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] Department of Computer Science, University of Arizona
alon@email.arizona.edu, poorna@email.arizona.edu
[2] Department of Computer Science, Braunschweig University of Technology
s.fekete@tu-bs.de
[3] Department of Applied Mathematics and Statistics, Stony Brook University
jsbm@ams.stonybrook.edu
[4] Helsinki Institute for Information Technology HIIT,
University of Helsinki and Helsinki University of Technology
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. 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. We present a 3.11-approximation algorithm, and show that the problem 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 have been considered in the literature [1,2,3,4,5,6,7,8]. 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.

D. Halperin and K. Mehlhorn (Eds.): ESA 2008, LNCS 5193, pp. 356–367, 2008.

Formally, the input to the relay placement problem is a set of 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 only* 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).

The current best approximation ratio of 7 for one-tier relay placement is due to Lloyd and Xue [5]. For the two-tier version, Lloyd and Xue [5] suggested a $(5+\varepsilon)$-approximation algorithm for arbitrary $r \geq 1$; Srinivas et al. [6] gave a $(4+\varepsilon)$-approximation for the case $r \geq 2$. In this paper, we present a polynomial-time 3.11-approximation algorithm for the one-tier relay placement, and show that it admits no PTAS unless P = NP (assuming that r is part of the input). In the full version, we will present a PTAS for the two-tier version; the PTAS works for arbitrary $r \geq 1$.

2 Blobs, Clouds, Stabs, Hubs, and Forests

For two points x, y in the plane let $|xy|$ be the Euclidean distance between them. 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.

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

Analogously, we define a *cloud* $C \in \mathcal{C}$ as the union of the unit disks centered 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

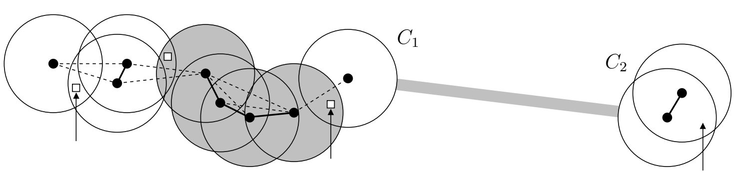

Fig. 1. Dots are sensors in V, solid lines are edges in E and F, and dashed lines are edges in F only. There are 5 blobs in $\mathcal{B}$ (one of them highlighted) and 2 clouds $C_1, C_2 \in \mathcal{C}$. Arrows are stabs, and small rectangles are hubs. The wide grey line is the only edge in MStFN($\mathcal{C}$),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

$C \in \mathcal{C}$ consists of one or more blobs $B \in \mathcal{B}$; we use $\mathcal{B}_C$ to denote the blobs that form the cloud C.

We define a *stab* to be a relay with an infinite communication range ($r = \infty$), and a *hub* as a relay without the ability to communicate with the other relays (thus hubs can enable communication within one cloud, but are of no use in communicating between clouds). As will be shown, a solution to stab or hub placement can be used as a 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. 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, the number of additional hubs needed is less than the number of 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. □

2.1 Steiner Forests and Spanning Forests with Neighbourhoods

Let $\mathcal{P}$ be a collection of planar sets; call them *neighbourhoods*. (In Section 3 the neighbourhoods 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 (since it is not necessary to connect neighbourhoods beyond their boundaries).

Consider a complete weighted graph whose vertices are the neighbourhoods in $\mathcal{P}$ and whose edge weights are the 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 $|\text{MSFN}(P)| \le (2/\sqrt{3})|\text{MStFN}(P)|$ 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* $\mathcal{P}$, $|\text{MSFN}(\mathcal{P})| \le (2/\sqrt{3})|\text{MStFN}(\mathcal{P})|$.

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. □

3 A 3.11-Approximation Algorithm

In this section we give a 3.11-approximation algorithm for one-tier relay placement. We focus on nontrivial instances with more than one blob.

Note that the *number* of relays in a solution may be exponential in the size of the input (number of bits). Our algorithm produces a succinct representation of the solution, 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.

3.1 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.

The 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.

3.2 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. (For any subset of i cells of the arrangement of unit disks centered on the sensors in C, we can consider placing the relays in the cells and check whether this stabs all blobs.) 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.

