The Complexity of Pure Nash Equilibria in Weighted Max-Congestion Games

Carme Àlvarez and Guillem Francès *

alvarez@lsi.upc.edu gfrances@lsi.upc.edu
Universitat Politècnica de Catalunya, ALBCOM Research group.
Edifici Ω, Campus Nord. Jordi Girona, 1-3, Barcelona 08034, Spain.

Abstract. We study *Network Max-Congestion Games* (NMC games, for short), a class of network games where each player tries to minimize the *most congested* edge along the path he uses as strategy. We focus our study on the complexity of computing a pure Nash equilibria in weighted NMC games. We show that, for single-commodity games with non-decreasing delay functions, this problem is in P when either all the paths from the source to the target node are disjoint or all the delay functions are equal. For the general case, we prove that the computation of a PNE belongs to the complexity class PLS through a new technique based on *semi-potential* functions and a slightly modified definition of the usual local search neighborhood. We further apply this technique to a different class of games (which we call *Pareto-efficient*) with restricted cost functions. Finally, we also prove some PLS-hardness results, showing that computing a PNE for Pareto-efficient NMC games is indeed a PLS-complete problem.

1 Introduction

During the last years, a great effort has been devoted to the development of mathematical tools for the modelling of computer networks such as the Internet. Given the decentralized and non-cooperative nature of these networks, Game Theory [1] arises as one of the most suitable instruments to characterize the behaviour of its selfish users[1]. *Network Congestion Games*, for instance, provide a useful means of studying the behaviour of such users when they try to minimize the total congestion of the network paths through which they route their packets. One of the fundamental and most studied concepts of Game Theory is that of

* Both authors partially supported by FET pro-actives Integrated Project 15964 (AEOLUS) and by Spanish CICYT under grants TIN2005-09198-C02-02 (ASCE) and TIN2007-66523 (MEFOALDISI). The second author is also supported by a UPC research grant.

[1] As stated by C. Papadimitriou on his seminal paper [2], "... the mathematical tools and insights most appropriate for understanding the Internet may come from a fusion of algorithmic ideas with concepts and techniques from Mathematical Economics and Game Theory".

the *Nash Equilibrium* [3], which intends to describe stable states of the system, i.e. states in which no user has an incentive to unilaterally change his route in the network.

In this article, we study a model of network games which has not received much attention until the moment, and which we call *Network Maximum Congestion Games* (NMC games). In this class of games, players try to minimize the congestion of the *most congested* link (instead of the sum of congestions of all links, as it happens in classical Network Congestion Games) of the network path they use. We believe that the study of this new model may be useful in a context where the bandwidth of the network is limited and users want to route their packets through paths with minimum bandwidth usage. Our study is focused on the analysis of the computational complexity of computing pure Nash Equilibria (PNE) in some types of Network Congestion Games.

Statement of results. We consider the general case where all players may have different weights, as well as different kinds of delay functions. We show that the cost functions, which depend essentially on these weights and delay functions, strongly affect the complexity of the PNE computation.

After giving some formal definitions in Sect. 2, in Sect. 3 we study the problem of computing a PNE for weighted single-commodity NMC games with non-decreasing delay functions. If we consider only network topologies where all the (s-t) paths are disjoint, then the problem can be solved in polynomial time. For general network topologies, the problem remains polynomial-time computable as long as all the edges have identical delay functions. For general NMC games, though, we prove that the same problem belongs to PLS. Since there are games of this kind which have no potential function [4], our proof relies on an alternative technique based on what we call *semi-potential* functions and on a particular neighborhood structure. As far as we know, this is the first time that the problem of computing a PNE for a game is expressed as a local search problem in which the cost function to be minimized is not a potential function.

In Sect. 4, we investigate the hardness of computing a PNE for a different kind of NMC games, which we call Pareto-efficient games and which have the property that no player can worsen the cost that other players assume by improving his own strategy. Following the aforementioned technique, we show that Pareto-efficient games always have a PNE and that its computation can also be expressed as a PLS optimization problem. Furthermore, we prove that is is a PLS-complete problem through a PLS-reduction from the Maxcut problem.

Related Work. With respect to the class of Network Congestion Games, most of the issues we study here have already been addressed. Any unweighted Congestion Game (including those defined over networks) possesses at least one Pure Nash Equilibria [5], whereas there exist weighted games with piecewise delay functions which possess no PNE [6]. However, weighted games with linear delay functions do always possess a PNE [7]. The computation of a Pure Nash Equilibria can be carried out in polynomial time in single-commodity unweighted

Network Congestion Games, while the same problem is, in general Network Congestion Games, PLS-complete (see [8] for the first proof or [9] for a simpler but later one).

