Improved Approximation Algorithms for Relay Placement

Alon Efrat[1], Sándor P. Fekete[2], Poornananda R. Gaddehosur[1], Joseph S. B. Mitchell[3], Valentin Polishchuk[4], and Jukka Suomela[4]

[1] The University of Arizona, alon@email.arizona.edu, poorna@email.arizona.edu
[2] Braunschweig University of Technology, s.fekete@tu-bs.de
[3] Stony Brook University, jsbm@ams.stonybrook.edu
[4] Helsinki Institute for Information Technology HIIT, valentin.polishchuk@cs.helsinki.fi, jukka.suomela@cs.helsinki.fi

Abstract. In the relay placement problem the input is a set of sensors and a number $r \geq 1$, the communication range of a relay. In the *one-tier* version of the problem the objective is to place a minimum number of relays so that between every pair of sensors there is a path *through sensors and/or relays* such that the consecutive vertices of the path are within distance r if both vertices are relays and within distance 1 otherwise. The *two-tier* version adds the restrictions that the path must go *through relays, and not through sensors*. We present a 3.11-approximation algorithm for the one-tier version and a PTAS for the two-tier version. We also show that the one-tier version admits no PTAS, assuming P $\neq$ NP.

1 Introduction

A sensor network consists of a large number of low-cost autonomous devices, called *sensors*. Communication between the sensors is performed by wireless radio with very limited range, e.g., via the Bluetooth protocol. To make the network connected, a number of additional devices, called *relays*, must be judiciously placed within the sensor field. Relays are typically more advanced and expensive than sensors. For instance, in addition to a Bluetooth chip, each relay may be equipped with a WLAN transceiver, enabling communication between distant relays. The problem we study in this paper is that of placing a *minimum number* of relays to ensure the connectivity of a sensor network.

Two models of communication between sensors have been considered in the literature [1–8]. In both models, a sensor and a relay can communicate if the distance between them as at most 1, and two relays can communicate if the distance between them is at most r, where $r \geq 1$ is a given number. The models differ in whether direct communication between sensors is allowed. In the *one-tier* model two sensors can communicate if the distance between them is at most 1. In the *two-tier* model the sensors do not communicate at all, no matter how close they are. In other words, in the two-tier model the sensors may only link to relays, but not to other sensors.

Formally, the input to the relay placement problem is a set of n sensors, identified with their locations in the plane, and a number $r \geq 1$, the communication range of a relay (w.l.o.g. the communication range of a sensor is 1). The objective in the *one-tier* relay placement is to place a minimum number of relays so that between every pair of sensors there exists a path, *through sensors and/or relays*, such that the consecutive vertices of the path are within distance r if both vertices are relays, and within distance 1 otherwise. The objective in the *two-tier* relay placement is to place a minimum number of relays so that between every pair of sensors there exists a path *through relays* such that the consecutive vertices of the path are within distance r if both vertices are relays, and within distance 1 if one of the vertices is a sensor and the other is a relay (going directly from a sensor to a sensor is forbidden).

Previous Work. The current best approximation ratio of 7 for one-tier relay placement is due to Lloyd and Xue [5]. For the two-tier placement Lloyd and Xue [5] gave a $(5+\varepsilon)$-approximation algorithm for arbitrary $r \geq 1$; Srinivas et al. [6] gave a $(4+\varepsilon)$-approximation for the case $r \geq 2$. See references in [5,6] for earlier works.

Contributions. We present new results on approximability of replay placement:

- In Section 3 we give a simple $O(n \log n)$-time 6.73-approximation algorithm for the one-tier version.
- In Section 4 we present a polynomial-time 3.11-approximation algorithm for the one-tier version.
- In Section 5 we show that there is no PTAS for one-tier relay placement (assuming that r is part of the input, and $\mathrm{P} \neq \mathrm{NP}$).
- In Section 6 we give a PTAS for two-tier relay placement.

Note that the *number* of relays in a solution may be exponential in the size of the input (number of bits). Our algorithms produce a succinct representation of the solution. The representation is given by a set of points and a set of line segments; the relays are placed on each point and equally-spaced along each segment.

2 Blobs, Clouds, Stabs, Hubs, and Forests

In this section we introduce the notions, central to the description of our algorithms for one-tier relay placement. We also provide lower bounds.

Blobs and Clouds. We write $|xy|$ for the Euclidean distance between x and y. Let V be a given set of sensors (points in the plane). We form a unit disk graph $\mathcal{G} = (V, E)$ and a disk graph $\mathcal{F} = (V, F)$ where $E = \{\{u, v\} : |uv| \leq 1\}$, $F = \{\{u, v\} : |uv| \leq 2\}$; see Fig. 1.

A *blob* is defined to be the union of the unit disks centred at the sensors that belong to one connected component of $\mathcal{G}$. We use B to refer to a blob, and $\mathcal{B}$ for the set of all blobs.

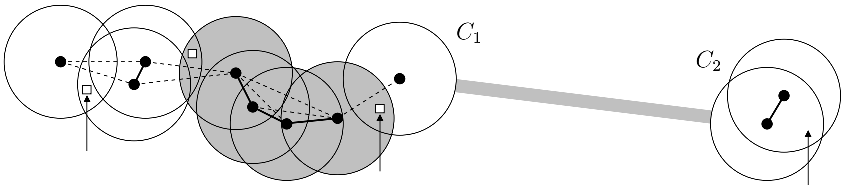

Fig. 1. Dots are sensors in V, solid lines are edges in E and F, and dashed lines are edges in F only. There are 5 blobs in $\mathcal{B}$ (one of them highlighted) and 2 clouds $C_1, C_2 \in \mathcal{C}$. Arrows are stabs in $\mathrm{Stab}(\mathcal{B})$, small rectangles are hubs. The wide grey line is the only edge in $\mathrm{MStFN}(\mathcal{C})$, which happens to be equal to $\mathrm{MSFN}(\mathcal{C})$ here.

Analogously, a *cloud* $C \in \mathcal{C}$ is the union of the unit disks centred at the sensors that belong to the connected component of the graph $\mathcal{F}$. The sensors in a blob can communicate with each other without relays, while the ones in a cloud might not, even though their disks may overlap. Each cloud $C \in \mathcal{C}$ consists of one or more blobs $B \in \mathcal{B}$; we use $\mathcal{B}_C$ to denote the blobs that form the cloud C.

Stabs and Hubs. A *stab* is a relay with an infinite communication range ($r = \infty$). A *hub* is a relay without the ability to communicate with the other relays. As we shall see, a solution to stab or hub placement can be used as the first step towards a solution for relay placement.

If we are placing stabs, it is necessary and sufficient to have a stab in each blob to ensure communication between all sensors (to avoid trivialities we assume there is more than one blob). Thus, stab placement is equivalent to the set cover problem: the universe is the blobs, and the subsets are sets of blobs that have a point in common. We use $\mathrm{Stab}(\mathcal{B}')$ to denote the minimum set of stabs that stab each blob in $\mathcal{B}' \subseteq \mathcal{B}$. In the example in Fig. 1 arrows show an optimal solution to the stab placement problem; 3 stabs are enough.

