Math Meth Oper Res (2010) 71:245–265
DOI 10.1007/s00186-009-0293-6

ORIGINAL RESEARCH

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

Heike Sperber

Received: 12 February 2009 / Accepted: 16 September 2009 / Published online: 27 September 2009

Abstract We study the complexity of finding extreme pure Nash equilibria in symmetric network congestion games and analyse how it is influenced by the graph topology and the number of users. In our context best and worst equilibria are those with minimum or maximum total latency, respectively. We establish that both problems can be solved by a Greedy type algorithm equipped with a suitable tie breaking rule on extension-parallel graphs. 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 concering 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.

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

H. Sperber (✉)
Department of Mathematics, University of Kaiserslautern, P.O. Box 3049, 67653 Kaiserslautern, Germany
e-mail: sperber@mathematik.uni-kl.de

Modern algorithmic game theory brings up additional questions such as computability of equilibria and the overall performance of the system under selfish behaviour. Pigou (1920) gave a first negative answer by stating that in general selfish non-cooperative behaviour does not lead to social optimal outcome. Papadimitriou (2001) introduced the coordination ratio as the quotient of the social cost of a worst Nash equilibrium and the minimum social cost. It is often called "price of anarchy" as it reflects the degradation in performance due to missing central regulation.

Koutsoupias and Papadimitriou (1999) established a model (KP-model) in which users of different sizes 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 (Czumaj and Vöcking 2002; Mavronicolas and Spirakis 2001; Feldmann et al. 2003; Fischer and Vöcking 2007).

The problems of finding extreme (best and worst) Nash equilibria concerning makespan social cost for the KP-Model was addressed by Fotakis et al. (2002), 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. (2007) show all Nash equilibria for unit-sized users on extension-parallel graphs, including the special case of parallel links, to have optimal makespan.

Unit-sized users traveling through more complex graphs are modeled by network congestion games. Rosenthal (1973) established the existence of pure Nash equilibria in these games and Fabrikant et al. (2004) gave a polynomial time algorithm to compute an 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. (2005).

The price of anarchy of these games was studied for two social objectives: Awerbuch et al. (2005) used the total latency as measure of social cost to establish that for affine latency functions the price of anarchy is bounded from above and below by $5/2$. This bound does also hold for the makespan objective, measuring the latency of the longest path chosen by a user, due to results by Christodoulou and Koutsoupias (2005).

Recently Gassner et al. (2008) analysed extreme Nash equilibria in network congestion games for makespan social cost 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.

1.1 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 extension-parallel graphs both problems can be solved by the algorithm GBR (Fotakis et al. 2005) 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-decreasing 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. On these graphs we can adapt the dynamic programming approach given by Gassner et al. (2008) for the makespan case to find extreme Nash equilibria in pseudo-polynomial time.

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:

Find a ... *on ...*	Nash equilibrium with minimum total latency	Nash equilibrium with maximum total latency
Extension-parallel graphs	Polynomially solvable for increasing latencies by Greedy$^{\text{min}}$ (Sect. 3.1)	Polynomially solvable for non-decreasing latencies by Greedy$^{\text{max}}$ (Sect. 4.1)
Series-parallel graphs	Polynomially solvable for two users by Greedy$^{\text{min}}$ NP-hard for three or more users (Sect. 3.2)	NP-hard for two or more users (Sect. 4.2)
	Pseudo-polynomially solvable by dynamic programming for fixed number of users (based on results of Gassner et al. (2008), Sects .3.2 and 4.2)	
General acyclic graphs		NP-hard in the strong sense (Sect. 4.3)

Road Map We start by introducing notation and preliminary results in Sect. 2 and then establish our results on finding a best Nash equilibrium in Sects. 3 and 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 is thus the set of 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* $\ell_e\colon \mathbb{N}_0 \to \mathbb{R}_+^0$ 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 $f\colon \mathcal{P} \to \mathbb{N}_0$ 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:

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

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. (2008) give an example that the property of a feasible flow to be at 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

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

Rosenthal (1973) 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, our social objective:

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:

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

While all Nash equilibria on parallel links have optimal makespan the situation is more difficult for total latency social cost already on this very easy topology:

Example 1 (Nash equilibria with different social cost) Consider the graph G in Fig. 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 $\ell_{e_2}(x) = 2x$. Two users want to travel from s to t.

