TEL-AVIV UNIVERSITY

RAYMOND AND BEVERLY SACKLER

FACULTY OF EXACT SCIENCES

SCHOOL OF COMPUTER SCIENCE

On the Price of Stability for Designing Undirected Networks with Fair Cost Allocations

This thesis was submitted in partial fulfillment of the requirements for the M.Sc. degree in

the School of School of Computer Science, Tel-Aviv University

by

Svetlana Olonetsky

The research work for this thesis has been carried out at Tel-Aviv University

under the supervision of Prof. Haim Kaplan and Prof. Amos Fiat

February 2006

Acknowledgements

I wish to thank Prof. Haim Kaplan and Prof. Amos Fiat for their guidance and help during the work on my thesis, and for introducing the study of game theory to me. I wish to thank them for spending so much time with me, discussing the problem and reading and commenting this manuscript.

Abstract

In this work we address the open problem of bounding the price of stability for a network design game with fair cost allocations in undirected graphs posed in [ADK$^+$04]. For the version of this problem that we consider, every vertex is associated with a selfish player, and there is a distinguished source vertex to which all players must connect. We show that the price of stability is $O(\log \log n)$.

Contents

Chapter 1

Introduction

In many networks, including the Internet, no central authority exists. The networks are built, maintained, and used by noncooperative selfish players. The lack of regulation typically degrades the performance of the network, since every player operates for her own benefit, and is not interested in the social optimum. A lot of research has been performed recently in trying to analyze and model networks subject to such selfish behavior. For that purpose concepts and techniques from game theory have been borrowed.

1.1 Nash equilibrium

Game theory can be roughly divided into two areas: *non-cooperative* (or strategic) games and *cooperative* (or coalitional) games. In this work we consider only non-cooperative games.

A key goal of game theory is to study the stable states where the players have reached some kind of equilibrium. Several types of equilibria have been defined and studied, among them the *Nash equilibrium* [Nas50] is the most famous and well studied.

Informally, a Nash equilibrium is a kind of optimal collective state in a game involving two or more players, where no player can gain by changing only her own strategy.

Formally, we need the following definitions. A non-cooperative game with n players consists of a set of pure strategies $\bar{S}_i$ for each player i. A *mixed strategy* of player i is a probability distribution on $\bar{S}_i$. We define a *pure state* S of the game to be a tuple $(S_1, S_2, \ldots, S_n)$, where $S_i \in \bar{S}_i$ is a pure strategy of player i. More generally a *mixed state* of the game is a tuple $(P_1, P_2, \ldots, P_n)$ where P_i is a mixed strategy of player i. We can represent every mixed state as a probability distribution (the joint distribution $P_1 \times P_2 \cdots \times P_n$) on the pure states. The cost of player i in every pure state S is specified as part of the definition of the game and denoted by $C_S(i)$. The cost of player i in a mixed state is defined to be the appropriate convex combination (defined by $P_1 \times P_2 \cdots \times P_n$) of the costs of the pure states.

A state S is a *Nash equilibrium* if for every player i, and for every state S' that differs from state S only by the strategy of player i (that is $S_i \neq S'_i$ and $S_j = S'_j$, for $j \neq i$), we have that $C_S(i) \leq C_{S'}(i)$. If S is pure we say that the *Nash equilibrium is pure* and otherwise we say that the *Nash equilibrium is mixed*. Nash [Nas50] proved that every game with a finite number of strategies has at least one Nash equilibrium (that could be mixed).

Often there is a natural global cost function that assigns a cost to each state. (For example one such cost function could be the sum of the costs of all players in the state.) When there is such a cost function we define a *social optimum* to be a state that minimizes this global cost function. We denote such a state by OPT, and the associated cost by $C(OPT)$. In order to compare the cost of selfish behavior to $C(OPT)$ the following concepts were defined.

1. The *price of anarchy* [KP99, Pap01] is the ratio between the cost of the most expensive Nash equilibria and the cost of the social optimum.

$$\text{Price Of Anarchy} = \max_{S \text{ is a Nash E.}} \frac{C(S)}{C(OPT)} \tag{1.1}$$

2. The *price of stability* [ADK$^+$04][1] is the ratio between the cost of the least expensive Nash

[1] The best Nash equilbria or "optimistic price of anarchy" was also considered in [ADTW03, CSM04b, CK05a].

equilibria and the cost of the social optimum.

$$\text{Price Of Stability} = \min_{S \text{ is a Nash E.}} \frac{C(S)}{C(OPT)} \; . \tag{1.2}$$

The price of stability is motivated by the scenario where one may have some centralized control for a limited time when the network is set-up. But, once the network is up and running, it should be stable without central control. Of course, the price of stability is not larger than the price of anarchy.

1.2 Congestion games

Much of the work so far on network games has been focused on congestion games [Ros72, MS96]. In a congestion game there are n players, and a set E of resources. The possible strategies of a player is to select a subset of the resources out of a collection of possible subsets. In a congestion game every resource $e \in E$ charges each player i such that $e \in S_i$. The cost of player i, $C_S(i)$, is the sum of the charges that i gets from all resources in S_i.

In an *unweighted congestion* game the amount that resource e charges each player using it depends only on the total number of the players using e. Specifically, each resource e has a function c_e such that if m is the number of players using resource e then $c_e(m)$ is the charge of resource e to each these m players. Given a state $S = (S_1, S_2, \ldots, S_n)$, let $x_s(e)$ be the number of players that use resource $e \in E$ in S, that is $x_s(e) = |\{i : e \in S_i\}|$. So formally the cost of player i in state S is $C_S(i) = \sum_{e \in S_i} c_e(x_s(e))$. In an unweighted congestion game the quantity $x_s(e)$ is referred to as the *congestion* of e in state S.

In a *weighted congestion* game, every player i has a weight w_i. The cost charged by resource e to a player using it, depends both on the weight of the player, and the total weight of all players using e. In a weighted congestion game c_e is a function with two parameters, defined such that

$c_e(w, W)$ is the charge to a player of weight w using resource e when the total weight of all players using e is W. In a weighted congestion game the cost of player i is $C_S(i) = \sum_{e \in S_i} c_e(w_i, w_s(e))$ where $w_s(e)$ is the total weight of players using e in state S. In a weighted congestion game the quantity $w_s(e)$ instead of $x_s(e)$ is referred to as the *congestion* of e in state S.

We recall that Nash proved [Nas50] that every game with finite number of strategies has a Nash equilibrium. Rosenthal [Ros72] proved that every unweighted congestion game has a pure Nash equilibrium.

Monderer and Shapley [MS96] defined several kinds of *potential games*. A game is an *ordinal potential game* if there exists a function Φ mapping each state to a real number with the following property. For every two states S and S', that differ only in the strategy of player i, and $C_{S'}(i) < C_S(i)$, then $\Phi(S') < \Phi(S)$. A game is an *exact potential game* if $C_S(i) - C_{S'}(i) = \Phi(S) - \Phi(S')$. The function Φ of a potential game is called a *potential function.* The most important property of potential games is that they have a pure Nash equilibrium. In fact, every series of improving moves (each of a single player) converges to a pure Nash equilibrium.

It is straightforward to check that every unweighted congestion game is an exact potential game [MS96] with respect to the potential function

$$\Phi(S) = \sum_{e \in E} \sum_{j=1}^{x_s(e)} c_e(j) \ . \tag{1.3}$$

Libman and Orda [LO01] give an example of a weighted congestion game that doesn't have a pure Nash equilibrium. Fotakis *et al.* [FKS04] considered a weighted congestion game such that a resource charges the players using it by a linear (in the congestion) cost function. Specifically, the cost charged by resource e to player i with weight w_i is defined to be $c_e(w_i, W) = w_i(a_e W + b_e)$ where a_e and b_e are nonnegative constants associated with resource e, and W is a total weight of players that use e. Fotakis *et al.* proved that the function

$$\Phi(S) = \sum_{i=1}^{n} \sum_{e \in S_i} c_e(w_i, w_s(e) + w_i) = \sum_{e \in E} (a_e w_s^2(e) + 2b_e w_s(e)) + \sum_{i=1}^{n} \sum_{e \in S_i} a_e w_i^2 \ ,$$

is an ordinal potential function of this game. (Recall that $w_s(e) = \sum_{i|e \in S_i} w_i$.) Therefore this game has a pure Nash equilibrium.

1.3 Network Congestion Games

We are interested in a special subclass of congestion games called *network congestion games*. In a network congestion game the resources are the edges of some directed or undirected network $G = (V, E)$. Player i is associated with two vertices $s_i, t_i \in V$ that she wants to connect. The set of pure strategies of player i is the set of all paths that connect s_i and t_i.

1.3.1 Network congestion games on a graph with parallel links.

Koutsoupias and Papadimitriou [KP99] considered a network congestion game on a network with two vertices s, and t and m parallel edges between them. Each of n players wants to connect s to t. In the unweighted version of this game the cost that edge e charges each player using it is determined by the linear function $c_e(m) = a_e m$ where m is the total number of players using e. In the weighted version of the same the cost charged by edge e to each player using it is $c_e(W) = a_e W$ where W is the total weight of the players using e. When the constants a_e are equal for all edges, we call these edges *identical*.

Koutsoupias and Papadimitriou considered the social cost function which is the maximum cost of a player. That is a social optimum is a state where the maximum cost of a player is minimized. This game corresponds to a scheduling problem with m machines (edges) and n independent tasks (players). If all the edges are identical, players are unweighted, and each player chooses an edge uniformly at random then this problem is identical to throwing n balls uniformly at random into m bins. The social cost function is then the expected maximum number of balls in a bin.