With respect to Network Max-Congestion games, their study has been initialized in [10, 4, 11, 12]. All these papers analyze the existence of PNE as well as other aspects. Libman and Orda [10] prove the existence of a PNE for weighted single-commodity NMC games with increasing delay functions. They also analyze parallel links games with a broad class of residual capacity functions, a case where max-congestion games cannot be distinguished from traditional congestion games. Fotakis *et al.* [13] deal with weighted parallel links NMC games with a specific type of linear delay functions, and show that a PNE always exists and can be computed in polynomial time.

Caragiannis *et al.* [4] present some results with respect to the existence and complexity of PNE for NMC games with again a specific type of linear delay functions. It is shown that these games always have PNE, and that the computation of the best PNE is NP-complete for weighted single-commodity NMC games but can be done in polynomial time for unweighted NMC games where all users share either the source or the target nodes.

Finally, Banner and Orda [11] consider both weighted NMC games with splittable and unsplittable traffic, the former being games where the players can split their traffic among different paths, the latter being our scenario. They prove the existence of PNE for both cases when the delay functions are continuous and increasing, and show that best response dynamics for the unsplittable case always converge, though it may take an exponential number of steps to do so. They also investigate the efficiency of the PNE with respect to the social optimum and show that computing an optimal PNE is NP-hard.

2 Definitions and Preliminaries

A *Strategic game* can be formally defined as a tuple $\Gamma = (N, (P_i)_{i \in N}, (c_i)_{i \in N})$ where N is a set of $n \geq 2$ *players* and, for each player $i \in N$, P_i is a set of *actions* or *strategies* available to him and $c_i : \Pi = P_1 \times \cdots \times P_n \to \mathbb{R}$ his *cost function*. Each element π of this set Π is said to be a *strategy profile* (profile, for short) and reflects the choice of each player. Given a profile $\pi = (p_1, \ldots, p_n)$, we say that $p_i \in P_i$ is the strategy chosen by player $i \in N$ in π, while $c_i(\pi)$ is the cost assumed by the same player under that same profile π. We denote by (π_{-i}, p) the strategy profile resulting from replacing by p the strategy of player i in π.

If the cost functions of a strategic game are defined in such a way that all the improving movements that a player may do provoke no harm to the other players, that is, if for any profile π, player i and strategy $p \in P_i$ it holds that $c_i(\pi_{-i}, p) < c_i(\pi) \Rightarrow \forall j \in N \;\; c_j(\pi_{-i}, p) \leq c_j(\pi)$, then we say that the game is *Pareto-efficient*.

A strategy profile $\pi = (p_1, \ldots, p_n)$ is a *Pure Nash Equilibrium* (PNE) if for each player $i \in N$ and for each strategy $p \in P_i$ that player i may adopt, it holds that $c_i(\pi) \leq c_i(\pi_{-i}, p)$ (that is, all players are satisfied with their current strategies).

Both weighted *Congestion Games* and *Max-Congestion Games* are strategic games defined by a tuple $\Gamma = (N, E, (P_i)_{i \in N}, (w_i)_{i \in N}, (d_e)_{e \in E})$, where now the set of actions $P_i \subseteq 2^E$ available to each player $i \in N$ is composed by a number of subsets of the *set of resources* E. In addition, each player i has a *weight* $w_i \in \mathbb{N}^+$. We say that a (max-) congestion game is *unweighted* if all players have equal weight[2]. Finally, each resource $e \in E$ has a *delay function* $d_e : \mathbb{N} \to \mathbb{R}$, which we assume to be polynomial-time computable. When considering only games with non-decreasing delay functions, we call them *non-decreasing delay* games.

In this two classes of games, the cost function is given implicitly by means of these weights and delay functions. We say that, under a given strategy profile $\pi = (p_1, \ldots, p_n)$, the *load* $\lambda_e(\pi)$ of the resource $e \in E$ is the total weight of the players using it, $\lambda_e(\pi) = \sum_{\{i \in N |\ e \in p_i\}} w_i$, and the delay (or the congestion) caused by resource e is $d_e(\lambda_e(\pi))$. The cost assumed by the players is then computed as a function of the delays caused by the different resources he is using. For congestion games, the cost assumed by player i under profile π is $c_i(\pi) = \sum_{e \in p_i} d_e(\lambda_e(\pi))$, whereas for max-congestion games this cost is defined as $c_i(\pi) = \max_{e \in p_i} d_e(\lambda_e(\pi))$.