If we are placing hubs, it is necessary (assuming more than one blob in the cloud), but not sufficient, to have a hub in each blob to ensure communication between sensors within one cloud. In fact, hub placement can be interpreted as a special case of the *connected* set cover problem [9,10]. In the example in Fig. 1 small rectangles show an optimal solution to the hub placement problem for the cloud $C = C_1$; in this particular case, 2 stabs within the cloud C were sufficient to "pierce" each blob in $\mathcal{B}_C$, however, an additional hub is required to "stitch" the blobs together. The next Lemma shows that in general there may be at most as many additional hubs needed as there were stabs:

Lemma 1. *Given a feasible solution S to stab placement on $\mathcal{B}_C$, we can obtain in polynomial time a feasible solution to hub placement on $\mathcal{B}_C$ with $2|S|-1$ hubs.*

Proof. Let $\mathcal{H}$ be the graph, whose nodes are the sensors in the cloud C and the stabs in S, and whose edges connect two devices if either they are within distance 1 from each other or if both devices are stabs (i.e., there is an edge between *every* pair of the stabs). Switch off communication between the stabs, thus turning them into hubs. Suppose that this breaks $\mathcal{H}$ into k connected components. There must be a stab in each connected component. Thus, $|S| \geq k$.

If $k > 1$, by the definition of a cloud, there must exist a point where a unit disk covers at least two sensors from two different connected components of $\mathcal{H}$. Placing a hub at the point decreases the number of the connected components by at least 1. Thus, after putting at most $k-1$ additional hubs, all connected components will merge into one.

Steiner Forests and Spanning Forests with Neighbourhoods. Let $\mathcal{P}$ be a collection of planar subsets; call them *neighbourhoods.* (In Section 3 the neighbourhoods will be the clouds, in Section 4 they will be "clusters" of clouds.) For a plane graph G, let $\mathcal{G}_{\mathcal{P}} = (\mathcal{P}, E(G))$ be the graph whose vertices are the neighbourhoods and two neighbourhoods $P_1, P_2 \in \mathcal{P}$ are adjacent whenever G has a vertex in P_1, a vertex in P_2, and a path between the vertices.

The *Minimum Steiner Forest with Neighbourhoods* on $\mathcal{P}$, denoted MStFN($\mathcal{P}$), is a *minimum-length* plane graph G such that $\mathcal{G}_{\mathcal{P}} = (\mathcal{P}, E(G))$ is connected. The MStFN is a generalisation of the Steiner tree of a set of points. Note that MStFN is slightly different from Steiner tree with neighbourhoods (see, e.g., [11]) in that we are only counting the part of the graph *outside* $\mathcal{P}$ towards its length.

Consider a complete weighted graph whose vertices are the neighbourhoods in $\mathcal{P}$ and whose edge weights are the shortest distances between them. A minimum spanning tree in the graph is called the *Minimum Spanning Forest with Neighbourhoods* on $\mathcal{P}$, denoted MSFN($\mathcal{P}$). A natural embedding of the edges of the forest is by the straight-line segments that connect the corresponding neighbourhoods; we will identify MSFN($\mathcal{P}$) with the embedding. (As with MStFN, we count the length of MSFN only *outside* $\mathcal{P}$.)

We denote by $|\text{MStFN}(\mathcal{P})|$ and $|\text{MSFN}(\mathcal{P})|$ the total length of the edges of the forests. It is known that $|\text{MSFN}(P)| \leq (2/\sqrt{3})|\text{MStFN}(P)|$ for a *point* set P, where $2/\sqrt{3}$ is the *Steiner ratio* [12]. The following lemma generalises this to neighbourhoods.

Lemma 2. *For any* $\mathcal{P}$, $|\text{MSFN}(\mathcal{P})| \leq (2/\sqrt{3})|\text{MStFN}(\mathcal{P})|$.

Proof. If $\mathcal{P}$ is erased, MStFN($\mathcal{P}$) falls off into a forest, each tree of which is a minimum Steiner tree on its leaves; its length is within the Steiner ratio of minimum spanning tree length.

Lower Bounds on the Number of Relays. Let R^* be an optimal set of relays. Let $\mathcal{R}$ be the communication graph on the relays R^* alone, i.e., without sensors taken into account; two relays are connected by an edge in $\mathcal{R}$ if and only if they are within distance r from each other. Suppose that $\mathcal{R}$ is embedded in the plane with vertices at relays and line segments joining communicating relays. The embedding spans all clouds, for otherwise the sensors in a cloud would not be connected to the others. Thus, in $\mathcal{R}$ there exists a forest $\mathcal{R}'$, whose embedding also spans all clouds. Let $|\mathcal{R}'|$ denote the total length of the edges in $\mathcal{R}'$. By definition of MStFN($\mathcal{C}$), we have $|\mathcal{R}'| \geq |\text{MStFN}(\mathcal{C})|$.

Let m, v, and k be the number of edges, vertices, and trees of $\mathcal{R}'$. Since each edge of $\mathcal{R}'$ has length at most r, we have $|\mathcal{R}'| \leq mr = (v-k)r$. Since $v \leq |R^*|$, since there must be a relay in every blob and every cloud, and since the clouds are disjoint, it follows that

$$|R^*| \geq |\text{MStFN}(\mathcal{C})|/r, \qquad |R^*| \geq |\text{Stab}(\mathcal{B})|, \qquad |R^*| \geq |\mathcal{C}|. \tag{1}$$

3 A 6.73-Approximation for One-Tier Relay Placement

In this section we give a simple $O(n \log n)$-time 6.73-approximation algorithm for relay placement. We first find an approximately optimal stab placement. Then we turn a stab placement into a hub placement within each cloud. Then a spanning tree on the clouds is found and "Steinerised".

Finding an approximately optimal stab placement is a special case of the set cover problem. The maximum number of blobs pierced by a single stab is 5. Thus, in this case the greedy heuristic for the set cover has an approximation ratio of $1 + 1/2 + 1/3 + 1/4 + 1/5 = 137/60$ [13, Thm. 35.4].

Based on this approximation, a feasible hub placement R_C within one cloud $C \in \mathcal{C}$ can be obtained by applying Lemma 1; for this set of hubs it holds that $|R_C| \leq 137|\text{Stab}(\mathcal{B}_C)|/30 - 1$. We can now interpret hubs R_C as relays; if the hubs make the cloud C connected, surely it holds for relays.

Let $R' = \bigcup_C R_C$ denote all relays placed this way. Since the blobs $\mathcal{B}_C$ for different C do not intersect, $|\text{Stab}(\mathcal{B})| = \sum_C |\text{Stab}(\mathcal{B}_C)|$, so

$$|R'| \leq \textstyle\sum_C |R_C| \leq \sum_C (137|\text{Stab}(\mathcal{B}_C)/30| - 1) = 137|\text{Stab}(\mathcal{B})|/30 - |\mathcal{C}|. \tag{2}$$

Next, we find $\text{MSFN}(\mathcal{C})$ and place another set of relays, R'', along its edges. Specifically, for each edge e of the forest, we place 2 relays at the endpoints of e, and $\lfloor |e|/r \rfloor$ relays every r units starting from one of the endpoints. This ensures that all clouds communicate with each other; thus $R = R' \cup R''$ is a feasible solution. Since the number of edges in $\text{MSFN}(\mathcal{C})$ is $|\mathcal{C}| - 1$,

$$|R''| = 2(|\mathcal{C}| - 1) + \textstyle\sum_e \lfloor |e|/r \rfloor < 2|\mathcal{C}| + |\text{MSFN}(\mathcal{C})|/r. \tag{3}$$

We obtain $|R| = |R'| + |R''| \leq (137/30 + 1 + 2/\sqrt{3})|R^*| < 6.73|R^*|$ from (1), (2), (3) and Lemma 2.

