Maximum Congestion Games on Networks: How Can We Compute Their Equilibria? *

Carme Àlvarez and Guillem Francès

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

Abstract. We study Network Maximum Congestion Games, a class of network games where players choose a path between two given nodes in order to minimize the congestion of the bottleneck (the most congested link) of their path. For single-commodity games, we provide an algorithm which computes a Pure Nash Equilibrium in polynomial time. If all players have the same weight, the obtained equilibrium has optimum social cost. If players are allowed to have different weights, the obtained equilibrium has social cost at most $\frac{4}{3}$ times worst than the optimum. For multi-commodity games with a fixed number of commodities and a particular graph topology, we also provide an algorithm which computes a Pure Nash Equilibria in polynomial time. We also study some issues related to the quality of the equilibria in this kind of games.

Keywords. *Maximum Congestion, Network Congestion Games, Nash equilibria, Complexity, Edge-Disjoint Paths.*

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 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 congestion of the network paths through which they route their packets.

One of the fundamental concepts of Game Theory is that of the *Nash Equilibrium* [3], which intends to describe stable configurations of the system, i.e. configurations in which no user has an incentive to unilaterally change his route

* Work partially supported by FET pro-actives Integrated Project 15964 (AEOLUS), by Spanish CICYT under grant TIC2004–C02-02 (ASCE) and Spanish CICYT under grant TIN2004-07925-C03-01 (GRAMMARS)

[1] As stated by C. Papadimitriou on his 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".

in the network. Other useful concepts, such as that of the *Price of anarchy* of a game, relate to the quality of equilibria, i.e. to their cost measured from a global point of view.

In this article, we study a model of Network Games which has received little attention until the moment, and which we call *Network Maximum Congestion Games* (NMC games). In this class of games, the players try to minimize the congestion of the most congested link of the network path they use, instead of trying to minimize the sum of congestions of all the links (as it happens in classical Network Congestion Games). 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 in some types of Network Congestion Games, as well as on some other issues related to the quality of these Nash Equilibria.

With respect to the class of classical Network Congestion Games, most of the issues we study here have already been addressed. Any unweighted Congestion Game (including those defined over networks) possess at least one Pure Nash Equilibria [4], whereas there exist games with piecewise delay functions which possess no PNE, [5]. However, [6] proves that weighted games with linear delay functions do always possess a PNE. The computation of a Pure Nash Equilibria can be done in polynomial time in single-commodity unweighted Network Congestion Games, whereas the same problem is, in general Network Congestion Games, PLS-complete [7]. Finally, the Price of anarchy of single-commodity Network Congestion Games is at most $PA = \Omega(\sqrt{n})$, and this bound is tight [8]. If we consider only single-commodity games, this upper tight bound can be set to $PA = \frac{5}{2}$.

For single-commodity unweighted NMC games, we prove that PNE always exist and that their computation can be done in polynomial time. Furthermore, we provide an algorithm which actually computes a PNE with optimum social cost in polynomial time. If we consider single-commodity weighted NMC games, our algorithm turns to be a $\frac{4}{3}$-approximation for the problem of computing an optimum PNE. With respect to the quality of the configurations, we prove that the Price of anarchy ([9]) in single-commodity games is at most equal to the maximum number of edge-disjoint paths between the two distinguished nodes s and t of the game.

For the case of unweighted multi-commodity NMC games, we limit our study to a particular class of games, which we call *l-common-edges* games, and provide an algorithm which computes a PNE in polynomial time, as long as we consider that the number of commodities of the input game is fixed (i.e. there is a constant number of source-destination pairs in the network).

During the development of our work, three papers [10–12] have been published which independently study the same class of Network Maximum Congestion Games (albeit with different denominations). Although they all prove the existence of PNE for the class of NMC games, only [10] gives a constructive proof for the single-commodity case. However, that proof is completely independent of

ours, and we believe that the conceptual clarity of our algorithm is interesting enough to be presented here. On the other hand, none of these papers tackles the multi-commodity case from a constructive point of view, as we do.

The rest article is organized as follows: in section 2 we give some formal definitions and specify the notation we will be using in the rest of the article. In section 3 we study single-commodity games, whereas in section 4 we tackle the study of *l-common-edges* games. Finally, in section 5 we conclude the article with some remarks. Some of the proofs have been omitted due to the lack of space.

2 Definition of the model

A *Game* is formally defined as a tuple $\Gamma = (N, (P_i)_{i \in N}, (u_i)_{i \in N})$ with a set N of n players and, for each player $i \in N$, a set of *actions* or *strategies* P_i available to him and a *payoff function* $u_i : P_1 \times \cdots \times P_n \to \mathbb{R}$. Each element π of this set $P_1 \times \cdots \times P_n$ is said to be a *configuration* or *strategy profile* and reflects the choice of each player. Given a configuration $\pi = (p_1, \ldots, p_n)$, we say that $p_i \in P_i$ is the strategy chosen by player $i \in N$ in π, while $u_i(\pi)$ is the benefit obtained by the same player also under π[2].

A NMC game is a tuple $\Gamma = (N, G, ((s_i, t_i))_{i \in N}, (w_i)_{i \in N})$, where N is the set of players, G is a directed graph and $(s_i, t_i) \in V(G) \times V(G)$ and $w_i \in \mathbb{N}_+$ are, respectively, the *commodity* and the *weight* of player i. The set P_i of actions available to each player $i \in N$ (which is implicitly defined by G and the pair (s_i, t_i)) is the set of all the paths between s_i and t_i (we will often denote that set by $P^{(s_i,t_i)}$, and we will use the expression $(s-t)$ *path* to mean *a path between nodes s and t*). Given a configuration $\pi = (p_1, \ldots, p_n)$, we will denote by $\Lambda_e(\pi)$ the *set of users* of each edge $e \in E(G)$ under π (i.e., $\Lambda_e(\pi) = \{i \in N \mid e \in p_i\}$), and we will denote by $\Lambda_p(\pi)$ the *set of users* of path p (i.e. $\Lambda_p(\pi) = \{i \in N \mid p_i = p\}$). The *congestion* l_e of the edge $e \in E(G)$ under π is the sum of the weights of the users of e, $l_e(\pi) = \sum_{i \in \Lambda_e(\pi)} w_i$. The *congestion* L_p of any path p of the graph is then defined as the maximum of the congestions of the edges it contains, $L_p(\pi) = max\{l_e(\pi) \mid e \in p\}$. Finally, the cost assumed by player $i \in N$ under the configuration $\pi = (p_1, \ldots, p_n)$ is exactly the congestion of the path p_i he uses in π, $c_i(\pi) = L_{p_i}(\pi)$. We will use the standard notation (π_{-i}, p) to refer to the configuration resulting from replacing by p the strategy of player i in π.

A NMC game is said to be *single-commodity* if all the players select paths between the same pair (s, t) of nodes[3]; otherwise, the game is said to be *multi-*

[2] It is not uncommon to describe a game in terms of *costs* instead of benefits. In that case, for each player $i \in N$ there is a *cost function* $c_i : P_1 \times \cdots \times P_n \to \mathbb{R}$, and $c_i(\pi)$ is the cost assumed by player i under the configuration π.

[3] For the sake of simplicity, we will often describe the games of this class as tuples $\Gamma = (N, G, (s, t), (w_i)_{i \in N})$

commodity. Also, a NMC game is said to be *unweighted* if the weight of all players equals one[4]; otherwise, the game is said to be *weighted*.

