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

Amos Fiat, Haim Kaplan, Meital Levy, Svetlana Olonetsky, and Ronen Shabo

School of Computer Science, Tel-Aviv University,Israel

Abstract. In this paper we address the open problem of bounding the price of stability for network design with fair cost allocation for undirected graphs posed in [1]. For the version of this problem that we consider, every vertex is associated with a selfish agent, and there is a distinguished source node to which all agents must connect. We show that the price of stability is $O(\log \log n)$. We prove this by defining a particular improving dynamics in a related graph. This proof technique may have other applications and is of independent interest.

1 Introduction

The Internet is built, maintained, and used by noncooperative selfish agents. The Nash equilibria is a stable state in a noncooperative game in the sense that no agent can gain from unilaterally changing its behavior. Koutsoupias and Papadimitriou [13] suggested the study of the so-called *price of anarchy*. This is the ratio between the cost of the worst-case Nash equilibria and the cost of the social optimum.

An alternative notion to the price of anarchy is the *price of stability* [1][1]. This is the ratio between the cost of the least expensive Nash equilibria and the cost of the social optimum. 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.

We consider the game of network design with fair cost allocation introduced in [1]. In this game, agent i has to choose a path (strategy) from source node s_i to destination node t_i. The cost of an edge e, $c(e)$, is shared equally by all agents i whose chosen path $p_i = s_i, ..., t_i$ includes e.

It follows from the potential function arguments of [16, 15] that pure strategy Nash equilibria always exist for general congestion games, and in particular for the network design game that we consider here (both directed and undirected versions)[2]. In the following, we consider the price of stability for this network design game with respect to pure strategies.

The social optimum for this game is a minimum Steiner network connecting all source-destination pairs. Anshelevich *et al.* [1] show that the price of stability of this game is at most $H(n) = 1 + 1/2 + \cdots + 1/n$, where n is the number of agents. They also exhibit a directed network where this bound is tight.

For undirected graphs the upper bound of $H(n)$ on the price of stability still holds but the lower bound does not. Furthermore, for the case of two players and an undirected graph with a single source Anshelevich *et al.* [1] prove a tight bound on the price of stability of 4/3 which is less than $H(2) = 3/2$. Thus, [1] left open the question of whether there is a tighter bound for undirected graphs.

[1] The best Nash equilbria or "optimistic price of anarchy" was also considered in [2, 8, 5].

[2] Some weighted congestion games do not have Nash equilibria in pure strategies.

Our results We prove that for undirected graphs with an agent in every vertex and a distinguished source vertex r to which all agents must connect, the price of stability of the network design game of [1] is $O(\log \log n)$ where n is the number of agents. In contrast, in directed graphs even when there is a single source and an agent in every vertex the price of stability is still $\Theta(\log n)$. This follows by a slight modification of the lower bound example of [1].

Related work on Network games Much of the work on network games has focused on congestion games [16, 15]. In particular, latency minimization and some network construction/design games can be modelled as congestion games or weighted congestion games.

Latency issues in the simplest network models (parallel links) can be modelled as games for makespan minimization for parallel machine scheduling/load balancing [13, 14, 10, 12, 4, 9, 11, 19]. Minimizing the total latency in general networks when there are many players, each controlling a negligible amount of flow, was studied by Roughgarden and Tardos [18]. This is called the *non-atomic* model. Roughgarden and Tardos proved that for linear latency functions the price of anarchy is exactly 4/3. The social cost of the maximum latency (rather than total latency) has also been considered in [7, 17].

Recently, Awerbuch, Azar, and Epstein [3], and Christodoulou and Koutsoupias [6] considered the *atomic* version of game in general networks where each agent controls a fixed amount of flow which is referred to the "demand" of the agent. For linear latency functions and equal demands the price of anarchy is 2.5 and for general demands it is 2.618, both for pure and mixed strategies. They also generalize the results for polynomial latency functions [6, 3], and considered the maximum latency rather than the sum [6]. Recently, Christodoulou and Koutsoupias [5], also gave an upper bound of 1.6 on the price of stability for linear latency functions.

The game we consider here is also a congestion game where players are source-destination pairs and a strategy of a player is a single path from the source to the destination. The difference is that the cost a player pays for each edge e on its path is $c(e)/x_e$ where x_e is the number of players using the edge. The price of anarchy for this game can be high as shown in [1]. But we are interested in the price of stability.

