Approximating Dense Cases of Covering Problems

Marek Karpinski and Alexander Zelikovsky

ABSTRACT. We study dense instances of several covering problems. An instance of the set cover problem with m sets is dense if there is $\epsilon > 0$ such that any element of the ground set X belongs to at least $\epsilon \cdot m$ sets. We show that the dense set cover problem can be approximated with the performance ratio $c \cdot \log |X|$ for any $c > 0$ though it is unlikely to be NP-hard. A polynomial-time approximation scheme is constructed for the dense Steiner tree problem in n-vertex graphs, i.e. for the case when each terminal is adjacent to at least $\epsilon \cdot n$ vertices. We also study the vertex cover problem in dense graphs, i.e. graphs with the bounded minimum or average vertex degree. A better approximation algorithm is suggested. Its performance ratio is $\frac{2}{1+\epsilon}$ and $\frac{2}{2-\sqrt{1-\epsilon}}$ for graphs with $\epsilon|V|$-bounded minimum and average vertex degree, respectively. We also consider *superdense* (complementary to sparse) cases of all these problems.

1. Introduction

There has been some recent success in designing approximation algorithms for dense cases of certain problems that are hard to approximate in general. In papers [**AKK95, AFK96, FK96, FV95**] polynomial-time approximation schemes where suggested for such problems as Max Cut and special cases of Quadratic Assignment Problems. On the other hand, these problems are MAX SNP-hard and existence of polynomial-time approximation schemes for general cases of such problems will imply that NP equal to P [**ALMSS92**].

In this paper we address dense cases of several covering problems for which methods developed recently in [**AKK95, AFK96, FK96**] do not apply. We also introduce *superdense* graphs (i.e. graphs that are complement to sparse graphs) and discuss approximation complexity of superdense cases of covering problems.

In the next section we deal with the dense set cover problem.

Set Cover Problem. Let $X = \{x_1, ..., x_k\}$ be a finite set and $P = \{p_1, ..., p_m\} \subseteq 2^X$ be a family of its subsets. Find minimum size sub-family M of P such that $X \subseteq \cup\{p | p \in M\}$.

An instance of the set cover problem is ϵ-*dense* if there is $\epsilon > 0$ such that any element of X belongs to at least $\epsilon \cdot m$ sets from P. We show that the dense set cover problem can be approximated with the performance ratio $c \cdot \log k$ for any

$c > 0$ though it is unlikely to be NP-hard. This is in contrast to the fact that the set cover problem cannot be approximated to within factor $(1 - o(1)) \ln k$ unless $NP \subseteq DTIME[n^{\log \log n}]$ [**F96**].

The δ-*superdense* set cover problem is the case of the set cover problem where each element of the ground set X is covered by at least $m - o(m^{\delta})$ sets of P for some $\delta < 1$. We prove that the δ-superdense set cover problem can be solved efficiently.

Section 3 is devoted to the

Steiner Tree Problem. Given a connected graph $G = (V, E)$ and a set of distinguished vertices $S \subseteq V$. Find a minimum size tree within G that spans all distinguished vertices from S.

In the ϵ-*dense* instance of the Steiner tree problem any distinguished vertex (also called *terminal*) is adjacent to at least $\epsilon \cdot |V \setminus S|$ non-terminals, i.e. vertices from $V \setminus S$.

We construct a polynomial-time approximation scheme for the dense Steiner tree problem. On the other hand, the general Steiner tree problem is MAX SNP-hard even in the case when any two terminals have a common neighbor [**BP89**].

Similarly to the set cover problem, we define a δ-*superdense* Steiner tree problem to be the case of Steiner tree problem where any terminal has at least $|V| - o(|V|^{\delta})$ neighbors outside S. The δ-superdense Steiner tree problem also can be solved efficiently.

In the last section we address the

Vertex Cover Problem. Given a graph $G = (V, E)$, find a minimum size vertex set $OPT \subseteq V$ such that at least one end of any edge belongs to OPT.

