On Rajagopalan and Vazirani's $\frac{3}{2}$-Approximation Bound for the Iterated 1-Steiner Heuristic

Romeo Rizzi

BRICS*
Department of Computer Science
University of Aarhus
Ny Munkegade
DK-8000 Aarhus C, Denmark
e-mail: `rrizzi@science.unitn.it`

Abstract

Let $G = (V, E)$ be an undirected graph with costs on the edges specified by $w : E \mapsto \mathbb{R}_+$. A *Steiner tree* is any tree of G which spans all nodes in a given subset R of V. When $V \setminus R$ is a stable set of G, then (G, R) is called *quasi-bipartite*. In [3], Rajagopalan and Vazirani introduced the notion of quasi-bipartiteness and showed that the Iterated 1-Steiner heuristic always produces a Steiner tree of total cost at most $\frac{3}{2}$ the optimal when (G, R) is quasi-bipartite and w is a metric. In this paper, we give a more direct and much simpler proof of this result. Next, we show how a bit scaling approach yields a polynomial time implementation of the Iterated 1-Steiner heuristic. This gives a $\frac{3}{2}$-approximation algorithm for the problem considered by Rajagopalan and Vazirani. (We refer however to the recent and independent developments in [4] for better bounds and algorithms). Finally, our bit scaling arguments are not standard and we are the first to adapt bit scaling techniques to the design of approximation algorithms.

Key words: Steiner tree, local search, Iterated 1-Steiner heuristic, bit scaling.

1 Introduction

Let $G = (V, E)$ be an undirected graph with weights on the edges specified by a weighting function $w : E \mapsto \mathbb{R}_+$. A *Steiner tree* is any tree of G which spans all nodes in a given subset R of V. The *metric Steiner tree problem* asks for a Steiner tree of minimum weight, given that w is a metric. (The weight of a tree T is defined as $w(T) = \sum_{e \in T} w(e)$). When $V \setminus R$ is a stable set of G, then (G, R) is called *quasi-bipartite*. In [3], Rajagopalan and Vazirani introduced the notion of quasi-bipartiteness and gave a $(\frac{3}{2} + \epsilon)$-approximation algorithm for the metric Steiner tree problem, when (G, R) is quasi-bipartite. As a byproduct of their achievement, the Iterated 1-Steiner heuristic of Kahng and Robins [1, 2] always produces a Steiner tree of total cost at most $\frac{3}{2}$ the optimal when (G, R) is quasi-bipartite and w is a metric. In this paper, we give a more direct and much simpler derivation of this result. Next, we show how a bit scaling approach yields a polynomial time implementation of the Iterated 1-Steiner heuristic. This gives a $\frac{3}{2}$-approximation algorithm for the problem considered by

*Basic Research in Computer Science,
Centre of the Danish National Research Foundation.

Rajagopalan and Vazirani. We refer however to the recent and independent developments in [4] for better bounds and algorithms. In [4], an independent proof of the $\frac{3}{2}$ bound for the Iterated 1-Steiner heuristic is also given in the context of a more general approach. Still, our direct and simple proof remains of independent interest. Within this proof, we emphasize the role of a fundamental lemma (Lemma 2) whose statement will be shortly introduced just after first notation is given. We finally underline that our bit scaling arguments are not standard and that we are actually the first to adapt bit scaling techniques to the design of approximation algorithms. In particular, no polynomial time implementation of the Iterated 1-Steiner Heuristic of Kahng and Robins [1, 2] was previously known.

notation and key lemma. Denote by $mst(G, w)$ the minimum weight of a spanning tree for (G, w). Where $S \subseteq V$, the *subgraph of G induced by S* is the graph obtained from G by discarding all nodes in $\overline{S} := V \setminus S$ and all edges with at least one endnode in $\overline{S}$. The subgraph of G induced by S can be denoted either by $G[S]$ or by $G \setminus \overline{S}$. In the following, $R \subseteq V$ and $X := V \setminus R$ is assumed to be a *stable set* of G, that is, $G[X] = G \setminus R$ has no edges.

Let T be a minimum weight spanning tree for $(G[R], w)$. Our arguments are based on the following simple and fundamental lemma: if $mst(G[R \cup \{x\}], w) \geq w(T)$ for every $x \in X$, then $mst(G[R \cup X'], w) \geq w(T)$ for every $X' \subseteq X$. Note that, for this lemma to hold, the requirement that X is a stable set can not be omitted.

