the vertices.

3 Best Nash Equilibrium

3.1 Parallel links

On parallel links all local optima of Rosenthal's potential function are global optimum and thus all Nash flows can be found by the min-cost flow instance introduced by Fabrikant et al. [4]. In particular on this very easy topology every Nash flow can be found by GBR. Which Nash equilibrium is found depends on the tie breaking rule applied.

We start with the case of increasing functions and establish that Greedy^{min} really finds a best Nash equilibrium.

Theorem 1. *Greedy^{min} solves BNash for increasing latency functions on parallel links.*

Proof. We use induction on the number of users to show that Greedy^{min} succeeds in finding a best Nash equilibrium. The base case is easy, as sending one user on any candidate edge will induce the same cost.

Let f_{i+1} be the result of Greedy^{min} after $(i+1)$ iterations and e_{i+1} the edge chosen in the $(i+1)$ st iteration. Let g_{i+1} denote an arbitrary Nash flow in the same graph with also $(i+1)$ users which is not equal to f_{i+1} . (If there is no such flow f_{i+1} is best anyway.) Denote by $\bar{E} := \{e \in E : \ell_e(g_{i+1,e}) = C_{\max}(g_{i+1})\}$ the set of critical edges of g_{i+1} . Remember that the candidate set of edges for iteration $(i+1)$ is

$$E_i^+ := \{e_{i+1} \in E : \ell_{e_{i+1}}(f_{i,e_{i+1}} + 1) = \min_{e \in E} \ell_e(f_{i,e} + 1)\}.$$

In a first step, we show that there exists an edge \bar{e} with $\bar{e} \in \bar{E} \cap E_i^+$.

For the sake of a contradiction assume that for all edges $\bar{e} \in \bar{E}$ we have $\bar{e} \notin E_i^+$ and let \bar{e} be an arbitrary such edge. Thus we know $\ell_{\bar{e}}(g_{i+1,\bar{e}}) = C_{\max}(g_{i+1}) = C_{\max}(f_{i+1})$

and $\ell_{\bar{e}}(f_{i,\bar{e}} + 1) > C_{\max}(f_{i+1})$. Monotonicity of the latency functions thus implies $f_{i,\bar{e}} + 1 > g_{i+1,\bar{e}}$. By integrality of the flow values we conclude

$$f_{i+1,e} \geq f_{i,\bar{e}} \geq g_{i+1,\bar{e}} \text{ for all } e \in \bar{E}. \quad (2)$$

Let $e \in E \setminus \bar{E}$ be an arbitrary edge that is not critical for g_{i+1} and thus $\ell_e(g_{i+1,e}) < C_{\max}(g_{i+1})$. g_{i+1} being Nash additionally gives us $\ell_e(g_{i+1,e} + 1) \geq C_{\max}(g_{i+1}) = C_{\max}(f_{i+1}) \geq \ell_e(f_{i+1,e})$. As ℓ_e is increasing and both flows g_{i+1}, f_{i+1} are Nash we can conclude either if $\ell_e(f_{i+1,e}) = C_{\max}(f_{i+1})$ then $g_{i+1,e} + 1 = f_{i+1,e}$ or if $\ell_e(f_{i+1,e}) < C_{\max}(f_{i+1})$ then $f_{i+1,e} = g_{i+1,e}$. Together this implies that

$$f_{i+1,e} \geq g_{i+1,e} \text{ for all } e \in E \setminus \bar{E}. \quad (3)$$

Equations (2) and (3) tells us that either $g_{i+1} = f_{i+1}$ or f_{i+1} sends more users from s to t . This contradicts the choice of g_{i+1} and thus completes the proof of the first step.

In the second step, we are now ready to compare the cost of f_{i+1} and g_{i+1} : We have just established that there exists $\bar{e} \in \bar{E} \cap E_i^+$. Hence, we know that $\ell_{\bar{e}}(g_{i+1,\bar{e}}) = C_{\max}(g_{i+1}) = C_{\max}(f_{i+1})$ and $\ell_{\bar{e}}(f_{i,\bar{e}} + 1) = C_{\max}(f_{i+1})$ and thus by $\ell_{\bar{e}}$ being increasing $g_{i+1,\bar{e}} = f_{i,\bar{e}} + 1$. We decompose g_{i+1} into one unit of flow on edge \bar{e} and a Nash flow g_i with i users. Hence we can compare the total latencies:

$$\begin{aligned} C(g_{i+1}) - C(f_{i+1}) &= C(g_i) - C(f_i) + \Delta c_{\bar{e}}(g_{i+1,\bar{e}}) - \Delta c_{e_{i+1}}(f_{i+1,e_{i+1}}) \\ &\geq \Delta c_{\bar{e}}(g_{i+1,\bar{e}}) - \Delta c_{e_{i+1}}(f_{i+1,e_{i+1}}) \\ &= \Delta c_{\bar{e}}(f_{i,\bar{e}} + 1) - \Delta c_{e_{i+1}}(f_{i,e_{i+1}} + 1) \\ &\geq 0 \end{aligned}$$

The first inequality follows from the induction hypothesis and the second one from the choice of edge e_{i+1} by Greedy^{min}. \square

The following example establishes that in general Greedy^{min} does not work for non-decreasing latencies:

Example 3 (Greedy^{min} fails for non-decreasing latency functions). Consider the graph G in Figure 3 with 5 parallel edges between s and t . There is one edge e_1 with latency $\ell_{e_1}(x) = \begin{cases} 3 & x \leq 1 \\ 6 & x > 1 \end{cases}$ and the remaining four edges have latency $\ell_{e_i}(x) = 2x + 2$, $i = 2, \dots, 5$. We want to send 9 users from s to t .