Basically, there are two definitions for dense graphs: either the density means that the *minimum* vertex degree is $\Omega(|V|)$, or the density means that the number of edges is $\Omega(|V|^2)$, i.e. the *average* vertex degree is $\Omega(|V|)$. Formally, for some $\epsilon > 0$, we call a graph G *strongly ϵ-dense* if any vertex in G has at least $\epsilon|V|$ neighbors, and we call a graph G *weakly ϵ-dense* if the average degree of a vertex in G is at least $\epsilon|V|$, i.e.

$$\frac{\sum_{v \in V} deg(v)}{|V|} \geq \epsilon \cdot |V|$$

Note that strong ϵ-density implies weak ϵ-density.

In the preliminary draft of this paper [**KZ96**] as well as in [**CT96**], it was shown that the vertex cover problem even for strongly ϵ-dense graphs is still MAX SNP-hard. We present an approximation algorithm with the improved performance bounds $\frac{2}{1+\epsilon}$ and $\frac{2}{2-\sqrt{1-\epsilon}}$ for strong and weak ϵ-dense graphs, respectively.

We say that an instance of the vertex cover problem is δ-superdense if the average vertex degree is at least $|V| - o(|V|^{\delta})$ for some $\delta < 1$. We show that the δ-superdense vertex cover problem has a polynomial-time approximation scheme.

2. Dense Set Cover Problem

In this section we analyze the greedy algorithm and its modifications applied to the ϵ-dense set cover problem. The greedy algorithm repeatedly stores a maximum size set of P in the solution and removes the elements of this set from X and from all other sets in P. This heuristic is one of the best and simplest polynomial-time algorithms for the set cover problem. Its approximation ratio has been recently improved from $1 + \ln k$ [**J74**] to $\ln k - \ln \ln k + \Theta(1)$ [**S96**] and, from the other side,

no other polynomial-time algorithm can have a substantially better performance ratio [**F96**].

LEMMA 2.1. *The size of the output of the greedy algorithm applied to the ϵ-dense set cover problem is at most $\log_{1/(1-\epsilon)} k$, where k is the size of the ground set.*

PROOF. We will show that the maximum size of a set in P is at least ϵk. Consider a bipartite graph $G = (P \cup X, E)$ where $x \in X$ and $p \in P$ are adjacent if and only if $x \in p$. The degree of any $x \in X$ is at least ϵm, so the number of edges in this graph is at least $\epsilon m k$ and, therefore, there is a set $p \in P$ with degree at least ϵm.

Each iteration of the greedy heuristic does not decrease density, since all elements which belong to the chosen set are removed from X. Therefore, the size of X after the ith iteration is at most $(1 - \epsilon)^i k$. The algorithm terminates when the size of X becomes less than 1. This means that for the last iteration i_{last} we get $(1 - \epsilon)^{i_{last} - 1} k \geq 1$ and

$$i_{last} \leq \lfloor \log_{1/(1-\epsilon)} k \rfloor$$

□

This lemma implies that the size of the optimal set cover is $O(\log k)$. So we cannot expect that the ϵ-dense set cover problem is NP-hard, since a simple $O(m^{O(\log k)})$-time exhaustive search selects the exact solution.

THEOREM 2.2. [**PY96**] *Unless $NP \subseteq DTIME[n^{\log n}]$, the ϵ-dense set cover problem is not NP-hard.*

Note that $O(\log k)$ is the tight bound for the performance ratio of the greedy heuristic applied to the ϵ-dense set cover problem. Indeed, let us consider an instance of this problem with the size of optimal solution of $\log k$. If we add two sets A and B such that $A \cup B = X$, $A \cap B = \emptyset$, then the size of the optimal solution will be two and the performance ratio of the greedy algorithm will be $\frac{1}{2} \log k$. On the other hand, unlike the general case of the set cover problem, we may decrease the constant factor as far as we want.

