An Efficient Approximation Algorithm for the Steiner Tree Problem

Chi-Yeh Chen
Department of Computer Science and
Information Engineering,
National Cheng Kung University
Taiwan.
chency@csie.ncku.edu.tw

ABSTRACT
The Steiner tree problem is one of the classic and most fundamental NP-hard problems: given an arbitrary weighted graph, seek a minimum-cost tree spanning a given subset of the vertices (terminals). Byrka et al. proposed a 1.3863+ \( \varepsilon \) -approximation algorithm in which the linear program is solved at every iteration after contracting a component. Goemans et al. shown that it is possible to achieve the same approximation guarantee while only solving hypergraphic LP relaxation once. However, optimizing hypergraphic LP relaxation exactly is strongly NP-hard. This article presents an efficient two-phase heuristic in greedy strategy that achieves an approximation ratio of 1.4299.

CCS Concepts
• Mathematics of computing → Trees • Mathematics of computing → Graph algorithms • Mathematics of computing → Approximation algorithms.

Keywords
Steiner trees; approximation algorithms; graph Steiner problem; network design.

1. INTRODUCTION
The Steiner tree problem is one of the classic and most fundamental NP-hard problems. Given an arbitrary weighted graph with a distinguished vertex subset, the Steiner tree problem asks for a shortest tree spanning the distinguished vertices. This problem is widely used in many fields, such as VLSI routing [12], wireless communications [15, 16], transportation [11], wirelength estimation [5], and network routing [14]. The Steiner tree problem is NP-hard even in the very special cases of Euclidean or rectilinear metrics [8]. In fact, it is NP-hard to approximate the Steiner tree problem within a factor 96/95 [6]. Hence, an approximation algorithm with a small and provable guarantee is strongly NP-hard. This article presents an efficient two-phase algorithm with cost at most \( \alpha \) times the optimum value.

Arora [1] established that Euclidean and rectilinear minimum-cost Steiner trees can be approximated in polynomial time within a factor of \( 1 + \varepsilon \) for any constant \( \varepsilon > 0 \). For arbitrary weighted graphs, a sequence of improved approximation algorithms appeared in the literatures [2, 4, 10, 13, 17, 18, 19, 20, 21] and the best approximation ratio achievable within polynomial time was improved from 2 to 1.39.

Byrka et al. proposed an LP-based approximation algorithm that achieves approximation ratio of \( \ln 4 + \varepsilon \) for general graphs [4]. However, the linear program is solved at every iteration after contracting a component. Goemans et al. [9] shown that it is possible to achieve the same approximation guarantee while only solving hypergraphic LP relaxation once. However, optimizing hypergraphic LP relaxation exactly is strongly NP-hard [9]. Borchers and Du [3] show that \( \rho_k \leq 1 + [\log_k{k}]^{-1} \) where \( \rho_k \) is the worst-case ratio of the cost of optimal \( k \)-restricted Steiner tree to the cost of optimal Steiner tree. We may therefore choose \( k = 2^{1/\varepsilon} \) appropriately to obtain a \( 1+\varepsilon \) approximation to hypergraphic LP relaxation, for any \( \varepsilon > 0 \). The number of variables and constraints will consequently be more than \( n^{21/\varepsilon} \) where \( n \) is the number of terminals [7].

2. NOTATION AND PRELIMINARIES
Given a graph \( G = (V, E) \) with nonnegative edge costs (or weights) cost: \( E \rightarrow R^+ \) and a subset \( R \subseteq V \) of terminals of the vertices of \( G \), the Steiner tree problem asks for a minimum-cost Steiner tree spanning \( R \). Any tree in \( G \) spanning \( R \) is called a Steiner tree, and any non-terminal vertices contained in a Steiner tree are referred to as Steiner points. The cost of a tree is the sum of its edge costs. The graph \( G \) is assumed to be a complete graph and let \( G_R \) be a complete graph that induced by \( R \).

For any graph \( H \), we denote by \( MST(H) \) a minimum spanning tree of a graph \( H \) and by \( cost(H) \) the sum of the costs of all edges in \( H \). We thus abbreviate \( \text{mst}(H) = cost(MST(H)) \), i.e., the cost of a minimum spanning tree of \( H \).