In [2] Anshelevitz *et al.* considered a connection game under a different cost model. In this model every agent must connect some set of terminals. Agents can offer to pay any fraction of the cost of an edge. If the total contribution of agents towards the purchase price of an edge is sufficient (reaches the price of the edge) then the edge is purchased. (In particular agents can offer nothing for an edge). A solution is an equilibrium if no agent can benefit from changing her strategy. This model can result in instances where only mixed strategy Nash equilibria exist. Arguably, mixed strategies are not applicable to this network construction application. While determining if a pure strategy Nash equilibria exists is *NP*-hard, for the single source case the price of stability is 1 [2].

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. [3]

[3] 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.

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. [15, 1], 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 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 em 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 $e = (x, y) \in S_u$. We say that u *uses e in the direction $x \to y$* if y is closer than x to the r on S_u. Similarly, we say that *u uses e in the direction $y \to x$* if x is closer than y to the r on S_u. We say that *e appears in S in the direction $x \to y$* (or simply $x \to y$ appears in S) if there is a player u such that e appears in on S_u in the direction $x \to y$.

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 $x \to y$ appears in S.

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

Remark 1. Note that if we start from OPT and perform only EE, OPT, and $\overline{\text{OPT}}$ moves then in the state that we reach, no edge $(x, y) \notin OPT$ appears in both directions, $x \to y$ and $y \to x$. It appears in the same direction determined by the $\overline{\text{OPT}}$ move that added (x, y).

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. We single out a specific Nash equilibrium, denoted by N, that we reach by carefully scheduling EE, OPT, and $\overline{\text{OPT}}$ moves. We then prove that the cost of N is larger than the cost of OPT by a factor of at most $O(\log \log n)$.

This scheduling has two effects which our proof exploits. Let $c(u, v) = z$. After an $\overline{\text{OPT}}$ move of a player u that adds an edge (u, v) into the current state, we make further OPT and EE moves so that more players use (u, v).

We traverse players in increasing distance from u in OPT. Each player that improves her strategy by using the path to u in OPT following by the strategy of u makes the corresponding improvement move.

1. If there are $O(\log n)$ players whose distance to u in OPT is no larger than $z/4$ then the potential decreases by $O(z \log n)$. Therefore, the total cost introduced into N by such edges is $O(OPT)$.
2. Edges in $N \setminus OPT$ cannot be too close to each other in the metric defined by OPT. This allows us to relate the cost of all other edges in $N \setminus OPT$ to the cost of OPT.

Due to the space limit some of the proofs are omitted.

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 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.

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 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 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 $e \notin E(S) \cup OPT$ 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 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. By our choice of S'_u the strategy S''_u can not be improving for u and therefore

$$c(e) + C^u_S(w) < C^u_S(\overline{v}) \leq C^u_S(v), \tag{1}$$

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

$$C_S(v) \leq c(e) + C^v_S(w). \tag{2}$$

Using inequalities (1) and (2), and the fact that $C^u_S(v) \leq C_S(v)$ we get that $C^u_S(w) < C^v_S(w)$. Since $LCA_S(v, w)$ (that must be 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 1.

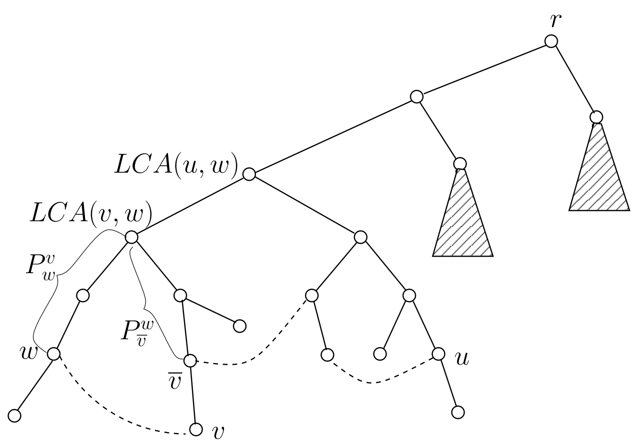


Fig. 1. Solid edges are in S. Dashed edges are in S'. The state S' differ from S only by the strategy of u that includes all dashed edges.

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 $C^v_S(w)$ is the same, we obtain that

$$c(e) + C^v_S(w) - C_S(v) = c(e) + \sum_{e' \in P^v_w} \frac{c(e')}{x_s(e') + 1} - \sum_{e' \in P^w_{\overline{v}}} \frac{c(e')}{x_s(e')}.$$

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

$$C_{S'}(u) - C_{S''}(u) \geq c(e) + \sum_{e' \in P^v_w} \frac{c(e')}{x_s(e') + 1} - \sum_{e' \in P^w_{\overline{v}}} \frac{c(e')}{x_s(e') + 1} \geq c(e) + C^v_S(w) - C_S(v).$$

The lemma now follows since $c(e) + C_S^v(w) - C_S(v) \geq 0$ by Inequality(2).