THEOREM 2.3. *For any $c > 0$ and any $\epsilon > 0$, there is an algorithm for the ϵ-dense set cover problem with the performance ratio $c \cdot \ln k$, where k is the size of the ground set.*

PROOF. Given an instance of the ϵ-dense set cover problem, we can check if there exists a solution containing at most r sets from P in time $O(m^r)$. Either we will find exact solution or will be sure that such a solution does not exist.

By Lemma 2.1, the greedy algorithm outputs at most

$$\log_{\frac{1}{1-\epsilon}} k = \frac{\ln k}{\ln \frac{1}{1-\epsilon}}$$

sets. Therefore, the approximation ratio of the greedy algorithm preprocessed with $O(m^r)$-exhaustive search has a performance ratio at most

$$\frac{1}{r \cdot \ln \frac{1}{1-\epsilon}} \ln k$$

□

Theorems 2.2 and 2.3 arise the following conjectures:

CONJECTURE 2.4. *The ϵ-dense set cover problem cannot be solved in polynomial time.*

CONJECTURE 2.5. *The ϵ-dense set cover problem cannot be approximated in polynomial time to within constant factor.*

Further densification leads to polynomial-time solvability of the set cover problem.

THEOREM 2.6. *The δ-superdense set cover problem can be solved in polynomial time.*

PROOF. Let each element of X is covered by at least $m - \gamma m^{\delta} = m(1 - \gamma m^{\delta-1})$ sets of P for some constant $\gamma > 0$.

We can solve exactly instances of the δ-superdense set cover problem with sufficiently small size of the family P, namely when $m \leq \gamma^{\frac{2}{1-\delta}}$. Therefore, from this point on we may assume that the size of P is large enough, i.e. $m > \gamma^{\frac{2}{1-\delta}}$. This means that the value $1 - \gamma m^{\delta-1}$ is positive and we may treat an instance of the δ-superdense problem as an instance of the ϵ-dense set cover problem with $\epsilon = 1 - \gamma m^{\delta-1}$.

By Lemma 2.1, the size of optimal solution, OPT, is at most $\log_{\gamma^{-1}m^{1-\delta}} k$. Therefore,

$$
\begin{aligned}
(\gamma^{-1}m^{1-\delta})^{OPT} &\leq k \\
OPT\log_m(\gamma^{-1}m^{1-\delta}) &\leq log_m k \\
OPT &\leq \frac{log_m k}{log_m(\gamma^{-1}m^{1-\delta})} \\
&= \frac{log_m k}{1 - \delta - \log_m \gamma}
\end{aligned}
$$

Since $m > \gamma^{\frac{2}{1-\delta}}$, we obtain $\log_m \gamma < \frac{1-\delta}{2}$ and the size of optimal solution is at most

$$
OPT < \frac{2}{1-\delta} \log_m k
$$

Thus, the exhaustive search for finding an exact solution has at most

$$
m^{OPT} < m^{\frac{2}{1-\delta}\log_m k} = k^{\frac{2}{1-\delta}}
$$

cases to consider.   □

3. Dense Steiner Tree Problem

A well-known *minimum spanning tree* heuristic [**TM80**] for the Steiner tree problem finds a minimum spanning tree M of a weighted complete graph $G' = (S, E', c)$, where the weight of any edge equals to the length of the shortest path between its ends in G. Then the minimum spanning tree heuristic replaces all edges of M with the corresponding paths in G and extracts a tree from the subgraph obtained. This heuristic can be implemented in time $O(|V|^2)$ [**M88**].

The minimum spanning tree heuristic gives a 2-approximation for the Steiner tree problem [**TM80**] and the best up-today polynomial-time approximation guarantee of Karpinski-Zelikovsky is about 1.644 [**KZ97**]. Unfortunately, the minimum

spanning tree heuristic has the same performance ratio of 2 for the dense Steiner tree problem. Note that for $\epsilon > \frac{1}{2}$, ϵ-dense Steiner tree problem is a sub-case of the network Steiner tree problem with distances 1 and 2 which is still MAX SNP-hard [**BP89**]. The Rayward-Smith heuristic [**R83**] was successfully applied to the latter problem in [**BP89**]. It gives a $\frac{4}{3}$-approximation while the minimum spanning tree heuristic still has the same tight bound of 2 as for the general case. We will use a combination of the exhaustive search, the minimum spanning tree heuristic and the Rayward-Smith heuristic.