A configuration π is a *Pure Nash Equilibrium* (PNE) if no player has an incentive (with respect to the cost he assumes) to change his strategy in π. Formally, we say that $\pi = (p_1, \ldots, p_n)$ is a PNE if for each player $i \in N$ and for each strategy $p \in P_i$ that player i may adopt it holds that $c_i((\pi_{-i}, p)) \geq c_i(\pi)$. Often, it is interesting to measure the quality of a PNE configuration from a global or social point of view. To this end, the social cost of a configuration π is defined as $SC(\pi) = max\{c_i(\pi) \mid i \in N\}$. In the case of NMC games, we can equivalently state the preceding definition in the following manner: $SC(\pi) = max\{l_e(\pi) \mid e \in E\}$. We will denote by SC^* the optimal social cost, $SC^* = min\{SC(\pi) \mid \pi \text{ is a config.}\}$. Notice that a game can have more than one configuration with optimal social cost. Related to the quality of the PNE configurations, the concept of the *Price of anarchy* (or *coordination ratio*) of a game [9] arises as a means of quantifying the degradation of the social cost due to the lack of coordination among players. Thus, it is formally defined as $PA = \max_{\pi \text{ is a PNE}} \frac{SC(\pi)}{SC^*}$.

3 Computation of Equilibria in Single-Commodity Games

First of all, we study some properties of PNE configurations in NMC games. In the unweighted case, it seems natural to think that in a PNE configuration the congestion of all the paths is almost the same. On the other hand, a configuration where all the paths have exactly the same congestion is a PNE.

Proposition 1. *Let $\pi = (p_1, \ldots, p_n)$ be a configuration for the unweighted NMC $\Gamma = (N, G, (s,t))$ and let $k = SC(\pi)$. Then,*

1. *If π is a PNE, then $\forall p \in P^{(s,t)}\ L_p(\pi) \in \{k, k-1\}$.*
2. *If $\forall p \in P^{(s,t)}\ L_p(\pi) = k$, then π is a PNE.*

Therefore, in order to compute a PNE we have to keep balanced the congestion of all paths in $P^{(s,t)}$. A simple distribution of players among paths would not keep that balance, since the edges shared by a greater number of paths would increase their congestion faster, thus provoking an uneven increase of the congestion of the different paths. However, this can be avoided if users select only edge-disjoint paths. In that case, the congestion of any edge can be proven to be equal to the congestion of the path where it belongs to:

Proposition 2. *Let $\pi = (p_1, \ldots, p_n)$ be a configuration for the weighted NMC $\Gamma = (N, G, (s,t), (w_i)_{i \in N})$. If $C = \{p_i \mid i \in N\}$ is a set of $(s-t)$ edge-disjoint paths, then $\forall p \in C\ \forall e \in p\ \Lambda_e(\pi) = \Lambda_p(\pi)$, and hence $l_e(\pi) = L_p(\pi) = \sum_{i \in \Lambda_p(\pi)} w_i$*

[4] In this case, we will describe games of this class as tuples $\Gamma = (N, G, ((s_i, t_i))_{i \in N})$

Notice that the last property holds for weighted games. It also implies that if we distribute players among a set of edge-disjoint paths, we may end up with a configuration where the congestion of those paths is fairly balanced. Intuitively, if the set of paths is maximal, then all $(s-t)$ paths will have a balanced congestion, and the configuration will probably be a PNE. If we want our distribution algorithm to be valid also for weighted games, we will have to carefully consider the order in which players are allocated. All those ideas are formalized in Algorithm 1, which allocates iteratively the player with the highest weight to the path with the lowest congestion.

Algorithm 1: Computation of a PNE (`w_round_robin` algorithm)

input : A weighted single-comm game $\Gamma = (N, G, (s,t), (w_i)_{i \in N})$
input : A set $C \subseteq P^{(s,t)}$ of paths in G
output : A configuration π for the given game

```
begin
    N' := N;
    while N' ≠ ∅ do
        Let i be a player from N' with maximum weight w_i;
        Let c be a path from C with minimum congestion L_p(π);
        Allocate player i to path c;
        N' := N' \ {i};
end
```

3.1 Correctness and Complexity Issues

In order to prove that the configuration computed by Algorithm 1 is a PNE, we need to introduce some notation. For all $j \in \{1, 2, \ldots, n+1\}$, let π^j denote the configuration computed by the algorithm just before the computation of the j-th iteration, and let $N^j \subseteq N$ denote the set of players which have been allocated to a path before that iteration. Finally, let i^j and p^j respectively denote the player and the path selected by the algorithm during the execution of the j-th iteration. Notice that, unless $j = n+1$, the configurations π^j are *incomplete* in the sense that they do not contain a strategy for all players, but only for players in N^j. For the remaining players, we can consider that their strategy is the empty path $\emptyset$. Hence, note that π^1, contains n empty paths.

The next lemma shows an important invariant of the algorithm.

Lemma 3. *Given a game $\Gamma = (N, G, (s,t), (w_i)_{i \in N})$ and a set $C \subseteq P^{(s,t)}$ of edge-disjoint $(s-t)$ paths, the algorithm `w_round_robin` on input (Γ, C) satisfies that, for any $j \in \{1, \ldots, n+1\}$, no player in N^j has an incentive to change his strategy in π^j for any other path in C. Formally,*

$$\forall i \in N^j \;\; \forall p \in C \;\; c_i((\pi^j_{-i}, p)) \geq c_i(\pi^j)$$

Proof. For $j = 1$, we have that $N^1 = \emptyset$, and the property trivially holds. Let us assume that $\forall i \in N^j \ \forall p \in C \ c_i((\pi^j_{-i}, p)) \geq c_i(\pi^j)$ and consider the iteration $j+1$. We can distinguish two cases:

1. Let's first consider the set of players $N' \subseteq N^{j+1}$ allocated to p^j in π^{j+1}. We will show that none of these players has an incentive to change his strategy. Let $i \in N'$ be one of them, and let $p \in (C \setminus \{p^j\})$ be an alternative strategy (of course, i has no incentive to change to path p^j, since that is his current strategy). We will show that $L_p((\pi^{j+1}_{-i}, p)) \geq L_{p^j}(\pi^{j+1})$.
   First, since i^j has been allocated to path $p^j \neq p$, the set of players of p remains unchanged, $\Lambda_p(\pi^{j+1}) = \Lambda_p(\pi^j)$. If player i changes p^j for p, then $\Lambda_p((\pi^{j+1}_{-i}, p)) = \Lambda_p(\pi^{j+1}) \cup \{i\} = \Lambda_p(\pi^j) \cup \{i\}$. By proposition 2 (which can be applied to π^j and π^{j+1} since they only contain paths from C), we then have that

$$L_p((\pi^{j+1}_{-i}, p)) = \sum_{k \in \Lambda_p((\pi^{j+1}_{-i}, p))} w_k = w_i + \sum_{k \in \Lambda_p(\pi^j)} w_k = w_i + L_p(\pi^j)$$

