

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/227294990>

# How Hard Is It to Find Extreme Nash Equilibria in Network Congestion Games?

Conference Paper in Theoretical Computer Science · December 2008

DOI: 10.1007/978-3-540-92185-1\_17

CITATIONS

4

READS

85

5 authors, including:



**Johannes Hatzl**

Graz University of Technology

20 PUBLICATIONS 195 CITATIONS

[SEE PROFILE](#)



**Sven O. Krumke**

RPTU - Rheinland-Pfälzische Technische Universität Ka...

224 PUBLICATIONS 2,382 CITATIONS

[SEE PROFILE](#)



**Heike Sperber**

RPTU - Rheinland-Pfälzische Technische Universität Ka...

8 PUBLICATIONS 75 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Project

Carsharing [View project](#)



Project

LZK-SIM [BAU] [View project](#)

# How Hard Is It to Find Extreme Nash Equilibria in Network Congestion Games?

Elisabeth Gassner<sup>1,\*</sup>, Johannes Hatzl<sup>1</sup>, Sven O. Krumke<sup>2</sup>, Heike Sperber<sup>2,\*\*</sup>,  
and Gerhard J. Woeginger<sup>3,\*\*\*</sup>

<sup>1</sup> Graz University of Technology, Institute of Optimization and Discrete  
Mathematics, Steyrergasse 30, Graz, Austria  
{gassner,hatzl}@opt.math.tu-graz.ac.at

<sup>2</sup> University of Kaiserslautern, Department of Mathematics, P.O.Box 3049,  
67653 Kaiserslautern, Germany  
{krumke,sperber}@mathematik.uni-kl.de

<sup>3</sup> Eindhoven University of Technology, Department of Mathematics and Computer  
Science, P.O.Box 513, 5600 MB Eindhoven, The Netherlands  
gwoegi@win.tue.nl

**Abstract.** We study the complexity of finding extreme pure Nash equilibria in symmetric (unweighted) network congestion games. In our context best and worst equilibria are those with minimum respectively maximum makespan. On series-parallel graphs a worst Nash equilibrium can be found by a Greedy approach while finding a best equilibrium is NP-hard. For a fixed number of users we give a pseudo-polynomial algorithm to find the best equilibrium in series-parallel networks. For general network topologies also finding a worst equilibrium is NP-hard.

**Keywords:** Network congestion game, unsplittable flow, makespan objective, extreme equilibria, complexity.

## 1 Introduction

In the last years there has been a lot of interest in algorithmic game theory combining aspects of game theory and computer science. Driven by growing demand for faster and larger communication networks more and more questions were asked: How do non-cooperative users interact in such networks where increasing load on individual parts of the network causes a degradation in service, often in the form of reduced transfer speed? How does this congestion effect influence the whole network? Is there some kind of self-regulation among the

---

\* This research is partially supported by the Austrian Science Fund Project P18918-N18 *Efficiently solvable variants of location problems*.

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

\*\*\* Supported by the Netherlands Organisation for Scientific Research (NWO), grant 639.033.403, and by BSIK grant 03018 (BRICKS: Basic Research in Informatics for Creating the Knowledge Society).

users? Classical game theory provides qualitative answers such as existence of equilibria, states of the network in which all users are satisfied, and computer scientists added more quantitative question and concepts. It is a well known fact (cf. Pigou [1]), that in general selfish non-cooperative behaviour does not lead to social optimal outcome. Papadimitriou [2] coined the term price of anarchy for the ratio of the social cost of a worst Nash equilibrium and the minimal social cost. The KP-Model named after Koutsoupias and Papadimitriou [3] describes the situation in which users of possibly different size assign their traffic to parallel links with linear latency functions. For pure assignments this corresponds to uniform/related machines in scheduling. Fotakis et al. [4] came up with the question whether a best or worst pure equilibrium w.r.t. to makespan can be computed efficiently and established that in the KP-Model both problems are strongly NP-hard. Gairing et al. [5] added that it is even hard to approximate the worst equilibrium social cost on identical links while there is a PTAS for the best equilibrium social cost. Fischer and Vöcking [6] considered the worst mixed equilibrium.

