A Note on Online Steiner Tree Problems

Gokarna Sharma* Costas Busch*

Abstract

We introduce and study a new Steiner tree problem variation called the *bursty Steiner tree problem* where new nodes arrive into bursts. This is an online problem which becomes the well-known *online Steiner tree problem* if the number of nodes in each burst is exactly one and becomes the *classical Steiner tree problem* if all the nodes that need to be connected appear in a single burst. In undirected graphs, we provide a tight bound of $\Theta(\min\{\log k, m\})$ on the competitive ratio for this problem, where k is the total number of nodes to be connected and m is the total number of different bursts. In directed graphs of bounded edge asymmetry α, we provide a near tight competitive ratio for this problem. We also consider a bursty variation of the terminal Steiner tree problem and provide the upper bound of $\min\{4\rho, 3\lambda m\}$ and the lower bound of $\min\{\rho/2, m/4\}$ on the competitive ratio in undirected complete graphs, where λ is the current best approximation for the terminal Steiner tree problem and $\rho = \frac{1}{2}\log k$. These are the first such results which provide clear performance tradeoffs for the novel Steiner tree problem variations that subsume both of their online and classical versions.

1 Introduction

The Steiner tree problem [8] and its variations have been extensively studied in the literature, e.g. [1–4, 7–9, 12, 13]. Given a set of k vertices $\mathcal{S} = \{v_1, v_2, \ldots, v_k\}$ in an arbitrary undirected graph $G = (V, E)$ (vertices in $\mathcal{S} \subseteq V$ are called *terminals* and other vertices $V \backslash \mathcal{S}$ are called *Steiner* or *optional* vertices) with a length (or weight) function $\mathfrak{w} : E \to \mathbb{R}^+$ on the edges, a *Steiner tree* for this terminal set $\mathcal{S}$ is a connected subgraph T that connects all k terminals minimizing the length. The length of a Steiner tree is defined to be the sum of the lengths of all its edges. Steiner trees and their variations are very useful due to their applications in VLSI design, network and group communication, multipoint and multicast routing, video broadcasting, computational biology, and so on; see [8, 9, 12].

The extensively studied Steiner tree problem is the *classical Steiner tree problem* (STP) [4, 8, 13]. STP is the problem of finding the minimum Steiner tree T (or a good approximation of it) for a set $\mathcal{S}$ of k terminals that are known in advance. Here the approximation means the ratio of the length of the tree T produced by an algorithm and the length of the optimal tree OPT for the terminals in $\mathcal{S}$. STP is an NP-hard [8] and MAX SNP-hard problem [3]. The current best approximation is $\sigma = \ln(4) + \epsilon < 1.39$ due to Byrka et al. [4].

One variation of STP that is also very well-studied is the *online Steiner tree problem* (OSTP) [2, 9]: given a sequence $\mathcal{S}$ of k terminals in G that appear one at a time sequentially, construct online a connected subgraph T that connects all the terminals minimizing its length. There exists a lower bound of ρ and a very simple greedy algorithm with the upper bound of 2ρ due to Imase and Waxman [9] in undirected graphs, where $\rho = \frac{1}{2}\log k$.

As actual communication networks may contain links that are asymmetric in the quality of service they offer, STP and OSTP are also studied in graphs where asymmetric links are present, e.g. [1, 2, 7, 12]. Let A denote the set of pairs of vertices in V such that if the pair u, v is in A, then either $(u, v) \in E$ or $(v, u) \in E$ (i.e., there is an edge from u to v or an edge from v to u or both). Then, the edge asymmetry is defined as $\alpha := \max_{\{u,v\} \in A} \frac{\mathfrak{w}(u,v)}{\mathfrak{w}(v,u)}$. According to this measure, undirected graphs are the class of graphs of asymmetry $\alpha = 1$. Directed graphs have asymmetry $\alpha = \infty$ when there is at least one pair of vertices u, v such that $(u, v) \in E$ but $(v, u) \notin E$. Between these two extremes, *graphs of bounded asymmetry* exist which are useful in modeling networks with certain type of link heterogeneity [2, 12]. The problem is then to find a minimum length *arborescence* rooted at some node $\wp$ which spans all the vertices in $\mathcal{S}$. For STP, there is an algorithm with a tight approximation factor of $\Theta(\min\{\alpha, k\})$ [7] in directed graphs of asymmetry α. For OSTP, there exists an approximation algorithm with the upper bound of $\mathcal{O}\left(\min\left\{\max\left\{\alpha\frac{\log k}{\log \alpha}, \alpha\frac{\log k}{\log\log k}\right\}, k\right\}\right)$ and the lower bound of $\Omega\left(\min\left\{\max\left\{\alpha\frac{\log k}{\log \alpha}, \alpha\frac{\log k}{\log\log k}\right\}, k^{1-\epsilon}\right\}\right)$ from a long series of work [2]. These bounds are optimal for $\alpha \in \mathcal{O}(k^{1-\epsilon})$ or $\alpha \in \Omega(k)$.