Koutsoupias and Papadimitriou [KP99] showed a lower bound of $\Omega(\frac{\log m}{\log\log m})$ on the price of anarchy of the weighted version of the game, when the edges are identical, and the number of players is equal to the number of edges (that is $n = m$). Mavronikolas and Spirakis [?] proved a matching upper bound of $\Theta(\frac{\log m}{\log\log m})$ on the price of anarchy, for any $n \leq m$. Their upper bound holds also for nonidentical edges and unweighted players. However, Mavronikolas and Spirakis proved the upper bound only for *fully-mixed Nash equilibrium* which are Nash equilibria in which every player uses each of her pure strategies with non-zero probability.

Czumaj and Vöcking [CV02] and independently Koutsoupias, Mavronikolas and Spirakis [KMS02] proved that the upper bound of $O(\frac{\log m}{\log\log m})$ on the price of anarchy holds for any Nash equilibrium not necessarily a fully-mixed Nash equilibrium, when edges are identical. Czumaj and Vöcking [CV02] also extended these results to m nonidentical edges and weighted players and showed that the price of anarchy in this case is $\Theta(\frac{\log m}{\log\log\log m})$.

1.3.2 Network congestion games on general graphs

Roughgarden and Tardos [RT02] considered the following network congestion game on a general graph. In this game we are given a set of source-sink pair (s_i, t_i) and a demand of traffic to send from s_i to t_i. We think of this traffic as split among infinitely many players, each responsible to deliver the same infinitesimally small share of the demand (which corresponds to the weight of the player in our terminology; all players have the same weight). Each player picks a path from s_i to t_i along which she delivers her flow. Each edge e charges each player using it by her infinitesimally small weight times some function $c_e(W)$ of the total weight W of all players using e (c_e is sometimes called the *latency of* e) Thereby the total charges accumulated by an edge e is

$W * c_e(W)$. Roughgarden and Tardos considered the social cost function which equals to the sum of the charges accumulated by all edges (which is the sum of the costs of all players). They proved that in this game the Nash equilibrium is unique, and therefore the price of stability is equal to the price of anarchy. Roughgarden and Tardos also proved that for linear cost functions, of the form $c_e(W) = a_e W + b_e$, where a_e and b_e are nonnegative constants, the price of anarchy is exactly 4/3. Roughgarden [Rou02] extended the results for polynomial cost functions of degree at most p, $(c_e(W) = \sum_{i=0}^{p} a_{e,i} W^i)$, and proved that in this case the price of anarchy is $O(\frac{p}{\log p})$. Roughgarden and Tardos [RT02] also proved that for general continuous and nondecreasing cost functions the total cost of the paths chosen by the selfish players is no more than the total cost incurred by optimally routing twice as much traffic between every source and its sink.

Roughgarden [Rou04] considered the same game with a social cost function that equals to the maximum cost of a player[2]. Roughgarden [Rou04] showed that if for every e the function c_e is continuous and nondecreasing, and there is only a single pair (s,t), then the price of anarchy is exactly $n-1$ where n is the number of vertices. On the other hand Correa, Schulz, and Moses [CSM04a] proved that there is an equilibrium whose maximum latency is within a constant factor of the optimal solution.

Recently, Awerbuch, Azar, and Epstein [AAE05], and Christodoulou and Koutsoupias [CK05b], studied the *atomic* version of the same game, where each player controls a fixed amount of traffic. The strategy of a player i is to choose a single path connecting pair of vertices s_i, and t_i and to pass all her traffic on this path. The players are *weighted* if each player may have a different demand to deliver from s_i to t_i, and *unweighted* if all demands are equal. Awerbuch, Azar, and Epstein [AAE05], and Christodoulou and Koutsoupias [CK05b] proved that for linear cost functions

[2]Since all players are equal this is the same as the maximum over every path P used by some player in state S of $\sum_{e \in P} c_e(w_s(e))$

and unweighted players the price of anarchy is 2.5. Awerbuch, Azar, and Epstein showed that for weighted players and linear cost functions the price of anarchy is 2.618. These bounds hold for both pure and mixed strategies, and they are tight for a pure Nash equilibrium. Recently, Christodoulou and Koutsoupias [CK05a] also gave an upper bound of 1.6 on the price of stability for linear cost functions. They also prove a lower bound of 1.577 on price of stability. It is an open problem to close this gap. In addition, they generalized the results for polynomial cost functions [AAE05, CK05b], and considered the social cost function which equals to the maximum latency of a player, rather than the sum of the latencies of all players [CK05b].

1.3.3 Network games with fair cost allocation

The game which we analyze was introduced by Anshelevich *et al.* [ADK+04]. They considered a network congestion game with n players and fair cost allocation. In this game, every player has a pair of vertices (s_i, t_i) that she wishes to connect. The cost of an edge e is shared equally by all players i whose chosen path $p_i = s_i, \ldots, t_i$ includes e. The cost of the player in this game is given by

$$C_S(i) = \sum_{e \in S_i} c_e(x_s(e)) = \frac{c_e}{x_s(e)}$$

where c_e is the cost of the edge $e \in E$ and $x_s(e)$ is the number of players that use edge e in state S. This game is a congestion game, hence it follows [Ros72, MS96] that a pure strategy Nash equilibrium always exists.

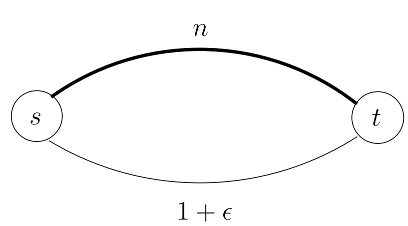


Figure 1.1: A network with price of anarchy $\Theta(n)$.

The social cost function associated with this game is the sum of the costs of all edges used by the players. The price of anarchy of this game is $\Theta(n)$. The upper bound follows since in equilibrium no player pays more that the length of the shortest path from s_i to t_i, and the length of each such shortest path is smaller than the cost of the optimal solution. To see the lower bound consider the network in Figure 1.1. Suppose n players wish to connect s to t. If all players use the edge of cost n between s and t, then every player pays 1, and her alternative strategy is of cost $1+\epsilon$. Therefore this state is a Nash equilibrium with cost n whereas in the optimal solution all players use the other edge and the social cost is $1+\epsilon$.

Anshelevich *et al.* considered the price of stability of this game. Substituting the cost function which we consider here in the standard potential function for congestion games (1.3) we obtain that the potential function of this game is

$$\Phi(S) = \sum_{e\in E}\sum_{j=1}^{x_s(e)} c_e(j) = \sum_{e\in E}\sum_{j=1}^{x_s(e)} \frac{c_e}{j} = \sum_{e\in E} c_e H(x_s(e)) \ , \tag{1.4}$$

where $H(n) = 1 + \frac{1}{2} + \ldots + \frac{1}{n}$. Since $x_s(e) \le n$, $H(x_s(e)) \le H(n)$, and

$$\Phi(S) \le \sum_{e\in E} c_e H(n) \le H(n)\sum_{e\in E} c_e \le H(n)C(S) \ .$$

If we start from OPT and make a sequence of improving moves we reach a state S such that $\Phi(S) \le \Phi(OPT) \le H(n)C(OPT)$. Since $\Phi(S) \ge \sum_{e\in S} c_e = C(S)$ we obtain that $C(S) \le H(n)C(OPT)$. Therefore the price of stability is $H(n) = O(\log n)$.

Anshelevich *et al.* [ADK+04] show that this bound is tight for a directed network, and left open the question of whether there is a tighter bound for undirected graphs. Furthermore, for the case of two players and an undirected graph with a single source Anshelevich *et al.* prove a tight bound on the price of stability of 4/3 which is less than $H(2) = 3/2$. Anshelevich *et al.* left open the question of whether there is a tighter bound for undirected graphs, in general.

Anshelevich *et al.* also considered the weighted version of this game. In the weighted version of the game player i has weight w_i and the cost of the player for using edge e is $C_S(i) = \frac{w_i}{w_s(e)} c_e$. Anshelevich *et al.* showed that for 2 weighted players it is a potential game, and hence a Nash equilibrium exists.

1.4 Other network games

In [ADTW03] Anshelevitz *et al.* considered a network congestion game under a different cost model. In this model every player must connect some set of terminals. Players can offer to pay any fraction of the cost of an edge. If the total contribution of players towards the purchase price of an edge is sufficient (reaches the price of the edge) then the edge is purchased. (In particular players can offer nothing for an edge.) A solution is an equilibrium if no player can benefit from changing her strategy. In this model there are instances where only mixed strategy Nash equilibria exist. Anshelevitz *et al.* show that it is NP-hard to determine if a pure strategy Nash equilibrium exists. The price of anarchy of this game is $\Theta(n)$. This follows using the network in Figure 1.1 and the same argument as in Section 1.3.3. When all players have a common source Anshelevitz *et al.* show that the price of stability is 1.

Fabrikant *et al.* [FLM$^+$03] considered another network game. In this game each player chooses a subset of the other players to connect with. Once every player chose her subset we have a graph, and the cost of a player is α times the number of edges that she chose plus the sum of the lengths of the shortest paths from her to every other player. The social cost that has been considered is the sum of the costs of all players. Fabrikant *et al.* [FLM$^+$03] bounded the price of anarchy for various values of α. Their results have been recently improved by Albers *et al.* [AEED$^+$06]. Albers *et al.* showed constant upper bound on the price of anarchy for $\alpha \leq \sqrt{n}$ and for $\alpha \geq 12n \log n$. For other values of α the price of anarchy is $O(1 + (min\{\frac{\alpha^2}{n}, \frac{n^2}{\alpha}\})^{1/3})$.

