Noname manuscript No.
(will be inserted by the editor)

Price of Stability in Survivable Network Design

Elliot Anshelevich · Bugra Caskurlu

Received: January 2010 / Accepted:

Abstract We study the survivable version of the game theoretic network formation model known as the Connection Game, originally introduced in [5]. In this model, players attempt to connect to a common source node in a network by purchasing edges, and sharing their costs with other players. We introduce the *survivable* version of this game, where each player desires 2 edge-disjoint connections between her pair of nodes instead of just a single connecting path, and analyze the quality of exact and approximate Nash equilibria. This version is significantly different from the original Connection Game and have more complications than the existing literature on arbitrary cost-sharing games since we consider the formation of networks that involve many cycles.

For the special case where each node represents a player, we show that Nash equilibria are guaranteed to exist and price of stability is 1, i.e., there always exists a stable solution that is as good as the centralized optimum. For the general version of the Survivable Connection Game, we show that there always exists a 2-approximate Nash equilibrium that is as good as the centralized optimum. To obtain the result, we use an approximation algorithm technique that compares the strategy of each player with only a carefully selected subset of her strategy space. Furthermore, if a player is only allowed to deviate by changing the payments on one of her connection paths at a time, instead of both of them at once, we prove that the price of stability is 1. We also discuss the time complexity issues.

1 Introduction

Network design is a fundamental problem for which it is important to understand the effects of strategic behavior. To accomplish this, algorithmic game theory has become a major tool for studying networks such as the Internet, which are developed, built, and maintained by a large number of independent agents, all of whom act in their own interests.

Elliot Anshelevich
Rensselaer Polytechnic Institute, Troy, NY.
Tel.: +1-518-2766491
Fax: +1-518-2764033
E-mail: eanshel@cs.rpi.edu

Bugra Caskurlu
Rensselaer Polytechnic Institute, Troy, NY.
Tel.: +1-518-2768988
Fax: +1-518-2764033
E-mail: caskub@cs.rpi.edu

In such networks, the global performance of the system may not be as good as in the case where a central authority can simply dictate a solution; rather, we need to understand the quality of solutions that are consistent with self-interested behavior. Much research in the theoretical computer science community has focused on this performance gap and specifically on the notions of the *price of anarchy* and the *price of stability* — the ratios between the costs of the worst and best Nash equilibrium[1], respectively, and that of the globally optimal solution. Both of these notions are important since as the "stable" points in a game, the Nash equilibria are often the only viable outcomes of agent interactions. We will only consider pure (i.e., deterministic) Nash equilibria, as mixed strategies do not make as much sense in our context.

Connection Game In this paper, we consider the price of stability of several important extensions of the *Connection Game*, which was first defined in [5], and later studied in a variety of papers including [3, 9, 11, 15, 18, 19]. This game represents a general framework where a network is being built by many different agents/players who have different connectivity requirements, but can combine their money to pay for some part of the network. The Connection Game models not only communication networks, but also many kinds of transportation networks that are built and maintained by competing interests. Specifically, each player in this game has some connectivity requirements in a graph $G = (V, E)$, i.e, she desires to connect a particular pair of nodes in this graph. With this as their goal, players can offer payments indicating how much they will contribute towards the purchase of each edge in G. If the players' payments for a particular edge e sum to at least the cost of e, then the edge is considered *bought*, which means that e is added to our network and can now be used to satisfy the connectivity requirements of any player.

Survivable Network Design One of the most important extensions of the Steiner Forest problem is Survivable Network Design (sometimes called the Generalized Steiner Forest problem), and for good reasons. In this problem, we must not simply connect all the desired pairs of terminals, but instead connect them using r edge-disjoint paths. This is generally needed so that in the case of a few edge failures, all the desired terminals still remain connected. Many nice results have been shown for finding the cheapest survivable network, including Jain's 2-approximation algorithm [22].

In this paper, we consider the *Survivable Connection Game,* where each agent/player wishes to connect to her destination using $r = 2$ edge-disjoint paths. The optimal (i.e., cheapest) centralized solution for this game is the optimal solution to Survivable Network Design, which we denote by OPT. Our goals are to understand the quality of exact and approximate Nash equilibria by comparing them to OPT, and thereby understand the efficiency gap that results because of the agents' self-interest. By studying the price of stability, we also seek to reduce this gap, as the best Nash equilibrium can be thought of as the best outcome possible if we were able to suggest a solution to all the players simultaneously.

Our Results We only consider the case where all players are attempting to connect to a single common source. For the single source version of the Connection Game, [5] proved that the price of stability is 1, and that in particular, a pure Nash equilibrium always exists. This is no longer true if the players have arbitrary connection requirements, as a pure Nash equilibrium is no longer guaranteed to exist. Figure 1(A) shows a game where this is the case. In Figure 1(A), everyone wishes to connect to s, and the edge costs are as shown. There is a player for each node v_i that wishes to connect v_i to s using a single path, and there is a player that wishes to connect u to s using 2 disjoint paths. It is easy to see that if the bottom path is entirely bought in a Nash equilibrium, then the v_i-players do not contribute to any edges, since they have 2 disjoint paths connecting them to s, when they only desire a single one, and they would be able to remove a

[1] Recall that a (pure-strategy) Nash equilibrium is a solution where no single player can switch her strategy and become better off, given that the other players keep their strategies fixed.

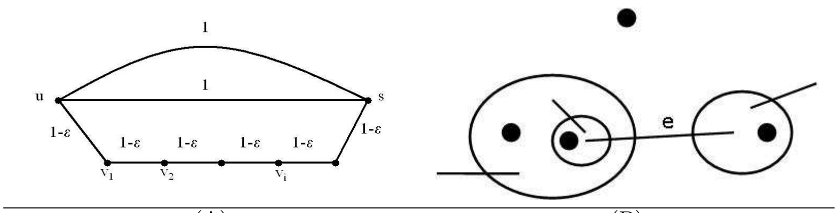


Fig. 1 (A) An instance with no pure Nash equilibrium. (B) Shows various witness sets of the directed arcs of an edge e.

payment to any edge without disconnecting themselves. The u-player would never pay more than 2 in total, however, so the bottom path cannot be fully bought in any equilibrium solution. This means that in any equilibrium solution, the u-player is using the top two paths (and is paying for them entirely, since no one else has a reason to help given that they only desire a single path), and is not paying anything for the bottom path (since the u-player is not using it). All but a single edge of the bottom path must be bought in any solution, however (to ensure the connectivity for the v_i-players), so the u-player has an incentive to switch from paying 1 for one of the top paths to buying the last edge of the bottom path for $1 - \varepsilon$. Similar examples show that a Nash equilibrium is not guaranteed to exist even for 2-player games, and that there may not exist an α-approximate[2] Nash equilibrium for any $\alpha < 2$.

The example in Figure 1(A) shows that adding stronger connectivity requirements to the Connection Game significantly changes it, since the prices of anarchy and stability become infinite. Instead of considering arbitrary connection requirements, therefore, we restrict our attention to the case where all terminals desire to connect using 2 disjoint paths. In this case, we prove results that are similar to the properties of the original Connection Game. Specifically, our main results are as follows:

– In the special case where all nodes are terminal nodes (i.e., there exists a player that desires to connect this node to the source), there always exists a Nash equilibrium that is as good as OPT.
– For the general Survivable Connection Game, there exists a 2-approximate Nash equilibrium that is as good as OPT.
– For the Survivable Connection Game, there is a polynomial time algorithm which finds a cheap $(2 + \epsilon)$-approximate Nash equilibrium.

Our approach for forming equilibrium payments is similar to [5], in that we show that either every edge can be fully bought using our equilibrium payment scheme, or that OPT is not actually an optimal solution. The fact that OPT is no longer a tree, however, significantly complicates matters, forcing the payment scheme to be a bit more clever and requiring different proof techniques. In order to prove the results:

– We consider a version of the Survivable Connection Game where each player is only allowed to deviate by changing the payments on a single path and show that for that version, there always exists a stable solution that is as good as the centralized optimal solution, i.e, the price of stability is 1. We obtain our grand result by proving that any stable solution of this version corresponds to a 2-approximate Nash equilibrium of the general Survivable Connection Game.
– We prove strong results about the almost laminar structure of survivable networks, which are stated in Section 2. These results are of independent interest, and give useful techniques for dealing with survivable networks in both game theoretic and more traditional contexts.

[2] An α-approximate Nash equilibrium is a solution where no player can save more than a factor of α by deviating.

Related Work Over the last few years, there have been several new papers about the Connection Game, e.g., [3,9,11,15,18,19]. Recently, Hoefer [17] proved some interesting results for a generalization of the game in [5], and showed an interesting relationship between the Connection Game and Facility Location. While the survivable network design games that we consider can be expressed as part of the framework in [17], these results do not imply ours, and our results cannot be obtained using their techniques. The state of the art for the usual (non-survivable) Connection Game is currently as follows [5]: (i) With a common source and every node being a terminal, a Nash equilibrium exists as good as OPT, and can be computed in polynomial time. (ii) With a common source but not every node being a terminal, a Nash equilibrium exists as good as OPT, and a $(1+\epsilon)$-approximate Nash equilibrium as good as OPT can be computed in polynomial time. (iii) Without a common source a 3-approximate Nash equilibrium exists (an exact one may not exist), and a $(3+\epsilon)$-approximate Nash equilibrium as good as OPT can be computed efficiently when the connectivity requirements of every player simply consist of connecting an arbitrary pair of nodes. Stronger bounds are known for tree games [18] and for geometric graphs [19].

The research on non-cooperative network design and formation games is too much to survey here, see [21,24,28,30,31] and the references therein. Fabrikant et al. [12] (see also [1]) studied the price of anarchy of a very different network design game, and [6] considered the price of stability of a network design game with local interactions, intended to model the contracts made by Autonomous Systems in the Internet.

A major part of the research on network games has focused on congestion games [4,8,10,27,30]. Probably the most relevant such model to our research is presented in [4] (and further addressed in [8,9,15]). In [4], extra restrictions of "fair sharing" are added to the Connection Game, making it a congestion game and thereby guaranteeing some nice properties, like the existence of Nash equilibria and a bounded price of stability. While the Connection Game is not a congestion game, and is not guaranteed to have a Nash equilibrium, it actually behaves much better than [4] when all the agents are trying to connect to a single common node. Specifically, the price of stability in that case is 1, while the best known bound for the model in [4] is $\frac{\log n}{\log \log n}$ [2]. Moreover, all such models (including cost-sharing models described below) restrict the interactions of the agents to improve the quality of the outcomes, by forcing them to share the costs of edges in a particular way. This does not address the contexts when we are not allowed to place such restrictions on the agents, as would be the case when the agents are building the network together without some overseeing authority. However, as [5] has shown for the Connection Game and we show for the Survivable version of it, it is still possible to nudge the agents into an extremely good outcome without restricting their behavior in any way.

The questions we consider bear some similarity to cost-sharing mechanisms for network design, such as [14,16,23,25]. Unlike non-cooperative games that we consider, cost-sharing games assume that there is a central authority that designs and maintains the network, and decides appropriate cost-shares for each agent, depending on the graph and all other agents, via a complex algorithm. The agents' only role is to report their utility for being included in the network. In contrast, in our game the agents contribute to individual edges directly, rather than contributing money to a central authority. Therefore, each player is free to choose her own paths, instead of having the central authority specify paths for each player. In our game there is no central authority designing the Steiner tree or cost shares. Rather, we study Nash equilibria of our game. In essence, cost-sharing games are appropriate when the network is being designed by a central authority, and the Connection Game is appropriate when the players can choose their own paths and edge contributions.

2 The Model and Preliminaries

We now formally define the Survivable Connection Game for N players. Let an undirected graph $G = (V, E)$ be given, with each edge e having a nonnegative cost $c(e)$, and let $s \in V$ be a special *root* (or source) node. Each player i has a single terminal node (also called *player node*) that she must connect to s using 2 edge-disjoint paths. The terminals of different players do not have to be distinct.

A strategy of a player is a payment function p_i, where $p_i(e)$ is how much player i is offering to contribute to the cost of edge e. Observe that players can share the cost of the edges. An edge e is considered *bought* if $\sum_i p_i(e) \geq c(e)$. Let G_p denote the subgraph of bought edges corresponding to the strategy vector $p = (p_1, \ldots, p_N)$. In the Survivable Connection Game, each player i wants to have 2 edge-disjoint paths between i and s, i.e., each player i requires that $M(G_p, i, s) \geq 2$, where $M(G_p, i, s)$ denotes the size of the min-cut between the nodes i and s on the graph G_p. While required to connect her terminals using at least 2 edge-disjoint connections, each player also tries to minimize her total payments, $\sum_{e \in E} p_i(e)$ (which we will denote by $|p_i|$). We conclude the definition of our game by defining the cost function for each player i as:

- $cost(i) = \infty$ if $M(G_p, i, s) < 2$
- $cost(i) = |p_i|$ otherwise.

The rest of this paper is mainly devoted to proving that there exist exact or approximate Nash equilibrium solutions that cost as much as OPT. To prove our results we give algorithms that either form Nash equilibrium strategies for the players that buy OPT, or give us a solution G_p that is cheaper than OPT and satisfies the connectivity requirements of all the players, i.e., $M(G_p, i, s) \geq 2 \forall i$, which contradicts with OPT being the socially optimal solution. We make use of the following observations in our arguments.

OPT and Nash Equilibrium OPT, by definition, is the cheapest network that satisfies the connection requirements of all the players. Therefore, for every edge e of OPT, there is a set of players whose connection requirements will be dissatisfied if e is deleted from OPT, since otherwise a cheaper network $OPT - e$ that satisfies the connectivity requirements of all the players can be obtained by simply deleting e from OPT. Note that in a Nash equilibrium, only this set of players can make payments on e, since all other players will deviate by setting their payment on e to 0 if they had a nonzero payment for e. The players in this set are therefore said to *witness* e, since without them, e would not be needed.

Witness Sets Let i be a player that witnesses $e = (j, k)$, i.e., i will have only 1 path to s if e is deleted. Observe that $M(OPT, i, s) = 2$ and $M(OPT - e, i, s) = 1$. Therefore, there is a cut C in OPT between i and s of size 2 such that e is one of the cut edges. Without loss of generality, assume j is on the same side of C as i. Since G is an undirected graph, each edge $e = (j, k)$ can be thought of as being composed of two directed arcs (j, k) and (k, j), with the connectivity requirement of a node i meaning that it desires to have 2 edge-disjoint directed paths to s. Notice that all the players in the set of nodes on the same side of C as i have to use arc (j, k) to satisfy their connectivity requirements. We call such a set of nodes a *witness set* of arc (j, k). The 2 arcs directed to outside of the set are called the boundary arcs of the witness set. Note that since every edge in OPT has necessarily a witnessing player, at least one of its corresponding directed arcs has a witness set, which can be constructed by the cut argument above. Figure 1(B) shows various witness sets of the directed arcs of an edge e. The black circles represent the player nodes.

Definition 1 A *witness set* of an arc (j, k) is a set of nodes including at least one player node and excluding s, with exactly 2 boundary arcs, one of which is (j, k).

