On the Approximability of Dense Steiner Problems

M. Hauptmann*

February 15, 2008

Abstract

The ϵ-Dense Steiner Tree Problem was defined by Karpinski and Zelikovsky [11] who proved that for each $\epsilon > 0$, this problem admits a PTAS. Based on their method we consider here dense versions of various Steiner Tree problems. In particular, we give polynomial time approximation schemes for the *ϵ-Dense k-Steiner Tree Problem*, the *ϵ-Dense Prize Collecting Steiner Tree Problem*, the *ϵ-Dense k-Steiner Tree Problem* and the *ϵ-Dense Group Steiner Tree Problem*. For the dense version of the *Steiner Forest Problem* we obtain an approximation algorithm that performs well if the number of terminal sets is small compared to the total number of terminals.

1 Introduction

Given a graph $G = (V, E)$, a cost function $c\colon E \to \mathbb{R}_+$ and a subset $S \subseteq V$ of the vertices of G, a Steiner Tree T for S in G is a subtree of G that all vertices from S. The elements of S are called terminals. The Steiner Tree Problem (STP) is: Given G, c and S as above, find a Steiner Tree T for S in G of minimum cost $c(T) = \sum_{e\in E(T)} c(e)$.

The Steiner Tree Problem is one of the fundamental network design problems with applications ranging from transportation networks, energy supply and broadcast problems to VLSI design and Internet Routing. The currently best known Steiner Tree approximation algorithm is due to Robins and Zelikovsky [13] and achieves a ratio of ≈ 1.55. On the other hand, Chlebik and Chlebikova [3] proved that the Steiner Tree Problem is NP-hard to approximate within ratio 1.01063.

*Dept. of Computer Science, University of Bonn. Email:hauptman@cs.uni-bonn.de

Numerous results for special cases of the Steiner Tree Problem are known. Robins and Zelikovsky [13] obtain a better approximation ratio for the quasi-bipartite case. Arora [2] gives a polynomial time approximation scheme (PTAS) for geometric instances in constant dimension. Karpinski and Zelikovsky [11] construct a PTAS for the ϵ-Dense Steiner Tree Problem.

In this paper we consider dense versions of the Steiner Tree Problem and of some of its most important generalizations, namely the Steiner Forest Problem, the k-Steiner Problem, the Prize Collecting Steiner Tree Problem and the Group Steiner Tree Problem. Let us give the definitions of these problems and of their dense versions and state results that are known for these problems.

The ϵ-Dense Steiner Tree Problem

The ϵ-Dense Steiner Tree Problem was introduced by Karpinski and Zelikovsky [11]. An instance of the Steiner Tree Problem in Graphs consisting of graph $G = (V, E)$ and terminal set $S \subseteq V$ is called ϵ-dense if each terminal $s \in S$ has at least $\epsilon \cdot |V \setminus S|$ neighbours in $V \setminus S$. Karpinski and Zelikovsky obtain the following result: For every $\epsilon > 0$ there is a polynomial time approximation scheme for the ϵ-Dense Steiner Tree Problem. The idea of their algorithm is to perform first a number of greedy steps contracting stars, where a star always consists of a non-terminal and all the terminals that are connected to it. The greedy phase reduces the number of terminals to a constant while adding extra cost $\delta \cdot |S|$. In a second phase, the remaining problem is solved by the Tree Enumeration Algorithm.

The following difficulty occurs. The greedy contractions are not necessarily disjoint and may therefore affect the density. Hence we will give a slightly modified approach where the greedy contractions are disjoint, hence the density condition is preserved. This ends in a remaining instance of $O(\log(|S|))$ terminals, and using the Dreyfus-Wagner algorithm we can solve the remaining problem to optimality. In this way we obtain a PTAS which is not efficient.

However, we will show that after such a greedy phase in a sense the density is not completely destroyed, and after $O(\log^*(|S|)$ such greedy phases we obtain a remaining instance with a constant number of terminals. This approach gives an efficient PTAS for the ϵ-Dense Steiner Tree Problem.

We will furthermore consider a relaxation of the density condition which we call log-density. Roughly, an instance is log-dense if all subsets of the terminal set of size at least $\log(|S|)$ satisfy the average-density condition. The precise definition is given in the next section. For this density condition we will also obtain a PTAS.

Steiner Forest Problem: Given a graph $G = (V, E)$ with edge costs $c\colon E \to \mathbb{R}_+$ and pairwise disjoint nonempty terminal sets $S_1, \ldots, S_n \subseteq V$, find a forest $F = (V(F), E(F)) \subseteq G$ of minimum cost such that for all $1 \leq i \leq n$ S_i is contained in a connected component of F. The best known approximation algorithms for the Steiner Forest Problem achieve ratio $2 \cdot (1 - |S|^{-1})$ and are based on the primal-dual method (see for instance [12], [6]). An instance of the Steiner Forest Problem is called ϵ-dense if it consist of an unweighted graph G and terminal sets $S_1, \ldots, S_m$ such that for every $1 \leq i \leq m$ and every $s \in S_i$ the number of neighbours of s in $V \setminus S_i$ is at least $\epsilon \cdot |V \setminus S_i|$ We will give a polynomial time approximation schemes for instances where the number of terminal sets is small compared to the total number of terminals. The precise statement is given below.

Prize Collecting Steiner Tree Problem (PSTP): Given a graph $G = (V, E)$ with edge costs $c\colon E \to \mathbb{R}_+$ and a terminal set $S \subseteq V$ with price function $p\colon S \to \mathbb{R}_+$, find a tree $T \subseteq G$ connecting a subset S' of S such as to minimize $c(T) + p(S \setminus S')$. Again, the best known approximation ratio for the general case is 2, based on the primal-dual method [6] Here we take the same density condition as for the Steiner Tree Problem: An instance is called ϵ-denes if every terminal has at least $\epsilon \cdot |V \setminus S|$ neighbours in $V \setminus S$. We will give a polynomial time approximation scheme for the ϵ-Dense PSTP.

k-Steiner Tree Problem (k-STP): Given a graph $G = (V, E)$ with edge costs $c\colon E \to \mathbb{R}_+$, a terminal set $S \subseteq V$ and a number $k \in [1, |S|]$, find a tree T in G of minimum cost $c(T)$ which connects at least k terminals from S. We will consider the same density condition as for the Steiner Tree Problem and for the Prize Collecting Steiner Tree Problem. We will give a PTAS for the ϵ-Dense k-Steiner Tree Problem.

Group Steiner Tree Problem: Given a graph $G = (V, E)$ with edge costs $c\colon E \to \mathbb{R}_+$ and a system of pairwise disjoint subsets $C_1, \ldots, C_n$ of the vertex set V, find a minimum cost tree T in G such that for each $1 \leq i \leq n$ T contains at least one vertex of C_i. The sets C_i are also called classes, hence T has to contain at least one representative for each class. This problem is easily seen to be at least as hard as the Set Cover Problem, and Halperin and Krauthgamer gave a polylogarithmic lower bound for approximability [8], while Garg and Konjevod [5] obtain a polylog-approximation algorithm. Nevertheless for the ϵ-dense version of the problem we are able to give a polynomial time approximation scheme. An instance of the Group Steiner Tree

Problem is called ϵ-dense if it consists of a graph $G = (V, E)$ (i.e. all edge weights are 1) and groups $C_1, \ldots, C_n$ such that for every $s \in S := \bigcup_i C_i$, $|N_{V \setminus S}(s)| \geq \epsilon \cdot |V \setminus S|$.
The following tables summarize our results.

