On the Complexity of Pure Nash Equilibria in Player-Specific Network Congestion Games*

Heiner Ackermann and Alexander Skopalik

Department of Computer Science
RWTH Aachen, D-52056 Aachen, Germany
{ackermann, skopalik}@cs.rwth-aachen.de

Abstract. Network congestion games with player-specific delay functions do not necessarily possess pure Nash equilibria. We therefore address the computational complexity of the corresponding decision problem, and show that it is NP-complete to decide whether such games possess pure Nash equilibria. This negative result still holds in the case of games with two players only. In contrast, we show that one can decide in polynomial time whether an equilibrium exists if the number of resources is constant.

In addition, we introduce a family of player-specific network congestion games which are guaranteed to possess equilibria. In these games players have identical delay functions, however, each player may only use a certain subset of the edges. For this class of games we prove that finding a pure Nash equilibrium is PLS-complete even in the case of three players. Again, in the case of a constant number of edges an equilibrium can be computed in polynomial time.

We conclude that the number of resources has a bigger impact on the computation complexity of certain problems related to network congestion games than the number of players.

1 Introduction

Network congestion games are a well-known and generally accepted approach to model resource allocation among selfish agents in large-scale networks like the internet. In these games agents share a network and each of them selects a path with minimum delay (cost, payoff) that connects an individual pair of nodes. The delay of a path equals the sum of delays of the edges in that path, and the delay of an edge depends on the number of players currently using that edge. In recent years network congestion games have been considered in various occurrences and with respect to different questions like the price of anarchy, the computational complexity of finding Nash equilibria[1], or certain network design

* This work was supported in part by the EU within the 6th Framework Programme under contract 001907 (DELIS) and the German Israeli Foundation (GIF) under contract 877/05.

[1] In this paper, the term *Nash equilibrium* always refers to a pure Nash equilibrium.

X. Deng and F.C. Graham (Eds.): WINE 2007, LNCS 4858, pp. 419–430, 2007.

problems. For an introduction into many of these questions we refer the reader to the forthcoming book of Nisan, Tardos, Roughgarden, and Vazirani [12].

In this paper we are interested in *player-specific network congestion games*. In such games we consider a finite set of players and assume that each of them is equipped with a set of player-specific delay functions. This is in contrast to the previously mentioned games in which all players sharing an edge observe the same delay. Player-specific network congestion games naturally arise when different players have different preferences on the edges of the network. Some players might prefer to use motor-ways, others might prefer to use scenic roads. It is well known that player-specific network congestion games do not necessarily possess Nash equilibria [11]. We therefore investigate the computational complexity of deciding whether such a game possesses a Nash equilibrium. We prove by a reduction from the problem Node-Disjoint Path that this problem is NP-complete. We also consider games with constant number of players or resources. In the first case the decision problem remains NP-complete even in the case of two players, whereas in the second case we present a polynomial time algorithm.

In order to bypass the limitations of general player-specific congestion games, we introduce a family of games for which the existence of a Nash equilibrium is guaranteed by Rosenthal's potential function [13]. We assume that all players sharing an edge observe the same delay, however, each player may only use a certain subset of the edges. Such games naturally arise when drivers are prohibited to use certain roads, e. g., trucks may be prohibited to use narrows roads, slow vehicles may be prohibited to use motor-ways. These games – in the following called *restricted network congestion games* – are closely related to standard network congestion games in which players compute their delays with respect to common delay functions and in which each player may use every edge. Fabrikant, Papadimitriou, and Talwar [6] introduce standard network congestion games and show that computing an equilibrium is PLS-complete, that is, computing a Nash equilibrium is "as hard to compute as any object whose existence is guaranteed by a potential function" [6]. Ackermann, Röglin, and Vöcking [1] present a simplified proof for this. Thus, computing a Nash equilibrium of a restricted network congestion game is PLS-complete, too. However, the previously mentioned proofs require an arbitrary number of players and resources. In this paper we consider games in which one of these two parameters is kept constant. In the case of a constant number of player we prove that computing a Nash equilibrium remains PLS-complete, whereas it is polynomial time solvable in the case of constant number of resources. The later result follows easily by a potential function argument and applies to every congestion game with common delay functions and with a constant number of resources. Unfortunately, we failed to prove PLS-completeness for computing Nash equilibria in standard network congestion games with a constant number of players. This question was our primary motivation and remains a challenging open problem.

To the best of our knowledge, this is the first paper systematically comparing the impact of the number of player and of the resources on the computational

complexity of certain problems related to player-specific network congestion games. From our results we conclude that the impact of the number of resources is much bigger than the impact of the number of players.

