The Complexity of Approximating the Class Steiner Tree Problem *†

Edmund Ihler ‡

Abstract

Given a connected, undirected distance graph with required classes of nodes and optional Steiner nodes, find a shortest tree containing at least one node of each required class. This problem called Class Steiner Tree is NP-hard and therefore we are dependent on approximation.

In this paper, we investigate various restrictions of the problem comparing their complexities with respect to approximability. A main result is that for an input of trees without Steiner nodes and unit edges only, Class Steiner Tree is as hard to approximate as Minimum Set Cover, for which no constant approximation is known, too. Further we prove that if this restricted version has an approximation scheme, all members of the optimization problem class MAX SNP do.

1 Introduction

A large number of important optimization problems have been shown to be NP-hard, and this number is increasing constantly. Therefore, good approximation algorithms are of great importance. Moreover, there is the question for a general theory, classifying optimization problems by their complexity with respect to approximation. Different approaches have been made toward such a general theory (see e.g. [PY88, PR90, Ihl90]).

In this paper, we want to investigate a minimization problem with respect to the following properties: Is it polynomially solvable or - if not - is

*This work was partly supported by the Deutsche Forschungsgemeinschaft under grant Wid 810/5.

†Appeared in: *Graph-Theoretic Concepts in Computer Science, WG91*, No. 570 of *Lecture Notes in Computer Science*, pages 85–96, Springer, 1992.

‡Institut für Informatik, Universität Freiburg, Rheinstr. 10–12, D-79104 Freiburg; email: ihler@informatik.uni-freiburg.de

an approximation scheme known or at least some constant approximation, or none of the three. Here, *constant approximation* means a polynomial time approximation algorithm whose relative error is bounded by a constant. An *approximation scheme* is a family of constant approximations, one for each arbitrary relative error bound $\epsilon > 0$. We study the frontiers between the mentioned types of complexity by investigating a certain minimization problem that is important in VLSI and network design.

We consider the problem of finding a shortest network, connecting classes of nodes of a graph. To be more precise: Given an undirected connected distance graph where its set of nodes is partitioned into a class of optional *Steiner nodes* and *required classes* of nodes, compute a tree of minimum total length that contains at least one node of each required class. We will call this problem Class Steiner Tree.

Various applications to this problem are possible. For example in the routing phase of VLSI design after the placement of components on a chip, pins of different components have to be connected by an electrical network of minimum total length [RW90]. Using the remaining freedom of rotating and mirroring the components leads to several positions for each required pin. So we can consider in a natural way the possible positions of such a pin as a required class of nodes.

As another possible application, consider the problem of connecting local networks by a minimum global network, independent of the local networks [IRW91]. Assume that we have e.g. existing local networks of computers and several possibilities for new connections with specific costs between computers and/or optional branching points. We can model this problem using the local networks as required classes of nodes with one node for each of their computers. The optional branching points are Steiner nodes.

Clearly, Class Steiner Tree is a natural generalization of the famous Steiner Tree. As Steiner Tree is NP-hard [Kar74], Class Steiner Tree is too. Therefore we depend on approximation. For Steiner Tree, polynomial time algorithms are known that compute an approximate Steiner tree less than twice as long as the minimal Steiner tree (e.g. [KMB81, Meh88]). But it is still an open question, whether there is an approximation scheme.

Some progress has been made toward this question in general. In [PY88], a complexity class MAX SNP for optimization problems has been introduced. Many well known problems like Max 3Sat or Maximum Independent Set and Node Cover for bounded node degree have been proved to be complete in this class under a certain kind of transformation that preserves approximation schemes. All these complete problems share the property that for none of it an approximation scheme has been found yet. But such a scheme for one of it would imply an approximation scheme for each problem in MAX SNP. Hence, for a problem proved to be complete or hard in this sense, there is little hope for an approximation scheme.

Steiner Tree has been shown MAX SNP-hard [BP89]. Similar to NP-hardness, this implies MAX SNP-hardness for Class Steiner Tree. MAX SNP-hard means that the problem is not in MAX SNP but that the reduction required for completeness is possible. There are many results showing that Class Steiner Tree is a harder problem than Steiner Tree. For certain restrictions, e.g., where no Steiner nodes are allowed or where an input graph has to be a tree, the complexities differ considerably: Steiner Tree gets polynomially solvable whereas Class Steiner Tree remains NP-hard [RW90]. Another indication for the higher complexity of Class Steiner Tree is that approximations with bounds for their relative error have been found [Ihl91], but none of the bounds is constant, i.e., independent of the input graph.

In Section 2, we introduce transformations between minimization problems that preserve these approximation properties like the existence of an approximation scheme or of a constant approximation. The main tools are *C-reductions*, transforming polynomial approximation algorithms with a constant bound for their relative error, and *S-reductions* that transform a whole approximation scheme.