Steiner Problems: General Case vs. Dense Case

Problem Definition	General Case		ϵ-Dense
	Upper	Lower	Upper
Steiner Tree Problem	≈ 1.55 [13]	≈ 1.01 [3]	PTAS [11]
Group Steiner Tree	polylog. [5]	polylog. [8]	**PTAS**
k-Steiner Problem	$2 + \delta$ [1]	≈ 1.01 [3]	**PTAS**
Prize Collecting STP	$2 \cdot \left(1 + \frac{1}{\lvert S\rvert}\right)$ [6]	≈ 1.01 [3]	**PTAS**
Steiner Forest Problem	$2 \cdot \left(1 + \frac{1}{\lvert S\rvert}\right)$ [6]	≈ 1.01 [3]	**improved**

Steiner Tree Problem: Notions of Density

Problem	Density Condition	Result
in ϵ-Everywhere Dense Graphs	$\forall\, v \in V\ d_G(v) \geq \epsilon \cdot n$	**PTAS**
in ϵ-Average Dense Graphs	$\lvert E\rvert \geq \epsilon \cdot n^2$	**APX-hard**
(ϵ, c)-log Density	$\forall S' \subseteq S$, $\lvert S'\rvert \geq c \cdot \log(\lvert S\rvert)$ implies $\lvert E(S', V \setminus S)\rvert \geq \epsilon \cdot \lvert S'\rvert \cdot \lvert V \setminus S\rvert$	**PTAS**

The rest of the paper is organized as follows. In section 2 we recall two exact algorithms for the Steiner Tree Problem, the Spanning Tree Heuristic and the Dreyfus -Wagner algorithm. In section 3 we consider the ϵ-Dense Steiner Tree Problem and the relaxation to the log-density mentioned above. Section 4 deals with the Dense Steiner Forest Problem. In section 5, 6 and 7 we construc polytime approximation schemes for the ϵ-dense versions of the prize Collecting Steiner Tree, k-Steiner Tree and Group Steiner Tree Problem respectively. A few open problems are stated in section 8.

2 Exact Algorithms for the Steiner Tree Problem

We mention here only two exact algorithms for the Steiner Tree Problem, the Tree Enumeration Heuristic and the Dreyfus-Wagner Algorithm.

The Tree Enumeration Heuristic [7] enumerates all subsets of $V \setminus S$ of size at most $|S| - 2$ and for each of them tries to compute a minimum spanning tree (MST). The running time is $O(|S|^2 \cdot 2^{n-|S|} + n^3)$, hence exponential in the number of non-terminals but polynomial in the number of terminals.

The Dreyfus-Wagner Algorithm [4] is based on a dynamic programming approach and achieves a running time exponential in the number of terminals but polynomial in the number of vertices in the graph. For a given instance $G = (V, E), S \subseteq V, c\colon E \to \mathbb{R}_+$ of the Steiner Tree Problem in edge-weighted graphs, the Dreyfus-Wagner algorithm St-DW computes an optimum Steiner tree $T = \text{St-DW}(S, c)$ for terminal set S in G, based on the following recursion. Given a node $v \in V$, $\text{St-DW}_2(v, S, c)$ denotes an optimum solution to the problem of constructing a Steiner tree for terminal set $\{v\} \cup S$ such that v has degree ≥ 2 in this tree. Then

$$\begin{aligned}
\text{St-DW}_2(v, S, c) &= \text{argmin}_{\emptyset \subsetneq X \subsetneq S} \{c(\text{St-DW}(\{v\} \cup X, c)) \\
&\qquad\qquad + c(\text{St-DW}(\{v\} \cup (S \setminus X)))\} \\
\text{St-DW}(\{v\} \cup S, c) &= \text{argmin}\,\{c(\text{St-DW}_2(v, S, c)), \\
&\qquad\qquad \min_{u \in V \setminus S}\{c(v, u) + c(\text{St-DW}_2(u, S, c))\}, \\
&\qquad\qquad \min_{u \in S}\{c(v, u) + c(\text{St-DW}(S))\}\,\}
\end{aligned}$$

Lemma 2.1. *[4] The running time of the Dreyfus-Wagner algorithm is in* $O(3^{|S|}n + 2^{|S|}n^2 + n^3)$.

For the rest of the paper, by *THD* we denote the Tree Enumeration Heuristic and by *St-DW* the Dreyfus-Wagner Algorithm.

3 The ϵ-Dense Steiner Tree Problem

In this section we will construct an efficient polynomail time approximation scheme for the ϵ-Dense Steiner Tree Problem. Furthermore we consider a relaxation of the density condition "towards average-density" which we call log-density.

Let us start with the ϵ-Dense Steiner Tree Problem. It was stated and used in [11] that if an instance $G = (V, E)$, $S \subseteq V$ satisfies the ϵ-density condition, then there exists some $v \in V \setminus S$ such that $|N_S(v)| \geq \epsilon \cdot |S|$. Here $N_S(v)$ denotes the set of all neighbors of v that are elements of S. if we pick such a node v and remove all its terminal neighbors from S, the resulting instance $G, S \setminus N_S(v)$ is still ϵ-dense. Hence we can iterate these greedy picks and collect the stars consisting of such a node v and the set $N_S(v)$. After i such picks, the size of the terminal set is reduced to at most $(1-\epsilon)^i \cdot |S|$. Hence if afterwards we contract all the stars picked so far and add these contracted nodes to the terminal set, this results in an instance of the Steiner Tree Problem with a terminal set of size at most $(1-\epsilon)^i \cdot |S| + i$. Hence after $O(\log |S|)$ greedy steps we are left with an instance of the Steiner Tree Problem with only $O(\log(|S|))$ terminals, which can be solved to optimality using the Dreyfus Wagner Algorithm DW [4]. This gives the following modified version of the Karpinski-Zelikovsky algorithm.

Algorithm DST
Input: $G = (V, E)$, $S \subseteq V$, $\delta > 0$
Output: $(1+\delta)$-approx. Steiner Tree for S in G
Phase 1: Greedy Picks
  while $|S| > k_\delta$
    $v := \text{argmax}\{|N_S(u)|, u \in V \setminus S\}$
    $ST_v :=$ the star consisting ov v and $N_S(v)$
    $S := S \setminus N_S(v)$
Phase 2:
  for each star ST_v collected in Phase 1
    Contract ST_v into s_v
    $S := S \cup \{s_v\}$
  Solve the remaining instance using
  the Dreyfus-Wagner Algorithm St-DW, obtain tree T_2.
Return $T_{DST} := T_2 \cup \bigcup_{\text{Phase 1}} ST_v$

Analysis. Let S_i denote the terminal set after i greedy picks in phase 1. Due to the density condition, $|S_i| \leq (1-\epsilon)^i \cdot |S|$. Hence for $i \geq \log\left(\frac{|S|}{k_\delta}\right) / \log\left(\frac{1}{1-\epsilon}\right)$, after i greedy steps, $|S_i| \leq k_\delta$.