A terminal-spanning tree is a Steiner tree that does not contain any Steiner points. Let \( mst \) be the cost of minimum terminal-spanning tree \( MST(G_R) \). A minimum-cost Steiner tree spanning subset \( R' \subseteq R \) in which all terminals are leaves is called a full component. Any Steiner tree can be decomposed into full components by splitting all the non-leaf terminals [18]. Our algorithm will start with a minimum-cost terminal spanning tree, and iteratively adds full components to improve it. Any full component is assumed to have its own copy of each Steiner point so that full components chosen by our algorithm do not share Steiner points.
Let $\Gamma(K)$ be the terminal set of a given full component $K$. Let $E_0(R')$ be the set of zero-cost edges in which all edges connect all pairs of terminals in $R'$. For brevity, let $E_0(H) = E_0(\Gamma(H))$. We call a Steiner tree $S$ is a well solution if $|\Gamma(K) \cap \Gamma(K')| \leq 1$ for any two full components $K_i$ and $K_j$ in $S$. Let Loss($K$) be the minimum-cost sub-forest of $K$. A simple method of computing Loss($K$) is given by the following lemma.

**Lemma 2.1** [18]. For any full component $K$, Loss($K$) = MST($K \cup E_0(K)$) − $E_0(K)$.

We denote the cost of Loss($K$) by loss($K$). Let $C[K]$ be a loss-contracted full component that can be obtained by collapsing each connected component of Loss($K$) into a single node. We denote by Opt$_k$ an optimal k-restricted Steiner tree. Let opt$_k$ and loss$_k$ be the cost and loss of Opt$_k$, respectively. Let opt be the cost of the optimal Steiner tree. For brevity, this article uses $T/E_0(R')$ to denote the minimum spanning tree of $T \cup E_0(R')$ for $R' \subseteq R$.

The gain of a full component $K$ with respect to $T$ is defined as $$\text{gain}_T(K) = \text{cost}(T) - \text{mst}(T \cup E_0(K)) - \text{cost}(T),$$

and the load of a full component $K$ with respect to $T$ is defined as $$\text{load}_T(K) = \text{cost}(K) + \text{mst}(T \cup E_0(K)) - \text{cost}(T).$$

Let $$\Psi_{T_1, T_2}(K) = \text{cost}(T_1) - \text{cost}(T_2) - \text{mst}(T_1 \cup E_0(K)) + \text{mst}(T_2 \cup E_0(K)).$$

The following lemma shows that if no full component can improve a terminal-spanning tree $T$, then cost($T$) ≤ opt$_k$.

**Lemma 2.2** [18]. Let $T$ be a terminal-spanning tree; if gain$_T(K)$ ≤ 0 for any $k$-restricted full component $K$, then cost($T$) ≤ opt$_k$.

### 3. TWO-PHASE ALGORITHM

This section proposes a $k$-restricted two-phase heuristic ($k$-TPH) which is described in **Algorithm 1**. Let $T^t$ be the terminal-spanning tree at the end of iteration $t$ and let $T^t$ be the chosen full component at the end of iteration $t$. The first phase finds a terminal-spanning tree $T_b$ such that no full component can improve it. The terminal-spanning tree $T_b$ is a based criterion for the second phase. We denote by $S_1$ the solution in the first phase, and by $S_2$ the solution in the second phase. The first phase is a loss-contracting algorithm. The criterion function of $K$ with respect to $T^{t-1}$ is defined as $$r = \frac{\text{gain}_{T^{t-1}}(K)}{\text{loss}(K)}.$$ A chosen full component $K_t$ may be modified by other chosen full components. Assume that some edges $\{e_1, e_2, \ldots\}$ in $T^{t-1}$ are deleted when adding $C[K_t]$ to $T^{t-1}$. Some components are obtained by $K_t - \{e_1, e_2, \ldots\}$ and each component can be replaced by a full component with same terminals. The full component $K_t$ is replaced by these full components. That is because we want to ensure that $1/2 \cdot \text{cost}(S) \leq \text{cost}(T_b)$. If no edge in $T^{t-1}$ is corresponding to $C[K_t]$, we keep a basic component from $K_t$ that is a Steiner point directly connect to two terminals in which an edge belongs to Loss($K_t$) and another edge belongs to $K_t - \text{loss}(K_t)$ (see Figure 1). It guarantees that the chosen full components never be chosen again. However, it may bring that some Steiner points are leaves in $S_1$. Fortunately, these Steiner points can be removed. Therefore, this paper assume that no Steiner point is leaf in $S_1$.