Smallest Witness Sets Observe that any player in a witness set of (j,k) witnesses (j,k), and any player witnessing (j,k) has to be involved in some witness set of (j,k). Intuitively, a player inside a witness set must use both of the arcs leaving it, since the witness set is a cut of size 2 and she needs 2 edge-disjoint paths. Among the sets witnessing (j,k), the smallest one in terms of the number of nodes included is called a *smallest witness set* of (j,k) and we denote it as $W(j,k)$. The smallest witness set for the reverse arc (k,j) is denoted as $W(k,j)$. If an arc (j,k) is witnessed, then the smallest witness set of (j,k), $W(j,k)$ exists and it is unique as proven by Lemma 1, the proof of which appears in Section 5.1. Observe that at least one of the arcs of every edge e of OPT has a smallest witness set. In the Nash equilibrium solutions formed by our algorithms, the cost of each edge e of OPT will be shared among the players that are in a smallest witness set of the arcs of e. In the rest of the paper, we will use W to denote the set of all smallest witness sets of all the arcs of OPT.

Lemma 1 *If an arc (j,k) is witnessed, then the smallest witness set of (j,k), $W(j,k)$, is unique.*

Laminarity A set of sets W is called a *laminar* set system if for any 2 elements of the set system $W_1, W_2 \in W$ either W_1 and W_2 are disjoint or one of them is a subset of the other. To see that the set of smallest witness sets is not necessarily laminar, consider a cycle s, t_1, v, t_2 with t_1 and t_2 being terminals and v being a non-terminal. In this example, $W(v,t_1) = \{v, t_2\}$, and $W(v,t_2) = \{v, t_1\}$. These two sets have a non-trivial intersection, and so are not laminar. However, if we contract the edges (t_1, v) and (v, t_2) by removing node v, then the witness set system of the new graph becomes laminar. Theorem 1, proof of which appears in Section 5 proves that this is essentially the only way that smallest witness sets may have a non-trivial intersection, and contracting paths composed of degree 2 non-player nodes makes the set of smallest witness sets become laminar. We say that the set of smallest witness sets W of a solution G_S is *laminar with path exceptions* if the set of smallest witness sets of G_S is laminar after contracting paths composed of degree 2 non-player nodes.

Theorem 1 *Let G^* be a graph obtained by replacing each path P of OPT composed of degree 2 non-player nodes by a single edge. The set of smallest witness sets of G^* is a laminar set system, i.e., the set of smallest witness sets of OPT is laminar with path exceptions.*

3 When All Nodes Are Terminals

For the Survivable Connection Game, we do not know whether there exists an exact Nash equilibrium for all possible instances of the problem. However, for the special case where each node of G is a player-node, a Nash equilibrium is guaranteed to exist. Specifically, there is a Nash equilibrium whose cost is as much as OPT, and therefore price of stability is 1. In this section, we will prove this result by forming stable payments on the edges of OPT.

To form the payments, we will give an algorithm that decides how the cost of each edge of OPT is shared among the players. For each edge e of OPT, our algorithm only asks the adjacent terminals to contribute to the cost of e. Recall that since we are trying to form a Nash equilibrium, each terminal can only contribute to the cost of an edge e only if it witnesses the outgoing arc of e. Though a terminal can have arbitrary number of outgoing arcs, it witnesses at most 2 of them. To see this, assume there is a player that witnesses more than 2 of its incident arcs. Then at least one of the connection paths of this terminal is using at least 2 of its incident arcs by the pigeonhole principle, which implies this connection path contains a cycle. Since that terminal is still 2-connected after removal of the cycle, it does not witness the incident arcs included in the cycle. That simple observation allows us to see a nice substructure in OPT, which we call *chains*.

A *chain* is a path with maximal length in OPT, where both of the arcs of each edge of the path are witnessed. Observe that each intermediate node of the chain is witnessing both of its outgoing arcs in the chain, since all the arcs of the chain are witnessed. Since a terminal can witness at most 2 of its incident arcs, no intermediate node of the chain witnesses any incident arcs except the ones in the chain. The boundary nodes of the chain are witnessing the arc of the chain they are adjacent to. Observe that boundary nodes of the chain may or may not witness any other arcs but if they do witness an arc (j, k), the reverse arc (k, j) does not have a witness set, since otherwise the edge (j, k) would have been part of the chain as well.

Observe that every edge e such that both of the arcs of e have smallest witness sets, is included in some chain. In the simplest case, where both of the adjacent nodes of e does not witness any other incident arcs or witness one other arc whose reverse arc does not have a witness set, we will have a chain that includes only one edge, namely e. Therefore, OPT is composed of chains and edges that has only one arc witnessed. To form the stable solution, we first form the payment on the edges that has only one arc witnessed, and then form the payments on the edges of the chains.

Since we are trying to form a Nash equilibrium, no player i should have an incentive of unilateral deviation from her strategy p_i when the algorithm terminates, i.e., $|p_i|$ should not be more than the cost of the best deviation of player i. Observe that a best deviation of player i, which we denote by $\chi_i(p_{-i})$, is the cheapest strategy of player i that satisfies her connection requirement given the strategies p_{-i} of other players. While such a deviation may not be unique, for our purposes it is enough to let $\chi_i(p_{-i})$ denote an arbitrary best response of player i to strategy p_{-i}. Since $\chi_i(p_{-i})$ is the strategy of i with smallest cost such that the payments $(p_{-i}, \chi_i(p_{-i}))$ buy 2 edge-disjoint paths from i to s, we will call the union of these two paths a "connection cycle of $\chi_i(p_{-i})$". We will show that $p_i + p_{-i}$ buys all the edges of OPT when our payment algorithm terminates, and that those payments form a Nash equilibrium.

Notation and Invariant Let p^* denote the cheapest strategy vector that buys all the edges of OPT, i.e., $p^*(e) = c(e)$ if e is in OPT and $p^*(e) = 0$ otherwise. Let $\overline{p_i}$ denote the minimum payment to be made by other players to buy all the edges of OPT given that player i plays the strategy p_i, i.e., $\overline{p_i} = p^* - p_i$. To have an easier analysis we want our algorithm to have a stronger property: we not only want it to ensure stability at termination but also at each intermediate step. In other words, at any step of the algorithm, the inequality $|p_i| \leq |\chi_i(\overline{p_i})|$ will hold, i.e., the payment strategy p_i assigned to i should be the cheapest strategy of i that satisfies her connectivity requirement, assuming the rest of the payments to buy all the edges of OPT are made by other players. Note that if all the edges of OPT are bought, i.e., $p_i + p_{-i} = p^*$, then $p_{-i} = \overline{p_i}$ and the invariant $|p_i| \leq |\chi_i(\overline{p_i})|$ turns into the Nash equilibrium condition. To show that our algorithm produces a Nash equilibrium as cheap as OPT, it is thus enough to prove the following two statements:

- The invariant $|p_i| \leq |\chi_i(\overline{p_i})|$ holds at every step of our algorithm for all players i.
- When the algorithm terminates, all the edges of OPT are bought by the players.

Proof Overview

1. We define the algorithm that constructs payments for the edges of OPT. By construction, it will be clear that this algorithm always satisfies the invariant above. Thus, if it succeeds in paying for all edges of OPT, then the payments form a Nash equilibrium.
2. In Theorem 2, we prove that this algorithm always terminates, and thus pays for all edges of OPT.

Computing deviations Here we discuss how deviations can be computed. When computing $\chi_i(\overline{p_i})$, note that all edges of OPT such that i is not contributing any payment to them can be used by i freely to satisfy her connectivity requirement. Therefore, when computing the cheapest deviation for a player i, we should not use the actual cost of the edges in G, but instead for each edge f, we should use the cost i would face if she is to use f, which will be referred to as *modified cost of f for i,* and denoted by $c'_i(f)$ in the rest of the paper. Specifically, for f not in OPT, $c'_i(f) = c(f)$, the

```
Input: The socially optimal solution OPT
Output: The payment scheme for
OPT that is a Nash equilibrium
Initialize p_i(e) = 0 for all players i and edges e;
Let d(e) denote the amount of payment i should make in order to buy e, i.e.,
d(e) = c(e) − Σ_{j≠i} p_j(e);
Loop through all edges e = (i, j) of OPT such that W(i, j) exists but
W(j, i) does not exist;
    If |χ_i(p̄_i, e)| − |p_i| ≥ d(e)
        Set p_i(e) = d(e);
    Else break;
Loop through all chains C of OPT;
    Let e_1 = (n_1, n_2), e_2 = (n_2, n_3), ..., e_k = (n_k, n_{k+1}) be the edges of the chain C;
    For each edge e = (i, j) of C from e_1 to e_k;
        Set p_i(e) = min{|χ_i(p̄_i, e)| − |p_i|, d(e)};
        If |χ_j(p̄_j, e)| − |p_j| ≥ d(e)
            Set p_j(e) = d(e);
        Else break;
```

Algorithm 1: Algorithm that generates payments on the edges of OPT

actual cost of f. For the edges f of OPT that i has not contributed anything to (i.e., $p_i(f) = 0$), we have that $c'_i(f) = 0$, since from i's perspective, she can use these edges for free because other players have paid for them. For all the other edges f that i is paying $p_i(f)$ for, $c'_i(f) = p_i(f)$, since that is how much it costs for i to use f in her deviation from the payment strategy p_i. Using these modified costs, $\chi_i(\overline{p_i})$ is simply the cheapest set of edges which fulfill player i's connectivity requirements.

We now present an algorithm which satisfies the two properties mentioned above (the pseudocode is shown in Algorithm 1). At the beginning of the algorithm $|p_i| = 0$ for all players i, and therefore the invariant is trivially satisfied by all the players. The algorithm first loops through the edges e of OPT such that exactly one of the corresponding arcs of e is witnessed. Let $e = (i, j)$ be one of those edges and without loss of generality assume $W(i, j)$ exists and $W(j, i)$ does not exist. Then the algorithm asks i to make a payment for e. Recall that the algorithm should never assign a payment for a player that violates the invariant $|p_i| \leq |\chi_i(\overline{p_i})|$. Let x denote the maximum amount of payment player i can make for e in order not to violate the invariant. If $x \geq c(e)$ then the algorithm asks i to pay $c(e)$ for e and break otherwise, and therefore the invariant is never violated throughout the first loop of the algorithm. In the second loop, the algorithm forms the payment on the edges of the chains. The algorithm loops through all the chains C of OPT. Let $e_1 = (n_1, n_2), e_2 = (n_2, n_3), \ldots, e_k = (n_k, n_{k+1})$ be the edges of a chain C. To form the payment on the edges of C, the algorithm loops through all the edges of C starting from the leftmost edge e_1 till the rightmost edge e_k. So the payment for e_i is decided after the payments for $e_1, e_2, \ldots, e_{i-1}$ are already decided. To form the payment for e_i, the algorithm asks n_i to make the maximum amount of payment x player n_i can make in order not to violate the invariant (or to pay as much as c(e), whichever is cheaper). The algorithm then asks n_{i+1} to pay for the rest of the cost of e_i if it will not violate the invariant. Therefore the invariant is never violated throughout the second loop of the algorithm as well.

What is this value x, however, and how to we compute it? For a strategy p_i of player i, let $\chi_i(\overline{p_i}, e)$ denote the cheapest deviation of player i from the strategy p_i that does not use the edge e, assuming that the rest of the players are paying for $\overline{p_i}$. Observe that $|\chi_i(\overline{p_i}, e)| \geq |\chi_i(\overline{p_i})|$. Then we argue below that $x \geq \min\{|\chi_i(\overline{p_i}, e)| - |p_i|, d(e)\}$. To see this, we consider the strategy $p_i + x$ where player i pays x for edge e. We are abusing notation here, since x is a number, not a payment vector. Formally, by $p_i + x$ we will mean the payment strategy which equals p_i everywhere except at e, with $(p_i + x)(e) = p_i(e) + x$.

Lemma 2 *Given payments p_i which do not violate the invariant, player i can increase her payments on edge e by $\min\{|\chi_i(\overline{p_i}, e)| - |p_i|, d(e)\}$ and not violate the invariant.*

Proof. To see this, suppose that x is the maximum amount that i can pay for edge e without violating the invariant, and $x < d(e)$. Now suppose to the contrary that $x < |\chi_i(\overline{p_i}, e)| - |p_i|$. This means that $|p_i + x| = |p_i| + x < |\chi_i(\overline{p_i}, e)|$. By Lemma 3, we know that $|\chi_i(\overline{p_i}, e)| = |\chi_i(\overline{p_i + x})|$. Therefore, $|p_i + x| < |\chi_i(\overline{p_i + x})|$, and so we can increase x and still not violate the invariant. This gives us a contradiction with x being maximum. ■

Lemma 3 *Let p_i denote the strategy of player i right before she is asked to make a payment for e and let $x < d(e)$ be the maximum amount of payment i can make for e without violating the invariant. Then, $|\chi_i(\overline{p_i}, e)| = |\chi_i(\overline{p_i + x})|$, i.e, at least one of the best deviations of player i right after she makes a payment of x for e does not use e.*

Proof. For the purpose of contradiction, assume that all of the cheapest deviations of player i after she pays x for e, i.e., $\chi_i(\overline{p_i + x})$, uses e. Let $\mu = |\chi_i(\overline{p_i + x}, e)| > |\chi_i(\overline{p_i + x})|$ be the cost of the cheapest deviation of player i that does not use e right after i pays x for e. Observe that if the payment of i for e is increased by $y = \min\{\mu - |\chi_i(\overline{p_i + x})|, d(e) - x\}$, the cost of all deviations of player i that use edge e increases by the same amount. The invariant is still not violated after this increase, which contradicts with the assumption that x is the maximum amount of payment she can make for e without violating the invariant. ■

Since the invariant is not violated at any step of the algorithm, the algorithm finds a Nash equilibrium whose cost is as much as OPT, if it does not break at any of the intermediate stages. To prove our result all we need to do is to prove that the algorithm never breaks at an intermediate stage. We will prove this by constructing a feasible network cheaper than OPT whenever the algorithm breaks, which will contradict the optimality of OPT.

Specifically, we will consider networks formed by players' deviations. Recall that by Lemma 3, when a player i cannot pay $d(e)$ but some amount $x < d(e)$ for an incident edge e without violating the invariant, she does have a deviation that costs as much as her strategy $p_i + x$ and does not use e. Define $X_i(p_i, e)$ to be the graph formed by removing the edges paid for by $p_i + x$ from OPT, as well as e, and then adding the edges paid for in $\chi_i(\overline{p_i}, e)$. In other words, $X_i(p_i, e)$ is the network of bought edges formed if player i deviates from her current strategy $p_i + x$ to $\chi_i(\overline{p_i}, e)$, with the payments of the other players being $\overline{p_i + x}$. The edges added to OPT by this deviation cost at most $|p_i + x|$, and the edges removed cost strictly greater than $|p_i + x|$. The cost is *strictly* greater because e is one of the edges removed, and player i does not fully pay for edge e in the payment $p_i + x$. Therefore, we know that the graph $X_i(p_i, e)$ is strictly cheaper than OPT. Since $\chi_i(\overline{p_i}, e)$ is a best response of player i, she has 2 edge-disjoint paths from i to s. Therefore, there exists a cycle in $X_i(p_i, e)$ containing both i and s, which we call the *connection cycle of* i in $X_i(p_i, e)$. To show feasibility of $X_i(p_i, e)$, we have to show that not only the deviating player but also all other players have 2 edge-disjoint paths to s. To do this, we often use the following easy lemma.

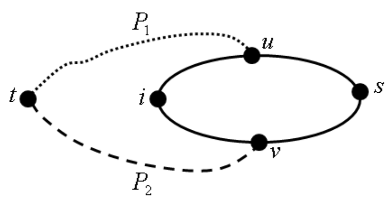


Fig. 2 Shows the connection paths of a player t that has 2 edge-disjoint paths to the connection cycle of player i.