The hardness proofs for extreme equilibria stated above are based on the users' different sizes, i.e., the amounts of unsplittable traffic they send through the network and the close relationship to scheduling and bin-packing problems. The question arises whether finding extreme Nash equilibria for unit-size users is substantially easier as for the unit-size case the corresponding scheduling and bin-packing instances become polynomially solvable. We will show that most versions of finding extreme equilibria are still NP-hard even for unit-size users. Up to now the complexity status of finding extreme equilibria with respect to the makespan was only considered for the KP-Model. However, in this case finding extreme equilibria for unit-size users is trivial because even for arbitrary non-decreasing latency functions on parallel links all Nash Equilibria have equal and minimal makespan as shown by Epstein et al. [7].

The game describing unit-size users sending their unsplittable traffic through arbitrary directed networks with latency functions on edges is called network congestion game and was already studied in the 1970's by Rosenthal [8]. He established that the more general congestion games possess pure strategy Nash equilibria. Fabrikant et al. [9] established that for symmetric (single-commodity) network congestion games an arbitrary equilibrium can be computed in polynomial time, but for asymmetric network congestion games or general symmetric congestion games it is PLS-complete to find an equilibrium. Fotakis et al. [10] introduced that the greedy approach yields a pure Nash equilibrium not only on parallel links but also on series-parallel graphs.

*Contribution.* We consider (unweighted) network congestion games with arbitrary non-decreasing latency functions on edges. Our negative results need only linear latencies  $\ell_e(x) = a_e x$ .

We establish that finding a best or a worst Nash equilibrium concerning makespan social cost is not equally hard in the following meaning: We prove that on series-parallel graphs finding a best equilibrium is NP-hard. It is strongly NP-hard if the number of users is part of the input and weakly NP-hard

otherwise. Moreover, we suggest a pseudo-polynomial time algorithm that determines a best Nash equilibrium on series-parallel graphs if the number of users is fixed. This indicates that this problem is not strongly NP-hard. In contrast to this we show that a worst pure equilibrium is found by the Greedy approach of Fotakis et al. [10] on these graphs.

In general networks also finding a worst equilibrium is NP-hard. In fact, we prove it to be NP-hard in the strong sense already for two users on an acyclic network with linear latencies.

*Road Map.* The paper is organized as follows: Section 2 introduces our notation and preliminary results such as existence of pure equilibria and computation of an arbitrary equilibrium. In Section 3 we discuss our results on finding a worst Nash equilibrium and in Section 4 for a best Nash equilibrium, respectively.

## 2 Preliminaries

We consider  $N$  users of the same size, i.e., each routing the same amount of unsplittable flow from a single source  $s$  to a single sink  $t$  through a directed graph  $G = (V, E)$ . The edges of  $G$  are equipped with *non-decreasing latency functions*  $\ell_e : \mathbb{N}_0 \rightarrow \mathbb{R}_0^+$  for all  $e \in E$  modelling the congestion effects. An instance of the game is thus given by  $(G = (V, E), (\ell_e)_{e \in E}, s \in V, t \in V, N)$ . By scaling the latency functions appropriately we assume without loss of generality all users to have *unit size*.

Let  $\mathcal{P}$  denote the set of all simple  $s$ - $t$ -paths in  $G$  and thus the strategy set of all users. 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:

$$\ell_P(f) := \sum_{e \in P} \ell_e \left( \sum_{P' \in \mathcal{P}: e \in P'} f_{P'} \right) \quad (1)$$

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. Example 1 shows that we need the information about paths for modelling the users' behaviour in our game.

A Nash equilibrium is a stable situation in which no user wants to deviate from her chosen path because she cannot decrease her experienced latency this way:

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

$$\ell_{P_1}(f) \leq \ell_{P_2}(\tilde{f}) \text{ with } \tilde{f}_P = \begin{cases} f_P - 1 & \text{if } P = P_1 \\ f_P + 1 & \text{if } P = P_2 \\ f_P & \text{otherwise} \end{cases} \quad (2)$$

**Existence of Nash Equilibria.** Rosenthal [8] used the following potential function  $\Pi : \mathcal{F} \rightarrow \mathbb{R}$  defined on the set of feasible flows  $\mathcal{F}$  to prove the existence of pure Nash equilibria in network congestion games:

$$\Pi(f) = \sum_{e \in E} \sum_{i=1}^{f_e} \ell_e(i) \quad (3)$$

Flows corresponding to local optima of this potential function constitute Nash equilibria. Fabrikant et al. [9] establish that one equilibrium can be computed in polynomial time because a min-cost flow in the following instance  $\text{MCF}(G)$  minimizes Rosenthal's potential function and is thus a Nash flow.