2 The Iterated 1-Steiner heuristic

Let (G, R, w) be a quasi-bipartite graph endowed by a metric. Let $\hat{T}$ be an optimal Steiner tree for (G, R, w). If we knew which of the nodes of X, say $\hat{I} \subseteq X$, are actually in $V(\hat{T})$, then we could find an optimal Steiner tree by computing a minimum spanning tree of $(G[R \cup \hat{I}], w)$. Moreover, since w is a metric, there always exists an optimal solution $\hat{T}$ such that no node in $\hat{I}$ is incident with less than 3 edges in $\hat{T}$. As observed by Rajagopalan and Vazirani in [3], the following local search algorithm returns a $\frac{3}{2}$-*optimal* Steiner tree, i.e. a Steiner tree $\tilde{T}$ with $w(\tilde{T}) \leq \frac{3}{2} w(\hat{T})$. The algorithm was first introduced by Kahng and Robins [1, 2] as an heuristic for the general Steiner tree problem and is nowadays considered as one of the most popular benchmarks in the field.

Algorithm 1 ITERATED_1-STEINER (G, R, w)

1. $I \leftarrow \emptyset$; $T \leftarrow$ any minimum spanning tree of $(G[R \cup I], w)$;
2. **while** $\exists x \in X \setminus I$ such that $mst(G[R \cup I \cup \{x\}], w) < w(T)$ **do**
3. $\quad$ $I \leftarrow I \cup \{x\}$; $T \leftarrow$ any minimum spanning tree of $(G[R \cup I], w)$;
4. $\quad$ remove from I all nodes with degree one in T; update T accordingly;
   (drop the corresponding leafs);
5. $\quad$ remove from I all nodes with degree two in T; update T accordingly;
   (shortcut the pairs of consecutive edges si and it with the single edges st);
6. **return** T;

We offer a direct and simple proof of the following result. The proof is based on Lemma 2. The lemma is stated below the theorem and proven in the next section.

Theorem 1 *The Steiner tree output by Algorithm 1 is within a factor of $\frac{3}{2}$ from optimum.*

Proof: Let $\tilde{T}$ be the Steiner tree output by Algorithm 1 and let $\hat{T}$ be an optimal Steiner tree for (G, R, w). Let $\tilde{I} = V(\tilde{T}) \setminus R$ and $\hat{I} = V(\hat{T}) \setminus R$. In $\tilde{T}$, consider the stars of the nodes in $\tilde{I}$. Since (G, R) is quasi-bipartite, then these stars are all disjoint. Moreover, by steps 4 and 5, each star contains at least three edges. For every $x \in \tilde{I}$, let e_x be any edge of $\tilde{T}$ incident with x and with smallest possible weight. By the above remarks, $\sum_{x \in \tilde{I}} w(e_x) \leq \frac{1}{3} w(\tilde{T})$. Since for every $x \in \tilde{I}$ one of the two endpoints of e_x is in R, then there exists a spanning tree T of $G[R \cup \hat{I} \cup \tilde{I}]$ with $w(T) \leq w(\hat{T}) + \frac{1}{3} w(\tilde{T})$. (Take any spanning tree in $\hat{T} \cup \{e_x \ : x \in \tilde{I}\}$). By step 3, $mst(G[R \cup \tilde{I} \cup \{x\}], w) \geq w(\tilde{T})$ for every $x \in X \setminus \tilde{I}$. By Lemma 2, $w(T) \geq w(\tilde{T})$. Combining, $w(\hat{T}) + \frac{1}{3} w(\tilde{T}) \geq w(\tilde{T})$. So, $w(\tilde{T}) \leq \frac{3}{2} w(\hat{T})$. □

Lemma 2 *Let $X := V \setminus R$ be a stable set of G. Assume $mst(G[R \cup \{x\}], w) \geq mst(G[R], w)$ for every node $x \in X$. Then $mst(G[R \cup X'], w) \geq mst(G[R], w)$ for every $X' \subseteq X$.*

Note that the assumption of X being stable can not be omitted in the above lemma.

3 A proof of Lemma 2

Let $G = (V, E)$ be an undirected graph and $w : E \mapsto \mathbb{R}_+$ be a non-negative weighting of the edges of G. To prove Lemma 2, we do not need w to be a metric. Instead, we need to introduce some further notation. In this section, it will be convenient to allow our graphs to possess parallel edges. The operation of *identifying a set of nodes S into a node s* amounts to: (1) remove all edges with both endnodes in S; (2) introduce a new node s; (3) for every edge with one endnode s' in S, move that endnode from s' to s; (4) remove all nodes in S. Where $V_1, \ldots, V_k$ is a partition of V, then $G < V_1, \ldots, V_k >$ denotes the graph obtained from G by identifying all nodes of V_i into v_i (for $i = 1, \ldots, k$). Usually, we consider a tree T to be just a set of edges. Sometimes however, and depending on our convenience, a tree T will be regarded as the graph $(V(T), T)$, where $V(T)$ is the set of endnodes of edges in T.