**Algorithm 1** The $k$-restricted two-phase heuristic ($k$-TPH)

---

**The first phase**

1. Initialize $T^0 = \text{MST}(G_2)$.
2. for $t = 1, 2, \ldots$ do
3.    Find a $k$-restricted full component $K_t = K$ with maximizes $r = \frac{\text{gain}_{T^{t-1}}(K)}{\text{loss}(K)}$.
4.      if $r \leq 0$ then
5.          return $T_b$.
6.      end if
7.      if there exist some edges $\{e_1, e_2, \ldots\} \subseteq T^{t-1} - \text{MST}(T^{t-1} \cup E_0(K_t))$ and $\{e_1, e_2, \ldots\} \subseteq C[K_t]$ for $i \neq t$ then
8.          Some components are obtained by $K_t - \{e_1, e_2, \ldots\}$ and each components can be replaced by a full component with same terminals.
9.      end if
10.     Replace the full component $K_t$ by these full components.
11.     (for convenient to describe algorithm, we reuse the notation $K_t$ to represent these full components.)
12. end if
13. $T^t = \text{MST}(T^0 \cup C[K_1] \cup \cdots \cup C[K_t])$.
14. $S_1 = \text{MST}(T^0 \cup K_1 \cup \cdots \cup K_t)$.
15. if no edge in $T^t$ is corresponding to $C[K_t]$ for $i \neq t$ then
16.     Keep a basic component from $K_t$.
17. end if
19. end for

---

**The second phase**

21. $S_2 = k$-ERGH($T_b$).
22. return the minimum-cost tree $S$ between $S_1$ and $S_2$.

The second phase calls the $k$-restricted enhanced relative greedy heuristic ($k$-ERGH), which is described in **Algorithm 2**, to obtain a Steiner tree $S_2$. The $k$-ERGH iteratively finds a full component $K$ for modifying the terminal-spanning trees $T_b^0 = \text{MST}(G_2)$ and $T_b^0$. When a full component $K_t$ has been chosen, the algorithm contracts the cost of the corresponding edges in $T_b^{t-1}$ to zero, that is, $T_b^0 = \text{MST}(T_b^{t-1} \cup E_0(K_t))$. Similarly, $T_b^t = \text{MST}(T_b^{t-1} \cup$
The criterion function of $K$ with respect to $T^{T-1}_0$ and $T^{b-1}_b$ is defined as

$$f(K) = \frac{\text{load}_{T^{T-1}_0}(K)}{\psi_{T^{b-1}_b}(K)}$$

The following steps analyze the complexity of $k$-TPH. Recall that, $n$ is the number of terminals. In the first phase, the number of iterations cannot exceed the number of full Steiner components $O(n^2)$. The gain of a full component $K$ can be found in time $O(k)$ after precomputing the longest edges between any pair of nodes in the current minimum spanning tree, which may be accomplished in time $O(n \log n)$ [18]. Thus, the runtime of all the iterations in the first phase is $O(k n^{2k+1} \log n)$.

We also can obtain the runtime of all the iterations in the second phase is bounded by $O(k n^{2k+1} \log n)$. Thus, the total runtime is $O(k n^{2k+1} \log n)$.

**Algorithm 2** The $k$-restricted enhanced greedy heuristic ($k$-ERGH)

**Require:** $T_b$

1. $T^0_b = T_b$ and $T^0_0 = \text{MST}(G_2)$
2. for $t = 1, 2, \ldots$ do
3. Find a $k$-restricted full component $K_t = K$ which minimizes
   $$f(K) = \frac{\text{load}_{T^{T-1}_0}(K)}{\psi_{T^{b-1}_b}(K)}$$
4. $T^t_b = \text{MST}(T^{t-1}_b \cup E_0(K_t))$
5. $T^t_0 = \text{MST}(T^{t-1}_0 \cup E_0(K_t))$
6. if $c(T^t_b) = c(T^t_0)$ then
   7. return $\text{MST}(T^0_0 \cup K_t \cup K_{t-1} \cup \ldots \cup K_1)$
8. end if
9. end for

### 4. APPROXIMATION RATIO OF THE $k$-TPH

This section shows the approximation result of the $k$-TPH. When a full component $K$ has been chosen, the following lemma shows that the first phase never chooses the full component $K$ even it has been replaced by some full components.

