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
December 18, 2007

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 that 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 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. In particular, we show that network congestion games are PLS-complete for directed and undirected networks even in case of linear latency functions.

1 Introduction

Congestion games are a natural and generally accepted approach to model resource allocation among selfish or myopic players. In a congestion game we have a set of resources. 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 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 problems in distributed systems.

Rosenthal [12] 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. [4] shows, however, that these best response sequences may require an exponential number of iterations. Their analysis relates congestion games to local search problems and it shows that it is PLS-complete to compute a Nash equilibrium for general congestion games. The completeness proof is based on a *tight* PLS-reduction preserving lower bounds on the lengths of improvement sequences. This way, 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. [4] 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, Fabrikant et al. 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. [7] 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

Congestion Games. A *congestion game* Γ is a tuple $(\mathcal{N}, \mathcal{R}, (\Sigma_i)_{i \in \mathcal{N}}, (d_r)_{r \in \mathcal{R}})$ where $\mathcal{N} = \{1, \ldots, n\}$ denotes the set of players, $\mathcal{R}$ with $|\mathcal{R}| = m$ the set of resources, $\Sigma_i \subseteq 2^{\mathcal{R}}$ the strategy space of player i, and $d_r \colon \mathbb{N} \to \mathbb{N}$ 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 $S = (S_1, \ldots, S_n)$ 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 $S' = (S_1, \ldots, S_{i-1}, S_i', S_{i+1}, \ldots, S_n)$, 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 wish to play strategies $S_i \in \Sigma_i$ that minimize their individual delays. The delay $\delta_i(S)$ of player i is given by $\delta_i(S) = \sum_{r \in S_i} d_r(n_r(S))$. Given a state S, we call a strategy $S_i^* \in \Sigma_i$ a *best response* of player i to S if, for all $S_i' \in \Sigma_i$, $\delta_i(S \oplus S_i^*) \leq \delta_i(S \oplus S_i')$. 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 $S_i' \in \Sigma_i$, $\delta_i(S) \leq \delta_i(S \oplus S_i')$. Rosenthal [12] shows that every congestion games possesses at least one Nash equilibrium by considering the potential function $\phi \colon \Sigma_1 \times \cdots \times \Sigma_n \to \mathbb{N}$ with $\phi(S) = \sum_{r \in \mathcal{R}} \sum_{i=1}^{n_r(S)} d_r(i)$. For a state S, we call $\gamma(S) = \sum_{i \in \mathcal{N}} \delta_i(S)$ the *social delay of state S*.

Local Search Problems. 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 $\mathcal{F}(I)$, an objective function $c \colon \mathcal{F}(I) \to \mathbb{N}$, 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 that does not have a strictly better neighbor. A neighbor S' of a solution S is strictly better if the objective value $c(S')$ is larger or smaller than $c(S)$ in the case of a maximization or minimization problem, respectively. The class PLS is defined by Johnson, Papadimitriou, and Yannakakis [8] and it contains all local search problems with polynomial time searchable neighborhoods. Formally it is defined as follows.

Definition 1.1. *A local search problem Π belongs to PLS if there exist polynomial time algorithms for the following tasks:*

1. *an algorithm A that computes for every instance I of Π an initial feasible solution $S^0 \in \mathcal{F}(I)$,*

2. *an algorithm B that computes for every instance I of Π and every feasible solution $S \in \mathcal{F}(I)$ the objective value $c(S)$,*

3. *an algorithm C that 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.*

Given an instance I of a local search problem Π, we denote by $TG(I)$ the *transition graph* that contains a node $v(S)$ for every feasible solution $S \in \mathcal{F}(I)$ and a directed edge from a node $v(S^1)$ to a node $v(S^2)$ if S^2 is in the neighborhood of S^1 and if the objective value $c(S^2)$ is strictly better than the objective value $c(S^1)$.

Johnson, Papadimitriou, and Yannakakis [8] introduce the notion of a PLS-*reduction* and of PLS-*complete problems*.

Definition 1.2. *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:*

1. *f maps instances I of Π_1 to instances $f(I)$ of Π_2,*

2. *g maps pairs (S^2, I), where S^2 denotes a solution of $f(I)$, to solutions S^1 of I,*

3. *for all instances I of Π_1 and all solutions S^2 of $f(I)$, if S^2 is a local optimum of instance $f(I)$, then $g(S^2, I)$ is 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.

Schäffer and Yannakakis [13] introduce the notion of a *tight* PLS-*reduction*, which ensures several properties of the corresponding transition graphs.

Definition 1.3. *A PLS-reduction (f, g) is tight if for any instance $I \in \mathcal{I}_{\Pi_1}$ one can choose a subset $\mathcal{Q}$ of feasible solutions for the image instance $J = f(I) \in \mathcal{I}_{\Pi_2}$ such that the following properties are satisfied:*

1. *$\mathcal{Q}$ contains all local optima of J.*

2. *There exists a polynomial time algorithm which constructs for every feasible solution $S^I \in \mathcal{F}(I)$, a feasible solution $S^J \in \mathcal{Q}$ of J such that $g(S^J, I) = S^I$.*

3. *Suppose that the transition graph $TG(J)$ of J contains a directed path from $S^{J,1}$ to $S^{J,2}$ such that $S^{J,1}, S^{J,2} \in \mathcal{Q}$, but all internal path vertices are outside $\mathcal{Q}$, and let $S^{I,1} = g(S^{J,1}, I)$ and $S^{I,2} = g(S^{J,2}, I)$ be the corresponding feasible solutions of I. Then either $S^{I,1} = S^{I,2}$ or $TG(I)$ contains an edge from $S^{I,1}$ to $S^{I,2}$.*

An important property of tight PLS-reductions is that they do not shorten paths in the transition graph. That is, if the graph $TG(I)$ for an instance I of Π_1 contains a node whose shortest distance to a local optimum is z, then $TG(f(I))$ contains a node whose shortest distance to a local optimum is at least z. Moreover, consider the following problem: given an instance I of Π_1 and an initial solution S^0 of I, find a local optimum that is reachable from S^0 in the transition graph. If this problem is PSPACE-complete for Π_1 and there exists a tight PLS-reduction from Π_1 to Π_2, then the problem is also PSPACE-complete for Π_2.

We introduce two variants of tight PLS-reductions, which we call *embedding* PLS-*reduction* and *isomorphic* PLS-*reduction*.