We start with the exhaustive search. Note that a (sub)optimal *Steiner tree T* (i.e. a tree that spans S) may contain also non-terminals. Each non-terminal of degree at least 3 in T is called a *Steiner vertex* of T. If we guess a right selection of Steiner vertices, then we can find an optimal Steiner taking the minimum spanning tree of terminals and Steiner vertices. It is easy to see that any Steiner tree may contain at most $|S| - 2$ Steiner vertices. The exhaustive search of Steiner vertices combining with the minimum spanning tree heuristic takes $O(|V|^{|S|})$ runtime.

LEMMA 3.1. *An optimal Steiner tree can be found exactly in $O(|V|^{|S|})$ time.*

This lemma implies that for a sufficiently small number of terminals, e.g. $|S| \leq \frac{1}{\epsilon}$, we can find an exact solution in polynomial time. Now we will show how we reduce the size of the given terminal set using a variation of Rayward-Smith heuristic (or the greedy algorithm [**Z93**]).

Consider the subgraph $G(S)$ of the graph G induced by the set of terminals S. If the graph $G(S)$ is connected, then any spanning tree of $G(S)$ is an optimal Steiner tree since it has $|S| - 1$ edges. In general, let $\mathcal{C}[S] = \{C_1, ..., C_k\}$ be the partition of S into vertex sets of the connected components of $G(S)$. We construct a spanning tree $T(C_i)$ for each component C_i. Each such tree is an optimal Steiner tree for C_i since the number of edges $e(T(C_i)) = |C_i| - 1$ is the minimum possible. Then we collapse each component C_i into one vertex which is adjacent to all neighbors of each element of C_i.

Let $OPT(\mathcal{C}[S])$ be an optimal Steiner tree spanning collapsed terminals C_i. It is not difficult to see that $OPT(\mathcal{C}[S])$ combined with $T(C_i)$ gives an optimal solution for the original problem. Unfortunately, if the size of $\mathcal{C}[S]$ is not bounded we cannot find $OPT(\mathcal{C}[S])$ with the exhaustive search efficiently.

Now we are ready to describe our heuristic for the ϵ-dense Steiner tree problem.

Algorithm DSTP

(0) $S_{final} \leftarrow S$
(1) while $|\mathcal{C}[S_{final}]| > \frac{1}{\epsilon}$ do
    find $v \in V \setminus S_{final}$ that minimizes the size of $\mathcal{C}[S_{final} \cup v]$
    $S_{final} \leftarrow S_{final} \cup v$
(2) find an optimal Steiner tree $OPT(\mathcal{C}[S_{final}])$ for the collapsed
    $C \in \mathcal{C}[S_{final}]$ using exhaustive search
(3) output $APPR = \cup_{C \in \mathcal{C}[S_{final}]} T(C) \cup OPT(\mathcal{C}[S_{final}])$

Note that the number of edges in $T(C)$ for each $C \in \mathcal{C}[S_{final}]$ equals to $|C| - 1$, therefore, the number of edges in the output tree $APPR$ is

$$e(APPR) = |S_{final}| - |\mathcal{C}[S_{final}]| + e(OPT(\mathcal{C}[S_{final}]))$$

Now we need to bound the number of edges in the optimal solution OPT. For the purposes of analysis, let us modify the graph G by inserting edges between all pairs of vertices from the same component $C \in \mathcal{C}[S_{final}]$. This may only decrease the minimum number of edges that are necessary to span S. In the modified graph, $|\mathcal{C}[S]| = |\mathcal{C}[S_{final}]|$ and the tree $OPT(\mathcal{C}[S])$ cannot have less edges than $OPT(\mathcal{C}[S_{final}])$.