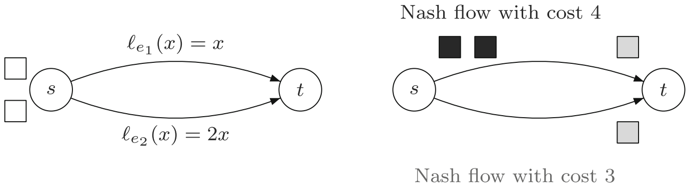

Fig. 1 Two Nash equilibria on parallel links need not have the same total latency (Example 1)

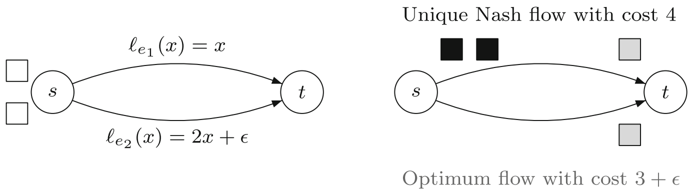

Fig. 2 Nash equilibria on parallel links need not be optimal concerning total latency (Example 2)

In this setting there are two Nash equilibria: One sending all flow on edge e_1 for costs of four and the second one sending one user on each edge e_1 and e_2 resulting in lower costs of three. The latter one is also optimal.

In general no Nash equilibrium is optimal concerning total latency even on parallel links:

Example 2 (No optimal Nash equilibrium) Consider a slight modification of the previous Example 1 given in Fig. 2. We again consider the graph G with two parallel links but with latencies $\ell_{e_1}(x) = x$, $\ell_{e_2}(x) = 2x + \epsilon$ for $0 < \epsilon < 1/2$. Two users travel from s to t. The unique Nash equilibrium sends both users on edge e_1 for costs of four, 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 extension-parallel graphs including parallel links, series-parallel and arbitrary (acyclic) graphs as the complexity of the problems defined above varies on these topologies. We start with the recursive definition of series-parallel and extension-parallel graphs:

Definition 5 *(Series-parallel and extension-parallel graphs)* 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.

Extension-parallel graphs are a special case of series-parallel ones, in which every series composition is only an extension composition $E(G_1, e = (s_2, t_2))$ or $E(e = (s_1, t_1), G_2)$, namely an extension-parallel graph is extended by adding a single edge e either to its source or sink.

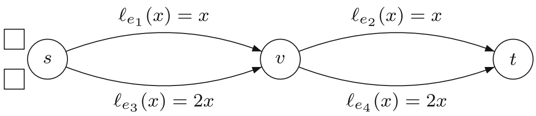

Fig. 3 Series-parallel but not extension-parallel graph

The smallest graph that is series-parallel but not extension-parallel is the one shown in Fig. 3 with three vertices s, v and t and four edges, two connecting each s to v and v to t, respectively. The restriction of the more general series composition to extension compositions is crucial: On extension-parallel graphs all Nash equilibria have equal and optimal makespan (Epstein et al. 2007) while already on the smallest only series-parallel graph mentioned above this is not true for all choices of latency functions on edges. In particular, in the instance given in Fig. 3 one Nash equilibrium f uses paths $P_1 := \{e_1, e_2\}$, $P_2 \coloneqq \{e_3, e_4\}$ with $C_{\max}(f) = 4$ but a second Nash flow g sending users on paths $Q_1 := \{e_1, e_4\}$ and $Q_2 := \{e_3, e_2\}$ has makespan $C_{\max}(g) = 3$.

To clearly express the relation of flows in a series- or extension-parallel graph and its components, we denote the restriction of a flow f in graph $G = P(G_1, G_2)$ or $G = S(G_1, G_2)$ to one of the components G_i by $f|_{G_i}$. In parallel composed graphs $G = P(G_1, G_2)$ we use the notation $f = f_1 \cup f_2$ to express that f sends users on exactly those paths chosen by f_1 in G_1 and f_2 in G_2. In contrast to that we refer to the canonical summation of two flows f, g in one graph G as $f + g$ with $(f + g)_P = f_P + g_P$ for all paths P in G. Additionally we denote by δ_P the flow that sends one unit along path P and zero flow on all other paths.

For our positive results we modify the algorithm GBR introduced by Fotakis et al. (2005) 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 ith iteration, f_0 the constant zero flow on all edges and

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

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. (2005) 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 select 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 the 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 $\Delta c_e : \mathbb{N} \to \mathbb{R}_+^0$ 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*.

To ease the notation if flows are constructed out of several components we sometimes use $\Delta c_e(f)$ for a flow f with the natural meaning of $\Delta c_e(f_e)$, i.e., we leave out the restriction to edge e.

Observe that the running time of Greedy$^{\min}$ [Greedy$^{\max}$] is still 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.

To show that our variants of GBR have the desired properties we later use the following result of Epstein et a. (2007) about Nash equilibria on extension-parallel graphs.