We sketch here an $O(n \log n)$-time implementation of the algorithm. From the Delaunay triangulation (DT) of the sensors one can identify the blobs and the clouds, and from the 5-order Voronoi diagram — the possible locations for the stabs/hubs. Since each such location belongs to at most 5 blobs, our set cover instance has linear size and can be solved in linear time [13, Problem 35.3-3]. Finally, $\text{MSFN}(\mathcal{C})$ is a subgraph of DT and can be found in $O(n \log n)$ time.

4 A 3.11-Approximation for One-Tier Relay Placement

In this section we first take care of clouds whose blobs can be stabbed with few relays, and then find an approximation to the hub placement by greedily placing the hubs themselves, without placing the stabs first, for the rest of the clouds. Together with a refined analysis, this gives a polynomial-time 3.11-approximation algorithm. We focus on nontrivial instances with more than one blob.

Overview. The basic steps of our algorithm are as follows:

1. Compute optimal stabbings for clouds which can be stabbed with few relays.
2. Connect the blobs in each of these clouds, using Lemma 1.
3. Greedily connect all blobs in each of the remaining clouds ("stitching").
4. Greedily connect clouds into clusters, using 2 additional relays per cloud.
5. Connect the clusters by a spanning forest.

Our algorithm constructs a set A_r of "red" relays (for connecting blobs in a cloud, i.e., relays added in steps 1–3), a set A_g of "green" relays (two per cloud, added in steps 4–5) and a set A_y of "yellow" relays (outside of sensor range, added in step 5). In the analysis, we compare an optimal solution R^* to our approximate one by subdividing the former into a set R_d^* of "dark" relays that are within reach of sensors, and into a set R_ℓ^* of "light" relays that are outside of sensor range. We compare $|R_d^*|$ with $|A_r| + |A_g|$, and $|R_\ell^*|$ with $|A_y|$, showing in both cases that the ratio is less than 3.11.

Clouds with Few Stabs. For any constant k, it is straightforward to check in polynomial time whether all blobs in a cloud $C \in \mathcal{C}$ can be stabbed with $i < k$ stabs. Using Lemma 1, we can connect all blobs in such a cloud with at most $2i - 1$ red relays. We denote by $\mathcal{C}^i$ the set of clouds where the minimum number of stabs is i, and by $\mathcal{C}^{k+}$ the set of clouds that need at least k stabs.

Stitching a Cloud from $\mathcal{C}^{k+}$. We focus on one cloud $C \in \mathcal{C}^{k+}$. For a point y in the plane, define $\mathcal{B}(y) = \{B \in \mathcal{B}_C : y \in B\}$ to be the set of blobs that contain the point; obviously $|\mathcal{B}(y)| \leq 5$ for any y. For a any subset of blobs $\mathcal{T} \subseteq \mathcal{B}_C$, define $\mathcal{S}(\mathcal{T}, y) = \mathcal{B}(y) \setminus \mathcal{T}$ to be the set of blobs *not from* $\mathcal{T}$ containing y, and define $V(\mathcal{T})$ to be the set of sensors that form the blobs in $\mathcal{T}$.

Within C, we place a set of red relays $A_r^C = \{y_j : j = 1, 2, \dots\}$, as follows:

1. Choose arbitrary $B_0 \in \mathcal{B}_C$. Initialise $j \leftarrow 1$, $\mathcal{T}_j \leftarrow \{B_0\}$.
2. While $\mathcal{T}_j \neq \mathcal{B}_C$:

$$y_j \leftarrow \arg\max_y \{|\mathcal{S}(\mathcal{T}_j, y)| : \mathcal{B}(y) \cap \mathcal{T}_j \neq \emptyset\},$$
$$\mathcal{S}_j \leftarrow \mathcal{S}(\mathcal{T}_j, y_j), \quad \mathcal{T}_{j+1} \leftarrow \mathcal{T}_j \cup \mathcal{S}_j, \quad j \leftarrow j + 1.$$

By induction on j, after each iteration, there exists a path through sensors and/or relays between any pair of sensors in $V(\mathcal{T}_j)$. By the definition of a cloud, there is a line segment of length at most 2 that connects $V(\mathcal{T}_j)$ to $V(\mathcal{B}_C \setminus \mathcal{T}_j)$; the midpoint of the segment is a location y with $\mathcal{S}(\mathcal{T}_j, y) \neq \emptyset$. Thus each iteration increases the size of $\mathcal{T}_j$ by at least 1, the algorithm terminates in at most $|\mathcal{B}_C| - 1$ iterations, and $|A_r^C| \leq |\mathcal{B}_C| - 1$. The sets $\mathcal{S}_j$ form a partition of $\mathcal{B}_C \setminus \{B_0\}$.

We can prove the following performance guarantee. (The proof is similar to the analysis of greedy set cover; see the Appendix for details.)

Lemma 3. *For each cloud C we have $|A_r^C| \leq 37|R_d^* \cap C|/12 - 1$.*

Green Relays and Cloud Clusters. At any stage of the algorithm, we say that a set of clouds is *interconnected* if, with the current placement of relays, the sensors in the clouds can communicate with each other. Now, when all clouds have been stitched (so that the sensors within any one cloud can communicate), we proceed to interconnecting the clouds. First we greedily form the collection of cloud *clusters* (interconnected clouds) as follows. We start from assigning each cloud to its own cluster. Whenever it is possible to interconnect two clusters by placing one relay within each of the two clusters, we do so. These two relays are coloured green. After it is no longer possible to interconnect 2 clusters by placing just 2 relays, we repeatedly place 4 green relays wherever we can use them to interconnect clouds from 3 different clusters. Finally, we repeat this for 6 green relays which interconnect 4 clusters.

On average, we put 2 green relays every time the number of connected components in the communication graph on sensors plus relays decreases by one. Thus, the total number of green relays placed so far is twice the number of clouds that have been interconnected into the clusters.

Interconnecting the Clusters. Now, when the sensors in each cloud and the clouds in each cluster are interconnected, we interconnect the clusters by MSFN. We find MSFN on the clusters and place relays along edges of the forest just as we did in the simple algorithm from the previous section. This time though we assign colours to the relays. Specifically, for each edge e of the forest, we place 2 green relays at the endpoints of e, and $\lfloor |e|/r \rfloor$ yellow relays every r units starting from one of the endpoints (and when we find MSFN, we minimise the total number of yellow relays that we need). As with interconnecting clouds into the clusters, when interconnecting the clusters we use 2 green relays each time the number of connected components of the communication graph decreases by one. Thus, overall, we use at most $2|\mathcal{C}|$ green relays.

Analysis: Red and Green Relays. Recall that for $i < k$, $\mathcal{C}^i$ is the class of clouds that require precisely i relays for stabbing, and $\mathcal{C}^{k+}$ is the class of clouds that need at least k relays for stabbing. An optimal solution R^* therefore contains at least $|R^*_d| \geq k|\mathcal{C}^{k+}| + \sum_{i=1}^{k-1} i|\mathcal{C}^i|$ dark relays (relays inside clouds, i.e., relays within reach of sensors). Furthermore, $|R^*_d \cap C| \geq 1$ for all C.