In Section 3, we present a strong restricted version of Class Steiner Tree that is polynomially solvable, and how we can modify this problem to an NP-hard but constant approximable version. In Section 4, we work in the other direction: How strong can we restrict Class Steiner Tree preserving the complexity of the general version. We prove in Section 4 that Class Steiner Tree for VLSI oriented graphs where the edge lengths are integers polynomially bounded by the number of nodes (*similarity assumption*, see e.g. [Gab85]) is not easier to approximate if we consider only graphs without Steiner nodes and with all edges of length 1. We show that it is not more complex to find an approximation scheme for graphs where the edge lengths are integers, the similarity assumption holds, and the node degree is bounded by some integer, than for graphs with node degree bounded by 3, without Steiner nodes and with unit edges only.

In Section 5, we give strong evidence that an approximation scheme or at least a constant approximation even for strongly restricted versions is not achievable. There we prove that Class Steiner Tree for trees without Steiner nodes and unit edges only is as hard to approximate as Minimum Set Cover. In Minimum Set Cover, a cover for a set is given by a collection of subsets and we have to find a minimum number of subsets, covering the same original set. The power of this result depends on the fact that it is an old open question whether there exists a constant approximation for Minimum Set Cover and that Minimum Set Cover has been shown to be MAX SNP-hard [PY88].

2 Basic definitions

We describe a *minimization problem* Π by a set $\mathbf{I}$ of *instances*, a collection of finite nonempty sets containing a set $\mathbf{S}(I)$ of *solutions* for each instance $I \in \mathbf{I}$, and a *solution value function* $/\cdot,\cdot/$, mapping the pairs (I, S) with $I \in \mathbf{I}$ and $S \in \mathbf{S}(I)$ into the rational numbers. The problem asks for each instance $I \in \mathbf{I}$ for a minimal solution $S^* \in \mathbf{S}(I)$, i.e., a solution minimizing the solution value function (see e.g. [GJ79]).

A polynomial time algorithm A is an *approximation algorithm (PTA)* for a minimization problem Π, if A finds for each instance $I \in \mathbf{I}$ a solution $A(I) \in \mathbf{S}(I)$. A PTA is called a $(1+\epsilon)$-*CPTA* or just a *CPTA* for Π, if for each $I \in \mathbf{I}$: $/I, A(I)/ \leq (1+\epsilon)/I, S^*(I)/$, and $\epsilon > 0$ is a constant.

A *polynomial time approximation scheme (PTAS)* for a minimization problem Π is a family $(A_\epsilon)_{\epsilon>0}$ of CPTAs, containing a $(1+\epsilon)$-CPTA for Π for each $\epsilon > 0$.

We will now define transformations between minimization problems, that preserve CPTAs or whole PTASs. For two minimization problems Π, Π', we call a pair of polynomial transformations (f, g) with an *instance transformation* $f : \mathbf{I} \to (\mathbf{I}')^m$, and a *solution transformation* $g : \mathbf{I} \times (\mathbf{I}')^m \times (\mathbf{S}'(\mathbf{I}'))^m \to \mathbf{S}(\mathbf{I})$ a *(multi-) C-reduction*, denoted by $\Pi \leq_C \Pi'$, if whenever A' is a CPTA for Π', then $g \circ (\cdot, f, A' \circ f)$ is a CPTA for Π. Here $A' \circ f$ is defined by $(A' \circ f_1, \ldots, A' \circ f_m)$ where $f = (f_1, \ldots, f_m)$.

If $(f_\epsilon, g_\epsilon, e)_{\epsilon>0}$, where $(f_\epsilon, g_\epsilon)_{\epsilon>0}$ is a family of C-reductions, one for each $\epsilon > 0$, and e, called the *error transformation*, is a polynomial transformation $e : (0,\infty) \to (0,\infty)^m$, then this triple is called a *(multi-) S-reduction*, denoted by $\Pi \leq_S \Pi'$, if whenever $(A'_{\epsilon'})_{\epsilon'>0}$ is a PTAS for Π', then $g_\epsilon \circ (\cdot, f_\epsilon, A'_{e(\epsilon)} \circ f_\epsilon)_{\epsilon>0}$ is a PTAS for Π. Here $A'_{e(\epsilon)} \circ f_\epsilon$ is defined for each $\epsilon > 0$ by $(A'_{e_1(\epsilon)} \circ f_{\epsilon 1}, \ldots, A'_{e_m(\epsilon)} \circ f_{\epsilon m})$ where $f_\epsilon = (f_{\epsilon 1}, \ldots, f_{\epsilon m})$ and $e = (e_1, \ldots, e_m)$. If e is the identity mapping (in all components), then we call $(f_\epsilon, g_\epsilon)_{\epsilon>0}$ a *(multi-) S^{id}-reduction.* Usually $(f_\epsilon, g_\epsilon, e)_{\epsilon>0}$ is given by a triple (f, g, e), independent of ϵ, and $m = 1$. If we have $\Pi \leq_{S^{id}} \Pi'$ and $\Pi' \leq_{S^{id}} \Pi$, then we write $\Pi \equiv_{S^{id}} \Pi'$. It is an important consequence of our definitions that C-reductions as well as S-reductions compose. To see this, note that the transformations f, e can have at most a polynomial number of polynomial components, because f, e itself have to be polynomially computable. This implies that the corresponding $\leq$-relations are transitive.