Theorem 1 *(Epstein et al. 2007) On extension-parallel graphs all Nash equilibria of a symmetric network congestion game have the same makespan which equals the optimal one.*

Observing that $L^+(f)$ of a Nash flow f for N users is the makespan of a Nash equilibrium with $N+1$ users yields the following corollary.

Corollary 1 *On extension-parallel graphs any two Nash equilibria f and g of a symmetric network congestion game have $L^+(g) = L^+(f)$.*

We are now prepared to study first best and then worst Nash equilibria on several graph topologies.

3 Best Nash equilibrium

3.1 Extension-parallel graphs

In this section we want to establish, that on these graphs we can find a best Nash equilibrium in polynomial time by applying Greedy$^{\min}$ if the latency functions are increasing. Unfortunately Greedy$^{\min}$ fails for only non-decreasing functions.

Theorem 2 *On extension-parallel graphs Greedy$^{\min}$ solves* BNash *for increasing latency functions.*

Proof We prove the result by induction on the number of composition step necessary to construct the graph G of the underlying network congestion game $[G = (V, E), (\ell_e)_{e \in E}, s \in V, t \in V, N]$.

The base case of the induction on a single edge is trivial. The extension composition $G = E(G_1, e_2)$ is also easy, as the cost of any equilibrium f in G decomposes into the cost of the flow restricted to G_1 and the cost on e_2, which is $N \cdot \ell_{e_2}(N)$ where the first summand is minimum among the Nash equilibria on G_1 by induction hypothesis.

The interesting case is that of a parallel composition $G = P(G_1, G_2)$: Let f be the result of Greedy$^{\min}$ on G and $f_i := f|_{G_i}, i = 1, 2$ the flows in the two components. W.l.o.g. we assume $C_{\max}(f) = C_{\max}(f_1)$ and thus f_1 sends $0 < k \leq N$ users from s to t.

Let g be another Nash equilibrium of the game and $g_i := g|_{G_i}$ the flows in the two graph components, where g_1 sends $0 \leq l \leq N$ users from s to t. We want to show that $C(f) \leq C(g)$.

If $l = k$ we are done, as by induction hypothesis we know that f_i does not induce higher costs than g_i, $i = 1, 2$ and hence

$$C(f) = C(f_1) + C(f_2) \leq C(g_1) + C(g_2) = C(g).$$

For the case $l < k$ consider the flow $\bar{g}$ constructed from the two flows $\bar{g}_i$, $i = 1, 2$ in the two graph components as the result of Greedy$^{\min}$ with l and $N - l$ users, respectively. Hence $\bar{g}$ sends as many users as g in the two component graphs but the paths are chosen according to Greedy$^{\min}$. Using the induction hypothesis on each graph component we conclude

$$C(g) = C(g_1) + C(g_2) \geq C(\bar{g}_1) + C(\bar{g}_2) = C(\bar{g}).$$

Thus it suffices to compare the cost of $\bar{g}$ and f to prove the result.

Observe that $\bar{g}$ is Nash: First of all $\bar{g}_1$ and $\bar{g}_2$ are Nash so no user wants to change her strategy within her graph component. Additionally no user can benefit from changing to the other component as no user can do so in g and $C_{\max}(g_i) = C_{\max}(\bar{g}_i)$ as well as $L^+(g_i) = L^+(\bar{g}_i)$, $i = 1, 2$ due to Theorem 1 and Corollary 1.

The two Nash flows f and $\bar{g}$ share l paths in G_1 and $N - k$ paths in G_2. This "basic" flow sends in all $N - k + l$ users via $\bar{g}_1 \cup f_2$. The Nash flow f sends additional $k - l$ users on paths $P_1, \ldots, P_{k-l}$ through G_1 and $\bar{g}$ sends the same number of additional users through G_2 on paths $Q_1, \ldots, Q_{k-l}$. Thus we know

$$f = \left(\bar{g}_1 + \sum_{i=1}^{k-l} \delta_{P_i}\right) \cup f_2 \quad \text{and} \quad \bar{g} = \bar{g}_1 \cup \left(f_2 + \sum_{i=1}^{k-l} \delta_{Q_i}\right)$$

Informally, the basic idea of the next and final steps is to show that all paths $P_1, \ldots, P_{k-l}, Q_1, \ldots, Q_{k-l}$ have makespan latency once they are chosen and hence Greedy$^{\min}$ working on G and not on the components separately chooses the cheaper ones due to its tie breaking.