**Definition 2 (Min-cost Flow Instance,  $\text{MCF}(G)$ ).** *Given a network congestion game  $(G = (V, E), (\ell_e)_{e \in E}, s \in V, t \in V, N)$  construct the corresponding min-cost flow instance as follows:*

*For every edge  $e \in E$  we need  $N$  copies with costs  $c_{e_i} = \ell_e(i)$ ,  $i = 1, \dots, N$ . The capacities of all edges are 1 and we send  $N$  units of flow from  $s$  to  $t$ .*

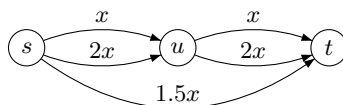
Observe that every path decomposition of every optimal solution of the min-cost flow instance  $\text{MCF}(G)$  yields a Nash equilibrium as the negative cycle optimality condition for optimal min-cost flows directly implies that no user wants to deviate from her chosen strategy. However, not every Nash equilibrium is also an optimal solution of the min-cost flow instance (cf. Examples 3 and 4).

Note that there are instances and Nash flows (not global but local optima of Rosenthal's potential) such that a different path decomposition of the flow on edges induced by a Nash flow is not again Nash (cf. Example 1). Thus, it is necessary to have the information about the flow on paths as the output of the game.

*Example 1 (Nash equilibria and flow decompositions).* Consider the instance given in Figure 1 in which two users travel from  $s$  to  $t$ . The latency functions are given as edge labels. In order to distinguish parallel edges  $(s, u)$  (or  $(u, t)$ ), we call them upper and lower edge between  $s$  and  $u$  ( $u$  and  $t$ ).

Observe that the flow sending the first user on edge  $(s, t)$ , the second user on the path consisting of the upper edge from  $s$  to  $u$  and the lower edge from  $u$  to  $t$  and the third user on the path containing the so far unused edges is a Nash equilibrium. The flow on every edge is equal to 1.

But if we change the flow decomposition and send the second user on both upper and the third on the lower edges this last user becomes unsatisfied because she would be better off changing to edge  $(s, t)$ . Hence, not every path decomposition of a flow on edges yields a Nash equilibrium.



**Fig. 1.** Nash equilibrium property might depend on flow decomposition (Example 1)

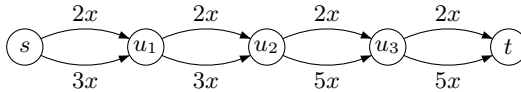
**Social Cost.** In this paper, we consider minimizing the makespan as the social objective function. This notion comes from scheduling and is a priori only applicable to parallel link networks. Flows minimizing the following more general makespan definition are sometimes also called min-max flows.

**Definition 3 (Makespan, Social Cost).** *Given a flow on paths  $f = (f_P)_{P \in \mathcal{P}}$  the makespan is given by*

$$C_{\max}(f) := \max_{P \in \mathcal{P}: f_P > 0} \ell_P(f). \tag{4}$$

Epstein et al. [7] showed that on parallel links all Nash equilibria have equal makespan but this does not hold in general:

*Example 2 (Nash equilibria with different non-optimal makespans).* Consider the instance given in Figure 2 for two users. If every edge is used by exactly one user and the paths are alternating between upper and lower edges then an optimal solution with makespan 12 is achieved. Observe that in any Nash equilibrium there is exactly one user on every edge between  $s$  and  $u_1$  and between  $u_1$  and  $u_2$  and there are two users on the upper edges between  $u_2$  and  $u_3$  and between  $u_3$  and  $t$ . A best Nash equilibrium with makespan 13 can be obtained if both users alternate between upper and lower connection on the first two edges. However, one user may also choose the lower connections on both first edges. This yields again a Nash equilibrium, which is worst and has makespan 14.



**Fig. 2.** Instance with several Nash equilibria

As in general the makespan of different Nash equilibria as well as an optimum makespan are not equal, we are now interested in computing two extreme Nash equilibria.

**Extreme Nash Equilibria.** We introduce the following two problems of finding a best or worst pure equilibrium, respectively.

**WORST NASH EQUILIBRIUM (W-NE for short):**

Given: Network congestion game  $(G = (V, E), (\ell_e)_{e \in E}, s \in V, t \in V, N)$   
 Output: Nash equilibrium  $f$  with maximal makespan among all Nash equilibria.