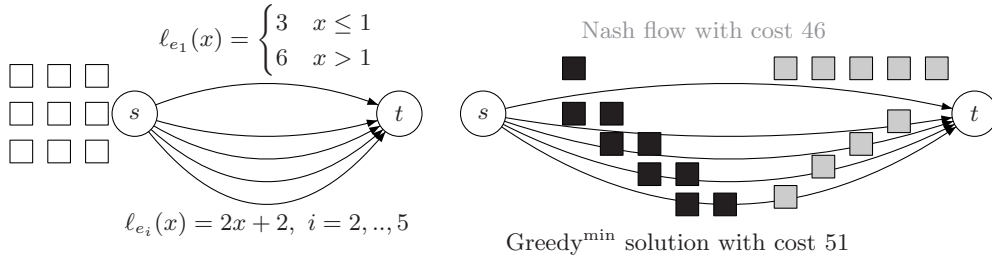


Fig. 3. Greedy^{min} fails for non-decreasing latencies on parallel links. (Example 3).

Greedy^{min} assigns the first user to edge e_1 , then in some order adds one user to each of the other edges. In the sixth iteration all edges are candidate edges and the algorithm compares the cost increase:

$$\Delta c_{e_1}(2) = 9 > 8 = \Delta c_{e_i}(2) \quad i = 2, \dots, 5$$

Thus Greedy^{min} assigns the remaining four users one to each of the edges e_2, \dots, e_5 which results in total latency 51. But assigning only one job to every edge e_2, \dots, e_5 and five users to e_1 is also Nash and has costs of 46.

3.2 Series-parallel networks

In series-parallel graphs the best Nash equilibrium is not guaranteed to be found by GBR (with any tie breaking rule) for three or more users, even for increasing latency functions.

Example 4 (Best Nash flow not found by GBR). Consider the graph shown in Figure 4 for three users.

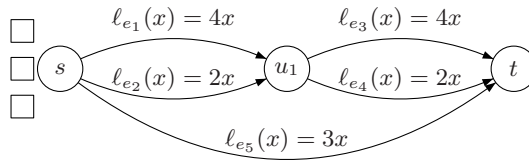


Fig. 4. Best Nash equilibrium not found by GBR. (Example 4).

The solution of GBR for the graph given in Figure 4 is unique and has a unique path decomposition sending one user on the lower edges e_2 and e_4 and the other two on the direct edge e_5 with costs $C(f^*) = 16$. However, there is a Nash equilibrium f with cost $C(f) = 15$ which sends one user on the direct edge e_5 and both other users on one upper e_1 (e_3) and one lower edge e_4 (e_2) such that all edges are used by exactly one user.

The good news is that for two users we can use Greedy^{min} to find a best Nash equilibrium on series-parallel graphs in polynomial time.

Theorem 2. *Greedy^{min} solves BNash for two users on series-parallel graphs.*

Proof. We use induction on the number of composition steps necessary to construct G . If G is a single edge every Nash equilibrium send both users over this one edge and thus Greedy^{min} finds a best one.

For the induction step let f be the result of Greedy^{min} which sends the users on paths P_1, P_2 and g an arbitrary equilibrium in G which uses paths Q_1, Q_2 . We analyse the cost of f in comparison to that of g in several cases depending on the allocation of the four paths in G_1 and G_2 .

1. $G = S(G_1, G_2)$: As Greedy^{min} on G_i chooses exactly the parts of P_1, P_2 in G_i we know by induction hypothesis that $f|_{G_i}$ is a best equilibrium on G_i , $i = 1, 2$. As g also decomposes into two Nash equilibria in G_1, G_2 and the cost of the flow in G is just the sum of the costs of the flows in G_1, G_2 we conclude that $C(f) \leq C(g)$.
2. $G = P(G_1, G_2)$:
 - 2.1. $P_1 \in G_1$ and $P_2 \in G_2$:
 - 2.1.1. $Q_1 \in G_1$ and $Q_2 \in G_2$: To be Nash both f and g can only use shortest paths w.r.t. $\ell_e(1)$ in both components and the costs are just the sum of the latency of these paths. Hence $C(f) = C(g)$.
 - 2.1.2. W.l.o.g. $Q_1, Q_2 \in G_1$: Let f_1 be the result of Greedy^{min} on G_1 . f_1 chooses paths P_1 and \bar{P}_2 in G_1 . Greedy^{min} on G did not choose \bar{P}_2 instead of P_2 thus we have one of the following two cases:

2.1.2.1. $\ell_{\bar{P}_2}(f_1) > \ell_{P_2}(f)$: We can conclude from the construction of f_1 and the induction hypothesis for Greedy^{min} on G_1 that

$$C(f) = \ell_{P_1}(f) + \ell_{P_2}(f) < \ell_{P_1}(f_1) + \ell_{\bar{P}_2}(f_1) = C(f_1) \leq C(g)$$

2.1.2.2. $\ell_{\bar{P}_2}(f_1) = \ell_{P_2}(f)$ and $\sum_{e \in \bar{P}_2} \Delta c_e(f_{1,e}) \geq \sum_{e \in P_2} \Delta c_e(f_e)$: Again by applying the induction hypothesis we can compare the cost of f and g :

$$\begin{aligned} C(f) &= \sum_{e \in P_1} \ell_e(1) + \sum_{e \in P_2} \ell_e(1) \\ &= \sum_{e \in P_1} \ell_e(1) + \sum_{e \in P_2} \Delta c_e(f_e) \\ &\leq \sum_{e \in P_1} \ell_e(1) + \sum_{e \in \bar{P}_2} \Delta c_e(f_{1,e}) \\ &= C(f_1) \\ &\leq C(g) \end{aligned}$$

2.2. W.l.o.g. $P_1, P_2 \in G_1$:

2.2.1. $Q_1, Q_2 \in G_1$: As we can restrict both flows to G_1 we know by induction hypothesis that $C(f) \leq C(g)$.

2.2.2. $Q_1, Q_2 \in G_2$: Due to the Nash property of g , no user wants to change to a path in G_1 , and in f no user wants to change to G_2 . Together with monotonicity of the latency functions this implies

$$\ell_{Q_j}(g) \leq \ell_{P_i}(f) \text{ and } \ell_{P_i}(f) \leq \ell_{Q_j}(g) \text{ for all } i, j \in \{1, 2\}$$

Hence $\ell_{P_1}(f) = \ell_{P_2}(f) = \ell_{Q_1}(g) = \ell_{Q_2}(g)$ and so $C(f) = C(g)$.

2.2.3. $Q_1 \in G_1$ and $Q_2 \in G_2$: Greedy^{min} on G did not choose Q_2 instead of P_2 so we have exactly one of the following cases:

2.2.3.1. $\ell_{Q_2}(g) > \ell_{P_2}(f)$: In the following we show that this case contradicts g being Nash. For this purpose we construct a path \bar{Q}_2 in G_1 and show that the user on path Q_2 would want to change to \bar{Q}_2 .

Let $v_1 = s, v_2, \dots, v_k = t$ be the vertices in P_2 and $v_{i_1} = s, v_{i_2}, \dots, v_{i_l} = t$ those that also lie on P_1 . Let $\pi_{j,1}, \pi_{j,2}$ be the parts between v_{i_j} and $v_{i_{j+1}}$ of P_1, P_2 , respectively for $j = 1, \dots, l-1$. Let \bar{Q}_2 be the path through vertices v_{i_1}, \dots, v_{i_l} using path segments $\bar{\pi}_j, j = 1, \dots, l-1$ where $\bar{\pi}_j = \pi_{1,j}$ if there is an edge $e \in \pi_{j,2}$ with $e \in (P_2 \cap Q_1) \setminus P_1$ and $\bar{\pi}_j = \pi_{2,j}$ otherwise. Let g_1 be the flow that uses paths Q_1, \bar{Q}_2 .

Let $j \in \{1, \dots, l-1\}$ be arbitrary. If $\bar{\pi}_j = \pi_{1,j}$ we know that Q_1 does not intersect $\pi_{1,j}$ as it shares an edge with $\pi_{j,2}$ and G is series-parallel. We additionally use that P_1 is a shortest s - t -path w.r.t. $\ell_e(1)$ and conclude

$$\sum_{e \in \bar{\pi}_j} \ell_e(g_{1,e}) = \sum_{e \in \pi_{j,1}} \ell_e(1) \leq \sum_{e \in \pi_{j,2}} \ell_e(1) \leq \sum_{e \in \pi_{j,2}} \ell_e(f_e)$$

If $\bar{\pi}_j = \pi_{2,j}$ we have

$$\begin{aligned} \sum_{e \in \bar{\pi}_j} \ell_e(g_{1,e}) &= \sum_{e \in \pi_{j,2}} \ell_e(g_{1,e}) \\ &= \sum_{e \in \pi_{j,2} \setminus Q_1} \underbrace{\ell_e(1)}_{\leq f_e} + \sum_{e \in \pi_{j,2} \cap Q_1 \cap P_1} \underbrace{\ell_e(2)}_{= f_e} \\ &\leq \sum_{e \in \bar{\pi}_{j,2}} \ell_e(f_e) \end{aligned}$$

By summing over j we conclude $\ell_{\bar{Q}_2}(g_1) \leq \ell_{P_2}(f) < \ell_{Q_2}(g)$ which contradicts g being Nash.