   Since the algorithm guarantees that $w_i \geq w_{i^j}$ and that $L_p(\pi^j) \geq L_{p^j}(\pi^j)$, we have that $w_i + L_p(\pi^j) \geq w_{i^j} + L_{p^j}(\pi^j)$. Thus, showing that this last expression is exactly $L_{p^j}(\pi^{j+1})$ will suffice to prove that $L_p((\pi^{j+1}_{-i}, p)) \geq L_{p^j}(\pi^{j+1})$. Since the set of users of p^j in π^{j+1} is $\Lambda_p(\pi^{j+1}) = \Lambda_p(\pi^j) \cup \{i^j\}$ (and, by definition, $i^j \notin \Lambda_p(\pi^j)$), by proposition 2 we have that

$$L_{p^j}(\pi^{j+1}) = \sum_{k \in \Lambda_p(\pi^{j+1})} w_k = w_{i^j} + \sum_{k \in \Lambda_e(\pi^j)} w_k = w_{i^j} + L_{p^j}(\pi^j)$$

2. Let's now consider the rest of players in N^{j+1}, i.e. the ones who are not allocated to p^j in π^{j+1}. Again, we will show that none of them has an incentive to change his strategy. Notice that, given that $i^j \in N'$, $N^{j+1} \setminus N' = (N^j \cup \{i^j\}) \setminus N' = N^j \setminus N'$. Let $i \in N^j \setminus N'$ and let $p \in C$. We want to show that $L_p((\pi^{j+1}_{-i}, p)) \geq L_{p_i}(\pi^{j+1})$, p_i being the strategy of player i in π^{j+1}. But using proposition 2 and our inductive hypothesis, it can be proven that $L_p((\pi^{j+1}_{-i}, p)) \geq L_p((\pi^j_{-i}, p)) \geq L_{p_i}(\pi^j) = L_{p_i}(\pi^{j+1})$, which is what we wanted to prove.

Since, by definition, $N^{j+1} = N' \cup (N^{j+1} \setminus N')$, we have proven that

$$\forall i \in N^{j+1} \ \forall p \in C \ L_p((\pi^{j+1}_{-i}, p)) \geq L_{p_i}(\pi^{j+1})$$

or, equivalently,

$$\forall i \in N^{j+1} \ \forall p \in C \ c_i((\pi^{j+1}_{-i}, p)) \geq c_i(\pi^{j+1})$$

□

Once we have shown the invariant property satisfied after each iteration of the algorithm, the next property of the strategy profile computed by the algorithm follows immediately.

Corollary 4. *Let $\pi = (p_1, \ldots, p_n)$ be the configuration obtained by the algorithm* `w_round_robin` *applied to a game $\Gamma = (N, G, (s,t), (w_i)_{i \in N})$ and a set of edge-disjoint paths $C \subseteq P^{(s,t)}$. Then, no player has incentive to change his strategy in π for any other path in C. Formally, $\forall i \in N \ \ \forall p \in C \ \ c_i((\pi_{-i}, p)) \geq c_i(\pi)$.*

Notice that if C is equal to the set of all $(s-t)$ paths in G then the configuration computed by the algorithm is a PNE, but if there exist other $(s-t)$ paths the configuration might not be a PNE. However, if C is a *maximal* set of edge-disjoint $(s-t)$ paths in G, then the configuration computed by `w_round_robin` on input (Γ, C) can be proven to be a PNE.

Theorem 5. *The configuration $\pi = (p_1, \ldots, p_n)$ computed by the algorithm* `w_round_robin` *applied to a game $\Gamma = (N, G, (s,t), (w_i)_{i \in N})$ and to a maximal set of edge-disjoint paths $C \subseteq P^{(s,t)}$ is a PNE.*

Proof. Let $i \in N$ and let $p \in P^{(s,t)}$. We will first show that there exists a path $c^* \in C$ such that $L_p((\pi_{-i}, p)) \geq L_{c^*}((\pi_{-i}, c))$.

Since C is a maximal set, p shares at least one edge with at least one path in C. Let $c^* \in C$ be one of these paths and let $e^* \in (p \cap c^*)$ be one of the shared edges. Let $\pi_p = (\pi_{-i}, p)$ and $\pi_{c^*} = (\pi_{-i}, c^*)$ be the corresponding configurations obtained when player i changes his strategy in π for p and c^*, respectively. Since $\Lambda_{e^*}(\pi_p) = \Lambda_{e^*}(\pi_{c^*}) = \Lambda_{e^*}(\pi) \cup \{i\}$ and players in π_{c^*} select only paths from a set of disjoint paths, by proposition 2 all the edges in c^*, even e^*, have exactly the same congestion $L_{c^*}(\pi_{c^*})$. Hence,

$$L_p(\pi_p) \geq \ \ l_{e^*}(\pi_p) = \sum_{i \in \Lambda_{e^*}(\pi_p)} w_i = \sum_{i \in \Lambda_{e^*}(\pi_{c^*})} w_i = l_{e^*}(\pi_{c^*}) = L_{c^*}(\pi_{c^*})$$

By corollary 4 we have that $L_{c^*}(\pi_{c^*}) \geq L_{p_i}(\pi)$. Thus, $c_i(\pi_p) = L_p(\pi_p) \geq L_{c^*}(\pi_{c^*}) \geq L_{p_i}(\pi) = c_i(\pi)$, which proves that π is a PNE. □

A simple analysis of the computational complexity of our algorithm proves that the problem of computing a PNE for a given weighted single-commodity game[5] can be solved in polynomial time with respect to the size of the input game. Notice that we have to take into account not only the complexity of our algorithm (which is clearly polynomial), but also the complexity of computing a maximal set C of disjoint paths. Indeed, the computation of a *maximum* set of disjoint paths between two given nodes of a graph can be done in polynomial time, as shown in [13] (pp. 338-346 and proposition (7.44)).

3.2 Quality of the Equilibria

In this section, we show that, in the general case, our algorithm can be seen as a $\frac{4}{3}$-approximation to the problem of computing a configuration with optimum social cost, provided that we apply it to a maximum set of edge-disjoint paths.

[5] Thus also for an unweighted single-commodity game, since it is a special case of a weighted game.

Proposition 6. *Let $\Gamma = (N, G, (s,t), (w_i)_{i \in N})$ be a weighted game and let C be a maximum set of edge-disjoint $(s-t)$ paths in G. The configuration π computed by the algorithm* `w_round_robin` *on input (Γ, C) has a social cost $SC(\pi) \leq \frac{4}{3} \cdot SC^*$.*

Contrasting with this, it is not hard to prove that, when the input of the algorithm is an unweighted single-commodity game $\Gamma = (N, G, (s,t))$ and a maximum set C of (s,t) edge-disjoint paths, the algorithm computes a PNE with optimum social cost (which is, in turn, $\frac{|N|}{m}$, where m is the maximum number of $(s-t)$ disjoint paths).

Once we have analyzed the social cost of the PNE computed by our algorithm, we study the price of anarchy. Obtaining an upper and a lower bound of the social cost of any configuration of a weighted NMCG, it can be shown that the Price of anarchy is at most equal to the maximum number of edge-disjoint $(s-t)$ paths in the graph that describes the network.

Theorem 7. *Any weighted game $\Gamma = (N, G, (s,t), (w_i)_{i \in N})$ satisfies that $PA \leq m$, where m is the maximum number of $(s-t)$ edge-disjoint paths in G.*

4 Computation of Equilibria in Multi-Commodity Games

In this section we study the equilibria of unweighted multi-commodity NMC games. The technique developed in the previous section, which consisted in distributing players among a set of disjoint paths, can't be applied to this kind of games, since edge-disjoint paths among different commodities may not exist (and even if they do, a simple distribution of players among paths may not lead to a PNE). For that reason, we focus our analysis on games with a particular graph topology, or, as we call them, *l-common-edges* games.