**BEST NASH EQUILIBRIUM (B-NE for short):**

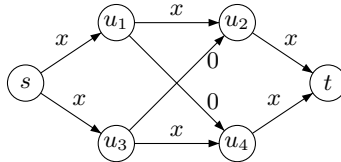
Given: Network congestion game  $(G = (V, E), (\ell_e)_{e \in E}, s \in V, t \in V, N)$   
 Output: Nash equilibrium  $f$  with minimal makespan among all Nash equilibria.

Note that the decision versions of these two problems are in NP for acyclic networks  $G$  as in those networks for a given flow  $f$  a longest path w.r.t. to the fixed edge lengths  $\ell_e(f_e)$  can be computed in polynomial time [11].

Unfortunately, it can be shown that in general neither a best nor a worst Nash equilibrium is an optimal solution of  $\text{MCF}(G)$ :

*Example 3 (Best Nash flow not optimal in  $\text{MCF}(G)$ ).* Reconsider the instance of Example 2 and observe that the unique solution of  $\text{MCF}(G)$  is the Nash equilibrium with makespan 14 and thus not the best one.

*Example 4 (Worst Nash flow not optimal in  $\text{MCF}(G)$ ).* In case of the worst Nash equilibrium consider the instance given in Figure 3 for two users:



**Fig. 3.** The unique worst Nash equilibrium does not imply an optimal min-cost flow (Example 4)

The optimal solution  $f^*$  of  $\text{MCF}(G)$  for the graph given in Figure 3 is unique and has a unique path decomposition sending one user on  $Q_1 = (s, u_1, u_4, t)$  and  $Q_2 = (s, u_3, u_2, t)$  each with makespan  $C_{\max}(f^*) = 2$ . However,  $f$  with  $f_{P_1} = f_{P_2} = 1$  where  $P_1 = (s, u_1, u_2, t)$  and  $P_2 = (s, u_3, u_4, t)$  is a Nash equilibrium with  $C_{\max}(f) = 3$ .

The fact that in general no worst Nash equilibrium is an optimal min-cost flow in  $\text{MCF}(G)$  is quite interesting because in the special case of series-parallel graphs there always exists a worst Nash equilibrium that is an optimal solution of the min-cost flow problem  $\text{MCF}(G)$ . This follows from the result that the Greedy approach determines a worst Nash equilibrium in series-parallel graphs (cf. Section 3).

**Series-Parallel Graphs.** As already mentioned we consider not only arbitrary network topologies but also series-parallel networks. Series-parallel graphs can be defined inductively. 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). And the graph  $G = P(G_1, G_2)$  obtained by identifying  $s_1$  as  $s_2$  and also  $t_1$  as  $t_2$  is a series-parallel graph (parallel composition). This graph has  $s_1(=s_2)$  and  $t_1(=t_2)$  as its terminals (cf. [10]).

This class of graphs has some very nice properties: Bein et al. [12] established that the Greedy approach solves the min-cost flow problem in series-parallel graphs. Combined with the min-cost flow instance introduced by Fabrikant et al. [9] this yields that the greedy approach of iteratively assigning the users to a shortest path with respect to the latency induced by the current flow plus an

additional user on every edge yields a Nash equilibrium on series-parallel graphs. This result was also obtained by Fotakis et al. [10] who call this algorithm GBR (greedy best response) and we keep this notation.

### 3 Worst Pure Nash Equilibrium

In this section the complexity status of determining a worst Nash equilibrium is investigated. We prove that a Greedy strategy solves the problem on series-parallel graphs and show strong NP-hardness for the problem on general graphs.

**Special Case of Series-Parallel Graphs.** In the following we show that the Greedy Best Response (GBR) algorithm introduced by Fotakis et al. [10] always leads to a worst Nash equilibrium in series-parallel graphs. The idea of this algorithm is as follows: If one considers a setting where the users arrive consecutively, a new user routes her path such that her personal latency is minimized given the flow induced by the users currently in the network. This choice is irrevocable, i.e., no user can change the strategy in the future. More formally, let us denote by