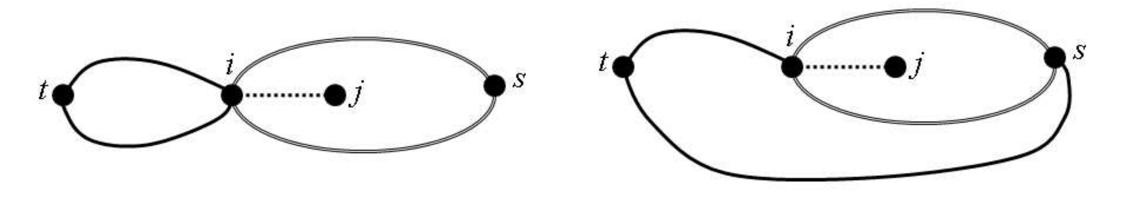


Fig. 3 A player t witnessing the arc (i, j) in OPT (A) either has 2 edge-disjoint paths to i, or (B) 1 path to i and 1 path to s, which are mutually edge-disjoint.

Lemma 4 *Let C_i be a connection cycle of a player i, i.e., a cycle containing nodes i and s. If a player t has 2 edge-disjoint paths to C_i, then t has 2 edge-disjoint paths to s as well. Moreover, if players i and j each have connection cycles C_i and C_j, and t has 2 edge-disjoint paths: one ending at a node of C_i and one at a node of C_j, then t also have a connection cycle to s.*

Proof. Assume there are 2 edge-disjoint paths P_1 and P_2 from t to C_i, the connection cycle of player i, as depicted in Figure 2. Let u and v be the nodes of C_i, where P_1 and P_2 reaches C_i respectively. Since all u, v, and s are nodes of the cycle C_i, there is a path from u to v on C_i that includes s and let this path be called P_3. Since P_1 and P_2 does not contain any node of C_i other than u and v respectively, P_3 is necessarily disjoint from P_1 and P_2. Therefore $P_1 \cup P_2 \cup P_3$ is a cycle containing both t and s and constitutes 2 edge-disjoint paths from t to s. Thus, the first part of the lemma holds.

To prove the second part of the lemma, suppose to the contrary that i and j have connection cycles to s, but t does not. This means that there is a cut $(S, V - S)$ with $t \in S$, $s \in V - S$, and with less than 2 edges crossing this cut. S cannot contain any nodes of C_i or C_j, since then it will have two edges of C_i or C_j leaving it. Therefore, C_i and C_j are fully contained in $V - S$, and so the two disjoint paths from t to C_i and C_j must cross the cut, yielding a contradiction with the cut having less than 2 edges. ■

Now we are ready to prove our exact Nash equilibrium result.

Theorem 2 *The payment algorithm does not break in any intermediate stages, and therefore the payments form a Nash equilibrium. Since there is a Nash equilibrium whose cost is as much as OPT, price of stability is 1.*

Proof. Let us first prove that the payment algorithm does not break while executing the first loop. For the purpose of contradiction, assume the contrary, i.e., the payment algorithm breaks while deciding the payment on an edge $e = (i, j)$ such that $W(i, j)$ exists but $W(j, i)$ does not exist. Notice that $d(e) = c(e)$ at the time when the algorithm breaks, since i is the only player that is asked to pay for the cost of e. Since the algorithm breaks, player i has a deviation $\chi_i(\overline{p_i}, e)$ such that $|\chi_i(\overline{p_i}, e)| < |p_i| + c(e)$. Recall that $X_i(p_i, e)$ is the graph of bought edges for the payment vector $p^* - p_i + \chi_i(\overline{p_i}, e)$, but with edge e also removed. In other words, $X_i(p_i, e)$ is the network that would be bought if player i unilaterally deviates her strategy from p_i to her best response $\chi_i(\overline{p_i}, e)$, with all other players paying for $\overline{p_i} - e$. Notice that the cost of the network $X_i(p_i, e)$ is $c(OPT) - |p_i| - c(e) + |\chi_i(\overline{p_i}, e)|$. Since $|\chi_i(\overline{p_i}, e)| < |p_i| + c(e)$, then $X_i(p_i, e)$ is strictly cheaper than OPT.

In order to obtain a contradiction, we will show that all the players have 2 edge-disjoint paths to s on network $X_i(p_i, e)$, which will be a contradiction since in that case it is a feasible network cheaper than OPT. First consider the case where e is the only edge that player i has contributed to at the time the algorithm brakes. Then, e is the only edge of G that is in OPT but not in $X_i(p_i, e)$. i has 2 edge-disjoint paths to s in $X_i(p_i, e)$, since $\chi_i(\overline{p_i}, e)$ is a best response of player i; let C_i be a

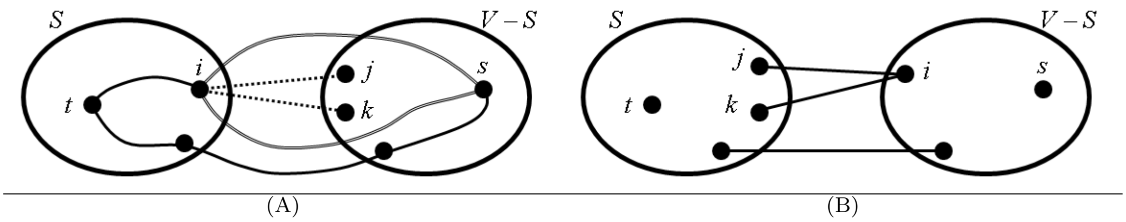


Fig. 4 (A) Explains the structure of the connection paths of t for the case $i \in S$, (B) Illustration of why it cannot be the case $i \notin S$.

connection cycle of player i in $X_i(p_i, e)$. For the purpose of contradiction, suppose there is a player t, which does not have 2 edge-disjoint paths in $X_i(p_i, e)$. Then t is one of the players that witness the arc (i, j) in OPT since the arc (j, i) is not witnessed in OPT. This implies that there are two edge-disjoint paths in OPT from t to s, with one of these paths using the arc (i, j). Since (i, j) is the only edge in OPT but not in $X_i(p_i, e)$, this implies that there are either two edge-disjoint paths from t to i in $X_i(p_i, e)$ (see Figure 3 (A)), or one path from t to i and one path from t to s (see Figure 3 (B)). In either case, Lemma 4 implies that t has 2 edge-disjoint paths to s in $X_i(p_i, e)$, since both i and s are in C_i. This gives us a contradiction for the case where i has contributed only to edge e at the time that the algorithm breaks.

Since a player can witness at most 2 of its incident edges, player i may be witnessing and therefore may have paid for one more adjacent edge, which we will call $f = (i, k)$. Recall that the payment on edge f is decided while the algorithm executes the first loop, if and only if $W(i, k)$ exists and $W(k, i)$ does not exist. Since the algorithm asks a player to pay for the cost of adjacent edges only, e and f are the only two edges that i may have contributed to. Therefore, e and f are the only two edges that are in OPT but not in $X_i(p_i, e)$. Suppose to the contrary that there do not exist 2 edge-disjoint paths between some player t and s in $X_i(p_i, e)$. Then, there is a cut $(S, V - S)$ of OPT with size at most 3 such that $t \in S$, $s \in V - S$, and e and f are two of the cut edges, since removing both e and f from OPT must result in a cut between t and s of size less than 2. If $i \in S$ (see Figure 4 (A)), then there must be 2 edge-disjoint paths from t to s in OPT such that at least one of them passes through i, simply because OPT is a feasible solution for t. Thus, by the same argument as above, t has 2 edge-disjoint paths to nodes of C_i, and thus still fulfills its connectivity requirement in $X_i(p_i, e)$ by Lemma 4, giving a contradiction. If, on the other hand, $i \notin S$ as depicted in Figure 4 (B), then we will argue that i cannot be witnessing both of the arcs (i, j) and (i, k). The fact that i is witnessing both of them means that one of her paths to s in OPT traverses the arc (i, j) and her other path to s traverses the arc (i, k). Therefore, both connection paths of i in OPT enter S through 2 boundary edges of S. However, there are at most 3 boundary edges of S, and $s \notin S$, so there is no way for both these paths to leave S in an edge-disjoint manner. Therefore, i could not have been feasible in OPT, which gives us a contradiction.

We have now shown that the algorithm does not break while executing the first loop. Now let us show that the algorithm will not break while executing the second loop. First, to help our understanding of chains, we prove the following lemma.

Lemma 5 *Let C be an arbitrary chain in OPT. Then all intermediate nodes of C have exactly* 2 *incident edges in OPT.*

Proof. Let i be an intermediate node of a chain C. By definition of chains, i has 2 incident edges in the chain and witnessing the corresponding outgoing arcs of both of her incident edges. For the purpose of contradiction, suppose that i has a third incident edge (i, j) in OPT as depicted in Figure 5. Since i does not witness the arc (i, j), i.e., $W(i, j)$ does not exist, then $W(j, i)$ must necessarily exist. Let C_j be a simple connection cycle of player j in OPT. Since j witnesses the arc (j, i), then the edge (i, j) and node i are contained in C_j. Since i witnesses the corresponding

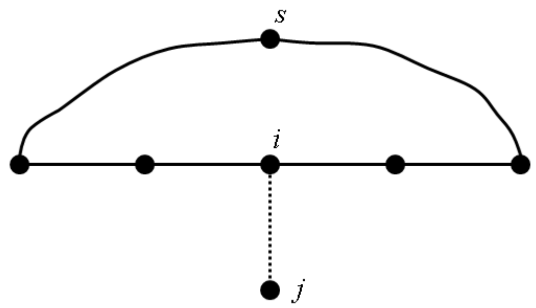


Fig. 5 An intermediate node i of a chain C has only 2 incident edges on OPT.

arcs of its 2 incident edges of the chain, any connection cycle containing i will also contain the 2 incident edges of i in C. Therefore, i has 3 incident edges in C_j. This is a contradiction since all nodes in a simple cycle have exactly 2 incident edges. ■

Let $e_1 = (n_1, n_2), e_2 = (n_2, n_3), \ldots, e_k = (n_k, n_{k+1})$ be the edges of a chain C and without loss of generality assume n_i and n_{i+1} could not pay for e_i, and this caused the algorithm to break. Consider a node n_j with $j \leq i$. Unless n_j agrees to pay for the entire edge e_j, it must have a deviation $\chi_{n_j}\left(\overline{p_{n_j}}, e_j\right)$ preventing it from contributing any more to e_j. Call this deviation χ_j. In addition, n_{i+1} has a deviation $\chi_{n_{i+1}}\left(\overline{p_{n_{i+1}}}, e_i\right)$ (which we will denote by χ_{i+1}) which is preventing it from contributing more to e_i. The fact that players n_i and n_{i+1} have deviations χ_i and χ_{i+1} preventing them from contributing more to edge e_i is what causes the algorithm to break.

In the rest of this proof, define the player n_β with $\beta \leq i$ to be the player that is closest to i such that one of the following holds: (1) χ_ℓ for some $\beta < \ell \leq i+1$ passes through n_β, (2) χ_β passes through $e_{\beta-1}$, or (3) n_β does not contribute anything to $e_{\beta-1}$ (i.e., $p_{n_\beta}(e_{\beta-1}) = 0$). If no node n_β satisfying one of the above conditions exists, we will say that we are in case (4), addressed separately in Lemma 8. We now form a solution OPT' as follows.

1. Start with OPT.
2. Add all edges contained in (i.e., bought by) $\chi_{\beta+1}, \ldots, \chi_i, \chi_{i+1}$.
3. If we are in the cases (2) or (3), also add all edges contained in χ_β.
4. Remove edges $e_\beta, e_{\beta+1}, \ldots, e_i$.

We now show that this network OPT' is still feasible, and is cheaper than OPT, thus giving us a contradiction.

Lemma 6 *OPT' is strictly cheaper than OPT.*

Proof. Let p_j denote the payments p_{n_j} of node n_j when the algorithm breaks. By the invariant of our algorithm we know that $|p_j| \leq |\chi_j|$. Moreover, for any $\beta \leq j < i+1$, we now show that $|p_j| = |\chi_j|$. Suppose to the contrary, that $|p_j| < |\chi_j|$. The only way this is possible in our algorithm is if n_j pays entirely for edge e_j, i.e., $p_j(e_j) = c(e_j)$. But in this case, n_{j+1} contributes nothing to edge e_j, and so n_{j+1} satisfies Condition (3), contradicting the maximality of β. Thus, for $\beta \leq j < i+1$, we have that $|p_j| = |\chi_j|$.

We now prove the lemma. First consider the case where n_β satisfies Condition (1). Then,

$$c(OPT') \leq c(OPT) + \sum_{j=\beta+1}^{i+1} |\chi_j| - \sum_{j=\beta}^{i} c(e_j),$$

since we obtain OPT' by first adding the edges bought by $\chi_{\beta+1}, \ldots, \chi_i, \chi_{i+1}$, and then removing edges $e_\beta, e_{\beta+1}, \ldots, e_i$ from OPT. Thus, to show that $c(OPT') < c(OPT)$, we must prove that

$$\sum_{j=\beta+1}^{i+1} |\chi_j| < \sum_{j=\beta}^{i} c(e_j). \tag{1}$$

Recall that $|\chi_{i+1}| < c(e_i) - p_i(e_i)$, since the algorithm broke when deciding the payment of e_i. Thus we know that

$$\sum_{j=\beta+1}^{i+1} |\chi_j| < \sum_{j=\beta+1}^{i} |p_j| + c(e_i) - p_i(e_i). \tag{2}$$

Since, by the construction of our algorithm, a payment p_j for $j > \beta$ only contributes to edges e_{j-1} and e_j, then

$$\sum_{j=\beta+1}^{i} |p_j| + p_\beta(e_\beta) + c(e_i) - p_i(e_i) \leq \sum_{j=\beta}^{i} c(e_j), \tag{3}$$

and so Inequality (1) holds, as desired.

Now we consider the case when n_β satisfies Condition (2). Recall that if n_β satisfies Condition (2), then the edges bought by χ_β are also added to OPT when constructing OPT'. Since χ_β passes through $e_{\beta-1}$ when Condition (2) holds, and thus pays $p_\beta(e_{\beta-1})$ for $e_{\beta-1}$, then the cost of the edges added to OPT' because of this is at most $|\chi_\beta| - p_\beta(e_{\beta-1})$. By the argument above, we know that $|\chi_\beta| = |p_\beta|$, and so the extra cost being added to OPT' in the case of Condition (2) is equal to $|p_\beta| - p_\beta(e_{\beta-1}) = p_\beta(e_\beta)$. Thus to prove that when Condition (2) holds, then $c(OPT') < c(OPT)$, we must prove that $\sum_{j=\beta+1}^{i+1} |\chi_j| + p_\beta(e_\beta) < \sum_{j=\beta}^{i} c(e_j)$. This is true once again by Inequalities (2) and (3).

Finally, if n_β satisfies Condition (3), then the edges bought by χ_β are also added to OPT when constructing OPT', similar to the above case. Thus, we must show that $\sum_{j=\beta}^{i+1} |\chi_j| < \sum_{j=\beta}^{i} c(e_j)$. Since Condition (3) means that $p_\beta(e_{\beta-1}) = 0$, then we know that $|p_\beta| = p_\beta(e_\beta)$, and since $|\chi_\beta| = |p_\beta|$, then we need to show that $\sum_{j=\beta+1}^{i+1} |\chi_j| + p_\beta(e_\beta) < \sum_{j=\beta}^{i} c(e_j)$ in order to prove that $c(OPT') < c(OPT)$. Once again, this holds by Inequalities (2) and (3). ■

We must now show that OPT' is a feasible solution. To do this, we must show that every player node t has a connection cycle to s in OPT'.