For distance graphs (V, E, l), we use standard notation, where V is the set of *nodes*, E is the set of *edges*, and l maps the edges to nonnegative *edge lengths.* We describe *simple paths* between two nodes $v, w \in V$ by a sequence of nodes $(v, v_1, \ldots, v_k, w)$, where each node of the sequence appears once. The *(total) length* of a graph, a subgraph, or a path is given by the sum of the lengths of all its edges.

We define the problem Class Steiner Tree as follows. Let $G = (V, E, l)$ be a connected, undirected distance graph, where $V = S \cup \bigcup_{i=1}^{c} R_i$ is partitioned into disjoint sets: the *required classes* R_i, $1 \le i \le c$, and the *Steiner nodes* S. A *class Steiner tree* is a tree that contains at least one node for each required class. We ask for a class Steiner tree of minimum total length. If we restrict Class Steiner Tree to classes where $|R_i| = 1$ for $1 \le i \le c$, this defines the classical Steiner Tree problem.

After all this definitions, let us first give two simple versions of Class Steiner Tree in the next section: A polynomial solvable one and a NP-hard one, together with a CPTA.

3 A polynomial and a constant approximable version

For many restrictions of the input graph, Class Steiner Tree remains NP-hard, whereas other classical NP-complete problems become polynomially solvable. This holds e.g. for input graphs that are trees without Steiner nodes, having unit edges only, a node degree and a cardinality for the required classes both bounded by 3 [RW90, IRW91]. Also versions of Class Steiner Tree embedded in the plane and in the unit grid have been proved NP-hard [IRW91]. Further restrictions are possible.

But where is the frontier between NP-hard and polynomial versions? Consider a restriction where the node degree is bounded by 2. Because the graph has to be connected, this implies a linear order to the nodes. So we can compute a minimum class Steiner tree as follows. For each node, we add recursively the next node and the edge connecting these nodes until all classes are represented. Then we just have to take a shortest of these subgraphs, and we have found a (degenerated) minimum class Steiner tree.

There is another less trivial polynomial version given in [Ihl91], where the number of classes is bounded. Slightly modified, this version becomes NP-hard, but we can give a CPTA. Here, not the total number of required classes is bounded by k, but the number of required classes that contain more than one node. This new version Class Steiner Tree$(|\{R_i : |R_i| > 1\}| \le k)$ is a generalization of Steiner Tree and therefore is NP-hard. But there is a simple way to apply any CPTA known for Steiner Tree to this problem.

Theorem 1 Class Steiner Tree*$(|\{R_i : |R_i| > 1\}| \le k)$ $\equiv_{S^{id}}$* Steiner Tree.

Proof: Let the graph G have n nodes and let the required classes R_i have n_i nodes, then $\prod_{i=1}^{c} n_i \le n^k$ different tuples $(r_1, \ldots, r_c) \in R_1 \times \cdots \times R_c$ are possible. A transformation f can now create by its component f_i for each such tuple i a graph G'_i for Steiner Tree, by using the original graph, and interpreting the nodes of the tuple as required nodes and the remaining nodes as Steiner nodes. If a CPTA computes a Steiner tree for each of the graphs G'_i, a transformation g can pick a shortest one of the computed trees and reinterpret it for G. Clearly, this describes a CPTA for G, because the number of graphs G'_i is bounded by $O(n^k)$, and there is a tuple i where the length of the minimal Steiner tree for G_i equals this of the minimal class Steiner tree for G. The relative error bound is the same. Hence (f, g) is a multi S^{id}-reduction from Class Steiner Tree to Steiner Tree, and by a trivial S^{id}-reduction in the other direction we have proved the theorem. □

It can easily be seen that the same version without Steiner nodes gets polynomially solvable using $O(n^k)$ minimal spanning tree computations.

In this section, we considered very restricted versions. Slightly more general cases of Class Steiner Tree become considerably more complex. In the next section, we give a restricted version that is as hard to approximate as the general one.

4 Versions of equal complexity

How strong can we restrict Class Steiner Tree preserving the complexity with respect to approximation of the general version? Sometimes, it is easier to find a CPTA or a PTAS for a restricted version, but can we apply this to more general cases? The following theorems aim at this questions. First, we show that a PTAS for Class Steiner Tree restricted to graphs without Steiner nodes gives a PTAS for the general Class Steiner Tree and vice versa.

Theorem 2 Class Steiner Tree $\equiv_{S^{id}}$ Class Steiner Tree*($S = \emptyset$).*

Proof: We have to construct S^{id}-reductions in both directions.