Player-specific Network Congestion Games. A player-specific network congestion game Γ consists of four components: (1) a network $G = (V, E)$ with m directed edges, (2) a set $\mathcal{N} = \{1, \ldots n\}$ of n players, (3) for every player i a source-sink pair $(s_i, t_i) \in V \times V$, and (4) for every player i and every edge $e \in E$ a non-decreasing delay function $d_i^e \colon \mathbb{N} \to \mathbb{N}$. The *strategy space* of player i equals the set of paths connecting source s_i with target t_i. We denote by $S = (P_1, \ldots, P_n)$ *a state of the game* where player i chooses path P_i. Furthermore, we denote by $n_e(S) = |\{i \in \mathcal{N} \mid e \in P_i\}|$ the congestion on edge e in state S, that is, $n_e(S)$ equals the number of players sharing edge e in state S. Players act selfishly and choose paths with minimum delay given fixed choices of the other players. The delay $\delta_i(S)$ of player i in state S equals $\sum_{e \in P_i} d_i^e(n_e(S))$. Finally, we call a state S a *Nash equilibrium* if no player has an incentive to change her strategy.

It is well know that, in general, a pure Nash equilibrium is not guaranteed to exist. However, if the network consists of parallel links only or if for every edge e the player-specific delay functions d_i^e are identical, then a Nash equilibrium is guaranteed to exist [10,13]. In the following, we consider also network congestion games with common delay functions and assume that each player is restricted to a certain subset of the edges. We call such a game a *restricted network congestion game*. Such a game can easily be interpreted as a player-specific game by defining player-specific delay functions in the following way. If a player is allowed to use an edge, her delay functions equals the common one, if a player is not allowed to use an edge, she observes delay ∞ for every congestion on that edge. By Rosenthal's potential function argument [13], every restricted network congestion games possess a Nash equilibrium.

The Complexity Class PLS. A local search problem Π is given by its set of instances $\mathcal{I}_\Pi$. For every instance $I \in \mathcal{I}_\Pi$, we are given a finite set of feasible solutions $\mathcal{F}(I) \subseteq \{0,1\}^*$, an objective function $c : \mathcal{F}(I) \to \mathbb{N}$, and for every feasible solution $S \in \mathcal{F}(I)$ a neighborhood $\mathcal{N}(S, I) \subseteq \mathcal{F}(I)$. Given an instance I of a local search problem, we seek for a *locally optimal solution* S^*, i. e., a solution which does not have a strictly better neighbor with respect to the objective function c.

A local search problem Π belongs to PLS if the following polynomial time algorithms exist: an algorithm A which computes for every instance I of Π an initial feasible solution $S_0 \in \mathcal{F}(I)$, an algorithm B which computes for every instance I of Π and every feasible solution $S \in \mathcal{F}(I)$ the objective value $c(S)$, and an algorithm C which determines for every instance I of Π and every feasible solution $S \in \mathcal{F}(I)$ whether S is locally optimal or not, and finds a better solution in the neighborhood of S in the latter case.

Johnson et al. [8] introduce the notion of a PLS-*reduction*. A problem Π_1 in PLS is PLS-reducible to a problem Π_2 in PLS if there exist polynomial-time computable functions f and g such that f maps instances I of Π_1 to instances

$f(I)$ of Π_2, g maps pairs (S_2, I) where S_2 denotes a solution of $f(I)$ to solutions S_1 of I, and for all instances I of Π_1, if S_2 is a local optimum of instance $f(I)$, then $g(S_2, I)$ is a local optimum of I. A local search problem Π in PLS is PLS-*complete* if every problem in PLS is PLS-reducible to Π. PLS-completeness results for various local search problems can be found in [8,14].

Related Work. Most closely related to our work is the work of Milchtaich [10,11] and the work of Dunkel and Schulz [5]. Milchtaich [10] introduces player-specific network congestion games on parallel links and proves that every such game possesses a Nash equilibrium if the player-specific delay functions are non-decreasing. In [11] he presents some network topologies such that every player-specific network congestion game on such a topology possesses an equilibrium without any assumption on the delay functions except monotonicity. Dunkel and Schulz [5] consider the computational complexity of deciding whether a weighted network congestion games possesses a Nash equilibrium. In such games players sharing an edge observe the same delay, however, the congestion on an edge depends on the weighted number of players. They prove that this decision problem is NP-complete. Ackermann, Röglin, and Vöcking [2] prove that every player-specific matroid congestion game possesses a Nash equilibrium. In such games the players' strategy spaces are the sets of bases of a matroid on the resources. Additionally, they show that the matroid property is the maximal property on the strategy spaces guaranteeing the existence of equilibria. Chakrabarty, Mehta, and Nagarajan [4] consider player-specific network congestion games on (a constant number of) parallel links from a global optimization perspective, and investigate whether one can compute social optimal states of such games efficiently. Anshelevich et al. [3] and Meyers [9] consider several problem involving congestion games with a constant number of players.