Definition 1.4. *We call a problem* Π_1 *in* PLS *embedding* PLS-*reducible to a problem* Π_2 *in* PLS *if*

1. Π_1 *is* PLS-*reducible to* Π_2,

2. *for every instance I of Π_1, the transition graph $TG(I)$ is isomorphic to a subgraph $TG^*(f(I))$ of $TG(f(I))$ that contains all local optima of the instance $f(I)$ and has no outgoing edges. Furthermore, when restricted to this subgraph, the function $g(\cdot, I)$ must be an isomorphism between $TG^*(f(I))$ and $TG(I)$ and its inverse must be computable in polynomial time.*

We call Π_1 isomorphic PLS-*reducible to Π_2 if Π_1 is embedding* PLS-*reducible to Π_2, and if for every instance I of Π_1 the transition graphs $TG(I)$ and $TG(f(I))$ are isomorphic.*

It is easy to verify that every embedding PLS-reduction is also a tight PLS-reduction. In order to see this, let $J = f(I)$ and let $TG^*(J)$ be the subgraph of $TG(J)$ to whom $TG(I)$ is isomorphic. Observe that the set of feasible solutions that corresponds to the nodes of $TG^*(J)$ defines a set $\mathcal{Q}$ with the properties in Definition 1.3. The first and the second condition are directly satisfied by the definition of an embedding PLS-reduction. The third condition is satisfied because $TG^*(J)$ has no outgoing edges and therefore paths with the properties as in condition 3 cannot have internal path vertices and must therefore be single edges with both endpoints from $TG^*(J)$. Since $TG^*(J)$ is isomorphic to $TG(I)$, the third condition must be satisfied for these edges.

1.2 Our Results

Upper and Lower Bounds on the Convergence Time. We show that the analysis of Ieong et al. [7] can be generalized towards matroid congestion games, 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 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. We show that this characterization holds even for ε-greedy players, i.e., for players who only have an incentive to change their strategy if this decreases their delay by at least a factor of $\varepsilon > 1$.

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 [16]. There are quite a few more interesting applications as even simple matroid structures like uniform matroids, which 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 [5, 9], and scheduling games in which each player has to injectively allocate a given set of tasks (services) to a given set of machines (servers). We

also analyze the complexity of finding socially optimal states in matroid congestion games. In particular, we show that computing a Nash equilibrium with minimum social delay is NP-hard in spanning tree congestion games, and we observe that for *weakly convex* delay functions a state with socially optimal delay can be computed in polynomial time.

Our negative result for the convergence time in 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 $(1, 2)$-exchanges as basic building blocks.

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. [4]. Unfortunately, the analysis in [4] is not very instructive as it completely reworks the very involved completeness proof of PosNAE3Flip (Not-All-Equal-3SAT with weighted clauses and positive literals only) from [13] and adds some further complications. (According to [4] already the analysis from [13] 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 PosNAE3Flip to MaxCut (which is equivalent to PosNAE2Flip) [13]. We show that MaxCut can be reduced to so-called *(quadratic) threshold games*. The strategy space of each player in a threshold game corresponds to a $(1, k)$-exchange. Quadratic threshold games have further restrictions on the global structure of these games. Despite their simple structure, threshold games are a natural and interesting class of games. Our main interest, however, stems from the fact that quadratic threshold games are a good starting point for PLS-reductions because of their simple structure. 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 polynomially bounded costs),
- overlay network design games, 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 and linear delay functions.

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 embedding reductions, 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 briefly introduce matroids. For a detailed discussion of matroids we refer the reader to [14].

Definition 2.1. *A tuple $M = (\mathcal{R}, \mathcal{I})$ is a matroid if $\mathcal{R}$ 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 $i \in I \setminus J$ with $J \cup \{i\} \in \mathcal{I}$.*

Let $M = (\mathcal{R}, \mathcal{I})$ be a matroid, and 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* $\mathrm{rk}(M)$ of the matroid. A maximal independent set B is called a *basis* of M. In the case of a weight function $w \colon \mathcal{R} \to \mathbb{N}$, we call a matroid *weighted*, and wish 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. In the following, we state two additional useful properties of matroids. We denote by $\mathcal{B}$ the set of bases of a matroid M and assume that $B_1, B_2 \in \mathcal{B}$.

Proposition 2.2 ([14])**.** *Let $r_2 \in B_2 \setminus B_1$, then there exists some $r_1 \in B_1 \setminus B_2$ such that $B_1 \cup \{r_2\} \setminus \{r_1\} \in \mathcal{B}$.*

We denote by $G(B_1 \Delta B_2)$ the bipartite graph (V, E) with $V = (B_1 \setminus B_2) \cup (B_2 \setminus B_1)$ and $E = \{\{r_1, r_2\} \mid r_1 \in B_1 \setminus B_2, r_2 \in B_2 \setminus B_1, B_1 \cup \{r_2\} \setminus \{r_1\} \in \mathcal{B}\}$.

Proposition 2.3 ([14])**.** *There exists a perfect matching in the graph $G(B_1 \Delta B_2)$.*

We are now ready to define matroid congestion games.

Definition 2.4. *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 and Σ_i is the set of bases of M_i. Additionally, we denote by $\mathrm{rk}(\Gamma) = \max_{i \in \mathcal{N}} \mathrm{rk}(M_i)$ the rank of the matroid congestion game Γ.*

2.1 Fast Convergence

Ieong at al. [7] show that in singleton games players reach a Nash equilibrium after at most n^2m best responses. Note that singleton games are matroid congestion games with $\mathrm{rk}(M_i) = 1$ for every player i. We now extend this analysis to general matroid congestion games.

Theorem 2.5. *Let Γ be a matroid congestion game. Then players reach a Nash equilibrium after at most $n^2m \cdot \mathrm{rk}(\Gamma)$ best responses. In the case of identical delay functions, players reach a Nash equilibrium after at most $n^2 \cdot \mathrm{rk}(\Gamma)$ best responses.*

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

Lemma 2.6. *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. the delays d_r which strictly decreases the delay of player i. Then S_i^* also strictly decreases the delay of player i w.r.t. the delays $\tilde{d}_r$.*

Proof. Consider the bipartite graph $G(S_i^* \Delta S_i)$, which contains a perfect matching P_M due to Proposition 2.3. Let $S^* = S \oplus S_i^*$ and observe that for every edge $\{r^*, r\} \in P_M$, with $r^* \in S_i^* \setminus S_i$ and $r \in S_i \setminus S_i^*$, $d_{r^*}(n_{r^*}(S^*)) \le d_r(n_r(S^*) + 1) = d_r(n_r(S))$ since, otherwise, S_i^* is not a best response w.r.t. the delays d_r. Additionally, there exists at least one edge with $d_{r^*}(n_{r^*}(S^*)) < d_r(n_r(S^*) + 1) = d_r(n_r(S))$ 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. $\square$

Now due to Lemma 2.6, whenever a player plays a best response w.r.t. the delays d_r, Rosenthal's potential decreases w.r.t. the delays $\tilde{d}_r$. Since there are at most nm different delays, $\tilde{d}_r(i) \le nm$ for all resources $r \in \mathcal{R}$ and for all possible congestion values i. Hence,

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

where the second inequality holds as each of the n player occupies at most $\mathrm{rk}(\Gamma)$ resources. 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 nm, which implies the second part of the theorem. $\square$

Note that Theorem 2.5 is independent of the delay functions. In particular, we do not assume monotonicity or that all delays have the same sign.