The remaining instance consists of at most $i + k_\delta$ terminals. For each star ST_v picked in phase 1, we add edges that form an MST for the terminals of ST_v. This does not increase the cost of an optimum Steiner tree. We may assume that the optimum Steiner tree for this modified instance consists of the MSTs for the stars and a tree T_0 connecting these. Note that the cost of T_2 is bounded by the cost of T_0. Let T^* denote an optimum Steiner tree for S in G. Let $ST_1, \ldots, ST_i$ be the stars cllected in phase 1, let S_j denote the terminal set of ST_j $(1 \leq j \leq i)$. Hence we obtain

$$\frac{c(T_{DST})}{c(T^*)} \leq \frac{\sum_{j=1}^{i} |S_j| + i + k_\delta}{\sum_{j=1}^{i} |S_j| + k_\delta}$$

We want to choose k_δ such as to bound the right hand side of this inequality by $1+\delta$. It suffices to choose $k_\delta \geq \frac{|S|}{2^{\log(1/(1-\epsilon)) \cdot |S| \cdot \delta - 1}}$, which can be bounded by a constant only depending on ϵ and δ. The running time of phase 1 is obviously polynomial in the size of G and S. Since $i = O(\log(|S|))$ for fixed δ, the running time of phase 2 is also polynomial in the input size. Hence we obtain

Theorem 3.1. *Algorithm DST is a polynomial time approximation scheme for the ϵ-Dense Steiner Tree Problem.*

Since each terminal was involved in at most one contraction, afterwards each of the new terminals has at least $\epsilon|V \setminus S| - 1$ non-terminal neighbors and the number of non-terminals is $|V \setminus S| - i$. Hence the resulting instance is ϵ'-dense for $\epsilon' = \frac{\epsilon \cdot |V \setminus S| - i}{|V \setminus S| - i}$. We can now iterate the greedy process $O(\log^*(|S|)$ times, where the j-th run of the greedy phase takes as its terminal set the remaining terminals and the contracted stars from the previous phase. This reduces the number of terminals to a constant, and therefore we can afterwards use the Tree Enumeration Algorithm [7] instead of the Dreyfus-Wagner algorithm to solve the remaining instance. This yields

Theorem 3.2. *For each $\epsilon > 0$, there is an efficient polynomial time approximation scheme for the ϵ-dense Steiner Tree Problem.*

3.1 Towards Average-Density: Relaxing the Density Condition

We consider now instances of the Graph Steiner Tree Problem for which some kind of average density condition is satisfied. A natural definition of average density is as follows: We call an instance $G = (V, E), S \subseteq V$ of the Graph Steiner Tree Problem *average ϵ-dense* if

$$|E(S, V \setminus S)| \quad \geq \quad \epsilon \cdot |S| \cdot |V \setminus S|. \tag{1}$$

Note that ϵ-Density as defined in [11] implies Average-ϵ-Density. Furthermore average ϵ-density implies existence of a vertex $v \in V \setminus S$ with at least $\epsilon \cdot |S|$ neighbours in S, i.e. a *good pick*. However average ϵ-density is not preserved by good picks. Here is an easy example: If there exists a subset S' of S of size $\epsilon \cdot |S|$ such that every vertex $v \in V \setminus S$ is connected to all terminals from S' and to none of the terminals from $S \setminus S'$, this instance is average ϵ-dense. However after one good pick the density condition is not valid anymore.

Nevertheless we will now relax the density condition "towards average-density" and give a PTAS for this relaxed version. Let us first give some motivation. We observe that ϵ-density does not only imply (1) but also the following more general property:

$$|E(S', V \setminus S)| \quad \geq \quad \epsilon \cdot |S'| \cdot |V \setminus S| \quad \text{for all } S' \subseteq S \tag{2}$$

Indeed (2) is equivalent to ϵ-density. We may now ask how far we can relax (2). Relaxing here means to consider instances where (2) does not necessarily hold for all subsets S' of S but for all subsets with cardinality at least some prespecified lower bound. In (2) the lower bound is 1 (or 0) while in the average-density condition (1) it is $|S|$. The question now is: How much can we increase the lower bound (starting from 1) and still get a PTAS ? Actually we do not know the answer but at least we can relax up to logarithmic size:

Definition 3.1. (log-**Density**)
An instance $G = (V, E), S$ of the Graph Steiner Tree Prolem is called $(\epsilon, c)-$ log-**dense** *iff for all subsets $S' \subseteq S$ of the terminal set with $|S'| \geq c \cdot \log(|S|)$*

$$|E(S', V \setminus S)| \quad \geq \quad \epsilon \cdot |S'| \cdot |V \setminus S|. \tag{3}$$

Theorem 3.3. *For each $\epsilon > 0, c > 0$ there is a PTAS for the $(\epsilon, c) -$* log-*Dense Steiner Tree Problem.*

Let us first give some ideas and then give the precise proof of Theorem 3.3. Our approach is quite similar to that of Karpinski and Zelikovsky [11]. The most important difference is that when performing greedy steps and picking Steiner points, after contracting a star consisting of a vertex from $V \setminus S$ and all its neighbours in S the resulting supernode will be removed from S and hence not be considered in further greedy steps anymore. This alternative method has basically two effects: First the density condition for the actual terminal set is preserved and second afterwards we are left with a "residual" terminal set of logarithmic instead of constant size. Hence in order to solve the remaining problem we will take the Dreyfus Wagner algorithm instead of the Tree Enumeration algorithm since its running time is polynomial in the number of non-terminals (and exponential in the number of terminals, hence polynomial in the initial input size). We are now ready to describe our algorithm.

Algorithm LDSTP
Input: an instance of the (c, ϵ)-log-dense Steiner Tree Problem consisting of graph $G = (V, E)$, terminal set $S \subseteq V$
Output: Steiner tree T for S in G

(0) $\mathcal{C} := \{\{s\} : s \in S\}$ set of terminal components
$\mathcal{C}_a := \mathcal{C}$ set of active terminal components
(1) **while** $E(\mathcal{C}_a) \neq \emptyset$ **do**
Pick $e \in E(\mathcal{C}_a)$ connecting two terminal components $C_1, C_2 \in \mathcal{C}_a$.
Let $\mathcal{C}_a := (\mathcal{C}_a \setminus \{C_1, C_2\}) \cup \{C_1 \cup C_2\}$, update $\mathcal{C}$ accordingly.
(2) **while** $|\mathcal{C}_a| \geq c \cdot \log(|S|) \cdot K$ **do**
Find $v \in V \setminus S$ with the maximum number of neighbours in $\mathcal{C}_a$.
Contract the star $T(v)$ consisting of v and its neighbours $N(v, \mathcal{C}_a)$ in $\mathcal{C}_a$. Update $\mathcal{C}$ and $\mathcal{C}_a$ accordingly:
$\mathcal{C}_a := \mathcal{C}_a \setminus N(v, \mathcal{C}_a)$, $\mathcal{C} := (\mathcal{C} \setminus N(v, \mathcal{C}_a)) \cup \{\bigcup_{C \in N(v, \mathcal{C}_a)} C\}$.
(3) Find an optimum Steiner tree $T^\star$ for $\mathcal{C}$.
(4) **Return** $T_{LD} := T^\star \cup \bigcup_{v \text{ picked in } (1)} T(v) \cup \{e | \, e \text{ picked in } (1)\}$.