A STP is a *terminal Steiner tree problem* (TSTP) or a *full Steiner tree problem* if all terminals $\mathcal{S}$ are the leaves of the Steiner tree [11]. Terminal Steiner trees have applications in the reconstruction of evolutionary trees in biology. TSTP is shown to be NP-hard and MAX SNP-hard [10], even when the lengths of edges

*School of Electrical Engineering and Computer Science, Louisiana State University, {gokarna, busch}@csc.lsu.edu

are restricted to be either 1 or 2 [11]. There exists an algorithm which achieves the approximation ratio of $\lambda = 2\sigma - \frac{(\sigma\beta^2-\beta\sigma)}{(\beta+\beta^2)(\sigma-1)+2(\beta-1)^2}$ for TSTP and this is the current best for $\beta = 3$ and 4 [5].

In this paper, we consider following two variations:

i. **BSTP** (Bursty Steiner tree problem)
Instance: A graph (directed or undirected) $G = (V, E)$ with $\mathfrak{w} : E \to \mathbb{R}^+$, and a subset $\mathcal{S} \subset V$, $|\mathcal{S}| = k$, of terminals appearing online in a sequence of m groups (or bursts) $\mathcal{B} = \{B_1, B_2, \ldots, B_m\}, B_i \subseteq \mathcal{S}, 1 \leq i \leq m$, one burst at a time.
Question: Find a Steiner tree (arborescence in directed case) for $\mathcal{S}$ in G with minimum length.

ii. **BTSTP** (Bursty terminal Steiner tree problem)
Instance: A complete undirected graph $G = (V, E)$ with $\mathfrak{w} : E \to \mathbb{R}^+$, and a proper subset $\mathcal{S} \subset V, |\mathcal{S}| = k$, of terminals appearing online in a sequence of m groups (or bursts) $\mathcal{B} = \{B_1, B_2, \ldots, B_m\}, B_i \subseteq \mathcal{S}, 1 \leq i \leq m$, one burst at a time, where the length function $\mathfrak{w}$ is metric.
Question: Find a terminal Steiner tree for $\mathcal{S}$ in G with minimum length.

BSTP (similarly BTSTP) is a new natural variation of both STP (TSTP) and OSTP (online TSTP (OTSTP)). BSTP and BTSTP model the online situations in which terminals in $\mathcal{S}$ arrive in groups, called *bursts*, one burst after another (instead of only one new node at a time in OSTP and OTSTP). We have that terminals in each burst $B_i \subseteq \mathcal{S}$ and $\cup_{i=1}^{m} B_i = \mathcal{S}$. BSTP (BTSTP) become OSTP (OTSTP) if $|B_i| = 1$ for $1 \leq i \leq m$ (one terminal in each burst) and becomes STP (TSTP) if $m = 1$ (a single burst). These problems are interesting in the sense that they provide clear trade-offs to both of their classical and online versions. Since BSTP (BTSTP) allows more than one nodes to join the connection at a time, it provides a flexibility to OSTP (OTSTP). Therefore, BSTP (BTSTP) subsumes its two existing variations STP (TSTP) and OSTP (OTSTP) and generalizes them by capturing an intermediate variation which is not completely online like OSTP (OTSTP) and not completely offline like STP (TSTP).

We provide lower and upper bounds for both BSTP and BTSTP. We consider BSTP in both undirected and directed graphs of bounded asymmetry α. We consider BTSTP in complete graphs whose length function is metric. By definition, any terminal Steiner tree $T = (\mathcal{S}, E_{\mathcal{S}})$ for $\mathcal{S}$ in $G = (V, E)$ contains no edge $\{(u, v)|u, v \in \mathcal{S}, u \neq v\}$ in $E_{\mathcal{S}}$. Moreover, BSTP (BTSTP) is online, we need to construct a Steiner tree (a terminal Steiner tree) incrementally, as terminals (bursts) appear one after another. At the end of step i, a minimum length tree T_i of terminals (bursts) is constructed without the knowledge of terminals v_j (terminal set B_j) for any $j > i$. Denote by $\mathcal{B}_i = \{B_1, B_2, \ldots, B_i\} \subseteq \mathcal{S}$ the sequence of bursts that arrived in the system from step 1 up to step $i \leq m$. Given an instance I of BSTP (BTSTP) and an algorithm $\mathcal{A}$, let $\mathcal{C}_{\mathcal{A}}(T_i)$ be the length of the tree generated by $\mathcal{A}$ for the terminal burst set $\mathcal{B}_i$ and let OPT be the length of an optimal Steiner tree (terminal Steiner tree) for $\mathcal{B}_i$. Then, the performance of $\mathcal{A}$ on instance I of BSTP (BTSTP) is measured by $CR_{\mathcal{A}}(I) = \max_{1 \leq i \leq m} \frac{\mathcal{C}_{\mathcal{A}}(T_i)}{OPT}$, which is the competitive ratio of $\mathcal{A}$. For all instances I, we consider the supremum $CR_{\mathcal{A}}(I)$ over all instances I, which is denoted by $CR_{\mathcal{A}}$. When the context is clear, we denote $C_{\mathcal{A}}(T_i)$ by $C_{\mathcal{A}}(\mathcal{B}_i)$ and $CR_{\mathcal{A}}(I)$ by $CR_{\mathcal{A}}(\mathcal{B})$.

