The Complexity of Equilibria in Cost Sharing Games

Vasilis Syrgkanis*

Department of Computer Science, Cornell University, Ithaca, NY 14853
vasilis@cs.cornell.edu

Abstract. We study Congestion Games with non-increasing cost functions (Cost Sharing Games) from a complexity perspective and resolve their computational hardness, which has been an open question. Specifically we prove that when the cost functions have the form $f(x) = c_r/x$ (Fair Cost Allocation) then it is PLS-complete to compute a Pure Nash Equilibrium even in the case where strategies of the players are paths on a directed network. For cost functions of the form $f(x) = c_r(x)/x$, where $c_r(x)$ is a non-decreasing concave function we also prove PLS-completeness in undirected networks. Thus we extend the results of [7, 1] to the non-increasing case. For the case of Matroid Cost Sharing Games, where tractability of Pure Nash Equilibria is known by [1] we give a greedy polynomial time algorithm that computes a Pure Nash Equilibrium with social cost at most the potential of the optimal strategy profile. Hence, for this class of games we give a polynomial time version of the Potential Method introduced in [2] for bounding the Price of Stability.

Keywords: Cost Sharing, PLS-completeness, Price of Stability, Congestion Games.

1 Introduction

The rapid and overwhelming expansion of the Internet has transformed it into a completely new economic arena where a large number of self-interested players interact. The lack of central coordination has rendered classic optimization problems insufficient to capture Internet interactions and has given rise to new game-theoretic models. In this work we study a general such model, namely Congestion Games with non-increasing cost functions (Cost Sharing Games), from a complexity perspective and we resolve the computational hardness of computing a Pure Nash Equilibrium (PNE) in such games, which has been an open problem. The computational hardness is an important aspect of an equilibrium concept since it indicates whether it is a reasonable outcome in real world settings.

We start with a motivating example: a group of Internet Service Providers (ISP) wants to create a new network on a set of nodes (possibly different set for each provider). Each ISP's goal is that any two of his nodes are connected

* Supported in part by NSF grant CCF-0729006.

A. Saberi (Ed.): WINE 2010, LNCS 6484, pp. 366–377, 2010.

by a path. For practical reasons, each provider can build edges only between two nodes in his set and his clients can use a link only if the ISP helped build it. Moreover, we assume that ISPs are clever enough and when they decide to build the same link as others then they all build one link and share the cost. The moment we add this last specification, the problem faced by an ISP is no longer an optimization problem and the setting becomes a game which from now on we will call the ISP Network Creation Game. ISP Network Creation Games can be easily modeled as Cost Sharing Games.

Congestion Games in general has been a widely studied game theoretic model. In Congestion Games a set of players allocate some set of shared resources. The cost incurred from using a resource is a function of the number of players that have allocated the resource and the total cost of a player is the sum of his costs on all the resources he has allocated. A reasonable outcome of such a setting is a Pure Nash Equilibrium (PNE): a strategy profile such that no player can profit from deviating unilaterally. In a seminal paper, Rosenthal [13] gives a proof that Congestion Games always possess a PNE. To achieve this, he introduces a potential function and shows that the change in the potential induced by a unilateral move of some player is equal to the change of that player's utility. Several aspects of the PNE of Congestion Games have been studied in the literature.

An interesting research area has been the complexity of computing a PNE in Congestion Games. In a seminal paper Fabrikant et al. [7] proved that the above problem is PLS-complete even in the case where the strategies of the players are paths in a directed network. Later, Ackermann et al. [1] extended the above result to the case of undirected networks with linear cost functions. However, both results use cost functions on the resources that are non-decreasing (delays) and do not carry over to Cost Sharing Games. The complexity of computing a PNE in Cost Sharing Games has been an open question.

Another interesting line of research has been measuring the inefficiency that arises from selfishness. An important concept in that direction (especially in the case of Cost Sharing Games) has been the Price of Stability (PoS), which is the ratio of the quality (sum of players' costs) of the best PNE over the socially optimal outcome ([2]). One major motivation for the PoS is that it is the socially optimal solution subject to the constraint of unilateral stability. If there was a third-party that could propose to players a solution to their problem, then the optimal stable solution he could propose would be the best PNE. This motivation raises an interesting open question: Given an upper bound on the PoS for a class of games, is there a polynomial-time algorithm for computing a PNE with cost comparable to that bound?