$$L^+(f) := \min_{P \in \mathcal{P}} \sum_{e \in P} \ell_e(f_e + 1) \quad (5)$$

the minimum latency for a new  $(N+1)^{\text{st}}$  user given a flow  $f$  sending  $N$  users from  $s$  to  $t$ . According to GBR the new user chooses her path  $P_{N+1}$  such that the latency of  $P_{N+1}$  is  $L^+(f)$ . If a flow  $f'$  is obtained by a given flow  $f$  where a single user is added according to GBR we use  $f' = f \oplus P_{N+1}$ . For series-parallel graphs it has been shown in [10] that if  $f$  is an arbitrary Nash equilibrium then  $f' = f \oplus P_{N+1}$  is again a Nash equilibrium. Note that this property does not hold in general graphs. As a consequence GBR always leads to a Nash equilibrium if all users have the same size and the underlying network is series-parallel. In this paper, we strengthen this result and show that the obtained Nash equilibrium is always a worst Nash equilibrium. This holds for all latency functions that are non-decreasing. The next lemma, which is a key point in order to prove our result, has already been used implicitly in [10]. It states that if we start with a Nash equilibrium and add one more user according to GBR then the latency of the new user is not less than the latency of all the previous users in the new flow.

**Lemma 1.** *Let  $G = (V, E)$  be a series-parallel graph and  $f$  a Nash equilibrium for  $N$  users. If we choose  $P_{N+1} \in \mathcal{P}$  according to GBR we obtain a new Nash equilibrium  $f' = f \oplus P_{N+1}$  such that*

$$\ell_{P_{N+1}}(f') = C_{\max}(f').$$

The next two lemmata are dealing with the two compositions in the definition of series-parallel graphs. In fact, we give a characterization of a Nash equilibrium in  $S(G_1, G_2)$  and  $P(G_1, G_2)$ . Before the results are stated the following notation is introduced. Let  $G_i$  be a series-parallel graph and  $f_i : \mathcal{P}_i \rightarrow \mathbb{N}_0$  a flow in  $G_i$  for



$i = 1, 2$ . Then the set of all simple  $s$ - $t$ -paths in  $P(G_1, G_2)$  is given by  $\mathcal{P}_1 \cup \mathcal{P}_2$ . We define a new flow  $f$  in  $P(G_1, G_2)$  by  $f := f_1 \cup f_2$ , where  $f : \mathcal{P}_1 \cup \mathcal{P}_2 \rightarrow \mathbb{N}_0$  and  $f|_{\mathcal{P}_i} = f_i$  for  $i = 1, 2$ .

**Lemma 2.** *Let  $f_i$  be a flow in a series-parallel graph  $G_i$  for  $i = 1, 2$ . Then  $f = f_1 \cup f_2$  is a Nash equilibrium in  $P(G_1, G_2)$  if and only if the following conditions are satisfied:*

1.  $f_i$  is a Nash equilibrium in  $G_i$  for  $i = 1, 2$ ,
2.  $L_{G_1}^+(f_1) \geq C_{\max}(f_2)$  and  $L_{G_2}^+(f_2) \geq C_{\max}(f_1)$ .

We want to establish a similar result for the series composition. Therefore let  $G_i$  be series-parallel and  $f_i : \mathcal{P}_i \rightarrow \mathbb{N}_0$  a flow in  $G_i$  for  $i = 1, 2$  for  $N$  users. Let us assume without loss of generality that the users choose the paths  $P_1, \dots, P_N$  ( $Q_1, \dots, Q_N$ ) in  $G_1$  ( $G_2$ ). For each permutation  $\phi$  of  $\{Q_1, \dots, Q_N\}$  we can obtain a new flow  $f$  in  $S(G_1, G_2)$  if we define a new path for user  $i$  by  $\bar{P}_i = P_i \cup Q_{\phi(i)}$ . The set of all flows that can be obtained this way will be denoted by  $f_1 \otimes f_2$ .

**Lemma 3.** *Let  $f_i$  be a flow in  $G_i$  for  $i = 1, 2$ . Let  $f \in f_1 \otimes f_2$  then  $f$  is a Nash flow in  $S(G_1, G_2)$  if and only if  $f_i$  is a Nash equilibrium in  $G_i$  for  $i = 1, 2$ .*

Using these lemmata we are able to prove the following theorem by induction on the composition steps. The detailed proof is omitted due to space restrictions.

**Theorem 1.** *If  $G$  is a series-parallel graph then the Nash equilibrium obtained by GBR is a worst Nash equilibrium.*

**Complexity Status on General Graphs.** Before proving NP-hardness of the problem of finding a worst Nash equilibrium, we consider a related problem that is called Blocking Path problem:

BLOCKING PATH PROBLEM (BlockP for short):

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  $t$ ?