Analysis. The log-density condition (3) directly implies that initially

$$|E(\mathcal{C}', V \setminus \mathcal{C}_a)| \quad \geq \quad \epsilon \cdot |\mathcal{C}'| \cdot |V \setminus \mathcal{C}_a| \tag{4}$$

for all subsets $\mathcal{C}'$ of $\mathcal{C}_a$ of size at least $c \cdot \log(|S|)$. We will now prove that (4) is preserved by the picks of edges in phase (1) and stars in phase (2) of algorithm LDSTP. Indeed, if an edge connecting two active components is picked, then subsets $\mathcal{C}'$ of $\mathcal{C}_a$ of size at least $c \cdot \log(|S|)$ after the pick correspond to subsets of $\mathcal{C}_a$ of size $\in [\, |\mathcal{C}'|, |\mathcal{C}'| + 1]$ before the pick with the same neighbourhood in $V \setminus \mathcal{C}$, and since $|V \setminus \mathcal{C}|$ does not change, (4) still holds. On the other hand, if a vertex $v \in V \setminus \mathcal{C}$ is picked and the star consisting of v and $N(v, \mathcal{C}_a)$ is contracted, then the resulting supernode is removed from $\mathcal{C}_a$ and the cardinality of $V \setminus \mathcal{C}_a$ remains the same, hence also in this case (4) is preserved.

Let k be the number of picks of stars $T(v, N(v, \mathcal{C}_a))$ in phase (2), let $T_1 = T(v_1, N_1), \ldots, T_k = T(v_k, N_k)$ denote these stars and let $e_1, \ldots, e_l$ denote the single edges connecting active components picked in phase (1) of algorithm LDSTP.

Now construct graph G' from G by adding edges connecting the set $N(v, \mathcal{C}_a)$ by a spanning tree for each pick v in phase (1) of algorithm LDSTP. Note that $OPT(G', S) \leq OPT(G, S)$. There exists an optimum tree $T^{\star\star}$ in G' consisting of spanning trees T_i' for the sets $N_i = N(v_i, \mathcal{C}_a)$ of picks in phase (2), the set E_1 of all edges connecting two active components in phase (1) and a tree T' connecting the set of components $\mathcal{C}$ at the end of phase (2). Hence we can bound the approximation ratio of algorithm LDSTP as follows:
$$\frac{\text{cost}(T_{LD})}{\text{cost}(T^{\star\star})} \leq \frac{\text{cost}(T^\star) + \sum_{i=1}^k \text{cost}(T_i) + |E_1|}{\text{cost}(T') + \sum_{i=1}^k \text{cost}(T_i') + |E_1|} \leq \frac{\sum_{i=1}^k |N_i|}{\sum_{i=1}^k (|N_i| - 1)} \leq 1 + \frac{k}{(\sum_{i=1}^k |N_i|) - k}.$$

Hence let us assume we start with a (c,ϵ)-dense instance with no edges between teminals. Each pick of a star reduces the cardinality of $\mathcal{C}_a$ by a factor ϵ. Let $\mathcal{C}_a(i)$ denote the set $\mathcal{C}_a$ after i picks of a star, then $|\mathcal{C}_a(i)| \leq (1-\epsilon)^i|S|$. We obtain $|\mathcal{C}_a(k)| < c \cdot \log(|S|) \cdot K$ for $k \geq \frac{|S|}{c\cdot\log(|S|)\cdot K} \cdot \frac{1}{\log(1/(1-\epsilon))}$, hence we assume $k \leq \frac{|S|}{c\cdot\log(|S|)\cdot K} \cdot \frac{1}{\log(1/(1-\epsilon))} + 1$. Since

$$\sum_{i=1}^{k} |N_i| \quad \geq \quad \sum_{i=1}^{k}(1-\epsilon)^i \cdot \epsilon \cdot |S| = \epsilon \cdot |S| \cdot \sum_{i=1}^{k}(1-\epsilon)^i = \epsilon \cdot |S| \cdot \frac{1-(1-\epsilon)^k}{\epsilon}$$

we obtain the following bound for the approximation ratio of algorithm LDSTP:

$$\begin{aligned}
\frac{\mathrm{cost}(T_{LD})}{\mathrm{cost}(T^{**})} &\leq 1 + \frac{2k}{\sum_{i=1}^{k}|N_i|} \leq 1 + \frac{2k}{|S|\cdot(1-(1-\epsilon)^k)} \\
&\leq 1 + \frac{2\cdot(\frac{|S|}{c\cdot\log(|S|)\cdot K}\cdot\frac{1}{\log(1/(1-\epsilon))}+1)}{|S|\cdot(1-(1-\epsilon)^k)} \\
&\leq 1 + \frac{2\cdot(\frac{|S|}{c\cdot\log(|S|)\cdot K}\cdot\frac{1}{\log(1/(1-\epsilon))}+1)}{|S|\cdot\epsilon} \qquad (5)
\end{aligned}$$

since we may assume $k \geq 1$ (in case $k = 0$ algorithm LDSTP computes an optimum solution, namely a spanning tree for S). Using $\frac{1}{|S|} \leq \frac{1}{\log(|S|)}$ we obtain $\frac{\mathrm{cost}(T_{LD})}{\mathrm{cost}(T^{**})} \leq 1 + \frac{1}{\epsilon} \cdot \left(\frac{1}{c\cdot K\cdot\log(\frac{1}{1-\epsilon})} + 1\right) \cdot \frac{1}{\log(|S|)}$. For given $\delta > 0$ we will now choose K such that the approximation ratio is bounded by $1+\delta$, i.e. $K \geq \left(c \cdot \log\left(\frac{1}{1-\epsilon}\right) \cdot (\epsilon \cdot \delta \cdot \log(|S|) - 1)\right)^{-1}$. Hence choosing $K = \left(c \cdot \log\left(\frac{1}{1-\epsilon}\right)\right)^{-1}$, solving the Steiner Tree instance exactly by brute force for $|S| \leq 2^{2/(\epsilon\cdot\delta)}$ and applying LDSTP for all other instances yields a PTAS for the (c,ϵ)-log-dense Steiner Tree Problem. □

4 The Dense Steiner Forest Problem

In this section we will consider the ϵ-Dense Steiner Forest Problem. Currently we are not able to provide a PTAS for this problem, for the following reason: All the variants of the methods of [11] we have discussed so far (and will discuss in subsequent sections) are based on the approach of performing greedy steps until the problem size is sufficiently small and then applying some exact algorithm for the remaining instance. In the Steiner Forest Case the kind of greedy steps we have in mind reduce each single terminal set to constant size, but the number of terminal sets might not be reduced at all. On the other hand we do not know how to justify contraction steps that reduce the number of terminal sets, since melting j of them into a single terminal set might produce an additive cost of j. However we will now give an approximation algorithm for the Dense Steiner Forest Problem with approximation ratio $1 + O((\sum_{i=1}^{n} \log(|S_i|))/(\sum_{i=1}^{n} |S_i|))$, where $S_1, \ldots S_n$ are the given terminal sets. Intuitively this provides good approximation in case sufficiently many terminal sets are large, and we will make this precise in this section.