To be more precise, we show for $i = 1, \ldots, k - l$ that

$$\ell_{P_i}\left(\bar{g}_1 + \sum_{j=1}^{i} \delta_{P_j}\right) = C_{\max}(f) \quad \text{and} \quad \ell_{Q_i}\left(f_2 + \sum_{j=1}^{i} \delta_{Q_j}\right) = C_{\max}(f).$$

We first analyze the latency of paths $P_1, \ldots, P_{k-l}$ in the iteration they are chosen by Greedy$^{\min}$ on G and can bound it from above and below by $C_{\max}(f)$ as follows:

$$\ell_{P_i}\left(\bar{g}_1 + \sum_{j=1}^{i} \delta_{P_j}\right) \leq \ell_{P_i}(f_1) \leq C_{\max}(f)$$

$$\ell_{P_i}\left(\bar{g}_1 + \sum_{j=1}^{i} \delta_{P_j}\right) \geq \ell_{P_i}(\bar{g}_1 + \delta_{P_i}) \geq L^+(\bar{g}_1) \geq C_{\max}(\bar{g}) = C_{\max}(f)$$

Analogous arguments apply to paths $Q_1, \ldots, Q_{k-l}$ in the iteration they are chosen by Greedy$^{\min}$ on G_2 and this also tells us, that all paths $P_1, \ldots, P_{k-l}, Q_1, \ldots, Q_{k-l}$ are pairwise edge disjoint for increasing latency functions.

Now we can use the disjointness of paths and the tie breaking rule of Greedy$^{\min}$ to compare the costs of f and $\bar{g}$:

$$\begin{aligned} C(f) - C(\bar{g}) &= C(\bar{g}_1 \cup f_2) + \sum_{i=1}^{k-l} \sum_{e \in P_i} \Delta c_e \left(\bar{g}_1 + \sum_{j=1}^{i} \delta_{P_j}\right) \\ &\quad - \left[C(\bar{g}_1 \cup f_2) + \sum_{i=1}^{k-l} \sum_{e \in Q_i} \Delta c_e \left(f_2 + \sum_{j=1}^{i} \delta_{Q_j}\right)\right] \\ &= \sum_{i=1}^{k-l} \left[\sum_{e \in P_i} \Delta c_e(\bar{g}_1 + \delta_{P_i}) - \sum_{e \in Q_i} \Delta c_e(f_2 + \delta_{Q_i})\right] \\ &\leq 0 \quad \text{due to tie breaking of Greedy}^{\min} \text{ on } G \end{aligned}$$

The case $k > l$ can be treated analogously and this completes the proof. □

The following example establishes that in general Greedy$^{\min}$ does not work for non-decreasing latencies and thus the assumption of Theorem 2 is necessary:

Example 3 (Greedy$^{\min}$ fails for non-decreasing latency functions) Consider the graph G in Fig. 4 with five parallel edges between s and t. There is one edge

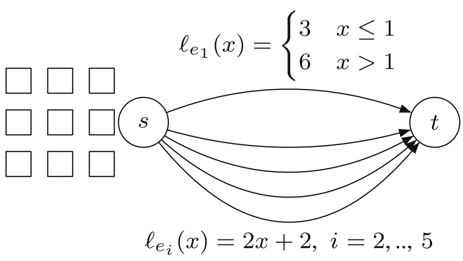

Fig. 4 Greedy$^{\min}$ fails for non-decreasing latencies on parallel links (Example 3)

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 $\ell_{e_i}(x) = 2x + 2$, $i = 2, \ldots, 5$. We want to send nine users from s to t.

Greedy$^{\text{min}}$ 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:

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

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

In the proof of Theorem 2 we used strict monotonicity of the latency functions to establish that the paths chosen by users in Nash flow f but not in g are edge disjoint. This is not true for Example 3 as here the two Nash equilibria differ by more than one user on edge e_1.

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 Fig. 5 for three users.

The solution of GBR for the graph given in Fig. 5 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$^{\text{min}}$ to find a best Nash equilibrium on series-parallel graphs in polynomial time.

Theorem 3 *Greedy*$^{\text{min}}$ *solves* BNash *for two users on series-parallel graphs.*

Proof Again, 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$^{\text{min}}$ finds a best one.