2 General Player-Specific Network Games

In this section, we consider the complexity of deciding whether a general player-specific network congestion game possesses a Nash equilibrium.

Theorem 1. *It is* NP*-complete to decide whether a player-specific network congestion game with* two players *possesses a Nash equilibrium.*

Proof. Obviously, the decision problem belongs to NP as one can decide in polynomial time whether a given state S of such a game is a Nash equilibrium. In order to prove that the problem is complete, we present a polynomial time reduction from the Node-Disjoint Path problem. An instance of the Node-Disjoint Path problem consists of a directed graph $G = (V, E)$ and two pairwise disjoint node pairs (s_1, t_1) and (s_2, t_2). Given such an instance, we like to decide whether there exist node-disjoint paths from s_1 to t_1 and from s_2 to t_2. It is well know that this problem is NP-complete [7].

Given an instance $(G, (s_1, t_1), (s_2, t_2))$ of the problem Node-Disjoint Path we construct a player-specific network congestion game with two players as follows. Given $G = (V, E)$ we substitute every node $v \in V$ by the gadget G_v

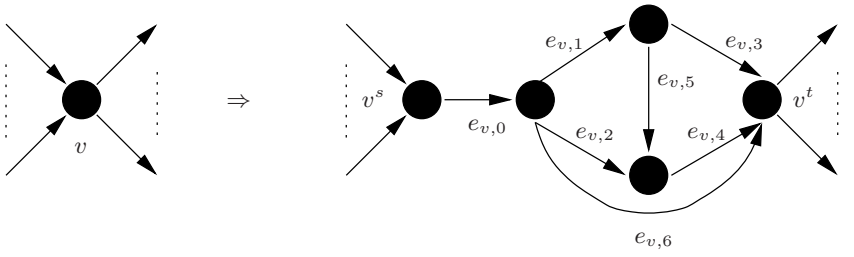

Fig. 1. The gadget G_v of a node v

presented in Figure 1 in order to obtain the network $G_\Gamma = (V_\Gamma, E_\Gamma)$ on which the game is played. Player $i \in \{1, 2\}$ wants to allocate a path between the nodes s_i^s and t_i^t in G_Γ. Observe that this construction ensures a one-to-one corresponds between the paths in G and in G_Γ in the natural way if we ignore the precise subpaths through every gadget. The player-specific delay functions are chosen as follows. For every edge $e = (v_i^t, v_j^s)$, i. e., for edges that represent edges from the original graph G, we assume that for each player and every congestion the delay on such an edge equals 0. In the following, let M be a sufficiently large number. Then, the player-specific delay functions of edges $e_{v,i}$, $i \in \{0, \ldots, 6\}$, are defined as presented in Figure 2. Observe that every gadget G_v implements a subgame that is played by the players if both want to allocate a path connecting the nodes v^s and v^t. If only one player wants to allocate such a path, then she allocates a player-specific shortest path from v^s to v^t. If we choose M sufficiently large, such that the second player will never allocate one of the edges $e_{v,5}$ or $e_{v,6}$, then the cost of these shortest path are 56 and 62. Suppose now, that the two players play such a subgame. In this case, it is not difficult to verify that the subgame possesses no Nash equilibrium. Note that a game that is similar to the gadgets presented here can be found in [11].

	$e_{v,0}$		$e_{v,1}$		$e_{v,2}$		$e_{v,3}$		$e_{v,4}$		$e_{v,5}$		$e_{v,6}$	
congestion	1	2	1	2	1	2	1	2	1	2	1	2	1	2
player 1	0	M	20	28	45	45	48	48	20	30	16	16	65	65
player 2	0	M	14	18	45	45	48	48	20	30	M	M	M	M

Fig. 2. The player-specific delay functions of the edges e_v, i

Suppose now, that we are given two node-disjoint paths P_1 and P_2 in G connecting s_1 and t_1, and s_2 and t_2. We map these paths to paths in G_Γ in the natural way, and choose player-specific shortest paths through every gadget. Let $n(P_i)$ be the number of nodes on the path P_i. Thus, player 1 has delay $56 \cdot n(P_1)$, and player 2 has delay $62 \cdot n(P_2)$. If one of the players had an incentive to change her strategy, then she will only choose a path in which she shares no gadget with the other player, as otherwise her delay would increase to at least M. This is true as in this case the players would share at least one edge $e_{v,0}$. This also implies that the delay of the other player does not increase due to the strategy

change of the first player. Observe that this holds for any further best response. Thus, the players converge to an equilibrium after $O(n)$ best responses as the delay of a player decreases by at least the cost of the shortest path through a gadget.