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 3).

Lemma 3. *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 a 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 $\overline{G}$ 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.

We define EE, OPT, or $\overline{\text{OPT}}$ moves in $\overline{G}$ the same as we defined them in Section 2 where by edges of OPT in $\overline{G}$ we refer to both copies of each edge of OPT 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 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 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 $\text{OptFollow}(CurS, v, p(v))$ is an improving strategy for v. If it is improving then v changes her strategy to $\text{OptFollow}(CurS, v, p(v))$. 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 $\text{Follow}(CurS, v, w)$ is an improving strategy for v, then v changes her strategy to $\text{Follow}(CurS, v, w)$. 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 moves 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 3.

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 [1], 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 2 the resulting state is a Nash equilibrium.

6 The price of stability

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

Lemma 4. *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 $C_S(v) \le C_S^v(w) + d_{opt}(v, w)$ 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 $C_{S'}(v) \le C_S^v(w) + d_{opt}(v, w)$, 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'. The intuition of why this holds is as follows: From Lemma 4 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 5. *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

$$C_{S'}(v) + z < C_S(u). \tag{3}$$

Since in S there are no improving OPT moves and no improving EE moves, then, by Lemma 4,

$$C_S(u) \leq C_S^u(w) + d_{opt}(u, w). \tag{4}$$

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 2). 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,

$$C_S^u(w) = \sum_{e \in P_1} \frac{c(e)}{x_s(e) + 1} + \sum_{e \in P_2} \frac{c(e)}{x_s(e)} \leq \sum_{e \in S'_w} \frac{c(e)}{x_{s'}(e)} = C_{S'}(w), \tag{5}$$

as we claimed. From inequalities (4) and (5) we obtain

$$C_S(u) \leq C_{S'}(w) + d_{opt}(u, w). \tag{6}$$

Considering inequalities (3) and (6) 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 $d_{opt}(u, w) \leq \frac{z}{4}$,

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

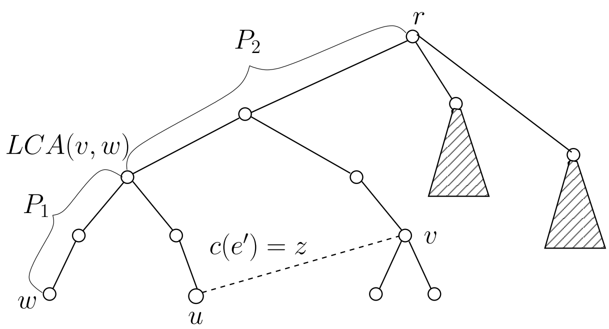
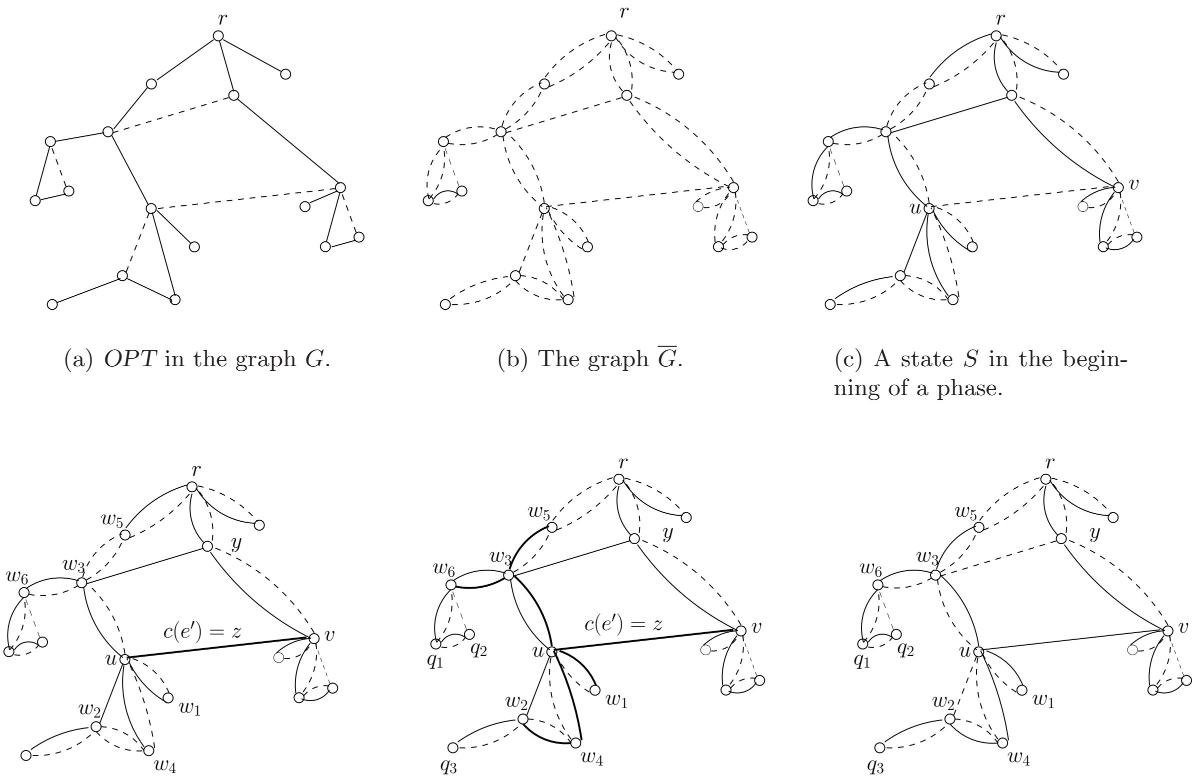