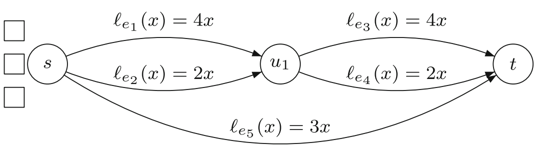

Fig. 5 Best Nash equilibrium not found by GBR on series-parallel network (Example 4)

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 $C(f) \le C(g)$.
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

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

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

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

   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 $C(f) \le C(g)$.
      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

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

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

Let $\bar{g}$ be the flow that uses paths Q_1, $\bar{Q}_2$ and $j \in \{1, \ldots, l-1\}$ 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

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

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

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

By summing over j we conclude $\ell_{\bar{Q}_2}(\bar{g}) \le \ell_{P_2}(f) < \ell_{Q_2}(g)$ which contradicts g being Nash as the user on path Q_2 wants to switch to $\bar{Q}_2$.

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

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

□

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 4 *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 NP-hard even–odd partition problem as stated in Garey and Johnson (1979):

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

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

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 $\ell_{e^+}(x) = (B/2)x$. Three users travel from $s = v_1$ to $t = v_{n+1}$. We denote $G \backslash \{e^+\}$ 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 $C(f) \le 5B/2$.

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 (2) and as both chosen paths have latency B they do not want to change to edge e^+ either. The user on edge e^+ also has no incentive to change to any path in G'.

Conversely assume f to be a Nash flow with cost $C(f) \le 5B/2$.

Case 1: $f_{e^+} = 3$
Let P be the path in G' whose edges have slopes $\min\{w(a_{2i-1}), w(a_{2i})\}$. 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 $\ell_{e^+}(f) = \ell_{e^+}(3) = 3B/2$. Hence f can't be Nash in this case.

Case 2: $f_{e^+} = 2$
Let P_1 be the path chosen by the user not traveling on e^+ and P as in Case 1. We know about the cost of f:

$$C(f) = 2 \cdot \ell_{e^+}(2) + \ell_{P_1}(f) \geq 2 \cdot B + \sum_{e \in P} \ell_e(1) \tag{3}$$

Equation (2) implies

$$\begin{aligned} \sum_{e \in P} \ell_e(1) &= \sum_{i=1}^{n} \min\{w(a_{2i-1}), w(a_{2i})\} \\ &> \sum_{i=1}^{n} \frac{1}{2} \max\{w(a_{2i-1}), w(a_{2i})\} \\ &\geq \frac{1}{2} B \end{aligned} \tag{4}$$

Using Eqs.(3) and (4) 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 (2) we know that $P_1 \cap P_2 = \emptyset$ and hence

$$\ell_{P_1}(f) + \ell_{P_2}(f) = 2B \tag{5}$$

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

$$B = \ell_{e^+}(f_{e^+} + 1) \geq \ell_{P_i}(f) \quad \text{for } i = 1, 2 \tag{6}$$

Combining Eqs.(5) and (6) 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 cannot constitute a Nash equilibrium, as at least one user wants to change to edge e^+. □

Gassner et al. (2008) give a dynamic programming approach on series-parallel graphs that tests for a fixed number of users N in pseudo-polynomial time which of the multisets $\{\ell_1, \ldots, \ell_N\}$ with $0 \leq \ell_i \leq |V| \cdot \max_{e \in E} \ell_e(N)$ correspond to the latency experienced by the 1 to N users in a Nash equilibrium of the network congestion game on this graph. The social cost of a Nash equilibrium, in our meaning of total latency, can be equivalently formulated using the latencies of the users having chosen paths $P_1, \ldots, P_N$ as

$$C(f) := \sum_{e \in E} \ell_e(f_e) \cdot f_e = \sum_{i=1}^{N} \ell_{P_i}(f).$$

Hence using the data generated by the dynamic programming approach, we simply have to sum up the entries of each multiset that corresponds to a Nash flow in G and take the minimum and this does not affect the pseudo-polynomial running time.

4 Worst Nash equilibrium

4.1 Extension-parallel graphs

Analogously to the case of finding the best equilibrium in Theorem 2 we can establish a result for finding a worst Nash equilibrium by Greedy$^{\max}$ and in this case we don't need increasing latencies as stated in the following theorem:

Theorem 5 *Greedy*$^{\max}$ *solves* WNash *on extension-parallel graphs for non-decreasing latency functions.*

Proof The proof is similar to the one for Greedy$^{\min}$ but with one crucial difference, as Theorem 5 holds for non-decreasing latencies in contrast to Theorem 2.

Observe that the base case of the induction on the number of decompositions necessary to construct G and the induction step if the last composition is an extension composition is easily verified.