Let us consider a NMC game Γ and let M be the set of the m commodities (s_i, t_i) of Γ. For any commodity $i \in M$, we denote by N_i the set of players playing in commodity i, i.e., the set of players choosing paths between the two nodes of commodity i. For any subset of commodities $S \subseteq M$, we denote by N_S the set of players playing in any commodity $i \in S$, $N_S = \bigcup_{i \in S} N_i$. In addition, $n_i = |N_i|$ and $n_S = |N_S|$ denote the cardinality of these sets. A *l-common-edges* game is a game whose graph has the following topology: for each commodity $i \in M$ there is a set L_i containing l_i paths between s_i and t_i which are edge-disjoint with respect to all other paths in the graph[6]. In addition to this set L_i of *private* paths, there also exists a set L'_i of l *public* $(s-t)$ paths which are edge-disjoint with respect to the other paths in the same set L'_i, but intersect with paths of the remaining commodities in M. More formally, there exists a set E^* of l *common* edges such that for each edge $e \in E^*$ and each commodity $i \in M$ there is exactly one path p_i^e in L'_i such that $e \in p_i^e$, whereas the remaining

[6] It could be shown that if these paths were the only ones in the graph, then a PNE could be computed in polynomial time applying the algorithm of the previous section to each one of the commodities.

edges in p_i^e are not part of any other path in the strategy set of any player. Hence, $L'_i = \{p_i^e \mid e \in E^*\}$. For each $S \subseteq M$, we denote by L'_S the set of all the public paths of commodities in S, $L'_S = \bigcup_{i \in S} L'_i$. Given an edge $e \in E^*$, then, we can define a set of m paths $P_e = \bigcup_{i \in M} p_i^e$ which contains, for each commodity $i \in M$, the only path p_i^e in L'_i which uses the edge e. Notice that, given the particular structure of this class of graphs, the congestion of any path $p_i^e \in L'_i$ will always be equal to the congestion of the edge e which, in turn, will be equal to the number of players allocated to paths in P_e.

4.1 Correctness and Complexity Issues

Let's now analyse the algorithm 2. In this analysis, we denote by k_S the quotient $\frac{n_S}{l+l_S}$. Given an unweighted multi-commodity NMC game, the algorithm starts (lines 2-5) by searching a subset of commodities which may be able to *dominate* all the set of common edges, that is, to be the only commodities whose players use those edges in such a way that players from other commodities can be satisfied even if they only use their private paths.

Lemma 8. *After the execution of lines 2-5 of Algorithm 2, the following statements hold:*

1. $\forall i \in M \setminus S \quad \left(\lfloor k_S \rfloor \geq \left\lceil \frac{n_i}{l_i} \right\rceil - 1\right)$
2. $\forall S' \subseteq M \left(|S'| < |S| \Rightarrow \exists i \in M \setminus S' \left(\lfloor k_S \rfloor < \left\lceil \frac{n_i}{l_i} \right\rceil - 1\right)\right)$
3. $\forall i \in S \left(\frac{n_i}{l_i} \geq k_S\right)$

Once the algorithm has determined which subset $S \subseteq M$ dominates the set E^* of common edges, it distributes players among paths in a particular manner. In the first place (l. 6-25), it distributes all the players from the commodities in S. The loop in lines 6-8 allocates $\lfloor k_S \rfloor$ players to every path of every commodity $i \in S$. Since, as we have seen, $n_i \geq l_i \cdot k_S$, enough players exist to allow this allocation. After the execution of the loop, $l_S \cdot \lfloor k_S \rfloor$ players have been allocated. Thus, there remain at least $k_S \cdot l$ unallocated players. This fact ensures the correct execution of the second loop (lines 9-10), given that it allocates exactly $\lfloor k_S \rfloor \cdot l$ players to paths in L'_S. After the execution of these loops, every path in L_S, as well as every edge in E^*, have a congestion of exactly $\lfloor k_S \rfloor$ units. Next (l. 11-14), the algorithm iterates again through all the private paths of every commodity in S, allocating –when possible– one more player to each of them. Similarly, the loop of lines 15-17 allocates to each group of paths P_e (for all $e \in E^*$) at most one more player from N_S. Therefore, after the execution of these two loops every path in L_S and every edge in E^* have congestion either $\lfloor k_S \rfloor$ or $\lfloor k_S \rfloor + 1$.

Let's now analyze the loop of lines 19-25:

Lemma 9. *At each iteration of the loop of lines 19-25 of Algorithm 2, the following propositions hold:*

1. *There exists a commodity $i \in S$ such that*

Algorithm 2: Computation of a PNE in multi-commodity games.

input : An unweight. l-common-edges game $\Gamma = (N, G, ((s_i, t_i))_{i \in N})$
output : A configuration π for the given game

begin
  `// Selection of a proper subset of commodities:`
  foreach $t := 1, \ldots, m$ **do**
    foreach $S \subseteq M$ *s.t.* $|S| = t$ **and while** S *is not proper* **do**
      if $\forall i \in M \setminus S \ \left(\lfloor k_S \rfloor \geq \left\lceil \frac{n_i}{l_i} \right\rceil - 1 \right)$ **then**
        S is a proper subset
  `// Distribution of players among paths:`
  foreach $i \in S$ **do**
    foreach $p \in L_i$ **do**
      Allocate $\lfloor k_S \rfloor$ players from N_i to path p
  foreach $e \in E^*$ **do**
    Allocate $\lfloor k_S \rfloor$ players from N_S to the paths in P_e
  foreach $i \in S$ **do**
    foreach $p \in L_i$ **do**
      if *there remain unallocated players in* N_i **then**
        Allocate one player from N_i to path p
  foreach $e \in E^*$ **do**
    if *there remain unallocated players in* N_S **then**
      Allocate one player from N_S to a suitable path $p \in P_e$
  Let N'_S be the set of players from N_S still not allocated
  while $N'_S \neq \emptyset$ **do**
    Let j be a player from N'_S
    Let $i \in S$ be a commodity such that
      1) There exists a path $p \in L_i$ with only $\lfloor k_S \rfloor$ allocated players, and
      2) There exists a player $j' \in N_i$ already allocated to a path $p' \in L'_i$
    Allocate player j' to path p and player j to path p'
    $N'_S := N'_S \setminus \{j\}$
  foreach $i \in M \setminus S$ **do**
    Allocate players in N_i to paths in L_i in a *Round-Robin* fashion
end

   (a) *There exists a path* $p \in L_i$ *with only* $\lfloor k_S \rfloor$ *allocated players, and*
   (b) *There exists a player* $j' \in N_i$ *already allocated to a path* $p' \in L'_i$
2. *Every path in* L_S *and every edge in* E^* *(and thus every path in* L'_S *have congestion either* $\lfloor k_S \rfloor$ *or* $\lfloor k_S \rfloor + 1$

At the end of the loop of lines 19-25, then, we have that for any commodity $i \in S$, any player $j \in N_i$ is allocated to a path with congestion at most $\lfloor k_S \rfloor + 1$. Since all the other paths available to j are in $L_S \cup L'_S$ and thus have congestion at least $\lfloor k_S \rfloor$, player j has no incentive to change his strategy.