**Lemma 4.1** The first phase never choose the chosen full components again.

**Proof:** Assume that the first phase choose a full component $K_t = K$. If $\text{MST}(T \cup_{i=1}^t C[K_i])$ contain all edges in $C[K]$ in the iteration $t > t$, $\text{gain}_t(K) \geq 0$ and the first phase never choose the full component $K$ again. If $\text{MST}(T \cup_{i=1}^t C[K_i])$ does not contain some edge $e \in C[K]$ in the iteration $t > t$, the edge $e$ has been improved by a chosen full component. The full component $K$ is divided into two components by removing the edge $e$. Let $A$ and $B$ be two connected components of $K - \{e\}$. The full component $K_t$ is replaced by two full components $K_A$ and $K_B$ with terminals sets $\Gamma(A)$ and $\Gamma(B)$, respectively. We have $T' = \text{MST}(T \cup_{i=1}^t C[K_i] \cup K_A \cup K_B \cup_{i=1}^t C[K_i'] \cup C[K'])$, $\text{gain}_t(K) \leq \text{gain}_t(A \cup B) \leq \text{gain}_t(A) + \text{gain}_t(B)$ and $\text{loss}(K) = \text{loss}(A) + \text{loss}(B)$. Finally,

$$\frac{\text{gain}_t(K)}{\text{loss}(K)} \leq \frac{\text{gain}_t(A) + \text{gain}_t(B)}{\text{loss}(A) + \text{loss}(B)} \leq \max \left\{ \frac{\text{gain}_t(A)}{\text{loss}(A)}, \frac{\text{gain}_t(B)}{\text{loss}(B)} \right\}$$

We show that $K_A$ never be replaced by $A$. We knows that $\text{cost}(K_A) \leq \text{cost}(A)$ and $\text{gain}_t(A) \leq 0$. The full component $K_A$ is superior to $A$. We also can obtain that $K_B$ never be replaced by $B$. The first phase never chooses the full component $K$ again.

**Lemma 4.2** $\text{cost}(T^0_0) \geq \frac{1}{2} \cdot \text{cost}(S_1)$.

**Proof:** The cost of the Steiner tree in the first phase is $\text{cost}(S_1) = \text{cost}(T^0_0) + \sum_{k=0}^{n} \text{cost}(K_k)$.

Since $\text{cost}(K) \geq \frac{1}{2} \cdot \text{cost}(K)$ [18] for any full component $K$,

$$\text{cost}(S_1) \leq \text{cost}(T^0_0) + \sum_{k=0}^{n} \frac{1}{2} \cdot \text{cost}(K_k) \leq \text{cost}(T^0_0) + \frac{1}{2} \cdot \text{cost}(S_1)$$

which yields $\text{cost}(T^0_0) \geq \frac{1}{2} \cdot \text{cost}(S_1)$.

**Lemma 4.3** If no full component can improve the terminal-spanning tree $T$.

$$\text{load}_T \left( \bigcup_{i=1}^{n} K_i \right) \geq \sum_{i=1}^{n} \text{load}_T(K_i)$$

for full components $K_1, K_2, \ldots, K_n$.

**Proof:** The proof can be obtained by the following chain of inequalities:

$$\text{load}_T \left( \bigcup_{i=1}^{n} K_i \right) = \text{cost} \left( \bigcup_{i=1}^{n} K_i \right) + \text{mst} \left( T \cup_{i=1}^{n} E_0(K_i) \right) - \text{cost}(T)$$

$$\geq \sum_{i=1}^{n} \text{cost}(K_i) + \text{mst} \left( T \cup_{i=1}^{i} E_0(K_i) \right) - \text{cost}(T) \geq \sum_{i=1}^{n} \text{cost}(K_i) + \text{mst} \left( T \cup_{i=1}^{n} E_0(K_i) \right) - \text{cost}(T) \geq \sum_{i=1}^{n} \text{load}_T(K_i)$$

The following lemma guarantees that the solution of $k$-TPH at the second phase is a well solution.

**Lemma 4.4** For any chosen full components $K_i$ and $K_j$, $|\Gamma(K_i) \cap \Gamma(K_j)| \leq 1$.  

181
Proof. Assume that \(|\Gamma(K_i) \cap \Gamma(K_j)| = 2\) and \(j < i\). Both \(T_{0}^{i-1} - \text{MST} (T_{0}^{i-1} \cup E_0(K_j))\) and \(T_{0}^{i-1} - \text{MST} (T_{0}^{i-1} \cup E_0(K_j))\) contain a zero-cost edge that is from \(E_0(K_j)\). Since any full component cannot improve \(T_{0}^{i-1}, \text{MST}(T_{0}^{i-1} \cup K) = T_{0}^{i-1} \cup \text{Loss}(K)\) for any full component \(K\). We can find an edge \(e \in K \setminus \text{Loss}(K_i)\) such that

\[
\psi_{T_{0}^{i-1},T_{0}^{i-1}}(K) = \psi_{T_{0}^{i-1},T_{0}^{i-1}}(A) + \psi_{T_{0}^{i-1},T_{0}^{i-1}}(B)
\]

and

\[
\text{load}_{T_{0}^{i-1}}(K) \geq \text{load}_{T_{0}^{i-1}}(A \cup B) \geq \text{load}_{T_{0}^{i-1}}(A) + \text{load}_{T_{0}^{i-1}}(B)
\]

