The Hardness of Network Design for Unsplittable Flow with Selfish Users

Yossi Azar[1] and Amir Epstein[2]

[1] School of Computer Science, Tel-Aviv University, Tel-Aviv, 69978, Israel.
azar@tau.ac.il *

[2] School of Computer Science, Tel-Aviv University, Tel-Aviv, 69978, Israel.
amirep@tau.ac.il **

Abstract. In this paper we consider the network design for selfish users problem, where we assume the more realistic unsplittable model in which the users can have general demands and each user must choose a single path between its source and its destination. This model is also called atomic (weighted) network congestion game. The problem can be presented as follows : given a network, which edges should be removed to minimize the cost of the worst Nash equilibrium?

We consider both computational issues and existential issues (i.e. the power of network design). We give inapproximability results and approximation algorithms for this network design problem. For networks with linear edge latency functions we prove that there is no approximation algorithm for this problem with approximation ratio less then $(3+\sqrt{5})/2 \approx 2.618$ unless $P = NP$. We also show that for networks with polynomials of degree d edge latency functions there is no approximation algorithm for this problem with approximation ratio less then $d^{\Theta(d)}$ unless $P = NP$. Moreover, we observe that the trivial algorithm that builds the entire network is optimal for linear edge latency functions and has an approximation ratio of $d^{\Theta(d)}$ for polynomials of degree d edge latency functions. Finally, we consider general continuous, non-decreasing edge latency functions and show that the approximation ratio of any approximation algorithm for this problem is unbounded, assuming $P \neq NP$. In terms of existential issues we show that network design cannot improve the maximum possible bound on the price of anarchy in the worst case.

Previous results of Roughgarden for networks with n vertices where each user controls only a negligible fraction of the overall traffic showed optimal inapproximability results of $4/3$ for linear edge latency functions, $\Theta(d/\ln d)$ for polynomial edge latency functions and $n/2$ for general continuous non-decreasing edge latency functions. He also showed that the trivial algorithm that builds the entire network is optimal for that case.

* Research supported in part by the German-Israeli Foundation.
** Research supported in part by the German-Israeli Foundation.

1 Introduction

1.1 Selfish Routing

A major component of any large-scale network system is the routing mechanism, namely choosing a communication path between a sender and a receiver of traffic. In most cases, such as the Internet, wireless networks, or overlay networks built on top of the Internet, traffic from a sender to a receiver is sent over a *single* path; splitting the traffic causes the problem of packet reassembly at the receiver and thus is generally avoided. When choosing routing paths, the typical objective is to minimize the total latency. In most of these network systems it is infeasible to maintain one centralized authority that imposes efficient routing strategies on the network traffic. As a result users act independently and “selfishly”: each user tries to minimize his traffic cost based on current network traffic.

This problem can be mathematically formalized using classical game theory as follows. The network users are viewed as independent agents participating in a non-cooperative game. Each agent wishes to use the minimum latency path from its source to its destination, given the link congestion caused by the rest of the agents. This system is said to be in Nash Equilibrium if no agent has an incentive to change his path from its source to its destination. It is well known that Nash Equilibria do not in general optimize the social welfare (see, e.g, ”The Prisoner’s Dilemma” [7, 15]) and can be far from the global optimum.

Equilibria can be defined for pure strategies, where a single path is chosen by each user and for mixed strategies, where a probability distribution over the paths is used instead of a single path. Our hardness results hold for pure strategies and hence also for mixed strategies. Nash equilibrium requires mixed strategies, but in some cases pure strategies suffice [9, 14, 17].

The degradation of network performance caused by the lack of a centralized authority can be measured using the worst-case coordination ratio (price of anarchy) suggested by Koutsoupias and Papadimitriou [10] and Papadimitriou [16] which is the ratio between the worst possible Nash Equilibrium and the social optimum, see, e.g., [1, 4–6, 10, 11, 16, 19–21].

Braess’s paradox is the counterintuitive phenomenon that removing edges from a network can improve its performance. This paradox was first discovered by Braess [3] and later reported by Murchland [12]. Braess’s paradox motivates the following network design problem for improving the performance of a network with selfish users: How can we design selfish users networks to minimize the inefficiency inherent in Nash equilibrium?

Previous results of Roughgarden [18] for networks of n vertices with single source-sink pair where each user controls only a negligible fraction of the overall traffic showed optimal inapproximability results of $4/3$ for linear edge latency functions, $\Theta(d/\ln d)$ for polynomials of degree d edge latency functions and $n/2$ for general continuous non-decreasing edge latency functions. He also showed that the trivial algorithm that builds the entire network is optimal. For linear and polynomial edge latency functions these follow from price of anarchy results of Roughgarden and Tardos [21].

1.2 Our Results

We prove the following results for the network design problem for general networks with unsplittable flow:

- For linear latency functions we prove that for any $\epsilon > 0$ there is no $(\beta - \epsilon)$-approximation algorithm for network design where $\beta = (3 + \sqrt{5})/2 \approx 2.618$, assuming $P \neq NP$. Price of anarchy results appearing in [1] imply that this hardness result is optimal.
- For latency functions which are polynomials of degree d we prove that there is no approximation algorithm for network design, with approximation ratio less then $d^{\Theta(d)}$, assuming $P \neq NP$. Price of anarchy results appearing in [1] imply that the trivial algorithm has an approximation ratio of $d^{\Theta(d)}$. We note that our hardness result is $\Omega(d^{d/4})$ where the trivial algorithm's approximation ratio is $O(2^d d^{d+1})$.
- For general continuous, non-decreasing latency functions we show that the approximation ratio of any polynomial time approximation algorithm for NETWORK DESIGN is unbounded, assuming $P \neq NP$.