With respect to the players of commodities in $M \setminus S$, the algorithm (l. 26-27) simply allocates them to paths in L_i in a Round-Robin fashion. We then have that, if n_i is divisible by l_i, all paths in L_i will have congestion $\frac{n_i}{l_i}$. Thus, for any commodity $i \in M \setminus S$ and any player $j \in N_i$, player j will have no incentive to change his strategy, since all the other paths in L_i have the same congestion $\frac{n_i}{l_i}$, whereas all the paths in L'_i have an edge in E^* with congestion at least $\lfloor k_S \rfloor \geq \left\lceil \frac{n_i}{l_i} \right\rceil - 1 = \frac{n_i}{l_i} - 1$ (recall lemmas 8 and 9).

On the other hand, if n_i is not divisible by l_i, all paths in L_i will have congestion either $\left\lfloor \frac{n_i}{l_i} \right\rfloor$ or $\left\lfloor \frac{n_i}{l_i} \right\rfloor + 1$. Thus, for any commodity $i \in M \setminus S$ and any player $j \in N_i$, player j will have no incentive to change his strategy, since the paths in L_i will have congestion at least $\left\lfloor \frac{n_i}{l_i} \right\rfloor$, and –as we have already seen– the paths in L'_i have congestion at least $\lfloor k_S \rfloor \geq \left\lceil \frac{n_i}{l_i} \right\rceil - 1 = \left\lfloor \frac{n_i}{l_i} \right\rfloor$. As no player has an incentive to change his strategy, we can conclude that the configuration computed by our algorithm is a PNE. Furthermore, if we consider that the number of commodities is fixed, the running time of the algorithm can be proven to be polynomial. Therefore, the following theorem is proven.

Theorem 10. *Let* $\Gamma = (N, G, ((s_i, t_i))_{i \in N})$ *be an unweighted* l-common-edges *game. Algorithm 2 computes a PNE for* Γ*; furthermore, if we consider the number of commodities to be a fixed constant* k*, the algorithm solves the problem of computing a PNE for unweighted* k*-commodities* l-common-edges *games in polynomial time.*

5 Conclusions

In this work we have studied *Network Maximum Congestion Games*, which model the behaviour of selfish users when they try to minimize the bottleneck of the network paths through which they route their packets. Even though the existence of Nash Equilibria is guaranteed for this kind of games (see [10–12]), we have taken a constructive approach and studied the computational complexity of computing a PNE, as well as the quality of the computed equilibria.

In the case of single-commodity NMC games, we have designed a polynomial time algorithm that computes a PNE for the general case of weighted users; this algorithm is a $\frac{4}{3}$-approximation for the problem of computing an optimum PNE, but if we do not allow players to have weights, then the algorithm directly

computes an optimum PNE. We have also proven that the Price of anarchy in single-commodity games is at most equal to the maximum number of edge-disjoint paths between the two distinguished nodes s and t of the game. In the case of multi-commodity NMC games, we have focused our research on some particular network topologies. We have given a polynomial time algorithm that computes a PNE in polynomial time when the number of commodities is constant and all the players have the right to use either the set of *private* edge-disjoint paths of their commodities or a set of *public* paths (each one containing one edge which is shared by all the users). A major direction of future research will be to characterize other cases in which the computation of a PNE can be done efficiently.

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. Rosenthal, R.W.: A class of games possessing pure-strategy Nash equilibria. International Journal of Game Theory **2**(1) (1973) 65–67
5. Fotakis, D., Kontogiannis, S., Spirakis, P.: Selfish unsplittable flows, Springer (2004) 593–605
6. Fotakis, D., Kontogiannis, S., Spirakis, P.: Symmetry in network congestion games: Pure equilibria and anarchy cost. Volume 3879., Springer (2006) 161–175
7. 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
8. Christodoulou, G., Koutsoupias, E.: The price of anarchy of finite congestion games. In: STOC'05, New York, ACM Press (2005) 67–73
9. Koutsoupias, E., Papadimitriou, C.: Worst-case equilibria. In: STACS'99. Volume 1563., Springer (1999) 404–413
10. Caragiannis, I., Galdi, C., Kaklamanis, C.: Network load games. LNCS **3827** (2005) 809–818
11. Busch, C., Magdon-Ismail, M.: Atomic Routing Games on Maximum Congestion. Lecture notes in computer science **4041** (2006) 79–91
12. Banner, R., Orda, A.: Bottleneck Routing Games in Communication Networks. Proceedings of the 25th INFOCOM Conference (2006)
13. Kleinberg, J., Tardos, E.: Algorithm Design. Addison-Wesley (2006)
14. Garey, M., Johnson, D.: Computers and Intractability: A Guide to the Theory of NP-Completeness. WH Freeman & Co. New York, NY, USA (1979)
15. Graham, R.: Bounds on Multiprocessing Timing Anomalies. SIAM Journal on Applied Mathematics **17**(2) (1969) 416–429

6 Appendix: Omited Proofs

We present in this appendix those proofs which, due to space restrictions, have been omitted in the previous sections.

Proposition 1. *Let $\pi = (p_1, \ldots, p_n)$ be a configuration for the unweighted NMC $\Gamma = (N, G, (s,t))$ and let $k = SC(\pi)$. Then,*

1. *If π is a PNE, then $\forall p \in P^{(s,t)}\ L_p(\pi) \in \{k, k-1\}$.*
2. *If $\forall p \in P^{(s,t)}\ L_p(\pi) = k$, then π is a PNE.*

Proof. We are going to show each part separately.

1. Since $k = SC(\pi)$, then any path $p \in P^{(s,t)}$ satisfies that $L_p(\pi) \leq k$. It remains to show that $L_p(\pi) \geq k-1$. Let us assume that there is a path $p \in P^{(s,t)}$ with congestion $L_p(\pi) < k-1$. Let $i \in N$ be one of the players with maximum cost, i.e. $c_i(\pi) = L_{p_i}(\pi) = k$. If player i changes strategy p_i for p then $L_p((\pi_{-i}, p)) \leq L_p(\pi)+1 < k \leq L_{p_i}(\pi)$. Hence, $c_i((\pi_{-i}, p)) < c_i(\pi)$ and this contradicts that π is a PNE.
2. Let us consider that any path $p \in P^{(s,t)}$ satisfies $L_p(\pi) = k$. If any player i changes his strategy p_i for p, the congestion of p is incremented at most by 1. Formally, $\forall i \in N\ \forall p \in P^{(s,t)}\ L_p((\pi_{-i}, p)) \in \{L_p(\pi), L_p(\pi)+1\} = \{k, k+1\}$. Hence, each player i has no incentive to change unilaterally his strategy, $L_p((\pi_{-i}, p)) \geq k = L_{p_i}(\pi)$, and then π is a PNE.