(from Lemma 4.3) where \(A\) and \(B\) are two connected components of \(K_i \setminus \{e\}\). Finally,

\[
\frac{\text{load}_{T_{0}^{i-1}}(K)}{\psi_{T_{0}^{i-1},T_{0}^{i-1}}(K)} \geq \frac{\text{load}_{T_{0}^{i-1}}(A) + \text{load}_{T_{0}^{i-1}}(B)}{\psi_{T_{0}^{i-1},T_{0}^{i-1}}(A) + \psi_{T_{0}^{i-1},T_{0}^{i-1}}(B)} \geq \min \left\{ \frac{\text{load}_{T_{0}^{i-1}}(A)}{\psi_{T_{0}^{i-1},T_{0}^{i-1}}(A)}, \frac{\text{load}_{T_{0}^{i-1}}(B)}{\psi_{T_{0}^{i-1},T_{0}^{i-1}}(B)} \right\}
\]

which contradicts the choice of \(K_i\).

**Lemma 4.5** For any Steiner tree \(S\), \(\text{load}_{T_{0}^{i-1}}(S) \geq \text{load}_{T_{0}^{i-1}}(S/\cup_{j=1}^{i-1} E_0(K_j))\)

Proof. Since no full component can improve the terminal-spanning tree \(T_{0}^{i-1}\),

\[
\text{cost}(S) - \text{cost}(T_{0}^{i-1}) - \text{mst} \left( \bigcup_{j=1}^{i-1} E_0(K_j) \right) \\
+ \text{mst} \left( T_{0}^{i-1} \cup \bigcup_{j=1}^{i-1} E_0(K_j) \right) \geq 0
\]

The proof can be obtained by the following chain of inequalities:

\[
\text{load}_{T_{0}^{i-1}}(S) = \text{cost}(S) - \text{cost}(T_{0}^{i-1}) \\
= \text{mst} \left( \bigcup_{j=1}^{i-1} E_0(K_j) \right) \\
- \text{mst} \left( T_{0}^{i-1} \cup \bigcup_{j=1}^{i-1} E_0(K_j) \right) + \text{cost}(S) \\
- \text{cost}(T_{0}^{i-1}) - \text{mst} \left( \bigcup_{j=1}^{i-1} E_0(K_j) \right) \\
+ \text{mst} \left( T_{0}^{i-1} \cup \bigcup_{j=1}^{i-1} E_0(K_j) \right) \\
\geq \text{mst} \left( \bigcup_{j=1}^{i-1} E_0(K_j) \right) \\
- \text{mst} \left( T_{0}^{i-1} \cup \bigcup_{j=1}^{i-1} E_0(K_j) \right) \\
= \text{load}_{T_{0}^{i-1}}(S/\bigcup_{j=1}^{i-1} E_0(K_j))
\]

**Lemma 4.6** If \(\psi_{T_{0}^{i-1},T_{0}^{i-1}}(C) \leq \psi_{T_{0}^{i-1},T_{0}^{i-1}}(C)\) for any full components \(C \) and \(K\),

\[
\frac{\text{load}_{T_{0}^{i-1}}(K)}{\psi_{T_{0}^{i-1},T_{0}^{i-1}}(K)} \leq \frac{\text{load}_{T_{0}^{i-1}}(K)}{\psi_{T_{0}^{i-1},T_{0}^{i-1}}(K)}
\]

Proof. Since \(\text{load}_{T_{0}^{i-1}}(K) \leq \psi_{T_{0}^{i-1},T_{0}^{i-1}}(C)\) and \(\text{cost}(T_{0}^{i-1}) - \text{mst}(T_{0}^{i-1} \cup E_0(C) \cup E_0(K)) \leq \text{cost}(T_{0}^{i-1}) - \text{mst}(T_{0}^{i-1} \cup E_0(K))\), the proof can be obtained by the following inequality:

\[
\frac{\text{load}_{T_{0}^{i-1}}(K)}{\psi_{T_{0}^{i-1},T_{0}^{i-1}}(K)} \leq \frac{\text{load}_{T_{0}^{i-1}}(K)}{\psi_{T_{0}^{i-1},T_{0}^{i-1}}(K)}
\]

Based on the analysis in [21], the bound on the cost of our solution is as follows.