The above results deal with the computational issues related to the power of network design. We also consider the existential issues. Specifically we also consider the question whether network design can reduce the maximum bound on the price of anarchy in the worst case. We answer this negatively.

- For linear edge latency functions there is a network with coordination ratio at least $\beta - \epsilon$ where $\beta = (3 + \sqrt{5})/2 \approx 2.618$ for any $\epsilon > 0$, for polynomials of degree d edge latency functions there is a network with coordination ratio at least $\Omega(d^{d/4})$ and for general latency functions (continuous and non-decreasing) there is a network with unbounded coordination ratio such that in these networks network design cannot decrease the cost of the worst Nash equilibrium.

All our results hold for pure strategies and hence also for mixed strategies, since these are hardness and non existential results.

Techniques: To prove our hardness results we first prove hardness results to SELECTIVE NETWORK DESIGN which is an harder problem than NETWORK DESIGN. Then we show a general way to transform many types of hardness results of selective network design to hardness results of network design.

1.3 Paper structure

The paper is organized as follows. Section 2 includes formal definitions and notations. In section 3 we prove inapproximability results for NETWORK DESIGN and observe the approximation ratio of the trivial algorithm for linear and polynomial latency functions. In section 4 we consider the existential issues of NETWORK DESIGN and show that it cannot reduce the maximum bound on the price of anarchy.

2 Definitions and preliminaries

2.1 The Model

We consider the following model which is called weighted network congestion game: there is a directed graph $G = (V, E)$. Each edge $e \in E$ is given a load-dependent latency function $f_e : \mathcal{R}^+ \to \mathcal{R}^+$. There are n users, where user j ($j = 1, \ldots, n$) has a bandwidth request defined by a tuple (s_j, t_j, w_j), where $s_j, t_j \in V$ are the source/destination pair, and $w_j \in \mathcal{R}^+$ corresponds to the required bandwidth. We denote the set of (simple) $s_j - t_j$ paths by $\mathcal{Q}_j$. Request j can be assigned to any path Q from the set of paths $\mathcal{Q}_j$, such that the required bandwidth w_j has to be reserved along the path Q.

We assume that the users are non-cooperative and each one wishes to minimize its own cost with no regard to the global optimum. In Pure strategies user j selects a single path $Q \in \mathcal{Q}_j$ and assigns his request to it. Each user is aware of the choices made by all other users when making his decision.

2.2 Pure strategies definition

First, we give some simpler notations we use for a system $\mathcal{S} = (Q_1, \ldots, Q_n)$ of pure strategies. Let Q_j be the path associated with request j. We define $J(e) = \{j | e \in Q_j\}$ the set of requests assigned to a path containing the edge e. The load on edge e is defined by: $l_e = \sum_{j \in J(e)} w_j$.

For the optimal routes let Q_j^* be the path associated with request j. We define $J^*(e) = \{j | e \in Q_j^*\}$ the set of requests assigned to a path containing the edge e. We denote the load on edge e by l_e^*.

Definition 1. *The latency of user j for assigning his request in system $\mathcal{S}$ to path Q (instead of path Q_j) is defined as:*

$$c_{Q,j} = \sum_{(e \in Q) \wedge (e \in Q_j)} f_e(l_e) + \sum_{(e \in Q) \wedge (e \notin Q_j)} f_e(l_e + w_j). \tag{1}$$

2.3 Nash equilibrium and Coordination ratio

Nash equilibrium is characterized by the property that there is no incentive for any user to change its strategy and defined as follows

Definition 2. (Nash Equilibrium) *A system $\mathcal{S}$ is said to be in pure Nash Equilibrium if and only if for every $j \in \{1, \ldots, n\}$ and $Q \in \mathcal{Q}_j$, $c_{Q_j,j} \leq c_{Q,j}$.*

Definition 3. *The cost $C(\mathcal{S})$ for a given system $\mathcal{S}$ of pure strategies is defined as the total latency incurred by $\mathcal{S}$, that is $C(\mathcal{S}) = \sum_{e \in E} f_e(l_e) l_e$.*

We are interested in estimating the worst-case coordination ratio when pure Nash equilibrium exists. We denote the optimal system of pure strategies by $\mathcal{S}^*$.