2.2 Socially Optimal States and Nash Equilibria

Theorem 2.5 states that an arbitrary Nash equilibrium of a matroid congestion game can be computed in polynomial time. A natural problem related to this one is to consider the complexity of computing a *socially optimal Nash equilibrium*, i.e., a Nash equilibrium S minimizing the social delay $\gamma(S) = \sum_{i \in \mathcal{N}} \delta_i(S)$ among all Nash equilibria, or a socially optimal state, i.e., a state S minimizing the social delay among all states. Chakrabarty, Mehta, and Nagarajan [3] present an efficient algorithm for computing a socially optimal state of a singleton congestion game with monotone delay functions. Ieong et al. [7] present an efficient algorithm for computing a socially optimal Nash equilibrium of a singleton game. In the following, we prove that both problems are NP-hard for spanning tree congestion games with non-monotone delay functions.

Theorem 2.7. *For spanning tree congestion games with non-monotone delay functions it is* NP-*hard to compute a socially optimal state or a socially optimal Nash equilibrium.*

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 we assume that G is connected. We construct a spanning tree congestion game by setting $n = |V|$, $\mathcal{R} = E$, $\Sigma_i = \{T \mid T$ is a spanning tree of $G\}$ and $d_r(n_r) = (n_r - n + 1)^2$.

First observe that if G contains a Hamiltonian cycle, then there is a Nash equilibrium S of the game with $\gamma(S) = 0$. In this state, every player allocates the edges of the Hamiltonian cycle, except for one of them. If each player removes an individual edge from the cycle, the congestion on each edge is either $n - 1$ or 0, and therefore each player has delay 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 for every resource $r \in \mathcal{R}$. Now consider the subgraph $G' = (V', E')$ of G which only contains the edges with $n_r = n - 1$. Observe that $V' = V$, $|E'| = n$, and G' is connected. This implies that G' is the union of a single spanning tree T and one extra edge r not contained in T. Note that $T \cup \{r\}$ 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 players have allocated the edge incident to this node, which is a contradiction to our construction of G'. Hence, if there exists a state S with $\gamma(S) = 0$, which is necessarily a Nash equilibrium, then G contains a Hamiltonian cycle. $\square$

Note that the previous theorem relies on the fact that the delay functions are non-monotone. It remains an open question to settle the complexity of these problems in the case of monotone delay functions. In the case of *weakly convex delay functions*, however, the problems become easier. A delay function is called weakly convex if $n_r \cdot d_r(n_r) - (n_r - 1) \cdot d_r(n_r - 1) \leq (n_r + 1) \cdot d_r(n_r + 1) - n_r \cdot d_r(n_r)$ for all $1 \leq n_r < n$. Observe that polynomial delay functions with positive coefficients are weakly convex. Werneck and Setubal [16] consider spanning tree congestion games with such delay functions and show how to compute a socially optimal solution efficiently. Since one can easily extent their algorithm to arbitrary matroid congestion games, the following theorem follows.

Theorem 2.8. *There exists a polynomial time algorithm for the problem of computing a socially optimal state of an arbitrary matroid congestion game with weakly convex delay functions*

3 Non-Matroid Congestion Games

In the previous section, we showed that the matroid property is a sufficient condition on the combinatorial structure of the players' strategy spaces guaranteeing fast convergence to Nash equilibria. In this section, we prove that the matroid property is also necessary to guarantee fast convergence. In the following, let Σ be a set system on a set $\mathcal{R}$ of resources. We call Σ an *anti-chain* if for every $X \in \Sigma$, no proper superset $Y \supset X$ belongs to Σ. Moreover, we call Σ a *non-matroid set system* if the tuple $(\mathcal{R}, \{X \subseteq S \mid S \in \Sigma\})$ is not a matroid.

Theorem 3.1. *Let Σ be an a non-matroid anti-chain on a set of resources $\mathcal{R}$. Then, for every $n \in \mathbb{N}$, there exists a congestion game Γ with*

- *$4n$ players each of which having a strategy space isomorphic to Σ, and*
- *$O(n \cdot |\mathcal{R}|)$ resources with non-negative and non-decreasing delay functions*

such that there exists a best response sequence of length 2^n.

Thus, given a non-matroid anti-chain we can always construct a congestion game with an exponentially long best response sequence. We are only interested in the combinatorial structure of the strategy spaces, and we assume that the strategy spaces of different players can be interweaved arbitrarily. Observe that this matches the setting of our upper bound in Theorem 2.5. The assumption that Σ is an anti-chain is natural when all delays are non-negative as, in this case, supersets are dominated by subsets so that, w. l. o. g., supersets are never used as best responses. Hence, we can conclude the following corollary.

Corollary 3.2. *The matroid property is the maximal property on the individual players' strategy spaces that guarantees polynomial convergence in congestion games.*

3.1 A Characterization of Non-Matroid Set Systems

The key insight for proving Theorem 2.5 is that best responses in matroid congestion games 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 set system, and show that it is sufficient to construct congestion games with exponentially long best response sequences.

Definition 3.3 ($(1, 2)$-exchange property)**.** *Let Σ be an anti-chain on a set of resources $\mathcal{R}$. We say that Σ satisfies the $(1, 2)$-exchange property if we can identify three distinct elements $a, b, c \in \mathcal{R}$ with the property that for any given $k \in \mathbb{N}$, we can choose a delay $d(r) \in \{0, k+1\}$ 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 S from Σ with minimum delay, $a \in S$ and $b, c \notin S$. If $d(a) > d(b) + d(c)$, then for every set S from Σ with minimum delay, $a \notin S$ and $b, c \in S$.

Lemma 3.4. *Let Σ be an anti-chain on a set of resources $\mathcal{R}$. Furthermore, 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 Σ possesses the $(1, 2)$-exchange property.*

In [1] we introduce a version of the $(1, 2)$-exchange property in which we assume that all delays $d(r)$ with $r \in \mathcal{R} \setminus \{a, b, c\}$ are strictly positive, and we prove that every non-matroid anti-chain possesses this property. Although, the proof of Lemma 3.4 follows the same arguments, we present it here for the sake of completeness. The following proposition will be crucial for its proof.

Proposition 3.5 ([14])**.** *Let $\mathcal{R}$ be a set of resources and let Σ be a nonempty collection of subsets of $\mathcal{R}$. The following statements are equivalent:*

- *Σ is the set of bases of a matroid over $\mathcal{R}$.*
- *If $B_1, B_2 \in \Sigma$ and $r_1 \in B_1 \setminus B_2$, then there exists $r_2 \in B_2 \setminus B_1$ such that $B_1 \cup \{r_2\} \setminus \{r_1\} \in \Sigma$.*

Proof of Lemma 3.4. Since $(\mathcal{R}, \mathcal{I})$ is not a matroid, there exist due to Proposition 3.5 two sets $X, Y \in \Sigma$ and a resource $x \in X \setminus Y$ such that for every $y \in Y \setminus X$, the set $X \setminus \{x\} \cup \{y\}$ is not contained in Σ.

Let X and Y be such sets and let $x \in X$ be such a resource. Consider all subsets Y' of the set $X \cup Y \setminus \{x\}$ with $Y' \in \Sigma$. Every such set Y' can be written as $Y' = X \setminus \{x = x_1, \ldots, x_l\} \cup \{y_1, \ldots, y_{l'}\}$ with $x_i \in X \setminus Y$ and $y_i \in Y \setminus X$ and $l + l' > 2$. This is true since $l \geq 1$ holds per definition and $l' \geq 1$ holds because Σ is an anti-chain. Furthermore l and l' cannot both equal 1 as otherwise we obtain a contradiction to the choice of X, Y, and x. Among all these sets Y', let $Y_{\min}$ denote one set for which l' is minimal. Observe that we can replace Y by $Y_{\min}$ without changing the aforementioned properties of X, Y, and x. Hence, in the following, we assume that $Y = Y_{\min}$, that is, we assume that $Y \setminus X = Y' \setminus X$ for all of the aforementioned sets Y'.