Lemma 7 *Nodes n_ℓ with $\ell \in \{\beta \ldots i+1\}$ each have a connection cycle to s in OPT'.*

Proof. We prove this for nodes n_ℓ with $\ell = \beta \ldots i+1$, inductively, starting with n_{i+1}. To show that n_{i+1} has a connection cycle in OPT', we only need to show that the connection cycle of χ_{i+1} does not use any edges $e_\beta, e_{j+1}, \ldots, e_i$. Since $\chi_{i+1} = \chi_{n_{i+1}}\left(\overline{p_{n_{i+1}}}, e_i\right)$, then by definition it does not use the edge e_i. If χ_{i+1} used some edge e_j with $\beta \leq j < i$, then node n_{j+1} would satisfy Condition (1) above, contradicting the maximality of β. Therefore, χ_{i+1} does not use any edges that are in OPT but not in OPT', and thus the connection cycle of χ_{i+1} still exists in OPT'.

Now assume that all nodes $n_{\ell+1} \ldots n_{i+1}$ have a connection cycle in OPT', and let us prove the same for n_ℓ with $\ell > \beta$. Consider the connection cycle of χ_ℓ: first we show that it does not use any edges e_j with $\beta \leq j < \ell$. If it uses an edge e_j with $j \leq \ell - 2$ then node n_{j+1} satisfies Condition (1) and we obtain a contradiction as above; if it uses edge $e_{\ell-1}$ then node n_ℓ satisfies Condition (2), once again contradicting the definition of β. Hence, the connection cycle of χ_ℓ does not use any edge to the left of n_ℓ in the chain that is removed from OPT. It also does not use edge e_ℓ, since χ_ℓ is defined as $\chi_{n_\ell}(\overline{p_{n_\ell}}, e_\ell)$, and does not use edge e_ℓ by definition. Consider the 2 edge-disjoint paths from n_ℓ to s in the connection cycle of χ_ℓ. In OPT', each of these paths either reaches s, or stops at a node in $\{n_{\ell+1} \ldots n_{i+1}\}$ since the next edge this path must follow has been removed from the graph. By the inductive hypothesis, each of these nodes has a connection cycle in OPT', and it is easy to see that so does n_ℓ by Lemma 3.

Finally, we show that the lemma holds for n_β. If Condition (1) holds, then node n_β is on the connection cycle of χ_ℓ for some $\beta < \ell \leq i+1$. This cycle cannot use edge e_β, since this would imply that node $n_{\beta+1}$ satisfies Condition (1), and contradicts the maximality of β. Call this cycle C_ℓ, and note that it provides 2 edge-disjoint paths from n_β to s. Trace both of these paths starting from n_β in OPT'. Each of these paths will either reach s using edges of OPT', or will stop at a node in $n_{\beta+1} \ldots n_{i+1}$ since the next edge this path must follow has been removed from the graph. By our inductive hypothesis, each such node has a connection cycle in OPT', and so using the same argument as above, we have that n_β has a connection cycle in OPT'. If Condition (2) or (3) hold, then we add the edges of the connection cycle of χ_β to OPT' in step 2. Call this cycle C_β. C_β does not use the edge e_β, since if it did, then n_β would pay for the entire cost of e_β when asked, which would mean that $n_{\beta+1}$ would satisfy Condition (3), and this would contradict maximality of β. Thus, we can trace the paths of C_β in OPT' and end at either s or at nodes $n_{\beta+1} \ldots n_{i+1}$, and use the same argument as above to conclude that n_β has a connection cycle in OPT'. ■

Now consider an arbitrary player t, and let C_t be its connection cycle in OPT. If C_t does not use edges of $e_\beta, e_{\beta+1}, \ldots, e_i$, then trivially t has a connection cycle in OPT' as well. Trace the paths of C_t starting at t in OPT'. Each of these paths either reaches s using edges of OPT', or stops early because the next edge in it does not exist in OPT'. Since the only edges removed from OPT are $e_\beta, e_{\beta+1}, \ldots, e_i$, and our chain is simply a path in OPT by Lemma 5, then these paths must stop at nodes n_β or n_{i+1}. By the above Lemma 7, both of these nodes have connection cycles in OPT', and thus so does t by Lemma 4. Since this means that OPT' is a feasible solution, then by Lemma 6, we have a contradiction.

We have now proven that if our algorithm breaks in its first loop, or in its second loop with Condition (1), (2), or (3) being satisfied, then there exists a feasible solution cheaper than OPT, and thus the algorithm cannot break in this manner. All that is left to prove is that the same holds if Condition (4) is satisfied instead.

Lemma 8 *If Condition (4) is satisfied, then there exists a feasible solution cheaper than OPT.*

Proof. Most of this proof is analogous to cases (1)-(3) above, with $n_\beta = n_1$. The details are as follows. Denote by f the edge paid for by n_1 that is only witnessed in one direction, if such an edge exists. We form a solution OPT' slightly differently for this case:

1. Start with OPT.
2. Add all edges contained in (i.e., bought by) $\chi_1, \ldots, \chi_i, \chi_{i+1}$.
3. Remove edges $e_1, e_2, \ldots, e_i$.
4. If player n_1 pays for a positive amount of edge f during the first loop of the algorithm (i.e., if $p_{n_1}(f) > 0$), and if the deviation χ_1 does not use the edge f, then remove edge f.

Showing that OPT' is strictly cheaper than OPT requires essentially the same arguments as Lemma 6. OPT' is at most the set of bought edges if we add the payments $\chi_1, \ldots, \chi_{i+1}$ to p^*, and then remove the payments $p_{n_1}, \ldots, p_{n_{i+1}}$. This results in edge f being bought only if $p_{n_1}(f) = 0$ or if χ_1 uses edge f. The latter is because if χ_1 uses edge f in its connection cycle, then $\chi_1(f) = p_{n_1}(f)$, and so adding the payments of χ_1 and removing the payments of p_{n_1} cancels out for f.

Now we prove that OPT' is a feasible solution. The only changes needed to the proof of Lemma 7 are as follows. To show that n_{i+1} has a connection cycle in OPT', we already argued in Lemma 7 that χ_{i+1} does not contain any edges $e_1, e_2, \ldots, e_i$. Also notice that χ_{i+1} does not contain the edge f, since if it did, then χ_{i+1} would contain node n_1, and thus Condition (1) would hold for node n_1. This is also the reason why χ_ℓ for $1 < \ell \leq i$ does not use edge f. To show that n_1 has a connection cycle in OPT', we simply note that if the connection cycle of χ_1 contains f, then f still exists in OPT'. Therefore, if we trace the paths of χ_1 from n_1 in OPT', each of these paths either reaches

s on edges of OPT', or stops at a node in $\{n_2, \ldots, n_{i+1}\}$, which already have connection cycles in OPT' due to the inductive hypothesis. The rest of the argument for why nodes $n_1, \ldots, n_{i+1}$ have a connection cycle in OPT' is identical to the proof of Lemma 7.

Now consider an arbitrary player t, and let C_t be its connection cycle in OPT. If C_t does not use edges of $e_1, e_2, \ldots, e_i$, or edge f then trivially t has a connection cycle in OPT' as well. Trace the paths of C_t starting at t in OPT' (call these paths P_1 and P_2). Each of these paths either reaches s using edges of OPT', or stops early because the next edge in it does not exist in OPT'. Since the only edges removed from OPT are $e_1, e_2, \ldots, e_i, f$, and our chain is simply a path in OPT by Lemma 5, then these paths must stop at nodes n_1, n_{i+1}, or j where $f = (n_1, j)$. Suppose that P_1 stops at j, meaning that it uses the arc (j, n_1) in the path from t to s. Since no players witness arc (j, n_1) (i.e., $W(j, n_1)$ does not exist), then t does not witness (j, n_1), so there must be another connection cycle of t in OPT (call it C'_t) that does not contain this arc (j, n_1). Using the same argument as above for connection cycle C'_t, we obtain two disjoint paths in OPT' that either reach s, or stop at nodes n_1 or n_{i+1}. Since nodes n_1 and n_{i+1} were already shown to have connection cycles in OPT', then t has a connection cycle in OPT' as well by Lemma 3. The rest of the argument is the same as for Cases (1)-(3). ■

Since we proved that the algorithm does not break, then it must pay for the entire solution OPT, and as we argued above, its payments form a Nash equilibrium. Since there is a Nash equilibrium that buys the socially optimal network, then price of stability is 1. ■

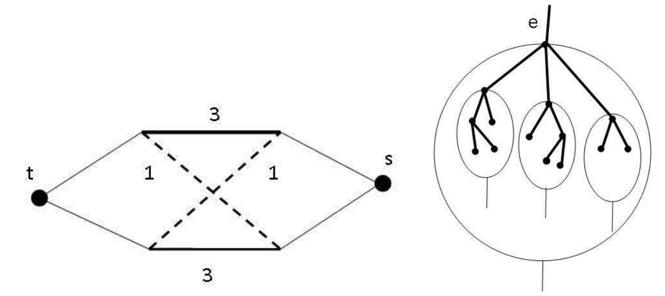
4 Good Stable Solutions When Not All Nodes Are Terminals

Approximation Algorithm Technique To prove our main result, we define a restricted version of the Survivable Connection Game such that this version of the game is identical to the original game, except that for each strategy p_i of a player i, the set of deviations she can take are restricted. In this restricted version of the game, a player is only allowed to deviate by changing the payments on one of her paths, instead of both of them at once, i.e., for any strategy profile $p = (p_1, \ldots, p_n)$, p'_i is a deviation for player i from her strategy p_i if for each edge e along *one* path from i to s in G_p, $p_i(e) = p'_i(e)$. In this restricted version of the game, each player should also determine her connection paths as well as the payment it makes on the edges as part of its strategy. In order to avoid ambiguity, in the rest of the paper we will use the term *stable solutions* for the equilibria of the restricted version of the game and the results we obtain for the stability of the restricted version of the game will also imply results about the Nash equilibria of the original Survivable Connection game due to Theorem 3.

Theorem 3 *A stable solution p is a 2-approximate Nash equilibrium of the original Survivable Connection Game.*

Proof. By a 2-approximate Nash equilibrium, we mean that no player can save more than a factor of 2 in her cost by changing all of her payments at once. Suppose to the contrary that player i can reduce her cost by at least a factor of 2 by switching her payments from $p_i(e)$ to $p'_i(e)$ on all edges e, i.e. $\sum_{e \in E} p'_i(e) < \frac{1}{2} \sum_{e \in E} p_i(e)$. Since in a stable solution player i makes a strictly positive payment only for the edges she uses for her connection paths, there exists a connection path P of player i such that $\sum_{e \in E} p'_i(e) < \sum_{e \in P} p_i(e)$. Then p is not a stable solution since player i can reduce her cost by replacing the payments on the edges of P by $p'_i(e)$. ■

The restricted version of the Survivable Connection Game is also of independent interest since it models the scenarios where a player wishes to keep one of her paths the same as before the deviation, the case where a complex deviation involving re-routing of both of the player paths is too much for a player, or the case where each path of a single player is managed by a different entity, which is possible when a player represents a large company.



(A) (B)

Fig. 6 (A) An example illustrating our stable solution concept. (B) Result of the Tree Generation algorithm. The ovals represent smallest witness sets.

In Figure 6(A), we have a game with one player that wants to connect from t to s through 2 edge-disjoint paths. Each thick edge has a cost of 3, each dashed edge has a cost of 1 and the total cost of the thin edges is ϵ. Any feasible solution has to include all 4 of the thin edges. Let p be a strategy of the player where she buys the 2 thick edges and all 4 of the thin edges where she uses the upper path and the lower path in Figure 6(A) as her connection paths. Please note that, though the connection paths are uniquely determined on this game for this set of bought edges, this is not true in general, therefore they are to be specified as part of the strategy. Let p' be a strategy where the player buys the dashed and the thin edges as well as the top thick edge and for each connection path she uses a dashed edge and its 2 incident thin edges. Observe that in this strategy player buys the top thick edge although she does not use it. If the player switches her strategy from p to p', she reroutes both of her connection paths. However, p' is considered a valid deviation since she keeps the payments on one of her connection paths in p the same.

Recall that because of Theorem 3, we can restrict our attention to *stable* solutions as defined above, as results about such solutions would also imply results about approximate Nash equilibria of the general Survivable Connection Game. In the following discussion we will use the terms *price of anarchy* and *price of stability* for the ratio of the worst and best *stable* networks to the socially optimal network instead of the ratio of the Nash equilibria to the socially optimal network. The objective function that we use to measure the quality of a solution is social welfare, which for our game is the same as the total cost of the network. As we show below the price of anarchy cannot be more than $2N$ and the bound is tight. Because of this, we focus on the price of stability in the rest of the paper.

4.1 Price of Anarchy

Observe that the price of anarchy for the *Survivable Connection Game* cannot be more than $2N$. If it was more than $2N$ there would be a player whose cost is more than 2 times the socially optimal network. Since in a stable solution players would not contribute to the payment of any edge that are not on their paths, the payment of this player along one of its paths will be more than the whole socially optimal network. By replacing the payments on this path with the whole socially optimal network, this player would reduce its payments and stay feasible. Therefore, the price of anarchy cannot be more than $2N$. As Figure 7(A) demonstrates, this bound is actually tight and it is $2N$. If each player is able to change the payments on both paths at the same time, then the same argument applys with the bound of N, instead of $2N$.

In Figure 7(A), the cost of a dashed edge is 1, the cost of a thick edge is $2N$ and the total cost of the other solid edges are ϵ. Consider the strategy vector where every player contributes 2 to each thick edge, a total of $\frac{\epsilon}{N}$ to the solid edges and nothing to the dashed edges. Observe that this

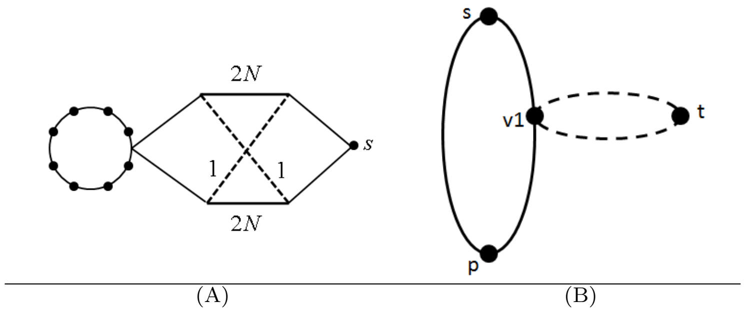


Fig. 7 (A) An instance where Price of Anarchy is $2N$. The terminal nodes are the nodes on the circle. (B) The solid paths are the connection paths of player p. The dashed lines are the connection paths of player t, which intersect the connection cycle of p at a point v_1.

strategy vector describes a stable solution. Consider a possible deviation of a player. Since it has to keep its payments on one of its paths, it has to keep the payment made to one of the thick edges. It cannot change its payments to solid edges since any feasible solution should have them. All it can do is to release its payments on one of the thick edges. However, to satisfy the feasibility it has to buy both of the dashed edges which has a total cost of 2. The players does not have an incentive for unilateral deviation since all they can do is to replace a payment of 2 to one of the thick edges with a payment of 2 to the dashed edges. Therefore there is a stable solution with total cost $4N+\epsilon$. Observe that the socially optimal network would only buy the solid and the dashed edges with total cost of $2+\epsilon$. Therefore, the price of anarchy can actually be $2N$ for some instances of this problem. Notice also that Figure 7(A) gives an example where a stable solution is a 2-approximate Nash equilibrium, and no less, showing that the bound in Theorem 3 is tight.