"$\leq$": For our instance transformation f, we start with the graph $G = (S \cup \bigcup_{i=1}^{c} R_i, E, l)$ where the number of Steiner nodes is s. Introduce for each $v \in S$ a new class and call these classes $R_{c+1}, \ldots, R_{c+s}$. Interprete each Steiner node as a node of its corresponding new required class. We call all nodes in this slightly modified graph *original nodes*. Now replace each edge $(v, w) \in E$ by a simple path $p(v, w) = (v, r_1, \ldots, r_s, w)$. The nodes $r_1, \ldots, r_s$ should be new nodes with $(r_1, \ldots, r_s) \in R_{c+1} \times \cdots \times R_{c+s}$ for each edge in E, the length of each of the new edges be $l((v, w))$. Hence, each of the classes R_{c+i}, $1 \leq i \leq s$, consists of $|E| + 1$ nodes, and the number of edges in the new graph G' is $|E|(s + 1)$ (see Figure 1).

The solution transformation g works as follows. Given a class Steiner tree T' for G', take the original nodes that are in T' and reinterpret it. Whenever two original nodes are connected in T' by a simple path of $s + 1$

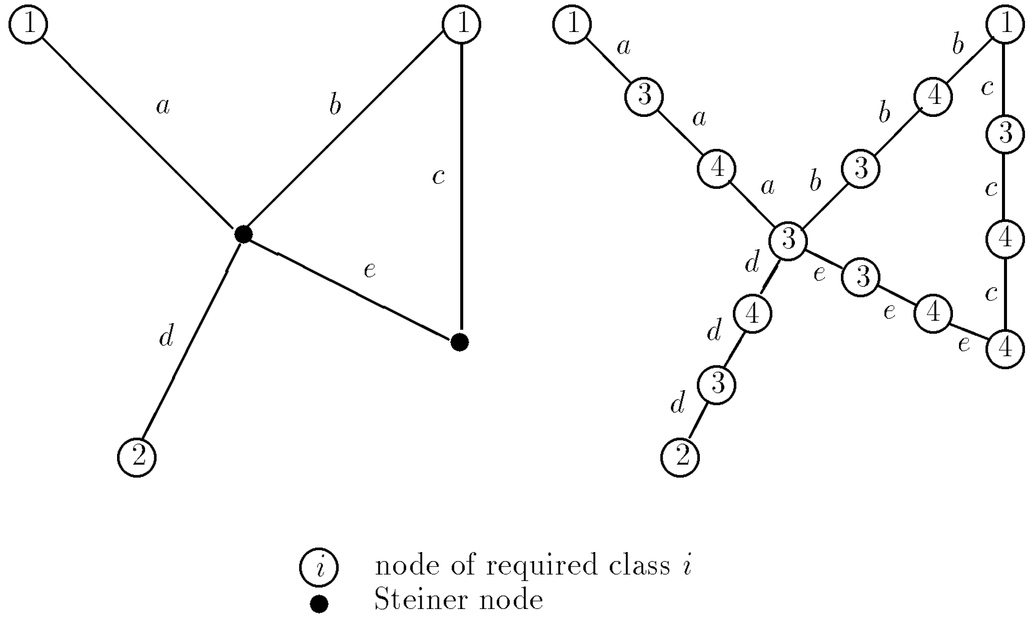

Figure 1: An example graph G and its image G' transformed by f of the proof for Theorem 2.

edges, then choose the corresponding edge in E. Clearly, these edges and the reinterpreted nodes build a class Steiner tree T for G, and $(s+1)l(T) \leq l(T')$.

Assume that we have given a $(1+\epsilon')$-CPTA A' that computes a class Steiner tree $T'_{A'}$ for each input graph G' of Class Steiner Tree($S = \emptyset$), then this implies a $(1+\epsilon')$-CPTA $A = g \circ (\cdot, f, A' \circ f)$ computing T_A for G, because of the following: A minimum class Steiner tree T_{min} for G leads to a class Steiner tree T' for G' by taking for each edge (v, w) of T_{min} the corresponding simple path $p(v, w)$ in G'. Hence, for a minimum class Steiner tree T'_{min} of G' this yields $(s+1)l(T_{min}) = l(T') \geq l(T'_{min})$, and altogether $\frac{l(T_A)}{l(T_{min})} \leq \frac{(s+1)}{(s+1)} \frac{l(T'_{A'})}{l(T'_{min})} \leq 1 + \epsilon'$. Note that both, f and g can be computed in polynomial time.

"$\geq$": Obvious by a trivial S^{id}-reduction. □

The effect of this theorem is, that all results that we can derive for the restricted version where no Steiner nodes are allowed, also hold for the general version.

A similar theorem is possible for Class Steiner Tree(l : *integers*, *sim. ass.*), where the lengths of all edges are integers and bounded by a polynomial function in the number of nodes, and Class Steiner Tree($l = 1$), a restriction to graphs with unit edges only. This polynomially bounding of the edge lengths is called *similarity assumption* (see e.g. [Gab85]). Both assumptions are very natural and hold, e.g., for graphs derived from the VLSI layout geometry, where the edge lengths are multiples of a certain layout unit.