Definition 4. (Coordination Ratio) *The coordination ratio is defined as $R = max_{\mathcal{S}} \frac{C(\mathcal{S})}{C(\mathcal{S}^*)}$, where the maximum is taken over all strategies $\mathcal{S}$ in Nash equilibrium.*

2.4 Formalizing the Network Design Problem

Let $C(H, \mathcal{S})$ be the total latency incurred by a given system $\mathcal{S}$ of pure strategies in Nash equilibrium for a subgraph H of G. If there is a user j such that $\mathcal{Q}_j = \emptyset$ in the subgraph H then $C(H, \mathcal{S}) = \infty$. We denote by $C(H)$ the maximum cost obtained for the graph H, where the maximum is taken over all strategies $\mathcal{S}$ in Nash equilibrium for the graph H. We note that for unsplittable flow we do not know how to compute the value $C(H)$ in polynomial time, while for the case of splittable flow (or alternatively where each user controls a negligible amount of the traffic) the value $C(H)$ can be recovered from the subgraph H in polynomial time via convex programming for positive convex functions (see [2]). Now we define the network design and selective network design problems for unsplittable flow.
The Network Design Problem: Given a weighted network congestion game with directed graph $G = (V, E)$, find a subgraph H of G that minimizes $C(H)$.
The Selective Network Design Problem: Given a weighted network congestion game with directed graph $G = (V, E)$ and $E_1 \subseteq E$, find a subgraph H of G containing the edges of E_1 that minimizes $C(H)$.

The above formulation of the SELECTIVE NETWORK DESIGN problem is itself interesting, but the main purpose of the presentation of this problem is for proving inapproximability results for the NETWORK DESIGN problem. In particular we first prove hardness results for the selective network design problem (which is a harder problem than the network design problem and hence it is easier to show hardness results for this problem) and then we modify the instance of the selective network design problem used in the proof of inapproximability of selective network design to an instance of the network design problem to show its inapproximability result.

3 Inapproximabilty of Network Design

In this section we consider the computational issues of NETWORK DESIGN. Specifically we prove inapproximabilty results for NETWORK DESIGN and observe the approximation ratio of the trivial algorithm for linear and polynomial latency functions.

3.1 Linear Latency Functions

In this section we consider the case where the latency of each edge is linear in the edge congestion. Specifically $f_e(x) = a_e x + b_e$ for each edge $e \in E$, where a_e and b_e are nonnegative reals. Let $\beta = (3 + \sqrt{5})/2 \approx 2.618$.

A trivial algorithm for the problem outputs the entire network G. We begin by observing that this trivial algorithm for NETWORK DESIGN is a β-approximation algorithm, where the latency functions are linear. This will follow easily from a result of Awerbuch et al. [1]. They proved that in every network with linear latency functions and unsplittable flow, the cost of unsplittable flow at Nash equilibrium is at most β times that of every other feasible unsplittable flow.

Proposition 1. *([1]) For linear latency functions and weighted demands let S^* be a system of strategies and let S be a system of strategies in Nash equilibrium. Then $C(S) \leq \beta \cdot C(S^*)$.*

Corollary 1. *The trivial algorithm is a β-approximation for linear latency functions and weighted demands.*

Proof. Consider an instance of the problem with subgraph H of G minimizing $C(H)$. Let S and S^* denote systems of strategies at Nash equilibrium for the graphs G and H, respectively. Since S^* can be viewed as a system of strategies for the graph G, it follows from proposition 1 that $C(G, S) \leq \beta \cdot C(G, S^*)$ and hence $C(G) \leq \beta \cdot C(H)$.

The main result of this section is a lower bound on the approximation ratio of any polynomial algorithm (unless P=NP).

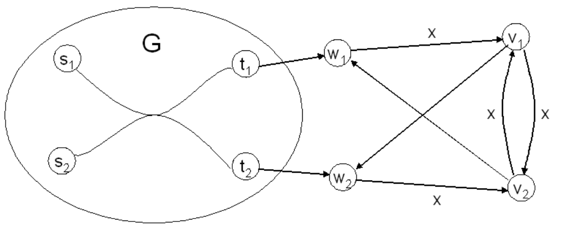

Fig. 1. Proof of Theorem 1

Theorem 1. *For linear latency functions and weighted demands assuming $P \neq NP$ there is no $(\beta - \epsilon)$-approximation algorithm for SELECTIVE NETWORK DESIGN (recall that $\beta = (3 + \sqrt{5})/2 \approx 2.618$).*

Proof. We reduce from the problem 2 Directed Disjoint Paths (2DDP): Given a directed graph $G = (V, E)$ and distinct vertices $s_1, s_2, t_1, t_2 \in V$, are there s_i-t_i paths P_1 and P_2, such that P_1 and P_2 are vertex disjoint? Fortune et al. [8] proved that this problems is NP-complete. We will show that for linear latency functions and weighted demands $(\beta - \epsilon)$-approximation algorithm for the SELECTIVE NETWORK DESIGN problem can be used to distinguish "yes" and "no" instances of 2DDP in polynomial time. Consider an instance I of 2DDP, as above. We add the vertices w_1, w_2, v_1 and v_2 to the vertex set V and include directed edges (t_1, w_1), (t_2, w_2), (w_1, v_1), (w_2, v_2), (v_1, v_2), (v_2, v_1), (v_1, w_2) and (v_2, w_1) as shown in Figure 1. We denote the new network by $G' = (V', E')$. Let $E_1 := E' - E$ be the group of edges that the subgraph H of G' should contain. We define the following linear latency functions f for the edges of E': the edges

$(w_1, v_1), (w_2, v_2), (v_1, v_2), (v_2, v_1)$ are given the latency functions $f(x) = x$ and all other edges are given the latency functions $f(x) = 0$. We later choose $\phi = \frac{1+\sqrt{5}}{2}$ which is the golden ratio. We consider an atomic weighted network congestion game with six players that uses the network G'. Player 1 has a bandwidth request (s_1, v_1, ϕ) (player 1 has to move ϕ units of bandwidth from s_1 to v_1), player 2 has a bandwidth request (s_2, v_2, ϕ), player 3 has a bandwidth request $(v_1, v_2, 1)$, player 4 has a bandwidth request $(v_2, v_1, 1)$, player 5 has a bandwidth request $(s_1, t_1, 1)$ and player 6 has a bandwidth request $(s_2, t_2, 1)$. The new instance I' can be constructed from I in polynomial time. To complete the proof, it suffices to show the following two statements.