In this work we make significant progress in both directions described above. We prove the first PLS-hardness results for Cost Sharing Games. Our results show that the non-increasing case is not easier than the non-decreasing. Moreover, we give the first polynomial-time algorithm that computes a PNE with quality equal to the known bound on the PoS for a significant class of Cost Sharing Games that contains, for example, the ISP Network Creation Game.

Results

- Our first main result is that a greedy approach leads to a polynomial time algorithm that computes a PNE of any Matroid Cost Sharing Game, with cost equal to the potential of the socially optimal solution. The quality of such a PNE is no worse than any bound on the PoS that can be proved via the Potential Method. Hence, for this class we give a polynomial time equivalent to the Potential Method. Matroid Cost Sharing Games are Cost Sharing Games where the strategy space of each player is exactly the set of bases of a player-specific matroid. The existence of algorithms like the one given here has been an interesting open question [18]. From previous work [5], we know that computing the global potential minimizer is NP-hard even for Singleton Cost Sharing Games. Also we note here that the same holds for the minimum social cost PNE. Hence it is surprising that we can achieve such an efficiency guarantee.
- The above result directly implies the logarithmic bound on the PoS for Matroid Cost Sharing Games with cost functions of the form $f(x) = c_r(x)/x$, where $c_r(x)$ is a nondecreasing concave function.
- For the case of Singleton Cost Sharing Games our algorithm does not output just a PNE but a Strong Nash Equilibrium. Hence this extends the results in [6] on the existence of Strong Nash Equilibria in Cost Sharing Games.
- Our second main result is that computing a PNE in the class of Network Cost Sharing Games where the cost functions come from the Shapley Cost Sharing Mechanism, $f(x) = c_r/x$ (Fair Cost Allocation) is PLS-complete. The hardness results are based on a tight PLS-reduction from MAX CUT. The result is not restrictive to Fair Cost Allocation and holds for almost any reasonable set of decreasing functions.
- The tightness of our reduction also shows that there exist instances of Network Cost Sharing Games with Fair Cost Allocation, where best response dynamics will certainly need exponential time to reach a PNE. This gives a negative answer to an interesting open question proposed in [2] of whether there exist a scheduling of best response moves that lead to a PNE in polynomially many steps.
- For cost functions of the form $f(x) = c_r(x)/x$, where $c_r(x)$ is a nondecreasing concave function we also prove PLS-completeness for the case of Undirected Network Cost Sharing. This result is not restricted to the above class of functions but generalizes to any class of cost functions that contains almost constant functions.
- The new techniques that we introduce can be used to simplify the existing reductions for the non-decreasing case.

Techniques. To prove our main hardness result we introduce a new class of Congestion Games called k-Congestable Congestion Games. In a k-Congestable Congestion Game the resources of any two strategies of a player are disjoint and each resource is contained in some strategy of at most k players. Thus at most k players can share a resource in any strategy profile. These games generalize k-Threshold Games introduced by Ackermann et al. [1].

We show how to reduce the computation of a PNE in a 2-Congestable Congestion Game with cost functions that satisfy certain assumptions, to the same problem in a Network Congestion Game with the same set of cost functions. If the class of cost functions is general enough to contain almost constant functions with arbitrary high cost, then we can reduce 2-Congestable Congestion Games to Undirected Network Games. We notice that the MAX CUT reduction of Fabrikant et al [7] constructs a 2-Congestable Congestion Game hence our techniques can be applied to simplify the PLS-completeness proofs for the non-decreasing case.

A lot of the proofs of our results had to be omitted due to space limitations. See the full version of the paper for the proofs that had to get deleted from this proceedings version.

Related Work

Complexity of Equilibria. Apart from the results mentioned in the introduction [7, 1], there has been several works on the relation between PLS and PNE. Skopalik et al [16] proves that even computing an approximate PNE is PLS complete for Congestion Games. Their techniques can also be used to prove PLS-completeness of approximate equilibria in Bottleneck Games (player cost is maximum of cost of allocated resources) as noted independently by Syrgkanis [17] and Harks et al [9].

For Cost Sharing Games Chekuri et al [5] prove that it is NP-hard to compute the global potential minimizer for Multicast Games with Fair Cost Allocation. Hansen et al [8] give an exponential sequence of best response moves for the case of Metric Facility Location Games and provide a polynomial time algorithm for computing approximate equilibria in that class.