2.2.3.2. $\ell_{Q_2}(g) = \ell_{P_2}(f)$ and $\sum_{e \in \bar{Q}_2} \Delta c_e(g_e) \geq \sum_{e \in P_2} \Delta c_e(f_e)$: Using that P_1 and Q_1 are both shortest paths w.r.t. $\ell_e(1)$ we have

$$C(f) = \sum_{e \in P_1} \ell_e(1) + \sum_{e \in P_2} \Delta c_e(f_e) \leq \sum_{e \in Q_1} \ell_e(1) + \sum_{e \in Q_2} \Delta c_e(g_e) = C(g)$$

□

In contrast to this positive result we can use a construction similar to Example 4 to establish (weak) NP-hardness of the problem to find a best Nash equilibrium for three or more users.

Theorem 3. *The problem BNash is NP-hard on series-parallel graphs for three or more users.*

Proof. The proof of weak NP-completeness of the decision version of BNash uses a reduction from the Even-Odd Partition Problem:

EVEN-ODD PARTITION (EOP for short):

Given: Finite set $A = \{a_1, a_2, \dots, a_{2n}\}$, a size $w(a_i) \in \mathbb{Z}^+$ for each $a_i \in A$ and $2B = \sum_{i=1}^{2n} w(a_i)$.

Question: Does there exist a subset $A' \subset A$ with $\sum_{a \in A'} w(a) = B$ and A' contains exactly one element of $\{a_{2i-1}, a_{2i}\}$ for $i = 1, \dots, n$.

We may assume without loss of generality that

$$w(a_{2i-1}) < 2w(a_{2i}) \quad \text{and} \quad w(a_{2i}) < 2w(a_{2i-1}) \quad \text{holds for } i = 1, \dots, n. \quad (4)$$

Given an instance $I(\text{EOP})$ then an instance $I(\text{BNash})$ is defined by a graph $G = (V, E)$ with $V = \{v_1, v_2, \dots, v_n, v_{n+1}\}$ with two parallel edges between v_i and v_{i+1} for $i = 1, \dots, n$ and an edge $e^+ = (s, t)$. The latency functions of the two edges between v_i and v_{i+1} are $\ell(x) = w(a_{2i-1})x$ and $\ell(x) = w(a_{2i})x$ for $i = 1, \dots, n$ and $\ell_{e^+}(x) = \frac{1}{2}Bx$. Three users travel from $s = v_1$ to $t = v_{n+1}$. We denote $G \setminus \{e^+\}$ by G' .

We show that $I(\text{EOP})$ is a YES-instance if and only if there exists a Nash equilibrium f in G of $I(\text{BNash})$ with $C(f) \leq \frac{5}{2}B$.

Let $I(\text{EOP})$ be a YES-instance and A' its solution. Construct a flow f in G by sending one user along the edges with slope $w(a')$ for $a' \in A'$, the second user on the remaining edges of G' and the third user on edge e^+ .

The cost of f can be calculated as follows

$$C(f) = \sum_{e \in E} \ell_e(f_e) \cdot f_e = \sum_{e \in E} \ell_e(1) = \sum_{a \in A} w(a) + \frac{1}{2}B = 2B + \frac{1}{2}B = \frac{5}{2}B$$

Additionally f is a Nash flow: Both users in G' can not benefit from changing their path in G' due to (4) and as both chosen paths have latency B they do not want to change to edge e^+ either. The user on edge e^+ has also no incentive to change to any path in G' .

On the contrary assume f to be a Nash flow with cost $C(f) \leq \frac{5}{2}B$.

Case 1: $f_{e^+} = 3$

Let P be the path in G' consisting of the edges with latency $\min\{w(a_{2i-1}), w(a_{2i})\}$.

The latency on path P for one user would be less or equal to B due to the choice of edges and thus at least one user wants to change from e^+ to P as $\ell_{e^+}(f) = \ell_{e^+}(3) = \frac{3}{2}B$. Hence f can't be Nash in this case.

Case 2: $f_{e^+} = 2$

Let P_1 be the path chosen by the third user and P as in Case 1. We know about the cost of f :

$$C(f) = 2 \cdot \ell_{e^+}(2) + \ell_{P_1}(f) \geq 2 \cdot B + \sum_{e \in P} \ell_e(1) \quad (5)$$

Equation (4) implies

$$\sum_{e \in P} \ell_e(1) = \sum_{i=1}^n \min\{w(a_{2i-1}), w(a_{2i})\} > \sum_{i=1}^n \frac{1}{2} \max\{w(a_{2i-1}), w(a_{2i})\} \geq \frac{1}{2}B \quad (6)$$

Using Equations (5) and (6) we derive a contradiction to the assumption about the cost of f :

$$C(f) \geq 2 \cdot B + \sum_{e \in P} \ell_e(1) > \frac{5}{2}B$$

Case 3: $f_{e^+} = 1$

Denote by P_1 and P_2 those paths in G' chosen by two of the users. From Equation (4) we know that $P_1 \cap P_2 = \emptyset$ and hence

$$\ell_{P_1}(f) + \ell_{P_2}(f) = 2B \quad (7)$$

In a Nash equilibrium no user wants to change from P_i , $i = 1, 2$ to edge e^+ :

$$B = \ell_{e^+}(f_{e^+} + 1) \geq \ell_{P_i}(f) \text{ for } i = 1, 2 \quad (8)$$

Combining Equations (7) and (8) we conclude that $\ell_{P_1}(f) = B = \ell_{P_2}(f)$ and hence these disjoint paths give rise to an even-odd partition.

