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]. We consider the case where there is an agent in every vertex. 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 *price of stability* [1] of a noncooperative game is the ratio between the cost of the least expensive Nash equilibria and the cost of the social optimum. The price of stability for network design games 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* [6] which is the ratio of the cost of the most expensive Nash Equilibrium and the cost of the social optimum.

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 [7,8] 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)[1]. 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

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

M. Bugliesi et al. (Eds.): ICALP 2006, Part I, LNCS 4051, pp. 608–618, 2006.

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.

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 [7,8]. In particular, latency minimization and some network construction/design games can be modeled as congestion games or weighted congestion games.

Most of the previous work has been focused on bounding the price of anarchy. The main focus was latency minimization for linear and polynomial latency functions [3,5,9]. The price of stability for linear latency functions has been studied by Christodoulou and Koutsoupias [4].

As most of previously considered games the game that 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 that 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. The price of stability of a different connection game was also considered by Anshelevitz *et al.* [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. [2]

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. [1,7], that maps every state S into a numeric value: $\Phi(S) = \sum_{e \in E} c(e) H(x_s(e))$, where $H(n) = 1 +$

[2] 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.

$1/2 + 1/3 + \cdots + 1/n$ 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.

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 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 S_u in the direction $x \to y$.

In the following definitions assume that v is the only player making the change, and we denote the new state by S' which is identical to S except that we replace S_v by S'_v. We say that a player v makes an *improvement move* when the player chooses a new strategy S'_v such that $C_{S'}(v) < C_S(v)$. We limit player v to choose strategies S'_v of the following three types.

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 $E(S') \nsubseteq E(S)$. Furthermore, if S'_v uses an edge $e = (x, y) \notin OPT$ in the direction $x \to y$ then $x \to y$ 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).

Overview. In Section 3 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 $O(\log \log n)$.

After an $\overline{\text{OPT}}$ move of a player u that adds the 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.

Let $c(u, v) = z$. This scheduling has two effects which our proof exploits.

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.

Our scheduling algorithm is described in Section 4. In Section 5 we prove the bound on the price of stability of the Nash Equilibrium obtained by the scheduler. Due to the space limit some of the proofs are omitted.

3 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. □

Lemma 2. *Let S be a state in which no player can make an OPT, $\overline{\text{OPT}}$, or EE improvement move. Then S is in a Nash equilibrium.*

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

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. We can show that:

Lemma 3. *For every Nash equilibrium in $\overline{G}$ there is a Nash equilibrium in G of the same cost.*

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.

The scheduler: We start the scheduler on $\overline{G}$ from an initial state isomorphic to OPT. We define the initial state S to consist of all edges $e^1 \in \overline{G}$ 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 $\overline{G}$. 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 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 scheduler: 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.

$\overline{\text{OPT}}$-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 that includes 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 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.

5 The Price of Stability

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

We introduce the following definitions. Let S be the state which is a tree. Assume we root the tree at r. 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 $P_w^v = P(w, LCA(v, w))$ and 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. *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'. 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

$$C_{S'}(v) + z < C_S(u) \ . \tag{1}$$

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

$$C_S(u) \leq C_S^u(w) + d_{opt}(u, w) \ . \tag{2}$$

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 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,

$$C_S^u(w) = \sum_{e \in P_1} \frac{c(e)}{x_s(e) + 1} + \sum_{e \in P_2} \frac{c(e)}{x_s(e)} \leq \sum_{e \in S'_w} \frac{c(e)}{x_{s'}(e)} = C_{S'}(w) \ , \tag{3}$$

as we claimed. From inequalities (2) and (3) we obtain

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

Considering inequalities (1) and (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 $d_{opt}(u, w) \leq \frac{z}{4}$,

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

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 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.

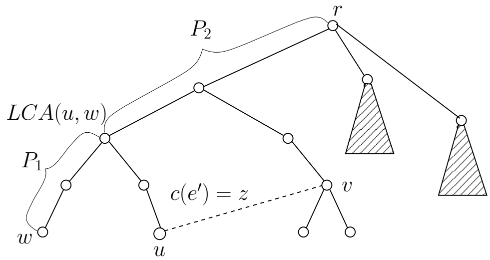

Fig. 1. Player u makes an $\overline{\text{OPT}}$-move and buys edge $e' = (u, v)$ of cost z. We assume that $d_{opt}(u, w) \leq \frac{z}{4}$.

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.*

Remark 2. 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 $\ell > i$. If these edges are not in S_v, and are not on the path between w_ℓ and u in *OPT* then this may lower the cost of S_{w_ℓ} such that when the scheduler gets to w_ℓ in the OPT-loop, her alternative OptFollow move is not improving.

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 $w \in D$. By the definition of the scheduler, the first edge in the strategy of w in S'' 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.

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 $e \notin S''$.*

The total cost of the edges in $N \cap OPT$ is no larger than the cost of OPT. We associate each edge $(u, v) \in N \setminus OPT$ 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. *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.

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. According to Lemma 4, $z_v < d_{opt}(v, r)$, so 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 $v_1, \ldots, v_m$ be the vertices of A_ξ in the order that their light edges $e_{v_1}, \ldots, e_{v_m}$ 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

$$d_{opt}(v_i, v_j) > \frac{z_{v_j}}{4} \ . \tag{5}$$

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

$$d_{opt}(v_i, v_j) \leq \frac{z_{v_i}}{12} + \frac{z_{v_j}}{12} \ . \tag{6}$$

Substituting $j = i + 1$ and combining the Inequalities (5) and (6), 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 6 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 $i < \log\log n$, 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* $O(\log\log n)$.

Acknowledgement

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. 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.

5. G. Christodoulou and E. Koutsoupias. The price of anarchy of finite congestion games. In *Proc. of 37th STOC*, pages 67–73, 2005.
6. E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Proc. of 16th STACS*, pages 404–413, 1999.
7. D. Monderer and L. Shapley. Potential games. *Games and Economic Behavior*, pages 14:124–143, 1996.
8. R. Rosenthal. A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, pages 2:65–67, 1972.
9. T. Roughgarden and E. Tardos. How bad is selfish routing? *Journal of the ACM*, pages 49(2):236–259, 2002.