On the positive side, Ieong et al. [10] give a dynamic programming algorithm for computing the optimal PNE in the class of Symmetric Singleton Games with arbitrary cost functions. Moreover, Ackermann et al [1] introduce Matroid Congestion Games as a class of games where best response dynamics converge in polynomially many steps.

Quality of Equilibria. Cost Sharing Games in the form studied in this work were introduced by Anshelevich et al. [2]. One of their main results is that the PoS is $O(\log(n))$ (where n is the number of players) for Cost Sharing Games where the cost functions have the form $f(x) = c_r(x)/x$, where $c_r(x)$ is a nondecreasing concave function. Their proof introduces the Potential Method, a way of bounding the PoS by showing that the global minimizer of Rosenthal's potential (which is a PNE) has cost close to the optimal.

Several other works have dealt with Cost Sharing Games from the perspective of bounding the inefficiency that arises from selfishness. Chekuri et al [5] study Multicast and Facility Location Games when players arrive sequentially and then perform best response. They prove that the quality of the resulting PNE is at most $O(\sqrt{n}\log^2 n)OPT$. Later Charikar et al. [4] improve this bound to $O(\log^3 n)OPT$ and also make progress for the case when best response and sequential arriving is interleaved. Epstein et al [6] study the quality of Strong Nash Equilibria of Cost Sharing Games with Fair Cost Allocation. Strong Nash Equilibria allow for group moves of players, therefore they are a solution concept robust to collusion. However, they do not always exist in Cost Sharing Games. When they exist Epstein et al [6] show that their worst case quality matches the PoS bound of H_n. Balcan et al [3] study Cost Sharing Games with Fair Cost Allocation under the perspective of Learning Agents. They prove that if players perform best response but at each step with a small fixed probability chose their strategy in a nearly optimal outcome, then the expected quality of the resulting PNE is $O(\log(n)\log(n|F|))OPT$, where $|F|$ is the number of resources.

2 Definitions and Notation

Definition 1. *A **Congestion Game**(CG), denoted* $\langle N, F, (S_i)_{i\in N}, (r_f)_{f\in F}\rangle$*, consists of: A set of N players and a set of facilities F. For each player i a set of strategies* $S_i \subseteq 2^F$*. For each facility f a cost function* $r_f(x)$*. Given a strategy profile s, the cost of a player i is* $C_i(s_i, s_{-i}) = \sum_{f\in s_i} r_f(n_f(s))$*, where* $n_f(s)$ *(congestion) is the number of players using facility f in strategy profile s. The Social Cost of s is:* $SC(s) = \sum_{i\in[N]} C_i(s)$ *and the Potential of s is:* $\Phi(s) = \sum_{f\in F}\sum_{k=1}^{n_f(s)} r_f(k)$*.*

Definition 2. *A **Cost Sharing Game** (CSG) is a CG where the facility cost functions* $r_f(x)$ *are non-increasing. Any Cost Sharing Game may also be augmented by the property of **Fair Cost Allocation** which means that the cost functions have the specific form of* $r_f(x) = \frac{c_f}{x}$*.*

Definition 3. *A **Network Cost Sharing Game** is a CSG, where the strategy space of each player i is the set of all possible paths between two nodes* (s_i, t_i) *on a directed network* $G = (V, E)$*. If we assume an undirected network then we have the class of **Undirected Network Cost Sharing Games**. If all players share a common sink then we have the case of a **Multicast Cost Sharing Game** either on a directed or undirected network.*

Definition 4. *A **Matroid Cost Sharing Game** is a CSG, where for each player* $i \in [N]$*,* S_i *is the set of bases of a matroid* $\mathcal{M}_i = (F, l_i)$*, where* l_i *is the set of independent sets [15]. Additionally we denote by* $rk(G) = max_{i\in[N]} rk(\mathcal{M}_i)$ *the rank of the game G, where* $rk(\mathcal{M}_i)$ *is the rank of matroid* $rk(\mathcal{M}_i)$*.*

Definition 5. *A **Singleton Cost Sharing Game** is a CSG, where for each player* $i \in [N]$*,* S_i *consists of singleton sets. For this class of games we will use an equivalent model that consists of: n jobs and m machines and an arbitrary bipartite graph* $\mathcal{G}$ *on nodes* $[n] \cup [m]$*. The jobs are the players of the game and their strategies is to choose one of the machines they are connected to in* $\mathcal{G}$*. We denote with* M_j *the set of neighbors of job j, i.e. the set of possible machines job j can choose from. We denote with* N_i *the set of neighbors of machine i, i.e. the set of jobs machine i can be picked by.*

3 Computing a Good Pure Nash Equilibrium

Matroid Cost Sharing Games is a subclass of Matroid Congestion Games, hence by Ackermann et al. [1] we have that best response dynamics converge to a PNE in at most $n^2m\ rk(G)$ steps. Thus one polynomial algorithm that gives a PNE starts from an arbitrary configuration and performs a best response in each step.