Jackson [Jac03] considered several other network games in which a player's strategy is to choose a subset of the other players to connect with. Jackson studies pairwise stable states. A state S is *pairwise stable* if the following conditions hold: (i) for every edge $e = (i, j)$, $e \in S$, $C_S(i) \leq C_{S-\{e\}}(i)$ and $C_S(j) \leq C_{S-\{e\}}(j)$, and (ii) for every edge $e = (i, j)$, $e \notin S$, if $C_{S\cup\{e\}}(i) < C_S(i)$ then $C_{S\cup\{e\}}(j) > C_S(j)$. The first part of the definition requires that no player wishes to delete an edge that she uses, and from the second part of the definition follows, that if an edge is not in the network then at least one of the players suffers from adding it. Jackson studied the existence and efficiency (with respect to a social optimum) of a pairwise stable state for several different cost functions.

1.5 Our results

We consider the price of stability in undirected graphs with fair cost allocation, as it was defined by Anshelevich *et al.* [ADK$^+$04]. We prove that for undirected graphs with at least one player in every vertex and a distinguished source vertex r to which all players must connect, the price of stability of the network design game defined in Section 1.3.3 is $O(\log \log n)$ where n is the number of players.

For directed graphs even if there is a single source and we must have a player in every vertex, the bound $\Theta(\log n)$ on the price of stability is tight. To see that we slightly modify the example given by Anshelevich *et al.* [ADK$^+$04]. Consider the graph in Figure 1.2, with a player in every vertex that must connect to the source r. Player $u_i \in \{u_1, u_2, \ldots, u_{n-1}\}$ has a path of cost $\frac{1}{i}$ to the source. On the other hand all players can share a path of cost $2 + \epsilon$ to the source. The optimal solution connects all the players to the source through the common path of cost $2 + \epsilon$. However, in the unique Nash equilibrium, player $u_i \in \{u_1, u_2, \ldots, u_{n-1}\}$ uses her alternative path of cost $\frac{1}{i}$. Thus, the unique Nash equilibrium has a cost $H(n-1) + 2 + \epsilon = O(\log n)$.

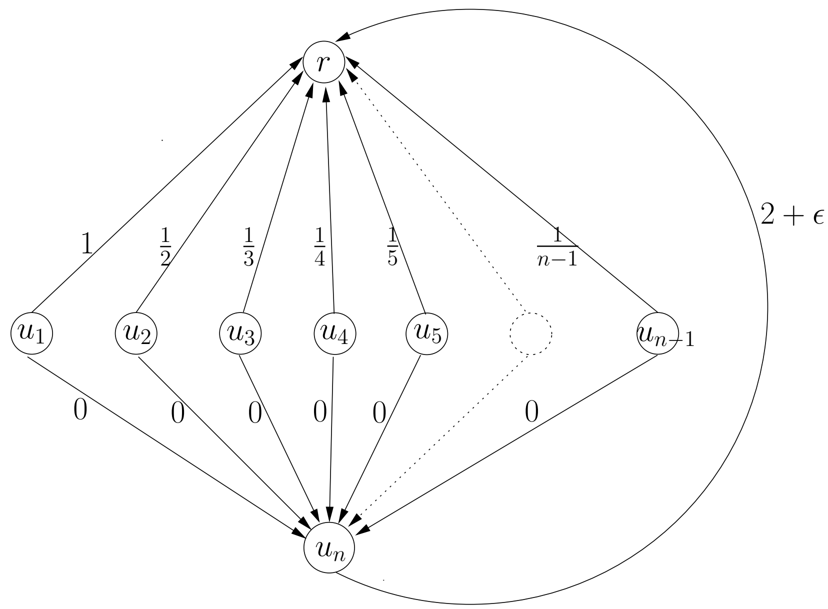


Figure 1.2: A network where the price of stability $O(\log(n))$. There is a player in every vertex that has to connect to r

The social optimum of this game is a minimum spanning tree. (Recall that the social cost function is the sum of the costs of all players.) We constructively prove the existence of an efficient Nash equilibrium by using the following techniques:

1. Rather than consider the original network, we define a new modified network, a multigraph (introducing some parallel edges). This modified network is an artifact of our proof technique. However, both these networks preserve two important properties: a social optimum in one network is mapped into a social optimum in the other with the same cost, and a Nash equilibrium in one is likewise easily mapped into a Nash equilibrium in the other with the same cost.

2. We initially assign to each player a strategy consistent with the social optimum. Of course, in general, this initial state may be unstable.

3. We now follow a careful schedule of "improvement moves". *I.e.*, we repeatedly change the

strategies of players subject to the constraint that no player is ever assigned a strategy inferior to her current strategy. The strategy chosen for a player is not necessarily a best response strategy, but must decrease the cost of the player.

4. When no further moves can be made, the set of current strategies used by the players is in a state of Nash equilibrium.

5. We argue that the social cost of this final Nash equilibrium is no more than a $O(\log \log n)$ factor times the minimum social cost.

Chapter 2

Preliminaries

Our input is an undirected graph $G = (V, E)$, along with a distinguished source vertex $r \in V$, and a cost function $c : E \mapsto R^+$. We will refer to $c(e)$, $e \in E$, as the *cost* of the edge e.

Associated with every vertex $v \in V$ is a selfish player. The network design game defines a strategy of a player v, to be a simple path in G connecting v to the source r. Let S_v denote the strategy chosen by player v, we define the *state* S to be the set of all paths S_v, for all players v. We define $E(S)$ to be the set of edges that appear in one or more of the paths in state S. [1]

It follows that the graph $(V, E(S))$ is a subgraph of G. In state S, let $x_s(e)$ be the number of players whose strategy contains edge $e \in E$. We define the cost of player v in state S, $C_S(v)$, to be $\sum_{e \in S_v} c(e)/x_s(e)$. A state S is in a Nash equilibrium if no player can lower her cost by unilaterally changing her path to the source r.

We shall use the standard potential function Φ, see e.g. [MS96, ADK+04] and 1.2, that maps every state S into a numeric value: $\Phi(S) = \sum_{e \in E} c_e H(x_s(e))$, where $H(n) = 1+1/2+1/3+\cdots+1/n$ is the n'th Harmonic number. If a single player v changes her strategy then the difference between

[1] Note that if one allow non simple paths as strategies then for every non simple strategy there is always a simple one which is strictly better.

the potential of the new state and the potential of the original state is exactly the change in the cost of player v. This implies that the improving response dynamics converges to a Nash equilibrium in pure strategies.

Notice that the sum of the costs of all players in state S is exactly the sum of the costs of the edges of $E(S)$. It follows that if the social cost function is the sum of the costs of all players then the social optimum of this game is a minimum spanning tree of the graph. We denote by OPT an arbitrary but fixed minimum spanning tree. Let p be the path from vertex u to vertex v in OPT. We define the *distance between u and v in OPT*, denoted by $d_{opt}(u, v)$, to be the sum of the costs of the edges between vertex u and vertex v along p.

We say that a player v makes an improvement move when the player chooses a new strategy S'_v such that $C_{S'}(v) < C_S(v)$ where S' is the new state in which v chooses S'_v and for every $w \neq v$, $S'_w = S_w$.

Let S be a state and let $e = (x, y) \in E(S)$. We say that e *appears in S in the direction $x \to y$* if there is some player u that uses a strategy S_u containing e, such that y is closer to the root than x on S_u. Similarly we say that e *appears in S in the direction $y \to x$* if there is some player u that uses a strategy S_v containing e for some player u, such that x is closer to the root than y on S_u.

We limit player v to choose strategies S'_v of the following three types. In the following definitions assume that v is the only player making the change, and as before we denote the new state by S' which is identical to S except that we replace S_v by S'_v.

EE (Existing Edges) – An improvement move such that $E(S') \subseteq E(S)$. Furthermore, if S'_v uses an edge $e = (x, y)$ in the direction $x \to y$ then $x \to y$ already appear in S.

OPT – An improvement move such that $E(S') \subseteq E(S) \cup OPT$, but $E(S') \nsubseteq E(S)$. Furthermore, if S'_v uses an edge $e = (x, y) \notin OPT$ in the direction $x \to y$ then $x \to y$ already appear in S.

$\overline{\textbf{OPT}}$ – The first edge $e = (v, w)$ on S'_v is not in $E(S) \cup OPT$, and $E(S') - \{e\} \subseteq E(S)$. Furthermore, if S'_v uses an edge $e' = (x, y)$, $e' \neq e$ in the direction $x \to y$ then $x \to y$ already appear in S.

Remark 1. *Note that if we start from OPT and restrict ourselves to using only EE,OPT, and $\overline{OPT}$ moves in every state S that we can reach no edge $(x, y) \notin OPT$ can appear in both directions, $x \to y$ and $y \to x$. It appears in the same direction determined by the $\overline{OPT}$ move that added (x, y).*

Chapter 3

Overview

In Section 4 we prove that if no player has an improvement move of type EE, OPT, or $\overline{\text{OPT}}$ then the state is a Nash equilibrium. The main result of this work, stated in Theorem 6.8, is the (constructive) proof that the price of stability for the network design game is $O(\log\log n)$. We prove this by carefully scheduling improvement moves until none are possible (Section 5.2). The resulting Nash equilibrium, N, has cost no more than $O(\log\log n)$ times the cost of the edges in OPT. In the rest of this chapter we give a high-level discussion of our proof while avoiding some subtle technicalities that we deal with later.

Clearly, edges in N, that are also in OPT, contribute, in total, no more than the cost of OPT. Now, let edge $e = (u, v) \in N$ be an arbitrary edge of cost $c(e)$. We want to justify the cost of any edge $e = (u, v)$ by mapping it to a subset of the edges of the social optimum OPT with the same cost. If we manage to find mapping without paying for several different edges in N with the same edges of OPT, this would give a constant ratio between the two costs.