1. If I is a "yes" instance of 2DDP, then G' contains a subgraph H of G' with $C(H) = 2\phi^2 + 2$.
2. If I is a "no" instance of 2DDP, then $C(H) \geq 2(\phi+1)^2+2\phi^2$ for all subgraphs H of G'.

Recall that the subgraph H of G' should contain the edges in E_1. To prove (1), let P_1 and P_2 be vertex-disjoint paths in G, respectively, and obtain H by deleting all edges of G not contained in some P_i. Then, H is a subgraph of G' that contains the paths $s_1 - t_1 - w_1 - v_1$, $s_2 - t_2 - w_2 - v_2$, $v_1 - v_2$, $v_2 - v_1$, $s_1 - t_1$ and $s_2 - t_2$. These paths are the direct paths of players $1 - 6$ respectively. The optimal solution S_1 is obtained when each player chooses its direct path and this solution is the only Nash equilibrium for I' in which the costs of players $1 - 6$ are $\phi^2, \phi^2, 1$, 1,0 and 0 respectively. The total cost $C(H, S_1) = 2\phi^2 + 2$. This solution is the unique Nash Equilibrium, since the dominant strategy of each of the players $1, 2, 5, 6$ is to choose its direct path which is its unique simple path and given these strategies of players $1, 2, 5, 6$ the best response of each of the players 3 and 4 is its direct path. For (2), we may assume that H contains $s_1 - t_1$ and $s_2 - t_2$ paths. In this case the paths $s_1 - t_1$ and $s_2 - t_2$ are not disjoint and hence H must contain $s_1 - t_2$ and $s_2 - t_1$ paths. Let S_2 be the system of strategies where player 1 uses its indirect path $s_1 - t_2 - w_2 - v_2 - v_1$, player 2 uses its indirect path $s_2 - t_1 - w_1 - v_1 - v_2$, player 3 uses its indirect path $v_1 - w_2 - v_2$, player 4 uses its indirect path $v_2 - w_1 - v_1$, player 5 uses its direct path $s_1 - t_1$ and player 6 uses its direct path $s_2 - t_2$. Then this is a Nash equilibrium and the costs of players $1 - 6$ are $2\phi + 1$, $2\phi + 1$, $\phi + 1$, $\phi + 1$, 0 and 0 respectively. The total cost $C(H, S_2) = 2(\phi + 1)^2 + 2\phi^2$. The ratio of the total costs $C(H, S_2)$ and $C(H, S_1)$ is :

$$\frac{2(\phi + 1)^2 + 2\phi^2}{2\phi^2 + 2}.$$

We choose $\phi = \frac{1+\sqrt{5}}{2}$ which is the golden ratio and get a ratio $\beta = \phi+1 \approx 2.618$. This completes the proof.

We call a family X of latency functions **nice** if all of its functions are non-negative, continuous and non-decreasing and the family is closed under non-negative linear combinations. Note that ,obviously, linear and polynomial latency functions satisfy this definition.

The following Lemma provides a way to transform inapproximability result of SELECTIVE NETWORK DESIGN to inapproximability result of NETWORK DESIGN.

Lemma 1. *Given a direct reduction from a hard problem Q to SELECTIVE NETWORK DESIGN for a nice family of latency functions that shows that it is hard to c-approximate selective network design, then one can create a similar reduction from Q to NETWORK DESIGN for the same family of latency functions that shows that it is hard to c-approximate network design, if the following condition applies : for every instance of selective network design created by the reduction with weighted network congestion game consisting of graph $G' = (V', E')$, $E_1 \subseteq E'$ and every subgraph $H \subseteq G'$ that has been considered in the proof (i.e. that contains E_1) it holds that in the worst Nash equilibrium each player has a unique best response (best strategy).*

Proof. For every instance of SELECTIVE NETWORK DESIGN created by the reduction with weighted network congestion game consisting of graph $G' = (V', E')$, $E_1 \subseteq E'$ and every subgraph $H \subseteq G'$ that has been considered in the proof (i.e. that contains E_1) we do the following. Let $\delta > 0$. For each edge $e \in E_1$ we make the following local modification. First we split the edge by adding a new vertex w_e and replacing the edge $e = (u, v)$ by the two edges $e_1 = (u, w_e)$ and $e_2 = (w_e, v)$. The new edges e_1 and e_2 will posses the latency function $\frac{1}{2} f_e$. Then we add two players with requests (u, w_e, δ) and (w_e, v, δ). We denote the modified network created from H by $H^* = (V^*, E^*)$. Since the costs of the players change continuously as a function of δ, for sufficiently small constant δ it holds that in the new weighted network congestion game the worst Nash equilibrium remains a Nash equilibrium where each player uses its original strategy and this strategy is its unique best response (the new players choose their unique strategy). Moreover, the total cost changes continuously as a function of δ and hence the new total cost is arbitrarily close to the original total cost as a function of δ. Additionally, each of the edges in E_1 cannot be deleted since it is a unique strategy of a new player. Hence the inapproximablity proof for SELECTIVE NETWORK DESIGN is also a proof for NETWORK DESIGN.