Fig. 2. Player u makes an $\overline{\text{OPT}}$-move and buys edge $e' = (u, v)$ of cost z. We assume that $d_{opt}(u, w) \leq \frac{z}{4}$.



(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.

Fig. 3. 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 $d_{opt}(u, w) \leq \frac{z}{4}$, 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). 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 $M_{w_1} = M_{w_3} = M_{w_4} = M_{w_5} = \emptyset$ (See Lemma 8).

Remark 2. Let u be the player making an improving $\overline{\text{OPT}}$ 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 5, $d_{opt}(u, w) > z/4$.

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 $d_{opt}(u, w_i) \leq d_{opt}(u, w_{i+1})$. In particular $w_0 = u$, and the vertex w_1 is adjacent to u in OPT. Lemma 5 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 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. *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 traversed by 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,

$$C_{S^i}(w_\ell) > C_{S^i}(v) + \frac{z}{2} + d_{opt}(u, w_\ell) \tag{7}$$

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 (7) holds by Lemma 5. Assume that Inequality (7) 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 (7) holds for S^{i+1} and for every $\ell \geq i + 1$.

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

$$C_{S^{i+1}}(w_\ell) - C_{S^i}(w_\ell) \geq C_{S^{i+1}}(v) - C_{S^i}(v) \tag{8}$$

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 (8) by showing that: (i) each edge that contributes a negative amount to the left side of (8) contributes the same negative amout to the right side of (8), (ii) no edge contributes positive amount to the right side of (8).

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 hypothesis 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 by the path S_v.

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

To further illustrate the proof of Lemma 6 consider the phase of the scheduler illustrated in Figure 3. The state $S' = S^1$ is the state after u changes her strategy to $(u \to v \to y \to r)$ shown in 3(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 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 7. *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, 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 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.

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 3(e), and 3(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 8. *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 3.

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 F. 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

$$\sum_{e \in T_x} \frac{c(e)}{x_T(e)} > \sum_{e \in F} \frac{c(e)}{x_{F(T)}(e)} \ . \tag{9}$$

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

$$\sum_{e \in T_x} \frac{c(e)}{x_{T'}(e)} > \sum_{e \in F} \frac{c(e)}{x_{F(T')}(e)} \ . \tag{10}$$

For every edge e, that q stops using, 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 10 holds assuming that Inequality 9 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.

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

Proof. Let e be a crowded edge in $N \setminus OPT$. By Lemma 7, 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 10 bounds the total cost of light edges.

Lemma 10. *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 4, $C_S(v) \leq C_S(r) + d_{opt}(v, r) = d_{opt}(v, r)$. 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 a subset $F \subseteq U$ as follows. Start with $T = U$ and $F = \emptyset$. 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 $\overline{\text{OPT}}$ 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 8, we have

$$d_{opt}(v_i, v_j) > \frac{z_{v_j}}{4}. \tag{11}$$

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

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

Substituting $j = i + 1$ and combining the Inequalities (11) and (12), 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 $z_{v_{i+1}} < z_{v_1}$, so by applying Equation

12 to v_{i+1} and v_1 we get $d_{opt}(v_{i+1}, v_1) \leq z_{v_1}/6$. 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 9 and 10 and is the main result of this work.