Theorem 3 CLASS STEINER TREE*(l : integers, sim. ass.)* $\equiv_{S^{id}}$ CLASS STEINER TREE*($l = 1$).*

Proof: Again, we have to construct S^{id}-reductions in both directions.

"$\leq$": We give a new transformation $f : G \mapsto G'$ for a graph $G = (S \cup \bigcup_{i=1}^{c} R_i, E, l)$. Replace each $(v, w) \in E$ by a simple path $p(v, w) = (v, s_1, \ldots, s_{l(v,w)-1}, w)$ where all nodes $s_1, \ldots, s_{l(v,w)-1}$ are new Steiner nodes and all edges have length 1.

Our solution transformation g starts with a given class Steiner tree T' for G'. For each edge $(v, w) \in E$ test, whether the corresponding path $p(v, w)$ is completely contained in T'. All such edges determine the tree T.

Analogously to the proof of Theorem 2, we can prove that (f, g) is an S^{id}-reduction. It is important that f and g are polynomial transformations, as the similarity assumption holds.

"$\geq$": This direction is obvious. $\Box$

Corollary 1 CLASS STEINER TREE*(l : integers, sim. ass.)* $\equiv_{S^{id}}$ CLASS STEINER TREE*($S = \emptyset, l = 1$).*

Proof: We use a composition of the S^{id}-reductions given in the proves of Theorem 3 and 2. $\Box$

We further show equivalence by S-reduction between CLASS STEINER TREE($l = 1$, $deg(v) \leq k$), where all edges have unit length and the node degree is bounded by a constant k, and the restriction CLASS STEINER TREE(l : *integers*, *sim. ass.*, $deg(v) \leq 3$). To this end, we want to use a certain family of graphs. Each of the graphs, called *i-star*, consists roughly spoken of three binary trees, where the root nodes are connected to a common *center node*. We recursively define this family as follows. A 1-star consists of a single center node, and an i-star S_i for $i \geq 2$ is built from an $(i-1)$-star S_{i-1} by finding a node v of S_{i-1} with minimum distance from the center node and with a node degree $deg(v) < 3$, and adding a new node w by a new edge (v, w) of length 1. Clearly, an i-star S_i is a tree, has i nodes and a total length of $(i-1)$. If we allow a node degree of 3 for all the nodes, then we can connect exactly $(2 + i)$ external edges to S_i. We call this number the *connectivity degree* of S_i, and the nodes with node degree less than 3 *connector nodes* (see Figure 2).

Theorem 4 CLASS STEINER TREE*($l = 1$, $deg(v) \leq k$)* $\equiv_S$ CLASS STEINER TREE*(l : integers, sim. ass., $deg(v) \leq 3$), for $k > 3$.*

Proof: "$\leq$" We transfer an instance graph G for an arbitrary $\epsilon > 0$ by the following f_ϵ to G'. Each node has a degree not exceeding k. Hence, we replace each node v of G by a $(k-2)$-star. Hereby we can connect each edge incident to v to a connector node of the $(k-2)$-star, because its connectivity

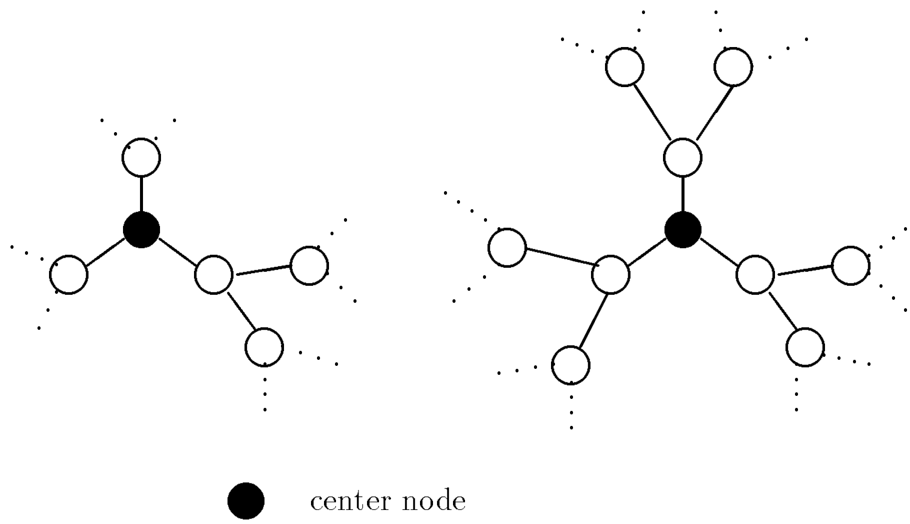

Figure 2: A 6-star and a 10-star.

degree is k. If v is a required node, all nodes, especially the center node, of the $(k-2)$-star are added to the required class of its corresponding node v; otherwise all nodes are additional Steiner nodes. At last we change the lengths of the old unit edges to $2m(k-3)$, where m is an arbitrary positive integer with $m \geq \frac{1}{\epsilon}$. Hence, we have integer lengths for all edges and the similarity assumption holds for any constant ϵ, too.