Definition 4.1. *An instance $G = (V, E), S_1, \ldots, S_n$ of the SFP is called* **ϵ-dense'** *iff for all $1 \leq i \leq n$ and $S' \subseteq S_i$ there exists a vertex $v \in V \setminus S_i$ such that $|N(v) \cap S'| \geq \epsilon \cdot |S'|$.*

Lemma 4.1. *For every $\epsilon > 0$, every ϵ-dense instance of the SFP is ϵ-dense'.*

Proof: Let $G = (V, E), S_1, \ldots, S_n$ be ϵ-dense, let $i \in \{1, \ldots, n\}$ and $S' \subseteq S_i$. Then for all $s \in S'$ it holds $|N(s) \cap (V \setminus S_i)| \geq \epsilon \cdot |V \setminus S_i|$.
From $\sum_{v \in V \setminus S_i} |N(v) \cap S'| = \sum_{s \in S'} |N(s) \cap (V \setminus S_i)| \geq |S'| \cdot \epsilon \cdot |V \setminus S_i|$ we conclude that there exists at least on $v \in V \setminus S_i$ such that $|N(v) \cap S'| \geq \epsilon \cdot |S'|$.
$\square$

Algorithm A_k:
Input: $G = (V, E), \mathcal{S} := \{S_1, \ldots, S_n\} \subseteq P(V)$ instance of the ϵ-Dense SFP
Output: Set of edges $F \subseteq E$ defining a Steiner Forest for $S_1, \ldots, S_n$

(0) Let $F := \emptyset$ and $S_{i,act} := S_i, 1 \leq i \leq n$.
(1) **while** $\max_{1 \leq i \leq n} |S_{i,act}| \geq k$ **do**
  Pick $i \in \{1, \ldots, n\}$ and $v \in V \setminus S_{i,act}$
    such as to maximize $|N(v) \cap S_{i,act}|$.
  Let $\tilde{S} := N(v) \cap S_{i,act}$ and $F := F \cup \{\{v, s\} : s \in \tilde{S}\}$.
  $S_{i,act} := S_{i,act} \setminus \tilde{S}$. Contract $\tilde{S} \cup \{v\}$.
(2) Solve the remaining instance using the Primal-Dual algorithm.

Lemma 4.2. *At the beginning of every call of the while-loop the sets $S_{i,act}$ are ϵ-dense'.*

Proof: The initial sets $S_{i,act} = S_i$ are ϵ-dense and therefore ϵ-dense'. Since in every iteration the removed set $\tilde{S}$ does not contain elements from $\bigcup_{s \in S_{i,act} \setminus \tilde{S}} N(s) \setminus S_{i,act}$, for every subset S' of $S_{i,act} \setminus \tilde{S}$ existence of a vertex v

in $V \setminus S_{i,act}$ and hence in $V \setminus S'$ with many neighbours in S' is not disturbed.$\square$

Analysis of Algorithm A_k. First note that $\sum_{i=1}^n (|S_i| - 1) =: L$ is a lower bound for the cost of an optimum solution. We will now estimate the cost of the solution produced by algorithm A_k. For $1 \leq i \leq n$ let $j(i)$ denote the number of contractions of subsets of $S_{i,act}$ in phase (1) of the algorithm, and let $S_i^1, \ldots, S_i^{j(i)}$ be the subsets being contracted. Let $S_{i,rem} := S_i \setminus (S_i^1 \cup \ldots \cup S_i^{j(i)})$ the remaining set of S_i. Then the number of edges added to F in phase (1) is given by $\text{cost}_1 = \sum_{i=1}^n \sum_{l=1}^{j(i)} |S_i^l| = \sum_{i=1}^n (|S_i| - |S_{i,rem}|)$. At the end of phase (1), for $1 \leq i \leq n$ the size of S_i is given by $s(i) := j(i) + |S_i| - \sum_{l=1}^{j(i)} |S_i^l| = j(i) + |S_{i,rem}|$. Furthermore the size of $S_{i,act}$ after l contractions is bounded by $|S_i|(1-\epsilon)^l$, hence $s(i) \leq j(i) + |S_i|(1-\epsilon)^{j(i)}$. Hence if we let cost_2 denote the number of edges picked by the 2-Approximation Algorithm in phase (2) of A_k, then cost_2 is bounded as follows: $\text{cost}_2 \leq 2 \cdot \sum_{i=1}^n \left(j(i) + |S_i|(1-\epsilon)^{j(i)}\right)$. Let $x_{i,l} := |S_i^l|, 1 \leq i \leq n, 1 \leq l \leq j(i)$. An upper bound for the cost of solution generated by algorithm A_k is then given by the following optimization problem:

$$
\begin{aligned}
& \max \sum_{i=1}^{n} \sum_{l=1}^{j(i)} x_{i,j} \; + \; 2 \cdot \left(|S_i| - \sum_{l=1}^{j(i)} x_{i,l} \; + j(i) \right) \\
& \text{s.t.} \quad \sum_{l=1}^{j(i)} x_{i,l} \geq |S_i| \cdot \left(1 - (1-\epsilon)^{j(i)}\right), \; 1 \leq i \leq n \\
= \quad & \max \sum_{i=1}^{n} \left(2|S_i| - \sum_{l=1}^{j(i)} x_{i,l} \; + 2j(i) \right) \\
& \text{s.t.} \quad \sum_{l=1}^{j(i)} x_{i,l} \geq |S_i| \cdot \left(1 - (1-\epsilon)^{j(i)}\right), \; 1 \leq i \leq n
\end{aligned}
$$

Let $\text{save}_i := \sum_{l=1}^{j(i)} x_{i,l} - 2j(i)$, $1 \leq i \leq n$. We give a lower bound for save_i as a function of $|S_i|$. Let $X = \sum_{l=1}^{j(i)} x_{i,l}$ the total number of terminals removed from S_i in phase (1) of algorithm A_k, then $X > (\epsilon|S_i| - k)/\epsilon = |S_i| - k/\epsilon$. Furthermore $k > \epsilon \cdot |S_i| \cdot (1-\epsilon)^{j(i)}$ from which we conclude $j(i) < \log\left(\frac{k}{\epsilon|S_i|}\right) / \log(1-\epsilon)$. Hence the total cost of the solution generated by algorithm A_k is bounded by
$\sum_{i=1}^n \left(|S_i| + \frac{k}{\epsilon} + 2 \cdot \frac{\log\left(\frac{k}{\epsilon|S_i|}\right)}{\log(1-\epsilon)} \right) = \sum_{i=1}^n \left(|S_i| + \frac{k}{\epsilon} + 2 \cdot \frac{\log\left(\frac{\epsilon|S_i|}{k}\right)}{\log\left(\frac{1}{1-\epsilon}\right)} \right) =: tc(\epsilon, k)$.
Since $\frac{d}{dk} tc(\epsilon, k) = n \cdot \left(\frac{1}{\epsilon} - \frac{2}{\log\left(\frac{1}{1-\epsilon}\right)} \cdot \frac{1}{k} \right) = 0$ for $k^* = \frac{2\epsilon}{\log\left(\frac{1}{1-\epsilon}\right)}$ and

$\frac{d^2}{dk^2}tc(\epsilon, k^*) > 0$, we choose $k = k^*$ and finally obtain

Theorem 4.1. *For each $\epsilon > 0$ there is a polynomial time approximation algorithm for the ϵ-Dense Steiner Forest Problem with approximation ratio $1 + O\left(\frac{\sum_{i=1}^{n} \log(|S_i|)}{\sum_{i=1}^{n} |S_i|}\right)$.*

5 Dense Prize Collecting Steiner Tree Problem

In this section we describe a polynomial time approximation scheme for the ϵ-Dense Prize Collecing Steiner Tree Problem.