Consider what happened when $e = (u, v)$ joined N. Improvement moves we allow introduce new edges only if the player that made the move, say player u, is adjacent to this edge. It is natural to try to amortize the cost of edge e with the neighborhood of u in OPT of radius $c(e)$. In order not to pay with the same edge in OPT for too many edges in $N \setminus OPT$ we make use of the following amortization techniques:

1. **Local amortization - "light edges"**: if the neighborhood of radius $c(e)$ about u within OPT contains relatively few players ($< \log n$), then we will argue that the "overcounting" we do in amortizing $c(e)$ is no more than by a factor of $\log \log n$. This local amortization is addressed in Lemma 6.7.

2. **Global amortization - "crowded edges"**: if u has many neighbor players in the ball of radius (approximately) $c(e)$, then we show that by carefully scheduling improvement moves, the potential function Φ(1.4) drops by a factor of $c(e) \log n$. As the potential function is always at least the cost of OPT and no more than $\log n$ times this cost, it follows that the total cost for such edges e is no more than the cost of OPT. This global amortization is addressed in Lemma 6.6.

Intuitively, the arguments for both local and global amortization make use of the following observations:

1. We schedule an $\overline{\text{OPT}}$ improvement move (*i.e. adding an edge not in the OPT tree*) only when no other improvement move is possible. Thus, just before we make an $\overline{\text{OPT}}$ improvement move, we know a great deal about the structure of the current state. In particular, we know that the difference in the cost between two players u and v is no more than $d_{opt}(u, v)$, see Lemma 6.1.

2. It also follows that when a player u does make an $\overline{\text{OPT}}$ move, say, adds edge (u, v), other players in her neighborhood (players w such that $d_{opt}(u, w)$ is proportional to $c(u, v)$) can reduce their cost. They reduce their cost by some constant fraction of $c(u, v)$, by choosing a path along OPT to u followed by the path from u to the source r. See Lemma 6.2.

Chapter 4

Improvement moves result in Nash equilibria

We now show that if no player has an improvement move of type EE, OPT, or $\overline{\text{OPT}}$ then the current set of strategies is a Nash equilibrium.

Lemma 4.1. *Let S be a state such that no player has an improving move of type EE. Then $(V, E(S))$ is a tree.*

Proof. Assume that $(V, E(S))$ is not a tree. Since our strategies are simple paths there must be some vertex w from which one can follow two paths to r; one path is the strategy S_w of w, and the other path, denoted by $\hat{S}_w$, is a suffix of some path S_u of a vertex u that goes through w. If $\sum_{e \in \hat{S}_w} c(e)/x_s(e) \leq \sum_{e \in S_w} c(e)/x_s(e)$ then w has an improving EE move in which she replaces her path by $\hat{S}_w$ which is a contradiction. On the other hand, if $\sum_{e \in S_w} c(e)/x_s(e) \leq \sum_{e \in \hat{S}_w} c(e)/x_s(e)$ then u has an improving EE move in which she replaces the suffix $\hat{S}_w$ of S_u by S_w. $\square$

Let S be the state which is a tree with root r (this would be the case when no EE move is possible). Let $P_S(v, w)$ be the path from vertex v to w in state S and let $LCA_S(v, w)$ be the lowest common ancestor of v and w in state S, when we root the tree at r. We remove the subscript S when it is clear from the context.

Let $P_w^v = P(w, LCA(v,w))$. Define $C_S^v(w) = \sum_{e \in P_w^v} \frac{c(e)}{x_S(e)+1} + \sum_{e \in S_w - P_w^v} \frac{c(e)}{x_S(e)}$, where S_w is the strategy of w in state S. In other words, we take into account an additional player on the path from w to $LCA(v,w)$ in S. One can think of $C_S^v(w)$ as the cost of w after v changes her strategy to a strategy in which she takes some path to w and then continues to the source according to S_w. It is clear that $C_S^v(w) \leq C_S(w)$ since the share of w in the cost of each edge on P_w^v in $C_S^v(w)$ is smaller than in $C_S(w)$.

Lemma 4.2. *Let S be a state in which no player can make an OPT, $\overline{OPT}$, or EE improvement move. Then S is in a Nash equilibrium.*

Proof. As there are no EE improving moves, by Lemma 4.1, state S is a tree. Assume that S is not in a Nash equilibrium. Then some player u has path S'_u such that the cost of u in the new state S', where only u changes her path to S'_u, is lower. Since S'_u is not an EE, OPT, or $\overline{\text{OPT}}$ move then S'_u must contain either (i) an edge $e' \in E(S)$ which is used in a direction that does not appear in S or (ii) an edge $e \notin E(S)$ which is not the first in S'_u.

If there is an edge e' in $S'_u \cap S$ which is used in S'_u in a direction which does not appear in S, then there must be an edge $e \notin E(S')$ in S'_u closer than e to the root (since S is a tree and S'_u is a simple path).

If u has more than one improving strategy then let S'_u be the one containing the minimum number of edges not in S. Let $e = (v, w)$ be the edge in $S'_u \setminus S$ closest to the source r (see Figure 4.1). By the preceding argument, all edges following e in S'_u appear in the same direction in S.

Consider the strategy S''_u obtained by taking the path that follows S'_u until the first vertex $\overline{v}$ that is an ancestor of v concatenated with $S_{\overline{v}}$. Notice that since $e = (v, w)$ is not in S''_u, then S''_u uses fewer edges which are not in S than S'_u. Since player u in strategy S'_u chooses to add edge e

followed by S_w and not to use strategy S''_u, we have

$$c(e) + C_S^u(w) < C_S^u(\overline{v}) \leq C_S^u(v), \tag{4.1}$$

as otherwise u has an improving strategy that has a smaller number of edges not in S. In state S no player can make OPT or $\overline{\text{OPT}}$ improvement moves, so for player v the strategy that consists of the edge $e = (v, w)$ followed by the path S_w is not an improving strategy, thus

$$C_S(v) \leq c(e) + C_S^v(w). \tag{4.2}$$

Using inequalities (4.1) and (4.2), and the fact that $C_S^u(v) \leq C_S(v)$ we get that $C_S^u(w) < C_S^v(w)$. Since $LCA_S(v, w)$ (that equal to $LCA_S(\overline{v}, w)$) and $LCA_S(u, w)$ are both on S_w, this is only possible if $LCA_S(u, w)$ is proper ancestor of $LCA_S(v, w)$. See Figure 4.1.

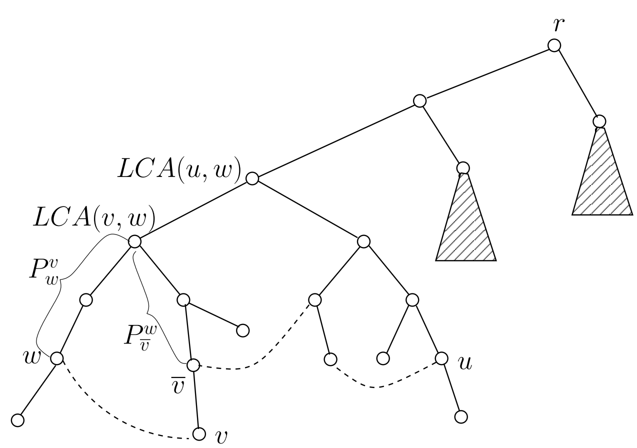


Figure 4.1: Solid edges are in S. Dashed edges are in S'. The state S' differ from S only by the strategy of u that includes all dashed edges.

Let S'' be the state identical to S except that player u uses S''_u rather than S_u. We now argue that $C_{S''}(u) \leq C_{S'}(u)$, in contradiction to the minimality of S'_u with respect to edges not in S'.

Since the contribution of each edge on the path from $LCA_S(v, w)$ to r to the costs $C_S(v)$ and

$C_S^v(w)$ is the same, we obtain that

$$c(e) + C_S^v(w) - C_S(v) = c(e) + \sum_{e' \in P_w^v} \frac{c(e')}{x_s(e')+1} - \sum_{e' \in P_v^w} \frac{c(e')}{x_s(e')}.$$

Since each edge on the path from u to $\overline{v}$ and on the path from $LCA_S(\overline{v}, w)$ to r contributes the same to $C'_S(u)$ and $C''_S(u)$ we obtain that

$$C_{S'}(u) - C_{S''}(u) \geq c(e) + \sum_{e' \in P_w^v} \frac{c(e')}{x_s(e')+1} - \sum_{e' \in P_{\overline{v}}^w} \frac{c(e')}{x_s(e')+1} \geq c(e) + C_S^v(w) - C_S(v).$$

The lemma now follows since $c(e) + C_S^v(w) - C_S(v) \geq 0$ by Inequality(4.2). $\square$

Chapter 5

Scheduling $\overline{\text{OPT}}$, OPT, and EE Improvement Moves

5.1 Modifying the underlying network

For technical reasons that we will elaborate on later, instead of considering the stability problem on the graph G, we switch to a related multigraph, $\overline{G}$. It would be clear from the definition of $\overline{G}$ that every minimum spanning tree in $\overline{G}$ corresponds to a minimum spanning tree in G with the same cost and vice versa. We also argue that a Nash equilibrium in the multigraph gives us a Nash equilibrium in the original graph with the same cost.

We define $\overline{G}$ as follows. Associate with every edge $e \in G$, not in OPT, an identical edge $e' \in \overline{G}$. Replace an edge $e \in G$ that is in OPT by parallel edges e^1 and e^2 in $\overline{G}$, each of weight $c(e)$. We say that e^1 and e^2 are *associated with* e and vice versa (See Figure 6.2).

Lemma 5.1. *For every Nash equilibrium in $\overline{G}$ there is a Nash equilibrium in G of the same cost.*

Proof. Let $\overline{N}$ be a Nash equilibrium in $\overline{G}$. Let $\overline{N}_u$ be the strategy of player u in $\overline{N}$. We convert $\overline{N}_u$ to the strategy N_u in G by replacing each edge e^1 or e^2 by the associated edge e in $G \cap OPT$, and each other edge by its identical copy in $G \setminus OPT$.