□

Proposition 2. *Let $\pi = (p_1, \ldots, p_n)$ be a configuration for the weighted NMC $\Gamma = (N, G, (s,t), (w_i)_{i \in N})$. If $C = \{p_i \mid i \in N\}$ is a set of $(s-t)$ edge-disjoint paths, then $\forall p \in C\ \forall e \in p\ \Lambda_e(\pi) = \Lambda_p(\pi)$, and hence $l_e(\pi) = L_p(\pi) = \sum_{i \in \Lambda_p(\pi)} w_i$*

Proof. Since all the paths $p \in C$ are edge-disjoint, then every edge e of any path $p \in C$, does not belong to any other path $p' \in C$. Hence, if a player i uses edge e and $e \in p$, it means that player i has selected the path p, and then $\Lambda_e(\pi) \subseteq \Lambda_p(\pi)$. On the other hand, if a player i has selected path $p \in C$ then i uses edge $e \in p$. Then $\Lambda_p(\pi) \subseteq \Lambda_e(\pi)$. So we have that $\forall p \in C\ \ \forall e \in p\ \ \Lambda_e(\pi) = \Lambda_p(\pi)$.

Furthermore, if we apply the definition of edge congestion, we have that for any $e \in p\ l_e(\pi) = \sum_{i \in \Lambda_e(\pi)} w_i = \sum_{i \in \Lambda_p(\pi)} w_i$ Hence, applying the definition of the path congestion we have that $L_p(\pi) = \max_{e' \in p}\ l_{e'}(\pi) = l_e(\pi) = \sum_{i \in \Lambda_p(\pi)} w_i$

□

In order to show Proposition 6 first we need to remind some properties related to directed graphs, the definition of a classical scheduling problem and, finally, we also need to prove properties related to social cost.

Theorem 11 (Menger's Theorem). *Let $G = (V, E)$ be a directed graph and let $s, t \in V$ be two nodes of G. The maximum number of edge-disjoint paths from s to t is equal to the minimum number of edges that should be eliminated in order to disconnect s from t.*

Proof. A nice proof of this theorem can be found in [13] (theorem (7.45)). □

Corollary 12. *For any directed graph $G = (V, E)$ and for any pair of nodes $s, t \in V$, there exists a set of edges $B \subseteq E$ such that every $(s-t)$ path contains at least one of its edges and the cardinality of B is equal to the maximum number of $(s-t)$ edge-disjoint paths in G. Moreover, every path in a maximum set of $(s-t)$ edge-disjoint paths contains exactly one of the edges of B*

Proof. Let $B \in E$ be a set with the minimum number of edges that disconnect s from t when they are eliminated. By Menger's Theorem, the cardinality of B is equal to the maximum number of $(s-t)$ edge-disjoint paths in G. Besides this, every path $p \in P^{(s,t)}$ contains at least one of the edges of B, otherwise if we delete all the edges of B, s will not be disconnected from t.

Finally, if $M \subseteq P^{(s,t)}$ is a set with a maximum number of $(s-t)$ edge-disjoint paths, then every path $p \in M$ has an edge in B and no one of these edges can appear in more than one path of M. Hence, there are as many disjoint paths in M as edges in B. □

Proposition 13. *Let $\Gamma = (N, G, (s,t), (w_i)_{i \in N})$ be a NMC game and let SC^* be the optimum social cost. Let C be a maximum set of $(s-t)$ edge-disjoint paths in G. Then there exists a configuration $\pi = (p_1, \ldots, p_n)$ such that for all i, $p_i \in C$ and $SC(\pi) = SC^*$.*

Proof. Let $\pi' = (p'_1, \ldots, p'_n)$ be a configuration with minimum social cost, i.e. $SC(\pi') = SC^*$. We are going to see that from this π' (in fact from any configuration π) we can define a new configuration π'' which only contains paths that belong to C and so that $SC(\pi'') = SC(\pi')$ (and in the case that π is any configuration then $SC(\pi'') \leq SC(\pi)$).

Let $G = (V, E)$ and let $B \subseteq E$ be the edge set of corollary 12. Then it satisfies that $|B| = |C|$ and each path $p \in C$ contains exactly one edge $e \in B$. Then we can define a bijection $f : B \to C$ so that for every $e \in B$, $f(e)$ is the only path in C that contains the edge e. Due to corollary 12 we also know that any path p'_i contains at least one edge of B. For each p'_i, let us select (arbitrarily) one of these edges and denote it by e_i.

Now, we can define a configuration π'' that only uses paths of C: $\pi'' = (f(e_1), f(e_2), \ldots, f(e_n))$. Notice that every player i that uses path $f(e_i)$ in π'' uses edge e_i in π'. Hence, by proposition 2, for all $i \in N$ we have that

$$L_{f(e_i)}(\pi'') = \sum_{k \in \Lambda_{f(e_i)}(\pi'')} w_k \leq \sum_{k \in \Lambda_{e_i}(\pi')} w_k = l_{e_i}(\pi') \leq SC(\pi')$$

Hence,

$$SC(\pi'') = \max_{i \in N} \; c_i(\pi'') = \max_{i \in N} \; L_{f(e_i)}(\pi'') \leq SC(\pi').$$

Since $SC(\pi') = SC^*$ then $SC(\pi'') = SC^*$. Notice that π'' is the desired configuration, it has an optimal social cost and all its paths p''_i belong to C. □

Let us remind a well-known NP-complete problem named as MINIMUM MULTIPROCESSOR SCHEDULING (see [14], problem [SS8]): Given a set $M = \{1, 2, \ldots, l\}$ of identical machines and given a set $J = \{1, 2, \ldots, n\}$ of tasks where each task j requires a processing time t_j, the problem consists in assigning a machine to each task so that the load of all machines has to be kept as balanced as possible. Formally, the problem consists in computing $\mu = (m_1, \ldots, m_n)$ (where $m_j \in M$ is the machine assigned to task $j \in J$) so that minimizes the *makespan* defined as

$$MS(\mu) = \max_{m \in M} \sum_{j \in \Lambda_m(\mu)} t_j$$

where $\Lambda_m((m_1, \ldots, m_n)) = \{j \in J \mid m_j = m\}$.

In order to find an upper bound of the ratio between the social cost of the configuration computed by the algorithm `w_round_robin` and the optimum social cost, we use a result published in [15]. This result states that the algorithm LPT computes an assignment that is a $\frac{4}{3}$-approximation to the makespan optimum.

Definition 14. *Let $\Gamma = (N, G, (s,t), (w_i)_{i \in N})$ a NMC game and let C be a set of $(s-t)$ edge-disjoint paths of G. Let $\Omega = (J, M, (t_j)_{j \in J})$ be an instance of the* MINIMUM MULTIPROCESSOR SCHEDULING *problem. We say that Ω and (Γ, C) are* equivalent *if $J = N$, $t_j = w_j$ for all $j \in J$ and $|M| = |C|$. If (Γ, C) and Ω are equivalent, we also define the equivalence between a configuration of Γ and an assignment for Ω. Let $\phi : C \to M$ be the bijection that assigns each path $c_j \in C$ to the machine $m_j \in M$. We say that a configuration $\pi = (p_1, \ldots, p_n)$ of Γ and an assignment $\mu = (m_1, \ldots, m_n)$ for Ω are equivalent if $m_i = \phi(p_i)$ for all $i \in N$. Notice that π can only be composed of paths that belong to C. And since ϕ is a bijection, $\pi = (p_1, \ldots, p_n)$ can only be equivalent to $\mu = (\phi(p_1), \ldots, \phi(p_n))$ and vice versa.*