However, one interesting question is whether we can find a good quality PNE, for example the one that globally minimizes the potential function or at least a PNE with good social cost characteristics. Chekuri et al. [5] prove that computing the global potential minimizer for the class of Singleton Cost Sharing Games with cost functions of the form $f(x) = 1/x$ is NP-hard. The proof is based on a gap introducing reduction by Lund and Yannakakis [12]. We remark here that this reduction can also be used to prove that computing the socially optimal PNE is NP-hard.

Nevertheless, these hardness results do not exclude the possibility of computing a PNE with social cost comparable with the bound on the PoS produced by the Potential Method. Specifically the Potential Method works as follows: Suppose that for any profile s: $SC(s) \leq \Phi(s) \leq \alpha SC(s)$. Then if we find the global potential minimizer $\hat{s}$ and denoting with s^* the optimal, we get that: $SC(\hat{s}) \leq \Phi(\hat{s}) \leq \Phi(s^*) \leq \alpha SC(s^*)$, hence the PoS is at most α. Now we know that computing the global potential or social cost minimizer is NP-hard, however if we manage to find a PNE $\hat{s}'$ such that: $SC(\hat{s}') \leq \Phi(s^*)$, then we would have: $SC(\hat{s}') \leq \alpha SC(s)$ and we would get the same upper bound. This is the guarantee that we will achieve for the algorithms that follow.

3.1 Singleton Cost Sharing Games

In this section we present the polynomial time algorithm that computes a good PNE for the class of Singleton Cost Sharing Games. We start from Singleton Cost Sharing Games to give a clear intuition for the case of Matroid Cost Sharing Games that will be a generalization of the results presented in this section.

Singleton Cost Sharing Games can also be viewed as a Multicast Cost Sharing Game on a directed network. Given an instance of our model we create a multicast game as follows: Set a common sink t. Create a machine node v_i for each machine i and connect it with t with an edge of cost r_i. Create one source node s_j for each job j. Then for each $j \in N_i$ create an edge of cost 0 from source node s_j to machine node v_i. It could also be viewed as a Multicast Cost Sharing Game on an undirected network. Instead of setting the edge costs of edges (s_j, v_i) to 0 we set it to some huge number q. However, the Price of Anarchy and PoS bounds do not carry over in this case.

In the special case of Fair Cost Allocation, the social cost is the sum of the costs of the machines used and the optimization problem of computing a strategy profile with minimum social cost is a problem equivalent to SET COVER.

Our algorithm (Alg. 1) works as follows: Each time pick the machine that incurs the minimum player cost if it is assigned all the possible unassigned jobs and assign to that machine all possible jobs. We iterate until all jobs are assigned.

Algorithm 1. Poly-time algorithm for good PNE

Require: An instance of $G = \langle \mathcal{G} = ([n] \cup [m], E), (r_i)_{i\in[m]}\rangle$
$\mathcal{G}^1 = \mathcal{G}$
for $t = 1$ to m **do**
Let $d_i^t = |N_i^t|$ be the degree of machine i in $\mathcal{G}^t$. Let $i^t = \arg\min_{i\in[m]} r_i(d_i^t)$
For all $j \in N_{i^t}^t$ set $s_j = i^t$
Remove nodes $N_{i^t}^t \cup i^t$ from $\mathcal{G}^t$ to obtain $\mathcal{G}^{t+1}$
end for
return Strategy profile s

Theorem 1. *For any instance of Singleton Cost Sharing Games, Algorithm (1) computes a PNE that is as good as the potential of the optimal allocation.*