Contributions: The first result is the tight bound of $\Theta(\min\{\log k, m\})$ on the competitive ratio of any deterministic algorithm for BSTP in undirected graphs. The second result is the upper bound of $\mathcal{O}\left(\min\left\{\max\left\{\alpha\frac{\log k}{\log \alpha}, \alpha\frac{\log k}{\log\log k}\right\}, k, \alpha \cdot m\right\}\right)$ and the lower bound of $\Omega\left(\min\left\{\max\left\{\alpha\frac{\log k}{\log \alpha}, \alpha\frac{\log k}{\log\log k}\right\}, k^{1-\epsilon}, \max\left\{\alpha, \alpha\frac{m}{\log \alpha}\right\}\right\}\right)$ on the competitive ratio of any deterministic algorithm for BSTP in directed graphs of bounded edge asymmetry α, where $0 < \epsilon < 1$ is a small constant. For $\alpha \in \mathcal{O}(k^{1-\epsilon})$ or $\alpha \in \Omega(k)$, this result is tight when $m \geq \log k$, and tight within a $\log \alpha$ factor when $m < \log k$. The third result is the lower bound of $\min\{\rho/2, m/4\}$ and the upper bound of $\min\{4\rho, 3\lambda m\}$ on the competitive ratio of any deterministic algorithm for BTSTP in undirected complete graphs, where λ is the approximation of TSTP and $\rho = \frac{1}{2}\log k$. All of these results are surprising because they show that algorithms whose competitive ratios depend on logarithmic of m (the number of bursts) do not exist for these bursty problems; however, for their purely online versions with one terminal at a time the competitive ratio of algorithms depend on the logarithm of k.

The main idea behind lower bound proofs is to construct online an adversarial set of m bursts of k terminals such that Steiner tree generated by any algorithm has length at least $\Omega(\min\{\log k, m\}$ times the optimal tree length. The main idea behind upper bound proofs is to show that there are algorithms which produce Steiner trees of length at most $\mathcal{O}(\min\{\log k, m\}$ times the optimal tree length in any scenario. Although we build on existing techniques, e.g. [7, 9], our results settle the performance efficiency of BSTP and BTSTP.

Paper organization: We give tight bounds for BSTP in undirected graphs in Section 2. In Section 3, we give almost-tight upper and lower bounds for BSTP in directed graphs of bounded edge asymmetry α. We then give tight bounds (within a constant factor) for BTSTP in undirected complete graphs in Section 4. Many proofs and details are deferred to a full version.

2 BSTP in Undirected Graphs

We first show that $\Omega(\min\{\log k, m\})$ is the lower bound and then give a deterministic approximation algorithm that matches the lower bound.

Lower bound: We create a sequence of instances I_ℓ for the terminal bursts that need to be connected online based on a sequence of graphs $G_\ell, \ell \geq 0$, borrowing the construction of Imase and Waxman [9]. We then apply an adversary argument to show that instances I_ℓ can be created such that the length for any algorithm $\mathcal{A}$ for BSTP matches our claimed result.

We construct graphs $G_\ell = (V_\ell, E_\ell), \ell \in \mathcal{Z}_0^+$, with a constant length function $\mathfrak{w}_\ell$ on the edges of G_ℓ. As an example, the construction of G_0, G_1, and G_2 is given in Fig. 1. The initial graph G_0 is the complete graph with only two nodes, say v_0 and v_1, and a single edge (v_0, v_1) with length 1, i.e., $\mathfrak{w}_0(v_0, v_1) = 1$. The only two nodes v_0 and v_1 in G_0 are called *level* 0 *nodes*. We now use a recursive definition to define graph G_ℓ, $\ell > 0$, based on graph $G_{\ell-1}$. We obtain G_ℓ from $G_{\ell-1}$ as follows. We introduce two distinct pair of nodes a and b for each edge $(u, v) \in G_{\ell-1}$ and replace the edge (u, v) with two paths (u, a, v) and (u, b, v). Here a path (x, y, z) is simply the concatenation of the edges connecting subsequent node pairs (x, y) and (y, z). There are two edges (u, a) and (a, v) connecting $u, v \in G_0$ following node a and, two edges (u, b) and (b, v) connecting $u, v \in G_0$ following node b. This path definition extends similarly for a path with more than two subsequent node pairs. We call these new nodes a, b introduced to get G_ℓ from $G_{\ell-1}$ the *level* ℓ *nodes* (see G_1 that is obtained from G_0 and G_2 that is obtained from G_1 in Fig. 1). Therefore, in $G_{i\geq 0}$, there are level 0 up to level i nodes. We denote by *level* of a node the level in G_i where that node belongs, e.g. the level of node a is 1 in Fig. 1. We assign the length of $1/2^\ell$ to each edge of G_ℓ, that is, $\mathfrak{w}_\ell(u, v) = 1/2^\ell$ for each edge (u, v) of G_ℓ (the length of each edge of G_1 is $1/2$ and the length of each edge of G_2 is $1/4$ in Fig. 1). Two level ℓ nodes that are added between any two level $\ell - 1$ nodes to obtain G_ℓ from $G_{\ell-1}$ are called *sister nodes*, e.g. a and b of G_1 in Fig. 1. Moreover, two nodes $u, v \in G_\ell$ are said to be i-*adjacent*, $0 \leq i \leq \ell$, if the level of both u and v is no more than i and there is a path from u to v which has no intermediate node from level j, $j \leq i$ [9].