Unfortunately we cannot use Lemma 1 to prove Theorem 2 according to the result of Theorem 1, hence we have to modify the weighed network congestion game used in the proof of Theorem 1 to satisfy the condition required by Lemma 1.

Theorem 2. *For linear latency functions and weighted demands assuming $P \neq NP$ there is no $(\beta - \epsilon)$-approximation algorithm for NETWORK DESIGN (recall that $\beta = (3 + \sqrt{5})/2 \approx 2.618$).*

Proof. We modify the weighted network congestion game defined in the proof of Theorem 1 as follows : Let $\epsilon > 0$. First we modify the network $G' = (V', E')$ shown in Figure 1 and obtain the network $G'' = (V'', E'')$ shown in Figure 2. Next we modify the requests of players 3 and 4. Player 3 has a bandwidth request

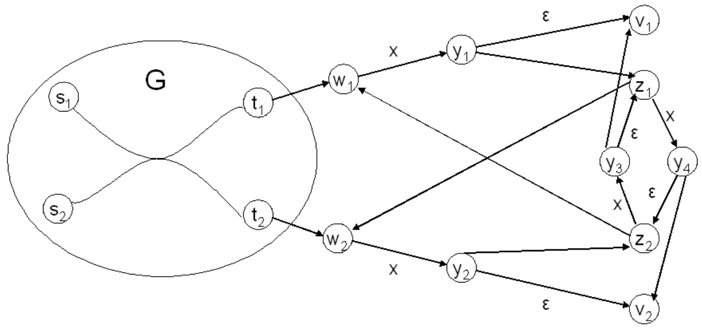

Fig. 2. Proof of Theorem 2

$(z_1, z_2, 1)$ (its previous request was $(v_1, v_2, 1)$) and player 4 has a bandwidth request $(z_2, z_1, 1)$ (its previous request was $(z_2, v_1, 1)$). The direct paths of players $1-6$ are $s_1 - t_1 - w_1 - y_1 - v_1$, $s_2 - t_2 - w_2 - y_2 - v_2$, $z_1 - y_4 - z_2$, $z_2 - y_3 - z_1$, $s_1 - t_1$ and $s_2 - t_2$ respectively. The indirect paths of players $1 - 4$ are $s_1 - t_2 - w_2 - y_2 - z_2 - y_3 - v_1$, $s_2 - t_1 - w_1 - y_1 - z_1 - y_4 - v_2$, $z_1 - w_2 - y_2 - z_2$, $z_2 - w_1 - y_1 - z_1$ respectively. Now it is easy to verify according to the proof of Theorem 1 that the following properties hold:

1. The optimum which is the best Nash equilibrium is obtained when each player chooses its direct path.
2. The worst Nash equilibrium is obtained when each of the players $1-4$ chooses its indirect path and players $5, 6$ choose their direct path.
3. In the best and worst Nash equilibria the total cost was increased by at most 8ϵ.
4. In the best and worst Nash equilibria each player has a unique best response (best setrategy).

Let $E_1 = E'' - E$ be the group of edges that the subgraph H of G'' should contain. It follows from the above properties and the proof of Theorem 1 that the above modified weighted network congestion game can be used to prove Theorem 1. It also follows that for every subgraph considered in the new proof of Theorem 1 which uses the modified weighted network congestion game, in the worst Nash equilibrium each player has a unique best response (best startegy). Applying Lemma 1 completes the proof.

3.2 Polynomial Latency Functions

In this section we consider the case where the latency of each edge is a polynomial of degree d in the edge congestion. Specifically $f_e(x) = \sum_i a_{e,i} x^i$ for each edge $e \in E$, where $a_{e,i}$ are nonnegative reals.

Proposition 2. *([1]) For polynomial of degree d latency functions and weighted demands let S^* be a system of strategies and let S be a system of strategies in Nash equilibrium. Then $C(S) \leq O(2^d d^{d+1}) \cdot C(S^*)$.*

Corollary 2. *The trivial algorithm is a $O(2^d d^{d+1})$-approximation for linear latency functions and weighted demands.*

The main results of this section are lower bounds on the approximation ratio of any polynomial algorithm for weighted demands (unless P=NP).