**Theorem 2.** *The Blocking Path Problem is strongly NP-complete even on acyclic networks.*

The proof is a reduction from 3SAT and due to lack of space postponed to the full version of this paper. The Blocking Path Problem is used to show that determining a worst Nash equilibrium in general networks is NP-hard even for two users.

**Theorem 3.** *Determining a Worst Pure Nash equilibrium is strongly NP-hard even for two users on acyclic networks and with linear latency functions.*

*Proof.* Consider an instance  $I(\text{BlockP})$  of the strongly NP-complete Blocking Path problem. Let  $G = (V, E)$  be the acyclic network of instance  $I(\text{BlockP})$  with

$s, t \in V$ . An instance of  $I(\text{W-NE})$  of determining a worst pure Nash equilibrium is defined as follows:  $I(\text{W-NE})$  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  $\pi$  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  $\ell_{(s,t)}(x) = (L^* + \frac{1}{2})x$ .

We show that there exists a blocking path  $P^*$  for  $I(\text{BlockP})$  if and only if the answer to the decision problem corresponding to  $I(\text{W-NE})$  is “yes” for  $K = L^* + \frac{1}{2}$ , i.e. there exists a Nash equilibrium  $f$  in  $G'$  with cost  $C_{\max}(f) \geq L^* + \frac{1}{2}$ .

Given a blocking path  $P^*$  in  $I(\text{BlockP})$  we construct a feasible flow  $f$  in  $G'$  by sending one user on  $P^*$  and the other on edge  $(s, t)$  inducing  $C_{\max}(f) = L^* + \frac{1}{2}$ . Observe that indeed both users are satisfied and this flow constitutes a Nash equilibrium.

On the other hand, assume that there exists a Nash equilibrium  $f$  with makespan  $C_{\max}(f) \geq L^* + \frac{1}{2}$ . Analysing the different cases of flow values on  $(s, t)$ , the Nash property of  $f$  together with the lower bound on  $C_{\max}(f)$  tell us that in this setting one user is sent over  $(s, t)$  and one on a path  $P^*$  in  $G'$ . The fact, that the user on  $(s, t)$  does not want to change to  $G'$  implies that  $P^*$  is in fact a blocking path. □

## 4 Best Pure Nash Equilibrium

In this section, we show several complexity results concerning the problem of determining a best Nash equilibrium. All results given in this section hold even for series-parallel graphs. We show that computing a best Nash equilibrium for  $N$  users is strongly NP-hard if  $N$  is part of the input. If the number of users is fixed then the problem remains weakly NP-hard. At least for series-parallel graphs this result is best possible because there exists a dynamic programming algorithm with pseudo-polynomial running time.

**Strong NP-Hardness Result.** In this subsection, we prove that finding a best Nash equilibrium on series-parallel graphs is strongly NP-hard if the number of users is part of the input. We show this by a reduction of the corresponding decision problem to the numerical 3-dimensional matching problem, which is known to be strongly NP-complete (see [11]).

NUMERICAL 3-DIMENSIONAL MATCHING (N3M for short):

Given: Disjoint sets  $X, Y, Z$ , each containing  $m$  elements, a weight  $w(a)$  for all elements  $a \in X \cup Y \cup Z$  and a bound  $B \in \mathbb{Z}^+$ .

Question: Does there exist a partition of  $X \cup Y \cup Z$  into  $m$  disjoint sets  $A_1, \dots, A_m$  such that each  $A_j$  contains exactly one element from each of  $X, Y$  and  $Z$  and  $\sum_{a \in A_i} w(a) = B$  for all  $1 \leq i \leq m$ .

**Theorem 4.** *Determining a best Nash equilibrium is strongly NP-hard on series-parallel graphs if the number of users is part of the input.*

*Proof.* Consider an instance  $I(\text{N3M})$  of N3M. Observe that we may assume without loss of generality that  $w(a) < 2w(b)$  and  $w(b) < 2w(a)$  for each pair  $a, b \in X$ . Otherwise a large number  $M$  can be added to all elements in the set  $X$  and to  $B$  until the desired condition is satisfied. An analogue property holds for  $Y$  and  $Z$ .

Based on this instance we construct the following series-parallel graph  $G = (V, E)$ : Let  $V = (s, u, v, t)$  and for each element in the set  $X$  ( $Y, Z$ ) we introduce a directed edge from  $s$  to  $u$  ( $u$  to  $v$ ,  $v$  to  $t$ ). The latency function of an edge  $e$  is given by  $\ell_e(x) = w(a)x$  where  $w(a)$  is the weight of the corresponding element in the instance  $I(\text{N3M})$ .