Sketch of Proof. Suppose in the end of the algorithm, some job j wants to move from his current machine i to some i'. Assume j was assigned to i at time step t_0, i.e. $i = i^{t_0}$. Therefore it was not connected to any machine i^t for $t < t_0$. Thus i' must correspond to some machine i^{t_1} for $t_1 > t_0$. Since j was not assigned to i' it means that at t_0, the degree of i' dropped by at least 1. Moreover, at each subsequent time step the degree of machine i' can only drop. Thus $d_{i'}^{t_1} \leq d_{i'}^{t_0} - 1$. Moreover, since i was selected at t_0 it means that: $r_i(d_i^{t_0}) \leq r_{i'}(d_{i'}^{t_0}) \leq r_{i'}(d_{i'}^{t_1} + 1)$, where the last inequality holds from the fact that $d_{i'}^{t_0} \geq d_{i'}^{t_1} + 1$ and r_i are non-increasing functions. In the end of the algorithm $n_i = d_i^{t_0}$ and $n_{i'} = d_{i'}^{t_1}$, hence $r_i(n_i) \leq r_{i'}(n_{i'} + 1)$, which is a contradiction.

For the efficiency guarrantee we work with a price scheme. Consider a machine i in the optimum solution and assume d players are assigned to it in OPT. Order these players in the order that the algorithm assigns them to some other machine. When the j-th player was assigned to another machine i' we know that at least $d - j + 1$ players can still be assigned to i. Since we assign all possible players to i' and we choose the machine with the smallest possible player cost we know that j-th player pays a cost of at most $r_i(d - j + 1)$ for being assigned to i'. Iterating this for all j and for all machines in the optimum solution we get the desired result. □

For the case of Fair Cost Allocation, Algorithm (1) is exactly the greedy H_n-approximation for SET COVER. Hence we immediately get a good efficiency guarantee. In addition it is interesting to notice that the tight example for the H_n-approximation given in Example 2.5 of [19] for the greedy approximation algorithm for SET COVER is the identical analogue of the tight example for the PoS given in [2].

We also note that the above algorithm actually computes a Strong Nash Equilibrium. For Matroid Cost Sharing Games such a guarantee cannot be achieved since the example of [6] that possesses no Strong Nash Equilibrium can be easily transformed into a Matroid Game.

3.2 Matroid Cost Sharing Games

In this section we present a generalization of Algorithm (1) that computes a good PNE for the class of Matroid Cost Sharing Games.

Algorithm 2. Poly-time algorithm for good PNE in Matroid Cost sharing Games

Require: An instance of $\langle N, F, (S_i)_{i \in [N]}, (r_f)_{f \in F} \rangle$
  $\forall i \in [N] : s_i^0 = \emptyset;\ \ \forall f \in F : N_f^0 = \{i \in [N] \mid f \in l_i\},\ d_f^0 = |N_f^0|,\ t = 0$
  while $\exists f \in F : d_f^t > 0$ **do**
    $f^t = \arg\min_{f \in F} r_f(d_f^t)$
    $\forall i \in N_{f^t}^t$ set $s_i^t = s_i^{t-1} \cup \{f^t\}$. $t = t + 1$
    $N_f^t = \{i \in [N] \mid s_i^{t-1} \cup \{f\} \in l_i\},\ d_f^t = |N_f^t|$
  end while
  return Strategy profile s^{t-1}

The algorithm (Alg. 2) works as follows: At each point we keep a temporary strategy for each player, starting from the empty strategy. At each iteration t we compute for each resource to how many players' strategy it could be added (d_f^t). Then we choose the resource that has minimum player cost if added to the strategy of all possible players ($\min_{f \in F} r_f(d_f^t)$) and we perform this addition. This happens until no player's strategy can be further augmented.

Assuming that checking whether some set is in l_i takes polynomial time in the size of the input, then it is clear that the above algorithm runs in polynomial time since the while loop is executed at most $n \cdot rk(G)$ times and during each time step we go over all the resources. For example the above property is true for the case where the strategy space of each player is the set of spanning trees on a set of nodes, like the ISP Network Creation Game.

Theorem 2. *For any Matroid Cost Sharing Game, Algorithm (2) computes a PNE with social cost at most the potential of the optimal allocation.*

Sketch of Proof. To prove that the resulting allocation is a PNE we use the matroid property that a base is minimum if and only if there is no $(1, 1)$ exchange of a facility that results to a better strategy. Then we argue that no profitable $(1, 1)$ exchange can exist in a player's strategy due to the way the algorithm works. To prove the efficiency guarantee we construct for every player a 1-1 mapping of the facilities in the algorithm's allocation and those in the optimal allocation with the following property: whenever a facility is assigned to the player then its mapping in the optimal allocation is also still an option at that time step. In this way we are able to simulate the same logic that we used in the proof of Theorem 1. □

4 Intractability of Cost Sharing Games

In this section we provide the PLS-hardness results. For a more detailed exposition of the class PLS and definitions and properties of PLS reductions the reader is redirected to the initial papers on PLS [11, 14] and to previous work on PLS hardness of congestion games [7, 1].

