Electronic Colloquium on Computational Complexity, Report No. 67 (2006)

On the Impact of Combinatorial Structure on Congestion Games *

Heiner Ackermann Heiko Röglin Berthold Vöcking

Department of Computer Science
RWTH Aachen, Germany
{ackermann,roeglin,voecking}@cs.rwth-aachen.de
April 12, 2006

Abstract

We study the impact of combinatorial structure in congestion games on the complexity of computing pure Nash equilibria and the convergence time of best response sequences. In particular, we investigate which properties of the strategy spaces of individual players ensure a polynomial convergence time. We show, if the strategy space of each player consists of the bases of a matroid over the set of resources, then the lengths of all best response sequences are polynomially bounded in the number of players and resources. We can also prove that this result is tight, that is, the matroid property is a necessary and sufficient condition on the players' strategy spaces for guaranteeing polynomial time convergence to a Nash equilibrium. In addition, we present an approach that enables us to devise hardness proofs for various kinds of combinatorial games, including first results about the hardness of market sharing games and congestion games for overlay network design. Our approach also yields a short proof for the PLS-completeness of network congestion games.

1 Introduction

Congestion games are a natural and generally accepted approach to model resource allocation among selfish and myopic players. In a congestion game we have a set of resources, and a strategy of a player corresponds to the selection of a subset of these resources. The strategy space is thus a set of sets of resources. The delay (cost, payoff) for each player from selecting a particular resource depends only on the number of players choosing that resource, and her total delay is the sum of the delays associated with her selected resources. Almost needless to say, congestion games are fundamental to routing, network design and other kinds of resource sharing in distributed systems.

Rosenthal [9] shows with a potential function argument that every congestion game possesses at least one pure Nash equilibrium. This argument does not only prove the existence of pure Nash equilibria but it also shows that such an equilibrium is reached in a natural way when players iteratively play best responses. A recent result of Fabrikant et al. [2] shows, however, that these best response sequences may require an exponential number of iterations. Their analysis relates congestion games to local search problems. They show that it is PLS-complete to compute a Nash equilibrium for general congestion games. Their completeness proof is based on a *tight* PLS-reduction preserving lower bounds on the length of best response sequences. Hence,

*This work was supported in part by the EU within the 6th Framework Programme under contract 001907 (DELIS) and by DFG grant Vo889/2-1.

ISSN 1433-8092

it follows from previous results about local search problems that there exist congestion games with initial configurations such that any best response sequence starting from these configurations needs an exponential number of iterations to reach a Nash equilibrium. Fabrikant et al. [2] are able to extend their negative results from general congestion games towards *network congestion games* in which each player aims at allocating a path in a network connecting a given source with a given destination node, provided that different players can have different source/destination pairs. The complexity changes if one assumes that all players have the same source/destination pair: For symmetric network congestion games, they present a polynomial time algorithm that computes a Nash equilibrium by solving a min-cost flow problem. This positive result leaves open, however, the question about the convergence time for best responses in symmetric network congestion games. As one of our results, we will see that, in contrast to the PLS-hardness results, the negative results for the convergence time of asymmetric network congestion games directly transfer to the symmetric case.

In this paper, we are interested in the question of which properties the combinatorial structure of a congestion game has to satisfy in order to guarantee that computing a Nash equilibrium has polynomial complexity and which properties ensure polynomial convergence time for best responses. In network congestion games, the strategy spaces of individual players have a very rich combinatorial structure: A best response requires to solve a shortest path problem. On the other extreme, we find *singleton games* in which all of the players' strategies consist only of single resources. Recently Ieong et al. [5] have shown that best response sequences for singleton games reach a Nash equilibrium after only a polynomial number of iterations. This result can be seen as a criterion on the strategy space of the players that guarantees fast convergence. In this paper, we systematically study how far such a sufficient criterion for fast convergence that is solely based on properties of the strategy spaces of individual players can go. More generally, taking into account also the global structure of the game, we investigate the question of what are the combinatorial properties that influence the complexity and the convergence time for structured congestion games.

1.1 Definitions and Notations

A *congestion game* Γ is a tuple $(\mathcal{N}, \mathcal{R}, (\Sigma_i)_{i\in\mathcal{N}}, (d_r)_{r\in\mathcal{R}})$ where $\mathcal{N} = \{1,\dots,n\}$ denotes the set of players, $\mathcal{R} = \{1,\dots,m\}$ the set of resources, $\Sigma_i \subseteq 2^{\mathcal{R}}$ the strategy space of player i, and $d_r : \mathbb{N} \to \mathbb{Z}$ a delay function associated with resource r. We call a congestion game *symmetric* if all players share the same set of strategies, otherwise we call it *asymmetric*. We denote by $S = (S_1,\dots,S_n)$ the *state of the game* where player i plays strategy $S_i \in \Sigma_i$. Furthermore, we denote by $S \oplus S_i'$ the state $S' = (S_1,\dots,S_{i-1},S_i',S_{i+1},\dots,S_n)$, i. e., the state S except that player i plays strategy S_i' instead of S_i. For a state S, we define the *congestion* $n_r(S)$ on resource r by $n_r(S) = |\{i \mid r \in S_i\}|$, that is, $n_r(S)$ is the number of players sharing resource r in state S. We assume that players act selfishly and like to play a strategy $S_i \in \Sigma_i$ that minimizes their individual delay. The delay $d_i(S)$ of player i is given by $d_i(S) = \sum_{r\in S_i} d_r(n_r(S))$. Given a state S, we call a strategy S_i^* a *best response* of player i to S if, for all $S_i' \in \Sigma_i$, $d_i(S \oplus S_i^*) \leq d_i(S \oplus S_i')$. In the following, we use the term *best response sequence* to denote a sequence of consecutive strategy changes in which each step is a best response which strictly decreases the delay of the corresponding player. Furthermore, we call a state S a *Nash equilibrium* if no player can decrease her delay by changing her strategy, i. e., for all $i \in N$ and for all $S_i' \in \Sigma_i$, $d_i(S) \leq d_i(S \oplus S_i')$. Rosenthal [9] shows that every congestion games possesses at least one Nash equilibrium by considering the potential function $\phi : \Sigma_1 \times \cdots \times \Sigma_n \to \mathbb{Z}$ with $\phi(S) = \sum_{r\in\mathcal{R}} \sum_{i=1}^{n_r(S)} d_r(i)$.

The class PLS contains all *local search problems* with certain properties. 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 set of resources $\mathcal{R}$, a set of feasible solutions $\mathcal{F}(I) \subseteq 2^{\mathcal{R}}$, an objective function $f : \mathcal{F}(I) \to \mathbb{Z}$, 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. A neighbor S' of a solution S is strictly better if the objective $f(S')$ is larger/smaller in the case of a maximization/minimization problem. A local search problem Π belongs to PLS if the following polynomial time algorithms exist: (1) an algorithm A which computes for every instance $I \in \mathcal{I}_\Pi$ an initial feasible solution $S \in \mathcal{F}(I)$, (2) an algorithm B which computes for every instance $I \in \mathcal{I}$ and every feasible solution $S \in \mathcal{F}(I)$ the objective value $f(S)$, and (3) an algorithm C, which determines for every instance $I \in \mathcal{I}_\Pi$ 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. [6] 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 are polynomial-time computable functions f and g such that (1) f maps instances $I \in \mathcal{I}_{\Pi_1}$ to instances $f(I) \in \mathcal{I}_{\Pi_2}$, (2) g maps solutions S_2 of $f(I)$ to solutions S_1 of I, and (3) for all instances $I \in \mathcal{I}_{\Pi_1}$ if S_2 is a local optimum of instance $f(I)$, then g maps S_2 to a local optimum of I. Additionally, a local search problem Π in PLS is PLS-complete if every problem in PLS is PLS-reducible to Π.

1.2 New Results

Upper and lower bounds on the convergence time. We show that the analysis of Ieong et al. [5] can be generalized towards matroids, that is, if the set of strategies of each player consists of the bases of a matroid over the set of resources, then the lengths of all best response sequences are polynomially bounded in the number of players and resources. This result holds regardless of the global structure of the game and for any kind of delay functions. We can show that the result is tight on the basis of instances with non-decreasing, non-negative delays: Any condition on the players' strategy spaces that yields a subexponential bound on the lengths of all best response paths implies that the strategy spaces after removing dominated strategies (w. r. t. non-negative delays) are the bases of matroids. In other words, the matroid property is a necessary and sufficient condition on the players' strategy spaces for guaranteeing polynomial time convergence to a Nash equilibrium.