The interesting case is the parallel composition $G = P(G_1, G_2)$: We adopt the notation of the proof of Theorem 2 and denote by f be the result of Greedy$^{\max}$ on G, $f_i := f|_{G_i}$, $i = 1, 2$ the flows in the two components with $C_{\max}(f) = C_{\max}(f_1)$ where f_1 sends $0 < k \leq N$ users from s to t. We also consider an arbitrary Nash equilibrium g of the game with $g_i := g|_{G_i}$, $i = 1, 2$ where g_1 sends $0 \leq l \leq N$. We want to show that $C(f) \geq C(g)$.

Again if $l = k$ we are done, as by induction hypothesis we know that f_i does not induce lower costs than g_i, $i = 1, 2$ and hence

$$C(f) = C(f_1) + C(f_2) \geq C(g_1) + C(g_2) = C(g).$$

For the case $l < k$ we use the flow $\bar{g}$ constructed from the two flows $\bar{g}_i$, $i = 1, 2$ in the two graph components as the result of Greedy$^{\max}$ with l and $N - l$ users, respectively. Applying the induction hypothesis on each graph component we conclude $C(g) = C(g_1) + C(g_2) \leq C(\bar{g}_1) + C(\bar{g}_2) = C(\bar{g})$ and hence we only have to compare the cost of $\bar{g}$ and f to prove the result.

This flow $\bar{g}$ is Nash because g is and $C_{\max}(g_i) = C_{\max}(\bar{g}_i)$ as well as $L^+(g_i) = L^+(\bar{g}_i)$ due to Theorem 1 and Corollary 1.

We decompose the two Nash flows f and $\bar{g}$ via a "basic" flow sending in all $N-k+l$ users by $\bar{g}_1 \cup f_2$. The Nash flow f sends additional $k-l$ users on paths $P_1, \ldots, P_{k-l}$

through G_1 and $\bar{g}$ sends the same number of additional users through G_2 on paths $Q_1, \dots, Q_{k-l}$. Thus we know

$$f = \left(\bar{g}_1 + \sum_{i=1}^{k-l} \delta_{P_i}\right) \cup f_2 \qquad \text{and} \qquad \bar{g} = \bar{g}_1 \cup \left(f_2 + \sum_{i=1}^{k-l} \delta_{Q_i}\right)$$

Analogously to the proof of Theorem 2 we can establish that for $i = 1, \dots, k-l$

$$\ell_{P_i}\left(\bar{g}_1 + \sum_{j=1}^{i} \delta_{P_j}\right) = C_{\max}(f) \quad \text{and} \quad \ell_{Q_i}\left(f_2 + \sum_{j=1}^{i} \delta_{Q_j}\right) = C_{\max}(f).$$

Hence, edges in the intersection of two paths Q_{i_1}, Q_{i_2} with $i_1 \neq i_2$ must have constant latency after the first of the two paths is chosen by Greedy$^{\max}$ on G_2. More precisely, we know for any $i \in \{1, \dots, k-l\}$ and any $e \in Q_i$ that

$$\ell_e(f_2 + \delta_{Q_i}) = \ell_e\left(f_2 + \sum_{j=1}^{i} \delta_{Q_j}\right) = \ell_e\left(f_2 + \sum_{j=1}^{k-l} \delta_{Q_j}\right)$$

and thus

$$\Delta c_e\left(f_2 + \delta_{Q_i}\right) \geq \Delta c_e\left(f_2 + \sum_{j=1}^{i} \delta_{Q_j}\right) \geq \Delta c_e\left(f_2 + \sum_{j=1}^{k-l} \delta_{Q_j}\right). \tag{7}$$

Hence the cost increase for a path Q_i can only decrease due to the additional choice of paths $Q_1, \dots, Q_{i-1}$.

Now we can use this observation and the tie breaking rule of Greedy$^{\max}$ to compare the costs of f and $\bar{g}$:

$$\begin{aligned}
C(f) - C(\bar{g}) &= C(\bar{g}_1 \cup f_2) + \sum_{i=1}^{k-l} \sum_{e \in P_i} \Delta c_e\left(\bar{g}_1 + \sum_{j=1}^{i} \delta_{P_j}\right) \\
&\quad - \left[C(\bar{g}_1 \cup f_2) + \sum_{i=1}^{k-l} \sum_{e \in Q_i} \Delta c_e\left(f_2 + \sum_{j=1}^{i} \delta_{Q_j}\right)\right] \\
&= \sum_{i=1}^{k-l}\left[\sum_{e \in P_i} \Delta c_e\left(\bar{g}_1 + \sum_{j=1}^{i} \delta_{P_j}\right) - \sum_{e \in Q_i} \Delta c_e\left(f_2 + \sum_{j=1}^{i} \delta_{Q_j}\right)\right] \\
&\geq \sum_{i=1}^{k-l}\left[\sum_{e \in P_i} \Delta c_e\left(\bar{g}_1 + \sum_{j=1}^{i} \delta_{P_j}\right) - \sum_{e \in Q_i} \Delta c_e(f_2 + \delta_{Q_i})\right] \quad \text{by (7)} \\
&\geq 0 \quad \text{due to tie breaking of Greedy}^{\max} \text{ on } G
\end{aligned}$$