Our algorithm places at most $2i-1$ red relays per cloud in $\mathcal{C}^i$, and not more than $37/12|R^*_d \cap C| - 1$ red relays per cloud in $\mathcal{C}^{k+}$. Including a total of $2|\mathcal{C}|$ green relays used for clouds interconnections, we get

$$\begin{aligned} |A_r| + |A_g| &\leq \textstyle\sum_{C \in \mathcal{C}^{k+}} (37|R^*_d \cap C|/12 - 1) + \sum_{i=1}^{k-1} (2i-1)|\mathcal{C}^i| + 2|\mathcal{C}| \\ &\leq 37(|R^*_d| - \textstyle\sum_{i=1}^{k-1} i|\mathcal{C}^i|)/12 + |\mathcal{C}^{k+}| + \sum_{i=1}^{k-1} (2i+1)|\mathcal{C}^i| \\ &\leq 37|R^*_d|/12 + |\mathcal{C}^{k+}| \ < \ (3.084 + 1/k)|R^*_d|. \end{aligned}$$

Analysis: Yellow Relays. As in the previous section, let $\mathcal{R}$ be the communication graph on the optimal set R^* of relays (without sensors). In $\mathcal{R}$ there exists a forest $\mathcal{R}'$ which makes the clusters interconnected. Let $R' \subset R^*$ be the relays that are vertices of $\mathcal{R}'$. We partition R' into "black" relays $R_b^* = R' \cap R_d^*$ and "white" relays $R_w^* = R' \cap R_\ell^*$ – those inside and outside the clusters.

Two black relays cannot be adjacent in $\mathcal{R}'$: if they are in the same cluster, the edge between them is redundant, if they are in different clusters, the distance between them must be larger than r as otherwise our algorithm would have placed two green relays to interconnect the clusters into one. By a similar reasoning, there cannot be a white relay adjacent to 3 or more black relays in $\mathcal{R}'$, and there cannot be a pair of adjacent white relays such that both of them are adjacent to 2 black relays. Finally, the maximum degree of a white relay is 5. Using these observations, we can prove the following lemma (see the Appendix).

Lemma 4. *There is a spanning forest with neighbourhoods on cloud clusters that requires at most* $(4/\sqrt{3} + 4/5)|R_w^*| < 3.11|R_w^*|$ *yellow relays on its edges.*

Then it follows that $|A_y| < 3.11|R_w^*| \leq 3.11|R_\ell^*|$. This completes the proof that the approximation ratio of our algorithm is less than 3.11.

5 Inapproximability of One-Tier Relay Placement

We have improved the best known approximation ratio for one-tier relay placement from 7 to 3.11. A natural question to pose at this point is whether we could make the approximation ratio as close to 1 as we wish. In this section, we show that no PTAS exists, unless P = NP. We prove the following theorem.

Theorem 1. *It is NP-hard to approximate one-tier relay placement within factor* $1 + 1/687$.

We present a reduction from minimum vertex cover in bounded-degree graphs. Let $\mathcal{G} = (V, E)$ be an instance of vertex cover; let $\Delta \leq 5$ be the maximum degree of $\mathcal{G}$. We construct an instance $\mathcal{I}$ of the relay placement problem as follows.

Fig. 2 illustrates the construction. Fig. 2a shows the *vertex gadget*; we have one such gadget for each vertex $v \in V$. Fig. 2b shows the *crossover gadget*; we have one such gadget for each edge $e \in E$. Small dots are sensors in the relay placement instance; each solid edge has length at most 1. White boxes are *good locations* for relays; dashed lines show connections for relays in good locations.

We set $r = 16(|V| + 1)$, and we choose $|E| + 1$ disks of diameter r such that each pair of these disks is separated by a distance larger than $|V|r$ but at most $\text{poly}(|V|)$. One of the disks is called $S(0)$ and the rest are $S(e)$ for $e \in E$. All vertex gadgets and one isolated sensor, called p_0, are placed within disk $S(0)$. The crossover gadget for edge e is placed within disk $S(e)$. There are noncrossing paths of sensors that connect the crossover gadget $e = \{u, v\} \in E$ to the vertex gadgets u and v; all such paths (*tentacles*) are separated by a distance at least 3. Good relay locations and p_0 cannot be closer than 1 unit to a disk boundary.

Fig. 2c is a schematic illustration of the overall construction in the case of $\mathcal{G} = K_5$; the figure is highly condensed in x direction. There are 11 disks. Disk $S(0)$ contains one isolated sensor and 5 vertex gadgets. Each disk $S(e)$ contains one crossover gadget. Outside these disks we have only parts of tentacles.

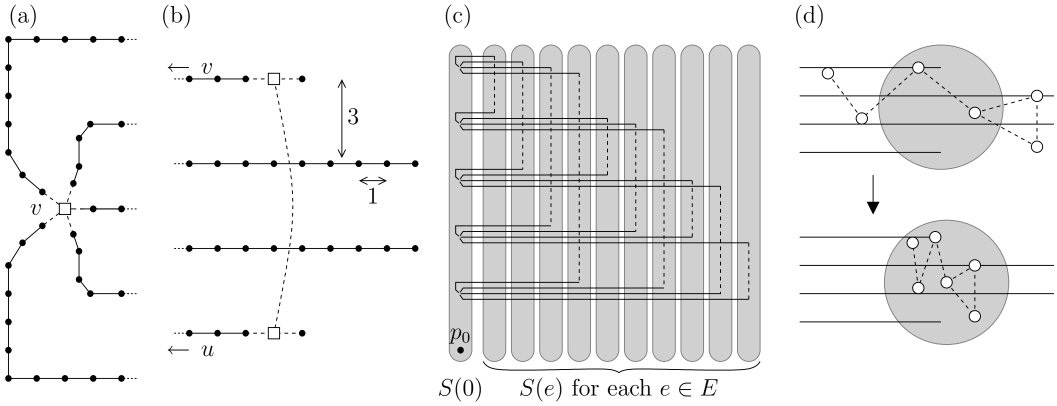

Fig. 2. (a) Vertex gadget for $v \in V$. (b) Crossover gadget for $\{v, u\} \in E$. (c) Reduction for K_5. (d) Normalising a solution, step 1.

There are $4|E| + 1$ blobs in $\mathcal{I}$. The isolated sensor p_0 forms one blob. For each edge there are 4 blobs: two tentacles from vertex gadgets to the crossover gadget, and two isolated sensors in the crossover gadget.

Theorem 1 now follows from the following two lemmata and inapproximability of vertex cover in graphs with maximum degree 4 [14, Thm. 3]; see the Appendix for the full proofs.

Lemma 5. *Let C be a vertex cover of $\mathcal{G}$. Then there is a feasible solution to relay placement problem $\mathcal{I}$ with $|C| + 2|E| + 1$ relays.*

Lemma 6. *Assume that there exists a feasible solution to relay placement problem $\mathcal{I}$ with $k + 2|E| + 1$ relays. Then $\mathcal{G}$ has a vertex cover of size at most k.*

Remark 1. We remind that throughout this work we assume that radius r is part of the problem instance. Our proof of Theorem 1 heavily relies on this fact; in our reduction, $r = \Theta(|V|)$. It is an open question whether one-tier relay placement admits a PTAS for a small, e.g., constant, r.