Network Congestion Games and *Network Max-Congestion Games* are congestion games where the sets of strategies of the players are concisely represented by means of a directed graph. These games are defined by a tuple $\Gamma = (N, G, ((s_i, t_i))_{i \in N}, (w_i)_{i \in N}, (d_e)_{e \in E(G)})$, where G is a directed graph, the resources are the edges of the graph and the set $P_i \subseteq 2^{E(G)}$ of strategies available to each player $i \in N$ is the set of all paths from his *origin node* s_i to his *destination node* t_i (that is, the set of all (s_i-t_i) paths, where by $(s - t)$ path we mean *path from* s *to* t). Note that these paths are not given explicitly. As before, we denote by $\lambda_e(\pi)$ the load of edge e in the strategy profile π, and each edge causes a delay $d_e(\lambda_e(\pi))$. The cost assumed by player i under profile π is $c_i(\pi) = \sum_{e \in p_i} d_e(\lambda_e(\pi))$ when we are dealing with network congestion games, whereas this cost is $c_i(\pi) = \max_{e \in p_i} d_e(\lambda_e(\pi))$ in the case of network max-congestion games.

When all players have the same pair of nodes $(s_i, t_i) = (s, t)$, we say that the network (max-) congestion game is *single-commodity*; otherwise, we say that it is *multi-commodity*. We will also consider the specific case of single-commodity games called *parallel links network* [14], where the graph is made up exclusively of a set of edges between the nodes s and t.

The function $\phi : \Pi \to \mathbb{R}$ is said to be an *ordinal potential* for a strategic game $\Gamma = (N, (P_i)_{i \in N}, (c_i)_{i \in N})$ if for every $\pi \in \Pi$, $i \in N$, $p \in P_i$, it holds that

[2] From now on, we will refer to weighted games unless explicitly stated.

$c_i(\pi_{-i}, p) < c_i(\pi) \Leftrightarrow \phi(\pi_{-i}, p) < \phi(\pi)$, and it is said to be an *exact potential* if for every $\pi \in \Pi$, $i \in N$, $p \in P_i$, it holds that $c_i((\pi_{-i}, p)) - c_i(\pi) = \phi((\pi_{-i}, p)) - \phi(\pi)$. It can be easily seen that every game that possesses an ordinal potential function (which is called an *ordinal potential game*) has a PNE [15]. It is also well-known that every unweighted congestion game is a *potential game* (i.e., a game possessing an exact potential function) [5], and that every (finite) potential game is isomorphic to an unweighted congestion game [15].

Finally, an optimization problem with a given neighborhood structure belongs to the complexity class PLS (for *Polynomial-time Local Search*) [16] if (a) all solutions are polynomially bounded and an initial solution can be computed in polynomial time, (b) the cost of any solution can be computed in polynomial time and (c) given a solution, we must be able in polynomial time to decide if it is locally optimal with respect to the given neighborhood structure and, if not, to compute a neighboring solution with better cost.

3 Non-decreasing Delay NMC Games

We first study NMC games with non-decreasing delay functions, and identify some cases for which the computation of a PNE can be done in polynomial time.

3.1 A polynomial time upper bound for some particular cases

We present an algorithm that computes PNE profiles for some kinds of weighted single-commodity non-decreasing delay. The algorithm BDP (for *Best Disjoint Path*) proceeds in two phases. First, it computes a maximal set M of $(s-t)$ disjoint paths, and then it assigns players to paths in M step by step: at each step, the player i with maximum weight (from the set of players still unassigned) is assigned to the path $p \in M$ that yields him the minimum cost, that is, the path that minimizes the value $\max_{e \in p} d_e(\lambda_e(\pi) + w_i)$, where π is the (partial) strategy profile we are constructing. Let's now analyze the properties of the profiles computed by the algorithm.