The obvious application of matroid congestion games are network design problems in which players compete for the edges of a graph in order to build a spanning tree [13]. There are quite a few more interesting applications as even simple matroid structures like uniform matroids that are rather uninteresting from an optimization point of view lead to rich combinatorial structures when various players with possibly different strategy spaces are involved. Illustrative examples based on uniform matroids are market sharing games with uniform market costs or scheduling games in which each player has to injectively allocate a given set of tasks (services) to a given set of machines (servers).

Our negative result for the non-matroid games does not have immediate implications for particular classes of structured congestion games as it is solely based on local properties of the players' strategy spaces and neglects the global structure. However, our proof technique can be transfered to various classes of games as it reveals a minimal substructure, so-called (1,2)-exchanges, that can be found in the strategy spaces of non-matroid congestion games. If a class of non-matroid games allows to interweave the individual strategy spaces in the right way, then one can construct exponentially long best response sequences in form of a counter with the (1,2)-exchanges or more general with (1,k)-exchanges as basic building blocks. This approach can be applied to various classes of congestion games even if the delay (payoff) functions are restricted like in the case of market sharing games. We obtain that market sharing games admit exponentially long best response paths, which answers an open question from [3, 7].

Symmetric network congestion games are the only known class of non-matroid congestion games for which a Nash equilibrium can be computed in polynomial time. We can show, however, by an embedding of asymmetric network games into particular starting configurations of symmetric network congestion games that symmetric network games do not only admit exponentially long best response paths but that there are initial configurations such that all best response sequences starting from these configurations have exponential length.

Hardness results for structured congestion games. The only known hardness result for a class of structured congestion games is the PLS-completeness result for network congestion games with directed edges by Fabrikant et al. [2]. Unfortunately, the analysis in [2] is not very instructive, as it completely reworks the very involved master reduction to POSNAE3FFLIP (not-all-equal-3SAT with weights on clauses with only positive literals) from [10] and adds some further complications. (According to [2] already the master reduction from [10] is possibly the most complex reduction in the literature, if one excludes PCP.) We present an alternative approach for proving hardness of structured congestion games that more directly reveals which kind of substructures cause the trouble, and that also shows the hardness of asymmetric network congestion games with undirected edges. There is a simple, elegant reduction from POSNAE3FFLIP to MAXCUT (which is equivalent to POSNAE2FFLIP) [10]. We show that MAXCUT can be reduced to so-called *threshold games*. The strategy space of each player in a threshold game corresponds to an (1,k)-exchange. Despite this simple structure, threshold games are a natural and interesting class of games. Our main interest, however, stems from the fact, that because of this simple structure, threshold games are a good starting point for further PLS-reductions. We demonstrate the applicability of our approach by showing reductions from threshold games to three classes of games with different kinds of combinatorial structure:

- market sharing games with generalized payoff functions and polynomially bounded market costs,
- overlay network design, where players have to build a spanning tree on a given subset of nodes that are (virtually) completely connected on the basis of fixed routing paths in an underlying communication network, and
- network congestion games with (un)directed edges.

Taking the PLS-completeness of POSNAE3FFLIP for granted, the latter of these results yields a significantly simpler proof for the PLS-completeness of network congestion games than the one in [2]. The second result might seem as a contradiction to the positive result about matroid congestion games. However, despite the fact that players only have to solve a spanning tree problem, their strategy spaces do not form a matroid over the set of resources but over subsets (paths) of resources. This rather small deviation from the matroid property results in the PLS-completeness of this seemingly harmless class of congestion games.

Finally, let us remark that all considered PLS reductions are tight, so that they do not only prove the PLS-hardness of the considered classes of games but, in addition, they show that these classes contain instances of games with initial configurations for which all best response sequences have exponential length. Furthermore, this kind of reduction implies that it is PSPACE-hard to compute a reachable Nash equilibrium for a given initial configuration of these games.

2 Matroid Congestion Games

In this section, we consider matroid congestion games. Before we give a formal definition of such games we shortly introduce matroids. For a detailed discussion of matroids we refer the reader to [11].

Definition 1. *A tuple $M = (\mathcal{R}, \mathcal{I})$ is a matroid if $\mathcal{R} = \{1, \ldots, m\}$ is a finite set of resources and $\mathcal{I}$ is a nonempty family of subsets of $\mathcal{R}$ such that if $I \in \mathcal{I}$ and $J \subseteq I$, then $J \in \mathcal{I}$, and if $I, J \in \mathcal{I}$ and $|J| < |I|$, then there exists an $i \in I \backslash J$ with $J \cup \{i\} \in \mathcal{I}$.*

Let $I \subseteq \mathcal{R}$. If $I \in \mathcal{I}$, then we call I an *independent set* of $\mathcal{R}$, otherwise we call it *dependent*. It is well known that all maximal independent sets of $\mathcal{I}$ have the same size which is usually denoted by the *rank* $rk(M)$ of the matroid. A maximal independent set B is called a *basis* of M. In case of a weight function $w : \mathcal{R} \to \mathbb{N}$, we call a matroid *weighted*, and seek to find a basis of minimum weight where the weight of an independent set I is given by $w(I) = \sum_{r \in I} w(r)$. It is well known that such a basis can be found by a greedy algorithm. We are now ready to define matroid congestion games.

Definition 2. *We call a congestion game* $\Gamma = (\mathcal{N}, \mathcal{R}, (\Sigma_i)_{i\in\mathcal{N}}, (d_r)_{r\in\mathcal{R}})$ *a matroid congestion game if for every player* $i \in \mathcal{N}$*,* $M_i = (\mathcal{R}, \mathcal{I}_i)$ *with* $\mathcal{I}_i = \{I \subseteq S \mid S \in \Sigma_i\}$ *is a matroid. In this case,* Σ_i *is the set of bases of* M_i*. Additionally, we denote by* $rk(\Gamma) = \max_{i\in\mathcal{N}} rk(M_i)$ *the* rank of a matroid congestion game Γ.

2.1 Fast Convergence for Matroid Congestion Games

Ieong at al. [5] show that in singleton games players reach a Nash equilibrium after at most $n^2 \cdot m$ best responses. Note that singleton games are matroid congestion games with $rk(M_i) = 1$ for every player i. We extend their analysis to general matroid congestion games.

Theorem 3. *Let* Γ *be a matroid congestion game. Then players reach a Nash equilibrium after at most* $n^2 \cdot m \cdot rk(\Gamma)$ *best responses. In the case of identical delay functions, players reach a Nash equilibrium after at most* $n^2 \cdot rk(\Gamma)$ *best responses.*

Proof. Consider a list of all delays $d_r(n_r)$ with $r \in \mathcal{R}$ and $1 \leq n_r \leq n$ and assume that this list is sorted in a non-decreasing way. For each resource r, we define an alternative delay function $\tilde{d}_r : \mathbb{N} \to \mathbb{N}$ where, for each possible congestion n_r, $\tilde{d}_r(n_r)$ equals the rank of the delay $d_r(n_r)$ in the aforementioned list of all delays. We assume that equal delays receive the same rank.

Lemma 4. *Let* S *be a state of a matroid congestion game and* $S_i^* \in \Sigma_i$ *a best response of player* i *to* S *w. r. t. delays* d_r *which strictly decreases the delay of* i*. Then* S_i^* *strictly decreases the delay of player* i *w. r. t. the delays* $\tilde{d}_r$*.*

Proof. Schrijver [11] introduces the bipartite graph $G(S_i^* \Delta S_i) = (V, E)$ with $V = (S_i^* \backslash S_i) \cup (S_i \backslash S_i^*)$ and $E = \{\{r^*, r\} \mid r^* \in S_i^* \backslash S_i, r \in S_i \backslash S_i^* : S_i^* \cup \{r\} \backslash \{r^*\} \in \Sigma_i\}$ and shows that there exists a perfect matching P_M in $G(S_i^* \Delta S_i)$. Now observe that for every edge $\{r, r^*\} \in P_M$, $d_{r^*}(n_{r^*}(S^*)) \leq d_r(n_r(S^*) + 1)$ since, otherwise, S_i^* is no best response w. r. t. the delays d_r. Additionally, there exists at least one edge with $d_{r^*}(n_{r^*}(S^*)) < d_r(n_r(S^*) + 1)$ since S_i^* strictly decreases the delay of player i. Finally, the same inequalities also hold for the delays $\tilde{d}_r$, as they correspond to the ranks of the original delays. Thus the claim follows. □

Now due to Lemma 4, whenever a player plays a best response w. r. t. the delays d_r, Rosenthal's potential decreases w. r. t. the delays $\tilde{d}_r$. Now, since there are at most $n \cdot m$ different $d_r(n_r)$-values, $\tilde{d}_r(n_r) \leq n \cdot m$ for all resources $r \in \mathcal{R}$ and for all possible congestion values n_r. Hence,

$$\tilde{\phi}(S) = \sum_{r\in\mathcal{R}} \sum_{i=1}^{n_r(S)} \tilde{d}_r(i) \leq \sum_{r\in\mathcal{R}} \sum_{i=1}^{n_r(S)} n \cdot m = n^2 \cdot m \cdot rk(\Gamma)$$

since we sum over $n \cdot rk(\Gamma)$ values. Since $\tilde{\phi}(S)$ is lower bounded by 0 and decreases by at least one if a player plays a best response w. r. t. the delays d_r, the first part of the theorem follows. In the special case of identical delay functions, there are at most n different delays instead of $n \cdot m$, and thus the second part of the theorem follows. □

Note that Theorem 3 is independent of the delay functions. In particular, we do not assume monotonicity or that all delays have the same sign.