6 A PTAS for Two-Tier Relay Placement

In the previous sections we studied one-tier relay placement, in which the sensors could communicate with each other, as well as with the relays. We gave a 3.11-approximation algorithm, and showed that the problem admits no PTAS (for general r). In this section we turn to the two-tier version, in which the sensors cannot communicate with each other, but only with relays. We give a PTAS for this version of the problem.

We first give a structural lemma showing that an optimal set of relays, R^*, can be replaced by a set of at most $(1+\varepsilon)|R^*|$ relays that lie in a polynomial-size set G, or that lie along a straight segment joining two points of G, with relays spaced at distance r along the segment (except, possibly, the last relay of the segment). Then, we utilise the m-guillotine structural theorem [15] to prove that an optimal tree T^* having $|R^*|$ relays can be converted to a tree, T_m, that is m-*guillotine*, spans V, and utilises at most $(1 + O(1/m))|R^*|$ relays, where we pick $m = \Theta(1/\varepsilon)$. A dynamic programming algorithm allows us to compute an optimal (minimum-relay) tree among all m-guillotine spanning trees.

Edges of T^* are of two types: (i) "red" edges of length at most 1, which join a sensor to a relay, and (ii) "blue" edges of length at most r, which join two relays. The proof of the following lemma is in the Appendix.

Lemma 7. *Given a set V of n sensors and any fixed $\varepsilon > 0$, one can compute, in polynomial time, a set G of* $\text{poly}(n)$ *candidate relay positions such that there exists a Steiner tree T spanning V with the following properties: (i) the Steiner points are points of G, or are spaced at distance r along a line segment joining two points of G, (ii) each edge of T that is incident to a sensor has length at most 1, and all other edges have length at most r, and (iii) there are at most $(1 + \varepsilon)|R^*|$ Steiner points in T.*

Consider a set A of segments (edges) with endpoints among the set $V \cup G$. A *window* is a rectangle whose coordinates are among the x- and y-coordinates of the points $V \cup G$. A *cut* is a horizontal or vertical line ℓ, through a point of $V \cup G$, that intersects $\text{int}(W)$, the interior of window W. The intersection, $\ell \cap (A \cap \text{int}(W))$, of a cut ℓ with $A \cap \text{int}(W)$ consists of a discrete (possibly empty) set of line segments (edges of A) and crossing points (where an edge of A crosses ℓ). Let the segment endpoints/crossing points be denoted by $p_1, \ldots, p_\xi$, in order along ℓ. For a positive integer m, we define the m-*span*, $\sigma_m(\ell)$, of ℓ (with respect to W) to be the empty set, if $\xi \leq 2(m-1)$, and to be the (possibly zero-length) line segment $p_m p_{\xi-m+1}$, otherwise. A cut ℓ is m-*perfect with respect to* W if $\sigma_m(\ell) \subseteq A$.

We say that a set A of edges is m-*guillotine with respect to window* W if either (1) $A \cap \text{int}(W) = \emptyset$; or (2) there exists an m-perfect cut, ℓ, with respect to W, such that A is m-guillotine with respect to windows $W \cap H^+$ *and* $W \cap H^-$, where H^+, H^- are the closed halfplanes defined by ℓ. We say that A is m-*guillotine* if A is m-guillotine with respect to the axis-aligned bounding box of V. In the Appendix we sketch the proof of the following lemma:

Lemma 8. *Let T^* be an optimal two-tier Steiner spanning tree for V, having $|R^*|$ relays (Steiner points). Then, for any positive integer m, there exists a bounded edge length spanning tree T_m for V whose blue edges (joining pairs of relays, each of length $\leq r$) is m-guillotine, having at most $(1 + \varepsilon)|R^*|$ relays, each of which lies on a polynomial-size grid G.*

Our PTAS uses dynamic programming to search for an optimal m-guillotine bounded edge length spanning tree with Steiner points on the polynomial-size grid G. A subproblem specifies a rectangle, a set of $O(m)$ segments (with endpoints in G) crossing its boundary, at most 4 bridges (each with a regular pattern of Steiner points $R_1(\lambda) \cup R_2(\lambda)$), and a set of connections among boundary-crossing segments that are required within the rectangle; the objective is to build a minimum-relay m-guillotine spanning tree, spanning all sensors within the rectangle, while obeying the connection pattern and the boundary conditions. We defer the details to the full paper. In conclusion, we have sketched the proof of the following main result:

Theorem 2. *There is a PTAS for two-tier relay placement.*

7 Discussion

In Section 3 we presented a simple $O(n \log n)$-time 6.73-approximation algorithm for the one-tier relay placement. If one is willing to spend more time finding the approximation to the set cover, one may use the semi-local optimisation framework of Duh and Furer [16], which provides an approximation ratio of $1 + 1/2 + 1/3 + 1/4 + 1/5 - 1/2$ for the set cover with at most 5 elements per set; hence we obtain a 5.73-approximation.

One can form a bipartite graph on the blobs and candidate stab locations as follows. Pick a point within each maximal-depth cell of the arrangement of the blobs (maximal w.r.t the blobs that contain the cell); call these points "red". Pick a point within each blob; call these points "blue". Connect each blue point to the red points contained in the blob, represented by the blue point. It is possible to pick the points so that the bipartite graph on the points is planar. Then the stab placement is equivalent to the Planar Red/Blue Dominating Set Problem [17] – find fewest red vertices that dominate all blue ones. We believe that the techniques of Baker [18] can be used to give a PTAS for the problem. Combined with the simple algorithm in Section 3, this would result in a 4.16-approximation for the relay placement.

A more involved geometric argument allows us to improve the analysis of yellow relays in Section 4. We can bring the constant 3.11 in Lemma 4 down to 3, improving the approximation factor to 3.09. Combining this with the possible PTAS for the Planar Red/Blue Dominating Set would yield an approximation factor of $3 + \varepsilon$. We believe that a different, integrated method would be needed for getting below 3: various steps in our estimates are tight with respect to 3.

Acknowledgments. We thank Guoliang Xue for suggesting the problem to us and for fruitful discussions, and Marja Hassinen for comments and discussions. Parts of this research were conducted at the Dagstuhl research center.

References

1. Chen, D., Du, D.Z., Hu, X.D., Lin, G.H., Wang, L., Xue, G.: Approximations for Steiner trees with minimum number of Steiner points. Journal of Global Optimization **18**(1) (2000) 17–33
2. Chen, D., Du, D.Z., Hu, X.D., Lin, G.H., Wang, L., Xue, G.: Approximations for Steiner trees with minimum number of Steiner points. Theoretical Computer Science **262**(1–2) (2001) 83–99