Lemma 1. *After each iteration of the BDP algorithm, no player already assigned to some path in the (partial) profile π has an incentive to change his strategy for another path from M.*

Proof. The claim obviously holds after the first iteration. By induction, suppose that it also holds for the (partial) profile π^k obtained after the k-th iteration, let i be the player assigned to path p on the iteration $k+1$ and let us consider the profile $\pi^{k+1} = (\pi^k_{-i}, p)$ computed after this $(k+1)$-th iteration. By construction, player i is satisfied with his strategy in π^{k+1}.

Let's now suppose that for some player j already assigned to p in π^k there is an incentive to change his strategy to another path $p' \in M$. This means that $\max_{e \in p'} d_e(\lambda_e(\pi^k) + w_j) < \max_{e \in p} d_e(\lambda_e(\pi^k) + w_i)$. Since $w_i \leq w_j$ and the delay

functions are non-decreasing, $\max_{e \in p'} d_e(\lambda_e(\pi^k)+w_i) \leq \max_{e \in p'} d_e(\lambda_e(\pi^k)+w_j)$, but this implies that $\max_{e \in p'} d_e(\lambda_e(\pi^k) + w_i) < \max_{e \in p} d_e(\lambda_e(\pi^k) + w_i)$, which contradicts the fact that p has been chosen to minimize this last value. Thus, no player assigned to p has incentive to change his strategy.

Now, let us suppose that for some player j assigned to a path $p' \neq p \in M$ there is an incentive to change his strategy to another path in M. The cost induced by paths different than p has clearly remained unchanged from π^k to π^{k+1} (since we are considering only disjoint paths), while the cost induced by path p cannot be lower in π^{k+1} than in π^k, for the delay functions are non-decreasing. Thus, this contradicts our inductive hypothesis that all players in π^k were satisfied with their path. Hence, all players in π^{k+1} are satisfied. □

For graphs where all the $(s-t)$ paths are disjoint (which include parallel links), the set M computed by the BDP algorithm coincides with the set of all the strategies available to players. Hence, the computed profile is a PNE.

Corollary 1. *For single-commodity NMC games where the graph consists only of a number of $(s-t)$ disjoint paths, the profile computed by the BDP algorithm is a PNE.*

Also, if we consider general graph topologies but we restrict the delay functions of the edges to be identical, we have that, although there may be strategies other than the disjoint paths, none of them will yield a better cost.

Corollary 2. *For single-commodity NMC games where all the delay functions are identical, the profile computed by the BDP algorithm is a PNE.*

Proof. Notice that the profiles computed by the algorithm have the property that, for any path $p \in M$, all edges $e \in p$ have the same load λ_e. Thus, we now have that the delay induced by all edges of p is equal. Since M is maximal, any $(s-t)$ path p' shares at least one edge e with some path $p \in M$, and this edge e induces a delay equal to the delay induced by the whole path p. Hence, a player choosing p' would have to assume a cost at least as large as if he chose p, which (by the previous lemma) is no better than the cost of the path he has been assigned to in π. □

Given that a maximal set of disjoint paths can be obtained in polynomial time and that our algorithm runs also in polynomial time, we can state the following theorem.

Theorem 1. *A PNE can be computed in polynomial time for both (a) Weighted single-commodity NMC games with identical non-decreasing delay functions and (b) Weighted single-commodity NMC games with non-decreasing delay functions where all the $(s-t)$ paths of the graph are disjoint.*

3.2 A PLS upper bound for the general case

Now we move to the study of multi-commodity NMC games. The next lemma states that there exists a total ordering among strategy profiles that has an important property.

Lemma 2. *Let $A(\pi)$ be a tuple $A(\pi) = (c_{i_1}(\pi), c_{i_2}(\pi), \dots c_{i_n}(\pi))$ such that $c_{i_1}(\pi) \geq c_{i_2}(\pi) \geq \cdots \geq c_{i_n}(\pi)$. Let $<_l$: $\mathbb{N}^n \to \mathbb{N}^n$ be the usual (total) lexicographical order defined over all possible pairs of tuples $A(\pi)$, $A(\pi')$. Let Γ be a non-decreasing delay NMC game and π, π' two strategy profiles such that $\pi = (\pi_{-i}, p)$ and $\pi' = (\pi_{-i}, p')$ for some strategies $p \neq p'$ of player i. Then,*

$$c_i(\pi') < c_i(\pi) \Rightarrow A(\pi') <_l A(\pi)$$

Proof. The proof of the statement follows from the proof of Theorem 3 in [11] (which can be directly generalized to the case of arbitrary non-decreasing delay functions). □

This ordering suggests a natural way to prove the existence of PNE profiles.