2.2 About the Complexity of Computing Socially Optimal Nash Equilibria in Spanning Tree Congestion Games

Theorem 3 states that a Nash equilibrium of a matroid congestion game can be found in polynomial time. A natural problem related to this is to consider the complexity of computing a socially optimal Nash equilibrium, i. e., a Nash equilibrium S with minimum social delay $\sum_{i=1}^{n} d_i(S)$. In general, computing such an equilibrium is NP-hard which can be shown by a reduction from the Hamiltonian Cycle problem to spanning tree congestion games.

Before we prove that, in general, computing a socially optimal Nash equilibrium of a spanning tree congestion game is NP-hard, we discuss several aspects of computing an arbitrary state S, i. e., not necessarily a Nash equilibrium, with minimum social delay $\gamma(S) = \sum_{i \in \mathcal{N}} d_i(S)$. Werneck et al. [13] consider classes of instances of the spanning tree congestion game with weakly convex delay functions d_r and show how to compute a socially optimal state efficiently. A delay function is weakly convex if $d_r(n_r+1) - d_r(n_r) \geq d_r(n_r) - d_r(n_r-1)$ for all $1 < n_r < n$. Note that in this case any socially optimal state is also a Nash equilibrium. Weak convexity implies monotonicity which is not satisfied in many applications. A more realistic scenario is to consider delay functions $d_r(n_r) = a_r \cdot (n_r - t_r)^{2d} + b_r$ for a threshold $t_r \in \mathbb{N}$, constants $a_r, b_r \in \mathbb{N}$ and an exponent $d \in \mathbb{N}$. Assume that, for each edge, players have to pay some money in order to activate the edge. These costs are shared among the players using an edge hence players prefer to share edges with other players. Additionally, assume that each edge has a limited bandwidth. If too many players use an edge, the edge is overloaded and thus less worth for the players. Unfortunately, for the aforementioned delay functions, computing a socially optimal solution is NP-hard.

Lemma 5. *For spanning tree congestion games Γ with n players and delay functions $d_r(n_r) = (n_r - n + 1)^{2d}$ for every resource $r \in \mathcal{R}$ and $d \in \mathbb{N}$ computing a socially optimal sate is NP-hard.*

Proof. We prove the theorem by a reduction from the Hamiltonian Cycle problem HC. Given an instance $G = (V, E)$ of HC we like to decide whether G contains a Hamiltonian cycle. Without loss of generality assume that G is connected. From this instance, we construct an instance of the spanning tree congestion game by setting $n = |V|$, $\mathcal{R} = E$, $\Sigma_i = \{T \mid T \text{ is a spanning tree of } G\}$ and $d_r(n_r) = (n_r - n + 1)^{2d}$ for an arbitrary $d \in \mathbb{N}$.

First observe that if G contains a Hamiltonian cycle then there is a state S of the game with $\gamma(S) = 0$; each player removes an individual edge from the cycle in order to receive a tree. Thus, the congestion on each edge is either $n - 1$ or 0.

Second let S be a state of the spanning tree congestion game with $\gamma(S) = 0$. Obviously n_r is either $n - 1$ or 0. Now consider the subgraph $G' = (V', E')$ of G which only contains edges with $n_r = n - 1$. Observe that $V' = V$, $|E'| = n$, and G' is connected. Furthermore, observe that G' is the union of a single spanning tree T and one extra edge r not contained in T. Note that $T + r$ contains a unique cycle. Now two cases can occur. Either all edges of G' form a single cycle or not. In the first case we have found a Hamiltonian cycle of G' and thus also of G. In the second case, observe that G' contains at least one node with degree 1. Thus, all n player have allocated the edge incident to this node, which is a contradiction to our construction of G'. □

Theorem 6. *For spanning tree congestion games, computing a Nash equilibrium with minimum social delay is NP-hard.*

Proof. Consider an instance of the spanning tree congestion game with delay functions $d_r(n_r) = (n_r - n + 1)^{2d}$ for every resource $r \in \mathcal{R}$ and $d \in \mathbb{N}$. Then observe that due to Lemma 5 there exists a state S with social delay $\gamma(S) = 0$ if and only if there exists a Hamiltonian cycle in the given graph. Obviously such a state in a Nash equilibrium, too, and thus the theorem follows. □

2.3 A Characterization of Games that Admit Fast Convergence

One can view congestion games as games in which every player solves an optimization problem over the set $\mathcal{R}$ of resources with the goal to find a feasible subset of $\mathcal{R}$ that minimizes her delay. Since resources are shared by different players, the delays observed by a player can change over time. In this section, we take a bottom-up view on congestion games, meaning that we first define the optimization problems of the players and after that, describe how the global structure of the game looks like, i. e., how the resources of the players are combined to a global set of resources and how the delays are chosen. To make this more formal, we assume that every player $i \in \mathcal{N}$ has a set of resources $\mathcal{R}_i$ and a set of strategies $\Sigma_i \subseteq 2^{\mathcal{R}_i}$. The pair $(\mathcal{R}_i, \Sigma_i)$ constitutes an instance of an optimization problem in the sense that for given delays on the resources $r \in \mathcal{R}_i$, the goal of player i is to find a strategy from Σ_i with minimum delay. Additionally, a global set $\mathcal{R}$ of resources and delay functions $d_r : \mathbb{N} \to \mathbb{Z}$ for $r \in \mathcal{R}$ are given. The local optimization problems of the players are combined into a global congestion game by specifying an injective mapping $f_i : \mathcal{R}_i \to \mathcal{R}$ for every player $i \in \mathcal{N}$. Now the delay player $i \in \mathcal{N}$ incurs in state $S = (S_1, \ldots, S_n) \in \Sigma_1 \times \cdots \times \Sigma_n$ is defined to be $\sum_{r' \in S_i} d_{f_i(r')}(n_{f_i(r')}(S))$ with $n_r(S) = |\{i \in N \mid \exists r' \in S_i : \; r = f_i(r')\}|$ for every $r \in \mathcal{R}$. We call a congestion game *freely configurable* if there are no restrictions on the functions f_i.

An *optimization problem* Π is specified by its set of instances, and each instance is defined by the number of resources m it contains and the set $\Sigma \subseteq 2^{[m]}$ of feasible solutions over the set of resources $[m] = \{1, \ldots, m\}$. For a given instance $([m], \Sigma)$ and a delay function $d : [m] \to \mathbb{Z}$, the goal is to find a solution from Σ with minimum total delay. We say that an optimization problem Π has a *matroid structure* if every instance $([m], \Sigma) \in \Pi$ is a matroid. An optimization problem which possesses an instance which is no matroid is called *non-matroid* problem. An optimization problem Π is called *inclusion-free* if it does not contain an instance $([m], \Sigma)$ with $X, Y \in \Sigma$ and $X \subsetneq Y$. Due to our bottom-up perspective on congestion games, every class $\mathcal{C}$ of optimization problems defines a class of congestion games simply by restricting the players to instances of optimization problems from $\mathcal{C}$.

We have shown that in freely configurable congestion games in which the optimization problem of every player has a matroid structure, every best response sequence has polynomial length. In this section, we show that for congestion games induced by inclusion-free optimization problems this result cannot be generalized to more general classes of optimization problems.