Suppose now, that we are given a Nash equilibrium of Γ. In this case the players do not share a gadget as otherwise the state is no Nash equilibrium. □

Theorem 2. *One can decide in polynomial time whether a player-specific network congestion game Γ with a constant number of resources possesses a Nash equilibrium.*

In order to prove Theorem 2, we generalize an algorithm introduced by Chakrabarty et al. [4] that computes a social optimal state of a player-specific network congestion game with a constant number of parallel links. Details of this approach can be found in a full version of this paper.

3 Restricted Network Congestion Games

In this section, we analyze the complexity of computing Nash equilibria of restricted network congestion games with a constant number of players or resources.

Theorem 3. *Computing a Nash equilibrium of a restricted network congestion games with k players is* PLS*-complete for any $k \geq 3$.*

Proof. We prove the theorem by a reduction from the local search problem *positive not-all-equal 2-satisfiability* PosNae2Sat which is known to be PLS-complete [14]. Let $x_1, \ldots, x_n$ be boolean variables. An instance φ of PosNae2Sat consists of a set of m weighted clauses C_j over the variables x_i which contain two positive literals each. We denote by w_j the (integer) weight of clause C_j. A clause is satisfied if and only if the two variables it contains have different values. By $\bar{X} = (X_1, \ldots, X_n) \in \{0,1\}^n$ we denote a bit assignment to the variables $x_1, \ldots, x_n$. The weight $w(\bar{X})$ of a bit assignment $\bar{X}$ is defined as the sum of the weights of all satisfied clauses. We denote the maximum weight by $W = \sum_{j=1}^{m} w_j$. By $\bar{X}_{X_i=b}$, we denote the bit vector $(X_1, \ldots, X_{i-1}, b, X_{i+1}, \ldots, X_n)$. A local optimum of φ is a bit assignment $\bar{X}$ whose weight cannot be increased by flipping a single variable x_i, i.e., $w(\bar{X}) \geq w(\bar{X}_{x_i=b})$ for all $1 \leq i \leq n$ and $b \in \{0,1\}$. Therefore, the neighborhood of an assignment is defined as the set of assignments with Hamming distance one.

Given an instance φ, we construct a restricted network congestion game Γ_φ such that one can easily construct a local optimum of φ given a Nash equilibrium of Γ_φ. Γ_φ simulates two copies of φ, which we call φ_A and φ_B, in parallel. Furthermore, the game consists of three players, a *bit player* and two *clause players*.

Every path the bit player can choose determines assignments $\bar{X}_A$ and $\bar{X}_B$ for φ_A and φ_B, respectively. The set of paths the bit player can choose from can be

divided into two disjoint sets $\mathcal{P}_1$ and $\mathcal{P}_2$. If she chooses a path from $\mathcal{P}_1$, $\bar{X}_A$ is the actual assignment for φ and $\bar{X}_B$ is a (probably better) neighboring assignment. For every path in $\mathcal{P}_2$ it is the other way round. The bit player switches between paths in $\mathcal{P}_1$ and $\mathcal{P}_2$ as long as she can switch to a better neighboring assignment.

The paths of the clause players lead through $2m$ gadgets. For both copies of φ there is one gadget for every clause. The two clause players simulate a clause by choosing paths through the corresponding gadget. We ensure that they always have an incentive to correctly simulate the clauses according to the assignments determined by the bit player.

To implement this, we introduce four levels of delays: large, medium, small, and tiny. If the bit player is on a path in $\mathcal{P}_1$ ($\mathcal{P}_2$) and the clause players do not correctly simulate the clauses of φ_A (φ_B) according to the assignment $\bar{X}_A$ ($\bar{X}_B$), at least one of them has large delay. If the bit player is on a path in $\mathcal{P}_1$ ($\mathcal{P}_2$) and the clause players simulate φ_A (φ_B) correctly, she observes medium delay proportional to the weight of the unsatisfied clauses according to the actual assignment $\bar{X}_A$ ($\bar{X}_B$). Furthermore, she has additionally small delay that is proportional to the weight of the unsatisfied clauses of the neighboring assignment $\bar{X}_B$ ($\bar{X}_A$). If the bit player is on a path in $\mathcal{P}_1$ ($\mathcal{P}_2$) and the clause players do not correctly simulate φ_B (φ_A), they additionally have tiny delays. This ensures that the clause players have an incentive to correctly simulate the clauses and that the bit player has an incentive to choose the best neighboring assignment.