Theorem 2 ([11]). *Any non-decreasing delay NMC game has a PNE.*

Proof. Since Π is finite, there must exist a profile $\pi^* \in \Pi$ such that $\forall \pi \in \Pi$ $A(\pi^*) <_l A(\pi)$. Suppose that, in π^*, a player i has incentive to change his strategy to p'; then, by Lemma 2 we have that $A(\pi^*_{-i}, p') <_l A(\pi^*)$, a contradiction. Thus, π^* is a PNE. □

Notice that, unlike what happens with unweighted congestion games, there exist unweighted non-decreasing delay NMC games which have no potential function [4], so the existence of PNE profiles cannot be proved directly by means of this potential function. Furthermore, the absence of a potential function also prevents us to directly prove (as it can be proved for congestion games, see [8]) that the computation of a PNE for this kind of games is in PLS.

However, consider the function γ that maps each profile π to the value of $A(\pi)$ seen as a n-digit number in base $D+1$ (that is, $\gamma(\pi) = \sum_{0 \leq j \leq n} A(\pi)_j \cdot (D+1)^j$), where $D = \max_{e \in E} d_e(W)$ and $W = \sum_{i \in N} w_i$. It can be seen that γ is computable in polynomial time and that $A(\pi') <_l A(\pi) \Leftrightarrow \gamma(\pi') < \gamma(\pi)$, for any $\pi, \pi' \in \Pi$. Recalling Lemma 2 and the definition of an ordinal potential function, it thus seems appropriate to call this function a *semi-potential* function, since it meets only one of the two implications of the definition. The use of this function (together with a special definition of the neighborhood structure of the search problem, as we will see) allows us to prove the aforementioned membership in PLS.

Theorem 3. *Computing a PNE for weighted multi-commodity non-decreasing delay NMC belongs to* PLS.

Proof. Consider the search problem of finding a local optimum of γ, where the set of feasible solutions contains all valid strategy profiles of our game and the neighborhood $N(\pi, \Gamma)$ of a solution π is the set of all profiles π' where exactly one player has changed his strategy for a better one:

$$N(\pi, \Gamma) = \{\pi' \in \Pi \mid \exists i \in N \;\; \exists p \in P_i \;\; \pi' = (\pi_{-i}, p) \wedge c_i(\pi') < c_i(\pi)\}$$

Notice that this last condition ($c_i(\pi') < c_i(\pi)$) implies that (i) the neighborhood of π is empty if and only if π is a PNE and (ii) For any neighbor $\pi' \in N(\pi, \Gamma)$ of a given profile π, it holds that $\gamma(\pi') < \gamma(\pi)$ (by definition of γ and Lemma 2). Thus, finding a PNE is equivalent to finding a local minimum of the function γ with respect to the neighborhood defined above.

This search problem belongs to PLS, since (a) an initial solution can be produced in polynomial time by assigning to every player an arbitrary strategy, (b) The cost $\gamma(\pi)$ of any profile π can be computed in polynomial time. (c) deciding whether $N(\pi, \Gamma) = \emptyset$ (i.e. π is a local optimum) or, if this is not the case, computing a strategy profile $\pi' \in N(\pi, \Gamma)$ s.t. $\gamma(\pi') < \gamma(\pi)$ can be done in polynomial time using a modification of the Dijkstra algorithm where the shortest path computation is done considering that the length of an edge e is $d_e(\lambda_e(\pi))$ and the length of a path p is $\max_{e \in p} d_e(\lambda_e(\pi))$. □

Notice that the previous proof can be immediately generalized to the case of general max-congestion games. We only have to see that, since the representation of the game explicitly contains the set of actions for each player, deciding if a given profile is a local optimum with respect to the neighborhood can be done in polynomial time by computing the cost of all neighbors of the profile.

As we stressed before, the neighborhood used in the above proof is not the neighborhood that one may initially think of when considering the nature of games (see, for instance, [8, 17]), where the neighbors of a given profile π are *all* profiles $\pi' = (\pi_{-i}, p)$ (for some $i \in N$ and $p \in P_i$). Given that max-congestion games are not potential games and that we only have at our disposal a *semi-potential* function, we have to restrict the neighbors to those profiles where the deviating player improves his cost in order to get a search problem whose local optima coincide with the PNE profiles of the game.