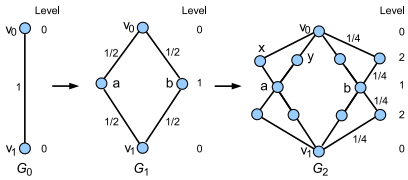


Figure 1: Illustration of sequences of graphs $G_\ell, \ell \geq 0$

Consider a sequence of m terminal bursts $\mathcal{B} = \{B_0, B_1, \ldots, B_m\}$ for graphs $G_i, i \geq 0$, respectively. Each B_i contains some specific number of level i nodes from $G_i, 0 \leq i \leq m$ (we give exact numbers soon). We create this sequence $\mathcal{B}$ in such a way that there exists a path p between v_0 and v_1 for all the nodes in $\mathcal{B}$ with length exactly 1 using an optimal algorithm.

We define a *minimal tree sequence* $\overline{\mathcal{T}} = \{\overline{T}_0, \overline{T}_1, \ldots, \overline{T}_m\}$ for graphs $G_i, i \geq 0$, with respect to a sequence of terminal bursts $\{B_0, B_1, \ldots, B_m\}$, such that, for $1 \leq i \leq m$, each tree $\overline{T}_i$ must contain tree $\overline{T}_{i-1}$ as a subgraph and connects all of the terminals in B_i. As we consider minimal tree sequence, no subgraph of $\overline{T}_i$ can satisfy this requirement. $\overline{\mathcal{T}}$ is constructed by an algorithm $\mathcal{A}$ based on the knowledge of only $\overline{T}_{i-1}$ and B_i in each step i to obtain $\overline{T}_i$. We choose the nodes to include in $\overline{T}_i$ based on the knowledge of $\overline{T}_{i-1}$ to maximize the length of $\mathcal{A}$.

Theorem 1 *The competitive ratio of every deterministic algorithm for BSTP is* $\Omega(\min\{\log k, m\})$ *in undirected graphs.*

Proof. When there is only one terminal in each burst B_i, BSTP becomes OSTP. Therefore, $\Omega(\log k)$ lower bound of Imase and Waxman [9] applies to BSTP. We now consider the case where the number of bursts $m < \lceil \log k \rceil$ for the set $\mathcal{B}$ of k terminals and prove the lower bound of $\Omega(m)$. We consider the burst set $\mathcal{B} = \{B_i, B_{i+1}, \ldots, B_{i+m-1}\}$, where $i \geq 2$, and B_i contains 2^{i-1} terminals in addition to v_0 and v_1, B_{i+1} contains 2^i terminals, and so on. According to this setting, there are strictly less than $\lfloor \log k \rfloor$ bursts for all k terminals in $\mathcal{B}$. Now for B_i we consider graph G_i, for B_{i+1} we consider graph G_{i+1}, and so on. We start from the construction of a minimal tree sequence $\{\overline{T}_i, \overline{T}_{i+1}, \ldots, \overline{T}_{i+m-1}\}$ by an algorithm $\mathcal{A}$. For all the terminals in the first burst B_i, the length of the tree $\overline{T}_i$ is such that $\mathcal{C}_\mathcal{A}(\overline{T}_i) \geq 1$. This is because all the 2^{i-1} terminals of B_i and two nodes v_0 and v_1 can contain in a path from v_0 to v_1 with length exactly 1 in G_i. As there are exactly double number of terminals in B_{i+1} comparing to the number of terminals in B_i except v_0 and v_1, $\mathcal{C}_\mathcal{A}(\overline{T}_{i+1}) \geq 1 + \frac{1}{2}$. We can achieve this length choosing level $i + 1$ nodes that are not in $\overline{T_i}$ to be in B_{i+1}. We can have a path in G_{i+1} that contains all the terminals in B_i and B_{i+1}, therefore $\{B_i, B_{i+1}\}$ is an initial segment of the sequence of m bursts.

Using inductive argument, assume that $\{B_i, B_{i+1}, \ldots, B_{i+j-1}\}$ is an initial segment for the sequence of m bursts $\mathcal{B}$. Since $\overline{T}_{i+j-1}$ is a minimal tree, each terminal must be in one of B_l, $i \leq l \leq i + j - 1$, and there are no cycles in $\overline{T}_{i+j-1}$. Consider a terminal $v \in B_{i+j-1}$. Node v is adjacent to exactly 4 nodes at level $i + j$ consisting of two sister pairs. If $\overline{T}_{i+j-1}$ contains both nodes of a level $i + j$

Algorithm 1: BSTP approximation algorithm

Input: An undirected (directed) graph $G = (V, E)$ with weight function $\mathfrak{w} : E \to \mathbb{R}^+$ and a set $\mathcal{S} \subseteq V$ of terminals that appear online in a sequence of m bursts $\mathcal{B} = \{B_1, B_2, \ldots, B_m\}$, $B_i \subseteq \mathcal{S}, 1 \leq i \leq m$, one burst at a time.