Case 4: $f_{e^+} = 0$

This case can not constitute a Nash equilibrium, as at least one user wants to change to edge e^+ . \square

4 Worst Nash Equilibrium

4.1 Parallel links

Analogously to the case of finding the best equilibrium in Theorem 1 we can establish a result for finding a worst Nash equilibrium for increasing latency functions by Greedy^{max}. But for this case we can get even more as stated in the following theorem:

Theorem 4. *Greedy^{max} solves WNash on parallel links for non-decreasing latency functions.*

We prepare the proof of Theorem 4 by a lemma. We adopt the notation of the Proof of Theorem 1 and add the set $M^- := \{e \in E \mid f_{i+1,e} < g_{i+1,e}\}$ to compare the flow on edges.

Lemma 1. *Let f_{i+1} be the result of GBR (with any tie-breaking rule) on parallel links. For every other Nash flow $g_{i+1} \neq f_{i+1}$ there exists an edge $\bar{e} \in \bar{E}$ with $\bar{e} \in M^- \cap E_i^+$.*

Proof. All edges $e \in M^-$ have latency $\ell_e(g_{i+1,e}) = C_{\max}(g_{i+1}) = C_{\max}(f_{i+1})$ and thus are elements of \bar{E} . Additionally we have $g_{i+1,e} > f_{i+1,e}$. If $e \neq e_{i+1}$ and thus $f_{i+1,e} = f_{i,e}$ this implies that $e \in E_i^+$. For $e = e_{i+1}$ we use $e_{i+1} \in E_i^+$. As $g_{i+1} \neq f_{i+1}$ we have $M^- \neq \emptyset$ and thus there must be an edge with the desired properties. \square

Proof (of Theorem 4). We analyse the costs by induction on the number of users assigned by Greedy^{max}. The base case is easy as any choice of a minimum latency edge induces the same cost for the first user.

Let f_{i+1} be the result of Greedy^{max} after $(i+1)$ iterations and e_{i+1} the edge chosen in the $(i+1)$ st iteration. Denote by g_{i+1} an arbitrary Nash flow in the same graph with also $(i+1)$ users which is not equal to f_{i+1} . (If no such g_{i+1} exists there is nothing left to prove.)

Using Lemma 1 we can decompose g_{i+1} in a way that the $(i+1)$ st user was sent on an edge $\bar{e} \in (\bar{E} \cap M^- \cap E_i^+)$. As both $e_{i+1}, \bar{e} \in E_i^+$ the tie breaking rule of Greedy^{max} implies that

$$\Delta c_{e_{i+1}}(f_{i+1,e_{i+1}}) = \Delta c_{e_{i+1}}(f_{i,e_{i+1}} + 1) \geq \Delta c_{\bar{e}}(f_{i,\bar{e}} + 1) \quad (9)$$

Hence we can compare the costs of g_{i+1} and f_{i+1} :

$$\begin{aligned} C(g_{i+1}) - C(f_{i+1}) &= \underbrace{C(g_i) - C(f_i)}_{\leq 0} + \Delta c_{\bar{e}}(g_{i+1,\bar{e}}) - \Delta c_{e_{i+1}}(f_{i+1,e_{i+1}}) \\ &\leq \Delta c_{\bar{e}}(g_{i+1,\bar{e}}) - \Delta c_{e_{i+1}}(f_{i+1,e_{i+1}}) \\ &= \Delta c_{\bar{e}}(g_{i,\bar{e}} + 1) - \Delta c_{e_{i+1}}(f_{i,e_{i+1}} + 1) \\ &\stackrel{(9)}{\leq} \Delta c_{\bar{e}}(g_{i,\bar{e}} + 1) - \Delta c_{\bar{e}}(f_{i,\bar{e}} + 1) \end{aligned} \quad (10)$$

As $\bar{e} \in M^-$ we know that $g_{i+1,\bar{e}} - 1 = g_{i,\bar{e}} \geq f_{i+1,\bar{e}}$. If $\bar{e} \neq e_{i+1}$ we conclude $g_{i,\bar{e}} \geq f_{i,\bar{e}}$ and if \bar{e} and e_{i+1} coincide even strict inequality holds.