Let $SV = S_{final} \setminus S$ be the set of Steiner vertices introduced by our algorithm. Note that $|\mathcal{C}[S_{final}]| \leq \frac{1}{\epsilon}$. Thus the approximation ratio of our algorithm is at most

$$
\begin{aligned}
\frac{e(APPR)}{e(OPT)} &\leq \frac{|S_{final}| - |\mathcal{C}[S_{final}]| + e(OPT(\mathcal{C}[S_{final}]))}{|S| - |\mathcal{C}[S]| + e(OPT(\mathcal{C}[S]))} \\
&\leq \frac{|S_{final}| - |\mathcal{C}[S_{final}]|}{|S| - |\mathcal{C}[S_{final}]|} \\
&\leq \frac{|S_{final}| - \frac{1}{\epsilon}}{|S| - \frac{1}{\epsilon}} \\
&= 1 + \frac{|SV|}{|S| - \frac{1}{\epsilon}}
\end{aligned}
$$

The size of SV equals to the number of iterations in the step (1) of our algorithm. Each iteration of (1) decreases the size of $\mathcal{C}[S_{final}]$ by at least $\epsilon|\mathcal{C}[S_{final}]| - 1$. Thus, after i-th iteration $|\mathcal{C}[S_{final}]| \leq (|S| - \frac{1}{\epsilon})(1 - \epsilon)^i + \frac{1}{\epsilon}$. The loop (1) terminates when $|\mathcal{C}[S_{final}]| < \frac{1}{\epsilon} + 1$, so

$$
(3.1) \qquad\qquad |SV| \leq \log_{1/(1-\epsilon)}(|S| - \frac{1}{\epsilon})
$$

Finally, we obtain the following

LEMMA 3.2. *An approximation ratio of Algorithm DSTP is at most*

$$
1 + \frac{\log_{1/(1-\epsilon)}(|S| - \frac{1}{\epsilon})}{|S| - \frac{1}{\epsilon}}
$$

Given an arbitrary approximation ratio $1 + \alpha$, $\alpha > 0$, our strategy is to solve exactly in polynomial time (for fixed ϵ and α) instances of the dense Steiner tree problem with small number of terminals, i.e. when $|S|$ satisfies the following inequality

$$
\frac{\log_{1/(1-\epsilon)}(|S| - \frac{1}{\epsilon})}{|S| - \frac{1}{\epsilon}} > \alpha.
$$

Otherwise, the number of terminals is sufficiently big, and we apply Algorithm DSTP with the approximation ratio at most $1 + \alpha$. Thus we obtain the following

THEOREM 3.3. *There is a polynomial-time approximation scheme for the ϵ-dense Steiner tree problem.*

It is not difficult to see that there is a polynomial time reduction of the ϵ-dense set cover problem to the ϵ-dense Steiner tree problem and vice versa. Therefore, we cannot expect that the ϵ-dense Steiner tree problem NP-hard.

Similarly to the set cover problem, further densification leads to polynomial-time solvability of the Steiner tree problem.

THEOREM 3.4. *The δ-superdense Steiner tree problem can be solved exactly in polynomial time.*

PROOF. Let any terminal be adjacent to at least $m - \gamma m^\delta$ vertices from $V \setminus S$ for some $\delta < 1$, where $m = |V \setminus S|$. We can solve the problem using the exhaustive search if

$$m^{1-\delta} \leq \max\{\gamma^2, 2\gamma\}.$$

Otherwise, we will try any possible set of vertices from $V \setminus S$. Thus it is sufficient to show that there is an optimal Steiner tree with sufficiently small number of such vertices.