Output: A Steiner tree (an arborescence) T for $\mathcal{S}$ in G.

When a new burst B_i arrives at step i:
  For each terminal $v \in B_i$ **do**
    Find a node x in T_i that is closest to v and join v to x with the shortest path;
$T := T_i$;

sister pair then $\overline{T}_{i+j-1}$ has a cycle or a leaf node at a level greater than $i + j - 1$ and hence $\overline{T}_{i+j-1}$ can not be the minimum tree. Therefore, $\overline{T}_{i+j-1}$ can only contain at most one of the nodes from each sister pair $(i + j)$-adjacent to v. Therefore, for each node v in burst B_{i+j-1}, we select one node from each sister pair of v that is not in $\overline{T}_{i+j-1}$ to place in the burst B_{i+j}. Note that, according to the construction, there is a path of length 1 which contains all the nodes in every $B_l, i \leq l \leq i + j$. Thus, $\{B_i, B_{i+1}, \ldots, B_{i+j}\}$ is an initial segment of the sequence of m bursts $\mathcal{B}$. Recall that the length of a shortest path from each of the nodes in B_{i+j} to a node in $\overline{T}_{i+j-1}$ is $1/2^{i+j}$. Moreover, we have 2^{i+j-1} level $i + j$ nodes in B_{i+j}. Thus, $\mathcal{C}_{\mathcal{A}}(\overline{T}_{i+j}) \geq \mathcal{C}_{\mathcal{A}}(\overline{T}_{i+j-1}) + \frac{1}{2} = 1 + \frac{j-2}{2} + \frac{1}{2} = 1 + \frac{j-2+1}{2} = 1 + \frac{j-1}{2}$. Therefore, as there are exactly m bursts in $\mathcal{B}$, the lower bound for any algorithm $\mathcal{A}$ is $\mathcal{C}_{\mathcal{A}}(\overline{T}_{i+m-1}) \geq 1 + \frac{m-1}{2} \geq \Omega(m)$. Finally, combining two different lower bounds, we obtain $\Omega(\min\{\log k, m\})$ lower bound. □

Upper bound: We outline a simple algorithm for BSTP which is asymptotically optimal in undirected graphs. For each burst B_i at step i, first find a node $v \in B_i$ that is closest to the current tree of node bursts up to B_{i-1} and connect that node v to the closest node of T_{i-1} with a shortest path. After that, repeat this process for rest of the nodes in B_i until all the nodes in the burst are connected to the current Steiner tree at step T_i at step i. Note that for every node in $v \in B_i$, except the first node, the current tree is the tree that is formed after the connection of nodes in the bursts up to B_{i-1} and the nodes in B_i that are already connected. The details are in Algorithm 1.

Theorem 2 *The competitive ratio of Algorithm 1 for BSTP is $\mathcal{O}(\min\{\log k, m\})$ in undirected graphs.*

Combining Theorems 1 and 2, we obtain the tight bound $\Theta(\min\{\log k, m\})$. As $m = 1$ in STP, we have from Theorem 1 that the lower bound for STP is $\Theta(\min\{\log k, 1\}) = \Theta(1)$. Similarly, as $m = k$ in OSTP, $\Theta(\min\{\log k, k\}) = \Theta(\log k)$ which matches the tight bound given in [9].

3 BSTP in Graphs of Bounded Edge Asymmetry

We first prove a lower bound and then give a deterministic approximation algorithm that matches the lower bound for certain values of m and α.