Let N be the state in G defined by the strategies N_u. The cost of N_u in N is the same as the cost of $\overline{N}_u$ for every player u. This follows since $\overline{N}$ is a tree and, therefore, contains only one of each pair of parallel edges e^1 and e^2 associated with an edge $e \in G \cap OPT$. So the number of players using each edge in N is equal to the number of players using the corresponding edge in $\overline{N}$.

If N is not a Nash equilibrium then some player u has an improving strategy N'_u. We map N'_u to a strategy $\overline{N}'_u$ of player u in $\overline{N}$ by replacing each edge $e \in OPT \cap N'_u$ by e^1 if $e^1 \in \overline{N}$, by e^2 if $e^2 \in \overline{N}$, and by either e^1 or e^2 if neither of them is in $\overline{N}$. One can verify that the cost of $\overline{N}'_u$ is the same as the cost of N'_u which contradicts the fact that $\overline{N}$ is a Nash equilibrium. $\square$

We define EE, OPT, or $\overline{\text{OPT}}$ moves in $\overline{G}$ the same as we defined them in Chapter 2 where by edges of OPT in $\overline{G}$ we refer to both copies of the associated edge in G.

5.2 Improvement scheduling algorithm

In this section we define our *Improvement scheduling algorithm* which we often refer to as the scheduler. We start the scheduler on G' from an initial state isomorphic to OPT. We define the initial state S to consist of all edges $e^1 \in G'$ associated with some $e \in OPT$. The scheduler halts and the process converges when no EE, OPT, or $\overline{\text{OPT}}$ moves are possible. The scheduler works in phases where in each phase we make a single $\overline{\text{OPT}}$ move.

Let S be some state, that includes strategy S_v for player v and S_w for player w. Given that w is a vertex on S_v, we define **Follow**(S, v, w) as a possible alternative strategy for vertex v. Strategy **Follow**(S, v, w) consists of the prefix of S_v up to and including vertex w, followed by S_w.

As an aid to the exposition, we use colors red and blue to label the parallel edges of G'. Initially, for every $e \in OPT$ we assign the edge e^1 the color red and the edge e^2 the color blue. In the beginning of a phase we may change the assignment of the red/blue colors to the parallel edges.

OptFollow(S, v, w) is a new strategy for player v that is defined if there is an edge (v, w) that is a copy of an edge in OPT colored blue. The strategy **OptFollow**(S, v, w) consists of the single edge (v, w) followed by S_w.

A phase of the Scheduling Algorithm: Let S be the state at the beginning of a phase. We maintain the invariant that in S no player can make an improving OPT or EE move, and thereby S is a tree according to Lemma 4.1. Before the phase starts we make a *Recoloring step*. In this step we recolor red each edge in S which is a copy of an edge in OPT, and we color blue the other copy of the edge which not in S (since S is a tree according to Lemma 4.1).

$\overline{\text{OPT}}$-move:

The phase starts with some player u changing her strategy by an improving $\overline{\text{OPT}}$ move. We denote by S' the state after this $\overline{\text{OPT}}$ move of u at the beginning of the phase.

OPT-loop:

Following this $\overline{\text{OPT}}$ move we start a breadth first search of OPT from u and for each player v in increasing order of $d_{opt}(u, v)$ we do the following.

Let $CurS$ be the state right before we process v, and let $p(v)$ be the parent of v in the breadth first search tree. We check if OptFollow$(CurS, v, p(v))$ is an improving strategy for v. If it is improving then v changes her strategy to OptFollow$(CurS, v, p(v))$. If it is not improving then we truncate the breadth first search at v. Note that all these OptFollow moves are defined since we started the phase with a recoloring step. We call this part of the phase of the scheduler the OPT-loop since all improvement moves made in this part are OPT moves. We denote by D the set of players consisting of u and players who performed an OPT move in the OPT-loop.

EE-loop:

For each player $w \in D$ let M_w be the subset of descendants of w in the tree S rooted at r, such that $v \in M_w$ if and only if $v \notin D$ and w is the first player in D along the path from v to r in S. In the second part of the phase we traverse the vertices in $\bigcup_{w \in D} M_w$. For each player $v \in M_w$, let $CurS$ be the state right after we process w, if the strategy Follow$(CurS, v, w)$ is an improving strategy for v, then v changes her strategy to Follow$(CurS, v, w)$. We call this part of the phase of the scheduler the EE-loop since all improvement moves made in this part are EE moves.

In the last part of the scheduler we perform any improving OPT or EE move until no such improving move exists. Then the phase ends, and we start the next one if there is an improving $\overline{\text{OPT}}$ move, or we stop if there isn't.

For an example of a phase of the scheduler see Figure 6.2.

The scheduler performs only $\overline{\text{OPT}}$, OPT, and EE moves. Since these moves are improvement moves, each such move causes a decrease of the potential function. Analogously to the argument in [ADK+04] (see also 1.2), a series of improvement moves in a finite potential game is finite. Therefore the scheduler halts. When the scheduler stops there are no $\overline{\text{OPT}}$, OPT or EE moves which are improving, so by Lemma 4.2 the resulting state is a Nash equilibrium.

Chapter 6

The price of stability

In this chapter we bound the cost of the Nash equilibrium reached by the scheduler.

Lemma 6.1. *Assume that no improving OPT moves, and no improving EE moves are possible in a state S. Then for every pair of players v and w the inequality $C_S(v) \leq C_S^v(w) + d_{opt}(v, w)$ holds.*

Proof. Suppose that $C_S(v) > C_S^v(w) + d_{opt}(v, w)$. Consider the strategy S'_v that consists of the path of OPT edges from v to w followed by the strategy of w. The strategy S'_v has cost $C_{S'}(v) \leq C_S^v(w) + d_{opt}(v, w)$, so it is an improving OPT move and we get a contradiction. □

Let S' be the state after player u performs an $\overline{\text{OPT}}$ move during the execution of the scheduler and let S be the state preceding this move. Let the cost of the newly used edge $e' = (u, v)$ be $c(e') = z$. In the following lemma we show that for every player w for which $d_{opt}(u, w) \leq \frac{z}{4}$, w would pay less if she takes the path in OPT to u and then continues as u in S'.

From Lemma 6.1 we know that when no OPT moves are possible the cost of u in S could not be much larger than the cost of w. The difference is about $d_{opt}(u, w) \leq \frac{z}{4}$. So if we make w go through u in S her cost may increase by at most $z/2$. It increases by at most $z/4$ for the path to get to u and by at most $z/4$ since the cost of u may be larger by at most $z/4$ from the cost of w.

In S' however w will split the cost of the edge (u, v) with u, paying only $z/2$ to go through it and thereby recovering the extra cost to get to u.

Lemma 6.2. *Let S be a state where no OPT moves and no EE moves which are improving are possible. Let S' be the new state after player u makes an improving $\overline{OPT}$ move defined by the edge $e' = (u, v)$. Let the cost of $c(e')$ be z. Then for every player w for which $d_{opt}(u, w) \leq \frac{z}{4}$, $C_{S'}(w) > C_{S'}(v) + \frac{z}{2} + d_{opt}(u, w)$.*

Proof. The strategy of player u in S' is the edge (u, v) followed by the strategy of player v, S_v, that is $C_{S'}(u) = C_{S'}(v) + z$. Since u performed an improving $\overline{\text{OPT}}$ move, $C_{S'}(u) < C_S(u)$, and thus

$$C_{S'}(v) + z < C_S(u). \tag{6.1}$$

Since in S there are no improving OPT moves and no improving EE moves, then, by Lemma 6.1,

$$C_S(u) \leq C_S^u(w) + d_{opt}(u, w). \tag{6.2}$$

We claim that $C_S^u(w) \leq C_{S'}(w)$. First note that the strategy S_w is equal to the strategy S'_w, since only the strategy of u is different in S and S'. The cost of w however may be different in S and S'. Split S_w into two pieces. One piece, denoted by P_1, from w to $LCA_S(u, w)$, and the other piece, denoted by P_2, from $LCA_S(u, w)$ to the source (see Figure 6.1). In S, player w shares with player u the cost of the edges in P_2, but this may not be true in S', so for $e \in P_2$, $x_s(e) \geq x_{s'}(e)$. Consider P_1. In S player w does not share with player u the cost of the edges on P_1, but she may share this cost with u in S'. So for $e \in P_1$ we have $x_s(e) + 1 \geq x_{s'}(e)$. In contrast $C_S^u(w)$ is the tentative cost of w assuming that she shares with u the cost for every edge of her strategy. Therefore,

$$C_S^u(w) = \sum_{e \in P_1} \frac{c(e)}{x_s(e) + 1} + \sum_{e \in P_2} \frac{c(e)}{x_s(e)} \leq \sum_{e \in S'_w} \frac{c(e)}{x_{s'}(e)} = C_{S'}(w), \tag{6.3}$$

as we claimed. From inequalities (6.2) and (6.3) we obtain

$$C_S(u) \leq C_{S'}(w) + d_{opt}(u, w). \tag{6.4}$$

Considering inequalities (6.1) and (6.4) we get $C_{S'}(w) + d_{opt}(u,w) > C_{S'}(v) + z$, and therefore

$$C_{S'}(w) > C_{S'}(v) + z - d_{opt}(u,w).$$

For player w for which $d_{opt}(u,w) \leq \frac{z}{4}$,

$$C_{S'}(w) > C_{S'}(v) + z - d_{opt}(u,w) \geq C_{S'}(v) + \frac{3z}{4} \geq C_{S'}(v) + \frac{z}{2} + d_{opt}(u,w).$$