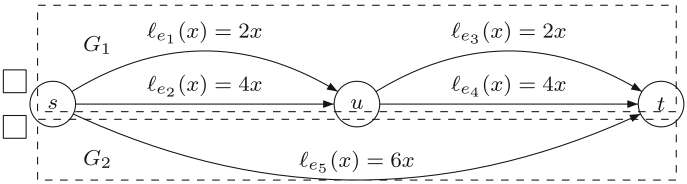

Fig. 6 Worst Nash equilibrium not found by GBR (Example 5)

Analogously we can treat the case $k > l$ and analyze the intersection edges of paths $P_1, \ldots, P_{k-l}$ and this completes the proof. □

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 Fig. 6 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 $C(f) = 10$.

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 eight and thus does not constitute a Nash equilibrium any more if 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 6 *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 4.

Given an instance $I(\mathsf{EOP})$ then an instance $I(\mathsf{WNash})$ is constructed as a graph $G = (V, E)$ with $V = \{v_1, v_2, \ldots, v_n, v_{n+1}\}$ with two parallel edges between v_i and v_{i+1} for $i = 1, \ldots, n$ 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 $i = 1, \ldots, n$ 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 $I(\mathsf{EOP})$ is a YES-instance if and only if there exists a Nash equilibrium f in G of $I(\mathsf{WNash})$ with costs $C(f) \geq 2B$.

Let $I(\mathsf{EOP})$ 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 (2) 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 (2) 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 "cheap" edges with slopes $w_i = \min\{w(a_{2i-1}), w(a_{2i})\}, i = 1, \ldots, n$. We know that $\sum_{i=1}^{n} w_i \leq B$ by choice of a_i and equation (2). 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 user not traveling on e^+. As f is Nash we conclude $\ell_{P_1}(f) \leq B$ as otherwise 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 a solution of $I(\mathsf{EOP})$.

□

As described in Sect. 3.2 we can use the dynamic programming approach of Gassner et al. (2008) to find a worst Nash equilibrium in pseudo-polynomial time.

4.3 General topologies

We strengthen the hardness result in Theorem 6 by showing that it is hard in the strong sense to find the worst Nash equilibrium in general graph topologies.

Theorem 7 *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) (Gassner et al. 2008):
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 there exists a topological sorting $\pi : V \to \{1, \ldots, n\}$ of the vertices, i.e., $\pi(i) < \pi(j)$ if $(i, j) \in E$. Given any such sorting π the latency functions are given by

$$\ell_e(x) = (\pi(j) - \pi(i)) \cdot x \quad \text{for } e = (i, j) \in E.$$

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 $\ell_{(s,t)}(x) = (L^* + 1/2)x$.

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 $C(f) \geq 2 \cdot L^* + 1/2$.

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 $C(f) = 2 \cdot L^* + 1/2$.

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: As every path P in G has latency L^* as long as there is one user on P_2 and $\ell_{P_2}(\bar{f}) = 2L^* + 1$ for $\bar{f}_{P_2} = 2$, the user on path P_1 is satisfied. For the user on P_2 compare the flow $\tilde{f}$ with $\tilde{f}_{P_1} = \tilde{f}_P = 1$ for some path P in G and 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 $\ell_{P_2}(f) = L^* + 1/2 < \ell_P(\tilde{f})$ 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 $C(f) \geq 2 \cdot L^* + 1/2$ we distinguish three cases to show that there exists a blocking path in $I(\text{BlockP})$:

Case 1: Both user 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 $L^* + 1/2$ this way.

Case 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 $\ell_{(s,t)}(\tilde{f}) = L^* + 1/2$. 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.

Case 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 $\ell_{P_2}(f) = L^* + 1/2 > L^* = \ell_{P'}(\tilde{f})$. This contradicts f being Nash and thus P_1 is a blocking path in G. □

5 Conclusion

We have provided a characterization of the complexity of finding extreme Nash equilibria w.r.t. total latency social cost. It turned out that finding best or worst equilibria concerning this social objective function is in general more complex than for makespan social cost:

On extension-parallel graphs all Nash equilibria are optimal w.r.t. makespan due to Epstein et al. (2007) but still easy to find by Greedy type algorithms for total latency social cost (Sects. 3.1 and 4.1).