**Theorem 7** The \(k\)-TPH finds a Steiner tree \(S\) such that

\[
\text{cost}(S) \leq \left( \text{opt}_{k} - \text{cost}(T_{0}^{i-1}) + 1 \right) \cdot \left( \text{opt}_{k} - \text{cost}(T_{0}^{i-1}) \right)
\]

Proof. Let \(M_i = \text{cost}(T_{0}^{i}) - \text{cost}(T_{0}^{i-1}) \) and \(m_i = M_{i-1} - M_i\).

Therefore, \(f(K_i) = \frac{\text{load}_{T_{0}^{i-1}}(K_i)}{m_i}\).

\[
\text{Opt}_{k} = \left( \text{opt}_{k} \bigcup_{i=1}^{r} E_0(K_i) \right) - \bigcup_{i=1}^{r} E_0(K_i)
\]

For \(i = 1, \ldots, r + 1 \) and \(\text{load}_{T_{0}^{i-1}}(\text{Opt_{k}}) \leq M_{i-1}, \) we have

\[
\text{load}_{T_{0}^{i-1}}(\text{Opt_{k}}) \leq \frac{\text{load}_{T_{0}^{i-1}}(\text{Opt_{k}})}{M_{i-1}}
\]

Replacing \(M_i = M_{i-1} - M_i\) into the above inequality yields

\[
M_i \leq M_{i-1} \left( 1 - \frac{\text{load}_{T_{0}^{i-1}}(K_i)}{\text{load}_{T_{0}^{i-1}}(\text{Opt_{k}})} \right)
\]

(1)

for \(i = 1, 2, \ldots, t\). From the inequality (1),

\[
M_t \leq M_0 \prod_{i=1}^{t} \left( 1 - \frac{\text{load}_{T_{0}^{i-1}}(K_i)}{\text{load}_{T_{0}^{i-1}}(\text{Opt_{k}})} \right).
\]
Taking the natural logarithms of both sides and using the inequality \(\ln(1 + x) \leq x\),
\[
\ln \frac{M_0}{M_r} \geq -\sum_{i=1}^{t} \ln \left(1 - \frac{\text{load}_{\phi}^{i-1}(K_i)}{\text{load}_{\phi}(\text{opt}_k)}\right) \geq \frac{\sum_{i=1}^{t} \text{load}_{\phi}^{i-1}(K_i)}{\text{load}_{\phi}(\text{opt}_k)}
\]  
(2)

Since \(k\)-TPA interrupts at \(M_t = c(T^0_k) = c(T^0_k) = 0\), there exists \(M_r > \text{load}_{\phi}(\text{opt}_k) \geq M_{r+1}\) for some \(r < t\).

The value \(m_{r+1}\) can be split into two values \(m^*\) and \(m'\) such that
\[
m^* = \text{load}_{\phi}^{r}(\text{opt}_k) - M_{r+1}
\]
(3)
\[
m' = \text{load}_{\phi}^{r}(\text{opt}_k) - M_{r+1}
\]
(4)
According to inequality (3), we have
\[
M_{r+1} = M_r - m' = \text{load}_{\phi}^{r}(\text{opt}_k)
\]
(5)
The value \(\text{load}_{\phi}^{r}(K_{r+1})\) also can be split into \(w^*\) and \(w'\) such that
\[
\text{load}_{\phi}^{r}(K_{r+1}) = \frac{m_{r+1}}{w^*} = \frac{m'}{w'}.
\]
Since \(\text{load}_{\phi}^{r}(K_{r+1}) = \frac{w^*}{w'}\), inequality (2) implies that
\[
\ln \frac{M_0}{M_{r+1}} \geq \frac{\sum_{i=1}^{r} \text{load}_{\phi}^{i-1}(K_i) + w^*}{\text{load}_{\phi}(\text{opt}_k)}
\]  
(6)
Since \(\text{load}_{\phi}^{r}(K_{r+1}) = \frac{m_{r+1}}{w^*} \leq \frac{m_r}{w'} \leq 1\), we have
\[
w' \leq m'
\]  
(7)
The ratio related to the cost of approximate Steiner tree after \(r+1\) iterations is at most
\[
\frac{\text{cost}(S_2) - \text{cost}(T^0_k)}{\text{opt}_k - \text{cost}(T^0_k)} \leq \frac{\text{cost}(S_2) - \text{cost}(T^0_k)}{\text{opt}_k - \text{cost}(T^0_k)} \leq 1 - \frac{(\text{cost}(T^0_k) - \text{opt}_k)}{\text{opt}_k - \text{cost}(T^0_k)} + \frac{\text{cost}(T^0_k)}{\text{opt}_k - \text{cost}(T^0_k)}
\]
\[
\frac{\text{cost}(S_2) - \text{cost}(T^0_k)}{\text{opt}_k - \text{cost}(T^0_k)} \leq \left(\frac{\text{cost}(T^0_k) - \text{opt}_k}{\text{opt}_k - \text{cost}(T^0_k)} + 1\right) \left(\frac{\text{opt}_k - \text{cost}(T^0_k)}{\text{opt}_k - \text{cost}(T^0_k)} + 1\right)
\]
which yields
\[
\text{cost}(S) \leq \text{cost}(S_2)
\]
\[
\text{cost}(S) \leq \left(\frac{\text{cost}(T^0_k) - \text{opt}_k}{\text{opt}_k - \text{cost}(T^0_k)} + 1\right) \left(\frac{\text{opt}_k - \text{cost}(T^0_k)}{\text{opt}_k - \text{cost}(T^0_k)} + 1\right)
\]
Since \(\text{cost}(T^0_k) \leq \text{opt}_k\) (from Lemma 2.2) and \(\text{cost}(T^0_k) \geq \frac{1}{2}\cdot \text{opt}_k\) (from Lemma 4.2), we can assume that \(\text{cost}(T^0_k) = \text{opt}_k\) for \(\alpha \in (0, 1)\). The following result can be obtained.