As long as there is a better neighboring assignment, the bit player can change from a path from $\mathcal{P}_1$ ($\mathcal{P}_2$) to a path from $\mathcal{P}_2$ ($\mathcal{P}_1$) by adopting the neighboring assignment as the actual assignment and by choosing a new neighboring assignment.

We are now ready to describe our construction in detail. We present the network of Γ_φ as two subnetworks. One subnetwork contains the edges the bit player is allowed to choose, the other subnetwork contains the edges the two clause players are allowed to choose. The edges that are contained in both networks are called *connection edges*. The connection edges are almost the only edges that cause delay to the players. Almost all other edges have delay 0 regardless of the number of players using it. To further simplify the presentation, we merge path segments into sets of edges and use dashed edges to indicate these path segments in Figure 3 and 4. The precise network can be constructed by concatenating the edges from a set in arbitrary order while adding an edge that is not contained in the other subnetwork between every pair of consecutive edges with constant delay 0.

The subnetwork of the bit player is depicted in Figure 3. We now define the corresponding sets of edges and the delays on the edges. Let $M \gg \alpha W \gg \alpha \gg \beta W \gg \beta \geq 4m$.

- $\mathcal{P}^A_{x_i=b} := \{u^A_{j,x_i=b}, t^B_{j,x_i=b} |$ for all clauses C_j with $x_i \in C_j\}$. Such a path segment corresponds to the fact that bit $x_i = b$ in the assignment $\bar{X}_A$. It also corresponds to the fact that $x_i = b$ in the assignment $\bar{X}_B$, unless the bit player chooses to flip this bit (see below). The u-edges have delay 0 for one player and delay M for two or more players. They induce large delay to clause players if they do not correctly simulate this bit assignment φ_A. The

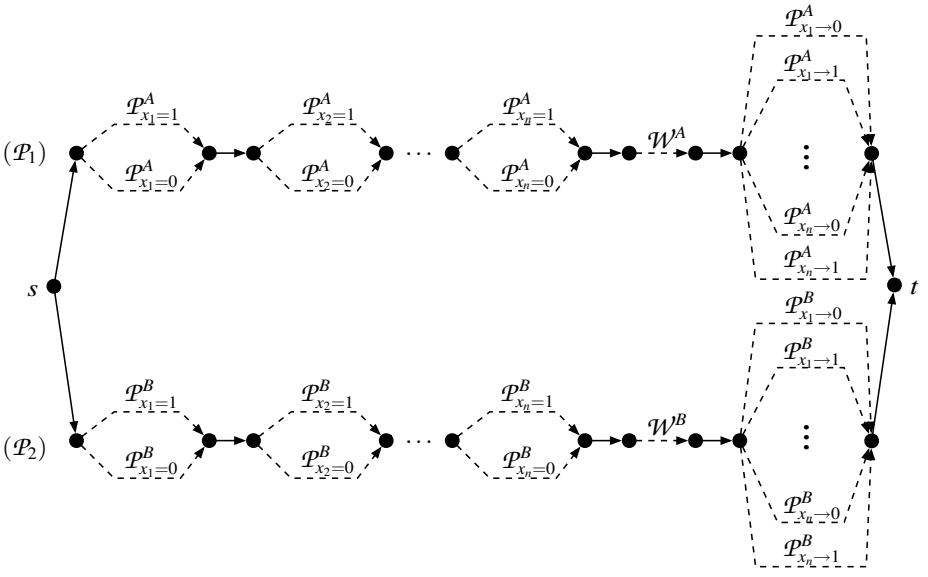

Fig. 3. The subnetwork of the bit player. The dashed edges correspond to set of edges.

t-edges have delay 0 for one player and delay 1 for two or more players. They induce tiny delay to the clause players if they do not correctly simulate the bit assignment φ_B.