4.2 Forming a Stable Solution on the Edges of OPT

In this section, we present an algorithm to find a stable strategy vector that buys OPT, which implies that the price of stability for the *Survivable Connection Game* is 1. Since a strategy of a player is composed of specifying 2 edge-disjoint paths to s and the amount of payment made on the edges, then we must specify both of these for every player.

Although our proof techniques do not require that the connection paths of the players be *node-disjoint* as well as edge-disjoint, having this property greatly simplifies the proofs. Though it may not be possible to find node-disjoint connection paths for all players in a feasible network, the following theorem states that there always exists an *equivalent graph* where each player has node-disjoint connection paths in OPT. Equivalence among the graphs means that the socially optimal network of the new graph costs as much as OPT, and that for each stable network in the new graph, there corresponds a unique stable network in the original graph with the same cost.

Theorem 4 *There exists an equivalent graph G' with a routing on the centrally optimal solution that is node-disjoint.*

Proof. To prove the result, we will give an algorithm that explicitly transforms G into G' and explicitly constructs the node-disjoint paths on it. We will first select a player node p on G arbitrarily and will try to find 2 node-disjoint connection paths for it on the socially optimal solution. If we can find such 2 connection paths, we will *mark* the edges on them and proceed with the next player. A mark on an edge means that that edge is being used by the connection path of a player. If we can't find 2 node-disjoint paths from p to s, we will select just 2 edge-disjoint connection paths. These connection paths cross at some nodes of G. For each node n where these connection paths are crossing, we will *split* n into two, namely n_1 and n_2.

Node Splitting When we split a node n, we will remove n from G and add 2 nodes instead. Two incident edges of n that are used by one of the connection paths are assigned as incident edges of n_1 and similarly 2 incident edges used by the other connection path of p are assigned to n_2. Observe that n may have had more than 4 incident edges. We assign all other incident edges of n to n_1 or n_2 arbitrarily. We will add 2 extra edges between n_1 and n_2 and set the cost of these edges to 0. These edges are necessary for the equivalence of the new graph, as will be explained shortly. Observe that n may have been a player node. If it was, we will assign one of n_1 or n_2 arbitrarily as the node of this player. Once we do the splitting we will *mark* the edges on the connection paths and proceed with the next player.

Before forming the connection paths of the other players, we would like to show that the socially optimal network is still feasible in this new graph. Since the network we started with was feasible, each player t had 2 edge-disjoint paths before we split n. If these paths were not crossing n, t clearly still has 2 edge-disjoint paths since we haven't made any changes on the nodes and the edges on its paths. Assume t had 2 connection paths such that one or both of the paths were crossing n. Observe that in this new graph, both n_1 and n_2 (and s, by definition) are on the connection cycle of p. Therefore, each connection path of t crosses the connection cycle of p and t still has 2 edge-disjoint paths to s by Lemma 4.

Now, we explain how we form the node-disjoint paths for the remaining players. We loop through all players and form the connection paths for them one at a time. When we form the connection paths for a player t, we first grow 2 paths from t (which we know exist since t is feasible). We stop when these connection paths cross a node adjacent to the marked edges. Let these nodes be v_1 and v_2 respectively. From there on, t connects to s through 2 node-disjoint paths by using only the marked edges. Let us explain how t does that.

Assume we had assigned a ranking to all the players. The first player whose connection paths are formed had rank 1, the next player whose connection paths are formed had rank 2, and so on. Let r_1 and r_2 be the ranking of the lowest ranked players whose connection paths pass through v_1 and v_2 respectively.

Let us first consider the case where $r_1 = r_2$. Then t has connected to a connection cycle of the player with rank r_1 at 2 nodes (v_1 and v_2) and can connect to s through this connection cycle by Lemma 4. Since the player ranked r_1 already has 2 node-disjoint paths (by construction in the previous steps), then so does t, if both of its paths are node-disjoint till it connects to v_1 and v_2. Therefore, the connection paths of t may pass through a common node only until it touches the connection cycle of r_1 or right at the node it connects to the cycle, i.e., if v_1 and v_2 are actually the same node as depicted in Figure 7(B). To make the connection paths of t node-disjoint, we split the nodes that are common to both connection paths of t. Observe that splitting a node does not violate feasibility of any players as we have proven above. Furthermore, splitting of any node except v_1 will not interfere with the connection paths we have already formed since these nodes are not adjacent to the already marked edges. However, splitting of v_1 may interfere with the already formed connection paths. To illustrate, a connection path of r_1 was passing through v_1, i.e., two incident edges of v_1 were on a connection path of r_1. After splitting v_1, each one of these two edges are assigned to the newly added nodes. If they are assigned to different nodes, then the r_1-path is interrupted. To ensure that r_1 still has 2 node-disjoint paths, we should find a path between these 2 nodes that were added to the graph to replace v_1. Observe that we have grown 2 paths from t and v_1 was the first node adjacent to the already marked edges that are touched by these paths. Therefore, r_1 can use this path through t between newly added nodes, and this portion of the path is still node-disjoint from the other path of player r_1, since it does not use any nodes adjacent to marked edges. After we have split the nodes and obtained node-disjoint paths for t, we mark the edges it uses as well.

Now, we need to address the case where $r_1 \neq r_2$. Without loss of generality, let $r_1 < r_2$. Since the connection paths of r_2 are formed after the connection paths of r_1, the connection paths of r_2

both connect to the connection cycle of r_1 by construction. The connection path of t that touches to the connection cycle of r_2 can follow the cycle in any direction until it encounters the connection cycle of r_1. It should choose the direction that will lead to a node in the cycle of r_1 other than v_1. Since t now has two paths connected to the connection cycle of r_1, it can connect to s through this connection cycle by Lemma 4. Once again, the connection paths of t may pass through a common node only until it touches v_1 and v_2 and we will apply a splitting on this common node in exactly the same way as explained in the above paragraph.

We have formed a new graph G' and showed that each player has 2 node-disjoint paths on it. We also need to show that G' is equivalent to G. That is, we need to show that the centrally optimum solution costs the same in G and G', and in fact that the optimum solution in G' consists of the exact same set of edges, on which we formed the node-disjoint routes above. Observe that G' has exactly the same set of edges as G except 2 free edges added at each node splitting among the newly introduced nodes. We have already shown how to form node-disjoint paths on G' for all players on the edges that were part of the socially optimal network on G. Therefore, the network we have formed on G' has exactly the same cost as the socially optimal network on OPT. We also need to show that the network we formed is indeed the socially optimal network on G' as well. For the purpose of contradiction, assume it is not, i.e., there exists a cheaper network on G'. On this new network, the connection paths of the players may not be node-disjoint but they are necessarily edge-disjoint. Observe that exactly the same set of edges except the free edges form a feasible solution on G as well, which would contradict the fact that the network we started with was optimal in G.

To finish our proof of equivalence, we also need to show that for any stable solution on G' the corresponding feasible solution in G is stable as well. Starting with a stable solution on G' and the corresponding solution in G, all we need to show that for all players and their all possible deviations in G, there corresponds an equal cost deviation in G'. If the deviation of a player t does not pass through a node that we have split, then the statement is trivially true. If the deviation of the player passes through a node v that we have split into v_1 and v_2, the corresponding deviation in G' has exactly the same cost due to the zero-cost edges added while transforming the graph.

Therefore, since the optimal solutions cost the same, and the stable solutions in G' have corresponding stable solutions in G, we know that the price of stability in G is at most that of G'. ■

As explained in the proof of Theorem 4, the equivalent graph and the node-disjoint connection paths can be efficiently determined. Since a stable solution in the new graph corresponds to a stable solution in the original graph, it is enough to form a stable solution in G', and so we assume that there is a routing on OPT that is node-disjoint. Fix such a routing. We will now show that it is possible to pay for the edges of OPT so that this routing forms a stable solution.

Let W be the set of all smallest witness sets of OPT, i.e., $W = \{W(i,j) | \text{for some } (i,j) \in OPT\}$. For ease of explanation, we will first consider the case where W is a *laminar* set system, i.e., for any 2 elements of W_1 and W_2 of W, either W_1 and W_2 are disjoint or one of them is a subset of the other. We will first prove our results under the assumption that W is a laminar set system. Theorem 6 describes how our algorithm can be modified for the case W is *laminar with path exceptions* and we conclude the proof of our main result by proving Theorem 1 in Section 5.

Proof Overview

1. We define the algorithm that constructs payments for the edges of OPT, assuming that W is laminar. By construction, it is clear that this algorithm always forms stable payments. Thus, if it succeeds in paying for all edges of OPT, then the payments form a Nash equilibrium.
2. For every edge $e = (i, j)$ of OPT, we do the following:

(a) Show how to generate trees T_i and T_j contained in $W(i,j)$ and $W(j,i)$ with all leaf nodes being player nodes, and all other nodes being non-player nodes. Lemmas 9, 10, and 11 establish the properties of these trees.

(b) Theorem 5 proves that the leaves of T_i and T_j are willing to pay for the cost of e in our payment algorithm. This implies that this algorithm pays for all edges.

3. Theorem 6 shows how to change our algorithm and proof to handle the general case when W is laminar with path exceptions, and not simply laminar.

Our payment scheme is formed by Algorithm 2. While deciding the payment on an edge $e = (u,v)$, the algorithm needs to form the cheapest deviation χ_i on $G-e$, for all players i in $W(u,v)$ and $W(v,u)$. For each player i in $W(u,v)$ or $W(v,u)$, we call the connection path of i that does not use e the *enduring path* of player i and denote it as E_i. To form the cheapest deviation χ_i in this algorithm, we need to be able to find the cheapest way for a player to form 2 edge-disjoint paths to s, while keeping the payments on E_i the same. As shown in Algorithm 2, this can be done by using modified costs $c'_i(f)$ for each edge f, that represent how much it costs for player i to use edge f in χ_i. Specifically, for f not in OPT, $c'_i(f) = c(f)$, the actual cost of f. For the edges f of OPT that i has not paid anything for, or for the edges in E_i, we have that $c'_i(f) = 0$, since from i's perspective, it can use these edges for free (it cannot change the payments on E_i, so from a deviational point of view, those edges are free for i to use in χ_i). For all the other edges f that i is paying $p_i(f)$ for, $c'_i(f) = p_i(f)$, since that is how much it costs for i to use f in its deviation χ_i.

Input: The socially optimal network OPT
Output: The payment scheme for
OPT
Initialize $p_i(e) = 0$ for all players i and edges e;
Loop until the payments for all edges are determined;
    Pick an edge $e = (u,v)$ whose payment scheme has not been decided yet;
    Pay for all the edges in e's smallest witness sets recursively;
    Loop through all terminals i of $W(u,v)$ and $W(v,u)$ until e is paid for;
        Define $p_i = \sum_{f\in(E\setminus E_i)} p_i(f)$;
        Define $p(e) = \sum_j p_j(e)$;
        Define $c'_i(f)$ to be the modified cost of f for i;
        Define χ_i as the cost of the cheapest deviation by player i in $G-e$ under c'_i;
        Set $p_i(e) = \min\{\chi_i - p_i, c(e) - p(e)\}$.

Algorithm 2: Algorithm That Generates the Payment Scheme

We first claim that if this algorithm terminates, then the resulting payment forms a stable solution. Consider the algorithm at some stage where we are determining player i's payment to e. The cost function c'_i reflects the costs player i faces if she deviates in the final solution (not counting the cost of E_i, which stays the same). χ_i is the cost of deviating while preserving the payments on the enduring path, and so is the smallest amount player i would have to pay if she wanted to deviate from the strategy we are forming. We never allow i to contribute so much to e that her total payments exceed the cost of her cheapest deviation. Therefore, it is never in player i's interest to deviate. Since this is true for all players, this algorithm forms a stable solution if it terminates.

To prove the algorithm succeeds in paying for OPT, we need to show that for any edge e, the terminals inside its smallest witness sets will be willing to pay for e. To show this, we will actually prove a stronger statement. Specifically, for every edge $e = (i,j)$ we will generate two trees T_i and T_j in $W(i,j)$ and $W(j,i)$ rooted at i and j respectively, such that the leaves of T_i and T_j are player nodes/terminals, and all other nodes are non-player nodes. We will show that just the leaves of these trees are willing to pay for all of e, and the other terminals in the smallest witness sets are not needed. In fact, we can just as easily make our algorithm only ask the players that are leaves of these trees to contribute to the payment of e.

Tree Generation Since W is laminar, we construct the trees $T_j(i,j)$ recursively, starting with the smallest sets in W, and continuing to the sets containing those. To construct $T_j(i,j)$, we start the search in $W(j,i)$ from j. If j is a player then the tree is just a single node. If it is a non-player node we add all its incident edges in $W(j,i)$, along with their corresponding trees in their smallest witness sets from the other side. That is, for every edge (j,k) inside $W(j,i)$, we add the edge (j,k) and the subtree $T_k(j,k)$. These trees must have already been generated, since those witness sets are contained inside $W(j,i)$. We presented the tree generation in terms of the smallest witness sets but indeed it is equivalent to making a breadth-first search in $W(j,i)$ starting from j, except we stop when a branch arrives at a player node.

When *Tree Generation Algorithm* reaches a non-player node j below an edge (j,i), it adds all incident edges (j,k) in $W(j,i)$ as well as the corresponding trees $T_k(k,j)$ of these edges in their witness sets $W(k,j)$, as shown in Figure 6(B). However, we haven't proven these smallest witness sets $W(k,j)$ indeed exist since the arc (k,j) does not necessarily have a witness set (although one of (k,j) and (j,k) must).

Lemma 9 *Any arc (k,i) of $T_i(e)$ generated by the Tree Generation Algorithm has a smallest witness set $W(k,i)$.*

Proof. If i is a player node then *Tree Generation Algorithm* does not expand to any other nodes and returns i as the tree $T_i(i,j)$. For the purpose of contradiction, suppose that i is a non-player node, and one of the incident edges $f=(i,k)$ in $W(i,j)$ does not have a witness set from the side of k, i.e., $W(k,i)$ does not exist. Since every edge must have a witness set, this implies $W(i,k)$ exists. Due to laminarity of smallest witness sets, we know that $W(i,k) \subseteq W(i,j)$. This implies that (i,j) is a boundary edge of $W(i,k)$, since i is contained inside it, and j is outside. Since the only boundary edges of $W(i,k)$ are (i,j) and (i,k), then every player node inside $W(i,k)$ must witness the edge (i,j). In fact, this means that $W(i,k)$ is a witness set of (i,j). $W(i,k)$ is strictly smaller than $W(i,j)$ (since it does not contain the node k), and so this contradicts $W(i,j)$ being the smallest witness set of arc (i,j). ■

Now that we know *Tree Generation Algorithm* is well-defined, we should make sure that it actually generates trees.

Lemma 10 *The structure $T_j(e)$ generated by the Tree Generation Algorithm for any edge $e=(i,j)$ of OPT is a tree such that all leaf-nodes are player nodes and all non-leaf nodes are non-player nodes.*

Proof. We will prove this result by induction on the size of $T_j(e)$ in terms of the nodes it contains. For the base case, when j is a player node, $T_j(e)$ is a single node and therefore it is clearly a tree. Assume all the structures generated by the *Tree Generation Algorithm* that have at most n nodes are trees, with all leaves being player nodes and all non-leaves being non-player nodes. Let $T_j(e)$ be a structure with $n+1$ nodes. Then clearly j is not a player node. Since j is a non-player node, *Tree Generation Algorithm* expands to all its incident edges in $W(j,i)$.