Concerning series-parallel graphs Gassner et al. (2008) proved that GBR finds a worst Nash equilibrium w.r.t. makespan but a best one is (weakly) NP-hard to find for any fixed number of users. In contrast to these results the complexity in our setting depends not solely on the graph topology but also on the number of users as we can find a best Nash equilibrium w.r.t total latency for two users in polynomial time but it is NP-hard to find for three or more (Sect. 3.2). Finding a worst Nash equilibrium w.r.t. total latency is also (weakly) NP-hard for any fixed number of users (Sect. 4.2).

On acyclic digraphs finding a worst Nash equilibrium w.r.t. either makespan or total latency is NP-hard in the strong sense (Gassner et al. 2008), Sect. 4.3). An interesting open question is, whether finding a best Nash equilibrium w.r.t. total latency is NP-hard in the strong sense either on acyclic graphs or even on series-parallel graphs if the number of users is part of the input. The latter is true for makespan social cost due to Gassner et al. (2008).

Acknowledgments I thank Sven O. Krumke and Florian Seipp for helpful discussions on the topic of this work.

References

Awerbuch B, Azar Y, Epstein A (2005) The price of routing unsplittable flow. In: Proceedings of 37th annual ACM symposium on the theory of computing (STOC), pp 57–66
Christodoulou G, Koutsoupias E (2005) The price of anarchy of finite congestion games. In: Proceedings of the 37th annual ACM symposium on the theory of computing (STOC), pp 67–73
Czumaj A, Vöcking B (2002) Tight bounds for worst-case equilibria. In: Proceedings of the 13th annual ACM–SIAM symposium on discrete algorithms (SODA), pp 413–420
Epstein A, Feldman M, Mansour Y (2007) Efficient graph topologies in network routing games. In: Joint workshop on economics of networked systems and incentive-based computing
Fabrikant A, Papadimitriou C, Talwar K (2004) The complexity of pure nash equilibria. In: Proceedings of the 36th annual ACM symposium on the theory of computing (STOC), pp 604–612
Feldmann R, Gairing M, Lücking T, Monien B, Rode M (2003) Nashification and the coordination ratio for a selfish routing game. In: Baeten JCM, Lenstra JK, Parrow J, Woeginger GJ (eds) Proceedings of the 30th international colloquium on automata, languages and programming (ICALP), LNCS vol. 2719, Springer, NY, pp 514–526
Fischer S, Vöcking B (2007) On the structure and complexity of worst-case equilibria. Theor Comput Sci 378(2):165–174
Fotakis D, Kontogiannis S, Koutsoupias E, Mavronicolas M, Spirakis P (2002) The structure and complexity of Nash equilibria for a selfish routing game. In: Widmayer P, Ruiz FT, Bueno RM, Hennessy M, Eidenbenz S, Conejo R (eds) Proceedings of the 29th international colloquium on automata, languages and programming (ICALP), LNCS vol. 2380, Springer, NY, pp 123–134
Fotakis D, Kontogiannis S, Spirakis P (2005) Symmetry in network congestion games: pure equilibria and anarchy cost. In: Erlebach T, Persiano G (eds) Proceedings of the 3rd workshop on approximation and online algorithms (WAOA), LNCS vol. 3879, Springer, NY, pp 161–175
Garey MR, Johnson DS (1979) Computers and intractability. A guide to the theory of NP-completeness. A series of books in the mathematical sciences. W.H. Freeman, San Francisco
Gassner E, Hatzl J, Krumke SO, Sperber H, Woeginger GJ (2008) How hard is it to find extreme Nash equilibria in network congestion games. In: Papadimitriou C, Zhang S (eds) Proceedings of the 4th international workshop on internet and network economics (WINE), LNCS vol. 5385, pp 82–93
Koutsoupias E, Papadimitriou C (1999) Worst-case equilibria. In: Proceedings of the 16th international symposium on theoretical aspects of computer science (STACS), LNCS vol. 1563, pp 404–413
Mavronicolas M, Spirakis P (2001) The price of selfish routing. In: Proceedings of the 33rd annual ACM symposium on the theory of computing (STOC), pp 510–519
Papadimitriou C (2001) Algorithms, games, and the internet. In: Proceedings of the 33rd annual ACM symposium on the theory of computing (STOC), pp 749–753
Pigou AC (1920) The economics of welfare. Macmillan, NY
Rosenthal RW (1973) A class of games possessing pure-strategy Nash equilibria. Int J Game Theory 2(1):65–67