Lower bound: We prove the lower bound of $\Omega\left(\min\left\{\max\left\{\alpha\frac{\log k}{\log \alpha}, \alpha\frac{\log k}{\log\log k}\right\}, k^{1-\epsilon}, \max\left\{\alpha, \alpha\frac{m}{\log \alpha}\right\}\right\}\right)$, where $0 < \epsilon < 1$ is a small constant. We construct a sequence of directed graphs $G_\ell, \ell \geq 0$, similar to [7]. We use the concept of d-tree defined as follows: A d-tree is a completely binary tree that has pairs of opposite directed edges (a d-tree with root at the bottom is shown inside a dotted triangle in Fig. 2). The d-tree guarantees that an algorithm for BSTP has two identical routing choices for each destination. The sequence of graphs $G_{\ell\geq 0}$ are then recursively generated similar to Section 2. We start with a pair of nodes, and insert a d-tree between the pair of nodes at each step ℓ. The depth of the d-tree that is inserted between the pair of nodes is set to $\gamma = \lfloor\alpha\rfloor$. We obtain G_1 after inserting a d-tree between the pair of nodes in G_0 (Fig. 2). According to this construction, there are $\ell \cdot (\gamma + 1) + 1$ number of layers in G_ℓ; initially, G_0 has only two layers. The number of layers gives the depth of the graphs $G_{\ell\geq 0}$. In G_0, we denote $v_{0,0}$ as the *source* node and $v_{0,1}$ as the *first* node. We set $\mathfrak{w}(v_{0,0}, v_{0,1}) = \rho_0$ and $\mathfrak{w}(v_{0,1}, v_{0,0}) = \alpha \cdot \rho_0$ for the two directed edges $(v_{0,0}, v_{0,1})$ and $(v_{0,1}, v_{0,0})$ in G_0. Moreover, a node of G_ℓ is denoted by $v_{\ell,\chi}$, where χ denotes the layer. Note that, irrespective of the notation $v_{0,0}, v_{1,0}$ etc., the source and first nodes are same in all G_ℓ. Between every pair of nodes in $G_{\ell-1}$, a d-tree is introduced. Therefore, in G_ℓ, $\mathfrak{w}(v_{\ell,\chi}, v_{\ell,\chi+1}) = \rho_\ell$ and $\mathfrak{w}(v_{\ell,\chi+1}, v_{\ell,\chi}) = \alpha \cdot \rho_\ell$ for the edges between adjacent nodes, where $\rho_\ell = \frac{\rho_0}{(\gamma+1)^\ell} = \frac{\rho_{\ell-1}}{\gamma+1}$. This construction guarantees that the asymmetry of $G_{\ell\geq 0}$ is α. Moreover, all paths between any pair of nodes are of equal length in each G_ℓ. The number of nodes (or layers) χ in a path from the source to the first node in any $G_{\ell\geq 0}$ is given by $\chi = (\gamma + 1)^\ell + 1$.

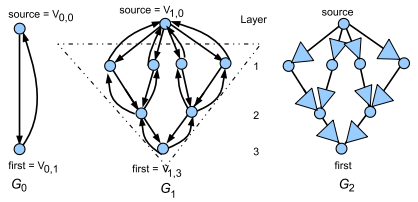
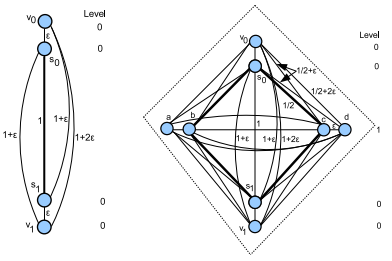


Figure 2: Illustration of sequences of directed graphs $G_i, i \geq 0$, with $\gamma = 2$

Theorem 3 *The competitive ratio of any algorithm for BSTP in any directed graph of edge asymmetry α is $\Omega\left(\min\left\{\max\left\{\alpha\frac{\log k}{\log \alpha}, \alpha\frac{\log k}{\log\log k}\right\}, k^{1-\epsilon}, \max\left\{\alpha, \alpha\frac{m}{\log \alpha}\right\}\right\}\right)$.*



(a) G_0 (b) G_1

Figure 3: Illustration of sequences of graphs G_0 and G_1.

Upper bound: We analyze Algorithm 1 for BSTP in graphs of bounded asymmetry α. Given a directed graph $G = (V, E)$ with asymmetry α, Algorithm 1 finds an arborescence T rooted at some node $\wp$ for $\mathcal{S}$ in G. The only extra information that needs to be provided to Algorithm 1 is the root node $\wp$.

Theorem 4 *The competitive ratio of Algorithm 1 for BSTP in any graph of edge asymmetry α is* $\mathcal{O}\left(\min\left\{\max\left\{\alpha\frac{\log k}{\log \alpha}, \alpha\frac{\log k}{\log\log k}\right\}, k, \alpha\cdot m\right\}\right)$.

Comparing with Theorem 3, this result is tight when $m \geq \log k$ and tight within a $\log\alpha$ factor when $m < \log k$ for $\alpha \in \mathcal{O}(k^{1-\epsilon})$ or $\alpha \in \Omega(k)$.

4 BTSTP in Complete Undirected Graphs

We first prove the lower bound of $\min\{\rho/2, m/4\}$ and give a deterministic algorithm which achieves the upper bound of $\min\{4\rho, 3\lambda m\}$, where $\rho = \frac{1}{2}\log k$.

Lower bound: We create a sequence of instances I_ℓ for the terminals that need to be connected online in a terminal Steiner tree based on a sequence of complete graphs $G_\ell, \ell \geq 0$, similar to the construction given in Section 2 for BSTP in undirected graphs, that we define below. We then apply an adversary argument to show that instances I_ℓ can be created such that the cost of any algorithm $\mathcal{A}$ for BTSTP matches our claimed result. We note here that the lower bound for OSTP due to Imase and Waxman [9] does not apply to BTSTP. The reason is that they considered planar graphs for the lower bound in which TSTP may not have a feasible solution [6, 10, 11]. Moreover, the approach of [9] does not produce a terminal Steiner tree.