We distinguish two cases:

Case 1: $g_{i,\bar{e}} = f_{i,\bar{e}}$ (and $e_{i+1} \neq \bar{e}$)

$$\Delta c_{\bar{e}}(g_{i,\bar{e}} + 1) - \Delta c_{\bar{e}}(f_{i,\bar{e}} + 1) = 0 \quad (11)$$

Case 2: $g_{i,\bar{e}} > f_{i,\bar{e}}$

This implies that $\ell_{\bar{e}}(g_{i,\bar{e}}) = C_{\max}(g_{i+1}) = C_{\max}(f_{i+1})$ and

$$\begin{aligned} \Delta c_{\bar{e}}(g_{i+1,\bar{e}}) - \Delta c_{\bar{e}}(f_{i,\bar{e}} + 1) &= C_{\max}(f_{i+1}) \cdot (g_{i+1,\bar{e}} - g_{i,\bar{e}} - f_{i,\bar{e}} - 1) + f_{i,\bar{e}} \cdot \ell_{\bar{e}}(f_{i,\bar{e}}) \\ &= f_{i,\bar{e}} \cdot \underbrace{(\ell_{\bar{e}}(f_{i,\bar{e}}) - C_{\max}(f_{i+1}))}_{\leq 0} \\ &\leq 0 \end{aligned} \quad (12)$$

Thus (11),(12) together imply that the difference in cost described in (10) is negative. This proves that f_{i+1} is “worse” than g_{i+1} . \square

4.2 Series-parallel networks

First of all we observe that GBR does in general not find a worst Nash equilibrium on series-parallel networks independent of the tie breaking rule applied:

Example 5 (Worst Nash flow not found by GBR). Consider the series parallel graph shown in Figure 5 in which we want to send two users from s to t .

GBR sends one user on path $P_1 = \{e_1, e_3\}$ and the other one on $P_2 = \{e_5\}$. The resulting Nash flow f has makespan $C_{\max}(f) = 6$ and cost $C(f) = 10$.

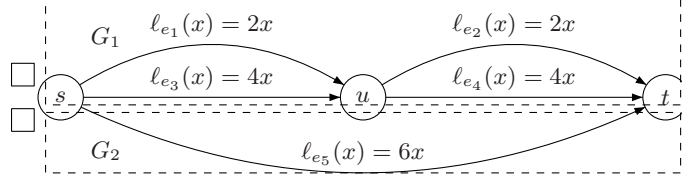


Fig. 5. Worst Nash equilibrium not found by GBR (Example 5).

We compare this to the Nash flow g with $g_{Q_1} = g_{Q_2} = 1$ with $Q_1 = \{e_1, e_4\}$ and $Q_2 = \{e_2, e_3\}$. This flow has also makespan $C_{\max}(g) = 6$ but cost $C(g) = 12$. Thus GBR does not find the worst equilibrium.

Observe that $g_1 := g|_{G_1}$ is not the worst Nash flow on G_1 for two users (neither for makespan nor cost), as sending both users on path P_1 induces costs of 16 but also a makespan of 8 and thus does not constitute a Nash equilibrium any more when G_1 and G_2 are composed in parallel. This tells us that a worst Nash flow in a parallel composed graph does not necessarily consist of worst Nash flows in the two components.

In the following we want to show that it is hard to find the worst Nash equilibrium in series-parallel graphs.

Theorem 5. *The problem WNash is NP-hard on series-parallel graphs even for two users.*

Proof. The proof of (weakly) NP-completeness of the decision version of WNash uses a reduction from the Even-Odd Partition Problem as introduced in the Proof of Theorem 3.

Given an instance $I(\text{EOP})$ then an instance $I(\text{WNash})$ is defined by a graph $G = (V, E)$ with $V = \{v_1, v_2, \dots, v_n, v_{n+1}\}$ with two parallel edges between v_i and v_{i+1} for $i = 1, \dots, n$ and an edge $e^+ = (s, t)$. The latency functions of the two edges between v_i and v_{i+1} are $\ell(x) = w(a_{2i-1})x$ and $\ell(x) = w(a_{2i})x$ for $i = 1, \dots, n$ and $\ell_{e^+}(x) = Bx$. Finally, two users travel from $s = v_1$ to $t = v_{n+1}$. We denote $G \setminus \{e^+\}$ by G' .

We show that $I(\text{EOP})$ is a YES-instance if and only if there exists a Nash equilibrium f in G of $I(\text{WNash})$ with costs $C(f) \geq 2B$.

Let $I(\text{EOP})$ be a YES-instance and A' its solution. Construct a flow f in G by sending one user along the edges with slope $w(a')$ for $a' \in A'$ and one user on the remaining edges of G' .

The cost of f can be calculated as follows

$$C(f) = \sum_{e \in E} \ell_e(f_e) \cdot f_e = \sum_{e \in E \setminus \{e^+\}} \ell_e(1) = \sum_{a \in A} w(a) = 2B$$

Additionally observe that f is a Nash flow: Both users can not benefit from changing their path in G' due to (4) and as both chosen paths have latency B they have no incentive to change to edge e^+ either.

On the contrary assume f to be a Nash flow with cost $C(f) \geq 2B$.

Case 1: $f_{e^+} = 0$

Denote by P_1, P_2 the paths chosen by the two users. No user choosing e^+ implies that $\ell_{P_i}(f) \leq B$, $i = 1, 2$. Additionally equation (4) tells us that $P_1 \cap P_2 = \emptyset$ in

f . We can conclude that the slopes on the edges of P_1 and P_2 form an even-odd partition.