$\square$

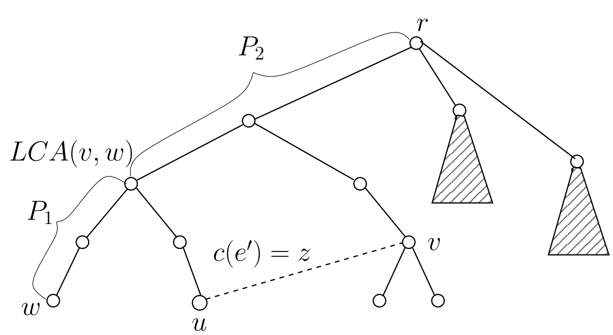
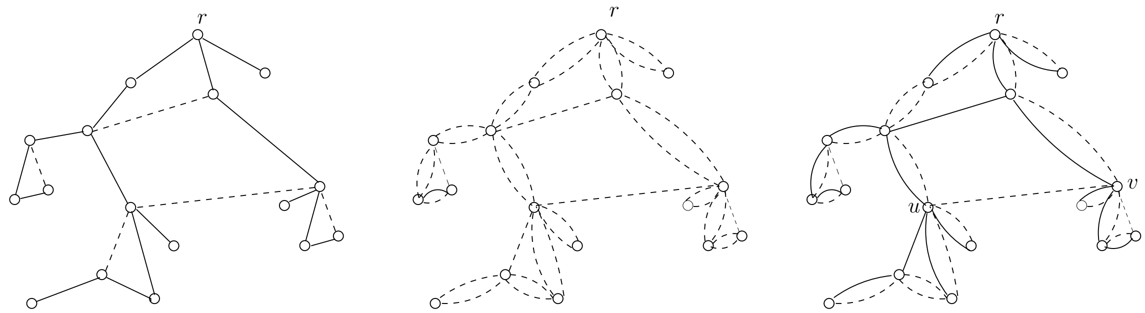


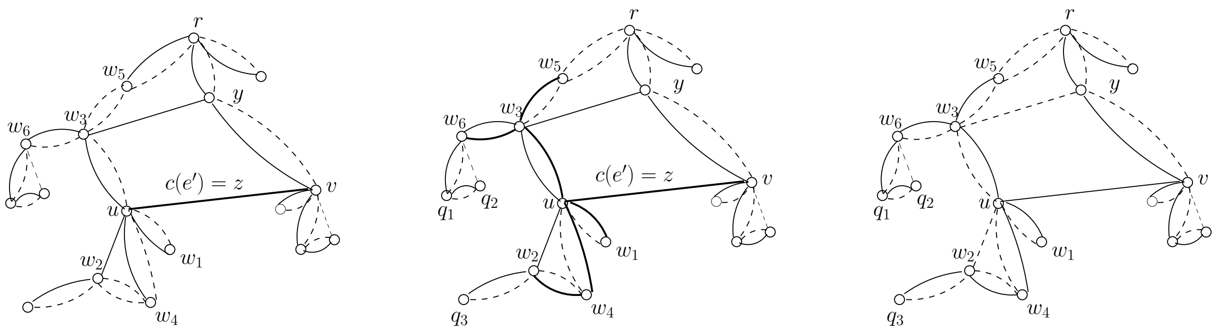
Figure 6.1: Player u makes an $\overline{\text{OPT}}$-move and buys edge $e' = (u,v)$ of cost z. The distance from w to u in OPT tree is $d_{opt}(u,w) \leq \frac{z}{4}$.

Remark 2. *Let u be the player making an improving $\overline{OPT}$ move using the edge $e' = (u,v)$. Let $z = c(e')$. For every $w \in S'_v$ (including v itself), $C_{S'}(w) \leq C_{S'}(v)$, and therefore by Lemma 6.2, $d_{opt}(u,w) > z/4$.*

Let S' be the state after player u performs an $\overline{\text{OPT}}$ move during the execution of the scheduler, defined by the edge $e_u = (u,v)$ whose cost is z. Let $w_0, w_1, w_2, \ldots, w_m$ be the vertices with $d_{opt}(u,w_i) \leq \frac{z}{4}$. Assume that $d(u,w_i) \leq d(u,w_{i+1})$. In particular $w_0 = u$, and the vertex w_1 is adjacent to u in OPT. Lemma 6.2 implies in particular that the strategy OptFollow(S, w_1, u) is improving for w_1. But what happens after w_1 changes her strategy? Can w_2 still make an OPT



(a) OPT in the graph G. (b) The graph $\overline{G}$. (c) A state S in the beginning of a phase.



(d) State S' after player u made an $\overline{\text{OPT}}$ move. (e) The state at the end of OPT-loop. (f) The state at the end of EE-loop.

Figure 6.2: In (a) solid edges are edges of OPT, and all other edges are dashed. In (c)-(f) solid edges are edges of the particular state shown, and all other edges are dashed. For $w \in \{w_1, w_2, \ldots, w_6\}$ we have $dist_{opt}(u, w) \leq \frac{z}{4}$, so they change their strategy to the strategy that uses OPT edges to connect to u followed by the strategy of u (See Lemma 6.3). Players q_1, q_2, q_3 change their strategy in the EE-loop, $M_{w_6} = \{q_1, q_2\}$, $M_{w_2} = \{q_3\}$, and $M_{w_1} = M_{w_3} = M_{w_4} = M_{w_5} = \phi$ (See Lemma 6.5).

move using some edge which is not in S and lower her cost? The following lemma shows that indeed this is the case.

Lemma 6.3. *Let w_k be the vertex following w_i on the path from w_i to u in OPT (that is, w_k is the parent of w_i in the BFS tree of the OPT-loop). Let S^i be the state just before the scheduler processes w_i in its OPT-loop. Then $C_{S^i}(w_i) > C_{S^i}(v) + \frac{z}{2} + d_{opt}(u, w_i)$, and therefore OptFollow(S^i, w_i, w_k) is an improvement move for w_i and the scheduler changes the state of w_i to this strategy.*

Proof. We prove by induction on i that in S^i,

$$C_{S^i}(w_\ell) > C_{S^i}(v) + \frac{z}{2} + d_{opt}(u, w_\ell) \tag{6.5}$$

for every $\ell \geq i$. From this the lemma clearly follows since the cost of OptFollow(S^i, w_i, w_k) is no larger than $C_{S^i}(v) + \frac{z}{2} + d_{opt}(u, w_i)$.

For $i = 1$ we have $S^1 = S'$ and Inequality (6.5) holds by Lemma 6.2. Assume that Inequality (6.5) holds for S^1, …, S^i, and for all relevant vertices in each of these states. We now show that after vertex w_i changes her strategy to OptFollow(S^i, w_i, w_k), where w_k is the vertex adjacent to w_i on the path from w_i to u in *OPT*, then Inequality (6.5) holds for S^{i+1} and for every $\ell \geq i + 1$.

Fix $\ell \geq i + 1$. We show that

$$C_{S^{i+1}}(w_\ell) - C_{S^i}(w_\ell) \geq C_{S^{i+1}}(v) - C_{S^i}(v) \tag{6.6}$$

from which the induction step follows.

Recall that the states S^i and S^{i+1} differ only in the strategy of player w_i. For each $\ell \geq i + 1$ the strategy of player w_ℓ in S^i and her strategy in S^{i+1} are still the same as her strategy was in S (and S'), and we denote it by S_{w_ℓ}. Similarly, by Remark 2, the strategy S_v of v in S does not change while processing w_1, …, w_m.

We establish Inequality (6.6) by showing that: (i) each edge that contributes a negative amount to the left side of (6.6) contributes the same negative amout to the right side of (6.6), (ii) no edge contributes positive amount to the right side of (6.6).

To prove (i), consider an edge $e \in S_{w_\ell}$ such that the number of players using e in S^{i+1} is larger (by 1) than the number of players using e in S^i since player w_i uses e after she makes the change. It follows that $e \in S_v$ since the only edges in the new strategy of w_i in S^{i+1} that were in S (and therefore could be in S_{w_ℓ}) are those edges in S_v. This holds since by the induction in S^{i+1} the strategy of each player w_j, where $j \leq i$, is a path consisting of blue copies of OPT edges that are not in S, up to vertex v, and continuing from there as the path S_v.

To prove (ii) consider an edge $e \in S_v$. Player v pays in S^{i+1} at most the share that she pays in S^i.

Consider the phase of the scheduler illustrated in Figure 6.2. The state $S' = S^1$ is the state after u changes her strategy to $(u \to v \to y \to r)$ shown in 6.2(d). At the first step of the OPT-loop w_1 changes her strategy from $(w_1 \to u \to w_3 \to y \to r)$ to $(w_1 \to u \to v \to y \to r)$, and we reach S^2. Now consider the cost $C_{S^2}(w_2)$ compared to the cost $C_{S^1}(w_2)$. The edge (v, y) contributes more to $C_{S^1}(w_2)$ than to $C_{S^2}(w_2)$ because w_1 uses this edge in S^2 but not in S^1. However, since $(v, y) \in S_v$, $C_{S^1}(v)$ is larger than $C_{S^2}(v)$ by the same amount. □

Remark 3. *To make Lemma 6.6 work we had to introduce $\overline{G}$. With one set of OPT edges it is possible that when w_i changes her strategy she uses OPT edges that can be part of the strategy of w_ℓ for some $\ell > i$ that are not in S_v, and are not on the path between w_ℓ and u in OPT. This may make the strategy of w_ℓ cheaper when she considers making her change.*

The following lemma gives a lower bound on the decrease in the potential during a phase of the scheduler.

Lemma 6.4. *Let u be the player making the $\overline{OPT}$ move at the beginning of a phase. Let $e' = (u, v)$*

be the first edge in the new strategy of player u, and let $z = c(e')$. Let m be the number of players at distance at most $\frac{z}{4}$ from player u in OPT (other than u itself). If $m \geq 2$ then the potential of the state at the end of the phase is smaller by $\Omega(zm)$ from the potential of the state at the beginning of the phase.