We claim that we can always identify resources $a, b, c \in X \cup Y$ such that either $a \in X \setminus Y$ and $b, c \in Y \setminus X$ or $a \in Y \setminus X$ and $b, c \in X \setminus Y$ with the property that for every $Z \subseteq X \cup Y$ with $Z \in \Sigma$, if $a \notin Z$, then $b, c \in Z$. In order to see this, we distinguish between the cases $l' = 1$ and $l' \geq 2$:

1. Let $Y \setminus X = \{y_1\}$ and hence $X \setminus Y = \{x = x_1, \ldots, x_l\}$ with $l \geq 2$. Then we set $a = y_1$, $b = x_1$, and $c = x_2$. Consider a set $Z \subseteq X \cup Y$ with $Z \in \Sigma$ and $a \notin Z$. Then $Z = X$ since Σ is an anti-chain, and hence $b, c \in Z$.

2. Let $Y \setminus X = \{y_1, \ldots, y_{l'}\}$ with $l' \geq 2$. Then we set $a = x$, $b = y_1$, and $c = y_2$. Consider a set $Z \subseteq X \cup Y$ with $Z \in \Sigma$ and $a \notin Z$. Since we assumed that $Y = Y_{\min}$, it must be $b, c \in Z$ as otherwise $Z \setminus X \neq Y \setminus X$.

Now we define delays for the resources in $\mathcal{R} \setminus \{a, b, c\}$ such that the properties in Definition 3.3 are satisfied. Let $k \in \mathbb{N}$ be chosen as in Definition 3.3, that is $d(a), d(b), d(c) \in$

11

$\{1, \ldots, k\}$. We set $d(r) = k + 1$ for every resource $r \notin X \cup Y$ and $d(r) = 0$ for every resource $r \in (X \cup Y) \setminus \{a, b, c\}$. First of all, observe that in the first case the delay of Y equals $d(a) \leq k$ and that in the second case the delay of X equals $d(a) \leq k$. Hence, a set $Z \in \Sigma$ that contains a resource $r \notin X \cup Y$ can never have minimum delay as its delay is at least $k + 1$. Thus, only sets $Z \in \Sigma$ with $Z \subseteq X \cup Y$ can have minimum delay. Since for such sets, $a \notin Z$ implies $b, c \in Z$, we know that every set with minimum delay must contain either a or it must contain b and c.

Consider the case $d(a) < d(b) + d(c)$ and assume for contradiction that there exists an optimal set Z^* with $a \notin Z^*$. Due to the choice of a, b, and c, the set Z^* must then contain b and c. Hence $d(Z^*) = d(b) + d(c)$. Furthermore, again due to the choice of a, b, and c, there exists a set $Z' \subseteq X \cup Y$ with $a \in Z'$ and $b, c \notin Z'$. The delay of Z' is $d(Z') = d(a) < d(b) + d(c) = d(Z^*)$, contradicting the assumption that Z^* has minimum delay. Hence every optimal set Z^* must contain a. If Z^* additionally contains b or c, then its delay is at least $d(a) + 1 > d(Z')$. Hence, in the case $d(a) < d(b) + d(c)$ every optimal set Z^* contains a but it does not contain b and c.

Consider the case $d(a) > d(b) + d(c)$ and assume for contradiction that there exists an optimal set Z^* with $b \notin Z^*$ or $c \notin Z^*$. Then Z^* must contain a and hence its delay is at least $d(a)$. Due to the choice of a, b, and c, there exists a set $Z' \subseteq X \cup Y$ with $a \notin Z'$ and $b, c \in Z'$. The delay of Z' is $d(Z') = d(b) + d(c) < d(a) \leq d(Z^*)$, contradicting the assumption that Z^* has minimum delay. Hence every optimal set Z^* must contain b and c. If Z^* additionally contains a, then its delay is at least $d(b) + d(c) + 1 > d(Z')$. Hence, in the case $d(a) > d(b) + d(c)$ every optimal set Z^* contains b and c but it does not contain a. □

3.2   Proof of Theorem 3.1

Proof of Theorem 3.1. A well known technique for constructing instances of local search problems with exponentially long sequences of local improvements is to construct instances that resemble the behavior of a binary counter (see, e.g., [2, 6, 10]). We construct a game that consists of n gadgets $G_0, \ldots, G_{n-1}$ that correspond to the bits of the counter. Each of these gadgets has a 0-state and a 1-state and for each gadget there exists a best response sequence from its 1-state to its 0-state when no other gadget interferes with it. A gadget that is in state 0 can be triggered by another gadget to change to state 1. The crucial property of our construction is that whenever a gadget G_i changes its state from 0 to 1, then it triggers gadget G_{i-1} twice. Hence, if G_{n-1} is triggered once, then every gadget G_i is triggered 2^{n-i-1} times. Thus the game possesses a best response sequence of length at least 2^n.

In the following, we denote by Σ_i a set system over a set of resources $\mathcal{R}_i$. We assume that Σ_i is isomorphic to Σ, and that Σ_i is the strategy space of player i. Due to Lemma 3.4, we can for every player $i \in \mathcal{N}$, identify three resources a_i, b_i, and $c_i \in \mathcal{R}_i$ with the properties as in Definition 3.3. These are the only resources of player i that she shares with other players. Resources in the set $\mathcal{R}_i \setminus \{a_i, b_i, c_i\}$ 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. The parameter k in Definition 3.3 is chosen as upper bound on the maximum delay of one of these three resources. 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 $\{b_i, c_i\}$. We have made no restrictions on the global structure of the game. Hence, we can interweave the resources a_i, b_i, c_i of different players in an arbitrary manner.

Each gadget G_i consists of 6 resources $r_i^0, r_i^1, r_i^2, r_i^3, r_i^4, r_i^5$ and 4 players, which we call Init$_i$-, Trigger$_i$-, P$_i^1$-, and P$_i^2$-player. Every player has two strategies, namely a 0-strategy and a 1-strategy. If all players of gadget G_i play their 0-strategies, then we say that gadget G_i is in its 0-state. Similarly, if all players play their 1-strategies, then we say that G_i is in its 1-state. If gadget G_i is in state 0, then Init$_i$ is the player who is triggered by the player Trigger$_{i+1}$ from gadget G_{i+1} and initiates a sequences of best responses resetting G_i to its 1-state.

For every player, her 0-strategy consists of one resource and her 1-strategy consists of two resources. The strategy spaces of the players are defined as follows:

1. $\Sigma_{\text{Init}_i} = \{\{r_i^0\}, \{r_i^1, r_i^2\}\}$,
2. $\Sigma_{\text{Trigger}_i} = \{\{r_i^1\}, \{r_i^3, r_{i-1}^0\}\}$,
3. $\Sigma_{\text{P}_i^1} = \{\{r_i^2\}, \{r_i^3, r_i^4\}\}$,
4. $\Sigma_{\text{P}_i^2} = \{\{r_i^4\}, \{r_i^1, r_i^5\}\}$.