A solution transformation g independent of ϵ takes all center nodes that are contained in a class Steiner tree T' of G'. A class Steiner tree T of G then consists of all nodes of G corresponding to these center nodes, and of the corresponding edges of T'. We can verify $2m(k-3)l(T) \leq l(T')$.

For the error transformation we use $e(\epsilon) = \frac{\epsilon}{2+\epsilon}$. All given transformations are polynomial in time by construction for any constant ϵ.

Assume a minimum class Steiner tree T_{min} for G, then we can build a class Steiner tree T' for G' by using for a leaf of T_{min} the corresponding connector node as a leaf for T', for the remaining nodes the whole corresponding $(k-2)$-star, and the corresponding edges of T_{min}. We have $2m(k-3)l(T_{min}) + (l(T_{min}) + 1 - 1)(k-3)) \geq l(T')$, because T_{min} consists of unit edges and therefore has $l(T_{min}) + 1$ nodes, each tree has at least one leaf, and the total length of a $(k-2)$-star is $(k-3)$. For a minimum class Steiner tree T'_{min} for G' this implies $(2m+1)(k-3)l(T_{min}) \geq l(T'_{min})$. The chosen e guarantees that a given PTAS $(A'_{\epsilon'})_{\epsilon'>0}$ computing a class Steiner tree $T'_{A'_{\epsilon'}}$ changes by $(f_\epsilon, g, e)_{\epsilon>0}$ to a PTAS $(A_\epsilon)_{\epsilon>0}$ for G computing T_{A_ϵ}:

$$\frac{l(T_{A_\epsilon})}{l(T_{min})} \leq \frac{(2m+1)(k-3)}{2m(k-3)} \frac{l(T'_{A'_{e(\epsilon)}})}{l(T'_{min})} \leq (1+\frac{\epsilon}{2})(1+\frac{\epsilon}{2+\epsilon}) = 1+\epsilon.$$

"$\geq$" Use the S-reduction given in the proof of Theorem 3. □

Corollary 2 CLASS STEINER TREE*(l : integers, sim. ass., $deg(v) \leq k$)* $\equiv_S$ CLASS STEINER TREE*($S = \emptyset$, $l = 1$, $deg(v) \leq 3$), for $k > 3$.*

Proof: Compose the S-reductions in the proofs of Theorems 3,4,3,2 in this order. □

Remember that an equivalence $\equiv_S$ guarantees that a PTAS for one of the problems implies a PTAS for the other problem. If a single $(1+\epsilon)$-CPTA is known for some ϵ for one of the problems, we have a CPTA for the other problem.

In the next section, we relate the complexity of approximating the CLASS STEINER TREE to the one of an old problem. This explains the lack of progress in finding a constant approximation for our problem.

5 Class Steiner Tree is hard to approximate

We saw that CLASS STEINER TREE remains NP-hard for strong restrictions, especially when the input graph is a tree, free of Steiner nodes, and all edges have length 1. In this section, we strengthen the conjecture that there is no CPTA for CLASS STEINER TREE even for the mentioned restrictions. We prove that such an algorithm would imply a CPTA for MINIMUM SET COVER, and exactly for this problem the question for a CPTA is still open.

MINIMUM SET COVER is the following minimization problem: Given a finite nonempty collection $C = \{C_1, \ldots, C_n\}$ of nonempty sets C_i, find a subcollection $S = \{S_1, \ldots, S_s\} \subseteq C$ of minimum cardinality $|S|$ being a cover, i.e. $\bigcup_{i=1}^{s} S_i = \bigcup_{i=1}^{n} C_i$.

To demonstrate the kind of reduction we will use, first consider CLASS STEINER TREE($S = \emptyset, tree, l = 0, 1$) where only trees without Steiner nodes and edge lengths 0 and 1 are allowed for input.

Theorem 5 MINIMUM SET COVER $\leq_{S^{id}}$ CLASS STEINER TREE*($S = \emptyset, tree, l = 0, 1$).*

Proof: We have to give polynomial transformations for the S^{id}-reduction. For the instance transformation f, create for each set C_i in the collection C a *set-node* r_i, for each member μ of each set C_i a *member-node* $r_{i\mu}$, and one *root-node* r_0. All these nodes form the set V. $\{r_0\}$, $\{r_i : 1 \leq i \leq n\}$ are two required classes. For each member $\mu \in \bigcup_{i=1}^{n} C_i$, we create a class $R_\mu = \{r_{i\nu} : \nu = \mu; 1 \leq i \leq n; r_{i\mu} \in V\}$. Connect each set-node r_i by an edge of length 1 to the root-node and each member-node $r_{i\mu}$ by an edge of length 0 to the corresponding set-node r_i. These edges form E, $G = (V, E, l)$ is a tree without Steiner nodes, and the constructed edges define l (see Figure 3).