Finally, let us say that the requirement of the delay functions being non-decreasing is essential to the proof of Lemma 2, so the above technique can not be trivially extended to games with arbitrary delay functions. In the next section, though, we will see a particular class of games with delay functions that are not restricted to be non-decreasing which does actually possess a semi-potential function. This will allow us to prove that the problem of computing a PNE for this class of games belongs to PLS.

4 Hard NMC Games

Let σ denote the function $\sigma(\pi) = \sum_{i \in N} c_i(\pi)$. For Pareto-efficient games, we have that whenever a player changes his strategy to decrease his cost, the cost

assumed by the rest of players does not increase. Thus, the sum of all costs strictly decreases.

Proposition 1. *Let Γ be a Pareto-efficient game and π, π' two strategy profiles such that $\pi = (\pi_{-i}, p)$ and $\pi' = (\pi_{-i}, p')$ for some strategies $p \neq p'$ of player i. Then,*

$$c_i(\pi') < c_i(\pi) \Rightarrow \sigma(\pi') < \sigma(\pi)$$

Note that the implication holds only in one direction, so all we can say about σ is that it is a semi-potential function, but not a potential one. The existence of this function makes it possible to use the same proof ideas that we used in the previous section to prove both the existence of PNE profiles and the membership in PLS of the problem of computing them in this kind of games.

Theorem 4. *Any Pareto-efficient game has a PNE.*

Proof. The proof is analogous to the proof of theorem 2; the profile π^* that minimizes $\sigma(\pi^*)$ is again a PNE. □

The problem of computing a PNE for Pareto-efficient NMC games can be seen to be equivalent to the problem of finding a local optimum of the function σ.

Theorem 5. *Computing a PNE for weighted multi-commodity Pareto-efficient NMC is in* PLS.

Proof. This problem is equivalent to the problem of finding a local minimum of the function σ with respect to the neighborhood defined in the proof of theorem 3.

Notice that this result can be directly extended to Pareto-efficient strategic games the sets of strategies of each player are explicitly given.

Corollary 3. *Computing a PNE for Pareto-efficient strategic games is in* PLS.

Although the Pareto-efficiency may seem too strong a restriction, we next prove that the computation of a PNE for Pareto-efficient weighted parallel links NMC games is a PLS-hard problem. The proof is based on a PLS-reduction from the Maxcut problem with the flip neighborhood, for which finding a local optimum is known to be PLS-hard [18]. In this problem, we are given an undirected graph with weights on the edges and we have to find a partition of the nodes into two disjoint sets A and B such that the cut of the partition (i.e. the sum of weights of edges between nodes assigned to different sets) cannot be increased by changing one single node from A to B or vice versa.

Theorem 6. *Computing a PNE for weighted Pareto-efficient parallel links NMC games is* PLS*-hard.*

Proof. Let's define W_T as the sum of weights of all edges and W_A (and analogously, W_B) as the sum of weights of edges with both endpoints in A (B). Then, the value of the cut is $W_T - (W_A + W_B)$, and maximizing it is equivalent to minimizing $W_A + W_B$, since W_T is fixed.

Now, given a Maxcut instance, we can define a weighted parallel links max-congestion game in the following way. There's one player for every node of the original graph, and the weight of each player is used as a means of identifying that player. Thus, the weight w_i of player i is a binary number of length n where the i-th digit is 1 and the rest of the digits are 0. The graph G of the game is a simple graph with two nodes, s and t, and two parallel links e_1 and e_2 from s to t. Finally, the delay function of both edges is defined as

$$d(l) = \sum_{\substack{1 \leq i < j \leq n \\ \text{s.t. } l_i = l_j = 1}} w_{i,j} + \sum_{\substack{1 \leq i < j \leq n \\ \text{s.t. } l_i = l_j = 0}} w_{i,j}$$

where l_i is the i-th digit of l and $w_{i,j}$ is the weight of edge (i, j). Intuitively, player i choosing edge e_1 can be thought of as node i being assigned to set A (equivalently for edge e_2 and set B). The delay of both edges is then $W_A + W_B$. Thus, in any PNE the value $W_A + W_B$ cannot be decreased by one player moving from one edge to the other, so the partition of nodes of the Maxcut problem induces a maximum cut with respect to the flip neighborhood. Besides, the game is clearly Pareto-efficient, so the proof is complete. □

The previous reduction implies the PLS-hardness of the computation of a PNE for (a) weighted parallel links NMC games with arbitrary delay functions (which strongly contrasts with the fact that the same problem is in P if we only allow non-decreasing delay functions, see Theorem 1), (b) Pareto-efficient weighted network congestion games (since there is no distinction between parallel links NMC games and parallel links network congestion games) and (c) general Pareto-efficient strategic games.