Case 2: $f_{e^+} = 2$

Consider the path P in G' including the edges with slopes $\bar{a}_i = \min\{a_{2i-1}, a_{2i}\}$, $i = 1, \dots, n$. We know that $\sum_{i=1}^n w(\bar{a}_i) \leq B$ by choice of a_i and equation (4). But as the users on e^+ experience latency $\ell_{e^+}(2) = 2B$ at least one user wants to change to P and hence this situation is not Nash.

Case 3: $f_{e^+} = 1$

Denote by P_1 the path in G' chosen by the second users. We know that $\ell_{P_1}(f) \leq B$ because otherwise f would not be Nash as the user on path P_1 would want to change to P as defined in Case 2. Taking into account that $P_1 \cap e^+ = \emptyset$, we can use the cost of f to find the matching lower bound for the latency on P_1 :

$$2B \leq C(f) = \ell_{P_1}(f) + \ell_{e^+}(f) = \ell_{P_1}(f) + B$$

Hence we have $\ell_{P_1}(f) = B$ and the slopes on P_1 form an even-odd partition. \square

4.3 General topologies

Even more, we want to show that it is hard in the strong sense to find the worst Nash equilibrium in general graph topologies.

Theorem 6. *The problem WNash is NP-hard in the strong sense even on acyclic graphs with two users.*

Proof. Consider an instance $I(\text{BlockP})$ of the strongly NP-complete Blocking Path problem:

BLOCKING PATH PROBLEM (BlockP for short) [9]:

Given: Digraph $G = (V, E)$ with source $s \in V$ and sink $t \in V$.

Question: Does there exist an s - t -path $P \in \mathcal{P}$ such that after deleting the edges of P there is no path from s to t ?

An instance $I(\text{WNash})$ of determining a worst pure Nash equilibrium is constructed as follows: $I(\text{WNash})$ is defined on a graph $G' = (V, E')$ which contains the same vertex set as G and $E' = E \cup \{(s, t)\}$. Since G' is acyclic it is possible to define a bijective function $\pi : V \rightarrow \{1, \dots, n\}$ such that $\pi(i) < \pi(j)$ if $(i, j) \in E$. Given any such bijection π the latency functions are given by

$$\ell_e(x) = (\pi(j) - \pi(i))x, \quad e = (i, j) \in E.$$

Observe that due to this definition of the latency functions of edges in G every path from s to t is a shortest path with respect to the edge lengths $\ell_e(1)$. Let L^* be the length of a shortest path from s to t in G with respect to edge lengths $\ell_e(1)$ for $e \in E$. Then the latency of (s, t) is defined by $\ell_{(s,t)}(x) = (L^* + \frac{1}{2})x$.

It remains to show that there exists a blocking path P^* for $I(\text{BlockP})$ if and only if there is a Nash equilibrium f in G' with cost $C(f) \geq 2 \cdot L^* + \frac{1}{2}$.

Given a blocking path P^* in $I(\text{BlockP})$ we choose paths $P_1 = P^*$ and $P_2 = (s, t)$ and construct a flow f with $f_{P_1} = f_{P_2} = 1$ while the flow on all other paths is equal to zero. Hence, we know that $C(f) = 2 \cdot L^* + \frac{1}{2}$.

Additionally f is a Nash equilibrium as neither the user on path P_1 nor the user on P_2 can benefit from changing to another path. First consider the user on path P_1 : Every path P in G has latency L^* as long as there is one user on P_2 . Moreover,

$\ell_{P_2}(\tilde{f}) = 2L^* + 1$ for $\tilde{f}_{P_2} = 2$. Hence, the user of P_1 is satisfied with her situation. In the second step, we check path P_2 : Compare the flow \tilde{f} with $\tilde{f}_{P_1} = \tilde{f}_P = 1$ for some path P in G . Observe that P shares at least one edge e' with the blocking path P_1 and since all slopes of latency functions in G are integral, we have $\ell_{e'}(1) \geq 1$. Therefore,

$$\begin{aligned} \ell_P(\tilde{f}) &= \sum_{e \in P} \ell_e(\tilde{f}) \geq \sum_{e \in P \setminus \{e'\}} \ell_e(1) + \ell_{e'}(2) \\ &= \sum_{e \in P} \ell_e(1) + \ell_{e'}(1) = L^* + \ell_{e'}(1) \geq L^* + 1. \end{aligned}$$

Since $\ell_{P_2}(f) = L^* + \frac{1}{2} < \ell_P(\tilde{f})$ holds for all alternative paths P in G we conclude that f is a Nash equilibrium.