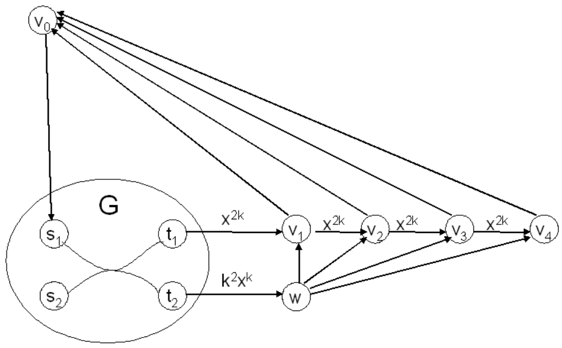

Fig. 3. Proof of Theorem 3. In this example $n = 4$

Theorem 3. *For polynomials of degree d latency functions and weighted demands assuming $P \neq NP$ there is a lower bound of $\Omega(d^{d/4})$ on the approximation ratio of any polynomial time approximation algorithm for SELECTIVE NETWORK DESIGN.*

Proof. Let $c = 2$,let $d = 2k$ (we can assume that d is even), let $n = k\sqrt{k}/c$. We reduce from the problem 2 Directed Disjoint Paths (2DDP): Given a directed graph $G = (V, E)$ and distinct vertices $s_1, s_2, t_1, t_2 \in V$, are there s_i-t_i paths P_1 and P_2, such that P_1 and P_2 are vertex disjoint? Fortune et al. [8] proved that this problems is NP-complete. We will show that for polynomials of degree d latency functions and weighted demands $O(d^{d/4})$-approximation algorithm for the SELECTIVE NETWORK DESIGN problem can be used to distinguish "yes" and "no" instances of 2DDP in polynomial time. Consider an instance I of 2DDP, as above. We now build the graph $G' = (V', E')$ shown in Figure 3. Let $E_1 = E' - E$ be the group of edges that the subgraph H of G' should contain. We begin by adding the vertices w and $v_0, \ldots, v_n$ to the vertex set V and include directed edges (v_0, s_1), (t_1, v_1), (t_2, w), (v_i, v_{i+1}) for $i = 1, \ldots, n-1$, (v_i, v_0) for $i = 1, \ldots, n$ and (w, v_i) for $i = 1, \ldots, n$. Next we add the edge latency functions. Edges (t_1, v_1) and (v_i, v_{i+1}) for $i = 1, \ldots, n-1$ will possess the latency function $f(x) = x^{2k}$, edge (t_2, w) will possess the latency function $f(x) = k^2x^k$, all other edges will possess the latency function $f(x) = 0$. Let $\delta > 0$ be sufficiently small. We consider an atomic weighted network congestion game with $n+3$ players that use the network G'. Player 1 has a bandwidth request (s_2, v_n, k). For $i = 2 \ldots n+1$ player i has a bandwidth request $(v_{i-2}, v_{i-1}, c\sqrt{k})$. Player $n+2$

has a bandwidth request (s_1, t_1, δ) and player $n+3$ has a bandwidth request (s_2, t_2, δ). The new instance I' can be constructed from I in polynomial time. To complete the proof, it suffices to show the following two statements.

1. If I is a "yes" instance of 2DDP, then G' contains a subgraph H of G' with $C(H) = k^{k+2}2^{2k} + k^{k+3}$.
2. If I is a "no" instance of 2DDP, then $C(H) \geq k^{2k+4}$ for all subgraphs H of G'.

To prove (1), let P_1 and P_2 be vertex-disjoint paths in G, respectively, and obtain H by deleting all edges of G not contained in some P_i. Then, H is a subgraph of G'. There is one simple path for each player. The optimal solution is obtained when each player chooses its direct path as follows. Player 1 chooses the path $s_2 - t_2 - w - v_n$,player 2 chooses the path $v_0 - s_1 - t_1 - v_1$, for $i = 3, \ldots n+1$ player i chooses the path $v_{i-2} - v_{i-1}$, player $n+2$ chooses the path $s_1 - t_1$ and player $n+3$ chooses the path $s_2 - t_2$. This solution is the only Nash equilibrium for I', in which $C(H, S) = \sum_{e \in E} f_e(l_e) l_e = n(2\sqrt{k})^{2k+1} + k^2 \cdot k^{k+1} = k\sqrt{k}/2(2\sqrt{k})^{2k+1} + k^2 \cdot k^{k+1} = k^{k+2}2^{2k} + k^{k+3}$. For (2), we may assume that H contains $s_1 - t_1$ and $s_2 - t_2$ paths. In this case H must contain $s_1 - t_2$ and $s_2 - t_1$ paths to satisfy the requests for paths $s_1 - t_1$ and $s_2 - t_2$. If player 1 uses its indirect path $s_2 - t_1 - v_1 - v_2 - \ldots - v_n$, for $i = 2 \ldots n+1$ player i uses its indirect path $v_{i-2} - v_0 - s_1 - t_2 - w - v_{i-1}$, player $n+2$ uses its direct path $s_1 - t_1$ which must exist and player $n+3$ uses its direct path $s_2 - t_2$ if it exists, otherwise it uses its indirect path $s_2 - t_1 - v_1 - v_0 - s_1 - t_2$, then this is a Nash equilibrium with $C(H, S) \geq k^2 \cdot k^{2k+2} + k \cdot k^{2k+3/2}/2 = k^{2k+4} + k^{2k+5/2}/2$. To show that this is a Nash equilibrium we have to show that no player benefits from changing its path. We assume that player $n+3$ uses its indirect path $s_2 - t_1 - v_1 - v_0 - s_1 - t_2$. The analysis of the case when player $n+3$ uses its direct path $s_2 - t_2$ follows from this case. The cost of player 1 on path $s_2 - t_1 - v_1 - v_2 - \ldots - v_n$ is $k\sqrt{k}/2 \cdot k^{2k} = k^{2k+3/2}/2$. The cost of player 1 on path $s_2 - t_2 - w - v_n$ is $k^2 \cdot (k^2 + k + \delta)^k > k^{2k+2}$, which is greater. For $i = 2 \ldots n+1$ the cost of player i on path $v_{i-2} - v_0 - s_1 - t_2 - w - v_{i-1}$ is $k^2 \cdot (k^2 + \delta)^k \geq k^{2k+2}$. The cost of player i on path $v_{i-2} - v_{i-1}$ is $(k + 2\sqrt{k})^{2k} > k^2 \cdot (k^2 + \delta)^k$ for sufficiently small δ (but at least one divided by a polynomial in k). Players $n+2$ and $n+3$ cannot decrease their cost by changing path (if one exists). This completes the proof.