Observe that in a best Nash equilibrium every edge is used by exactly one user. Hence, there is a one-to-one correspondence between the subsets  $A_i$   $i = 1, \dots, m$  and the paths of the users and therefore there exists a Nash equilibrium with  $m$  users in  $G$  with social cost at most  $B$  if and only if  $I(\text{N3M})$  is a YES-instance.  $\square$

**Weak NP-Hardness for Fixed Number of Users.** This subsection deals with the problem of determining a best Nash equilibrium if the number of users  $N$  is fixed. The proof is a reduction from Even-Odd Partition. As it works similar to that of the previous section it is omitted here.

**Theorem 5.** *Determining a best Nash equilibrium is weakly NP-hard even for two users and on series-parallel graphs.*

**A Pseudo-Polynomial Time Algorithm for Series-Parallel Graphs.** In this subsection, we discuss a dynamic programming approach to find a best Nash equilibrium in series-parallel graphs if the number  $N$  of users is not part of the input. Let  $f$  be a Nash equilibrium in a graph  $G$  for  $k$  users which choose the paths  $P_1, \dots, P_k$ . Then we define a multiset

$$C(f) := \{\ell_{P_1}(f), \dots, \ell_{P_k}(f)\}$$

which will be called cost profile of  $f$ . Note that several Nash equilibria can have the same cost profile. The idea of the algorithm is to decide if for a given multiset  $C = \{c_1, \dots, c_k\}$  with  $0 \leq k \leq N$  there exists a corresponding Nash flow  $f$  with  $k$  users. This is done using the inductive definition of series-parallel graphs. In order to decide if a cost profile can be realized by a Nash flow  $f = f_1 \cup f_2$  in  $P(G_1, G_2)$  we need to know  $L^+(f_i)$ . More formally, for a given multiset  $C = \{c_1, \dots, c_k\}$  and a graph  $G$  we define

$$S_G(C) := \max\{L^+(f) \mid C(f) = \{c_1, \dots, c_k\}, f \text{ is a Nash flow}\}.$$

If such a Nash equilibrium does not exist we set  $S_G(C) := -\infty$ . Hence, all cost profiles with  $S_G(C) \geq 0$  do have a corresponding Nash flow  $f$ . Let us discuss the algorithm in more detail.

1. A single edge  $(s, t)$

For the simplest series-parallel graph there is a unique flow for all  $0 \leq k \leq N$  and all users have latency  $\ell_{(s,t)}(k)$ . Thus, we obtain immediately

$$S_G(C) = \begin{cases} \ell_{(s,t)}(k+1) & \text{if } C = \{\ell_{(s,t)}(k), \dots, \ell_{(s,t)}(k)\}, |C| = k \leq N, \\ -\infty & \text{otherwise.} \end{cases}$$

2. The series composition

Let  $C = \{c_1, \dots, c_k\}$  be given. Note that this cost profile can only be obtained by a Nash flow  $f \in f_1 \otimes f_2$  with  $C_i := C(f_i) = \{c_1^i, \dots, c_k^i\}$  for  $i = 1, 2$  and  $C = \{c_1^1 + c_{\phi(1)}^2, \dots, c_k^1 + c_{\phi(k)}^2\}$  for some permutation  $\phi$ . If such a permutation exists we write  $C_1 \otimes C_2 = C$ . Moreover,  $L^+(f) = L^+(f_1) + L^+(f_2)$  because every  $s - t$  path in  $G$  has to pass the vertex  $t_1 = s_2$ . Thus, we obtain

$$S_G(C) = \max_{C_1 \otimes C_2 = C} \{S_{G_1}(C_1) + S_{G_2}(C_2)\}. \tag{6}$$

3. The parallel composition

Let  $C = \{c_1, \dots, c_k\}$  be given. A corresponding Nash flow  $f$  is of the form  $f_1 \cup f_2$  with  $C_1 := C(f_1) = \{c_1^1, \dots, c_{k_1}^1\}$ ,  $C_2 := C(f_2) = \{c_1^2, \dots, c_{k_2}^2\}$ ,  $k_1 + k_2 = k$  and  $C = C_1 \cup C_2$ . Moreover the conditions from Lemma 3 have to be satisfied, i.e.,  $\max\{c_1^1, \dots, c_{k_1}^1\} \leq S_{G_2}(C_2)$  and  $\max\{c_1^2, \dots, c_{k_2}^2\} \leq S_{G_1}(C_1)$ . The shortest path in  $G$  with respect to the flow  $f$  is given by  $\min\{L^+(f_1), L^+(f_2)\}$ , because the shortest path in  $P(G_1, G_2)$  chooses either a path with edges in  $G_1$  or in  $G_2$ . Thus,