**Theorem 8** If \(\text{cost}(T^0_k) = \text{opt}_k\) for \(\alpha \in (0, 1)\), the \(k\)-TPH finds a Steiner tree \(S\) such that
\[
\text{cost}(S) \leq \left(\frac{\text{cost}(T^0_k) - \text{opt}_k}{\text{opt}_k - \text{cost}(T^0_k)} + 1\right) \left(\frac{\text{opt}_k - \text{cost}(T^0_k)}{\text{opt}_k - \text{cost}(T^0_k)} + 1\right)
\]
and
\[
\text{cost}(S) \leq 2 \cdot \alpha \cdot \text{opt}_k.
\]

**Proof.** From Theorem 4.7, we have
\[
\text{cost}(S) \leq \left(\frac{\text{cost}(T^0_k) - \text{opt}_k}{\text{opt}_k - \text{cost}(T^0_k)} + 1\right) \left(\frac{\text{opt}_k - \text{cost}(T^0_k)}{\text{opt}_k - \text{cost}(T^0_k)} + 1\right)
\]
According to Lemma 4.2, \(\text{cost}(S) \leq 2 \cdot \text{cost}(T^0_k) = 2 \cdot \alpha \cdot \text{opt}_k\).

**5. PERFORMANCE OF THE K-TPH IN GENERAL GRAPHS**

The following corollary gives a bound on the cost of the Steiner tree generated by \(k\)-TPH.

**Corollary 1** The \(k\)-TPH has an approximation ratio of at most 1.4295.

**Proof.** We have \(\text{cost}(S) \leq 2 \cdot \text{opt}_k\) (see [19]). Theorem 4.8 yield
\[
\frac{\text{cost}(S)}{\text{opt}_k} \leq \left(\frac{2}{\text{opt}_k} - \frac{\text{opt}_k}{\text{opt}_k} + 1\right) \left(1 - \alpha\right)\frac{\text{opt}_k}{\text{opt}_k} + \alpha \cdot \frac{\text{opt}_k}{\text{opt}_k}
\]
\[
\frac{\text{cost}(S)}{\text{opt}_k} \leq \left(\frac{2}{\text{opt}_k} - \frac{\text{opt}_k}{\text{opt}_k} + 1\right) \left(1 - \alpha\right)\frac{\text{opt}_k}{\text{opt}_k} + \alpha \cdot \frac{\text{opt}_k}{\text{opt}_k}
\]
and
\[
\frac{\text{cost}(S)}{\text{opt}_k} \leq 2 \cdot \alpha \cdot \frac{\text{opt}_k}{\text{opt}_k}
\]
where \(\rho_k\) is the worst-case ratio of \(\text{opt}_k\) to \(\text{opt}_t\). Borchers and Du [3] show that \(\rho_k \leq 1 + \log_k 2\). When \(k \to \infty\), the approximation ratio of the \(k\)-TPH converges to
\[
A(\alpha) = \left(\frac{2}{1 - \alpha} + 1\right) \left(1 - \alpha\right)\frac{\text{opt}_k}{\text{opt}_k} + \alpha \cdot \frac{\text{opt}_k}{\text{opt}_k}
\]
and
\[
B(\alpha) \leq 2 \cdot \alpha.
\]
Since \(A(\alpha)\) is decreasing in \(\alpha\) and \(B(\alpha)\) is increasing in \(\alpha\), solving \(A(\alpha) = B(\alpha)\) yields \(\alpha' \approx 0.7147\). The \(k\)-TPH has an approximation ratio of at most \(A(\alpha') \approx 1.4295\).