The proof is similar to the proof of Theorem 2.6. Since $m^{1-\delta} > \gamma^2$, (3.1) yields that $|SV| \leq \log_{1/(1-\epsilon)}(|S|)$, and treating the problem as ϵ-dense problem with $\epsilon = 1 - \gamma m^{\delta-1}$ we obtain that

$$|SV| \leq \frac{2}{1-\delta} \log_m |S|$$

From the other side, since $m^{1-\delta} > 2\gamma$ we obtain $\epsilon > \frac{1}{2}$. Therefore, no other Steiner vertices are necessary to span $|S|$. Finally, we obtain that at most

$$m^{|SV|} \leq m^{\frac{2}{1-\delta} \log_m |S|} = |S|^{\frac{2}{1-\delta}}$$

cases should be considered. $\square$

4. Dense Vertex Cover Problem

The vertex cover problem is one of the first NP-hard problems for which approximation algorithms were suggested. The well-known 2-approximation algorithm picks up an arbitrary edge e of the given graph $G = (V, E)$, removes e together with all adjacent edges and stores the both ends of e in the solution. The algorithm repeats this procedure until the graph G runs out of edges. We will denote this algorithm $2VC$. The algorithm $2VC$ has the best up to-day approximation ratio for the general vertex cover problem.

A straightforward application of $2VC$ to ϵ-dense graphs has the same approximation ratio 2. In this section we suggest a better approximation algorithm. It combines $2VC$ with the exhaustive search.

In order to describe our heuristic we need some more denotations. Let $N(v)$ denote the set of neighbors of a vertex v. Let $G(V')$ be a subgraph induced by a subset of vertices $V' \subseteq V$. Let us apply the algorithm $2VC$ to the graph $G(V')$. A vertex cover of $G(V')$ obtained will be denoted by $2VC(V')$. Also, given $\epsilon > 0$, we denote by $H(\epsilon)$ the set of all vertices of degree at least $(1 - \sqrt{1-\epsilon}) \cdot |V|$.

Algorithm DVC

(1) for each vertex v from $H(\epsilon)$ do
$\qquad V' \leftarrow V \setminus (N(v) \cup \{v\})$
$\qquad VC(v) \leftarrow N(v) \cup 2VC(V')$
(2) $VC_1 \leftarrow$ the minimum size set among sets $VC(v)$, over all $v \in H(\epsilon)$
(3) $VC_2 \leftarrow H(\epsilon) \cup 2VC(V \setminus H(\epsilon))$
(4) $APPR \leftarrow$ the set of the minimum size among VC_1 and VC_2

Our first step in analysis of Algorithm DVC is to bound the size of a vertex cover which does not contain at least one high-degree vertex.

LEMMA 4.1. *Let OPT be an optimal vertex cover and let $v \in V \setminus OPT$. Then the vertex cover $VC(v)$ from Algorithm DVC cannot have size more than*

$$|VC(v)| \leq \frac{2}{1 + \frac{|N(v)|}{|V|}} \cdot |OPT|$$

PROOF. Since v does not belong to OPT, OPT should contain the neighborhood of v, $N(v)$, in order to cover all edges incident to v. Moreover, $N(v)$ covers all edges between $N(v)$ and $V' = V \setminus (N(v) \cup \{v\})$. Hence, all other vertices of OPT, denoted $OPT' = OPT - N(v)$, should cover only the edges of $G(V')$.

The vertex cover $2VC(V')$ has size at most $2|OPT'|$. Moreover, $|2VC(V')|$ cannot be more than $|V'|$. So the approximation ratio can be bounded as follows.

$$\frac{|VC(v)|}{|OPT|} \leq \frac{|N(v)| + \min\{2|OPT'|, |V'|\}}{|N(v)| + |OPT'|}$$

It is easy to see that the bigger the size of OPT' the worse upper bound we get while $|OPT'|$ is less than half of $|V'|$. Therefore, we meet the worst case of our upper bound when $2|OPT'| = |V'|$.

$$\begin{aligned}
\frac{|VC(v)|}{|OPT|} &\leq \frac{|N(v)| + |V'|}{|N(v)| + \frac{|V'|}{2}} \\
&\leq \frac{|V|}{\frac{|N(v)|}{2} + \frac{|V|}{2}} \\
&= \frac{2}{1 + \frac{|N(v)|}{|V|}}
\end{aligned}$$