Theorem 7. *Let $\mathcal{C}$ be a class of inclusion-free optimization problems that contains a non-matroid problem. Then the corresponding class of freely configurable congestion games with non-decreasing, non-negative delays contains games with exponentially long best response sequences.*

If we assume that every delay is non-negative, we can remove the assumption that $\mathcal{C}$ is inclusion-free from Theorem 7. In this case, if there are two strategies X and Y for a player with $X \subsetneq Y$, then she never has an incentive to play Y. Hence, we can prune the strategy space by removing any proper superset from the set of strategies without changing the best response dynamics. This way, we obtain a class $\mathcal{C}'$ of inclusion-free optimization problems. If $\mathcal{C}'$ contains only matroid optimization problems, we can apply Theorem 3, otherwise, we can apply Theorem 7.

The key insight for proving Theorem 3 is that best responses in matroids can be decomposed into sequences of pairwise exchanges of resources such that each of these exchanges does not increase the delay of the corresponding player. In the following, we show that this 1-1-*exchange property* is not only sufficient but also necessary for fast convergence. Therefore, we first define its negation formally, prove that it is satisfied by every non-matroid, and show that it is sufficient to construct congestion games with exponentially long best response sequences.

Definition 8 (1-2-exchange property)**.** *Let $(\mathcal{R}, \Sigma)$ be an instance of an optimization problem. We say that $(\mathcal{R}, \Sigma)$ satisfies the* 1-2-exchange property *if we can identify three distinct elements $a, b, c \in \mathcal{R}$ with the prop-*

erty that for any given $k \in \mathbb{N}$, we can choose a delay $d(r)$ for every $r \in \mathcal{R} \setminus \{a, b, c\}$ such that for every choice of the delays of a, b, and c with $1 \leq d(a), d(b), d(c) \leq k$, the following property is satisfied: If $d(a) < d(b)+d(c)$, then for every set I from Σ with minimum delay, $a \in I$ and $b, c \notin I$. If $d(a) > d(b) + d(c)$, then for every set I from Σ with minimum delay, $a \notin I$ and $b, c \in I$.

Lemma 9. *Let $(\mathcal{R}, \Sigma)$ be an inclusion-free instance of an optimization problem. Let $\mathcal{I} = \{X \subseteq S \mid S \in \Sigma\}$, and assume that $(\mathcal{R}, \mathcal{I})$ is not a matroid, i. e., that Σ is not the set of bases of some matroid. Then $(\mathcal{R}, \Sigma)$ possesses the 1-2-exchange property.*

Proof. If $(\mathcal{R}, \mathcal{I})$ is no matroid, then there must be two sets $X, Y \in \Sigma$ and an element $x \in X \setminus Y$ such that for every $y \in Y \setminus X$, the set $X \setminus \{x\} \cup \{y\}$ is not contained in Σ. Let such sets X and Y and such an element $x \in X$ be given, let k be as in Definition 8, and assume w. l. o. g. that $k > |\mathcal{R}|$. Furthermore, we can assume w. l. o. g. that there does not exist a set $Y' \subseteq (X \cup Y) \setminus \{x\}$ with $Y' \neq Y$ and $Y' \in \Sigma$. Otherwise, we could replace Y by Y' without changing the aforementioned properties of x, X, and Y. Due to the choice of the sets X and Y and the element $x \in X$, Y cannot be of the form $Y = X \setminus \{x\} \cup \{y\}$. Hence, $Y = X \setminus \{x_1, \ldots, x_l\} \cup \{y_1, \ldots, y_{l'}\}$ with $x_1 = x$, $x_i \in X \setminus Y$, $y_i \in Y \setminus X$, $l \geq 1$, $l' \geq 1$, and $l + l' > 2$.

We first consider the case that $l' = 1$ and $l \geq 2$. In this case, we identify elements a, b, and c and define a delay function d with the properties needed to satisfy Definition 8. We set $a = y_1$, $b = x_1$, and $c = x_2$ and we define $d(r) = M = 4k$ for every $r \notin X \cup \{a\}$ and to be $d(r) = 0$ for every $r \in X \setminus \{b, c\}$. Let $1 \leq d(a), d(b), d(c) \leq k$ be chosen arbitrarily and let X^* denote the optimal solution w. r. t. the delays d. If X^* contains a, then it does not contain b or c as, otherwise, Y would have less delay than X^*. If X^* does not contain a, then it must contain b and c, as otherwise $X^* \subsetneq X$. Hence, if $d(a) < d(b) + d(c)$, then the unique solution with minimum delay is Y, and if $d(a) > d(b) + d(c)$, then the unique solution with minimum delay is X.

Finally, we have to consider the case $l' > 1$. In this case, we set $a = x_1$, $b = y_1$, and $c = y_2$. Furthermore, we set $d(r) = M$ for $r \notin X \cup Y$ and $d(r) = 0$ for $r \in (X \cup Y) \setminus \{a, b, c\}$. Let $1 \leq d(a), d(b), d(c) \leq k$ be chosen arbitrarily and let X^* denote the optimal solution w. r. t. the delays d. If X^* contains a, then it does not contain b or c as, otherwise, X would have less delay than X^*. If X^* does not contain a, then it must contain b and c as, otherwise, $X^* \subseteq (X \cup Y) \setminus \{x\}$ with $X^* \neq Y$ which contradicts the assumption made above. Hence, if $d(a) < d(b) + d(c)$, then the unique solution with minimum delay is X, and if $d(a) > d(b) + d(c)$, then the unique solution with minimum is Y. ☐

Now we show that in congestion games in which the local optimization problem of every player satisfies the 1-2-exchange property, myopic players do not reach a Nash equilibrium in polynomial time in general if there are no restrictions on the global structure of the congestion game. Observe that the following lemma together with Lemma 9 directly implies Theorem 7.

Lemma 10. *Assume that a set of players $\mathcal{N}$ and for each player $i \in \mathcal{N}$, a set of resources $\mathcal{R}_i$ and a set of strategies $\Sigma_i \subseteq 2^{\mathcal{R}_i}$ are given such that $(\mathcal{R}_i, \Sigma_i)$ possesses the 1-2-exchange property. Then we can define a global set of resources $\mathcal{R}$, a delay function $d_r : \mathbb{N} \to \mathbb{N}$ for every resource $r \in \mathcal{R}$, and an injective mapping $f_i : \mathcal{R}_i \to \mathcal{R}$ for every player $i \in \mathcal{N}$ such that the resulting congestion game contains an exponentially long best response sequence.*

Proof. A well known technique for constructing examples with exponentially long best response sequences is to construct instances that resemble the behavior of a binary counter. Haken constructs such a counter for threshold logic networks [4] (see also [8]). The global structure of our counter follows the structure of a counter presented by Anshelevich et al. for a network design game [1]. However, the gadgets that represent the bits of the counter are different as we do not need to embed them into a network but into a structure that only allows 1-2 exchanges. Due to Lemma 9, we can for every player $i \in \mathcal{N}$, identify three resources a_i, b_i, and c_i in her set of resources $\mathcal{R}_i$ with the properties as in Definition 8. These are the only resources of player i that she shares with other players. The other resources are exclusively used by her. We choose their delays in such a way that the 1-2-exchange property is satisfied for a_i, b_i, and c_i. Hence, to simplify matters, we can assume w. l. o. g. that every player i is interested in only three resources, namely a_i, b_i, and c_i, and that she is only allowed to play either the strategy $\{a_i\}$ or the strategy $\{b_i, c_i\}$. Additionally, the congestion game we construct contains players $i \in \mathcal{N}$ with only two resources a_i and b_i who either play the strategy $\{a_i\}$ or the strategy $\{b_i\}$. In order to achieve this, resource c_i is also not shared with other players.

Let $G_0, \ldots, G_{n-1}$ denote the gadgets, and assume that G_0 represents the least significant bit and that G_{n-1} represents the most significant bit. Every gadget has three main configurations, namely a 0-state, a 1-state and a reset state. If a gadget G_i is in state 0 and no gadget G_j with $j > i$ is in its reset state, then there exists a best response sequence such that gadget G_i first changes to its reset state and after that to state 1. If a gadget G_i is in state 1 and at least one gadget G_j with $j > i$ is in its reset state, then there exists a best response sequence in which gadget G_i changes its state to 0. One can easily see that these properties imply the existence of an exponentially long best response sequence starting in the state in which every gadget is in state 0 and eventually leading to the state in which every gadget is in state 1.