Proposition 15. *Let $\Gamma = (N, G, (s,t), (w_i)_{i \in N})$ be a NMC game and let C be a set of $(s-t)$ edge-disjoint paths of G. Let $\Omega = (J, M, (t_j)_{j \in J})$ be an instance of the* MINIMUM MULTIPROCESSOR SCHEDULING *problem equivalent to (Γ, C). Let $\pi = (p_1, \ldots, p_n)$ be a configuration of Γ and $\mu = (m_1, \ldots, m_n)$ an assignment for Ω. If π and μ are equivalent, then $SC(\pi) = MS(\mu)$.*

Proof. It is not hard to see that if π i μ are equivalent, then the set of users of any path $p_i \in C$ is equal to the set of users of any machine $m_i \in M$:

$$\begin{aligned} \Lambda_{p_i}(\pi) &= \{k \in N \mid p_k = p_i\} = \{k \in J \mid \phi(p_k) = \phi(p_i)\} = \\ &= \{k \in J \mid m_k = m_i\} = \Lambda_{m_i}(\mu) \end{aligned}$$

Furthermore, since π and μ are equivalent, and the set of paths used by π is a set of edge-disjoint paths then we can apply proposition 2, obtaining that

$$\begin{aligned} SC(\pi) = \max_{i \in N} \; c_i(\pi) = \max_{i \in N} \; L_{p_i}(\pi) = \max_{i \in N} \sum_{k \in \Lambda_{p_i}(\pi)} w_k = \\ = \max_{i \in J} \sum_{k \in \Lambda_{m_i}(\mu)} t_k = \max_{m \in M} \sum_{k \in \Lambda_m(\mu)} t_k = MS(\mu) \end{aligned}$$

□

Now we have all what is necessary to show an upper bound of the ratio between the PNE computed by our algorithm and the social optimum.

Proposition 6. *Let $\Gamma = (N, G, (s,t), (w_i)_{i\in N})$ be a weighted game and let C be a maximum set of edge-disjoint $(s-t)$ paths in G. The configuration π computed by the algorithm* `w_round_robin` *on input (Γ, C) has a social cost $SC(\pi) \leq \frac{4}{3} \cdot SC^*$.*

Proof. Let $\Omega = (J, M, (t_j)_{j\in J})$ be an instance of the Minimum Multiprocessor Scheduling problem equivalent to (Γ, C). Let μ an assignment computed by LPT on input Ω. Notice that $\pi = w\_round\_robin(\Gamma, C)$ and μ are equivalent, the algorithms `w_round_robin` and LPT works in the same way assigning players/tasks to paths/machines.

In [15] it is shown that the *makespan* of μ is at most $\frac{4}{3}$ times greater than the *makespan* of the optimum. Let μ^* one of the assignments that induces the optimum *makespan*. Hence, $MS(\mu) \leq \frac{4}{3} \cdot MS(\mu^*)$. Let π^* the configuration of Γ equivalent to σ^*. By proposition 15, we have that $SC(\pi) = MS(\mu) \leq \frac{4}{3} \cdot MS(\mu^*) = \frac{4}{3} \cdot SC(\pi^*)$. It remains only to show that π^* is a configuration with an optimum social cost.

Let us suppose that π^* is not optimum, i.e. $SC^* < SC(\pi^*)$. By proposition 13, there exists another configuration π' all its paths belong to C and $SC(\pi') = SC^*$. Then μ' is equivalent to π' and by proposition 15, $MS(\mu') = SC(\pi') = SC^* < SC(\pi^*) = MS(\mu^*)$. But this is not possible, μ^* is the optimum assignment. Hence, π^* has an optimum social cost. □

In order to show Theorem 7 we need to prove the following results.

Proposition 16. *Let $\Gamma = (N, G, (s,t), (w_i)_{i\in N})$ be a NMC game. Let m be the maximum number of $(s-t)$ edge-disjoint paths in G and $w = \sum_{i\in N} w_i$. If π is any configuration of Γ, then $SC(\pi) \geq \lceil \frac{w}{m} \rceil$.*

Proof. By corollary 12 there exists a set $B \subseteq E$ of m edges so that any $(s-t)$ path contains at least one edge $e \in B$. Hence in any configuration π of Γ every one of its components contains at least one of the m edges of B. This fact implies that

$$\bigcup_{e\in B} \Lambda_e(\pi) = N$$

and then

$$\sum_{e\in B} l_e(\pi) = \sum_{e\in B} \sum_{i\in\Lambda_e(\pi)} w_i \geq \sum_{i\in\bigcup_{e\in B}\Lambda_e(\pi)} w_i = \sum_{i\in N} w_i = w$$

This means that there exists at least one edge $e^* \in B$ such that $l_{e^*}(\pi) \geq \left\lceil \frac{w}{|B|} \right\rceil = \lceil \frac{w}{m} \rceil$. Hence, we can bound the social cost as follows:

$$SC(\pi) = \max_{e\in E} \; l_e(\pi) \geq l_{e^*}(\pi) = \left\lceil \frac{w}{m} \right\rceil$$

□

Proposition 17. *Let $\pi = (p_1, \ldots, p_n)$ be a configuration of a NMC game $\Gamma = (N, G, (s,t), (w_i)_{i \in N})$. Then $SC(\pi) \leq w$, where $w = \sum_{i \in N} w_i$.*

Proof. Since for any $e \in E$ we have that $\Lambda_e(\pi) \subseteq N$ then

$$SC(\pi) = \max_{e \in E} \; l_e(\pi) = \max_{e \in E} \sum_{i \in \Lambda_e(\pi)} w_i \leq \sum_{i \in N} w_i = w$$

□

Once we have shown the two previous results, we can bound the Price of anarchy of weighted single-com NMC games:

Theorem 7. *Any weighted game $\Gamma = (N, G, (s,t), (w_i)_{i \in N})$ satisfies that $PA \leq m$, where m is the maximum number of $(s-t)$ edge-disjoint paths in G.*

Proof. Considering the bounds shown in Propositions 16 and 17:

$$PA = \max_{\pi \text{ s PNE}} \; \frac{SC(\pi)}{SC^*} \leq \frac{w}{\left\lceil \frac{w}{m} \right\rceil} \leq \frac{w}{\frac{w}{m}} = m$$

□

Lemma 8. *After the execution of lines 2-5 of Algorithm 2, the following statements hold:*

1. $\forall i \in M \setminus S \;\; \left(\lfloor k_S \rfloor \geq \left\lceil \frac{n_i}{l_i} \right\rceil - 1 \right)$
2. $\forall S' \subseteq M \left(|S'| < |S| \Rightarrow \exists i \in M \setminus S' \left(\lfloor k_S \rfloor < \left\lceil \frac{n_i}{l_i} \right\rceil - 1 \right) \right)$
3. $\forall i \in S \left(\frac{n_i}{l_i} \geq k_S \right)$

Proof. The first proposition is exactly the condition in line 4. Notice that this condition will evaluate to true sooner or later, since it always holds when $S = M$. The second proposition is directly implied by the order in which the algorithm considers the subsets of M.