Let $(j,n_1),\ldots,(j,n_l)$ be the incident edges to j in $W(j,i)$ and $W(n_1,j),\ldots,W(n_l,j)$ be their respective smallest witness sets. This is illustrated in Figure 8, where the ovals represent the smallest witness sets $W(n_1,j),\ldots,W(n_l,j)$. If one of these smallest witness sets is a subset of another smallest witness set, i.e., $W(n_u,j) \subset W(n_v,j)$ for some $u,v \in 1,\ldots,l$ then the two boundary edges of $W(n_v,j)$ must be (n_u,j) and (n_v,j), since n_v and n_u are inside this set, and j is outside. This implies that both of the connection paths of all player nodes in $W(n_v,j)$ would have j as an intermediate node, which would violate node-disjointness of the routing paths. Since none of these smallest witness sets is a subset of another they are all disjoint sets due to the laminar property of smallest witness sets.

Since j is a non-player node, *Tree Generation Algorithm* expands to all its incident edges in $W(j,i)$, and the tree structure will be composed of j and the structures of the incident edges which are trees due to our inductive assumption. Since all these structures live in disjoint sets due to laminarity of smallest witness sets, then $T_j(e)$ is a tree as well. Since all the leaves of $T_j(e)$ are leaves of the trees in the witness sets $W(n_1,j),\ldots,W(n_l,j)$, then we know that all leaves of $T_j(e)$ are terminals and all non-leaves are non-terminals. ■

Though we have stated above that *Tree Generation Algorithm* is indeed equivalent to making breadth-first search in $W(j,i)$ starting from j except we stop when a branch arrives at a player node, we have explained the expansion of trees in terms of the smallest witness sets for good reasons. We now know each player node t at the leaf of a tree $T_j(e)$ is in the smallest witness set of all the arcs of the path of the tree from her to j. This implies that every one of these edges *must* be used by t to connect to s, and since the connection paths of s are node-disjoint, this implies that one of the connection paths of t must simply proceed up the tree $T_j(e)$. Therefore, we know that the other connection path of t does not use any edge of this path. Lemma 11, which is one of the key lemmas for our proof, shows an even stronger property and states that the connection paths of all players leaving $W(j,i)$ through the other boundary edge (i.e., not the edge (i,j)) don't use any edge of $T_j(e)$ at all.

Lemma 11 *Let $W(j,i)$ be a smallest witness set of some arbitrary edge $e=(i,j)$. Let p be a player inside $W(j,i)$. Then the other connection path of p (that leaves $W(j,i)$ through the other boundary edge) does not use any of the edges of $T_j(e)$.*

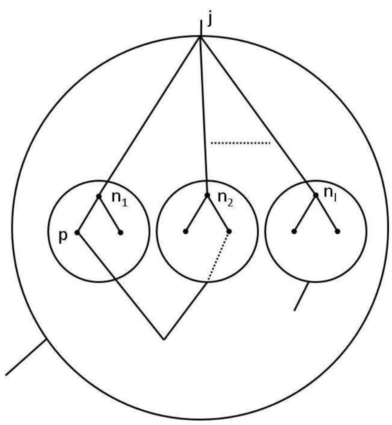


Fig. 8 Structure of Other Paths

Proof. For the purpose of contradiction, assume there exists a smallest witness set $W(j,i)$ of some edge $e=(i,j)$ that includes a player node p such that the other connection path of t uses some edges of $T_j(i,j)$. Let $W(j,i)$ be the *smallest* of such smallest witness sets in terms of the number of nodes included.

If j is a player node then this lemma trivially holds since the tree returned by the *Tree Generation Algorithm* is a single node and it does not have any edge. Therefore, consider the case where j is a non-player node. Let $(j,n_1),\ldots,(j,n_l)$ be the highest level edges of the tree $T_j(e)$ and $W(n_1,j),\ldots,$ $W(n_l,j)$ be their respective smallest witness sets. This is illustrated in Figure 8, where the ovals represent the smallest witness sets $W(n_1,j),\ldots,W(n_l,j)$. If one of these smallest witness sets is a subset of another smallest witness set, i.e., $W(n_u, j) \subset W(n_v, j)$ for some $u, v \in 1, \ldots, l$ then both of the connection paths of all player nodes in $W(n_u, j)$ would have j as an intermediate node which would violate node-disjointness of the routing paths. Since none of these smallest witness sets is a subset of another they are all disjoint sets due to the laminar property of smallest witness sets.

Let us first consider the case where p is in one of these smallest witness sets. Without loss of generality, let $p \in W(n_1, j)$, as shown in Figure 8. Observe that the other connection path of p cannot use any edge of T_j in $W(n_1, j)$ since $W(j, i)$ is assumed to be the smallest of such smallest witness sets. Then it uses some edges of T_j in one of the other smallest witness sets, w.l.o.g. $W(n_2, j)$. Since $W(n_2, j)$ has exactly two boundary edges, the other connection path of p must use them both to enter and exit $W(n_2, j)$. Since j is incident to one boundary edge of both $W(n_1, j)$ and $W(n_2, j)$, then both connection paths of p will have to contain j as an intermediate node, which contradicts node-disjointness of our connection paths.

Finally consider the case where $p \in W(j, i)$ but it is outside $W(n_1, j), \ldots, W(n_l, j)$. Observe that both of the connection paths of t cannot route through the smallest witness sets $W(n_1, j), \ldots, W(n_l, j)$, since otherwise both of the connection paths will include j as an intermediate node. Since only one of p's connection paths is routing through one of those smallest witness sets $W(n_1, j), \ldots, W(n_l, j)$, then the other connection path of p cannot use any edges of $T_j(e)$, since all the edges of $T_j(e)$ are either inside these smallest witness sets, or are incident to j. ■

Now that we generated the trees $T_i(e)$ inside each smallest witness set that are disjoint from the other connection paths of the player nodes, we are ready to state our theorem for the special case W is laminar. We prove the same result for the general case(when W is laminar with path exceptions) in Theorem 6.

Theorem 5 *If W is laminar Algorithm 2 fully pays for OPT, and so the price of stability is* 1*. Moreover, the leaves of $T_i(e)$ and $T_j(e)$ are willing to pay for an edge $e = (i, j)$ without help from any other players.*

Proof. For the purpose of contradiction, suppose that for some edge e, after all players have contributed to e, $p(e) < c(e)$. For each player k of $T_i(i, j)$, consider the longest subpath of A_k until it leaves $T_i(i, j)$. Call the highest ancestor of t_k on this subpath k's deviation point, denoted d_k. Let D^i be a minimum set of deviation points such that every player in $T_i(i, j)$ has an ancestor in D^i. The minimum set of deviation points D^j for the tree $T_j(i, j)$ is defined similarly.

First we consider the simpler case where e has a witness set from one side only; i.e. $T_i(i, j)$ exists but $T_j(i, j)$ does not exist. Let $D^i = \{d_1, d_2, \ldots, d_n\}$ be the set of highest deviation points and $t_1, t_2, \ldots, t_n$ be the players such that their alternate paths pass through $d_1, d_2, \ldots, d_n$ respectively. Make a new network called OPT' by replacing the portion of OPT above D^i by the alternate cycles of $t_1, t_2, \ldots, t_n$. Notice that the payment the players were making for e was not sufficient to buy it. In the construction of the new network, some of the players are deviating and others are sticking to their existing strategies, therefore, nobody is increasing its payment. Since they are able to buy this new network, OPT' is cheaper than OPT. Therefore, to form a contradiction we only need to show that OPT' is still feasible, i.e., all players are 2-connected to s.

First, consider the players $t_1, \ldots, t_n$. Since their alternate connection cycles connect them to s without using any more edges of the tree by Lemma 12 (which is defined below), then they are still 2-connected to s. The players not witnessing e also still satisfy their connection requirements, since the only edges taken away were edges of T_i, and if a player is witnessing an edge f in this tree, then she is witnessing all the edges between e and f since the tree generation algorithm is recursive. Therefore, we only have to make sure that the players that are witnessing e from the side of i (i.e., witnessing the arc (i, j)) but are not $t_1, \ldots, t_n$ also satisfy their connection requirements.

Consider a player node p that is a leaf of T_i, but not one of $t_1, \ldots, t_n$. We cannot immediately assume it is still 2-connected, since the alternate connection cycle including the highest deviation point above it in the tree might not be edge-disjoint from p's other connection path. Let P be the unique path from p to the highest deviation point d_k above it in T_i. P is necessarily edge-disjoint from the other connection path of p by Lemma 11, and it is still in OPT' since it is below the highest deviation points. Since P and the other connection path of p are disjoint and they connect to 2 nodes of the alternate connection cycle of a deviating player, i.e., P connects to d_k and the other connection path connects to s, p has 2 edge-disjoint paths to s by Lemma 4.

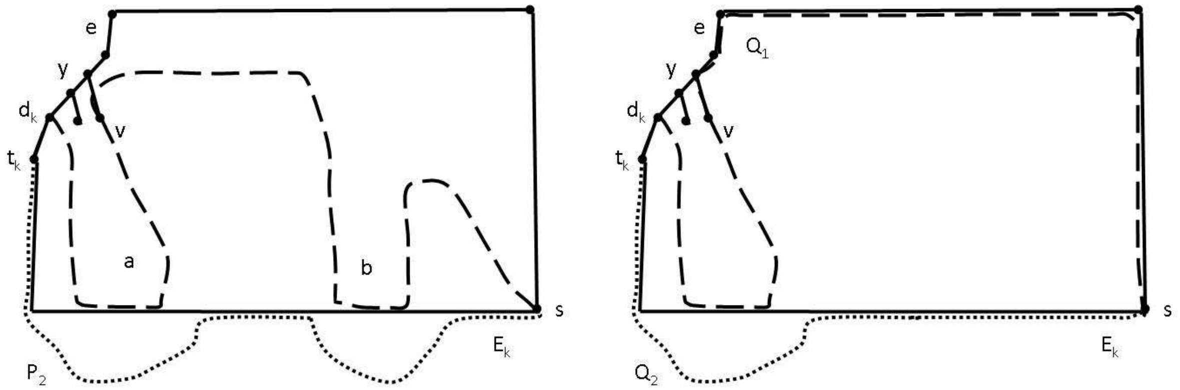
We have shown that the connection requirements of players in T_i are still satisfied, but we still need to address players that witness e but are not in T_i. The key observation here is that previously the paths of these terminals passing through e were first connected to T_i through a player. This property is due to the tree generation algorithm, since *Tree Generation Algorithm* expands to all the neighbors when it reaches to a non-player node. Consider a player p that witnesses e but does not belong to T_i, and let t be the first node of T_i on p's connection path that uses e. Let P be the union of the segment of p's connection path from p to t and the unique path in T_i between t and the highest deviation point above t, which we know must be disjoint from the other connection path of p. The other connection path of p is still in OPT' by Lemma 11, so we can use Lemma 4 to show feasibility of p in OPT', since P connects to a highest deviation point and the other connection path connects to s, both of which are in the alternate connection cycle of a deviating player.

In the more general case where both (i, j) and (j, i) have witness sets, i.e. both $T_i(i, j)$ and $T_j(i, j)$ exist, there are two sets of minimal deviation points $D^i = \{d_1^i, d_2^i, \ldots, d_n^i\}$ and $D^j = \{d_1^j, d_2^j, \ldots, d_m^j\}$ of $T_i(i, j)$ and $T_j(i, j)$ respectively. Let $t_1^i, t_2^i, \ldots, t_n^i$ and $t_1^j, t_2^j, \ldots, t_m^j$ be the corresponding players of $W(i, j)$ and $W(j, i)$. Observe from the case where an edge is only witnessed from one side, that when a player has taken its alternate connection cycle, it has only released its payments to the edges that are above its deviation point. Assume the alternate connection cycles of $t_1^i, t_2^i, \ldots, t_n^i$ do not use any edge of $T_j(i, j)$ above the set of points D^j and also assume alternate connection cycles of $t_1^j, t_2^j, \ldots, t_m^j$ do not use any edge of $T_i(i, j)$ above the set of points D^i. In this case, we can obtain a cheaper feasible network with exactly the same argument above by letting those players to deviate. Observe that the players witnessing e in the direction from i to j (i.e., witnessing the arc (i, j)) satisfy their connection requirements by Lemma 4 by connecting to 2 points of the alternate connection cycle of one of $t_1^i, t_2^i, \ldots, t_n^i$ through 2 edge-disjoint paths. Similarly, the players witnessing e in the direction from j to i satisfy their connection requirements by connecting to 2 points of the alternate connection cycle of one of $t_1^j, t_2^j, \ldots, t_n^j$ through 2 edge-disjoint paths.

Consider now the more complicated case where the alternate connection cycle of at least one of the players of $T_i(i, j)$ intersects $T_j(i, j)$ at a point y above D^j, without loss of generality let this player be t_1^i. Now, to construct OPT' let the players $t_1^i, t_2^i, \ldots, t_n^i$ take their alternate connection cycles similar to the previous cases, but have $t_1^j, t_2^j, \ldots, t_m^j$ stay with their existing strategies. The players witnessing e from the side of i (i.e., witnessing the arc (i, j)) fulfill their connectivity requirements by exactly the same argument above. For the players witnessing e from the side of j, feasibility can be shown with Lemma 4 again. All we need to show is to identify 2 edge-disjoint paths to the alternate connection cycle of t_1^i. Each player has a path to a player-node t in T_j and there is a path from t to y by using the edges of the tree. The union of these paths connects to y, which is a node of the alternate connection cycle of t_1^i, and by Lemma 11 it is necessarily edge-disjoint from the other connection path, which connects to s. ∎

Lemma 12 *Let player t_k be a leaf-node of $T_i(i, j)$. Then A_k, the alternate connection cycle of t_k, does not use any edge of $T_i(i, j)$ except in the subtree below d_k.*

Proof. By definition A_k, the alternate connection cycle of t_k, is a cycle including t_k, s and d_k and it uses the unique path in T_i between t_k and d_k. A_k is the connection cycle t_k will have if it takes the deviation of cost χ_k found by Algorithm 2 while we were deciding how the cost of e is to be shared among the players. Recall that the modified cost of χ_k is equal to whatever player t_k has paid so far in the previous iterations of the algorithm. Among various best deviations (all of which



(A) (B)

Fig. 9 (A) This figure shows the general alternate path structure of the deviation χ_k. Notice that, despite the way they appear in the figure, in general the paths P_1 and P_2 may not be using the edges of E_k in order. (B) Shows the construction of the deviating paths Q_1 and Q_2.

have the same cost), χ_k is the one whose corresponding alternate connection cycle includes as many ancestors of t_k as possible before including edges outside T_i. A_k is composed of 2 edge-disjoint paths between t_k and s, namely P_1 and P_2. Let P_1 be the path of A_k between t_k and s that uses the edges of T_i between t_k and d_k, and P_2 be the other connection path of A_k. We want to show that neither P_1 nor P_2 uses any edge of T_i that is not under the highest deviation point d_k.

For the purpose of contradiction, assume A_k, i.e., P_1 or P_2 or both, uses some edges of T_i that are not under d_k. Let v be the closest node to d_k in T_i such that v is not under the subtree rooted at d_k, and the alternate connection cycle passes through v. First let us consider the case where P_1 is the alternate connection path passing through v. Let a be the subpath of P_1 between t_k and v and b be the subpath of P_1 between v and s as shown in Figure 9(A). Note that P_2 is a path from t_k to s that is edge-disjoint from P_1 since P_1 and P_2 forms a deviation.