Proof. Let $w_1, \ldots, w_m$ be the players such that $d_{opt}(u, w_i) \leq \frac{z}{4}$. Assume that $d_{opt}(u, w_i) \leq d_{opt}(u, w_{i+1})$. Let S^i be the state right before the scheduler processes w_i in its OPT-loop.

By Lemma 6.3, when the scheduler processes player w_i we have that $C_{S^i}(w_i) > C_{S^i}(v) + \frac{z}{2} + d_{opt}(u, w_i)$. Also according to Lemma 6.3 players $w_1, \ldots, w_{i-1}$ already use the edge (u, v) in their strategy in S^i. Therefore the cost of the new strategy OptFollow(S^i, w_i, w_k) for w_i is at most $C_{S^i}(v) + \frac{z}{i+1} + d_{opt}(u, w_i)$. (Here w_k is the vertex adjacent to w_i on the path in OPT from w_i to u.) It follows that player w_i decreases her cost by at least $\frac{z}{2} - \frac{z}{i+1}$. Summing up the decrease in the cost of all m players $w_1, \ldots, w_m$, we get $\sum_{i=1}^{m} \frac{z}{2} - \frac{z}{i+1} = z(\frac{m}{2} - (H(m+1) - 1)) = \Theta(zm)$. This is also the decrease in the potential since when a single player changes her strategy the change in the potential is equal to the change in the cost of the player. $\square$

As before, let S' be the state after player u performs an $\overline{\text{OPT}}$ move and uses an edge $e' = (u, v) \notin OPT$. Let D be the set of vertices accumulated while the scheduler performed the OPT-loop, together with u, and let S'' be the state after the execution of the EE-loop. Consider an edge $e \notin OPT$ which was the first edge in the strategy S_w in state S, of some player w who is in D. By the definition of the scheduler, in S'', the first edge in the strategy of w, would be an edge in OPT (or e' for u) and not e. However it could be that some descendant of w still uses e in her strategy. We want to show that this could not be the case. That is, while performing the EE-loop all these descendants take an alternative strategy that does not use e.

For example consider the phase illustrated in Figure 6.2(e), and 6.2(f). After the OPT-loop w_3 does not use edge (w_3, y), but q_1 still uses this edge. We show that during the EE-loop q_1 would replace her strategy, and at the end of the phase she does not use (w_3, y).

Lemma 6.5. *Consider a phase of the scheduler. Let S be the starting state, and let D be the set of players that includes player u and the players that change their strategy in the OPT-loop. Let $e \notin OPT$ be the first edge in a strategy S_w, for some $w \in D$. Let S'' be the state after the execution of the EE-loop. Then $e \notin S''$.*

Proof. The only players that can use e in S'' are players that used e in S. Since S is a tree, each of these players is a descendant of player w in S. Descendants of w which are in D do not use e in their strategy in S''. So in the rest of the proof we consider only descendants of w not in D.

Let $x \notin D$ be a descendant of w. If x replaces her strategy in the EE-loop then x does not include e in her new strategy. This is because her new strategy contains the same path as in S to get to a vertex in D, then a path in OPT to get to u, continued with edge $e' = (u, v)$ and then the path S_v which is in S. None of these subpaths contains e.

We have to show that every descendant of w in S replaces her strategy in the EE-loop. For every vertex $w \in D$ let M_w be, as in Section 5.2, the subset of descendants of w such that $m \in M_w$ if and only if $m \notin D$ and w is the first player in D along the path S_m, from m to r in state S. Every descendant of w in S is either in M_w or in M_y for some descendant y of w in S. So every descendant of w could have changed her strategy to follow a vertex in D when the scheduler runs the EE-loop. See figure 6.2.

Let vertex x be a descendant of w in S, such that $x \in M_y$. Notice, that player x does not change her strategy from the beginning of the phase until we process her in the EE-loop. Let S^1 be the state right after y changes her strategy in the OPT-loop and let S^2 be the state in which we process x in EE-loop. Let F be the strategy Follow(S^1, x, y). Strategy F is an improving strategy for x in state S^1. We prove by induction that F remains an improving strategy of x in every state following S^1 and preceding S^2 (including S^2).

For any state T we denote by $F(T)$ the state obtained from T by changing the strategy of x to T. Assume that $C_T(x) > C_{F(T)}(x)$ holds for some state T following S^1, but preceding S^2. Let q be the next player that changes her strategy, and let T' be a state after it. We need to prove that $C_{T'}(x) > C_{F(T')}(x)$. Writing the inequality $C_T(x) > C_{F(T)}(x)$ explicitly we obtain

$$\sum_{e \in T_x} \frac{c(e)}{x_T(e)} > \sum_{e \in F} \frac{c(e)}{x_{F(T)}(e)}. \tag{6.7}$$

Similarly we can write the inequality $C_{T'}(x) > C_{F(T')}(x)$ explicitly as follows (recall that $T_x = T'_x$)

$$\sum_{e \in T_x} \frac{c(e)}{x_{T'}(e)} > \sum_{e \in F} \frac{c(e)}{x_{F(T')}(e)}. \tag{6.8}$$

For every edge e, that q stops using it, we have that $x_{F(T')}(e) = x_{F(T)}(e) - 1$, and $x_{T'}(e) = x_T(e) - 1$. Similarly for every edge e that q starts using, we have that $x_{F(T')}(e) = x_{F(T)}(e) + 1$, and $x_{T'}(e) = x_T(e) + 1$. For other edges $x_{F(T')}(e) = x_{F(T)}(e)$, and $x_{T'}(e) = x_T(e)$. We show that (i) every $e \in F$ such that $x_{F(T')}(e) = x_{F(T)}(e) - 1$ belongs to T_x, (ii) every $e \in T_x$ such that $x_{T'}(e) = x_T(e) + 1$ belongs to F. This implies that Inequality 6.8 holds assuming that Inequality 6.7 holds before the change of q.

To prove (i) consider an edge $e \in F$ such that $x_{F(T')}(e) = x_{F(T)}(e) - 1$. Then e must be in the prefix of F from x to y and therefore in T_x.

To prove (ii) consider an edge $e \in T_x$ such that $x_{T'}(e) = x_T(e) + 1$. Edge e must be on S_v. Since $S_v \subset F$, $e \in F$ as required. $\square$

Let N be the Nash equilibrium reached by the scheduler. We would like to relate the cost of N to the cost of OPT. So we partition the edges in N into two classes: those that are in OPT and those that are not in OPT. Clearly the total cost of the edges in $N \cap OPT$ is no larger than the cost of OPT. So our real concern are those edges in $N \setminus OPT$. Each such edge got into N by an $\overline{\text{OPT}}$ move performed by the scheduler at the beginning of some phase and remained there until the end of the process. We associate each such edge (u, v) with player u that actually improved her strategy by the $\overline{\text{OPT}}$ move that added the edge (u, v) to N. We further partition the edges $e = (u, v)$ in $N \setminus OPT$ according to the number of vertices in OPT in a neighborhood of size $c(e)/4$ around the associated player. Specifically, let $e = (u, v) \in N \setminus OPT$ be associated with player u. We say that e is *crowded* if $|\{w \mid d_{opt}(u, w) \leq \frac{c(e)}{4}\}| \geq \log n$, and we say that e is *light* otherwise.

Lemma 6.6. *The total cost of all crowded edges is $O(OPT)$.*

Proof. Let e be a crowded edge in $N \setminus OPT$. By Lemma 6.4, in the phase that started with the $\overline{\text{OPT}}$ move that put e into N, the potential dropped by $\Omega(c(e) \log n)$. Since initially the potential is at most $OPT \cdot \log n$, and is always decreasing, the lemma follows. □

Lemma 6.7 bounds the total cost of light edges.

Lemma 6.7. *The total cost of all light edges in N is $O(OPT \cdot \log \log n)$.*

Proof. Let U be the set of players assigned to light edges. For a player $v \in U$ we denote the associated light edge by e_v. We define *the cost of* v to be the cost of e_v and denote it by z_v.

First we show that for $v \in U$, $z_v \leq d_{opt}(v, r)$. Let S be the state right before edge e_v was added and let S' be the state after e_v was added. The scheduler only allows new $\overline{\text{OPT}}$ edges at the start of a phase, thus, S is a state at the beginning of a phase. According to Lemma 6.1, $C_S(v) \leq C_S(r) + d_{opt}(v, r) = d_{opt}(v, r)$. Since v made an improving move by changing her strategy S_v to strategy S'_v such that $e_v \notin S$ and $e_v \in S'$, then $z_v \leq C_{S'}(v) < C_S(v)$, and so $z_v \leq d_{opt}(v, r)$.

We choose the subset $F \subseteq U$ as follows. Start with $T = U$ and $F = \phi$. Let $v \in T$ be a player of maximum cost in T. Let $U_v = \{w \in U \mid d_{opt}(v, w) \leq z_v/4, z_w \leq z_v/\log n\}$. Add v to F and continue with $T = T \setminus (\{v\} \cup U_v)$ until T is empty.

Since every vertex $v \in F$ is a light vertex, the total cost of all vertices in U_v is at most z_v, so its enough to prove that the total cost of all vertices in F is $O(OPT \cdot \log \log n)$.

For $v \in F$, consider a ball, B_v, of radius $z_v/12$ around v in OPT. Since $z_v \leq d_{opt}(v,r)$, the ball B_v contains at least one path of length at least $z_v/12$.

We prove that every point $\xi \in OPT$ is contained in at most $\log\log n$ balls B_v for $v \in F$. Therefore the total cost of all vertices in F is $O(OPT \cdot \log\log n)$.