**6. CONCLUDING REMARKS**

This article developed a polynomial-time approximation algorithm to solve the Steiner tree problem. The proposed algorithm is an efficient two-phase heuristic in greedy strategy that achieves an approximation ratio of 1.4295.

**7. ACKNOWLEDGMENTS**

The work was supported by the Ministry of Science and Technology, Taiwan, under Grant no. MOST 107-2221-E-006 - 090 - .

**8. REFERENCES**

[1] S. Arora. Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. *Journal of the ACM*, 45(5):753–782, 1998.
[2] P. Berman and V. Ramaiyer. Improved approximations for the steiner tree problem. *Journal of Algorithms*, 17(3):381–408, 1994.

[3] A. Borchers and D.-Z. Du. The k-steiner ratio in graphs. *SIAM Journal on Computing*, 26(3):857–869, 1997.

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

[5] A. E. Caldwell., A. B. Kahng, S. Mantik, I. L. Markov, and A. Zelikovsky. On wirelength estimations for row-based placement. In *Proceedings of the 1998 international symposium on Physical design*, pages 4–11, 1998.

[6] M. Chlebík and J. Chlebíková. The steiner tree problem on graphs: Inapproximability results. *Theoretical Computer Science*, 406(3):207–214, 2008.

[7] A. E. Feldmann, J. Könemann, N. Olver, and L. Sanitá. On the equivalence of the bidirected and hypergraphic relaxations for steiner tree. *Mathematical Programming*, 160(1):379–406, Nov 2016.

[8] M. R. Garey and D. S. Johnson. *Computers and intractability*, volume 29. wh freeman, 2002.

[9] M. X. Goemans, N. Olver, T. Rothvoß, and R. Zenklusen. Matroids and integrality gaps for hypergraphic steiner tree relaxations. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, STOC ’12, pages 1161–1176, 2012.

[10] S. Hougardy and H. J. Prömel. A 1.598 approximation algorithm for the steiner problem in graphs. In *Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, pages 448–453, 1999.

[11] F. K. Hwang, D. S. Richards, and P. Winter. *The Steiner Tree Problem*. Elsevier Science Publishers, Amsterdam, 1992.

[12] A. B. Kahng and G. Robins. *On Optimal Interconnections for VLSI*. Kluwer Academic, 1995.

[13] M. Karpinski and A. Zelikovsky. New approximation algorithms for the steiner tree problems. *Journal of Combinatorial Optimization*, 1:47–65, 1997.

[14] B. Korte, H. J. Prömel, and A. Steger. Steiner trees in vlsi-layout. *Paths, Flows, and VLSI-Layout*, pages 185–214, 1990.

[15] W. Liang. Constructing minimum-energy broadcast trees in wireless ad hoc networks. In *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking &amp; Computing*, pages 112–122, 2002.

[16] M. Min, H. Du, X. Jia, C. X. Huang, S. C.-H. Huang, and W. Wu. Improving construction for connected dominating set with steiner tree in wireless sensor networks. *Journal of Global Optimization*, 35(1):111–119, May 2006.

[17] H. J. Prömel and A. Steger. Rnc-approximation algorithms for the steiner problem. In *Proceedings of the 14th Annual Symposium on Theoretical Aspects of Computer Science*, pages 559–570, 1997.

[18] G. Robins and A. Zelikovsky. Tighter bounds for graph steiner tree approximation. *SIAM Journal on Discrete Mathematics*, 19(1):122–134, 2005.

[19] H. Takahashi and A. Matsuyama. An approximate solution for the steiner problem in graphs. *Mathematica Japonica*, 24:573–577, 1980.

[20] A. Zelikovsky. An 11/6-approximation algorithm for the network steiner problem. *Algorithmica*, 9(5):463–470, May 1993.

[21] A. Zelikovsky. Better approximation bounds for the network and euclidean steiner tree problems. In *Technical report CS-96-06, University of Virginia*, 1996.