Now, we decribe the counter in more detail. Let $n = \Theta\left(\sqrt{|\mathcal{N}|}\right)$ and not equals the number of player as before, and $\alpha \in \mathbb{N}$ with $\alpha \geq 2$ be given. We construct a congestion game with n gadgets $G_0, \ldots, G_{n-1}$ representing a binary counter counting from 0 to $2^n - 1$. Each of these gadgets consists of $O(n)$ players and resources and represents one bit of the binary counter; G_0 represents the least and G_n the most significant bit. Each gadget contains two main players, the so-called *bit player* and the *reset player*. If these two players have chosen their strategies, then the best responses of the other players are uniquely determined. The only purpose of the additional players is to copy the decision of the *reset player*. We describe the state of a gadget by a pair of bits (x, y), meaning that the bit player plays her strategy x and that the reset player plays her strategy y. When describing the state of a gadget by such a pair, we assume that the other players have played their best responses according to strategy y. Let $i \in \{0, \ldots, n-1\}$ be fixed. If all reset players of the gadgets G_j with $j > i$ are on their 0-strategies, then there exists a sequence of best responses for gadget G_i starting in $(0, 0)$ and reaching $(0, 1)$, $(1, 1)$, and $(1, 0)$ in that order. If the reset player of at least one gadget G_j with $j > i$ is on her 1-strategy, then the 0-strategy of the bit player of gadget G_i is her best response. We say that the i-th bit is set if the bit player of gadget G_i plays her 1-strategy.

One can easily see that these properties suffice to resemble the behavior of a binary counter. In the initial state, every gadget G_i is in state $(0, 0)$. In this state, there exists a best response sequence for gadget G_0 leading to state $(1, 0)$, i. e., the least significant bit is set. After that, the players in the second gadget play their best responses, i. e., G_1 changes its state from $(0, 0)$ via $(0, 1)$ and $(1, 1)$ to $(1, 0)$. When G_1 is in state $(0, 1)$, gadget G_0 is reset again, i. e., its state is reset to $(0, 0)$. When G_1 is in state $(1, 0)$, gadget G_0 changes its state again from $(0, 0)$ to $(1, 0)$, i. e., the least significant bit is set again. After that, G_2 changes its state from $(0, 0)$ via $(0, 1)$ and $(1, 1)$ to $(1, 0)$. When G_2 is in state $(0, 1)$, the gadgets G_0 and G_1 are reset again, and so on.

Now we describe the gadgets in detail. Let $i \in \{0, \ldots, n-1\}$ be fixed and for $i > 0$, let $k = \lceil \log i \rceil + 1$. Gadget G_i contains the resources r_1^i, r_2^i, r_3^i and for $i > 0$, the resources $t_1^i, \ldots, t_{2^k-1}^i$. The bit player has two possible strategies, namely $\{r_1^i\}$ and $\{r_2^i\}$. The reset player decides between playing either $\{r_2^i\}$ or $\{r_3^i, t_1^i\}$. For every $j \in \{1, \ldots, 2^{k-1} - 1\}$, there is a player who decides between either playing $\{t_j^i\}$ or $\{t_{2j}^i, t_{2j+1}^i\}$. We refer to those players as *tree players* as they implicitly define a complete binary tree with height k (see Figure 1). Gadget G_i is connected to every gadget G_j with $j < i$ by a player who either plays $\{r_1^j\}$ or $\{t_{2^{k-1}+j-1}^i\}$. We call these players *connection players*. Let $k_i = 3(i+1) \log n$. We choose the delays of the resources as follows. There are $n - i$ players interested in resource r_1^i, namely $n - i - 1$ players from gadgets G_j with $j > i$ and the bit player of gadget G_i. If less than $n - i$ players are on resource r_1^i its delay is $\alpha^{k_i - 5}$. If exactly $n - i$ players are on resource r_1^i its delay is α^{k_i}. The delay of resource r_2^i is $\alpha^{k_i - 4}$ if one player is on that resource

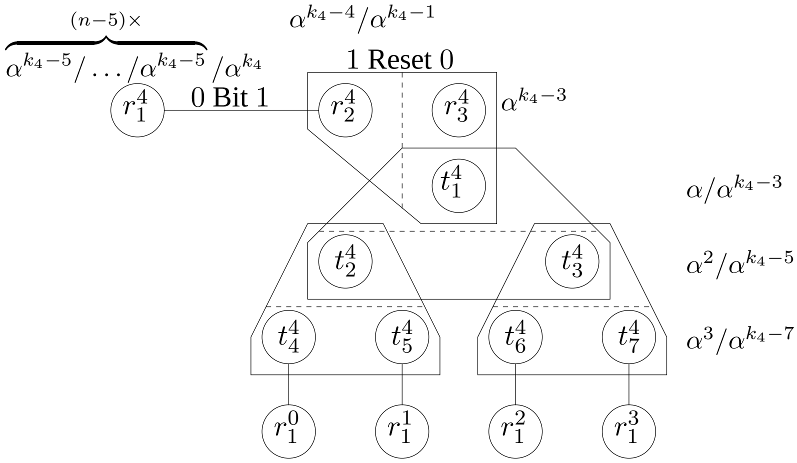

Figure 1: Illustration of gadget G_4. Resources are represented by circles, players who choose one of two possible resources are represented by lines connecting these resources, and players deciding between either taking one resource or two are represented by polygons.

alone and α^{k_i-1} if two players share that resource. The delay of r_3^i is α^{k_i-3}. Now we consider the resources at level $j \geq 0$ of the tree defined by the tree players. All resources at level j have delay α^{j+1} if they are allocated by one player alone and α^{k_i-3-2j} if they are shared by two players.

First of all, we show that the tree and the connection players of a gadget are solely controlled by the reset player of that gadget. Therefore, we consider the following two cases.

- The reset player is on her 0-strategy which is the strategy $\{r_3^i, t_1^i\}$. In this case the strategy $\{t_2^i, t_3^i\}$ is the best response of the first tree player as its delay is at most $2\alpha^{k_i-5}$ which is by at least a factor of α smaller than the delay on $\{t_1^i\}$ which is α^{k_i-3}. Now assume that the first tree player has played her best response. We can iteratively argue that the best responses of the second and the third tree player are then the strategies $\{t_4^i, t_5^i\}$ and $\{t_6^i, t_7^i\}$, respectively. These strategies guarantee a delay of at most $2\alpha^{k_i-9}$ while the strategies $\{t_4^i\}$ and $\{t_5^i\}$ each have a delay of α^{k_i-7}. Hence, allowing the tree players to play their best responses one after another results in a state in which every tree player is on her ,,down strategy" consisting of two resources. Observe that in this situation the best response of every connection player is to choose the strategy $\{r_1^j\}$ as this has delay at most $\alpha^{k_{i-1}}$ and the strategy $\{t^i_{2^{k-1}+j-1}\}$ has delay α^{k_i-3-2k}.

- The reset player is on her 1-strategy which is the strategy $\{r_2^i\}$. In this case the strategy $\{t_1^i\}$ is the best response of the first tree player as its delay is α and the delay of $\{t_2^i, t_3^i\}$ is at least $2\alpha^2$. Now assume that the first tree player has played her best response. We can iteratively argue that the best responses of the second and the third tree player are then the strategies $\{t_2^i\}$ and $\{t_3^i\}$, respectively. These strategies have a delay of α^2 while the strategies $\{t_4^i, t_5^i\}$ and $\{t_6^i, t_7^i\}$ have a delay of at least $2\alpha^3$. Hence, allowing the tree players to play their best responses one after another results in a state in which every tree player is on her ,,up strategy" consisting of a single resource. Observe that in this situation the best response of every connection player is to choose the strategy $\{t^i_{2^{k-1}+j-1}\}$ as this has a delay of α^{k+1} and the strategy $\{r_1^j\}$ has delay at least α^{k_0-5}.

In the following, we assume that immediately after each strategy change of the reset player, the tree and the connection players of the corresponding gadget also change their strategies appropriately. Hence, when we say that the reset player of gadget G_i is on her 0-strategy, we implicitly assume that all tree players of that gadget are on their down strategies and that all connection players are on their $\{r_1^j\}$ strategy. We study best response sequences of the bit and the reset players in more detail. Therefore, fix a gadget G_i.

- Assume that all reset players of the gadgets G_j with $j > i$ are on their 0-strategy and that gadget G_i is in state $(0, 0)$. In this case, the reset player can decrease her delay from $\alpha^{k_i-3} + \alpha$ to α^{k_i-4}. After that the gadget is in state $(0, 1)$, and the bit player can decrease her delay from α^{k_i} to α^{k_i-1}. After that the gadget is in state $(1, 1)$, and the reset player can again decrease her delay from α^{k_i-1} to $2\alpha^{k_i-3}$. After that the gadget is in state $(1, 0)$ and as long as no reset player of a gadget G_j with $j > i$ plays her 1-strategy it stays in this state.

