

Near-Optimal Network Design with Selfish Agents

Nuo Xu*

25.11.2024

1 Introduction

1.1 Classic Generalized Steiner Tree and Steiner Forest problem

In the classic generalized Steiner tree, we are given an undirected graph $G = (V, E)$ with non-negative edge cost $c_e \geq 0$ for every edge $c_e \in E$ and a set of terminals $R \subseteq V$. The goal is to compute the minimum-cost subgraph that spans all terminals. Whereas in the Steiner forest problem, we are given a collection of disjoint subsets of $V : V_1, V_2, V_3, \dots, V_n$. The goal is to compute a subgraph that any two vertices that belong to the same subset V_i are connected.

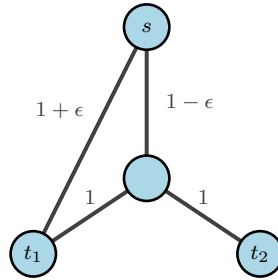
The Steiner Tree problem is a Steiner Forest problem with a single subset of V . We will denote the optimal solution of both problems as OPT. Finding OPT is NP-hard.

1.2 Introduction of Selfish Agents

In real life, the design of network often involves selfish agents. Imaging the train routes among major cities in Europe, Paris, Munich, Berlin, and Milan. Every city would like to connect to all other cities. However, there won't be a single company that can design the minimum cost network and implement such network. Instead, every city would like to seek the route that they can pay less rather than the route that requires more.

Given an undirected graph G with non-negative edge costs and N players, each player is interested in connecting a set of terminals (nodes in G) via buying a subgraph of G . Players offer each edge in G certain amount of money, and they would like to pay a little as possible.

*Department of Computer Science, Technical University of Munich. ge74mis@tum.de



1.3 Formal Definition of Connection Game

Here we formally define the connection game for N players as following:

- An undirected graph $G = (V, E)$.
- Non-negative edge cost $c_e \geq 0$ for every edge $c_e \in E$.
- A subset of V for each player that they must connect to.
- A payment function p_i indicates that player's payment strategy. $p_i(e)$ is the contribution that player i would like to offer for edge e .

If the sum of payment on certain edge e is larger than the cost on that edge c_e , this edge is considered as bought and can be used by all players no matter they contribute to it or not. The goal of all players is to connect all of their terminals. If in the end, a player's terminals are not fully connected, they will face an infinite penalty.

2 Nash Equilibrium in Connection Game

2.1 Existence of Pure Nash Equilibrium

A crucial idea of modern game theory introduced by von Neumann is mixed strategies i.e. randomization of pure strategies. Von Neumann's work mostly stayed in zero-sum games. John Nash further extended game theory into N players general games.

Theorem 2.1 (Nash's theorem). *With randomization, any game with finite number of players and actions has a mixed-strategy of Nash equilibrium.*

Definition 2.1 (Nash Equilibrium in Connection Game). *A Nash equilibrium of the connection game is a payment function p such that, if players offer payments p , no player has an incentive to deviate from their payment.*

Nash's theorem states that any finite game has at least one mixed strategies Nash equilibrium but no guarantee on pure strategy equilibrium. Here it is not hard to see that pure Nash equilibrium may not exist in the connection game.

2.2 Some Properties of Nash Equilibrium in Connection Game

- G_p is a forest.
- Let T^i be the smallest tree in G_p connecting all terminals of player i , then player i only contributes to edges in T^i .
- Each edge is either bought or not at all.

2.3 Fractional Nash Equilibrium

Definition 2.2 (Fractional Nash Equilibrium). *If Nash equilibrium requires players to split cost of some edge, such Nash Equilibrium is fractional.*

2.4 Price of Anarchy and Stability

As mentioned in Section 1, the introduction of selfish agents can lead to worse equilibrium than the best centralized optimum. The question is how bad an equilibrium can be.

Definition 2.3 (Price of Anarchy). *The price of anarchy of connection game is defined as the ratio of the cost of worst Nash equilibrium over the best centralized design.*

$$P_A = \frac{\sum_1^N p_i(e)}{OPT}$$

The price of anarchy can be as worst as N .

Definition 2.4 (Price of Stability). *Price of stability is a complementary concept of price of anarchy which evaluate how good the best equilibrium can be.*

$$P_A = \frac{\sum_1^N p_i(e)}{OPT}$$

3 Single Source Games

3.1 Definition of Single Source Games

We have already known that pure Nash Equilibrium might not exist in the general version of connection game i.e. N players with multiple terminals that they would like to connect. We have also already learned that determining the existence of Nash equilibrium in connection game is NP-hard. However, in the single source connection game, Nash equilibrium is determined to exist.

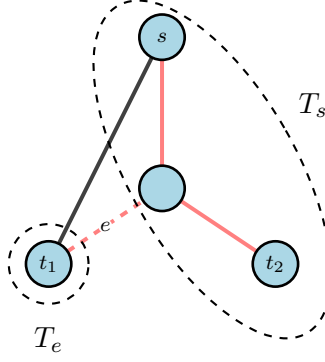
In the single source game, we only allow every player i has one unique terminal t_i that they all would like to connect a common terminal s . This can be considered as a special version of Steiner tree problem where $R = \{s, t_0, t_1, \dots, t_N\}$.

Definition 3.1 (Single source Game). *A single source game is a game in which all players share a common terminal s and in addition, each player i has exactly one other terminal t_i .*

As we have talked before, it is not much of our interests to study the worst equilibrium as it can be really bad. Instead, we would like to see how good the best equilibrium can be. One natural thought is can we achieve an equilibrium that equals to the OPT ? Now suppose the minimum cost Steiner tree T^* over all players' terminals is given on a silver plate, we are going to drive an algorithm that gradually assigns each player i a payment strategy on edge e . Then we are going to prove that the final payment function p will end up as an equilibrium and yield price of stability as 1.