The solution transformation g is obvious. Given a class Steiner tree T for G, pick all set-nodes r_i of T, then the corresponding sets S_i build a subcollection S which is really a cover for C and $|S| = l(T)$.

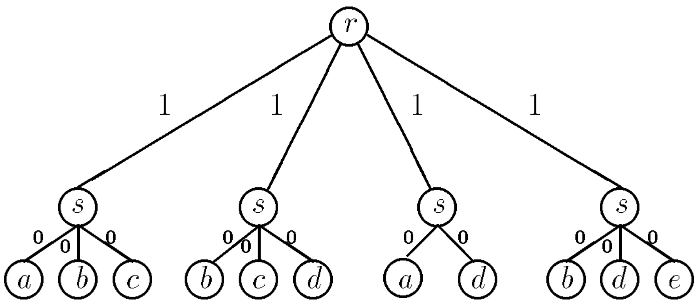

Figure 3: An example collection C and its image graph G transformed by f of the proof for Theorem 5.

Both, f and g, are polynomial time transformations and from a given $(1+\epsilon')$-CPTA A' for G computing a class Steiner tree $T_{A'}$, we derive a $(1+\epsilon')$-CPTA A computing a cover S_A using f and g: A minimum cover S_{min} for C implies a class Steiner tree T for G, if we use the set-nodes and member-nodes related to the sets of S_{min}, together with the root-node and all edges between these nodes. Clearly, for a minimum class Steiner tree T_{min}, we have $|S_{min}| = l(T) \geq l(T_{min})$, and therefore $\frac{|S_A|}{|S_{min}|} \leq \frac{l(T_{A'})}{l(T_{min})} \leq 1+\epsilon'$. $\square$

A simple but interesting corollary results from the fact that Minimum Set Cover, restricted to collections where for each set its cardinality is bounded by a constant, as well as the number of appearances of each member, was proved MAX SNP-complete [PY88].

Corollary 3 Class Steiner Tree*($S = \emptyset, tree, l = 0,1$) is MAX SNP-hard.*

Proof: MAX SNP-hard means that the problem is not in MAX SNP but that the reduction required for completeness is possible. The given S^{id}-reduction in the proof of Theorem 5 works for this bounded case of Minimum Set Cover, too, and has the form of the required reduction (see [PY88]). $\square$

As edges of the length 0 are not adequate for many of the applications of Class Steiner Tree, further restriction to trees built of unit edges are of interest. We show similar approximation results for this NP-hard version.

Theorem 6 Minimum Set Cover $\leq_S$ Class Steiner Tree*($S = \emptyset, tree, l = 1$).*

Proof: To describe the desired instance transformation f_ϵ for an arbitrary $\epsilon > 0$, recall the transformation f given in the proof of Theorem 5. As before, we create a tree with nodes r_0, r_i, $r_{i\mu}$, and edges of length 0 and 1, and with the classes $\{r_0\}$, $\{r_i : 1 \leq i \leq n\}$, and R_μ for $\mu \in \bigcup_{i=1}^{n} C_i$ by a given collection C. Then we choose an arbitrary integer $k \geq \frac{m}{\sqrt{1+\epsilon}-1}$, where m is the cardinality of $\bigcup_{i=1}^{n} C_i$. Replace each edge (r_0, r_i) of length 1 between the root-node r_0 and a set-node r_i by a simple path $p(r_0, r_i) = (r_0, v_{i1}, \ldots, v_{ik-1}, r_i)$ where all the v_{ij} are new nodes for each replaced edge and belong to the class of r_i. After this, replace all edges in the current graph by edges of length 1. In this way, we get a graph G that is a tree with the desired properties.

The solution transformation g_ϵ is given by g in the proof of Theorem 5, independent of ϵ. Pick the sets S_i related to the set-nodes r_i contained in a class Steiner tree T for G to generate a cover S. It is easy to verify that

$$k|S| + m \leq l(T). \tag{1}$$

We give the error transformation e by $e(\epsilon) = \sqrt{1+\epsilon} - 1$. Note, that each of $f_\epsilon, g_\epsilon, e$ can be computed in polynomial time.

It remains to show that our reduction $(f_\epsilon, g, e)_{\epsilon>0}$ transforms each given PTAS $(A'_{\epsilon'})_{\epsilon'>0}$ for Class Steiner Tree$(S = \emptyset, tree, l = 1)$ to a PTAS $(A_\epsilon)_{\epsilon>0}$ for Minimum Set Cover. Start with a minimum cover S_{min} and take for each set S_{min_i} of this cover the corresponding set-node r_i of G and each simple path $p(r_0, r_i)$. Each member of $\bigcup_{i=1}^{n} C_i$ lies in at least one set S_{min_i}, hence, we can easily contact a member node for each such member by an edge of length 1 to one of the chosen set-nodes. This implies for the constructed class Steiner tree T and a minimum class Steiner tree T_{min} of G

$$k|S_{min}| + m = l(T) \geq l(T_{min}). \tag{2}$$

In addition we know that $m \geq 1$, therefore $|S_{min}| \geq 1$. Remember that our k guarantees $k(\sqrt{1+\epsilon} - 1) \geq m$, hence the following holds:

$$\begin{aligned} k|S_{min}| &= \frac{1}{\sqrt{1+\epsilon}}(k|S_{min}| + k(\sqrt{1+\epsilon} - 1)|S_{min}|) \\ &\geq \frac{1}{\sqrt{1+\epsilon}}(k|S_{min}| + m) \end{aligned} \tag{3}$$