Theorem 4. *For polynomials of degree d latency functions and weighted demands assuming $P \neq NP$ there is a lower bound of $\Omega(d^{d/4})$ on the approximation ratio of any polynomial time approximation algorithm for NETWORK DESIGN.*

Proof. In any Nash equilibrium considered in the proof of Theorem 3 every player has a unique best response, hence the result follows from Lemma 1.

3.3 General Latency Functions

In this section we consider the case where the latency of each edge is continuous and non-decreasing in the edge congestion. We show that the approximation ratio of any approximation algorithm is unbounded even as a function of n.

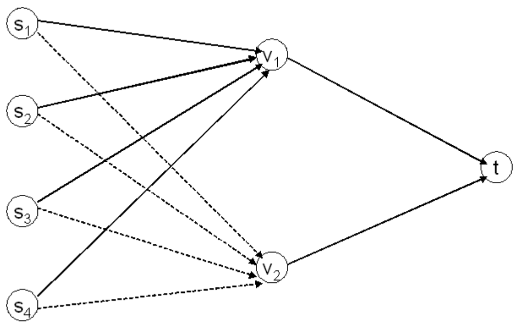

Fig. 4. Proof of Theorem 5. In this example $n = 4$

Theorem 5. *For general continuous, non-decreasing latency functions assuming $P \neq NP$ the approximation ratio of any polynomial time approximation algorithm for NETWORK DESIGN is unbounded.*

Proof. We show that it is NP-hard to differentiate between zero cost and positive cost. We reduce from the NP-complete problem PARTITION: we are given q positive integers $\{a_1, a_2, \dots, a_q\}$ and seek for a subset $T \subseteq \{1, 2, \dots, q\}$ such that $\sum_{j \in T} a_j = \frac{1}{2} \sum_{j=1}^{q} a_j$ [13]. Consider an instance I of PARTITION, as above. We now build the directed graph $G = (V, E)$ shown in Figure 4. Let $n = q$, let $A = \sum_{j=1}^{q} a_j$, $V = \{s, t, v_1, v_2, \dots, v_n\}$ and E includes the edges (s_i, v_1) for $i = 1, \dots, n$, (s_i, v_2) for $i = 1, \dots, n$, (v_1, t) and (v_2, t). The edges (v_1, t) and (v_2, t) will posses the latency function f satisfying $f(x) = 0$ for $x \leq A/2$ and $f(x) = x - A/2$ for $x \geq A/2$, all other edges will posses the latency function $f(x) = 0$. We consider an atomic weighted network congestion game with n players that uses the network G. For $i = 1 \dots n$ player i has a bandwidth request (s_i, t, a_i).

The new instance I' can be constructed from I in polynomial time. To complete the proof, it suffices to show the following two statements.

1. If I is a "yes" instance of PARTITION, then G contains a subgraph H of G with $C(H) = 0$.
2. If I is a "no" instance of PARTITION, then $C(H) > 0$ for all subgraphs H of G.

To prove (1), let the subset Y be the solution to the instance I, we obtain H by deleting all edges (s_i, v_2) for $i \in Y$ and deleting all edges (s_i, v_1) for i not in Y. Each player has a unique path (strategy) in the graph H. The load on each of the edges (v_1, t) and (v_2, t) is $A/2$ and hence $C(H, S) = 0$. For (2), we may assume that H contains $s_i - t$ path for each $i = 1, \dots, n$. Let Y' be the subset of players using paths containing the edge (v_1, t) (all other players use paths containing the edge (v_2, t)), then it holds that the load of one of the edges (v_1, t) and (v_2, t) is greater then $A/2$ and hence $C(H, S) > 0$.

4 The Limitation on the Power of Network Design

In this section we consider the existential issues of NETWORK DESIGN. Specifically we consider the question whether network design can reduce the maximum bound on the price of anarchy. We answer this negatively.

Theorem 6. *For any $\epsilon > 0$ and for linear latency functions there is a network with coordination ratio at least $\beta - \epsilon$ in which NETWORK DESIGN cannot decrease the cost of the worst Nash equilibrium (recall that $\beta = (3 + \sqrt{5})/2 \approx 2.618$).*

Proof. The proof follows from the weighted network congestion game with the graph G'' constructed in the proof of Theorem 2 where the graph G is contracted to a single vertex. For each edge in the graph G'' we apply the local modification described in the proof of Lemma 1 and obtain a new weighted network congestion game with coordination ratio at least $\beta - \epsilon$ where edges cannot be removed.