- Assume that gadget G_i is in state $(1, 0)$ and that at least one reset player of a gadget G_j with $j > i$ is on her 1-strategy. In this case, the number of players currently on resource r_1^i is at most $n - i - 2$. Hence, $\{r_1^i\}$ is the best response of the bit player as it has delay α^{k_i-5} whereas $\{r_2^i\}$ has a delay of α^{k_i-4}.

Altogether this shows that the aforementioned sequence of strategy changes which results in counting from 0 to $2^n - 1$ is a best response sequence. Furthermore it is even a best response sequence for α-greedy players. ☐

Mirrokni [7] introduces the notion of *(1+ε)-greedy players*, i. e., players who only change their strategy when this decreases their current delay by at least a factor of $1 + \varepsilon$. In general, these players do not reach a Nash equilibrium but a state in which no player can improve her delay by a factor of $1 + \varepsilon$, a so-called *(1+ε)-approximate Nash equilibrium*. The counter constructed in Lemma 10 possesses the property that a player who decreases her delay decreases it by a factor of at least α, where $\alpha \geq 2$ can be specified arbitrarily. Hence, the example shows that not even $(1 + \varepsilon)$-greedy players reach an approximate equilibrium in polynomial time.

3 Threshold Congestion Games

Threshold congestion games as defined below are a special class of congestion games. Assume that the set of resources $\mathcal{R}$ is divided into two disjoint subsets $\mathcal{R}_{\text{in}}$ and $\mathcal{R}_{\text{out}}$ with $|\mathcal{R}_{\text{out}}| = n$. Additionally assume that each player has only two strategies, namely a strategy $S_i^{\text{out}} = \{r_i\}$ for a unique resource $r_i \in \mathcal{R}_{\text{out}}$, and a strategy $S_i^{\text{in}} \subseteq \mathcal{R}_{\text{in}}$. Furthermore assume that no two players are interested in the same resource $r \in \mathcal{R}_{\text{out}}$. In a given state S, strategy S_i^{in} is a best response for player i if $d_i(S \oplus S_i^{\text{in}}) \leq d_{r_i}(1)$. Thus, the delay $d_{r_i}(1)$ on resource r_i is a threshold indicating whether i plays strategy S_i^{in} or not, and thus interferes with other players or not. We denote by $T_i = d_{r_i}(1)$ the threshold of player i.

We now introduce a further restriction on threshold games which is helpful for showing PLS-completeness of other classes of congestion games. We call a threshold congestion game a *2-threshold congestion game* if for each resource $r \in \mathcal{R}_{\text{in}}$ there are at most two players i with $r \in S_i^{\text{in}}$.

Theorem 11. *The problem of finding a Nash equilibrium of a 2-threshold congestion game Γ is* PLS*-complete.*

Proof. We prove the theorem by a PLS-reduction from the MAXCUT problem for which finding a local optimum is known to be PLS-complete [10]. Let $G = (V, E, (w_e)_{e\in E})$ denote a weighted graph. The goal is to find a partition $V = V_1 \dot{\cup} V_2$ of the vertices into two disjoint sets V_1 and V_2 such that the value of the cut, i. e., the sum of the weights of the edges having one endpoint in V_1 and one in V_2, cannot be improved by moving a single vertex from V_1 to V_2 or vice versa. Additionally, we denote by w_v the sum of the weights of the edges incident to vertex v.

From G we construct a 2-threshold congestion game Γ as follows. For every edge $e \in E$, there is a resource $r_e \in \mathcal{R}_{\text{in}}$ with delay $d_{r_e}(1) = 0$ and $d_{r_e}(2) = w_e$. For every vertex $v \in V$ there is a resource $r_v \in \mathcal{R}_{\text{out}}$

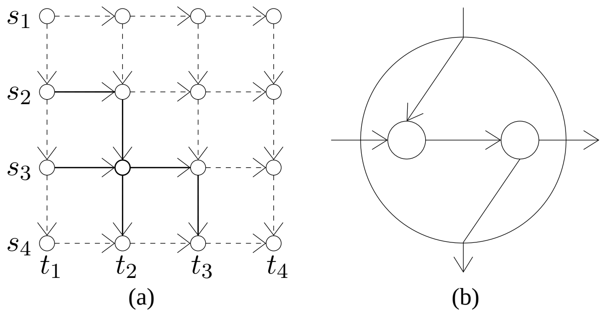

Figure 2: Illustration of the reduction in the proof of Theorem 12.

with delay $d_{r_v}(1) = w_v/2$. Additionally, for every vertex $v \in V$, there is a player i_v with $S^{\text{out}}_{i_v} = \{r_v\}$ and $S^{\text{in}}_{i_v} = \{r_e \mid e \text{ incident to } v\}$. Observe that this construction ensures that for every resource $r_e \in \mathcal{R}_{\text{in}}$, there are exactly 2 players i with $r \in S^{\text{in}}_i$. Thus, the constructed game is a 2-threshold congestion game.

Now let S be a Nash equilibrium of Γ. From S we construct a locally optimal partition of the vertices of the MAXCUT instance as follows. If player i_v plays strategy $S^{\text{out}}_{i_v}$, we put vertex v into V_1 otherwise into V_2. Observe that the total weight $w(v, V_2)$ of the edges incident to vertex v and to a vertex $u \in V_2$ in the constructed cut equals exactly the delay $d(S^{\text{in}}_{i_v})$ of the strategy $S^{\text{in}}_{i_v}$. Since S is a Nash equilibrium, if player i_v plays strategy $S^{\text{out}}_{i_v}$, then $d(S^{\text{in}}_{i_v}) \geq w_v/2$ and hence, in the constructed cut, $w(v, V_2) \geq w_v/2$. If player i_v plays strategy $S^{\text{in}}_{i_v}$, then $d(S^{\text{in}}_{i_v}) \leq w_v/2$ and hence, in the constructed cut, $w(v, V_2) \leq w_v/2$. Thus, the Nash equilibria of the 2-threshold congestion game coincide with the local optima of the MAXCUT instance. $\square$

The presented PLS-reduction is tight in the sense as defined by Schäffer and Yannakakis [10]. This means that there exist instances of 2-threshold games with states such that every best response sequence starting in such a state has exponential length. Furthermore, it implies that for two given states S and S^* it is PSPACE-complete to decide if S^* can be reached from S by a sequence of best responses.

4 Network Congestion Games

In this section, we present a proof that finding Nash equilibria in asymmetric network congestion games is PLS-complete. In the case of directed networks this has been proven before by Fabrikant, Papadimitriou, and Talwar, but we present a simplified proof which already contains the idea of how to prove PLS-completeness in the case of undirected networks.

Theorem 12 ([2])**.** *The problem of finding a Nash equilibrium of an asymmetric network congestion game* Γ *with directed edges is* PLS*-complete.*

Proof. We give a PLS-reduction from 2-threshold congestion games to asymmetric network congestion games. Let Γ be a 2-threshold congestion game and assume w. l. o. g. that for each pair of players $i, j \in \mathcal{N}$, there exists a unique resource $r_{i,j}$ that is contained in S^{in}_i and S^{in}_j. We can transform an arbitrary 2-threshold congestion game into this form by adding dummy resources with delay 0 and by combining a set of resources shared by two players into one resource whose delay equals the sum of the delays of these resources.

The directed graph $G = (V, E)$ that we construct is an n times n grid in which edges are directed downwards and from left to right. The source nodes of the players are the nodes in the first column; s_1 is the topmost node, s_2 the node in the second row, and so on. The target nodes are the nodes in the last row; t_1 is the leftmost node, t_2 the node in the second column, and so on. This construction is depicted in Figure 4 (a). For every player $i \in \mathcal{N}$, we denote by T_i her threshold, and we additionally add an edge from s_i to t_i. Observe that due to the directions of the edges in the grid, this edge can only be used by player i. Edges pointing downwards have always delay 0, and every edge from left to right in row i has always delay $i \cdot D$, where D denotes the sum of all delay values in the given 2-threshold congestion game. Furthermore, we associate with every node in the grid a delay function which can be accomplished by replacing each node by two nodes as depicted in Figure 4 (b). The delay function of the node $v_{i,j}$ in row i and column j equals the delay function of the resource $r_{i,j}$. Hence, if player i decides to route her traffic through the grid instead of using the direct edge from s_i to t_i, then she always uses the path $s_i = v_{i,1}, \ldots, v_{i,i}, v_{i+1,i}, \ldots, v_{n,i} = t_i$ as every other path has higher delay. Let D_i denote the delay on this path caused by the edges of the grid, i. e., $D_i = (i-1)i \cdot D$. We set the delay of the edge (s_i, t_i) to $D_i + T_i$.