It is trivial to see that every leaf node in T^* is a terminal as if not it is possible to simply discard this node and corresponding edge without affecting the connection of any terminals.

Fact 1. *Every leaf node in T^* is a terminal.*



3.2 Payment Strategy Design

We know that we are going to assign every edge in T^* but the problem is which edge we are going to assign first. The intuition is to assign edges that are known for sure which players are interested in. First, let T^* be rooted from s . We will visit T^* in reverse breadth-first-search order. For every edge e , suppose e is removed from T^* , T^* will be divided into two subtrees. Let the subtree that does not contain s be T_e and the one contains s be T_s . It is clear to see that players in T_s would not be interested in paying anything for e as discarding e will not affect their connection to s at all. Suppose a player can only connect to s through one path that no edges are shared by any other players, they would have to pay the full cost of every edge on that path. The main loop in our algorithms goes as following: we loop every edge $e \in T^*$ in reverse BFS order, then for every player $i \in T_e$ we determine $p_i(e)$.

Next we going to assign $p_i(e)$ so that players would stay in T^* . In every iteration, we modify the cost of every edge according to the following rules:

- edges $f \in T_e$ cost $p_i(f)$
- edges $f \in T^* \setminus T_e$ costs 0
- edges $f \notin T^*$ cost $c(f)$

Then we find the cheapest alternative path A_i from t_i to s from $F \setminus \{e\}$ under modified cost. Now we consider the cost player i needs to pay if they choose to deviate, $c(A_i) - \sum_{f \in T^*} p_i(T^*)$, and $c(e) - \sum_{j \in T_i, j \neq i} p_j(e)$. By choosing the minimum between these two values, we ensure that player i will never contribute to e more than the cost of deviation. In another word, it is always cheaper for players to stay in T^* . Therefore, p is indeed a Nash equilibrium.

Corollary 3.1. *The final payment function produced by the above algorithm is indeed a Nash equilibrium.*

3.3 Proof of Price of Stability Always Being 1

To prove that the price of stability is always 1, we need to demonstrate two things. First, every player is not going to buy any other edges that are not in T^* . Second, in the end of the game, every edge in T^* is fully paid. The first proof is relatively straight forward whereas the second one requires more work.

Corollary 3.2. *For any edge e , players with terminals in T_e will be willing to pay for e , i.e. they will choose to stay in OPT rather than deviation.*



To prove that every edge in T^* is fully paid in the end, we are going to assume that the following lemma is true.

Suppose A_i is i 's alternative path at some stage of the algorithm. Then suppose there are two nodes v and w on A_i that divides this path into three parts. Let edges on A_i from t_i to v be f_1 , from v to w be f_2 , from w to s be f_3 .

Lemma 3.1. *There must exist a pair of $\{v, w\}$ such that $f_1 \in T_e$, $f_2 \notin T^*$, $f_3 \in T^* \setminus T_e$.*

Lemma 3.2. *All edges in T^* are bought, i.e. $\sum p_i(e) = c(e)$ for any $e \in T^*$.*

Suppose there exists some edge e such that after all players in T_e have contributed to that edge, $p_i(e) < c(e)$, we can prove that T_e costs more than $\bigcup_{i \in T_e} A_i$ which is a contradiction of T^* is OPT .

3.4 Approximate Nash Equilibrium

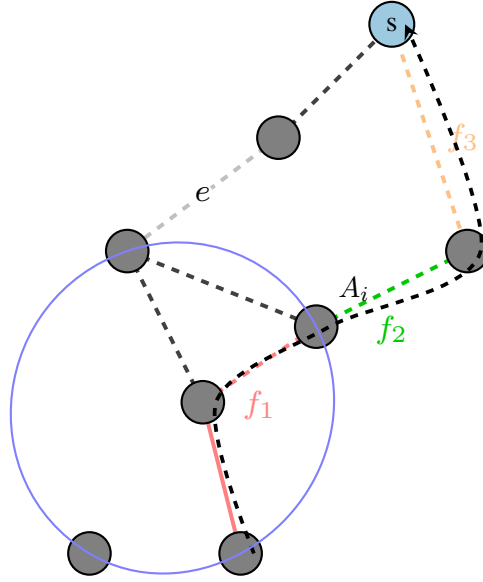
Although we proved that the determined existence of Nash equilibrium in single source game, the algorithm presented is not feasible as the finding the minimum cost Steiner tree is NP-hard by itself.

Definition 3.2 (Approximate Equilibrium). *A $(1 + \epsilon)$ -approximate Nash Equilibrium is a payment function p such that player i would not deviate their payment by a factor of $1 + \epsilon$.*

Theorem 3.1. *Given a single source game and α approximation of minimum-cost Steiner tree T ,*

4 Lower Bound of Approximate Nash Equilibrium

It has been proven that it is NP-hard to approximate Steiner tree problem within ratio 96/95. However, it is possible to obtain an approximate minimum-cost Steiner tree by modeling this problem into linear programming and randomized rounding. The closest approximation ratio has been obtained so far is 1.55.



4.1 Hardness Proof

Under suitable complexity assumption (unless $NP = coNP$), it is proven by xxxxx that computing Nash Equilibrium in non-zero-sum games is PPAD-hard (i.e. there is no polynomial-time algorithm).

References

- [1] Vazirani Vijay V. *Approximation Algorithms*. Chapters 3.1 and 22, Springer, 2003.