4.1 General Cost Sharing Games

We will prove that finding a PNE in the class of Cost Sharing Games is PLS-complete via a reduction from MAX CUT under the flip neighborhood.

Definition 6. *We say that a class of functions has the property $\mathcal{P}_1$ if for arbitrary $a > 0$ it contains a function $f(x)$ such that $f(1) = f(2) + a$.*

Theorem 3. *Computing a PNE for the class of Cost Sharing Games with a class of cost functions that has property $\mathcal{P}_1$ is PLS-complete.*

Proof. Assume an instance of MAX CUT on weighted graph $G = (V, E, (w_e)_{e\in E})$. Assume that $(i, j) \notin E \Rightarrow w_{ij} = 0$. We will create an instance of a Cost Sharing Game $\langle N, F, (S_i)_{i\in[N]}, (r_f)_{f\in F}\rangle$ such that from every PNE of the game we can construct in polynomial time a local maximum cut of the MAX CUT instance.

For each node $i \in V$ we add a player $P_i \in [N]$. We assume an ordering of the players $P_1, \dots, P_N$. For each unordered pair of players $\{i, j\}$ $(i < j)$ we add two facilities f^1_{ij} and f^2_{ij} in the set F, each with cost function r_{ij}, such that $r_{ij}(1) = r_{ij}(2) + w_{ij}$, which can be achieved due to the $\mathcal{P}_1$ property.

$$s_i^A = \{f^2_{ji} \mid j \in \{1 \dots i-1\}\} \cup \{f^1_{ij} \mid j \in \{i+1 \dots N\}\}$$
$$s_i^B = \{f^1_{ji} \mid j \in \{1 \dots i-1\}\} \cup \{f^2_{ij} \mid j \in \{i+1 \dots N\}\}$$

In other words for each pair of players $\{i, j\}$ if player i has facility f^1_{ij} in his s_i^A strategy then player j has facility f^2_{ij} in his s_j^A strategy and correspondingly for the B strategies.

Now given any strategy profile s we consider the following partition of the initial graph: If player $P_i \in N$ is playing s_i^A then place node i in partition $V_A(s)$ and to partition $V_B(s)$ otherwise. For every node $i \in V$ denote with $w_i = \sum_{j\in V} w_{ij}$, $w(i, V_A) = \sum_{j\in V_A} w_{ij}$, $w(i, V_B) = \sum_{j\in V_B} w_{ij}$.

Given any strategy profile s the cost of player P_i for playing each strategy is:

$$C_i(s_i^A, s_{-i}) = \sum_{j\in V_A(s)} r_{ij}(1) + \sum_{j\in V_B(s)} r_{ij}(2) = w(i, V_A(s)) + \sum_{j\neq i} r_{ij}(2)$$
$$C_i(s_i^B, s_{-i}) = \sum_{j\in V_A(s)} r_{ij}(2) + \sum_{j\in V_B(s)} r_{ij}(1) = w(i, V_B(s)) + \sum_{j\neq i} r_{ij}(2)$$

Thus if s is a PNE and P_i is playing s_i^A then: $C_i(s_i^A, s_{-i}) \leq C_i(s_i^B, s_{-i}) \implies w(i, V_A(s)) \leq w(i, V_B(s))$. Hence, switching node i from partition $V_A(s)$ to $V_B(s)$ will not increase the weight of the cut. Similarly if P_i is playing s_i^B we get the opposite inequality. Therefore, for any PNE s the corresponding partition $(V_A(s), V_B(s))$ is a local maximum of the initial MAX CUT instance. □

Corollary 1. *Computing a PNE for the class of Cost Sharing Games with Fair Cost Allocation is PLS-complete.*

4.2 Extending to Network Games

We will introduce k-Congestable Games. We will then notice that the game instance created in the MAX CUT reduction belongs to the class of 2-Congestable Cost Sharing Games with Fair Cost Allocation. We will then show how from any instance of a 2-Congestable Game we can create a Network Congestion Game that preserves the PNE.