3. Cheng, X., Du, D.Z., Wang, L., Xu, B.: Relay sensor placement in wireless sensor networks. Wireless Networks (2007) To appear.
4. Liu, H., Wan, P.J., Jia, X.: Fault-tolerant relay node placement in wireless sensor networks. In: Proc. 11th Annual International Computing and Combinatorics Conference (COCOON, Kunming, China, August 2005). Volume 3595 of Lecture Notes in Computer Science., Berlin, Germany, Springer-Verlag (2005) 230–239
5. Lloyd, E.L., Xue, G.: Relay node placement in wireless sensor networks. IEEE Transactions on Computers **56**(1) (2007) 134–138
6. Srinivas, A., Zussman, G., Modiano, E.: Mobile backbone networks – construction and maintenance. In: Proc. 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc, Florence, Italy, May 2006), New York, NY, USA, ACM Press (2006) 166–177
7. Zhang, W., Xue, G., Misra, S.: Fault-tolerant relay node placement in wireless sensor networks: Problems and algorithms. In: Proc. 26th IEEE International Conference on Computer Communications (INFOCOM, Anchorage, Alaska, USA, May 2007), Piscataway, NJ, USA, IEEE (2007) 1649–1657
8. Bredin, J.L., Demaine, E.D., Hajiaghayi, M., Rus, D.: Deploying sensor networks with guaranteed capacity and fault tolerance. In: Proc. 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc, Urbana-Champaign, IL, USA, May 2005), New York, NY, USA, ACM Press (2005) 309–319
9. Cerdeira, J.O., Pinto, L.S.: Requiring connectivity in the set covering problem. Journal of Combinatorial Optimization **9**(1) (2005) 35–47
10. Shuai, T.P., Hu, X.D.: Connected set cover problem and its applications. In: Proc. 2nd International Conference on Algorithmic Aspects in Information and Management (AAIM, Hong Kong, China, June 2006). Volume 4041 of Lecture Notes in Computer Science., Berlin, Germany, Springer-Verlag (2006) 243–254
11. Yang, Y., Lin, M., Xu, J., Xie, Y.: Minimum spanning tree with neighborhoods. In: Algorithmic Aspects in Information and Management. Volume 4508 of Lecture Notes in Computer Science., Berlin, Germany, Springer-Verlag (2007) 306–316
12. Du, D.Z., Hwang, F.: An approach for proving lower bounds: Solution of Gilbert–Pollak's conjecture on Steiner ratio. In: Proc. 31st Annual Symposium on Foundations of Computer Science (FOCS, St. Louis, MO, USA, October 1990), Piscataway, NJ, USA, IEEE (1990) 76–85
13. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. 2nd edn. The MIT Press, Cambridge, MA, USA (2001)
14. Berman, P., Karpinski, M.: On some tighter inapproximability results. In: Proc. 26th International Colloquium on Automata, Languages and Programming (ICALP, Prague, Czech Republic, July 1999). Volume 1644 of Lecture Notes in Computer Science., Berlin, Germany, Springer-Verlag (1999) 200–209
15. Mitchell, J.S.B.: Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, k-MST, and related problems. SIAM Journal on Computing **28**(4) (1999) 1298–1309
16. Duh, R.C., Fürer, M.: Approximation of k-set cover by semi-local optimization. In: Proc. 29th Annual ACM Symposium on Theory of Computing (STOC, El Paso, TX, USA, May 1997), New York, NY, USA, ACM Press (1997) 256–264
17. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Springer (1999)
18. Baker, B.S.: Approximation algorithms for NP-complete problems on planar graphs. Journal of the ACM **41**(1) (1994) 153–180

A Appendix

Proof of Lemma 3. For each $B \in \mathcal{B}_C \setminus \{B_0\}$, define the weight $w(B) = 1/|\mathcal{S}_j|$, where $\mathcal{S}_j$ is the unique set for which $B \in \mathcal{S}_j$. We also set $w(B_0) = 1$. We have

$$\textstyle\sum_{B \in \mathcal{B}_C} w(B) \;=\; |A_r^C| + 1. \tag{4}$$

Consider a relay $z \in R_d^* \cap C$, and find the smallest ℓ with $\mathcal{T}_\ell \cap \mathcal{B}(z) \neq \emptyset$, that is, $\ell = 1$ if $B_0 \in \mathcal{B}(z)$, and otherwise $y_{\ell-1}$ is the first relay that pierced a blob from $\mathcal{B}(z)$. Partition the set $\mathcal{B}(z)$ into $\mathcal{U}(z) = \mathcal{T}_\ell \cap \mathcal{B}(z)$ and $\mathcal{V}(z) = \mathcal{B}(z) \setminus \mathcal{U}(z)$. Note that $\mathcal{V}(z)$ may be empty, e.g., if $y_{\ell-1} = z$.

First, we show that $\sum_{B \in \mathcal{U}(z)} w(B) \leq 1$. We need to consider two cases. It may happen that $\ell = 1$, which means that $B_0 \in \mathcal{B}(z)$ and $\mathcal{U}(z) = \{B_0\}$. Then the total weight assigned to the blobs in $\mathcal{U}(z)$ is, by definition, 1. Otherwise $\ell > 1$ and $\mathcal{U}(z) \subseteq S_{\ell-1}$, implying $w(B) = 1/|S_{\ell-1}| \leq 1/|\mathcal{U}(z)|$ for each $B \in \mathcal{U}(z)$, and the claim follows.

Second, we show that $\sum_{B \in \mathcal{V}(z)} w(B) \leq 1/|\mathcal{V}(z)| + 1/(|\mathcal{V}(z)| - 1) + \cdots + 1/1$. At iterations $j \geq \ell$, the algorithm is able to consider placing the relay y_j at the location z. Therefore $|\mathcal{S}_j| \geq |\mathcal{S}(\mathcal{T}_j, z)|$. Furthermore, $\mathcal{S}(\mathcal{T}_j, z) \setminus \mathcal{S}(\mathcal{T}_{j+1}, z) = \mathcal{B}(z) \cap \mathcal{S}_j = \mathcal{V}(z) \cap \mathcal{S}_j$. Whenever placing the relay y_j makes $|\mathcal{S}(\mathcal{T}_j, z)|$ decrease by k, exactly k blobs of $\mathcal{V}(z)$ get connected to $\mathcal{T}_j$. Each of them is assigned the weight $w(C) \leq 1/|\mathcal{S}(\mathcal{T}_j, z)|$. Thus, $\sum_{B \in \mathcal{V}(z)} w(B) \leq k_1/(k_1 + k_2 + \cdots + k_n) + k_2/(k_2 + k_3 + \cdots + k_n) + \cdots + k_n/k_n$, where $k_1, k_2, \ldots, k_n$ are the number of blobs from $\mathcal{V}(z)$ that are pierced at different iterations, $\sum_i k_i = |\mathcal{V}(z)|$. The maximum value of the sum is attained when $k_1 = k_2 = \cdots = k_n = 1$ (i.e., every time $|\mathcal{V}(z)|$ is decreased by 1, and there are $|\mathcal{V}(z)|$ summands), and the claim follows.

Finally, since $|\mathcal{B}(z)| \leq 5$, and $\mathcal{U}(z) \neq \emptyset$, we have $|\mathcal{V}(z)| \leq 4$. Thus,

$$W(z) = \textstyle\sum_{B \in \mathcal{U}(z)} w(B) + \sum_{B \in \mathcal{V}(z)} w(B) \leq 1 + \dfrac{1}{4} + \dfrac{1}{3} + \dfrac{1}{2} + \dfrac{1}{1} = \dfrac{37}{12}. \tag{5}$$

The sets $\mathcal{B}(z)$, $z \in R_d^* \cap C$, form a cover of $\mathcal{B}_C$. Therefore, from (4) and (5),

$$\frac{37}{12}|R_d^* \cap C| \geq \sum_{z \in R_d^* \cap C} W(z) \geq \sum_{B \in \mathcal{B}_C} w(B) = |A_r^C| + 1.$$

□

Proof of Lemma 4. Let $\mathcal{D}$ be the set of cloud clusters. We partition $\mathcal{R}'$ into edge-disjoint trees induced by maximal connected subsets of white relays and their adjacent black relays. It is enough to show that for each such tree T which interconnects a subset of clusters $\mathcal{D}' \subseteq \mathcal{D}$, there is a spanning forest on $\mathcal{D}'$ such that the number of yellow relays on its edges is at most 3.11 times the number of white relays in T. As no pair of black relays is adjacent in $\mathcal{R}'$, these edge-disjoint trees interconnect all clusters in $\mathcal{D}$. The same holds for the spanning forests, and the lemma follows.