- $\mathcal{W}^A := \{w^A_{j,0}, w^A_{j,1} | \text{ for all } 1 \leq j \leq m\}$. If the clause players correctly simulate φ_A, this path segment induces medium delay proportional to the weight of the unsatisfied clauses of $\bar{X}_A$ to the bit player. The edges $w^A_{j,0}$ and $w^A_{j,1}$ have delay 0 for one or two players and delay αw_j for three players.
- $\mathcal{P}^A_{x_i\to b} := \{w^A_{j,0,x_i\to b}, w^A_{j,1,x_i\to b} | \text{ for all } 1 \leq j \leq m \text{ with } x_i \notin C_j\} \cup \{t^B_{j,x_i\to b}, w^A_{j,x_i\to b} | \text{ for all } 1 \leq j \leq m \text{ with } x_i \in C_j\}$. If the bit player chooses such a path segment, then she determines the neighboring assignment $\bar{X}_B$ to be obtained from $\bar{X}_A$ by flipping bit x_i to b. If the clause players correctly simulate φ_A, this path segment induces small delay proportional to the weight of the unsatisfied clauses of that neighboring assignment. For each $1 \leq j \leq m$ with $x_i \notin C_j$, the edges $w^A_{j,0,x_i\to b}$ and $w^A_{j,1,x_i\to b}$ have delay 0 for at most two players and delay βw_j for three. For each $1 \leq j \leq m$ with $x_i \in C_j$, the edge $w^A_{j,x_i\to b}$ has delay 0 for one player and delay βw_j for two or more players. The t-edges have delay 0 for one player and delay 2 for two or more players. They induce tiny delay to the clause players if they do not simulate this bit flip in φ_B.

Additionally, there are sets $\mathcal{P}^B_{x_i=b}$, $\mathcal{W}^B$, and $\mathcal{P}^B_{x_i\to b}$ which are defined in the same manner.

The two clause players are symmetric in the sense that they play on the same subnetwork and have the same source and target node. Their subnetwork is a concatenation of m A-gadgets and m B-gadgets. Figure 4 depicts such a pair of

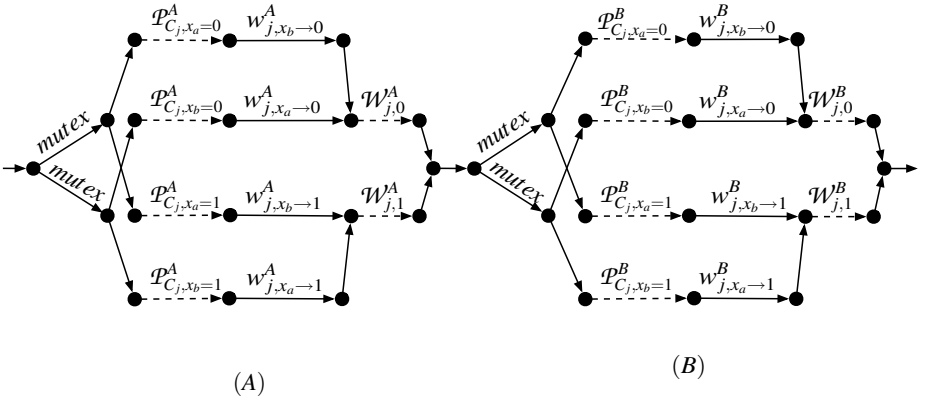

Fig. 4. This figure shows an A-gadget and a B-gadget for a clause $C_j = \{x_a, x_b\}$. There are four paths through each gadget. From top to bottom, we denote the paths with $x_a = 0$, $x_b = 0$, $x_a = 1$, and $x_b = 1$. The subnetwork of the two clause players is a concatenations of the A- and B-gadgets for all clauses.

gadgets. Their source-sink paths lead through all $2m$ gadgets. The edges labeled with *mutex* have delay 0 for one player and delay M^2 for two or more players. The dashed edges correspond to the following sets of connection edges:

- $\mathcal{P}^A_{C_j,x_i=b} := \{u^A_{j,x_i=1-b}, t^A_{j,x_i=1-b}, t^A_{j,x_i\to 1-b}\}$. A clause player using such a path segment simulates the assignment of b to x_i of $\bar{X}_A$ in the clause C_j of φ_A. In the following, we say she sets $x_i = b$ in this gadget. If this is not a correct simulation and the bit player is on a path from $\mathcal{P}_1$, then a u-edge induces large delay. If this is not a correct simulation and the bit player is on a path from $\mathcal{P}_2$, then a t-edge induces tiny delay.
- For each $d \in \{0,1\}$, $\mathcal{W}^A_{j,d} := \{w^A_{j,d}\} \cup \{w^A_{j,d,x_i\to b} |$ for all $b \in \{0,1\}$ and $1 \leq i \leq n$ with $x_i \notin C_j\}$. If and only if both players use the same $\mathcal{W}^A_{j,d}$ path segment, they simulate an unsatisfying assignment for C_j. If, additionally, the bit player chooses a path from $\mathcal{P}_1$, the edge $w^A_{j,d}$ has medium delay proportional to w_j. Furthermore, one of the edges $w^A_{j,d,x_i\to b}$ induces small delay if x_i is not in clause C_j. Note, that in the case that x_i is in the clause C_j, there are extra edges in the gadget.

The sets $\mathcal{P}^B_{C_j,x_i=b}$ and $\mathcal{W}^B_{j,d}$ are defined analogously.