Definition 7. *A k-Congestable Congestion Game is a Congestion Game where: (1) Any facility is used by at most k players. (2) The facilities on the different strategies of a player are disjoint. There is no restriction on the cost functions.*

Definition 8. *A class of cost functions has property $\mathcal{P}_2$ if for arbitrary $H > 0$ it contains a function $f(x)$ such that $\min_{k\in[1..N]} f(k) > H$ and any member of the class has bounded maximum in a finite integer range $[1..N]$.*

Theorem 4. *Given an instance of a 2-Congestable Congestion Game with cost functions from a class with property $\mathcal{P}_2$, we can create an instance of a Network Congestion Game on a directed network, where any PNE of the latter corresponds to a PNE of the former and the conversion can be computed in polynomial time. Moreover, the reduced game contains the same set of cost functions as the initial.*

It is easy to see that the game created in the MAX CUT reduction is a 2-Congestable Game and that the class of functions of the form $f(x) = c/x$ satisfy property $\mathcal{P}_2$ (see full version of the paper). Hence, we have the following corollary:

Corollary 2. *Computing a PNE in the class of Network Cost-Sharing Games with Fair Cost Allocation is PLS complete.*

An example of a reduction of MAX CUT on a three node graph to General Cost Sharing and Network Cost Sharing is depicted in Figure 1.

Now we describe for which classes of functions we can have PLS-completeness in undirected networks too (see full version for proofs).

Definition 9. *A class of functions has property $\mathcal{P}_3$ if for any $a, \epsilon > 0$ it contains a function $f(x)$ such that $f(1) = a$ and $\max_{k\in[1..N]} f(k) - \min_{k\in[1..N]} f(k) \leq \epsilon$.*

Theorem 5. *Given an instance of a 2-Congestable Congestion Game with cost functions from a class that has properties $\mathcal{P}_2$ and $\mathcal{P}_3$, we can create an instance of an Undirected Network Congestion Game, where any PNE of the latter corresponds to a PNE of the former and the conversion can be computed in polynomial time. Moreover, the reduced game contains the same set of cost functions as the initial.*

Corollary 3. *Computing a PNE in the class of Undirected Network Cost Sharing Games with functions of the form $f(x) = c_r(x)/x$, where $c_r(x)$ is a non-decreasing concave function, is PLS-complete.*

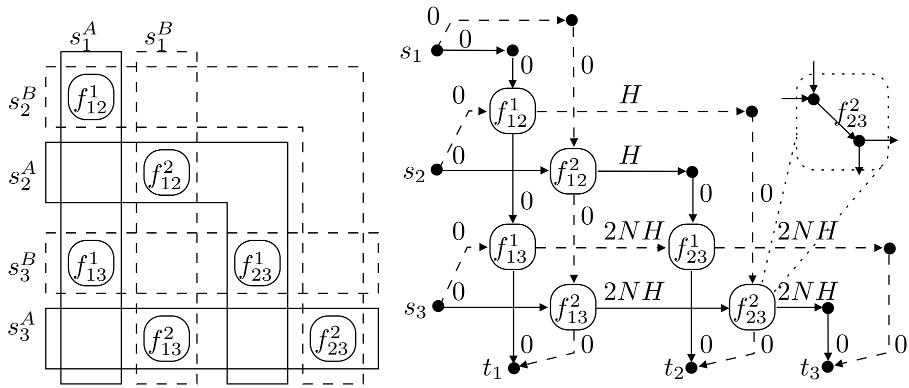

Fig. 1. Reduction from MAX CUT to Cost Sharing (left) and Network Cost Sharing (right) with Fair Cost Allocation. The facilities f_{ij}^k have cost the weight of the edge (i, j). The cost of the rest of the facilities-edges is depicted on the figure. H is a number much bigger than any cost imposed by facilities f_{ij}^k.

4.3 Tightness of PLS-Reductions

It is easy to observe that all the PLS reductions used in the previous sections are tight reductions as defined in [14]. From the initial works on PLS [11, 14] we know that tight reductions do not decrease the distance of an initial solution from a local optimum through local improvement steps. Moreover, we know that there exist instances of MAX CUT with initial configurations that have exponential distance from any local maximum. This directly implies that there exist instances of the class of games for which we prove PLS-completeness, with strategy profiles such that any sequence of best response moves needs exponential time to reach a PNE. Moreover, the tightness of our reductions shows that for the class of games we cope with it is PSPACE-complete to compute a PNE that is reachable from a specific initial strategy profile through best response moves.