Let $G = (V, E)$, $S \subseteq V$, $p\colon S \to \mathbb{R}_+$ be an instance of the ϵ-Dense Prize Collecing Steiner Tree Problem. We will construct a solution consisting of a tree T in G and a terminal set $S' \subseteq S$ such that T is a Steiner tree for the terminal set $S \setminus S'$. First we observe that if for some $s \in S$, $p(s) \leq 1$ then we may put s in S', i.e. pay the prize for terminal s, since otherwise we would have to spend at least one edge in order to connect s to the rest of the tree. Hence we may now assume that $S = \{s_1, \ldots, s_n\}$ such that $1 < p(s_1) \leq \ldots \leq p(s_n)$.

In a greedy phase we collect stars in the same way as for the ϵ-Dense Steiner Tree Problem. This will reduce the number of terminals from $|S|$ to $O_\epsilon(\log(|S|))$. Note that collected stars ST_i for terminal sets S_i will be contracted to super-terminals σ_i with prize $p(\sigma_i) := \sum_{s \in S_i} p(s)$.

In the second phase we run an exact algorithm for the Prize-Collecting Steiner Tree Problem in order to solve the residual instance to optimality. This algorithm is an extension of the Dreyfus-Wagner algorithm to the prize-collecting phase.

The rest of the section is oranized as follows: First we describe an exact algorithm P-DW for the Prize-Collecting Steiner Tree Problem. Then we give a description and analysis of the ptas for the ϵ-Dense Prize-Collecting Steiner Tree Problem.

An Exact Algorithm. Algorithm P-DW is based on a dynamic-programming approach and extends the Dreyfus-Wagner algorithm to the prize-collecting case. The algorithm is based on the following recursion formula.

$$\text{P-DW}(S, p, c) = \min_{\emptyset \subseteq S_p \subseteq S} (p(S_p) + \text{St-DW}(S \setminus S_p, c))$$

Instead of separate calls of the Dreyfus-Wagner algorithm St-DW for each of the instances $S \setminus S_p$, $S_p \subseteq S$ one can implement this recursion by building just one single dynamic-programming table. This gives the following result.

Lemma 5.1. *The algorithm P-DW can be implemented to run in time* $O(6^{|S|} \cdot n^2 + n^3)$.

PTAS for the ϵ-Dense Case. We are now ready to give a detailed pseudo-code description of the ptas for the ϵ-Dense Prize Collecting Steiner Tree Problem.

Algorithm $\mathcal{A}_{Prize}$
Input: $G = (V, E), S \subseteq V, p\colon S \to \mathbb{Q}_+,\ \delta > 0$
Output: tree T connecting a subset S_T of S in G
STARS := $\emptyset$ /⋆ *List of collected stars* ⋆/
$S_{act} := S \setminus \{s \in S \mid p(s) \leq 1\}$
(if $p(s) \leq 1$ then we will pay this prize)
While $|S_{act}| \geq k_\delta$
  $v := \text{argmax}_{u \in V \setminus S} |N_{S_{act}}(u)|$
  $ST_v :=$ star consisting of v and $N_{S_{act}}(v)$
  STARS := STARS$\cup\{ST_v\}$, $S_{act} := S_{act} \setminus N_{S_{act}}(v)$
Solve Residual Problem:
  Assume STARS$= \{ST_1, \ldots, ST_r\}$
  Contract each $ST_i = ST_{v_i}$ into single node σ_i
  Define $p(\sigma_i) = \sum_{s \in N_S(v_i)} p(s)$ $(1 \leq i \leq r)$
  $T_0 :=$ P-DW$(S_{act} \cup \{\sigma_1, \ldots, \sigma_r\}, p, c)$
  (where each edge e has cost $c(e) = 1$)
Return $T := T_0 \cup_{i=1}^{r} ST_i$

Lemma 5.2. *The algorithm $\mathcal{A}_{Prize}$ is a polynomial time approximation scheme for the ϵ-Dense Price Collecting Steiner Tree Problem.*

Proof: If G was the given graph, then obtain graph G' from it by adding for each star $ST_v \in$ STARS edges building an MST for the set $S_v := N_{S_{act}}(v)$. Obviously, $OPT_{G'}(S, p) \leq OPT_G(S, p)$. We may assume that an optimum solution $T^*_{G'}$ for the instance G', S, p has the following property: For each star ST_v being picked in the greedy phase, either $T^*_{G'}$ contains all terminals from $S_v := N_{S_{act}}(v)$ or none of them, since if it contains a proper nonempty subset from S_v, then each of the remaining terminals $s \in S_v \setminus T^*_{G'}$ from S_v would add cost 1 to the connection cost of the solution tree but save a prize $p(s) > 1$. Let *cost* denote the connection cost (number of edges) of a tree plus the sum of prizes of all terminals not being in that tree. Now $T^*_{G'}$ can be assumed to consist of some of the spanning trees $M_i, i \in J$ for terminal sets S_v of stars ST_v and a tree T^*_0 connecting these spanning trees. Hence we obtain the following bound for the approximation ratio:

$$\begin{aligned}\frac{cost(T)}{OPT_G(S,p)} &\le \frac{cost(T)}{OPT_{G'}(S,p)} \le \frac{c(T_0^*)+\sum_{i\in J}|S_{v_j}|}{c(T_0^*)+\sum_{i\in J}(|S_{v_j}|-1)} \\ &\le \frac{c(T_0^*)+\sum_{i\in J}|S_{v_j}|}{c(T_0^*)+\sum_{i\in J}|S_{v_j}|-r} \le \frac{\sum_{i\in J}|S_{v_j}|}{\sum_{i\in J}|S_{v_j}|-r}\end{aligned}$$

As in the Steiner Tree case, we can choose k_δ depending solely on ϵ and δ such that this term is bounded by $1+\delta$. □

6 The Dense k-Steiner Tree Problem

Another well known generalization of the Steiner Tree Problem is the *k-Steiner Tree Problem*: Given an instance of the Steiner Tree Problem and a number k, one has to construct a minimum cost tree connecting at least k elements from the terminal set. Note that k does not have to be constant. Here we consider the ϵ-dense version of this problem:

ϵ-Dense k-Steiner Tree Problem
Instance: Graph $G = (V, E)$ with terminal set $S \subseteq V$ such that each terminal has at least $\epsilon \cdot |V \setminus S|$ neighbours in $V \setminus S$, a number $k \in \{1, \ldots, |S|\}$.
Solution: Tree T in G connecting at least k terminals from S
Cost: Minimize the number of edges in T.