Lemma 3 *Let $v_1, \ldots, v_k$ the neighbors of a node x in G and $C_1, \ldots, C_k$ a partition of $V \setminus \{x\}$ such that $v_i \in C_i$ (for $i = 1, \ldots, k$). Assume $w(\delta(x)) < mst(G[V \setminus \{x\}] < C_1, \ldots, C_k >, w)$. Then $mst(G, w) < mst(G \setminus \{x\}, w)$.*

Proof: Let T be any spanning tree for $(G \setminus \{x\}, w)$. It suffices to show that there always exists a spanning tree F of $T < C_1, \ldots, C_k >$ such that $T \setminus F \cup \delta(x)$ contains a spanning tree of G. Let v be a leaf of T and let vu be the edge of T incident with v. W.l.o.g. assume $v \in C_1$.
Case 1: Assume that $v \neq v_1$. Let $T' = T \setminus \{vu\}$ and G' be the graph obtained from G by identifying $\{u, v\}$. Note that T' is a spanning tree for $G' \setminus \{x\}$. Let F be a spanning tree of $T' < C_1 \setminus \{v\}, C_2, \ldots, C_k >$ such that $T' \setminus F \cup \delta(x)$ contains a spanning tree of G'. But then, F is a spanning tree of $T < C_1, \ldots, C_k >$ such that $T \setminus F \cup \delta(x)$ contains a spanning tree of

G.
Case 2: Assume therefore that $v = v_1$. This time G' and T' are obtained from G and T as follows. First remove edge vu from T, and in G, remove xv and identify $\{u, v\}$. Next, as long as T contains an edge ab with $a \in C_1$, then remove ab from T and in G identify $\{a, b\}$. Let G' and T' be the graph and the tree so obtained. Note that T' is a spanning tree for $G' \setminus \{x\}$. Let F be a spanning tree of $T' < C_2, \ldots, C_k >$ such that $T' \setminus F \cup \delta(x)$ contains a spanning tree of G'. But then, $F \cup \{vu\}$ is a spanning tree of $T < C_1, \ldots, C_k >$ such that $T \setminus F \cup \delta(x)$ contains a spanning tree of G. □

Proof of Lemma 2: It suffices to show that if $mst(G[R \cup X'], w) < mst(G[R], w)$ for some $X' \subseteq X$ with $|X'| \geq 2$, then $mst(G[R \cup X''], w) < mst(G[R], w)$ for some proper subset X'' of X'. Let T' be a spanning tree of $G[R \cup X']$ with $w(T') < mst(G[R], w)$. In a minimal counterexample, we can always assume that $X' = X$ and that every edge of G with an endpoint in X is contained in T'. Let x be any node in X and $X'' = X \setminus \{x\}$. Let $v_1, \ldots, v_k$ be the neighbors of x in G. Consider the connected components $\tilde{C}_1, \ldots, \tilde{C}_k$ of the graph obtained by removing node x from the graph T'. (Assume w.l.o.g. that $v_i \in \tilde{C}_i$, for $i = 1, \ldots, k$). If $mst(G[R \cup X''], w) \leq w(T')$, then $mst(G[R \cup X''], w) < mst(G[R], w)$ and the proof is complete. Assume therefore $mst(G[R \cup X''], w) > w(T')$, which implies $w(\delta(x)) < mst(G[V \setminus \{x\}] < \tilde{C}_1, \ldots, \tilde{C}_k >, w)$. For $i = 1, \ldots, k$, let $C_i = \tilde{C}_i \cap R$. Since X is a stable set of G, then $v_i \in C_i$ for $i = 1, \ldots, k$. Clearly, $G[R] < C_1, \ldots, C_k >= G[V \setminus \{x\}] < \tilde{C}_1, \ldots, \tilde{C}_k >$. Hence $mst(G[R] < C_1, \ldots, C_k >, w) > w(\delta(x))$. By Lemma 3, $mst(G[R \cup \{x\}], w) < mst(G[R], w)$. □

4 Bit scaling and running time

In this section, we show how a bit-scaling technique can be employed to derive an implementation of Algorithm 1 with running time polynomial in the size of the input.