We begin by constructing a sequence of undirected complete graphs $G_\ell = (V_\ell, E_\ell), l \in \mathcal{Z}^+$, with a length function on the edges of G_ℓ. As for example, the construction of G_0 and G_1 is given in Fig. 3. The initial graph G_0 is a undirected complete graph with four nodes. Denote these four nodes by v_0, v_1, s_0, and s_1. In G_0, nodes v_0 and v_1 are the terminal vertices and s_0 and s_1 are Steiner vertices. There are six edges between these four nodes which are $(v_0, s_0), (v_1, s_1), (s_0, s_1), (v_1, s_0), (v_0, s_1)$, and (v_0, v_1). Each of these edges has length that satisfies triangle inequality. In particular, we have $\mathfrak{w}(v_0, v_1) = 1+2\epsilon$, $\mathfrak{w}(v_0, s_0) = \mathfrak{w}(v_1, s_1) = \epsilon$, $\mathfrak{w}(v_1, s_0) = \mathfrak{w}(v_0, s_1) = 1+\epsilon$, and $\mathfrak{w}(s_0, s_1) = 1$, where $0 < \epsilon < 1$ is a small positive constant which we set later. The only four nodes in G_0 are called *level 0 nodes*. G_0 as described above is given in Fig. 3a.

We now use a recursive definition to define graph $G_\ell, \ell > 0$, based on graph $G_{\ell-1}$. We obtain G_1 from G_0 as follows. We introduce four new nodes in V_0 to obtain V_1. We take edge (s_0, s_1) (the bold edge) and add two nodes on each side of it; a, b in the left side and c, d in the right side as shown in Fig. 3b. The two nodes in each side of (s_0, s_1) are connected with an edge of length ϵ between them, i.e., a and b are connected with an edge (a, b) such that $\mathfrak{w}(a, b) = \epsilon$. Moreover, c and d are connected and $\mathfrak{w}(c, d) = \epsilon$. The nodes a and d are the terminal nodes and b, c are their Steiner vertexes, respectively. All 8 nodes of G_1 are connected to each other with appropriate weights so that triangle inequality is satisfied. The weights of the edges between nodes are given in Fig. 3b.

The graph G_2 is obtained from G_1 as follows. We introduce four new nodes in each edge $(s_0, b), (b, s_1), (s_1, c)$, and (c, s_0) in a way similar as of constructing G_1 from G_0. The high level structure of G_2 is given in Fig. 4. The edges weights are also assigned proportionally to the level ℓ of G_ℓ. We call the new nodes introduced to get G_ℓ from $G_{\ell-1}$ the *level ℓ nodes*. Therefore, in $G_{\ell\geq 0}$, there are level 0 up to level ℓ nodes. We denote by *level* of a node the level in G_ℓ where that node belongs. Two level ℓ Steiner nodes that are added between any two level $\ell-1$ Steiner nodes in G_ℓ are called *sister nodes*, e.g. b and c of G_1 in Fig. 3. Moreover, i-adjacency of two Steiner nodes $u, v \in G_\ell$ is defined similarly as of Section 3.

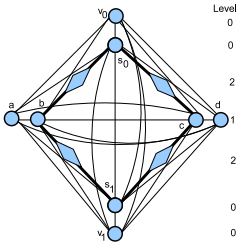


Figure 4: Illustration of graph G_2

Consider a sequence of m terminal sets $\mathcal{N} = \{N_0, N_1, \ldots, N_m\}$ for graphs $G_\ell, \ell \geq 0$, respectively. Each N_ℓ contains some specific number of level ℓ nodes from $G_\ell, 0 \leq \ell \leq m$ (we give exact numbers later). We create this sequence in such a way that there exists a comb structure between s_0 and s_1 with all the nodes in the terminal set $\mathcal{N}$ as its tooth and the length of the comb structure is exactly $1 + k\epsilon$ using an optimal algorithm (which assumes that all the terminals are known in advance). To show that the performance of any BTSTP algorithm $\mathcal{A}$ is at least our claimed bound, we choose the number of nodes in each set N_ℓ similar to bursts B_i of Section 2.

While connecting the terminals in a tree, they will not be connected by a direct edge between them (although

such edge exists) to satisfy the terminal Steiner tree criteria. In our lower bound construction, we connect terminals through Steiner nodes near to them. For example, if terminals v_1 and d need to be connected in G_1, then we use the path (v_1, s_1, c, d); the alternative path choices (v_1, s_1, d), (v_1, c, d), and (v_1, d) are not used. Our choice matches the length of other path choices and at the same time avoids the need of possible rearrangement due to future arrivals of terminals. Moreover, our path choice does not affect the lower bound.

We define a minimal terminal Steiner tree sequence $\overline{\mathcal{T}} = \{\overline{T}_0, \overline{T}_1, \ldots, \overline{T}_m\}$ for graphs $G_\ell, \ell \geq 0$, with respect to a sequence of terminal sets $\{N_0, N_1, \ldots, N_m\}$. Therefore, $\overline{T}_0$ is any tree that connects all the terminals in N_0 and there does not exist a proper subgraph of $\overline{T}_0$ that also connects all the terminals in N_0 satisfying the terminal Steiner tree criteria. For each $\overline{T}_\ell, 1 \leq \ell \leq m$, $\overline{T}_\ell$ must contain tree $\overline{T}_{\ell-1}$ as a subgraph and connect all of the terminals in N_ℓ. As we consider minimal terminal Steiner tree sequence, no subgraph of $\overline{T}_\ell$ can satisfy this requirement. The minimal terminal Steiner tree sequence $\overline{\mathcal{T}}$ is constructed by an algorithm $\mathcal{A}$ based on the knowledge of only $\overline{T}_{\ell-1}$ and N_ℓ in each step ℓ to obtain $\overline{T}_\ell$. We choose the nodes to include in $\overline{T}_\ell$ based on the knowledge of $\overline{T}_{\ell-1}$ to maximize the length using $\mathcal{A}$.