Lemma 6.1. *There is a polynomial time approximation scheme for the ϵ-Dense k-Steiner Tree Problem.*

Proof: If $k \le \epsilon \cdot |S|$ then for $k \le c_\delta$ we enumerate all subsets S' of S of size k and for each of them compute an optimum Steiner tree using the Dreyfus-Wagner algorithm D-St. Among these trees, one of minimum cost is an optimum solution for the k-Steiner Tree Problem.

If $c_\delta < k < \epsilon \cdot |S|$ then a single greedy pick gives a tree T centered at some non-terminal v which contains at least $\epsilon \cdot |S|$ terminals. Take a subtree of this star connecting k terminals, the cost is $k+1 \le (1+\delta) \cdot k \le (1+\delta) \cdot OPT$ provided $k \ge \delta^{-1}$, hence it suffices to choose $c_\delta = \delta^{-1}$.

If $k > \epsilon \cdot |S|$ then in a greedy phase we collect stars until the number of remaining terminals drops below some constant C_δ. Solving the residual problem reduces to the following generalizationn of the k-Steiner Forest Problem where terminals $s \in S$ have *values* $g_s \ge 1$ and the task is to compute a minimum cost Steiner tree for a subset of terminals of total value at least k. We call this the *Terminal-Weighted k-Steiner Tree Problem*.

An exact algorithm TW-k-DW for the *Terminal-Weighted k-Steiner Tree Problem* just enumerates all terminal subsets, and for each of them of total value at least k it computes an optimum Steiner tree using the Dreyfus-Wagner algorithm St-DW for the Steiner Tree Problem. Its running time is linear-exponential in the number of terminals but polynomial in the number of non-terminals of the instance. $\square$

7 The Dense Group Steiner Tree Problem

An instance of the Group Steiner Tree Problem is called ϵ-dense if it consists of a graph $G = (V, E)$ and groups $S_1, \ldots, S_n \subseteq V$ such that for each $s \in S := \bigcup_{i=1}^n S_i$,

$$|N(s) \setminus S| \geq \epsilon \cdot |V \setminus S|.$$

A feasible solution is a subtree T of G containing at least one node from every group S_i. In this case, the cost of the tree T is defined as $c(T) :=$ number of edges of T.

Let us call a group S_i *neighbor* of a vertex v if $S_i \cap N(v)$ is not empty. The idea of our algorithm is as follows: The density condition directly implies existence of a vertex $v \in V \setminus S$ which has many groups in its neighbourhood. Now in a greedy phase we perform the same kind of picks as in the Dense Steiner Forest algorithm (section 4): We maintain a set of *active classes*, starting with every group S_i being active. We pick a vertex v with maximum number of groups as neighbours. For each of these groups S_i neighbored to v we pick an element $s_i \in S_i \cap N(v)$. We contract these vertices, declare the associated groups as inactive and iterate. This guarantees that in every iteration we find a vertex v with many active groups in its neighbourhood. Afterwards we are left with a constant number of active groups S_i (i.e. groups not being involved in any contraction so far) and a logarithmic number of supervertices $\int_i$ resulting from contractions. The task is to choose representatives s_i of the active groups S_i and to construct a Steiner Tree for the terminal set consisting of the choosen representatives s_i and the supervertices $\int_j$. Note that there are only $\prod_{i:S_i \text{ active}} |S_i| = |V|^{O(1)}$ many different choices of representatives, and for each choice we are left with an instance of the Steiner Tree Problem with logarithmic number of terminals, which can be solved optimally using the primal-dual approach (cf. the remark in section 4). Since we have picked only logarithmic (in the number of classes) many Steiner points and the trivial lower bound is number of groups minus 1, this algorithm yields a PTAS for the ϵ-Dense Group Steiner

Tree Problem. We are now ready to give a detailed description and analysis of our algorithm.

Algorithm $\mathcal{A}_{DGL}$
Input: Graph $G = (V, E)$, groups $S_1, \ldots, S_n$, $\delta > 0$
Output: $(1+\delta)$-approximate Group Steiner Tree T
Initialization:
$\mathcal{C}_a := \{S_1, \ldots, S_n\}$ (set of active groups)
Choose $k := \max\left\{\frac{2}{\epsilon}, \frac{2\cdot(1+\delta)}{\delta\cdot\log(1/(1-\epsilon))}\right\}$
Choose $k_0 := k_0(\delta) := \frac{2k}{1-\epsilon}$
If $n \leq k_0$ **then**
For each system of representatives $s_i \in S_i, 1 \leq i \leq n$
Compute an optimum Steiner Tree for the instance
consisting of graph G and terminal set $\{s_1, \ldots, s_n\}$.
Return the best upon those Steiner Trees.
Phase 1:
while $|\mathcal{C}_a| \geq k$ **do**
Pick $v \in V \setminus S$ maximizing the cardinality of
$N(v, \mathcal{C}_a) = \{S_i \in \mathcal{C}_a : S_i \text{ is neighbour of } v\}$.
(Store and contract.)
Let $\mathcal{S}$ be the set of all supervertices constructed in phase 1.
Phase 2:
For each choice of representatives $s_i \in S_i$ $(S_i \in \mathcal{C}_a)$
Compute an optimum Steiner Tree T for the terminal set
$\{s_i : S_i \in \mathcal{C}_a\} \cup \mathcal{S}$.
$T_o :=$ a min-cost tree upon those constructed in the for-loop.
Return T_o.

Analysis. The density condition implies that every contraction in phase 1 of the algorithm reduces the set of active groups by an ϵ-fraction. Let $\mathcal{C}_a(i)$ be the set of active groups after i contractions and t be the total number of contractions in phase 1. Then $|\mathcal{C}_a(i)| \leq (1-\epsilon)^i \cdot n$, hence we have $|\mathcal{C}_a(i)| \leq k$ for $i \geq \log(n/k)/\log(1/(1-\epsilon))$, hence

$$t \leq 2 \cdot \log\left(\frac{n}{k}\right) / \log\left(\frac{1}{1-\epsilon}\right) =: t_u \quad \text{(upper bound for } t) \tag{6}$$

We assume $n \geq k_0 \geq 2k \cdot \frac{1}{1-\epsilon}$, this implies $\frac{2\log(n/k)}{\log(1/(1-\epsilon))} \geq \frac{\log(n/k)}{\log(1/(1-\epsilon))-1}$.

Since $|\mathcal{S}| = t = O(\log(n))$ and $|\mathcal{C}_a| \leq k = O(1)$ at the end of phase 1, there are only $\prod_{S_i\in\mathcal{C}_a} |S_i| = O\left(|V|^k\right)$ many choices of systems of representatives for the groups $S_i \in \mathcal{C}_a$. Hence each terminal set considered in the for-loop in phase 2 has size bounded by $k+t = O(\log(n))$, and the according Steiner Tree can be computed in polynomial time using the Dreyfus-Wagner algorithm. This establishes polynomial running time of the algorithm for fixed k.