We will prove such a deviation χ_k does not exist by constructing another valid deviation χ'_k whose modified cost is as much as the modified cost of χ_k, and which includes more ancestors of t_k in T_i before including edges outside T_i. Let the paths of the alternate connection cycle of χ'_k, namely P'_1 and P'_2, be as follows. Define P'_1 as using the unique path in T_i between t_k and v and the edges of b between v and s. We will form P'_2 by using only the edges of P_2 and the enduring path E_k of t_k. Let us now describe how we form P'_2.

Recall that P_2 is a path that starts at t_k and ends at s that is edge-disjoint from P_1. If P_1 were not using any edge of the enduring path then P_2 would follow the enduring path between t_k and s, since under the modified costs E_k would be the cheapest path between t_k and s that is edge-disjoint from P_1 (recall that under the modified costs c'_k, the edges of E_k are free). If P_1, either a or b or both, is using some edges of the enduring path, then P_2 would be composed of several (maybe 0) subpaths of the enduring path which are connected to each other by subpaths outside the enduring path as shown in Figure 9(A). Let s_1 and s_2 be two adjacent subpaths of the enduring path that are used by P_2. Then some of the edges of the enduring path between s_1 and s_2 have to be used by P_1, since otherwise one could obtain a deviation cheaper than χ_k in terms of modified costs by just replacing the edges of P_2 between s_1 and s_2 with the edges of the enduring path between them. Therefore, between any two adjacent subpaths of the enduring path used by P_2, there is a subpath of the enduring path which is used by P_1. Similarly, between any two adjacent subpaths of the enduring path used by P_1, there is a subpath of the enduring path which is used by P_2 since otherwise the modified cost of χ_k would be decreased by replacing the portion of P_1 between these two subpaths with the edges of the enduring path between them. Therefore, P_1 and P_2 are using the subpaths of the enduring path in *alternating* order. To illustrate, let $s_1, s_2, \ldots, s_n$ be the subpaths of the enduring path that are used by P_1 and P_2. Then odd indexed subpaths are used by P_2 while the even indexed subpaths are used by P_1, i.e., either by a or b. For convenience, we can notice that P_2 starts at t_k and ends at s, and so say that the first and last segment s_1 and s_n are always segments of P_2, even though these segments might only consist of a single node. We form P'_2 by using only the edges of P_2 and E_k as follows. P'_2 is obtained by joining the adjacent subpaths of P_2 on the enduring path that are separated by a subpath used by a, by the edges of the enduring path. This makes some edges of P_2 redundant, and so we set P'_2 to be the cheapest path (using modified costs c'_k) from t_k to s using only edges in $(E_k - b) \cup P_2$. If several such cheapest paths exist, we choose the one that uses fewest edges outside of E_k.

In order to complete the proof, we need to show χ'_k is a valid deviation, i.e., P'_1 and P'_2 are edge-disjoint paths from t_k to s, and the modified cost of χ'_k is no more than the modified cost of χ_k. P'_2 is using only the edges of P_2 and the edges of the enduring path that are not used by P'_1. Since P_1 and P_2 are edge-disjoint, then P'_2 is disjoint from b. And since v is the closest node of T_i above d_k that is being touched by A_k, then P_2 is disjoint from the path in the tree T_i from d_k to v, and so is P'_2. Therefore, P'_1 and P'_2 are edge-disjoint.

Let us now show that the modified cost of χ'_k is no more than the modified cost of χ_k. P'_1 was obtained from P_1 by replacing a with the unique path between t_k and v. P'_2 was obtained from P_2 by joining the adjacent subpaths of P_2 on the enduring path that were separated by a subpath used by a, by the edges of the enduring path. Therefore, to show that χ'_k is not more expensive than χ_k, all we need to show is that the modified cost of the unique path between t_k and v in T_i is no more than the sum of the modified costs of a and the edges of P_2 that are not used by P'_2. We prove this as a separate technical lemma.

Lemma 13 *The modified cost of the unique path between t_k and v in T_i is no more than the sum of the modified costs of a and the edges of P_2 that are not used by P'_2.*

Proof. Let y be the lowest common ancestor of t_k and v in T_i. Since t_k dos not witness the edges between y and v, Algorithm 2 has never asked t_k to contribute to the cost of these edges and therefore the modified cost of these edges is 0. All we need to show is that the payment t_k made for the edges of T_i between d_k and y is no more than the sum of the modified costs of a and the edges of P_2 that are not used by P'_2.

Consider the time when Algorithm 2 was paying for the edges of T_i between d_k and y. To bound the payment t_k made on these edges, consider the following deviation whose alternate connection paths are Q_1 and Q_2. We are going to form Q_1 and Q_2 by using only the edges of a, the edges of P_2 that are not used by P'_2, and the free edges. The cost of this deviation constrains the payment of t_k on the edges of T_i between d_k and y, since when forming these payments, t_k will pay no more than the best deviation available to her. Therefore, if we show that such a deviation exists, then the lemma holds.

We form Q_1 as follows. As shown in Figure 9(B), Q_1 uses the edges of a to reach from t_k to v and then follows the edges of T_i between v and y. To reach from y to s, Q_1 uses the edges of the connection path of t_k in OPT that leaves $W(i,j)$ through i, which we will refer as the *first connection path* of t_k. Note that the edges of T_i between v and y are not witnessed by t_k and therefore have a modified cost of 0. The modified cost of the edges above y of the first connection path of t_k used by Q_1 are 0 as well, since either t_k is not in one of the smallest witness sets of these edges, or the algorithm has not started paying for them yet.

We now describe how to form Q_2. Let $s_1, s_2, \ldots, s_n$ be the subpaths of P_2 on the enduring path such that s_1 includes t_k while s_n includes s. Recall that between every 2 adjacent subpaths of P_2 on the enduring path, there is a subpath used by a or b. We claim that there exist a path between s_1 and s_n (and therefore between t_k and s) that is edge-disjoint from a, using just the edges of the enduring path and the edges of $P_2 - P'_2$. We will set an arbitrary such path to be Q_2. We now prove the existence of such a path by induction.

As our inductive hypothesis, we will assume that there is a path between s_1 and s_ℓ that is edge-disjoint from a, using just the edges of E_k and $P_2 - P_2'$. If the subpath of P_1 between s_ℓ and $s_{\ell+1}$ is used by b, then we can obtain a path between s_ℓ and $s_{\ell+1}$ that is disjoint from a, by taking the path between these two subpaths on E_k. This gives us the desired path from s_1 to $s_{\ell+1}$. If instead the subpath of P_1 between s_ℓ and $s_{\ell+1}$ is used by a, then we will use the path that P_2 was using to connect them. The path that P_2 was using to connect s_ℓ and $s_{\ell+1}$ is not used by P_2', since P_2' does not have to be edge-disjoint from a and can use the edges of the enduring path between s_ℓ and $s_{\ell+1}$ freely. In either case, we form the desired path from s_1 to $s_{\ell+1}$.

Both Q_1 and Q_2 are paths between t_k and s. To show that they form a valid deviation, all we need to show is that they are edge-disjoint. Q_2 is edge-disjoint from a by construction. Q_2 does not use any of the edges of the tree between y and v since P_2 does not (by our choice of v), and E_k does not (by Lemma 11). Therefore, this Lemma holds if Q_2 is edge-disjoint from the portion of the first connection path of t_k between y and s. In fact, Q_2 *might not* be edge-disjoint from the portion of the first connection path of t_k between y and s, but then we can form another deviation, composed of Q_1' and Q_2', that is no more expensive.

Consider the first time Q_1 and Q_2 touch the portion of the first connection path of t_k between y and s. Let the nodes they touch be v_1 and v_2 respectively. Now consider only the subpaths of Q_1 and Q_2 between t_k and v_1 (and t_k and v_2). Both of these subpaths have passed through some nodes of the enduring path since they both start at t_k. Suppose the last node along the enduring path touched by these subpaths, v_3, is touched by Q_1. Then let Q_1' consist of the edges of Q_1 from t_k to v_3, followed by the edges of the enduring path between v_3 and s; and let Q_2' consist of the edges of Q_2 from t_k to v_2 followed by the edges of the first connection path of t_k between v_2 and s. Q_1' and Q_2' are disjoint paths, and they cost at most what Q_1 and Q_2 did, since all the new edges they are using are free. Suppose instead that v_3 is touched by Q_2. Then let Q_1' consist of the edges of Q_1 from t_k to v_1, followed by the edges of the first connection path of t_k between v_1 and s; and let Q_2' consist of the edges of Q_2 from t_k to v_3, followed by the edges of the enduring path of t_k between v_3 and s. Since Q_1' and Q_2' are edge-disjoint and they are using only the edges of Q_1, Q_2, and the free edges, then we have found a cheaper valid deviation. ■

The proof for the case where P_2 is the alternate connection path passing through v is analogous to the above discussion. We define a to be the subpath of P_2 from t_k to v, and b to be the rest of P_2. P_1' becomes the path in T_i from t_k to v, followed by b, while P_2' is made from the edges of P_1 and E_k in the same way as described above. ■

We have proven that the price of stability is 1 under the assumption that W is laminar. We next describe how we can modify our algorithm for the case W is not laminar.

Theorem 6 *There is a stable solution as good as OPT for the Survivable Connection Game.*

Proof. Theorem 5 proves that there is a stable solution as good as OPT for the Survivable Connection Game under the assumption that W is laminar. Now, we describe how we can modify our payment algorithm to obtain a stable solution as cheap as OPT even if W is laminar with path exceptions.

Let u and v be 2 player or high degree non-player nodes such that there is a path P on OPT between u and v, of which all intermediate nodes are degree 2 non-player nodes. Observe that if a player t witnesses any arc e of P, then it witnesses all of the edges of P in the same direction since any path between t and s that uses e has to use all the edges of P. By saying that an edge $e = (i, j)$ is *witnessed in the direction* from i to j, we will mean that the arc (i, j) has a witness set. Therefore, if an edge e of P is witnessed in both directions then all of the edges of P are witnessed in both directions. Similarly, if e is witnessed only in the direction $u \to v$ then all the edges of P are witnessed in the direction $u \to v$.

In proving Theorem 5, for each edge $e = (i, j)$ of OPT, we first generated the trees $T_i(e)$ and/or $T_j(e)$ in $W(i, j)$ and/or $W(j, i)$ such that all leaf-nodes of the trees are player nodes and all non-leaf nodes are non-player nodes. Once the payment algorithm have paid for the cost of all the edges in $W(i,j)$ and/or $W(j,i)$, the algorithm asked only the players at the leaves of $W(i,j)$ and/or $W(j,i)$ to pay for the cost of e.

For the case when W is not laminar, we will modify the generation of the trees as follows. First contract each path P of OPT composed of degree 2 non-player nodes into a single edge f. Let us call G^* to this new graph. For each edge $f = (u,v)$ of G^*(which corresponds to an edge or a path of OPT), generate the trees $T_u(f)$ and/or $T_v(f)$ in $W(u,v)$ and/or $W(v,u)$ such that all leaf-nodes of the trees are player nodes and all non-leaf nodes are non-player nodes as explained above. Observe that Lemma 10 and Lemma 11 trivially holds on G^* since the set of smallest witness sets of the edges of G^* is laminar. For each edge $f = (u,v)$ of G^* that corresponds to a path of OPT, uncontract f back with the actual path P of OPT. Note that we haven't generated the trees for actual edges of OPT yet. Let $e = (i,j)$ be an arbitrary edge of P and without loss of generality let i be the node closer to u. If the tree generation algorithm generated a tree $T_u(f)$ for f on G^* then the tree $T_i(e)$ is obtained by appending the portion of P between u and i to the tree $T_u(f)$. Note that $T_i(e)$ is actually a tree since it is obtained by appending a path outside the smallest witness set $W(u,v)$ on G^* with a tree contained in $W(u,v)$. We obtain $T_j(e)$ similarly if the tree generation algorithm have generated $T_v(f)$ for f on G^*. Observe that the smallest witness set of $W(i,j)$ on OPT is composed of the smallest witness set $W(u,v)$ on G^* and the set of nodes of P between u and i. Therefore, the set of terminals in $W(i,j)$ on OPT are exactly the same set of terminals in $W(u,v)$ on G^*. Since the other connection path of any player in $W(u,v)$ on G^* does not use any edge of $T_u(f)$ or f, the other connection path of any player in $W(i,j)$ on OPT does not use any edge of $T_i(e)$.

To pay for the cost of the edges of OPT, we pick an arbitrary edge $e = (i,j)$ whose payment scheme has not been decided yet. If e is not an edge of a path P composed of degree 2 nonterminal nodes, we follow exactly the same steps as in Algorithm 2. Therefore, we consider the case where e is an edge of a path P on OPT between u and v, of which all intermediate nodes are degree 2 non-player nodes. Assume e is witnessed in one direction, without loss of generality in the direction $i \to j$, where i is the adjacent node to e that closer to u on path P. Then we decide the payment of all the edges of P one after another starting from the edge closest to u till the edge incident on v. Observe that all of those edges has only one tree and the trees of all these edges contain the same set of players. If the algorithm can pay for the cost of all the edges of P then it proceeds with the next edge. Otherwise, we can construct a network cheaper than OPT with exactly the same arguments in the proof of Theorem 5. If e is witnessed in both directions, then we ask the players of $T_i(e)$ to pay for the edges of P one after another starting from the edge incident to u and we ask the players of $T_j(e)$ to pay for the edges of P one after another starting from the edge incident to v. If these payments “meet in the middle” to cover the cost of all edges in P, then we have paid for e and the algorithm will proceed with selecting another unpaid edge, and if they do not, then we can construct a network cheaper than OPT with exactly the same arguments in the proof of Theorem 5. Observe that the payment scheme we generate actually has a value $p_i(e)$ for every edge of OPT, not the contracted edges of G^*.

■

4.3 Polynomial-Time Results

We have shown that there exists a 2-approximate Nash equilibrium that is as good as OPT. Since computing OPT is computationally infeasible, we present the following result.

Theorem 7 *Suppose we have a Survivable Connection Game and an α-approximate socially optimal graph G_α. Then for any $\epsilon > 0$, there is a polynomial time algorithm which returns a $2(1+\epsilon)$-approximate Nash equilibrium on a feasible graph G', where $c(G') \leq c(G_\alpha)$.*

Proof. The proof of Theorem 6 suggests an algorithm which forms a cheaper network whenever a stable solution (a 2-approximate Nash equilibrium) cannot be found. The proof followed by contradiction since the network at hand was optimal. The algorithms algorithms and the proof of the theorem are based on two properties of OPT: every edge has a witness set, and the set of smallest witness sets is laminar with path exceptions. Observe that both of these properties hold for any minimal feasible network, i.e., any feasible network such that removal of any edge will violate feasibility, since every edge has a witness set by definition of minimality and the proof of Theorem 1 is general enough to be valid for all minimal feasible networks, though it was stated for only socially optimal networks. Therefore; given an α-approximate socially optimal graph G_α, Theorem 5 suggests an algorithm which forms a cheaper network whenever a stable solution cannot be found.

The proof is based on following this suggested algorithm to obtain a cheaper network whenever a stable solution cannot be found. However, the improvements we consider should be substantial enough to ensure the time-bound, while they should be small enough to ensure the approximation ratio.