Theorem 5 *The competitive ratio of every deterministic algorithm for BTSTP is at least* $\min\{\frac{\rho}{2}, \frac{m}{4}\}$ *in undirected complete graphs, where* $\rho = \frac{1}{2}\log k$.

Upper bound: We now outline a simple approximation algorithm for BTSTP which is asymptotically optimal within a constant factor. Before providing the details, we outline some assumptions. Let $D(v_i)$ be the set of neighbors of $v_i \in \mathcal{S}_{i-1}$ (i.e., $D(v_i) = \{r | (v_i, r) \in T_i, r \notin \mathcal{S}\}$ and its members are all Steiner vertices) and $D_1(v_i)$ be the nearest neighbor of v_i in T_i (i.e., $\mathfrak{w}(v_i, D_1(v_i)) = \min\{\mathfrak{w}(v_i, r) | r \in D(v_i)\}$). We assume without generality that $D_1(v_i)$ will not be the terminal of $\mathcal{S}$ in any step $l > i$.

As there are nodes arriving in a burst in BTSTP (i.e., a subset of $|B_i|$ nodes arriving at step i), our BTSTP algorithm deals with how to connect the burst at each step i to the existing tree formed due to the bursts from B_1 up to B_{i-1}. For each burst B_i at step i, our algorithm first removes all the edges from G that connect terminals in B_i to the previous arrived terminals in $\mathcal{B}_{i-1}$ and also the terminals in B_i. Then, it finds a node $v \in B_i$ that is closest to the current tree of node bursts up to B_{i-1} and connects that node v to the closest node of T_{i-1} with a shortest path. After that, it repeats this process for rest of the nodes in B_i until we connect all the nodes in the burst B_i to the current Steiner tree T_i at step i. Note that for every node in $v \in B_i$, except the first node, the current tree T_i is the tree that is formed after the connection of nodes in the bursts up to B_{i-1} and the nodes in B_i that are already connected to T_i. After all the nodes in B_i are connected, if any node $v_j \in \mathcal{B}_i$ is not a leaf in T_i, our algorithm performs star-replacement operations for v_j. We pick $D_1(v_j)$ and connect v_j and $D(v_j) \backslash D_1(v_j)$ to $D_1(v_j)$ removing the edge $(v_j, s), s \in D(v_j) \backslash D_1(v_j)$. Here $\mathcal{B}_i$ denotes the terminals in $\mathcal{S}$ that arrived up to step i.

Theorem 6 *The competitive ratio of the aforementioned algorithm for BTSTP is* $\min\{4\rho, 3\lambda m\}$*, where* λ *is the current best approximation of TSTP.*

References

[1] S. Angelopoulos. Improved bounds for the online steiner tree problem in graphs of bounded edge-asymmetry. In *SODA*, pages 248–257, 2007.

[2] S. Angelopoulos. A near-tight bound for the online steiner tree problem in graphs of bounded asymmetry. In *ESA*, pages 76–87, 2008.

[3] M. Bern and P. Plassmann. The steiner problem with edge lengths 1 and 2,. *Inf. Process. Lett.*, 32(4):171–176, 1989.

[4] J. Byrka, F. Grandoni, T. Rothvoss, and L. Sanita. Steiner tree approximation via iterative randomized rounding. *J. ACM*, 60(1):6:1–6:33, 2013.

[5] Y. Chen. An improved approximation algorithm for the terminal steiner tree problem. In *ICCSA*, pages 141–151, 2011.

[6] D. E. Drake and S. Hougardy. On approximation algorithms for the terminal steiner tree problem. *Inf. Process. Lett.*, 89(1):15 – 18, 2004.

[7] M. Faloutsos, R. Pankaj, and K. C. Sevcik. The effect of asymmetry on the on-line multicast routing problem. *Int. J. Found. Comput. Sci.*, 13(6):889–910, 2002.

[8] M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., New York, 1990.

[9] M. Imase and B. M. Waxman. Dynamic steiner tree problem. *SIAM J. Discrete Math.*, 4(3):369–384, 1991.

[10] G.-H. Lin and G. Xue. On the terminal steiner tree problem. *Inf. Process. Lett.*, 84(2):103–107, 2002.

[11] C. L. Lu, C. Y. Tang, and R. C.-T. Lee. The full steiner tree problem. *Theoretical Computer Science*, 306(1–3):55 – 67, 2003.

[12] S. Ramanathan. Multicast tree generation in networks with asymmetric links. *IEEE/ACM Trans. Netw.*, 4(4):558–568, 1996.

[13] H. Takahashi and A. Matsuyama. An Approximate Solution for the Steiner Problem in Graphs. *Math. Japonica*, 24:573–577, 1980.