Now we describe the aforementioned best response sequence of gadget G_i in detail. Assume that gadget G_i is in its 0-state, that is, every player plays her 0-strategy. If player Init$_i$ is triggered by the player Trigger$_{i+1}$ from gadget G_{i+1}, that is, if Trigger$_{i+1}$ allocates the resource r_i^0, then the following sequence of strategy changes can take place in gadget G_i.

1. Init$_i$ changes to her 1-strategy.
2. Trigger$_i$ changes to her 1-strategy.
3. P$_i^1$ changes to her 1-strategy.
4. Trigger$_i$ changes back to her 0-strategy.
5. P$_i^2$ changes to her 1-strategy.
6. Trigger$_i$ changes to her 1-strategy again.

Moreover, if all players play their 1-strategy and Init$_i$ is not triggered by the player Trigger$_{i+1}$, then there exists a sequence of best responses such that all players of G_i change back to their 0-strategies. In this sequence, Init$_i$ changes to her 0-strategy first, then P$_i^1$, P$_i^2$, and finally Trigger$_i$. We assume that gadget G_i performs this sequence of steps immediately after player Trigger$_{i+1}$ has left resource r_i^0. Observe that this construction ensures the property that gadget G_{i+1} resets gadget G_i twice from state 0 to state 1 every time it changes its own state from 0 to 1. The first time gadget G_i triggers gadget G_{i-1} is after the first two strategy changes of the aforementioned sequence have been performed. In the last step of this sequence, gadget G_i triggers G_{i-1} for the second time.

Hence, this construction ensures the existence of best response sequences of length at least 2^n. Therefore, assume that initially every gadget is in its 0-state and that gadget G_{n-1} is triggered to change its state to 1. This can be accomplished by, e. g., introducing one additional player who allocates resource r_{n-1}^0. If all players act according to the

aforementioned sequence of strategy changes, then every gadget G_i is reset from its 0-state to its 1-state 2^{n-i-1} times.

Let $*$ denote either 1 or 2. If the following inequalities are satisfied, then all six strategy changes in the aforementioned sequence of strategy changes are best responses:

1. $d_{r_i^0}(2) > d_{r_i^1}(2) + d_{r_i^2}(2)$,

2. $d_{r_i^1}(2) > d_{r_i^3}(1) + d_{r_{i-1}^0}(*)$,

3. $d_{r_i^2}(2) > d_{r_i^4}(2) + d_{r_i^3}(2)$,

4. $d_{r_i^1}(2) < d_{r_i^3}(2) + d_{r_{i-1}^0}(*)$,

5. $d_{r_i^4}(2) > d_{r_i^5}(1) + d_{r_i^1}(3)$,

6. $d_{r_i^1}(3) > d_{r_i^3}(2) + d_{r_{i-1}^0}(*)$.

Let $\alpha \geq 2$ be chosen arbitrarily, and for every gadget G_i, let $c_i = \alpha^{20i}$. We use c_i to scale the delays of the resources in such a way that the best response of the player Trigger_i is independent of the delay on the resource r_{i-1}^0. We set $d_{r_{i-1}^j}(1) = c_i \cdot \alpha^{2j}$ for every resource r_i^j and for every gadget G_i and furthermore $d_{r_i^0}(2) = c_i \cdot \alpha^{20}, d_{r_i^1}(2) = c_i \cdot \alpha^8, d_{r_i^1}(3) = c_i \cdot \alpha^{14}, d_{r_i^2}(2) = c_i \cdot \alpha^{18}, d_{r_i^3}(2) = c_i \cdot \alpha^{10}, d_{r_i^4}(2) = c_i \cdot \alpha^{16}$. One can easily verify that the aforementioned inequalities are all satisfied. Furthermore, observe that the second sequence of strategy changes in which G_i changes its state from 1 to 0 consists of best responses only since in this sequence every player changes to a resource that no other player has allocated. Finally, one can easily verify that all strategy changes do not only decrease the delay but decrease it even by a factor of at least α. $\qquad\square$

Finally, we like to comment on ε-*greedy players*, which have been introduced by Mirrokni [9]. An ε-greedy player is a player who only changes her strategy when this decreases her current delay by at least a factor of $\varepsilon > 1$. In general, these players do not reach a Nash equilibrium of a congestion game but a state in which no player can improve her delay by a factor of ε, a so-called ε-*approximate Nash equilibrium*. The instances constructed in the proof of Theorem 3.1 possess the property that a player who decreases her delay decreases it by a factor of at least α for an arbitrary given $\alpha \geq 2$. Hence, not even ε-greedy players necessarily reach an approximate equilibrium in polynomial time.

4 Complexity of Computing Equilibria

In this section, we analyze the complexity of computing Nash equilibria in various kinds of structured congestion games.

4.1 Threshold Games

Threshold games are a special class of congestion games in which the set of resources $\mathcal{R}$ is divided into two disjoint subsets $\mathcal{R}_{\text{in}}$ and $\mathcal{R}_{\text{out}}$. The set $\mathcal{R}_{\text{out}}$ contains a resource r_i for every $i \in \mathcal{N}$. This resource has a fixed delay T_i called the *threshold* of player i. Each player i has only two strategies, namely a strategy $S_i^{\text{out}} = \{r_i\}$ with $r_i \in \mathcal{R}_{\text{out}}$, and a strategy $S_i^{\text{in}} \subseteq \mathcal{R}_{\text{in}}$. The preferences of player i can be described in a simple and intuitive

way: Player i prefers strategy S_i^{in} to strategy S_i^{out} if the delay of S_i^{in} is smaller than the threshold T_i. *Quadratic threshold games* are a subclass of threshold games in which the set $\mathcal{R}_{\text{in}}$ contains exactly one resource r_{ij} for every unordered pair of players $\{i, j\} \subseteq \mathcal{N}$. For every player $i \in \mathcal{N}$ of a quadratic threshold game, $S_i^{\text{in}} = \{r_{ij} \mid j \in \mathcal{N}, j \neq i\}$.

We show that finding Nash equilibria in quadratic threshold games is PLS-complete despite the simple structure of these games. Our proof is by a reduction from MAXCUT with the *flip*-neighborhood. Consider an instance of the MAXCUT problem that, w. l. o. g., consists of a complete weighted graph $G = (V, E)$ with non-negative edge weights w_{ij}. The local search version of MAXCUT can be described as a game, the so-called *party affiliation game*, in which players correspond to nodes that can choose whether they belong to a set A or a set B. Edges reflect some symmetric kind of anti-sympathy, that is, a node seeks to choose the set A or B such that the weighted number of edges leading to the other set is maximized. Schäffer and Yannakakis [13] show that computing a locally optimal cut is PLS-complete.