Theorem 1. *For a graph with a source vertex and a player in every vertex the price of stability is $O(\log\log n)$.*

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 4. 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 11. *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 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.

It follows from Lemma 11, that the unique Nash equilibrium of this game has a cost $2n$, and therefore the price of stability is $\frac{2n}{(7/6)n} \approx \frac{12}{7} > 1.7$.

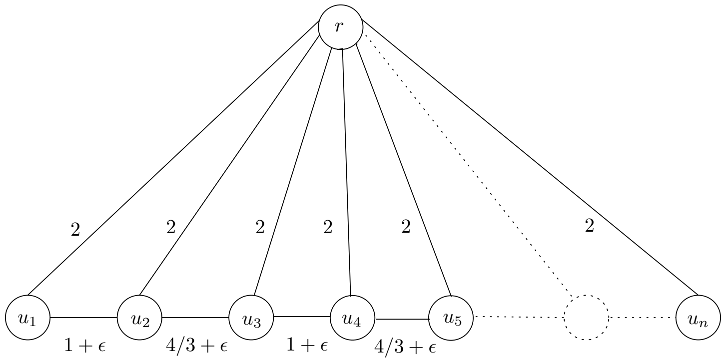


Fig. 4. Example of graph in which the price of stability > 1.7.

Acknowledgements

We would like to thank Micha Sharir for helpful discussions.

References

1. E. Anshelevich, A. Dasgupta, J. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. In *Proc. of 45th FOCS*, pages 295–304, 2004.
2. E. Anshelevich, A. Dasgupta, E. Tardos, and T. Wexler. Near-optimal network design with selfish agents. In *Proc. of 35th STOC*, pages 511–520, 2003.
3. B. Awerbuch, Y. Azar, and A. Epstein. Large the price of routing unsplittable flow. In *Proc. of 37th STOC*, pages 57–66, 2005.
4. B. Awerbuch, Y. Azar, Y. Richter, and D. Tsur. Tradeoffs in worst-case equilibria. In *Proc. of 1st WAOA*, pages 41–52, 2003.
5. G. Christodoulou and E. Koutsoupias. On the price of anarchy and stability of correlated equilibria of linear congestion games. In *Proc. of 13th ESA*, pages 59–70, 2005.
6. G. Christodoulou and E. Koutsoupias. The price of anarchy of finite congestion games. In *Proc. of 37th STOC*, pages 67–73, 2005.
7. J. Correa, A. Schulz, and N. S. Moses. Computational complexity, fairness, and the price of anarchy of the maximum latency problem.
8. J. Correa, A. Schulz, and N. S. Moses. Selfish routing in capacitated networks. *Journal of Mathematics of Operations Research*, pages 29(4):961–976, 2004.
9. A. Czumaj, P. Krysta, and B. Vocking. Selfish traffic allocation for server farms. In *Proc. of 34th STOC*, pages 287–296, 2002.
10. A. Czumaj and B. Vöcking. Tight bounds for worst-case equilibria. In *Proc. of 13th SODA*, pages 413–420, 2002.
11. M. Gairing, T. Lücking, M. Mavronicolas, and B. Monien. Computing nash equilibria for scheduling on restricted parallel links. In *Proc. of 36th STOC*, pages 613–622, 2004.
12. E. Koutsoupias, M. Mavronikolas, and P. Spirakis. Approximate equilibria and ball fusion. In *Proc. of 9th SIROCCO*, pages 223–235, 2002.
13. E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Proc. of 16th STACS*, pages 404–413, 1999.
14. M. Mavronicolas and P. Spirakis. The price of selfish routing. In *Proc. of 33rd STOC*, pages 510–519, 2001.
15. D. Monderer and L. Shapley. Potential games. *Games and Economic Behavior*, pages 14:124–143, 1996.
16. R. Rosenthal. A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, pages 2:65–67, 1972.
17. T. Roughgarden. The maximum latency of selfish routing. In *Proc. of 15th SODA*, pages 980–981, 2004.
18. T. Roughgarden and E. Tardos. How bad is selfish routing? *Journal of the ACM*, pages 49(2):236–259, 2002.
19. S. Suri, C. D. Toth, and Y. Zhou. Selfish load balancing and atomic congestion games. In *Proc. of 16th SPAA*, pages 188–195, 2004.