On the other hand, given a Nash equilibrium f with cost $C(f) \geq 2 \cdot L^* + \frac{1}{2}$ we distinguish three cases to show that there exists a blocking path in $I(\text{BlockP})$:

1. Both users share a path, i.e. there is one path P_1 with $f(P_1) = 2$:
But this situation is not stable and thus no Nash equilibrium: If $P_1 = (s, t)$ we know that $\ell_{P_2}(f) = 2L^* + 1$. However, one user would be better off by changing the flow to \tilde{f} with $\tilde{f}_{P_2} = \tilde{f}_P = 1$ for some path P in G as then $\ell_P(\tilde{f}) = L^* < \ell_{P_2}(f)$. If P_1 is a path in G , $\ell_{P_1}(f) = 2 \cdot L^*$ implies that one user wants to change to $P_2 = (s, t)$ because her latency would be only $L^* + \frac{1}{2}$ this way.
2. Both users travel on two distinct paths P_1, P_2 through G : W.l.o.g. we can assume that $\ell_{P_1}(f) \geq \ell_{P_2}(f)$. Consider the flow \tilde{f} with $\tilde{f}_{P_2} = \tilde{f}_{(s,t)} = 1$ with $\ell_{(s,t)}(\tilde{f}) = L^* + \frac{1}{2}$. Since f is a Nash equilibrium, we have $\ell_{P_2}(f) \leq \ell_{P_1}(f) \leq \ell_{(s,t)}(\tilde{f})$ and by integrality of the latencies we conclude

$$C(f) = \ell_{P_1}(f) + \ell_{P_2}(f) \leq 2 \cdot L^*.$$

This leads to a contradiction to the lower bound on the cost of f .

3. Only one user travels through G on a path P_1 and the second user on $P_2 = (s, t)$:
We show that P_1 is a blocking path in G . Assume that P_1 is not blocking. Then there exists a path P' from s to t in G such that P' does not share an edge with P_1 . Hence, the flow \tilde{f} with $\tilde{f}_{P_1} = \tilde{f}_{P'} = 1$ induces latency $\ell_{P'}(\tilde{f}) = L^*$ and implies $\ell_{P_2}(f) = L^* + \frac{1}{2} > L^* = \ell_{P'}(\tilde{f})$. This contradicts f being Nash and thus P_1 is a blocking path in G . \square

References

1. B. Awerbuch, Y. Azar, and A. Epstein, *The price of routing unsplittable flow*, Proceedings 37th Annual ACM Symposium on the Theory of Computing (STOC), 2005, pp. 57–66.
2. A. Czumaj and B. Vöcking, *Tight bounds for worst-case equilibria*, Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2002, pp. 413–420.
3. A. Epstein, M. Feldman, and Y. Mansour, *Efficient graph topologies in network routing games*, Joint Workshop on Economics of Networked Systems and Incentive-Based Computing, 2007.
4. A. Fabrikant, C. Papadimitriou, and K. Talwar, *The complexity of pure nash equilibria*, Proceedings of the 36th Annual ACM Symposium on the Theory of Computing (STOC), 2004, pp. 604–612.
5. R. Feldmann, M. Gairing, T. Lücking, B. Monien, and M. Rode, *Nashification and the coordination ratio for a selfish routing game*, Proceedings of the 30th International Colloquium on Automata, Languages and Programming (ICALP) (J. C. M. Baeten, J. K. Lenstra, J. Parrow, and G. J. Woeginger, eds.), LNCS, vol. 2719, Springer, 2003, p. 190.
6. S. Fischer and B. Vöcking, *On the structure and complexity of worst-case equilibria*, Theoretical Computer Science **378** (2007), no. 2, 165–174.

7. D. Fotakis, S. Kontogiannis, E. Koutsoupias, M. Mavronicolas, and P. Spirakis, *The structure and complexity of nash equilibria for a selfish routing game*, Proceedings of the 29th International Colloquium on Automata, Languages and Programming (ICALP) (P. Widmayer, F. T. Ruiz, R. M. Bueno, M. Hennessy, S. Eidenbenz, and R. Conejo, eds.), vol. 2380, Springer-Verlag, 2002, pp. 123–134.
8. D. Fotakis, S. Kontogiannis, and P. Spirakis, *Symmetry in network congestion games: Pure equilibria and anarchy cost*, Proceedings of the 3rd Workshop on Approximation and Online Algorithms (WAOA) (T. Erlebach and G. Persiano, eds.), LNCS, vol. 3879, Springer, 2005, pp. 161–175.
9. E. Gassner, J. Hatzl, S. O. Krumke, H. Sperber, and G. J. Woeginger, *How hard is it to find extreme nash equilibria in network congestion games*, Proceedings of the 4th International Workshop on Internet and Network Economics (WINE) (C. Papadimitriou and S. Zhang, eds.), LNCS, vol. 5385, 2008, pp. 82–93.
10. E. Koutsoupias and C. Papadimitriou, *Worst-case equilibria*, Proceedings of the 16th International Symposium on Theoretical Aspects of Computer Science (STACS), LNCS, vol. 1563, 1999, pp. 404–413.
11. M. Mavronicolas and P. Spirakis, *The price of selfish routing*, Proceedings of the 33rd Annual ACM Symposium on the Theory of Computing (STOC), 2001, pp. 510–519.
12. C. Papadimitriou, *Algorithms, games, and the internet*, Proceedings of the 33rd Annual ACM Symposium on the Theory of Computing (STOC), 2001, pp. 749–753.
13. A. C. Pigou, *The economics of welfare*, Macmillan, 1920.
14. R. W. Rosenthal, *A class of games possessing pure-strategy nash equilibria*, International Journal of Game Theory **2** (1973), no. 1, 65–67.