We transform a Nash equilibrium of the constructed network congestion game into a state of the 2-threshold game in the obvious way by setting exactly those players i to their strategy S_i^{out} who use the edge (s_i, t_i). In order to see that the state of the 2-threshold game obtained this way is a Nash equilibrium, observe that the delays player i incurs in the network congestion game when choosing the path through the grid or the edge (s_i, t_i) equal the delays of the corresponding player in the 2-threshold game when playing S_i^{in} or S_i^{out}, respectively, with an additional offset of D_i. Moreover, one can easily see that this reduction is tight. □

Our reduction can be extended to asymmetric network congestion games with undirected edges. Therefore, we modify the delays of the edges in order to achieve that there are only two best responses for each player.

Theorem 13. *The problem of finding a Nash equilibrium of an asymmetric network congestion game Γ with undirected edges is* PLS*-complete.*

Proof. We give a PLS-reduction from 2-threshold congestion games to asymmetric network congestion games with undirected edges. As before, let Γ be a 2-threshold congestion game and assume w. l. o. g. that for each pair of players $i, j \in \mathcal{N}$, there exists a unique resource $r_{i,j}$ that is contained in S_i^{in} and S_j^{in}. The undirected graph that we construct has the same structure as in the case of asymmetric networks with directed edges, except that we remove the directions of the edges, and split every edge $\{s_i, t_i\}$ into two edges by introducing a node s_i', i. e., we introduce the edges $\{s_i', s_i\}$ and $\{s_i', t_i\}$. The node s_i', and not the node s_i, will be the source of player i. In the previous reduction we could easily force a player to decide between two paths by considering directed edges. However, in the case of undirected edges we have to carefully introduce delays in order to achieve the same effect.

We now describe how we modify the delays of the vertical and horizontal edges in the grid given the delays of the reduction to network games with directed edges. Note that the delays of the nodes $v_{i,j}$ do not change. First we increase the delay of every vertical edge from 0 to D^2, where D is larger than the sum of all delays in the given 2-threshold game. Additionally, we define constant delays for the edges $\{s_i', s_i\}$ and $\{s_i', t_i\}$. Let the delay of the first edge be D^3 and the delay of the second be $D^3 + (n-i) \cdot D^2 + (i-1) \cdot i \cdot D + T_i$, where T_i denotes the threshold of player i in the given 2-threshold congestion game. Then for every player there are only 2 possible paths which can be best responses connecting s_i' and t_i. The delay of any other path is already larger than the largest possible delay on these two paths. We claim that player i either chooses the edge $\{s_i', t_i\}$ with delay $D^3 + (n-i) \cdot D^2 + (i-1) \cdot i \cdot D + T_i$, or the path $s_i = v_{i,1}, \ldots, v_{i,i}, v_{i+1,i}, \ldots, v_{n,i} = t_i$ with delay $D^3 + (n-i) \cdot D^2 + (i-1) \cdot i \cdot D + x$, where $x < D$ denotes the delay on the nodes of this path. If i would choose any other path, she would either pass a node s_j' with $j \neq i$, or she would allocate some additional vertical edges, or she would allocate some horizontal edges on lower levels. In all three cases her total delay is larger than the largest possible delay on the above mentioned paths.

Finally by the same arguments as before, a Nash equilibrium of the constructed network congestion game corresponds to a Nash equilibrium of the given 2-threshold congestion game. Moreover, one can easily see that this reduction is tight. □

In symmetric congestion games, a Nash equilibrium can be found in polynomial time [2]. Nonetheless, myopic players cannot find an equilibrium in polynomial time. We show this by simulating the behavior of the players in an asymmetric network congestion game in a symmetric network congestion game.

Theorem 14. *For every $n \in \mathbb{N}$ there exists a symmetric network congestion game Γ_{sym} (with directed or undirected edges) with n players, initial state S_{sym} and polynomially bounded network size such that every best response sequence starting in S_{sym} is exponentially long.*

Proof. We prove the theorem by simulating an asymmetric network congestion game by a symmetric one. In the case of asymmetric network congestion games, the existence of instances with the claimed properties follows since the reductions presented in the proof of Theorem 12 and in [2], and Theorem 13 are tight. Let Γ_{asym} be an asymmetric network congestion game and $S_{\text{asym}} = (P_1, \ldots, P_n)$ an initial state of Γ_{asym} such that every best response sequence starting in S_{asym} is exponentially long. Let $S(V)$ be the set of source and $T(V)$ the set of target nodes of the network G_{asym}. In order to receive a symmetric network congestion game, we introduce a common source s and a common target t such that s is connected to every source $s_i \in S(V)$ and such that every target $t_i \in T(V)$ is connected to t. For every new edge $e = (s, \cdot)$ and $e = (\cdot, t)$, we define the delay function d_e by $d_e(1) = 0$ and $d_e(n_e) = D$ for $n_e > 1$ with D being a number larger than the maximum total delay of every path in G_{asym}.

Assume that player i initially chooses path P_i with the additional edges (s, s_i) and (t_i, t), and let players iteratively play best responses. Obviously they behave in the same way as the do in the asymmetric case since no two players will share an edge $(s, \cdot)$ or $(\cdot, t)$. Thus, since in Γ_{asym} every best response path starting in S_{asym} is exponentially long, every best response path in Γ_{sym} starting in S_{sym} is exponentially long as well. ☐

5 Market Sharing Games

Market Sharing games have been introduced by Goemans et al. to model non-cooperative content distribution in wireless networks [3]. An instance of a market sharing game consists of a set $\mathcal{N} = \{1, \ldots, n\}$ of players, a set $\mathcal{M} = \{1, \ldots, m\}$ of markets, and a bipartite graph $G = (\mathcal{N} \cup \mathcal{M}, E)$. An edge between player i and market r indicates that player i is interested in market r. Furthermore, for each market r, costs c_r and a so-called query rate $q_r \in \mathbb{N}$ are given, and, for each player i, a budget B_i is specified. The query rate q_r determines the payoff of market r which is equally distributed among the players who have allocated that market, i. e., the payoff function of market r is given by $p_r(n_r) = q_r/n_r$. In terms of congestion games, the markets are the resources and the costs and budgets implicitly define the sets of feasible strategies. To be more precise, Σ_i consists of all sets $\mathcal{M}' \subseteq \mathcal{M}$ such that for all $r \in \mathcal{M}'$, $(i, r) \in E$ and $\sum_{r \in \mathcal{M}'} c_r \leq B_i$. Hence, market sharing games are congestion games in which, for each player, the set of strategies has a knapsack-like structure. In contrast to our definition of congestion games, the players are now interested in allocating a set of markets M' with maximum payoff instead of minimum delay. This can be achieved by considering payoffs to be negative delays.

If the costs c_r of every market r are 1, a market sharing game is called *uniform*. Goemans et al. give an algorithm for computing a Nash equilibrium of a uniform market sharing game in polynomial time. Observe that in uniform market sharing games, player i allocates an arbitrary subset of the markets she is interested in of size at most B_i. Hence Σ_i is a so-called B_i-*uniform matroid* on the set of markets in which player i is interested. If every payoff is non-negative, then only bases of this matroid can be best responses. Hence, we can apply Theorem 3 to obtain the following theorem.

Theorem 15. *In a uniform market sharing game Γ, players reach a Nash equilibrium after at most $n^2 \cdot m \cdot \max_{i \in \mathcal{N}} B_i$ best responses.*

If we allow arbitrary costs, then it becomes NP-hard to determine a best response since this corresponds to solving a knapsack problem, and hence the problem of finding a Nash equilibrium is not contained in PLS, unless P=NP. However, if the costs are polynomially bounded, then the problem of finding a Nash equilibrium is in PLS. In this case, we can easily enforce that a player $i \in \mathcal{N}$ decides between either allocating one market $\{a_i\}$ or a set of markets $\{b_i^{(1)}, \dots, b_i^{(k)}\}$ by setting the costs of market a_i to k, the costs of each market $b_i^{(j)}$ to one, and the budget of player i to k. If then every payoff is non-negative, the only possible best responses of player i are the strategies $\{a_i\}$ and $\{b_i^{(1)}, \dots, b_i^{(k)}\}$, regardless of the strategies of the other players. Hence, one can easily implement the counter presented in the proof of Theorem 10 in a market sharing game with general payoff functions $p_r : \mathbb{N} \to \mathbb{N}$ and polynomially bounded costs. When giving up the restriction of $(1+\varepsilon)$-greedy players, one can also implement the counter with standard payoff functions of the form $p_r(n_r) = q_r/n_r$ which answers an open question from Goemans et al. [3] and Mirrokni [7].