$\square$

THEOREM 4.2. *The algorithm DVC has an approximation ratio at most $\frac{2}{1+\epsilon}$ for strongly ϵ-dense graphs.*

PROOF. The strong ϵ-density bounds the minimum vertex degree by $\epsilon \cdot |V|$, from the other side, any optimal vertex cover cannot contain all vertices. Thus, Theorem follows from Lemma 4.1 straightforwardly. $\square$

THEOREM 4.3. *The algorithm DVC has an approximation ratio at most $\frac{2}{2-\sqrt{1-\epsilon}}$ for weakly ϵ-dense graphs.*

PROOF. If some $v \in H(\epsilon)$ does not belong to OPT, then Theorem follows from Lemma 4.1. But in the case of weak ϵ-density, we cannot guarantee that there is an optimal vertex cover which does not contain at least one high-degree vertex. We will show that the size of $H(\epsilon)$ is big enough and, in case of all vertices from $H(\epsilon)$ belong to OPT, the size of VC_2 is sufficiently small.

Indeed, the degree of any vertex in $H(\epsilon)$ is at most $|V|$ and any vertex in $V \setminus H(\epsilon)$ has degree at most $\alpha|V|$, where $\alpha = 1 - \sqrt{1 - \epsilon}$. Summing these upper bounds on vertex degrees over all vertices we obtain:

$$|H(\epsilon)| \cdot |V| + (|V| - |H(\epsilon)|)\alpha|V| \geq 2|E|$$

By definition of the weak ϵ-density, $2|E| \geq \epsilon|V|^2$. Hence,

$$\begin{aligned}
|H(\epsilon)| &\geq \frac{\epsilon - \alpha}{1 - \alpha}|V| \\
&= \frac{\epsilon - 1 + \sqrt{1 - \epsilon}}{\sqrt{1 - \epsilon}}|V| \\
&= (1 - \sqrt{1 - \epsilon})|V|
\end{aligned}$$

We may assume that the whole set $H(\epsilon)$ belongs to an optimal vertex cover OPT; otherwise, as we mentioned above, Theorem 4.3 follows from Lemma 4.1. We denote $OPT' = OPT - H(\epsilon)$. Similar to the proof of Lemma 4.1, we obtain

$$\begin{aligned}
\frac{|VC_2|}{|OPT|} &\leq \frac{|H(\epsilon)| + \min\{|2VC(V \setminus H(\epsilon))|, |V| - |H(\epsilon)|\}}{|H(\epsilon)| + |OPT'|} \\
&\leq \frac{|V|}{|H(\epsilon)| + \frac{|V| - |H(\epsilon)|}{2}} \\
&= \frac{2}{1 + \frac{|H(\epsilon)|}{|V|}} \\
&\leq \frac{2}{2 - \sqrt{1 - \epsilon}}
\end{aligned}$$

$\square$

Theorems 4.2 and 4.3 show that the density can help in approximating the vertex cover problem. On the other hand, we cannot expect any polynomial-time approximation scheme for the strongly (as well as weakly) dense vertex cover problem, since it is MAX NP-hard (see [**KZ96, CT96**]).

Further densification (as for the set cover problem and the Steiner tree problem) implies further decreasing of the approximation complexity. Let us consider the δ-superdense vertex cover problem, i.e. the case of the vertex cover problem when the average vertex degree is at least $|V| - \gamma|V|^\delta$ for some $\delta < 1$. We want to show that for any $\alpha > 0$ there exists a polynomial-time algorithm with the approximation ratio at most $1 + \alpha$.