Let $e \in OPT$ and let ξ be some point on edge e. Let A_ξ be the set of vertices whose balls contain ξ. We show that $|A_\xi| \leq \log\log n$. Let $v_1, v_2, \ldots, v_m$ be the vertices of A_ξ in the order that their light edges $e_{v_1}, e_{v_2}, \ldots, e_{v_m}$ were added to N (if some edge was added more than once, we consider the last time it was added). Let $1 \leq i < j \leq m$. By Remark 1 when v_j makes the $\overline{\mathrm{OPT}}$ move that adds e_{v_j}, v_i was using e_{v_i} in her strategy. Since $e_{v_i} \in N$, that is v_i did not change her strategy in the OPT-loop of the phase where v_j added e_{v_j}, according to Lemma 6.5, we have

$$d_{opt}(v_i, v_j) > \frac{z_{v_j}}{4}. \tag{6.9}$$

Since $d_{opt}(v_i, \xi) \leq z_{v_i}/12$ and $d_{opt}(v_j, \xi) \leq z_{v_j}/12$, we obtain

$$d_{opt}(v_i, v_j) \leq \frac{z_{v_i}}{12} + \frac{z_{v_j}}{12}. \tag{6.10}$$

Substituting $j = i+1$ and combining the Inequalities (6.9) and (6.10), we get $z_{v_{i+1}} < z_{v_i}/2$ and, by induction, $z_{v_{i+1}} < \frac{z_{v_1}}{2^i}$. In particular, for every i we have $z_{v_{i+1}} < z_{v_1}$, from which follows that $d_{opt}(v_{i+1}, v_1) \leq z_{v_1}/6$. Therefore, by the definition of F, it must be that $z_{v_{i+1}} > z_{v_1}/\log n$. Since

$$\frac{z_{v_1}}{\log n} < z_{v_{i+1}} \leq \frac{z_{v_1}}{2^i}$$

we get that $i \leq \log\log n$, and therefore $|A_\xi| \leq \log\log n$.

□

The following theorem follows from Lemmas 6.6 and 6.7 and is the main result of this work.

Theorem 6.8. *For a graph with a source vertex and a player in every vertex the price of stability is* $O(\log\log n)$.

Chapter 7

A lower bound on the price of stability

We give a family of undirected networks each with a player in every vertex that wishes to connect to the source r, in which the price of stability converges to $\frac{12}{7} > 1.7$.

Our network is shown in Figure 7. There is an edge of cost $\frac{4}{3} + \epsilon$ between vertices u_i and u_{i+1} if i is even and an edge of cost $1 + \epsilon$ between vertices u_i and u_{i+1} if i is odd. The optimal solution consists of one edge of cost 2 and all the edges of cost $\frac{4}{3} + \epsilon$ and $1 + \epsilon$ giving a total cost of $(1 + \epsilon + \frac{4}{3} + \epsilon)\frac{n-1}{2} + 2 \approx \frac{7}{6}n$.

We show that there is unique Nash equilibrium in this game.

Lemma 7.1. *For a graph as described above, there is unique Nash equilibrium, in which all players connect directly to r.*

Proof. As we discussed in Lemma 4.1, every Nash equilibrium for this game is a tree. Let T be a Nash equilibrium. We show that all players connect directly to r in T.

Let v be a vertex that connects directly to r in T such that the depth of its subtree in T is maximized. From the particular structure of G follows that the subtree of v consists of at most two paths P_1 and P_2 such that v is the only common vertex of P_1 and P_2. In case P_1 and P_2 are of different lengths, let u be the leaf at the end of the longer path among them. If both P_1 and P_2 are of the same length, then either the leaf at the end of P_1 or the leaf at the end of P_2 is adjacent to an edge of cost $4/3 + \epsilon$. Let u be that leaf.

If the length of the path from u to r is at least 4 and the first edge of the path is of cost $1 + \epsilon$, than u pays $(1 + \epsilon) + \frac{4/3+\epsilon}{2} + \frac{1+\epsilon}{3} > 2$ for the first 3 edges of the path, u can reduce her cost by connecting directly to r. Similarly, if the length of the path from u to r is at least 4 and the first edge of the path is of cost $4/3 + \epsilon$, than u pays at least $(4/3 + \epsilon) + \frac{1+\epsilon}{2} + \frac{4/3+\epsilon}{3} > 2$, and therefore u can reduce her cost by connecting directly to r.

If the length of the path from u to r is 3, than u pays at least $(1 + \epsilon) + \frac{4/3+\epsilon}{2}$ to connect to v and at least $\frac{2}{5}$ for using edge (v, r). Since $(1 + \epsilon) + \frac{4/3+\epsilon}{2} + \frac{2}{5} > 2$, and therefore u can reduce her cost by connecting directly to r.

If the length of the path from u to r is 2 and if v has two children (that is u has a sibling), then u pays $4/3 + \epsilon + 2/3 > 2$. If the length of the path from u to r is 2 and if v has only one child u, then u pays $(1 + \epsilon) + 1 > 2$. So, in both cases we get a contradiction that T is a Nash equilibrium.

$\square$

It follows from Lemma 7.1, that the unique Nash equilibrium of this game has a cost $2n$, and therefore the price of stability is $\frac{2n}{(7/6)n} \approx \frac{12}{7} > 1.7$.

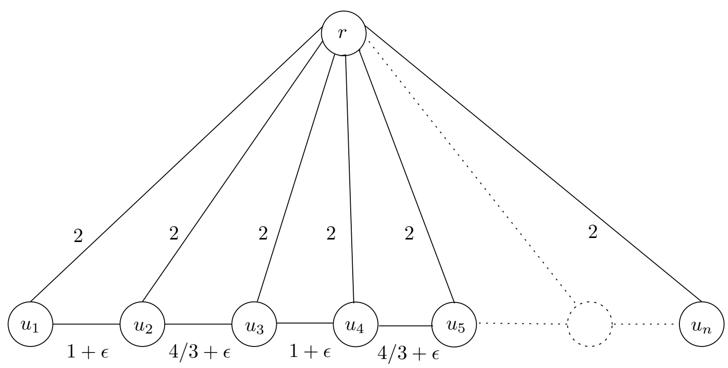


Figure 7.1: Example of graph in which the price of stability > 1.7.

Bibliography

[AAE05] Baruch Awerbuch, Yossi Azar, and Amir Epstein. Large the price of routing unsplittable flow. In *STOC '05: Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 57–66, 2005.

[ADK+04] Elliot Anshelevich, Anirban Dasgupta, Jon Kleinberg, Eva Tardos, Tom Wexler, and Tim Roughgarden. The price of stability for network design with fair cost allocation. In *FOCS '04: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 295–304, 2004.

[ADTW03] Elliot Anshelevich, Anirban Dasgupta, Eva Tardos, and Tom Wexler. Near-optimal network design with selfish agents. In *STOC '03: Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 511–520, 2003.

[AEED+06] S. Albers, S. Eilts, E. Even-Dar, Y. Mansour, and L. Roditty. On nash equilibria for a network creation game. In *SODA 06: Seventeenth ACM-SIAM Symposium on Discrete Algorithms (SODA), To appear.*, 2006.

[CK05a] George Christodoulou and Elias Koutsoupias. On the price of anarchy and stability of correlated equilibria of linear congestion games. In *ESA 2005: 13th Annual European Symposium, Proceedings*, volume 3669, 2005.

[CK05b] George Christodoulou and Elias Koutsoupias. The price of anarchy of finite congestion games. In *STOC '05: Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 67–73, 2005.

[CSM04a] J.R. Correa, A. Schulz, and N. Stier Moses. Computational complexity, fairness, and the price of anarchy of the maximum latency problem. In *Integer Programming and Combinatorial Optimization: 10th International IPCO Conference*, volume 3064 of *LNCS*, pages 59–73, 2004.

[CSM04b] J.R. Correa, A. Schulz, and N. Stier Moses. Selfish routing in capacitated networks. *Mathematics of Operations Research*, 29(4):961–976, 2004.

[CV02] Artur Czumaj and Berthold Vöcking. Tight bounds for worst-case equilibria. In *SODA '02: Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 413–420, 2002.

[FKS04] Dimitris Fotakis, Spyros Kontogiannis, and Paul Spirakis. Selfish unsplittable flows. 2004.

[FLM+03] Alex Fabrikant, Ankur Luthra, Elitza Maneva, Christos H. Papadimitriou, and Scott Shenker. On a network creation game. In *PODC '03: Proceedings of the twenty-second annual symposium on Principles of distributed computing*, pages 347–351, 2003.

[Jac03] M. Jackson. A survey of models of network formation: Stability and efficiency, 2003.

[KMS02] E. Koutsoupias, M. Mavronikolas, and P. Spirakis. Approximate equilibria and ball fusion. In *Proceedings of the 9th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, pages 223–235, 2002.

[KP99] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *STACS '99: Proceedings of the 16th Int. Symposium on Theoretical Aspects of Computer Science*, pages 404–413, 1999.

[LO01] Lavy Libman and Ariel Orda. Atomic resource sharing in noncooperative networks. pages 385–409, 2001.

[MS96] D. Monderer and L. Shapley. Potential games. *Games and Economic Behavior*, 14:124–143, 1996.

[Nas50] John F. Nash. Equilibrium points in n-person games. *Proc. of the National Academy of Sciences*, 36:48–49, 1950.

[Pap01] Christos Papadimitriou. Algorithms, games, and the internet. In *STOC '01: Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 749–753. ACM Press, 2001.

[Ros72] R. Rosenthal. A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, 2:65–67, 1972.

[Rou02] Tim Roughgarden. The price of anarchy is independent of the network topology. In *STOC '02: Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 428–437, 2002.

[Rou04] Tim Roughgarden. The maximum latency of selfish routing. In *SODA '04: Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 980–981, 2004.

[RT02] Tim Roughgarden and Eva Tardos. How bad is selfish routing? *J. ACM*, 49(2):236–259, 2002.