Theorem 16. *In market sharing games with polynomially bounded costs, myopic players do not find a Nash equilibrium in polynomial time in general.*

Proof. We give more details about how the counter constructed in Lemma 10 can be modified to work for payoff functions of the form $p_r(n_r) = q_r/n_r$. Therefore, first observe that in market sharing games, we do not need the tree players and the connection players. These were only necessary since only 1-2-exchanges were allowed. In market sharing games, we can, however, modify the reset-player of gadget G_i in such a way that she either plays the strategy $\{r_2^i\}$ or the strategy $\{r_3^i, r_1^{i-1}, \dots, r_1^0\}$ by appropriately choosing the costs of these resources and the budgets of the players. Hence, gadget G_i consists only of the resources r_1^i, r_2^i, and r_3^i and the bit and reset players. We define the costs of the markets as follows. In gadget G_i, we set $c_{r_1^i} = c_{r_3^i} = 1$ and $c_{r_2^i} = i+1$ and we set the budgets of the bit and reset players to $i+1$. The bit player is only interested in the resources r_1^i and r_2^i, and the reset player is interested in the resources r_2^i, r_3^i and r_1^j for every $j < i$. This way, we achieve that the bit player either plays $\{r_1^i\}$ or $\{r_2^i\}$ and the reset player plays either $\{r_2^i\}$ or $\{r_3^i, r_1^{i-1}, \dots, r_1^0\}$.

We scale the payoffs from gadget to gadget in such a way that the payoff which the reset player of gadget G_i gets from resources in gadgets G_j with $j < i$ are so small that they do not influence her decision. Without considering these payoffs, we obtain the following set of inequalities that have to be satisfied

$$\frac{q_1^i}{n-i} < \frac{q_2^i}{2} < q_3^i < q_2^i < \frac{q_1^i}{n-i-1} \ .$$

These inequalities cannot be satisfied. Therefore, we change the gadgets by adding players who are only interested in one particular resource. The inequalities can be satisfied if one adds n of these dummy players to r_2^i. This leads to the following satisfiable inequalities

$$\frac{q_1^i}{n-i} < \frac{q_2^i}{n+2} < q_3^i < \frac{q_2^i}{n} < \frac{q_1^i}{n-i-1} \ .$$

□

As mentioned above, it is easy to embed $(1,k)$-exchanges into market sharing games. Hence, for general payoff functions $p_r : \mathbb{N} \to \mathbb{N}$ and polynomially bounded costs, one can canonically reduce 2-threshold congestion games to market sharing games via a tight PLS-reduction.

Theorem 17. *In market sharing games with polynomially bounded costs and general payoff functions, it is* PLS*-complete to find a Nash equilibrium.*

6 Overlay Network Design

An overlay network is a network built on top of another network with fixed routing paths between all pairs of nodes. For example, Stoica et al. [12] suggest to generalize the Internet point to point communication to provide services like multicast, anycast, and mobility on the basis of overlay networks. In the case of multicast and anycast the overlay network is an aborescence connecting the source with the receivers. We simplify the scenario in many aspects and introduce the following overlay network congestion game: In an *overlay network design game* we are given an undirected graph $G = (V, E)$ with a delay function $d_e : \mathbb{N} \to \mathbb{N}$ for every edge $e \in E$ and a fixed routing path between any pair of nodes. For simplicity, we assume that the path from u to v corresponds to the path from v to u. Every player i wants to allocate a multicast tree $T_i = (V_i, E_i)$ on a subset $V_i \subseteq V$ of the nodes, where the edges in $E_i \subseteq V_i \times V_i$ form a spanning tree. Each edge $e \in E_i$ corresponds to the routing path in the network G, in particular, its delay equals the delay of the corresponding path. We show that finding a Nash equilibrium in an overlay network design game is PLS-complete, although, from a local point of view, every player solves a matroid optimization problem.

Theorem 18. *The problem of finding a Nash equilibrium in an overlay network design game is* PLS*-complete.*

Proof. We give a PLS-reduction from 2-threshold congestion games to overlay network design games. As in the proof of Theorem 12, we assume w. l. o. g. that for every pair $i, j \in \mathcal{N}$ of players, there is exactly one resource $r_{i,j}$ that is contained in S_i^{in} and S_j^{in}. We slightly modify the reduction presented in the proof of Theorem 12. We also take an $n \times n$-grid as basis of our construction, but now with undirected edges, and we use the identifiers $s_1, \ldots, s_n$ and $t_1, \ldots, t_n$ to denote the same nodes as before. The edges in the grid all have delay 0, the delay function of node $v_{i,j}$ still equals the delay function of resource $r_{i,j}$. Additionally, for each player $i \in \mathcal{N}$, we add a node t'_i and an edge (t_i, t'_i) with delay 0. Instead of having an edge (s_i, t_i) with delay $D_i + T_i$, we add an edge (s_i, t'_i) with delay T_i. In the network, the routing path between s_i and t_i is defined to be $s_i = v_{i,1}, \ldots, v_{i,i}, v_{i+1,i}, \ldots, v_{n,i} = t_i$. The routing paths between s_i and t'_i and between t_i and t'_i in the overlay network are defined to be the direct edges contained in the network G. Now, for every player i in the 2-threshold game, we define a player in the overlay network design game with $V_i = \{s_i, t_i, t'_i\}$.

Every best response of player i must contain the edge between t_i and t'_i since it has delay 0. Hence, every player decides between either taking the virtual edge between s_i and t_i in the overlay network or the edge between s_i and t'_i. In the former case, her message is routed along the path through the grid. Analogously to the proof of Theorem 12, this shows that it is PLS-complete to find a Nash equilibrium in an overlay network design game. □

References

[1] Elliot Anshelevich, Anirban Dasgupta, Jon Kleinberg, Eva Tardos, Tom Wexler, and Tim Roughgarden. The price of stability for network design with fair cost allocation. In *Proc. 45th Ann. IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 295–304, 2004.

[2] Alex Fabrikant, Christos Papadimitriou, and Kunal Talwar. The complexity of pure Nash equilibria. In *Proc. 36th Ann. ACM. Symp. on Theory of Comput. (STOC)*, pages 604–612, 2004.

[3] Michel X. Goemans, Erran L. Li, Vahab S. Mirrokni, and Marina Thottan. Market sharing games applied to content distribution in ad-hoc networks. In *MobiHoc*, pages 55–66, 2004.

[4] Armin Haken. Connectionist networks that need exponential time to stabilize. Unpublished manuscript. Department of Computer Science, University of Toronto, 1989.

[5] Samuel Ieong, Robert McGrew, Eugene Nudelman, Yoav Shoham, and Qixiang Sun. Fast and compact: A simple class of congestion games. In *20th National Conference on Artificial Intelligence (AAAI)*, 2005.

[6] David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. How easy is local search? *J. Comput. Syst. Sci.*, 37(1):79–100, 1988.

[7] Vahab S. Mirrokni. *Approximation Algorithms for Distributed and Selfish Agents*. PhD thesis, Massachusetts Institue of Technology, 2005.

[8] Pekka Orponen. Computing with truly asynchronous threshold logic networks. *Theor. Comput. Sci.*, 174(1-2):123–136, 1997.

[9] Robert W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.

[10] Alejandro A. Schäffer and Mihalis Yannakakis. Simple local search problems that are hard to solve. *SIAM J. Comput.*, 20(1):56–87, 1991.

[11] Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Volume B, Matroids, Trees, Stable Sets, Chapters 39-69. Springer, 2003.

[12] Ion Stoica, Daniel Adkins, Shelley Zhuang, Scott Shenker, and Sonesh Surana. Internet indirection infrastructure. *IEEE/ACM Trans. Netw.*, 12(2):205–218, 2004.

[13] Renato Werneck, João Setubal, and Arlindo da Conceicão. Finding minimum congestion spanning trees. *J. Exp. Algorithmics*, 5:11, 2000.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de/