Finally, let's prove the last proposition. Suppose that there exists a commodity $i \in S$ for which the proposition does not hold, i.e. $\frac{n_i}{l_i} < k_S$ or, equivalently, $n_i < k_S \cdot l_i$. Thus,

$$\begin{aligned} k_{S \setminus \{i\}} &= \frac{n_{S \setminus \{i\}}}{l + l_{S \setminus \{i\}}} = \frac{n_S - n_i}{l + l_S - l_i} > \frac{n_S - k_S \cdot l_i}{l + l_S - l_i} = \frac{n_S - (\frac{n_S}{l + l_S}) \cdot l_i}{l + l_S - l_i} \\ &= \frac{n_S (l + l_S - l_i)}{(l + l_S)(l + l_S - l_i)} = \frac{n_S}{l + l_S} = k_S \end{aligned}$$

Therefore, it holds that

$$k_{S \setminus \{i\}} > k_S > \frac{n_i}{l_i} \geq \left\lceil \frac{n_i}{l_i} \right\rceil - 1$$

Since $\left\lceil \frac{n_i}{l_i} \right\rceil - 1$ is an integer, $\lfloor k_{S\setminus\{i\}} \rfloor \geq \left\lceil \frac{n_i}{l_i} \right\rceil - 1$. On the other hand, thanks to the first proposition of the lemma we can state that

$$\forall i' \in M \setminus S \left(\lfloor k_{S\setminus\{i\}} \rfloor \geq \lfloor k_S \rfloor \geq \left\lceil \frac{n_{i'}}{l_{i'}} \right\rceil - 1 \right)$$

Combining the two last propositions, we have that

$$\forall i' \in M \setminus (S \setminus \{i\}) \left(\lfloor k_{S\setminus\{i\}} \rfloor \geq \left\lceil \frac{n_{i'}}{l_{i'}} \right\rceil - 1 \right)$$

but, since $|S \setminus \{i\}| < |S|$, that contradicts what we have shown in the second proposition of the lemma. Consequently, the third proposition must hold. □

Lemma 9. *At each iteration of the loop of lines 19-24 of Algorithm 2, the following propositions hold:*

1. *There exists a commodity $i \in S$ such that*
   (a) *There exists a path $p \in L_i$ with only $\lfloor k_S \rfloor$ allocated players, and*
   (b) *There exists a player $j' \in N_i$ already allocated to a path $p' \in L'_i$*
2. *Every path in L_S and every edge in E^* (and thus every path in L'_S have congestion either $\lfloor k_S \rfloor$ or $\lfloor k_S \rfloor + 1$*

Proof. 1. Consider the set $S^* \subseteq S$ of commodities $i \in S$ with at least one path in L_i having congestion $\lfloor k_S \rfloor$. Let's first see that, at any iteration of the loop, this set S^* is not empty. Since we have entered the loop, there remain some unallocated players in $N'_S \subseteq N_S$. That implies that all paths in L'_S have congestion $\lfloor k_S \rfloor + 1$. If there were a path in L'_S with congestion $\lfloor k_S \rfloor$, then there would be a path with that congestion in each L'_i, $i \in S$, due to the nature of our graph topology. But that is not possible, otherwise the loop of lines 15-17 would have allocated to these paths the unallocated players from N'_S. It is then impossible that all commodities $i \in S$ have congestion $\lfloor k_S \rfloor + 1$, otherwise the total number of players allocated would be $(l_S + l) \cdot (\lfloor k_S \rfloor + 1)$, which is strictly greater than n_S, the actual number of players. Thus, we have that S^* contains at least one commodity.
In turn, that fact implies that no player of N_{S^*} is in N'_S, the set of unallocated players; otherwise, the loop of lines 11-14 would have allocated him to one of the paths of his commodity with congestion $\lfloor k_S \rfloor$. All the players in N'_S, then, come from commodities in $S \setminus S^*$ (and that also proves that $S^* \subsetneq S$, that is, that $S \setminus S^*$ is not empty).
Let's now suppose that all players in N_{S^*} are allocated to paths in L_{S^*}. One one hand, that means that all shared paths in L'_S are allocated only to players in $N_{S\setminus S^*}$. Since, by definition of S^*, all paths in $L_{S\setminus S^*}$ have congestion $\lfloor k_S \rfloor + 1$, the number of players in $N_{S\setminus S^*}$ already allocated equals $(\lfloor k_S \rfloor + 1) \cdot (l + l_{S\setminus S^*})$. Furthermore, there is at least one more (unallocated) player in $N_{S\setminus S^*}$, otherwise we would have not entered the loop. Therefore, the total number of players in $N_{S\setminus S^*}$ is $n_{S\setminus S^*} \geq (\lfloor k_S \rfloor + 1) \cdot (l + l_{S\setminus S^*}) + 1 > (\lfloor k_S \rfloor + 1) \cdot (l + l_{S\setminus S^*})$, so we have that $\frac{n_{S\setminus S^*}}{l + l_{S\setminus S^*}} > \lfloor k_S \rfloor + 1$.

On the other hand, that means that the number of players in each commodity $i^* \in S^*$ is $n_{i^*} < (\lfloor k_S \rfloor + 1) \cdot l_{i^*}$. Therefore, $\frac{n_{i^*}}{l_{i^*}} < \lfloor k_S \rfloor + 1 < \frac{n_{S \setminus S^*}}{l + l_{S \setminus S^*}} = k_{S \setminus S^*}$, using the previous inequality. In turn, the total number of players in N_{S^*} is $n_{S^*} < (\lfloor k_S \rfloor + 1) \cdot l_{S^*}$, so we have that $\frac{n_{S^*}}{l_{S^*}} < (\lfloor k_S \rfloor + 1) < k_{S \setminus S^*}$, using the same inequality as before. Therefore,

$$k_S = \frac{n_S}{l + l_S} = \frac{n_{S^*} + n_{S \setminus S^*}}{l + l_{S^*} + l_{S \setminus S^*}} < \frac{n_{S \setminus S^*}}{l + l_{S \setminus S^*}} = k_{S \setminus S^*}$$

which makes use of the general fact that $\frac{c}{d} < \frac{a}{b} \Rightarrow \frac{a+c}{b+d} < \frac{a}{b}$.
Now, we can apply lemma 8 to state that

$$\forall i \in M \setminus S \quad \left(k_{S \setminus S^*} > k_S \geq \lfloor k_S \rfloor \geq \left\lceil \frac{n_i}{l_i} \right\rceil - 1 \right)$$

while, as we have just seen, $\forall i^* \in S^* \quad \left(k_{S \setminus S^*} > \frac{n_{i^*}}{l_{i^*}} > \left\lceil \frac{n_{i^*}}{l_{i^*}} \right\rceil \right)$. Merging both assertions, we have that

$$\forall i \in M \setminus (S \setminus S^*) \quad \left(k_{S \setminus S^*} > \left\lceil \frac{n_i}{l_i} \right\rceil - 1 \right)$$

which implies that

$$\forall i \in M \setminus (S \setminus S^*) \quad \left(\lfloor k_{S \setminus S^*} \rfloor \geq \left\lceil \frac{n_i}{l_i} \right\rceil - 1 \right)$$

since $\left\lceil \frac{n_i}{l_i} \right\rceil - 1$ is an integer.
This last assertion, however, contradicts the second proposition of lemma 8. For that reason, we have to conclude that there exists at least one player in N_{S^*} which is allocated to a path in L'_S, and the proposition is proven.

2. The second proposition is not difficult to prove, since the algorithm first reallocates a player from a path with congestion $\lfloor k_S \rfloor + 1$ to a path with congestion $\lfloor k_S \rfloor$ and then allocates a player to the first path.

□