We will now give a bound on the approximation ratio of the tree T_o computed by algorithm $\mathcal{A}_{DCL}$. Let $T^\star$ denote an optimum Group Steiner Tree for $S_1 \ldots, S_n$ in G. Note that T_o consist of a set of stars $ST_1, \ldots, ST_t$ generated by contractions in phase 1 and a set of edges T_0 connecting these stars and certain representatives $s_o(j)$ for the remaining groups S_j. Now construct graph G' as follows: Start with graph G and add the following edges:

(1) for each star ST_i edges forming a spanning tree for the terminals of ST_i,

(2) for each inactive class S_l (i.e. class being involved in a contraction in phase 1) edges from the terminal $s_l \in S_l$ which was picked to all neighbours of elements of class S_l in G.

Note that $\mathrm{OPT}(G') \leq \mathrm{OPT}(G)$. An optimum group Steiner tree $T^\star_{G'}$ for groups $S_i, 1 \leq i \leq n$ in graph G' can be obtained as a set of spanning trees T'_i for the sets of terminals R_i of stars ST_i, a set of representatives $\int_j$ for the still active groups S_j and a Steiner tree T' connecting these. Note that the cost of T_0 is bounded by the cost of T'. Hence we obtain

$$\begin{aligned}
\frac{\mathrm{cost}(T_o)}{\mathrm{cost}(T^\star)} &\leq \frac{\mathrm{cost}(T_o)}{\mathrm{cost}(T^\star_{G'})} = \frac{\mathrm{cost}(T_0) + \sum_{i=1}^t \mathrm{cost}(ST_i)}{\mathrm{cost}(T') + \sum_{i=1}^t \mathrm{cost}(T'_i)} \leq \frac{\sum_{i=1}^t \mathrm{cost}(ST_i)}{\sum_{i=1}^t \mathrm{cost}(T'_i)} \\
&= \frac{\sum_{i=1}^t |R_i|}{\sum_{i=1}^t (|R_i| - 1)} = \frac{\sum_{i=1}^t |R_i|}{\left(\sum_{i=1}^t |R_i|\right) - t} = \frac{n - |\mathcal{C}_a|}{n - |\mathcal{C}_a| - t} \leq \frac{n-k}{n-k-t}
\end{aligned}$$

We know that for $t \geq t_u$ $|\mathcal{C}_a| \leq k$ at the end of phase 1 of the algorithm. Hence we can upperbound $\frac{n-k}{n-k-t}$ by $\frac{n-k}{n-k-t_u}$. We want to choose k and k_0 such that (a) the upper bound estimate t_u in (6) is valid and (b) the approximation ratio is at most $1+\delta$:

$$\frac{n-k}{n-k-t_u} \leq 1+\delta \iff 0 \leq \delta(n-k) - (1+\delta) \cdot \frac{2 \cdot \log(n/k)}{\log(1/(1-\epsilon))}$$

which is equivalent to $\delta k - \frac{2\cdot(1+\delta)}{\log(1/(1-\epsilon))} \cdot \log(k) \leq \delta n - \frac{2\cdot(1+\delta)}{\log(1/(1-\epsilon))} \cdot \log(n)$ for $n > k_0$. Since the function $f(x) := \delta \cdot x - \frac{2\cdot(1+\delta)}{\log(1/(1-\epsilon))} \cdot \log(x)$ is unbounded and monotone increasing for $x > \frac{2\cdot(1+\delta)}{\delta\cdot\log(1/(1-\epsilon))}$, we also have to assure that $k \geq \frac{2\cdot(1+\delta)}{\delta\cdot\log(1/(1-\epsilon))}$. Hence we choose

$$k = \max\left\{\frac{2}{\epsilon}, \frac{2\cdot(1+\delta)}{\delta\cdot\log(1/(1-\epsilon))}\right\}, \quad k_0 = 2k\cdot\frac{1}{1-\epsilon} \tag{7}$$

and finally obtain

Theorem 7.1. *Algorithm $\mathcal{A}_{DCL}$ with the choices (7) for k and k_0 is a PTAS for the ϵ-Dense Group Steiner Tree Problem.*

8 Open Problems

We leave it as an open problem to construct a ptas for the ϵ-Dense Steiner Forest Problem or to show that such a ptas cannot exist (under some reasonable assumption). Another challenging open problem is to prove NP-hardness for the ϵ-dense version of at least one of the problems studied in this paper. Furthermore it would be interesting to investigate the fixed-parameter complexity of the ϵ-dense Steiner Tree Problem.

Acknowledgement. We would like to thank Marek Karpinski and Claus Viehmann for valuable discussions.

References

[1] S. Arora and G. George Karakostas. A 2+epsilon approximation for the k-mst problem. In *Proc. SIAM Symp. Discrete Algorithms (SODA)*, 2000.

[2] Sanjeev Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *Journal of the ACM*, 45(5):753–782, 1998.

[3] M. Chlebikov and J. Chlebikova. Approximation hardness of the steiner tree problem on graphs. In *Proceedings of 8th Scandinavian Workshop on Algorithm Theory (SWAT)*, pages 170–179. LNCS 2368, 2002.

[4] S.E. Dreyfus and R.A. Wagner. The steiner problem in graphs. *Networks*, 1:195–207, 1971.

[5] Garg, Konjevod, and Ravi. A polylogarithmic approximation algorithm for the group steiner tree problem. In *SODA: ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[6] M. Goemans and D. Williamson. A general approximation technique for constrained forest problems. In *SODA: ACM-SIAM Symposium on Discrete Algorithms*, 1992.

[7] S.L. Hakimi. Steiner's problem in graphs and its implications. *Networks*, 1:113–133, 1971.

[8] E. Halperin and R. Krauthgamer. Polylogarithmic inapproximability. In *In Proc. of STOC*, 2003.

[9] M. Hauptmann. *Approximation Complexity of Optimization Problems: Structural Foundations and Steiner Tree Problems*, PhD-Thesis, University of Bonn, 2004, URL: `http://hss.ulb.uni-bonn.de/diss_online/math_nat_fak/2004/hauptmann_mathias/0380.pdf`.

[10] M. Hauptmann. *On the Approximability of Dense Steiner Tree Problems*, Proc. Tenth Italian Conference on Theoretical Computer Science, p. 15–26, World Scientific, 2007.

[11] M. Karpinski and A. Zelikovsky. Approximating dense cases of covering problems. In *Proc. of the DIMACS Workshop on Network Design: Connectivity and Facilites Location, Princeton University, NJ, April, 1997. DIMACS series in Disc. Math. and Theor. Comp. Sci. 40*, pages 169–178, 1998.

[12] R. Ravi. A primal-dual approximation algorithm for the steiner forest problem. *Information Processing Letters*, 50(4):185–190, 1994.

[13] G. Robins and A. Zelikovsky. Improved steiner tree approximation in graphs. In *Proc. 11th Ann. ACM-SIAM Symp. on Discrete Algorithms*, pages 770–779. ACM-SIAM, 2000.