Theorem 7. *For polynomial of degree d latency functions there is a network with coordination ratio at least $\Omega(d^{d/4})$ in which NETWORK DESIGN cannot decrease the cost of the worst Nash equilibrium.*

Proof. The proof follows from the weighted network congestion game with the graph G' constructed in the proof of Theorem 3 where the graph G is contracted to a single vertex. For each edge in the graph G' we apply the local modification described in the proof of Lemma 1 and obtain a new weighted network congestion game with coordination ratio at least $\Omega(d^{d/4})$ where edges cannot be removed.

Theorem 8. *For general latency functions (continuous and non-decreasing) there is a network with unbounded coordination ratio such that in this network NETWORK DESIGN cannot decrease the cost of the worst Nash equilibrium.*

Proof. We prove the result by showing a weighted network congestion game for network with edges that cannot be removed (since each edge is a unique path of a player). In this game there is Nash equilibrium with zero cost and Nash equilibrium with positive cost as follows. We consider a weighted network congestion game that uses the network defined in the proof of Theorem 5 and shown in Figure 4. We denote the new network by $G = (V, E)$. Let the number of source vertices $n = 4$ and let $A = 12$. We define the following players: players $1-4$ have bandwidth requests $(s_1, t, 2)$, $(s_2, t, 3)$, $(s_3, t, 2)$, $(s_4, t, 3)$ respectively. For each $i = 1 - 4$ we add two players with requests $(s_i, v_1, 1)$ and $(s_i, v_2, 1)$. Additionally we add two players with requests $(v_1, t, 1)$ and $(v_2, t, 1)$. When players $1, 2$ choose their simple paths containing the edge (v_1, t), players $3, 4$ choose their simple paths containing the edge (v_2, t) and all other players use their unique path, then this is the optimal solution and it is also the best Nash equilibrium with cost $C(H, S_1) = 0$. Additional Nash equilibrium is obtained when players $1, 3$ choose their simple paths containing the edge (v_1, t), players $2, 4$ choose their simple path containing the edge (v_2, t) and all other players use their unique path. The cost of this Nash equilibrium $C(H, S_2) > 0$.

References

1. B. Awerbuch, Y. Azar, and A. Epstein. The price of routing unsplittable flow. In *Proc. 37th ACM Symp. on Theory of Computing*, 2005. To appear.
2. M. Beckmann, C.B. McGuire, and C.B. Winsten. *Studies in Economics of Transportation.* Yale University Press, 1956.
3. D. Braess. Uber ein paradoxon der verkehrsplanung. *Unternehmensforschung*, 12:258–268, 1968.
4. G. Christodoulou and E. Koutsoupias. The price of anarchy of finite congestion games. In *Proc. 37th ACM Symp. on Theory of Computing*, 2005. To appear.
5. A. Czumaj, P. Krysta, and B. Vöcking. Selfish traffic allocation for server farms. In *Proc. 34th ACM Symp. on Theory of Computing*, pages 287–296, 2002.
6. A. Czumaj and B. Vöcking. Tight bounds for worst-case equilibria. In *Proc. 13rd ACM-SIAM Symp. on Discrete Algorithms*, pages 413–420, 2002.
7. P. Dubey. Inefficiency of nash equilibria. *Mathematics of Operations Research*, 11(1):1–8, 1986.
8. S. Fortune, J.E. Hopcroft, and J.C. Wyllie. The directed subgraph homeomorphism problem. *Theoretical Computer Science*, 10(2):111–121, 1980.
9. D. Fotakis, S. Kontogiannis, and P. Spirakis. Selfish unsplittable flows. In *International Colloquium on Automata, Languages and Programming - ICALP '04*, 2004.
10. E. Koutsoupias and C.H. Papadimitriou. Worst-case equilibria. In *Proc. 16th Symp. on Theoretical Aspects of Comp. Science*, pages 404–413, 1999.
11. M. Mavronicolas and P. Spirakis. The price of selfish routing. In *Proc. 33rd ACM Symp. on Theory of Computing*, pages 510–519, 2001.
12. J.D. Murchland. Braess's paradox of traffic flow. *Transportation Research*, 4:391–394, 1970.
13. M.R. Garey nad D.S. Johnson. *Computers and Intractabilty: A Guide to the Theory of NP-Completeness.* Freeman, 1979.
14. J. F. Nash. Equilibrium points in n-person games. In *Proceedings of National Academy of Sciences*, volume 36, pages 48–49, 1950.
15. G. Owen. *Game Theory.* Academic Press, third edition, 1995.
16. C.H. Papadimitriou. Algorithms, games and the internet. In *Proc. 33rd ACM Symp. on Theory of Computing*, pages 749–753, 2001.
17. R. W. Rosenthal. A class of games possesing pure-strategy nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.
18. T. Roughgarden. Designing networks for selfish users is hard. In *Proc. 42nd IEEE Symp. on Found. of Comp. Science*, pages 472–481, 2001.
19. T. Roughgarden. The price of anarchy is independent of the network topology. In *Proc. 34th ACM Symp. on Theory of Computing*, pages 428–437, 2002.
20. T. Roughgarden. The maximum latency of selfish routing. In *Proc. 15rd ACM-SIAM Symp. on Discrete Algorithms*, pages 973–974, 2004.
21. T. Roughgarden and É. Tardos. How bad is selfish routing. In *Proc. 41st IEEE Symp. on Found. of Comp. Science*, pages 93–102, 2000.