Theorem 4.1. *Computing a Nash equilibrium of a quadratic threshold game with non-decreasing delay functions is* PLS*-complete.*

Proof. The preferences of a player in the party affiliation game can be described in the following way that points out what could be a suitable threshold for her. For player i, let W_i denote the sum of the weights of all of its incident edges and $W_i^{(B)}$ the sum of the weights of the edges that connect i with nodes in class B. Player i prefers strategy A to strategy B if $W_i^{(B)} > \frac{1}{2}W_i$, she prefers strategy B to strategy A if $W_i^{(B)} < \frac{1}{2}W_i$, and she is indifferent if $W_i^{(B)} = \frac{1}{2}W_i$.

Now we show how to represent the party affiliation game in form of a quadratic threshold game. Both games involve the same number of players. We identify the players in the two games. With each edge $e = \{i, j\}$, we associate the resource $r_{ij} \in \mathcal{R}_{\text{in}}$. The delay of this resource is 0 if the resource is used by only one player and its delay is w_{ij} if it is used by two players. We identify strategy B of player i in the party affiliation game with strategy S_i^{in} in the threshold game. Player i's strategy A in the party affiliation game corresponds to strategy S_i^{out} in the threshold game, and the delay of this strategy is $T_i = \frac{1}{2}W_i$. Observe that the players' preferences coincide with the preferences in the party affiliation game. Hence, there is a one-to-one correspondence between the transition graphs of both games so that our construction yields an isomorphic PLS-reduction. $\square$

In the following, we will use quadratic threshold games as the starting point for further PLS-reductions. For some of these reductions, it will be helpful to make some assumptions on the delay functions.

Remark 4.2. *Without loss of generality, each resource $r \in \mathcal{R}_{in}$ in a quadratic threshold game with non-decreasing delay functions has a linear delay function of the form $d_r(k) = a_r k$ with $a_r > 0$. Furthermore, all thresholds can be assumed to be positive.*

In the proof of Theorem 4.1, the delay function of a resource $r_{ij} \in \mathcal{R}_{\text{in}}$ has the form $d_{r_{ij}}(k) = w_{ij}k - w_{ij}$. The preferences of the players are not affected by adding w_{ij} to each delay function $d_{r_{ij}}$, which then becomes $d_{r_{ij}}(k) = w_{ij}k$, if one simultaneously increases each threshold T_i by $\sum_{j \neq i} w_{ij} = W_i$. After this transformation every resource r has a delay
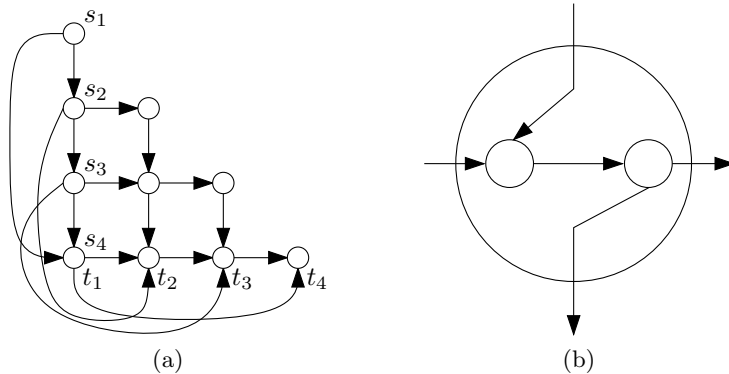
Figure 1: Illustration of the proof of Theorem 4.3.

function of the form $d_r(k) = a_r k$. Every resource r with $a_r = 0$ has constant delay 0 and hence, can be removed from the game without changing the preferences of the players.

4.2 Network Congestion Games

In a *network congestion game*, we are given a directed graph and, for each player, a source and a destination node. Every player seeks for a minimum delay path connecting her source with her destination. The delay of an edge depends on the number of players using that edge. Typically, it is assumed that the delay functions are non-decreasing. Fabrikant et al. [4] already proved that computing Nash equilibria of such network congestion games is PLS-complete. In the following, we present a much simpler proof for this. In contrast to the previous proof, our proof also holds for linear delay functions and it can be extended towards undirected networks.

Theorem 4.3. *Computing a Nash equilibrium for a network congestion game with non-decreasing linear delay functions is* PLS-*complete.*

Proof. Let Γ be a quadratic threshold game. We map Γ to a network congestion game as follows. The network consists of the lower-left triangle of an $n \times n$ grid (including the vertices on the diagonal) in which the vertical edges are directed downwards and the horizontal edges are directed from left to right. For every player i in Γ, we introduce a player i in the network congestion game whose source node s_i is the i-th node (from top to bottom) in the first column and whose target node t_i is the i-th node (from left to right) in the last row. For every player $i \in \mathcal{N}$, we add an edge from s_i to t_i, called *threshold edge*. Note that, due to the directions of the grid edges, the threshold edge of player i can only be used by player i. Figure 4.2 (a) illustrates our construction in the case of 4 players.

   Our first goal is to define delay functions in such a way that there are only two relevant strategies for player i: the threshold edge (s_i, t_i) and the *row-column path* from s_i to t_i, i.e., the path from s_i along the edges of row i until column i and then along the edges of column i to t_i. All other paths shall have such high delays that they are dominated by these two paths, regardless of the other players' choices. We achieve this goal by assigning the constant delay function 0 to all vertical edges and the constant delay function $D \cdot i$ to all horizontal edges in row i, where D denotes a large integer. Furthermore, for the

16

time being, we assume that the threshold edge (s_i, t_i) has the constant delay $D \cdot i \cdot (i-1)$. This way, each player i has only two undominated strategies: its threshold edge or its row-column path. The delays of these two alternative routes are so far identical.

Now we define additional delay functions for the nodes, that is, we view also the nodes as resources. (Figure 4.2 (b) describes how the nodes can be replaced by gadgets such that all resources are edges.) For $1 \leq i < j \leq n$, the node in column i and row j is identified with the resource $r_{ij} \in \mathcal{R}_{\text{in}}$ in the quadratic threshold game. In particular, we assume that the node has the same delay function as the corresponding resource in the threshold game. This way, the row-column path of player i corresponds to the strategy S_i^{in} of the threshold game. Furthermore, we increase the delay on the threshold edge of player i from $D \cdot i \cdot (i-1)$ to $D \cdot i \cdot (i-1) + T_i$, where T_i is the delay of resource $r_i \in \mathcal{R}_{\text{in}}$ in the threshold game. This way, the threshold edge of player i corresponds to the strategy S_i^{out} of the threshold game.

If we choose D larger than the sum of all delays in the threshold game, then for every player all strategies except for her row-column path and her threshold edge are dominated and, hence, can be ignored. Now, the remaining strategy spaces of the players and the corresponding delay functions are isomorphic to the strategies and delay functions of the threshold game. In particular, also the Nash equilibria of the two games coincide. Thus, our construction is a PLS-reduction. Moreover, one can easily verify that it is an embedding PLS-reduction. □

It is not difficult to modify the reduction above so that the linear delay functions have offset 0. In fact, one only needs to replace the constant delay functions of the form $d_e(n_e) = A$ by linear delay functions $d_e(n_e) = A \cdot n_e$. Thus, even congestion games in networks with *link speeds* are PLS-complete. Next we consider network congestion games with undirected edges and linear delay functions and prove that computing a Nash equilibrium remains PLS-complete for these games.