Trees with only one white relay (and thus exactly two black relays) are trivial: the spanning forest needs only one edge with one yellow relay (and one green in each end). Therefore assume that T contains at least two white relays.

We introduce yet another colour. For each white relay with two black neighbours, arbitrarily choose one of the black relays and change it into a "grey" relay. Let w be the number of white relays, let b be the number of remaining black relays, and let g be the number of grey relays in T.

First, we clearly have $b \leq w$. Second, there is no grey–white–white–grey path, each white relay is adjacent to another white relay, and the maximum degree of a white relay is 5 (geometry). Therefore the ratio $(b+g)/w$ is at most $9/5$. To see this, let w_2 be the number of white relays with a grey and a black neighbour, let w_1 be the number of white relays with a black neighbour but no grey neighbour, and let w_0 be the number of white relays without a black neighbour. By degree bound, $w_2 \leq 4w_1 + 5w_0 = 4w_1 + 5(w - w_2 - w_1)$; therefore $5w \geq 6w_2 + w_1$. And we know that, $w \geq w_2 + w_1$. Therefore $(9/5)w \geq (1/5)(6w_2 + w_1) + (4/5)(w_2 + w_1) = (w_2 + w_1) + w_2 = b + g$. (The worst case is a star of $1 + 4$ white relays, 5 black relays and 4 grey relays.)

Now consider the subtree induced by the black and white relays. It has fewer than $b + w$ edges, and the edge length is at most r. By Lemma 2, there is a spanning forest on the black relays with total length less than $(2/\sqrt{3})(b + w)r$; thus we need fewer than $(2/\sqrt{3})(b + w)$ yellow relays on the edges.

Now each pair of black relays in T is connected. It is enough to connect each grey relay to the nearest black relay: the distance is at most 2, and one yellow relay is enough. In summary, the total number of yellow relays is less than $(2/\sqrt{3})(b + w) + g \leq (2/\sqrt{3} - 1)2w + (14/5)w = (4/\sqrt{3} + 4/5)w < 3.11w$. □

Proof of Lemma 5. For each $v \in C$, place one relay at the good location of the vertex gadget v. For each $e \in E$, place two relays at the good locations of the crossover gadget e. Place one relay at the isolated sensor p_0. □

Proof of Lemma 6. If $k \geq |V|$, then the claim is trivial: $C = V$ is a vertex cover of size at most k. We therefore focus on the case $k < |V|$.

Let R be a solution with $k + 2|E| + 1$ relays. We transform the solution into a canonical form R' of the same size and with the following additional constraints: there is a subset $C \subseteq V$ such that at least one relay is placed at the good relay location of each vertex gadget $v \in C$; two relays are placed at the good locations of each crossover gadget; one relay is placed at p_0; and there are no other relays. If R' is a feasible solution, then C is a vertex cover of $\mathcal{G}$ with $|C| \leq k$.

Now we show how to construct the canonical form R'. We observe that there are $2|E| + 1$ isolated sensors in $\mathcal{I}$: sensor p_0 and two sensors for each crossover gadget. In the feasible solution R, for each isolated sensor p, we can always identify one relay within distance 1 from p (if there are several relays, pick one arbitrarily). These relays are called *bound relays*. The remaining $k < |V|$ relays are called *free relays*.

Step 1. Consider the communication graph formed by the sensors in $\mathcal{I}$ and the relays R. Since each pair of disks $S(i)$, $i \in \{0\} \cup E$, is separated by a distance larger than $|V|r$, we know that there is no path that extends from one disk to another and consists of at most k free relays (and possibly one bound relay in each end). Therefore we can shift each connected set of relays so that it is located within one disk (see Fig. 2d). While doing so, we do not break any relay–relay links: all relays within the same disk can communicate with each other. We can also maintain each relay–blob link intact.

Step 2. Now we have a clique formed by a set of relays within each disk $S(i)$, there are no other relays, and the network is connected. We move the bound relay in $S(0)$ so that it is located exactly on p_0. For each $e \in E$, we move the bound relays in $S(e)$ so that they are located exactly on the good relay locations. Finally, any free relays in $S(0)$ can be moved to a good relay location of a suitable vertex gadget. These changes may introduce new relay–blob links but they do not break any existing relay–blob or relay–relay links.

Step 3. What remains is that some disks $S(e)$, $e \in E$, may contain free relays. Let x be one of these relays. If x can be removed without breaking connectivity, we can move x to the good relay location of any vertex gadget. Otherwise x is adjacent to exactly one blob of sensors, and removing it breaks the network into two connected components: component A which contains p_0, and component B. Now we simply pick a vertex $v \in V$ such that the vertex gadget v contains sensors from component B, and we move x to the good relay location of this vertex gadget; this ensures connectivity between p_0 and B. □

Proof of Theorem 1. Let $\Delta, A, B, C \in \mathbb{N}$, with $\Delta \leq 5$ and $C > B$. Assume that there is a factor $\alpha = 1 + (C - B)/(B + \Delta A + 1)$ approximation algorithm $\mathcal{A}$ for relay placement. We show how to use $\mathcal{A}$ to solve the following *gap-vertex-cover* problem for some $0 < \epsilon < 1/2$: given a graph $\mathcal{G}$ with An nodes and maximum degree Δ, decide whether the minimum vertex cover of $\mathcal{G}$ is smaller than $(B + \epsilon)n$ or larger than $(C - \epsilon)n$.

If $n < 2$, the claim is trivial. Otherwise we can choose a positive constant ϵ such that $\alpha - 1 < (C - B - 2\epsilon)/(B + \epsilon + \Delta A + 1/n)$ for any $n \geq 2$. Construct the relay placement instance $\mathcal{I}$ as described above.

If minimum vertex cover of $\mathcal{G}$ is smaller than $(B + \epsilon)n$, then by Lemma 5, the algorithm $\mathcal{A}$ returns a solution with at most $b = \alpha((B + \epsilon)n + 2|E| + 1)$ relays. If minimum vertex cover of $\mathcal{G}$ is larger than $(C - \epsilon)n$, then by Lemma 6, the algorithm $\mathcal{A}$ returns a solution with at least $c = (C - \epsilon)n + 2|E| + 1$ relays. As $2|E| \leq \Delta An$, we have $c - b \geq (C - \epsilon)n + 2|E| + 1 - \alpha((B + \epsilon)n + 2|E| + 1) \geq (C - B - 2\epsilon - (\alpha - 1)(B + \epsilon + \Delta A + 1/n))n > 0$, which shows that we can solve the gap-vertex-cover problem in polynomial time.

For $\Delta = 4$, $A = 152$, $B = 78$, $C = 79$, and any $0 < \epsilon < 1/2$, the gap-vertex-cover problem is NP-hard [14, Thm. 3]. □

Proof of Lemma 7. *(Sketch.)* Fix a positive integer $m = \Theta(1/\varepsilon)$. Let T^* be an optimal tree, with $|R^*|$ relays, spanning V.