5 Discussion and Open Problems

We have presented a number of results related to the complexity of computing PNE strategy profiles, mainly for NMC games but also for general Pareto-efficient strategic games. On one hand, we have identified some types of single-commodity games for which the computation of a PNE profile can be done in polynomial time, and we have provided an algorithm to carry out this computation. These are weighted single-commodity non-decreasing delay NMC games where either all the delay functions are identical or all the $(s - t)$ paths of the graph are disjoint. Note that this result incidentally extends the class of parallel links *congestion* games for which the computation of a PNE was known to be in P [7], which now also includes games with arbitrary non-decreasing delay functions.

On the other hand, we have seen that the computation of PNE for two broad classes of NMC games (namely, non-decreasing delay NMC games and Pareto-efficient NMC games) is a PLS problem; we believe that the technique we have used to prove so can be extended to other classes of games for which there exists a function such as the one presented in proposition 1, provided that some other natural assumptions hold (for instance, we must be able to decide in polynomial time if a given profile is a PNE or, if not, to compute a profile where one player improves his cost). Thus, this semi-potential function technique may be useful to prove PNE existence and PLS membership when polynomial-time computable potential functions do not exist (or are not known to exist).

Finally, we have proved that there are some kinds of NMC games for which computing a PNE profile is PLS-complete, though it remains open the question about the hardness of this problem for general non-decreasing delay NMC games.

References

1. Osborne, M.: An introduction to game theory. Oxford University Press (2004)
2. Papadimitriou, C.: Algorithms, games, and the internet. In: STOC'01, ACM Press (2001) 749–753
3. Nash, J.: Equilibrium Points in n-Person Games. Proceedings of the National Academy of Sciences **36**(1) (1950) 48–49
4. Caragiannis, I., Galdi, C., Kaklamanis, C.: Network load games. LNCS **3827** (2005) 809–818
5. Rosenthal, R.W.: A class of games possessing pure-strategy Nash equilibria. International Journal of Game Theory **2**(1) (1973) 65–67
6. Fotakis, D., Kontogiannis, S., Spirakis, P.: Selfish unsplittable flows, Springer (2004) 593–605
7. Fotakis, D., Kontogiannis, S., Spirakis, P.: Symmetry in network congestion games: Pure equilibria and anarchy cost. Volume 3879., Springer (2006) 161–175
8. Fabrikant, A., Papadimitriou, C., Talwar, K.: The complexity of pure Nash equilibria. Proceedings of the thirty-sixth annual ACM symposium on Theory of computing (2004) 604–612
9. Ackermann, H., Roglin, H., Vocking, B.: On the Impact of Combinatorial Structure on Congestion Games. Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)-Volume 00 (2006) 613–622
10. Libman, L., Orda, A.: Atomic Resource Sharing in Noncooperative Networks. Telecommunication Systems **17**(4) (2001) 385–409
11. Banner, R., Orda, A.: Bottleneck Routing Games in Communication Networks. Proceedings of the 25th INFOCOM Conference (2006)
12. Busch, C., Magdon-Ismail, M.: Atomic Routing Games on Maximum Congestion. Lecture notes in computer science **4041** (2006) 79–91
13. Fotakis, D., Kontogiannis, S., Koutsoupias, E., Mavronicolas, M., Spirakis, P.: The structure and complexity of nash equilibria for a selfish routing game. In: ICALP '02: Proceedings of the 29th International Colloquium on Automata, Languages and Programming. Volume 2380., Springer (2002)
14. Koutsoupias, E., Papadimitriou, C.: Worst-case equilibria. In: STACS'99. Volume 1563., Springer (1999) 404–413

15. Monderer, D., Shapley, L.: Potential games. Games and Economic Behavior **14**(1) (1996) 124–143
16. Johnson, D., Papadimtriou, C., Yannakakis, M.: How easy is local search? J. Comput. Syst. Sci. **37**(1) (1988) 79–100
17. Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V.: Algorithmic Game Theory. Cambridge University Press New York, NY, USA (2007)
18. Schäffer, A., Yannakakis, M.: Simple local search problems that are hard to solve. SIAM Journal on Computing **20**(1) (1991) 56–87