Indeed, we can solve efficiently instances of the problem when $|V|^{1-\delta} \leq \gamma(\frac{2\alpha}{1+\alpha})^2$. Otherwise, we may view the superdense problem as the ϵ-dense vertex cover problem with $\epsilon = 1 - \frac{\gamma}{|V|^{1-\delta}} \geq 1 - (\frac{2\alpha}{1+\alpha})^2$. If we substitute the last lower bound on ϵ into the formula in Theorem 4.3, we obtain the upper bound $1 + \alpha$ on the approximation ratio of Algorithm DVC. Thus we proved

THEOREM 4.4. *The δ-superdense vertex cover problem has a polynomial-time approximation scheme.*

Acknowledgment

References

[AFK96] S. Arora, A. Frieze and H. Kaplan, *A new rounding procedure for assignment problem with applications to dense graph arrangement problems.* In Proc. of 37th Annual IEEE Symp. on Foundations of Computer Science, 21–30, 1996.

[AKK95] S. Arora, D. Karger, and M. Karpinski, *Polynomial Time Approximation Schemes for Dense Instances of NP-Hard Problems.* In Proc. of 27th ACM Symp. on Theory of Computing, 284–293, 1995.

[ALMSS92] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, *Proof verification and hardness of approximation problems.* In Proc. of 33d Annual IEEE Symp. on Foundations of Computer Science, 14–23, 1992.

[BP89] M. Bern and P. Plassmann, *The Steiner problems with edge lengths 1 and 2.* Inform. Process. Lett. **32**: 171–176, 1989.

[CT96] A.E.F. Clementi and L. Trevisan, *Improved non-approximability results for vertex cover with density constraints.* In: Computing and Combinatorics. Proc. of 2nd Annual International Conference. COCOON '96. Berlin, Germany: Springer-Verlag, 333-342, 1996.

[F96] U. Feige, *A threshold of* $\ln n$ *for approximating set cover.* In Proc. of 28th ACM Symp. on Theory of Comp., 314–318, 1996.

[FV95] W. Fernandez-de-la-Vega, *Max-Cut has a randomized approximation scheme in dense graphs.* Randomized Structures and Algorithms, to appear, 1995.

[FK96] A. Frieze and R. Kannan, *The regularity lemma and approximation schemes for dense problems.* In Proc. of 37th Annual IEEE Symp. on Foundations of Computer Science, 12–20, 1996.

[J74] D.S. Johnson, *Approximation algorithms for combinatorial problems.* Journal of Computer and System Sciences, **9**: 256–278, 1974.

[KZ96] M. Karpinski and A. Zelikovsky, *Approximating dense cases of covering problems.* Technical Report TR-96-059, International Computer Science Institute, Berkeley, 1996.

[KZ97] M. Karpinski and A. Zelikovsky, *New approximation algorithms for the Steiner tree problem.* Journal of Combinatorial Optimization **1**: 47–65, 1997.

[M88] K. Mehlhorn, *A faster approximation algorithm for the Steiner problem in graphs.* Inf. Process. Lett. **27**: 125–128, 1988.

[PY96] C.H. Papadimitriou and M. Yannakakis, *On limited nondeterminism and the complexity of the V-C dimension.* Journal of Computer and System Sciences, **53**: 161-170, 1996.

[S96] P. Slavik, *A tight analysis of the greedy algorithm for set cover.* In Proc. of 28th ACM Symp. on Theory of Comp., 435–439, 1996.

[R83] V. J. Rayward-Smith, *The computation of nearly minimal Steiner trees in graphs.* International J. Math. Ed. Sci. Tech. **14**: 15–23, 1983.

[TM80] H. Takahashi and A. Matsuyama, *An approximate solution for the Steiner problem in graphs.* Math. Japonica, **24**: 573–577, 1980.

[Z93] A. Z. Zelikovsky, *An 11/6-approximation algorithm for the network Steiner problem.* Algorithmica **9**: 463–470, 1993.

DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF BONN, 53117 BONN
E-mail address: `marek@cs.uni-bonn.de`

DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF VIRGINIA, CHARLOTTESVILLE, 22903 VIRGINIA
E-mail address: `alexz@cs.virginia.edu`