We know that in T^* each relay is of degree at most 5, and there are at most $n - 2$ relays of degree 3, 4 or 5. Relays of degree 2 in T^* form *relay-paths* in T^*. These relay-paths are potentially quite long (not bounded in terms of $n = |V|$). We can assume, w.l.o.g., that relay-paths consist of relays placed along a straight line segment, with relays evenly spaced at distance r along the segment (except that the last spacing along the segment may be $< r$).

Let R be a maximal subset of degree-≥ 3 relays in T^* such that every pair of relays of R is separated by distance (within T^*) at least $\Theta(m)$. Let $T^*_R \subseteq T^*$ be the minimal spanning tree of R within T^*; T^*_R interconnects nodes of R with paths of $\Theta(m)$ relays. Consider the polynomial-size regular square grid of spacing $\varepsilon \cdot \text{diam}(V)/n$ within the bounding box of V; we define the set G to include these grid points. Shifting the relays R to this grid adds Euclidean length at most $\varepsilon \cdot \text{diam}(V)$ to T^*_R; the total number of new relays needed to accommodate this change is bounded by $\varepsilon \cdot \text{diam}(S)/r$ plus $|R| - 1$ (since an additional relay may be needed on each interconnection path in T^*_R, due to rounding Euclidean lengths up to the nearest multiple of r). Since the $|R| - 1$ interconnection paths in T^*_R are each of length $\Theta(m)$, we see that $|R| - 1$ is a small fraction $(O(1/m))$ of the number of relays in T^*_R, so the total increase in the number of relays in T^* caused by this shifting is $O(\varepsilon \cdot |R^*|)$.

From the bounded degree property of T^*, we can show that there is a set E' of $O(|R^*|/m)$ (blue) edges whose removal breaks T^* into subtrees, each having $O(m)$ relays. We will (possibly) perturb the relays in each subtree, while not increasing their number in each subtree and while keeping all red edges of length at most 1 and blue edges of length at most r. This perturbation may, however, increase slightly the lengths of the edges E', causing us to have to add at most one more relay per edge of E'; since $|E'| = O(|R^*|/m)$, this addition can be afforded.

There are two types of subtrees: those consisting only of relays (Steiner points), and those having at least one sensor. Consider first those having at least one sensor. Then, we know that all relays of a subtree lie within distance $O(mr)$ of some sensor. The relay leaves of the subtree (at endpoints of edges in E') can be rounded to a regular grid of spacing $r/2$ centred on any one sensor; there are at most $O(m^2)$ such grid positions (per sensor), and rounding leaves to a grid of resolution $r/2$ will allow all edges of E' to be of length at most $2r$, so that at most $|E'|$ new relays are needed. The grid G is defined to include these grid positions. We now define an *iterated circle arrangement* as follows. First, we construct circles of radii $1, 1+r, 1+2r, 1+3r, \ldots, 1+mr$ centred at each sensor within the subtree and at each rounded leaf of the subtree. Then, in iteration 2, centred at each vertex (where two circles cross) of this arrangement, we construct circles of radii $r, 2r, \ldots, mr$. This continues for m iterations, resulting in an arrangement having a constant (function of $m = \Theta(1/\varepsilon)$) number of vertices. Now, we claim that the vertices of the iterated circle arrangement is a rich enough set of points that we can systematically perturb a subtree so that all of its relays lie at the vertices. (A subtree is a linkage, with upper bound constraints on link lengths; it has $O(m)$ degrees of freedom, so we can apply motion planning arguments to "pin" it.) We define the grid G to include the vertices of the iterated circle arrangement.

For subtrees consisting only of relays, there are two subcases: subtrees that are relay-paths (having only degree-2 relays), and all other subtrees. Subtrees that are relay-paths can be handled directly; in fact, we can consider separately the long relay-paths that consist of a maximal sequence of such subtrees, since

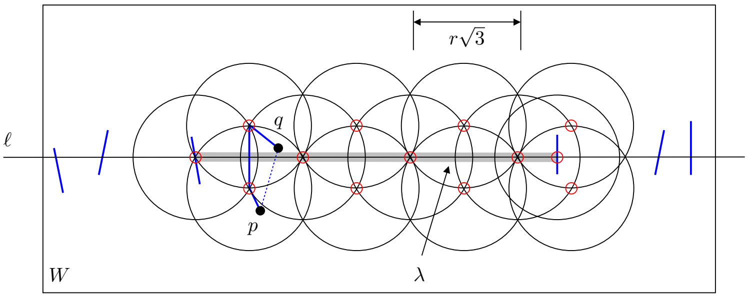

Fig. 3. Steiner points $R_1(\lambda)$ and $R_2(\lambda)$ along an m-span λ. Crossing blue edge pq is replaced by a path in the radius-r disk graph.

they will join points of G with a straight segment of relays spaced at distance r, once we round the subtrees that are not relay-paths. Thus, we consider a subtree τ consisting only of relays and having at least one relay, q, of degree ≥ 3. By adjusting the constants in the definition of the set R defined above, we can assume that there is at least one member of R in subtree τ; thus, q can be assumed to be at a grid point, allowing us to proceed as in the case of the subtree containing a sensor. □

Proof of Lemma 8. *(Sketch.)* The key lemma of [15] shows that the set of blue edges can be converted to an m-guillotine set by adding m-spans whose total length is at most $\sqrt{2}/m$ times the length of the original blue edges. (Further, the x- and y-coordinates of the cuts in the m-guillotine partition are from among the points of the polynomial-size grid G.) This implies that the added m-spans have total length $O(r|R^*|/m)$. Consider an m-span, λ, that was added, of length $|\lambda|$. We claim that we can place $O(|\lambda|/r)$ Steiner points along λ and modify the tree T^* so that no blue edges cross λ. We simply place $O(|\lambda|)$ Steiner (relay) points, $R_1(\lambda)$, along λ, at uniform spacing $r\sqrt{3}$, and also place an additional $O(|\lambda|)$ Steiner points, $R_2(\lambda)$, at the crossing points of radius-r circles centred at points R_1. See Fig. 3. (There is a technical issue that arises if λ is very short, since "rounding" up $|\lambda|$ introduces $O(1)$ additional Steiner points on each m-span. If $|\lambda| < \Theta(mr)$ then we show that $O(m)$ blue edges cross λ, so we can afford not to add the m-span and instead specify in the dynamic programming subproblems each of the $O(m)$ blue edges crossing the cut. If $|\lambda| > \Theta(mr)$, we can afford to add $O(1)$ Steiner points per m-span, charging them off to the length of the m-spans.) Clearly, the radius-r disk graph of $R_1(\lambda) \cup R_2(\lambda)$ is connected. Further, the union of radius-r disks centred at $R_1(\lambda) \cup R_2(\lambda)$ covers a strip of width $2r$ centred on λ. Thus, any blue edge pq of T^* that crosses λ can be replaced by a path of $O(1)$ edges in the radius-r disk graph of $\{p, q\} \cup R_1(\lambda) \cup R_2(\lambda)$, ensuring that the connectivity of T^* can be maintained using blue edges of length at most r, with no blue edges crossing λ. Thus, we have transformed T^* into a Steiner tree whose blue edges are m-guillotine, and we have done so by adding only $O(|R^*|/m)$ new Steiner points and keeping all edges of length at most r. □