To find a $2\,(1+\epsilon)$-approximate Nash equilibrium, i.e., a solution where no player can reduce its cost by more than a factor of $2\,(1+\epsilon)$ by taking any deviation, we start by defining $\gamma = \frac{c(G_\alpha)\epsilon}{\alpha(1+\epsilon)m}$, where m is the total number of edges of the graph. We now use Algorithm 2 to pay for all but γ of each edge in G_α. Since G_α is not optimal, it is possible that even with the γ reduction in price there will be some edge e that the players are unwilling to pay for. If this happens, the algorithm suggested by the proof of Theorem 5, indicates how we can rearrange G_α to reduce its cost. If we modify G_α in this manner, it is easy to show that we have reduced the cost by at least γ.

Each call to the payment algorithm can be made to run in polynomial time. Since each call which fails to form the payments reduces the cost by γ, we can have at most $\frac{\alpha(1+\epsilon)m}{\epsilon}$ calls. Therefore, in time polynomial in m and ϵ^{-1}, we obtained a network G' with $c\,(G') \leq c\,(G_\alpha)$ such that G' is a stable solution (2-approximate Nash equilibrium) if the cost of its edges were decreased by γ. (There exists constant factor approximation algorithms for this problem, including Jain's 2-approximation algorithm [22], i.e., alpha is a small constant).

For all players and for each edge e in G', we now increase $p_i(e)$ in proportion to p_i so that e is now fully paid for. Now clearly G' is fully paid for. Observe that the payment player i makes is increased to $\frac{c(G')p_i}{c(G')-m'\gamma}$, where m' denotes the number of edges in G'. To see that this is an $2(1+\epsilon)$-approximate Nash equilibrium, note that player i would not gain more than a factor of 2 by deviating before her payment was increased. Therefore, the cost of the best deviation of player i is at least $\frac{p_i}{2}$. Therefore, player i can gain at most a factor of

$$\frac{2c\,(G')}{c\,(G') - m'\gamma} \leq \frac{2c\,(G')}{c\,(G') - \frac{m'c(G_\alpha)\epsilon}{\alpha(1+\epsilon)m}} \leq \frac{2c\,(G')}{c\,(G') - \frac{c(G')\epsilon}{(1+\epsilon)}} = 2(1+\epsilon).$$

■

5 Smallest Witness Sets are Laminar with Path Exceptions

We now prove Theorem 1. Let S and T be two smallest witness sets of the edges of OPT. Since laminarity does not hold in general, all $S-T, T-S$ and $T \cap S$ can be nonempty, however as we will demonstrate below, if any two smallest witness sets intersect, then their intersection will be in one of two forms which enables us to generalize our results.

Let p_1, p_2, p_3, p_4, p_5 and p_6 denote the number of edges between these 3 sets and the exterior as depicted in Figure 10.

Observe that;

$$p_1 + p_2 + p_5 + p_6 = 2 \tag{4}$$

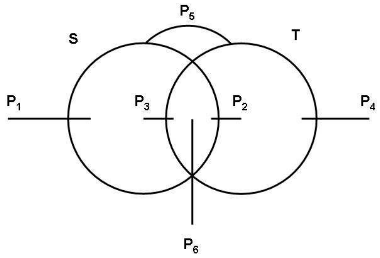


Fig. 10 General topology of two intersecting sets

and

$$p_3 + p_4 + p_5 + p_6 = 2 \tag{5}$$

since S and T are witness sets and therefore will have exactly 2 boundary edges.

Also;

$$p_1 + p_4 + p_6 \geq 2 \tag{6}$$

since there has to be at least 2 edges to leave $S \cup T$ for feasibility of the players in S and T. Observe that equations 4 and 6 together imply

$$p_4 \geq p_2 + p_5 \tag{7}$$

where equations 5 and 6 together imply

$$p_1 \geq p_3 + p_5. \tag{8}$$

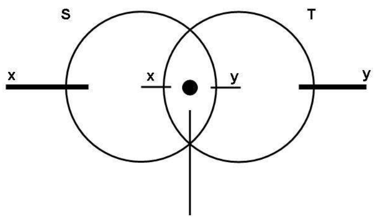


Fig. 11 Two intersecting witness sets that have a terminal in the intersection.

Assume there exists a terminal in $S \cap T$. Then; $p_2 + p_3 + p_6 \geq 2$ due to feasibility of this terminal. Observe that this equation together with equations 4 and 5 would imply $p_3 \geq p_1 + p_5$ and $p_2 \geq p_4 + p_5$ which would imply $p_1 = p_3, p_2 = p_4$ and $p_5 = 0$ when combined with equations 7 and 8. Therefore, the topology of these witness sets would be as described by Figure 11. In Figure 11 it is shown that $p_1 = p_3$ and they are both equal to x, and similarly $p_2 = p_4$ and they are both equal to some number y. Observe that S cannot be the smallest witness set of any of the edges leaving $S \cap T$ since $S \cap T$ is witnessing the same edges from the same side and has fewer nodes. For exactly the same reason, T cannot be the smallest witness set of any of the edges leaving $S \cap T$. Therefore, S is witnessing one of the x edges between $S - T$ and exterior while T is witnessing one of the y edges between $T - S$ and the exterior as depicted in Figure 11 (where the edges that S and T could be witnessing are shown in bold). Note that since $x + y + p_6 = 2$ (a smallest witness set will have exactly 2 boundary edges) and both x and y are greater than or equal to 1, $x = y = 1$.

Consider now the case where both $S - T$ and $T - S$ have at least one terminal. From feasibility of these terminals, we obtain $p_1 + p_3 + p_5 \geq 2$ and $p_2 + p_4 + p_5 \geq 2$ which in turn leads to $p_1 \geq p_4 + p_6$ and $p_4 \geq p_1 + p_6$ when combined with equations 5 and 4 respectively. When the last inequalities are combined together we obtain $p_6 = 0$ and $p_1 = p_4$ which implies $p_2 = p_3$ when we combine equations

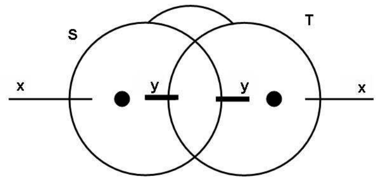


Fig. 12 Two intersecting witness sets that have a terminal in each difference.

4 and 5. Observe that S cannot be the smallest witness set of any of the $p_1 + p_5$ edges between $S - T$ and the exterior or $T - S$ because if S witnesses these edges, so does $S - T$ and it is smaller. Therefore, S witnesses one of the edges between $S \cap T$ and $T - S$. For exactly the same reason, T can only be the smallest witness set of one of the edges between $S - T$ and $S \cap T$. In Figure 12, the only edges that may be witnessed by S and T are drawn in bold. The key observation here is that there cannot be a terminal in $S \cap T$. To see this, assume the contrary. Then in Figure 12, $y \geq 1$ due to feasibility of this terminal. Since we know that $x \geq y$ from equation 7 or 8, and S has exactly 2 boundary edges, both x and y would be 1. However, observe that in this case S and T could not be the smallest witness sets of any of the edges since all possible edges would be witnessed by $S \cap T$ then as well. Therefore, $S \cap T$ does not have any terminal nodes. Observe that y has to be at least 1 since otherwise $S \cap T$ would not have any boundary edges and therefore would be empty. Since we know that $x \geq y$ from equation 7 or 8, and S has exactly 2 boundary edges, both x and y would be 1.

In the first case, when $S \cap T$ had at least one terminal that is demonstrated by Figure 11, we haven't mentioned anything about the nodes in $S - T$ and $T - S$. Due to the observations in Figure 12, we can now say that both of them cannot have a terminal since this would imply no terminals in $S \cap T$. Therefore, at least one of $S - T$ or $T - S$ would not have a terminal. In the complementary case, when $S \cap T$ does not have any terminal node, the situation is exactly as depicted in Figure 12, i.e., both $S - T$ and $T - S$ must have terminal nodes, since S and T are witness sets and they have to include terminal nodes by definition.

Since the cases above cover all possible situations, there are only two different situations possible when two smallest witness sets intersect but one is not contained in the other. In both of these cases, there is a "special" set which does not include any terminals and the number of edges entering and exiting the set are 1.

Since the special set does not have a terminal and there are exactly one entering and exiting edges, the special set is just a path of degree 2 non-terminals. After replacing the paths of degree 2 non-terminals with an edge the set smallest witness sets become laminar, and the desired structural properties hold, since all the nodes in the "special set" will disappear. Therefore, after replacing the paths of degree 2 non-terminals with an edge, the special set will be empty, which implies that at least one of $S - T$, $S \cap T$ or $T - S$ is empty.

5.1 Smallest Witness Sets are Unique

We will now prove Lemma 1, i.e., if an edge $e = (i, j)$ is witnessed in the direction say $i \rightarrow j$ (i.e., arc (i, j) is witnessed) then the smallest witness set of e in that direction, $W(i, j)$, is unique.

Let W_1 and W_2 be two smallest witness sets of an arbitrary arc (i, j). As shown in the proof of Theorem 1; for any 2 smallest witness sets S and T, all $S - T, T - S$ and $T \cap S$ can be nonempty since laminarity does not hold in general, however the intersection of S and T can be in one of the two forms depicted in Figure 11 and Figure 12. However, observe that in both of these forms, S and T are smallest witness sets of different edges. Therefore, since W_1 and W_2 are smallest witness sets of the same edge, all $W_1 - W_2, W_2 - W_1$ and $W_2 \cap W_1$ cannot be nonempty, i.e., they are either disjoint or one of them is a subset of the other. Since W_1 and W_2 are 2 smallest witness sets of the same arc (i, j), then i is included in both W_1 and W_2 (so W_1 and W_2 are not disjoint) and they contain an equal number of nodes. Therefore, $W_1 = W_2$.

6 Discussion and Open Problems

For the case where the connection requirements can be different for different players, we showed that Nash equilibria may not exist even for a single source, and that there may not even exist good approximate equilibria. For the connection requirement of 2 for all players, we have shown the existence of cheap exact equilibria when all nodes are player nodes, and of 2-approximate Nash equilibria as good as OPT if not. We do not know the price of stability for the general Survivable Connection Game, however. While we know there exist good 2-approximate Nash equilibria, there may still exist good exact stable solutions, even without the assumption that a player is allowed to only change payments on a single path at a time, and without all nodes being terminals.

An interesting future direction is to extend our results for the case where each player wishes to connect to s through r edge-disjoint paths. In this case, the set of smallest witness sets is still laminar with path exceptions. Our results do not easily generalize to cover that case, however, although it is possible that similar techniques can be used to take advantage of these laminar properties. The multi-source version of our game, where not everyone is trying to connect to a common source node, is also of interest. Unfortunately, we are able to construct examples where the smallest witness sets do not form a laminar set system, and so entirely different techniques may be needed for the analysis of such games.

References

1. S. Albers, S. Eilts, E. Even-Dar, Y. Mansour, and L. Roditty. On Nash Equilibria for a Network Creation Game. *SODA*, 2006.
2. A. Agarwal, M. Charikar. Unpublished manuscript.
3. E. Anshelevich, B. Caskurlu. Exact and Approximate Equilibria for Optimal Group Network Formation. In *Proc. 17th Annual European Symposium on Algorithms*, 2009.
4. E. Anshelevich, A. Dasgupta, J. Kleinberg, É. Tardos, T. Wexler, T. Roughgarden. The Price of Stability for Network Design with Fair Cost Allocation. In *Proc. 45th Annual IEEE Symposium on Foundations of Computer Science*, 2004.
5. E. Anshelevich, A. Dasgupta, É. Tardos, T. Wexler. Near-Optimal Network Design with Selfish Agents. In *Proc. 35th ACM Symposium on Theory of Computing*, 2003.
6. E. Anshelevich, B. Shepherd, G. Wilfong. Strategic Network Formation through Peering and Service Agreements. FOCS 2006.
7. V. Bala and S. Goyal. A Non-Cooperative Model of Network Formation. In *Econometrica* 68(2000), 1181–1229.
8. C. Chekuri, J. Chuzhoy, L. Lewin-Eytan, J. Naor, and A. Orda. Non-cooperative multicast and facility location games. Proceedings of the 7th ACM Conference on Electronic Commerce (EC), Ann Arbor, Michigan (2006), pp. 72-81.
9. H. Chen, T. Roughgarden. Network Design with Weighted Players. In *SPAA* 2006.
10. G. Christodoulou and E. Koutsoupias. On the price of anarchy and stability of correlated equilibria of linear congestion games. ESA, 2005.
11. A. Epstein, M. Feldman, and Y. Mansour. Strong Equilibrium in Cost-Sharing Connection Games. *EC* 2007.
12. A. Fabrikant, A. Luthra, E. Maneva, S. Papadimitriou, and S. Shenker. On a Network Creation Game. In *PODC*, 2003.
13. A. Fabrikant, C. Papadimitriou, K. Talwar. The complexity of pure Nash equilibria. In *STOC*, 2004.
14. J. Feigenbaum, C. Papadimitriou, S. Shenker. Sharing the Cost of Multicast Transmissions. *Journal of Computer and System Sciences* 63 (2001), 21–41.
15. A. Fiat, H. Kaplan, M. Levy, S. Olonetsky, R. Shabo. On the Price of Stability for Designing Undirected Networks with Fair Cost Allocations. *Proceedings of ICALP* 2006, pp. 608–618.
16. Shai Herzog, Scott Shenker, Deborah Estrin. Sharing the "Cost" of Multicast Trees: An Axiomatic Analysis. *IEEE/ACM Transactions on Networking*, Dec. 1997.
17. M. Hoefer. Non-cooperative Facility Location and Covering Games. In *ISAAC* 2006.
18. M. Hoefer. Non-cooperative Tree Creation. In *MFCS* 2006.

19. M. Hoefer, P. Krysta. Geometric Network Design with Selfish Agents. *COCOON* 2005.
20. R. Holzman, N. Law-Yone. Strong Equilibrium in congestion games. *Games and Economic Behavior,* 21, 1997.
21. M. Jackson, A survey of models of network formation: stability and efficiency. *Group Formation in Economics: Networks, Clubs and Coalitions*, eds. G. Demange and M.Wooders, Cambridge Univ. Press.
22. K. Jain. A Factor 2 Approximation Algorithm for the Generalized Steiner Network Problem. *Combinatorica,* 21, 2001.
23. K. Jain and V. Vazirani. Applications of Approximation Algorithms to Cooperative Games. In *STOC*, 2001.
24. R. Johari and J. Tsitsiklis. Efficiency loss in a network resource allocation game. *Mathematics of Operations Research,* 29 (3): 407–435.
25. J. Könemann, S. Leonardi, and G. Schäfer. A group-strategyproof mechanism for Steiner forests. *SODA* 2005.
26. Y.A. Korillis, A.A. Lazaar, and A. Orda. Achieving Network Optima Using Stackleberg Routing Strategies. *IEEE/ACM Transactions on Networking,* 5(1), 1997.
27. D. Monderer, L. Shapley. Potential Games. *Games and Economic Behavior* 14(1996), 124–143.
28. N. Nisan, T. Roughgarden, É. Tardos, and V. V. Vazirani (eds.), *Algorithmic Game Theory,* Cambridge University Press.
29. R. W. Rosenthal. The network equilibrium problem in integers. *Networks*, 3:53-59, 1973.
30. T. Roughgarden. *Selfish Routing and the Price of Anarchy,* MIT Press.
31. É. Tardos. Network Games. In *Proceedings of the 36th Annual ACM Symposium on the Theory of Computing*, 2004.