Consider the sequence of weightings $w = w_0, w_1, \ldots$, where, for $i > 0$, w_i is defined as follows: $w_i(e) = \lfloor \frac{w_{i-1}(e)}{2} \rfloor$. Let k be the smallest index for which $w_k(e) \leq 1$ for every edge e of G. Therefore $k \leq log_2 \max_{e \in E} w(e)$. When Algorithm 1 is executed on (G, R, w_k) as input, then loop 2–5 will cycle at most n times, since w_k is a $0,1$-vector. The output will be a tree $T^k{}_{APX}$. Note that $T^k{}_{APX}$ is a $\frac{3}{2}$-optimal Steiner tree for (G, R, w_k).

For $i = 0, 1, \ldots, k$, let $T^i{}_{OPT}$ be an optimal and $T^i{}_{APX}$ be a $\frac{3}{2}$-optimal Steiner tree in (G, w_i). Hence,

$$w_i(T^i{}_{APX}) - w_i(T^{i-1}{}_{OPT}) \leq w_i(T^i{}_{APX}) - w_i(T^i{}_{OPT}) \leq \frac{1}{2} w_i(T^i{}_{OPT})$$

Moreover, since every tree has less that n edges, we have:

$$w_{i-1}(T^i{}_{OPT}) - w_{i-1}(T^{i-1}{}_{OPT}) \leq \Big(2w_i(T^i{}_{OPT}) + n\Big) - 2w_i(T^{i-1}{}_{OPT}) \leq n$$

Therefore,

$$
\begin{aligned}
&w_{i-1}(T^i{}_{APX}) - w_{i-1}(T^{i-1}{}_{OPT}) \leq (2w_i(T^i{}_{APX}) + n) - 2w_i(T^{i-1}{}_{OPT}) \leq \\
&n + 2\left(w_i(T^i{}_{APX}) - w_i(T^{i-1}{}_{OPT})\right) \leq n + 2\left(\tfrac{1}{2}w_i(T^i{}_{OPT})\right) \leq n + \tfrac{2}{2}\left(\tfrac{1}{2}w_{i-1}(T^i{}_{OPT})\right) \leq \\
&n + \tfrac{1}{2}\Big(\ w_{i-1}(T^i{}_{OPT}) + w_{i-1}(T^{i-1}{}_{OPT}) - w_{i-1}(T^{i-1}{}_{OPT})\ \Big) \leq \\
&n + \tfrac{1}{2}w_{i-1}(T^{i-1}_{OPT}) + \tfrac{1}{2}\Big(\ w_{i-1}(T^i{}_{OPT}) - w_{i-1}(T^{i-1}{}_{OPT})\ \Big) \leq \\
&n + \tfrac{1}{2}w_{i-1}(T^{i-1}{}_{OPT}) + \tfrac{1}{2}(n) = \tfrac{3}{2}n + \tfrac{1}{2}w_{i-1}(T^{i-1}{}_{OPT})
\end{aligned}
$$

We conclude that $w_{i-1}(T^i{}_{APX}) \leq \frac{3}{2}n + \frac{3}{2}w_{i-1}(T^{i-1}{}_{OPT})$. Therefore, by executing loop 2–5 at most $\frac{3}{2}n$ times, then Algorithm 1 finds a $\frac{3}{2}$-optimal Steiner tree in (G, w_{i-1}) starting from any $\frac{3}{2}$-optimal Steiner tree in (G, w_i).

The above analysis yields the following result.

Theorem 4 *There is a $\frac{3}{2}$-approximation algorithm for the metric Steiner tree problem on quasi-bipartite graphs. In a straight forward implementation, the running time is $O(nkxT(n,m))$, where $n := |V|$ is the number of nodes, $m := |E|$ is the number of edges, $T(n,m)$ is the maximum time required by a minimum spanning tree computation on a graph with at most n nodes and m edges, $k := log_2 \max_{e \in E} w(e)$, and $x := |X|$ is the number of Steiner nodes.*

No polynomial time implementation of the Iterated 1-Steiner Heuristic of Kahng and Robins [1, 2] was previously known.

Acknowledgments

I thank Devdatt Dubashi for stimulating conversations and an helpful referee for valuable feedback.

References

[1] A.B. Kahng and G. Robins, A new class of iterative Steiner tree heuristics with good performance. *IEEE Transactions on Computer-Aided Design*, 11 (7), 1992, 893–902.

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

[3] S. Rajagopalan and V.V. Vazirani, On the bidirected cut relaxation for the metric Steiner tree problem. *SODA '99* (1999) 742–751.

[4] G. Robins and A. Zelikovsky, Improved Steiner tree approximation in graphs. *SODA 2000*.

[5] R. Rizzi, On the Steiner Tree $\frac{3}{2}$-Approximation for Quasi-Bipartite Graphs. BRICS Report Series, RS-99-39, November 1999.