Theorem 4.4. *Computing a Nash equilibrium for a network congestion game with undirected edges and non-decreasing linear delay functions is* PLS-*complete.*

Proof. We give a PLS-reduction from quadratic threshold games to network congestion games with undirected edges and linear delay functions. The reduction is similar to the one in Theorem 4.3 except that we slightly change the structure of the network and that we adapt the delay functions of the edges accordingly.

Let Γ be a quadratic threshold game. The undirected graph that we construct has the same structure as in the case of networks with directed edges, except that we remove the directions of the edges. Moreover we split every threshold edge $\{s_i, t_i\}$ into two edges by introducing a node s_i', i.e., we introduce the edges $e_i^s = \{s_i', s_i\}$ and $e_i^t = \{s_i', t_i\}$. Again, for every player i in Γ we introduce a player in the network congestion game. However, in this reduction the source node of player i is s_i' and her target node is t_i.

In the previous reduction we could force a player to decide between the threshold edge and the row-column path by considering directed edges and carefully designed delay functions. Now we have to achieve the same effect with the delay functions only. We do not change the delay functions of the nodes $v_{i,j}$ and of the horizontal edges. Thus, a horizontal edge in the i-th row has constant delay $i \cdot D$, and the delay function of node

17

$v_{i,j}$ equals the delay function of the resource $r_{i,j} \in \mathcal{R}_{\mathrm{out}}$. We change the delay function of every vertical edge from 0 to $n^n D$. Additionally, we set the delay of the edge $e_i^s = \{s_i, s_i'\}$ to $n^{6n}D$.

We claim that for every player i, if one excludes the direct edge between s_i' and t_i, the only path connecting s_i' and t_i which can be a best response is the row-column path. Let $D_i^{\max}$ denote the maximal delay that can occur on this path without taking into account the edge $\{s_i', s_i\}$ but including the delay caused by the nodes on the path. We can bound this delay by

$$D_i^{\max} \quad < \quad (i-1) \cdot i \cdot D + (n-i)n^n D + D \ .$$

If player i chooses any other path connecting s_i' and t_i, then she either passes a node s_j' with $j \neq i$, or she allocates more than $n - i$ vertical edges, or she allocates a horizontal edge in a row $j > i$.

In the first case, the delay caused by the edge $\{s_j', s_j\}$ is at least $n^{6n}D$. This delay is larger than $D_i^{\max}$ and hence choosing such a path cannot be a best response. In the second case, the delay is at least $(n - i + 1)n^n D$ which is also larger than $D_i^{\max}$. Finally, consider the third case and assume that player i allocates $n - i$ vertical edges but at least one edge from a row $j > i$. Then her delay is at least

$$(n-i)n^n D + (i-2)iD + jD \geq (n-i)n^n D + (i-1)iD + D,$$

which is also larger than $D_i^{\max}$.

Finally, we choose the delay of the edge e_i^t to be

$$T_i + n^{6n}D + (n-i)n^n D + (i-1)iD \ ,$$

where T_i denotes the threshold of player i in the given quadratic threshold game.

Now assume that every player plays a best response and hence, either uses the direct edge between s_i' and t_i or the row-column path. Observe that under this assumption, the delay of edge e_i^t equals the threshold of player i plus the delay caused by the grid edges of the aforementioned path and the edge $\{s_i', s_i\}$. Hence by the same arguments as in Theorem 4.3, a Nash equilibrium of the constructed network congestion game corresponds to a Nash equilibrium of the given quadratic threshold game. Moreover, this reduction is an embedding PLS-reduction. □

In symmetric network congestion games, a Nash equilibrium can be found in polynomial time [4]. Nonetheless, selfish players do not necessarily find an equilibrium in polynomial time.

Theorem 4.5. *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}, polynomially bounded network size, and linear delay functions 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 because the reductions presented in the

proofs of Theorems 4.3 and 4.4 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 $d_e(n_e) = n_e \cdot D$ 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 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. $\square$

The simulation of asymmetric networks by symmetric ones also implies the following theorem.

Theorem 4.6. *In network congestion games (with directed or undirected edges) with a common source and possibly different sinks (or vice versa), and with non-decreasing delay functions, a Nash equilibrium can be found in polynomial time.*

Proof. We use the same simulation as in the proof of Theorem 4.5. Assume that a network with a common source and $k \leq n$ different sinks is given. That is, different players may share a common sink. In the following, let k_s be the number of players sharing sink s. We make the network symmetric by introducing a new common sink and connecting this new sink to each original sink s by an edge with delay function d_e such that $d_e(n_e) = 0$ for $1 \leq n_e \leq k_s$ and $d_e(n_e) = n_e \cdot D$ for $n_e > k_s$. Again, D is larger than the sum of all delays in the original game. Due to Fabrikant et al. [4], a Nash equilibrium in this symmetric network congestion game with n players can be computed in polynomial time. Observe that each of the new edges connecting one of the original sinks with the new sink is used by exactly k_s player in every Nash equilibrium. Hence, every equilibrium of the symmetric game can be transformed into a Nash equilibrium of the original game in polynomial time. $\square$

4.3 Market Sharing Games

Market Sharing games have been introduced by Goemans et al. [5] to model non-cooperative content distribution in wireless networks. An instance of a market sharing game consists of a set $\mathcal{N} = \{1, \ldots, n\}$ of players, a set $\mathcal{M}$ with $|\mathcal{M}| = l$ of markets, and a bipartite graph $G = (\mathcal{N} \cup \mathcal{M}, E)$. An edge between player i and market m indicates that player i is interested in market m. Furthermore, for each market m, costs c_m and a so-called query rate $q_m \in \mathbb{N}$ are given, and, for each player i, a budget B_i is specified. The query rate q_m determines the payoff of market m which is equally distributed among the players who have allocated that market, i. e., the payoff function of market m is given by $p_m(n_m) = q_m/n_m$. 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 $m \in \mathcal{M}'$, $(i, m) \in E$ and $\sum_{m \in \mathcal{M}'} c_m \leq B_i$. Observe that the set of strategies has a knapsack-like structure. The players are now interested in allocating a set of markets M' with maximum payoff. Thus, we define the *delay of a market* to be equal to its negative payoff.

If the costs of all markets are equal to 1, a market sharing game is called *uniform*. Goemans et al. [5] 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 can allocate an arbitrary subset of the markets she is interested in of size at most B_i. Hence Σ_i is a B_i-*uniform matroid*. Since all query rates are positive, only bases of this matroid can be best responses. Consequently, we can apply Theorem 2.5 to obtain the following theorem.

Theorem 4.7. *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 computing a best response corresponds to solving an instance of the knapsack problem. As a consequence, 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 m_i or a set of markets $\{m_i^{(1)}, \ldots, m_i^{(k)}\}$ by setting the costs of market m_i to k, the costs of each market $m_i^{(j)}$ to one, and the budget of player i to k. This way, the only possible best responses of player i are the strategies $\{m_i\}$ and $\{m_i^{(1)}, \ldots, m_i^{(k)}\}$, regardless of the strategies of the other players. This observation shows how to implement $(1, k)$-exchanges in market sharing games, which is the main idea that is needed to prove that computing a Nash equilibrium of a market sharing game with polynomially bounded costs is PLS-complete.