We now prove that every Nash equilibrium of Γ_φ corresponds to a locally optimal assignment of φ. Consider a Nash equilibrium of Γ_φ and assume that the bit player chooses a path from the set $\mathcal{P}_1$. Let $\mathcal{P}^A_{x_1=X_1}, \ldots, \mathcal{P}^A_{x_n=X_n}, \mathcal{W}^A$, and $\mathcal{P}^A_{x_{i^*}\to b}$ be the path segments she chooses. Then the following properties hold.

Lemma 4.

a) In every A-gadget for every clause $C_j = \{x_a, x_b\}$ one clause player sets $x_a = X_a$ and the other player sets $x_b = X_b$.

b) In every B-gadget for every clause $C_j = \{x_a, x_b\}$ with $a, b \neq i^*$ one clause player sets $x_a = X_a$ and the other player sets $x_b = X_b$.
c) In every B-gadget for every clause $C_j = \{x_{i^*}, x_c\}$ one clause player sets $x_c = X_c$ and the other player sets $x_{i^*} = b$.

Proof. Observe that in any gadget for any clause $C_j = \{x_a, x_b\}$ one of the clause players chooses $x_a = 0$ or $x_a = 1$ whereas the other player chooses $x_b = 0$ or $x_b = 1$. Otherwise both have delay M^2 and, thus an incentive to change.

a) Consider the A-gadget of a clause $C_j = \{x_a, x_b\}$. Due to our assumptions, all edges of the path segment $\mathcal{P}^A_{C_j, x_a = X_a}$ are not used by the bit player and therefore have delay 0 for a single clause player, whereas the edge $u^A_{j, x_a = X_a}$ that is contained in the path segment $\mathcal{P}^A_{C_j, x_a = (1 - X_a)}$ is used by the bit player and therefore causes delay M to a clause player. The same is true for the path segments $\mathcal{P}^A_{C_j, x_b = X_b}$ and $\mathcal{P}^A_{C_j, x_b = (1 - X_b)}$, respectively. The delay of all other edges in the gadget sums up to less than M. Thus, in every Nash equilibrium, one of the clause players chooses $x_a = X_a$ and the other player chooses $x_b = X_b$.
b) In the B-gadgets all w^B-edges and all edges in the $\mathcal{W}^B$-sets are not used by the bit player and therefore have delay 0. Consider the B-gadget for a clause $C_j = \{x_a, x_b\}$ with $a, b \neq i^*$. All edges of the path segment $\mathcal{P}^B_{C_j, x_a = X_a}$ are not used by the bit player and therefore have delay 0 for a single clause player, whereas the edge $t^B_{j, x_a = X_a}$ that is contained in the path segment $\mathcal{P}^B_{C_j, x_a = (1 - X_a)}$ is used by the bit player and therefore has delay 1 for a clause player. The same is true for the path segments $\mathcal{P}^B_{C_j, x_b = X_b}$ and $\mathcal{P}^B_{C_j, x_b = (1 - X_b)}$, respectively.
c) Let $C_j = \{x_{i^*}, x_c\}$ be a clause that contains x_{i^*}. In the B-gadgets of clause C_j one clause player sets $x_c = X_c$ which has delay 0. The other clause player sets $x_{i^*} = b$ which has delay of at most 1. The path $x_{i^*} = 1 - b$ has delay of at least 2 due to the edge $t^B_{j, x_i \to b}$ which is currently used by the bit player.

□

Note that an equivalent version of Lemma 4 holds for Nash equilibria in which the bit player chooses a path from the set $\mathcal{P}_2$. The following corollary follows directly from Lemma 4.

Corollary 5. *In every Nash equilibrium the path segment $\mathcal{W}^A$ has delay $\alpha(W - w(\bar{X}))$ for the bit player. Furthermore, the delay on the path segment $\mathcal{P}^A_{x_{i^*} \to b}$ equals $\beta(W - w(\bar{X}_{x_{i^*} = b}))$ plus an additive term of at most $2m$ for the bit player.*

Lemma 6. *Every Nash equilibrium of Γ_φ corresponds to a local optimum of φ.*

Proof. For the purpose of contradiction, consider a Nash equilibrium that does not correspond to a local optimum of φ. Let $\mathcal{P}^A_{x_1 = X_1}, \ldots, \mathcal{P}^A_{x_n = X_n}$, $\mathcal{W}^A$, and $\mathcal{P}^A_{x_{i^*} \to b}$ be the path segments used by the bit player. By Corollary 5, we can conclude that $\bar{X}_{X_{i^*} = b}$ is the best neighboring assignment, otherwise the path segment $\mathcal{P}^A_{x_{i^*} \to b}$ has more delay then another path segment $\mathcal{P}^A_{x_{i^{**}} \to b^{**}}$ for the bit player. We show that this implies that the bit player can improve her delay

by choosing another path. The delays of all edges in the set $\mathcal{W}^A$ sum up to $\alpha(W - w(\bar{X}))$. Thus, the bit player has at least this amount of delay.