$$S_G(C) = \max_{\substack{C=C_1 \cup C_2 \\ |C_1|+|C_2|=k \\ \max\{c|c \in C_1\} \leq S_{G_2}(C_2) \\ \max\{c|c \in C_2\} \leq S_{G_1}(C_1)}} \min\{S_{G_1}(C_1), S_{G_2}(C_2)\} \tag{7}$$

is satisfied.

Note that it is straightforward to get the best Nash flow at the end if the corresponding flows which determine  $S_G(C)$  during the algorithm are stored as well. In order to analyze the running time of this algorithm note that for a graph  $G$  and a fixed number  $k$  of users there are at most  $\frac{(|V|L)^k}{k!} = \mathcal{O}((|V|L)^k)$  different multisets, where  $L := \max_{e \in E} l_e(N)$  is the maximum latency on an edge and a simple path can have at most  $|V|$  edges. Due to the fact that this is needed for all  $0 \leq k \leq N$  the number of multisets that have to be stored is at most  $N(|V|L)^N = \mathcal{O}((|V|L)^N)$ . It is easy to see that for the series and parallel composition (6) and (7) can be done in polynomial time with respect to the number of multisets. Thus, the proposed dynamic programming approach is pseudo-polynomial which implies that B-NE is indeed not NP-hard in the strong sense for series-parallel graphs.

## References

1. Pigou, A.C.: The economics of welfare. Macmillan, Basingstoke (1920)
2. Papadimitriou, C.: Algorithms, games, and the internet. In: Proceedings of the 33rd Annual ACM Symposium on the Theory of Computing, pp. 749–753 (2001)

3. Koutsoupias, E., Papadimitriou, C.: Worst-case equilibria. In: Meinel, C., Tison, S. (eds.) STACS 1999. LNCS, vol. 1563, pp. 404–413. Springer, Heidelberg (1999)
4. Fotakis, D., Kontogiannis, S., Koutsoupias, E., Mavronicolas, M., Spirakis, P.: The structure and complexity of nash equilibria for a selfish routing game. In: Widmayer, P., Triguero, F., Morales, R., Hennessy, M., Eidenbenz, S., Conejo, R. (eds.) ICALP 2002. LNCS, vol. 2380, pp. 123–134. Springer, Heidelberg (2002)
5. Gairing, M., Lücking, T., Mavronicolas, M., Monien, B., Spirakis, P.: The structure and complexity of extreme nash equilibria. *Theoretical Computer Science* 343(1-2), 133–157 (2005)
6. Fischer, S., Vöcking, B.: On the structure and complexity of worst-case equilibria. *Theoretical Computer Science* 378(2), 165–174 (2007)
7. Epstein, A., Feldman, M., Mansour, Y.: Efficient graph topologies in network routing games. In: Joint Workshop on Economics of Networked Systems and Incentive-Based Computing (2007)
8. Rosenthal, R.W.: A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory* 2(1), 65–67 (1973)
9. Fabrikant, A., Papadimitriou, C., Talwar, K.: The complexity of pure nash equilibria. In: Proceedings of the 36th Annual ACM Symposium on the Theory of Computing, pp. 604–612 (2004)
10. Fotakis, D., Kontogiannis, S., Spirakis, P.: Symmetry in network congestion games: Pure equilibria and anarchy cost. In: Erlebach, T., Persinao, G. (eds.) WAOA 2005. LNCS, vol. 3879, pp. 161–175. Springer, Heidelberg (2005)
11. Garey, M.R., Johnson, D.S.: Computers and intractability. A guide to the theory of NP-completeness. A Series of Books in the Mathematical Sciences. W. H. Freeman & Co., New York (1979)
12. Bein, W.W., Brucker, P., Tamir, A.: Minimum cost flow algorithm for series-parallel networks. *Discrete Applied Mathematics* 10, 117–124 (1985)