Theorem 4.8. *It is* PLS-*complete to compute a Nash equilibrium for market sharing games with polynomially bounded costs.*

Proof. We give a PLS-reduction from quadratic threshold games. Let Γ be a quadratic threshold game. Because of Remark 4.2, we can assume, w. l. o. g., that each resource $r \in \mathcal{R}_{\text{in}}$ has a linear delay function of the form $d_r(k) = a_r k$ with $a_r > 0$.

Following the arguments from the discussion about $(1, k)$-exchanges above, we construct a market sharing game Γ_M. For every resource $r_{ij} \in \mathcal{R}_{\text{in}}$, we introduce a market m_{ij} with cost 1 and query rate $q_{m_{ij}} = 2a_{r_{ij}}$. Furthermore, for every resource $r_i \in \mathcal{R}_{\text{out}}$, we introduce a market m_i with cost $|S_i^{\text{in}}|$ and query rate $3P_i - T_i$ where P_i denotes the sum of the coefficients $a_{r_{ij}}$ of the resources $r_{ij} \in S_i^{\text{in}}$. Observe that we can assume w. l. o. g. that $3P_i - T_i > 0$ since otherwise S_i^{in} is always the only best response for player i in Γ and hence, player i can be removed from the game. For every player i of Γ, there is also a player i in the market sharing game. This player has the budget $B_i = |S_i^{\text{in}}|$ and is interested in all markets that correspond to the resources in $S_i^{\text{out}} \cup S_i^{\text{in}}$. Observe that this construction yields a market sharing game with polynomially bounded cost.

Now let S be an arbitrary state of Γ_M. From S we construct a state $\tilde{S}$ of the quadratic threshold game as follows. If player i participates in market m_i, then we set the corresponding threshold game player i to its strategy S_i^{out}, otherwise to strategy S_i^{in}.

Fix a player i in Γ, let $\mathcal{R}_i^1$ denote the resources in S_i^{in} that she allocates alone in state $\tilde{S} \oplus S_i^{\text{in}}$, and let $\mathcal{R}_i^2$ denote the resources in S_i^{in} that she shares with another player in that state. Then her delay in state $\tilde{S} \oplus S_i^{\text{in}}$ can be written as $\sum_{r \in \mathcal{R}_i^1} a_r + 2\sum_{r \in \mathcal{R}_i^2} a_r = P_i + \sum_{r \in \mathcal{R}_i^2} a_r$. The strategy S_i^{in} is a best response in state $\tilde{S}$ if and only if $P_i + \sum_{r \in \mathcal{R}_i^2} a_r \leq T_i$. The payoff player i receives in state S when choosing all markets m_{ij} she is interested in can be written as $2\sum_{r \in \mathcal{R}_i^1} a_r + \sum_{r \in \mathcal{R}_i^2} a_r = P_i + \sum_{r \in \mathcal{R}_i^1} a_r$. This is a best response if and only if $P_i + \sum_{r \in \mathcal{R}_i^1} a_r \geq 3P_i - T_i$ which is equivalent to $P_i + \sum_{r \in \mathcal{R}_i^2} a_r \leq T_i$. Thus, S is a Nash equilibrium if and only if $\tilde{S}$ is a Nash equilibrium. Moreover, the reduction is an embedding reduction. □

4.4 Overlay Network Design Games

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. [15] 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 arborescence 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 $d_e \colon \mathbb{N} \to \mathbb{N}$ 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 $e = (u, v) \in E_i$ corresponds to the fixed routing path in the network G that is specified for the pair of nodes (u, v). In particular, its delay equals the delay of this 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 4.9. *The problem of finding a Nash equilibrium in an overlay network design game with linear delay functions is* PLS*-complete.*

Proof. We give a PLS-reduction from quadratic threshold congestion games to overlay network design games. As in the proof of Theorem 4.3, we use the lower-left triangle of an $n \times n$-grid as basis of our construction, but now with undirected edges, and we use the identifiers $s_1, \ldots, s_n, t_1, \ldots, t_n$, and $v_{i,j}$ 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), we add an edge (s_i, t_i') with delay function $d_{s_i, t_i'}(n_r) = n_r \cdot T_i$. In the network, the routing path between s_i and t_i is defined to be the row-colum path as in the proof of Theorem 4.3. 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 quadratic threshold game, we define a player in the overlay network design game with $V_i = \{s_i, t_i, t_i'\}$. Using the assumptions from Remark 4.2, our construction yields an overlay network design game with linear delay functions without offsets.

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 4.3, this shows that it is PLS-complete to find a Nash equilibrium in an overlay network design game. Moreover, observe that the reduction is embedding since the subgraph of the transition graph of the network design game that contains exactly those states in which every player i uses the edge (t_i, t'_i) is isomorphic to the transition graph of the quadratic threshold game, contains all local optima, and has no outgoing edges. $\square$

5   Conclusions

We investigated the impact of combinatorial structure on congestion games. On the positive side, we showed that best response sequences in matroid games have at most polynomial length. The crucial property leading to this result is that best responses in matroid games can be divided into exchanges of single resources, so-called $(1,1)$-exchanges. Moreover, we showed that matroids are the only combinatorial structure that can guarantee that all best response sequences are short. In order to see this, we first observed that in every non-matroid structure $(1,2)$-exchanges are possible and then we constructed instances of congestion games with exponentially long best response sequences using these $(1,2)$-exchanges as basic building blocks.

If one allows $(1,k)$-exchanges for arbitrary k, then one can even construct congestion games for which it is PLS-complete to find a Nash equilibrium. However, in our constructions k has to grow linearly with the number of players. It is an open question how large k has to be chosen in order to prove PLS-completeness. Since we showed that finding a Nash equilibrium in threshold congestion games is PLS-complete by a reduction from MaxCut, this question is closely related to the minimal node degree for which MaxCut is PLS-complete. To the best of our knowledge, there is still a considerable gap between the known results. On the one hand, the degree of the vertices in the MaxCut instances constructed in the PLS-completeness proof in [8] grows linearly with the number of vertices. On the other hand, Poljak [11] gives a polynomial time algorithm to find a locally optimal partition for cubic graphs.

References

[1] Heiner Ackermann, Heiko Röglin, and Berthold Vöcking. Pure Nash equilibria in player-specific and weighted congestion games. In *WINE*, pages 50–61, 2006.

[2] 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.

[3] Deeparnab Chakrabarty, Aranyak Mehta, and Viswanath Nagarajan. Fairness and optimality in congestion games. In *Proceedings of the 6th ACM conference on Electronic commerce (EC)*, pages 52–57, New York, NY, USA, 2005. ACM Press.

[4] 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.

[5] 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.

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

[7] 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.

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

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

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

[11] Svatopluk Poljak. Integer linear programs and local search for max-cut. *SIAM J. Comput.*, 24(4):822–839, 1995.

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

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

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

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

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