5 Discussion and Further Results

Another interesting fact that might be useful in other reductions is the following:

Theorem 6. *Computing a PNE in General Congestion Games where all players have two strategies and each facility is used by at most two players can be reduced to computing a PNE of a 2-Congestable Congestion Game. If the initial game contains a cost function $r_f(x)$ then the reduced game might contain cost functions of the form $r_f(x + k)$ for arbitrary k.*

Last, we observe that our reductions from 2-congestable games also show how one can conclude PLS completeness of Undirected Network Congestion Games with linear cost functions, directly from the MAX CUT reduction of [7] without introducing 2-threshold congestion games. It is easy to observe that the Congestion Game created in the reduction of [7] is a 2-Congestable Game and linear functions is a class that satisfies properties $\mathcal{P}_1$, $\mathcal{P}_2$ and $\mathcal{P}_3$.

Acknowledgements. I would like to deeply thank Eva Tardos for the many fruitful and insightful discussions on the subject. I would also like to thank the reviewers for the helpful comments.

References

[1] Ackermann, H., Röglin, H., Vöcking, B.: On the impact of combinatorial structure on congestion games. J. ACM 55(6) (2008)
[2] Anshelevich, E., Dasgupta, A., Kleinberg, J.M., Tardos, É., Wexler, T., Roughgarden, T.: The price of stability for network design with fair cost allocation. In: FOCS, pp. 295–304. IEEE Computer Society, Los Alamitos (2004)
[3] Balcan, M.-F., Blum, A., Mansour, Y.: Circumventing the price of anarchy: Leading dynamics to good behavior. In: ICS, pp. 200–213 (2010)
[4] Charikar, M., Karloff, H.J., Mathieu, C., Naor, J., Saks, M.E.: Online multicast with egalitarian cost sharing. In: SPAA, pp. 70–76. ACM, New York (2008)
[5] Chekuri, C., Chuzhoy, J., Lewin-Eytan, L., Naor, J., Orda, A.: Non-cooperative multicast and facility location games. IEEE Journal on Selected Areas in Communications 25(6), 1193–1206 (2007)
[6] Epstein, A., Feldman, M., Mansour, Y.: Strong equilibrium in cost sharing connection games. In: EC, pp. 84–92. ACM, New York (2007)
[7] Fabrikant, A., Papadimitriou, C.H., Talwar, K.: The complexity of pure nash equilibria. In: STOC, pp. 604–612. ACM, New York (2004)
[8] Hansen, T.D., Telelis, O.: Improved bounds for facility location games with fair cost allocation. In: COCOA 2009. LNCS, vol. 5573, pp. 174–185. Springer, Heidelberg (2009)
[9] Harks, T., Hoefer, M., Klimm, M., Skopalik, A.: Computing pure nash and strong equilibria in bottleneck congestion games. In: de Berg, M., Meyer, U. (eds.) ESA 2010. LNCS, vol. 6347, pp. 29–38. Springer, Heidelberg (2010)
[10] Ieong, S., McGrew, R., Nudelman, E., Shoham, Y., Sun, Q.: Fast and compact: A simple class of congestion games. In: AAAI, pp. 489–494 (2005)
[11] Johnson, D.S., Papadimitriou, C.H., Yannakakis, M.: How easy is local search? J. Comput. Syst. Sci. 37(1), 79–100 (1988)
[12] Lund, C., Yannakakis, M.: On the hardness of approximating minimization problems. J. ACM 41(5), 960–981 (1994)
[13] Rosenthal, R.W.: A class of games possessing pure-strategy nash equilibria. International Journal of Game Theory 2(1), 65–67 (1973)
[14] Schäffer, A.A., Yannakakis, M.: Simple local search problems that are hard to solve. SIAM J. Comput. 20(1), 56–87 (1991)
[15] Schrijver, A.: Combinatorial Optimization: Polyhedra and Efficiency. Matroids, Trees, Stable Sets, vol. B (2003)
[16] Skopalik, A., Vöcking, B.: Inapproximability of pure nash equilibria. In: STOC, pp. 355–364. ACM, New York (2008)
[17] Syrgkanis, V.: Equilibria in Congestion Game Models. Undergraduate Diploma Thesis (2009), `http://www.cs.cornell.edu/~vasilis/thesis.pdf`
[18] Tardos, E., Kleinberg, J.: Algorithm Design, ch. 12. Addison-Wesley, Reading (2005)
[19] Vazirani, V.V.: Approximation Algorithms. Springer, Heidelberg (2001)