Let S_{A_ϵ} be the cover derived by a given PTAS $(A'_{\epsilon'})_{\epsilon'>0}$ for G using $(f_\epsilon, g, e)_{\epsilon>0}$, then the inequalities (1),(2),(3), and the fact $k|S_{min}| > 0$ imply

$$\begin{aligned} \frac{|S_{A_\epsilon}|}{|S_{min}|} &< \frac{|S_{A_\epsilon}|}{|S_{min}|} + \frac{m}{k|S_{min}|} = \frac{k|S_{A_\epsilon}| + m}{k|S_{min}|} \\ &\leq \frac{l(T_{A'_{e(\epsilon)}})}{\frac{1}{\sqrt{1+\epsilon}}(k|S_{min}| + m)} \leq \frac{l(T_{A'_{e(\epsilon)}})}{l(T_{min})}\sqrt{1+\epsilon} \\ &\leq (1 + e(\epsilon))\sqrt{1+\epsilon} = (1 + \sqrt{1+\epsilon} - 1)\sqrt{1+\epsilon} \\ &= 1 + \epsilon \qquad \square \end{aligned}$$

Note that the instance transformations given in the proofs of Theorems 5 and 6 serve also as Turing reductions from MINIMUM SET COVER to the corresponding CLASS STEINER TREE version for an alternative proof of its NP-hardness. We can derive a similar corollary to Corollary 3.

Corollary 4 *If a PTAS for* CLASS STEINER TREE$(S = \emptyset, tree, l = 1)$ *is found, then a PTAS is found for each problem of MAX SNP.* □

6 Conclusion

We investigated the complexity of finding a constant approximation or an approximation scheme for CLASS STEINER TREE, a natural generalization of the STEINER TREE problem, where we have required classes of nodes instead of simply required nodes. We gave a polynomial time algorithm for a strong restricted version of CLASS STEINER TREE and a polynomial time approximation algorithm with constant error bound for an NP-hard version. But most nontrivial restrictions are NP-hard. We proved the equivalence of the complexity of approximation for some restrictions and the general version, and we proved that most of it are at least as hard to approximate as the well known problem MINIMUM SET COVER, for which no constant approximation is known yet, too. Moreover, an approximation scheme would imply such a scheme for all problems of the complexity class MAX SNP.

We have been still unsuccessful in proving this kind of hardness for NP-hard restrictions where the graph is embedded in the plane or in the unit grid. Maybe this gives some hope for constant approximability in these cases.

Acknowledgement

Thanks are due to P. Widmayer and G. Reich for helpful discussions and critical comments.

References

[BP89] Marshall Bern and Paul Plassmann. The Steiner problem with edge lengths 1 and 2. *Information Processing Letters*, 32:171–176, 1989.

[Gab85] H.N. Gabow. Scaling algorithms for network problems. *J. Comp. Sys. Sc.*, 31:148–168, 1985.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, 1979.

[Ihl90] Edmund Ihler. Approximation and existential second-order logic. Technical report, Institut für Informatik, Universität Freiburg, 1990.

[Ihl91] Edmund Ihler. Bounds on the quality of approximate solutions to the Group Steiner Problem. In *Graph-Theoretic Concepts in Computer Science, WG90*, volume 484 of *Lecture Notes in Computer Science*, pages 109–118. Springer, 1991.

[IRW91] Edmund Ihler, Gabriele Reich, and Peter Widmayer. On shortest networks for classes of points in the plane. Technical report, Institut für Informatik, Universität Freiburg, 1991.

[Kar74] Richard W. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1974.

[KMB81] L. Kou, G. Markowsky, and L. Berman. A fast algorithm for Steiner trees. *Acta Informatica*, 15:141–145, 1981.

[Meh88] Kurt Mehlhorn. A faster approximation algorithm for the Steiner problem in graphs. *Information Processing Letters*, 27:125–128, 1988.

[PR90] Alessandro Panconesi and Desh Ranjan. Quantifiers and approximation. In *Proc. 22th Annual ACM Symp. on Theory of Computing*, pages 446–456, 1990.

[PY88] Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. In *Proc. 20th Annual ACM Symp. on Theory of Computing*, pages 229–234, 1988.

[RW90] Gabriele Reich and Peter Widmayer. Beyond Steiner's problem: A VLSI oriented generalization. In *Graph-Theoretic Concepts in Computer Science, WG89*, volume 411 of *Lecture Notes in Computer Science*. Springer, 1990.