Now, observe that each path segment $\mathcal{P}^B_{x_i=X_1}$ with $i \neq i^*$ has delay 0 for the bit player since the clause players correctly simulate φ_B with the assignment $\bar{X}_{X_{i^*}=b}$. The path segment $\mathcal{P}^B_{x_{i^*}=b}$ has delay of at most m. The delays of all edges in the set $\mathcal{W}^B$ sum up to $\alpha(W - w(\bar{X}_{x_{i^*}=b}))$. The delay of any path $\mathcal{P}^B_{x_{i'}\to b'}$ is at most $\beta W + 2m$. Note that $\beta W + 3m < \alpha$. Thus, the bit player could decrease her delay by changing to such a path. This is a contradiction to the assumption that this is a Nash equilibrium. □

We conclude that every Nash equilibrium of G_φ corresponds to a locally optimal assignment of φ. Obviously, the construction of G_φ and the mapping of an equilibrium to a assignment of φ can be done in polynomial time. This conclude the proof of Theorem 3 □

It is an interesting open problem whether computing Nash equilibria for restricted network congestion games with two players remains PLS-complete. Moreover, it is an challenging open problem to prove any results in standard congestion games with a constant number of players.

Theorem 7. *One can compute a Nash equilibrium of a restricted network congestion game Γ with a constant number of resources in polynomial time.*

Theorem 7 is a consequence of the simple observation that there are only polynomial many different possible values for Rosenthal's potential functions. Again, details can be found in a full version of this paper.

Acknowledgment

The authors wish to thank Matthias Englert and Berthold Vöcking for helpful discussions.

References

1. Ackermann, H., Röglin, H., Vöcking, B.: On the impact of combinatorial structure on congestion games. In: Proc. of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS), pp. 613–622 (2006)
2. Ackermann, H., Röglin, H., Vöcking, B.: Pure Nash equilibria in player-specific and weighted congestion games. In: Spirakis, P.G., Mavronicolas, M., Kontogiannis, S.C. (eds.) WINE 2006. LNCS, vol. 4286, pp. 50–61. Springer, Heidelberg (2006)
3. Anshelevich, E., Dasgupta, A., Kleinberg, J.M., Tardos, É., Wexler, T., Roughgarden, T.: The price of stability for network design with fair cost allocation. In: Proc. of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS), pp. 295–304 (2004)
4. Chakrabarty, D., Mehta, A., Nagarajan, V.: Fairness and optimality in congestion games. In: Proc. of the 6th ACM conference on Electronic Commerce (EC), pp. 52–57. ACM Press, New York (2005)

5. Dunkel, J., Schulz, A.S.: On the complexity of pure-strategy Nash equilibria in congestion and local-effect games. In: Spirakis, P.G., Mavronicolas, M., Kontogiannis, S.C. (eds.) WINE 2006. LNCS, vol. 4286, pp. 62–73. Springer, Heidelberg (2006)
6. Fabrikant, A., Papadimitriou, C., Talwar, K.: The complexity of pure Nash equilibria. In: Proc. of the 36th Annual ACM Symposium on Theory of Computing (STOC), pp. 604–612 (2004)
7. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York (1979)
8. Johnson, D.S., Papadimtriou, C.H., Yannakakis, M.: How easy is local search? Journal on Computer and System Sciences 37(1), 79–100 (1988)
9. Meyers, C.: Network Flow Problems and Congestion Games: Complexitiy and Approximation. Massachusetts Institute of Technology (2006)
10. Milchtaich, I.: Congestion games with player-specific payoff functions. Games and Economic Behavior 13(1), 111–124 (1996)
11. Milchtaich, I.: The equilibrium existence problem in finite network congestion games. In: Spirakis, P.G., Mavronicolas, M., Kontogiannis, S.C. (eds.) WINE 2006. LNCS, vol. 4286, pp. 87–98. Springer, Heidelberg (2006)
12. Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V.: Algorithmic Game Theory. Cambridge University Press, Cambridge (forthcoming)
13. Rosenthal, R.W.: A class of games possessing pure-strategy Nash equilibria. Int. Journal of Game Theory 2, 65–67 (1973)
14. Schäffer, A.A., Yannakakis, M.: Simple local search problems that are hard to solve. SIAM Journal on Computing 20(1), 56–87 (1991)