3.3 Stitching a Cloud from $\mathcal{C}^{k+}$

We focus on one cloud $C \in \mathcal{C}^{k+}$. For a point y in the plane, let $\mathcal{B}(y) = \{B \in \mathcal{B}_C : y \in B\}$ be the set of blobs that contain the point; obviously $|\mathcal{B}(y)| \le 5$ for any y. For 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 $A^C_r = \{y_j : j = 1, 2, \dots\}$, as follows:

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

$$\begin{aligned} y_j &\leftarrow \arg\max_y \{|\mathcal{S}(\mathcal{T}_j, y)| : \mathcal{B}(y) \cap \mathcal{T}_j \ne \emptyset\}, \\ \mathcal{S}_j &\leftarrow \mathcal{S}(\mathcal{T}_j, y_j), \\ \mathcal{T}_{j+1} &\leftarrow \mathcal{T}_j \cup \mathcal{S}_j, \\ j &\leftarrow j + 1. \end{aligned}$$

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 $\mathcal{S}(\mathcal{T}_j, y) \ne \emptyset$. Since 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 $|A^C_r| \le |\mathcal{B}_C| - 1$. The sets $\mathcal{S}_j$ form a partition of $\mathcal{B}_C \setminus \{B_0\}$.

We prove the following performance guarantee.

Lemma 3. *For each cloud C we have $|A^C_r| \le 37|R^*_d \cap C|/12 - 1$.*

Proof. 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

$$\sum_{B \in \mathcal{B}_C} w(B) = |A^C_r| + 1. \tag{1}$$

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

$$\sum_{B \in \mathcal{U}(z)} w(B) \le 1.$$

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 $w(B) = 1/|S_{\ell-1}| \le 1/|\mathcal{U}(z)|$ for each $B \in \mathcal{U}(z)$.

Second, we show that

$$\sum_{B \in \mathcal{V}(z)} w(B) \le \frac{1}{|\mathcal{V}(z)|} + \frac{1}{|\mathcal{V}(z)| - 1} + \cdots + \frac{1}{1}.$$

Indeed, at iterations $j \ge \ell$, the algorithm is able to consider placing the relay y_j at the location z. Therefore $|\mathcal{S}_j| \ge |\mathcal{S}(\mathcal{T}_j, z)|$. 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 $w(C) \le 1/|\mathcal{S}(\mathcal{T}_j, z)|$. Thus, $\sum_{B \in \mathcal{V}(z)} w(B) \le k_1/(k_1 + k_2 + \cdots + k_n) + k_2/(k_2 + k_3 + \cdots + k_n) + \cdots + k_n/k_n$, 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).

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

$$W(z) = \sum_{B \in \mathcal{U}(z)} w(B) + \sum_{B \in \mathcal{V}(z)} w(B) \le 1 + \frac{1}{4} + \frac{1}{3} + \frac{1}{2} + \frac{1}{1} = \frac{37}{12}. \qquad (2)$$

The sets $\mathcal{B}(z)$, $z \in R_d^* \cap C$, form a cover of $\mathcal{B}_C$. Therefore, from (1) and (2),

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

□

3.4 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 by 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 place 2 green relays every time the number of connected components in the communication graph on sensors plus relays decreases by one.

3.5 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. 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 $2|\mathcal{C}| - 2$ green relays.

3.6 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 $\mathcal{C}^{k+}$. Adding a total of $2|\mathcal{C}| - 2$ green relays used for clouds interconnections, we get

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

3.7 Analysis: Yellow Relays

Let $\mathcal{R}$ be the communication graph on the optimal set R^* of relays 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. In $\mathcal{R}$ there exists a forest $\mathcal{R}'$ that 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, resp.

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 each of them is adjacent to 2 black relays. Finally, the maximum degree of a white relay is 5. Using these observations, we can prove the following lemma.

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.*

Proof. 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 $5w \geq 6w_2 + w_1$. We also 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 $(2/\sqrt{3})(b + w) + g \leq (2/\sqrt{3} - 1)2w + (14/5)w = (4/\sqrt{3} + 4/5)w < 3.11w$. □

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.

4 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.

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

The reduction is from minimum vertex cover in graphs of bounded degree. Let $\mathcal{G} = (V, E)$ be an instance of vertex cover; let $\Delta \le 5$ be the maximum degree of $\mathcal{G}$. We construct an instance $\mathcal{I}$ of the relay placement problem which has a feasible solution with $|C| + 2|E| + 1$ relays if and only if $\mathcal{G}$ has a vertex cover of size k.

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.

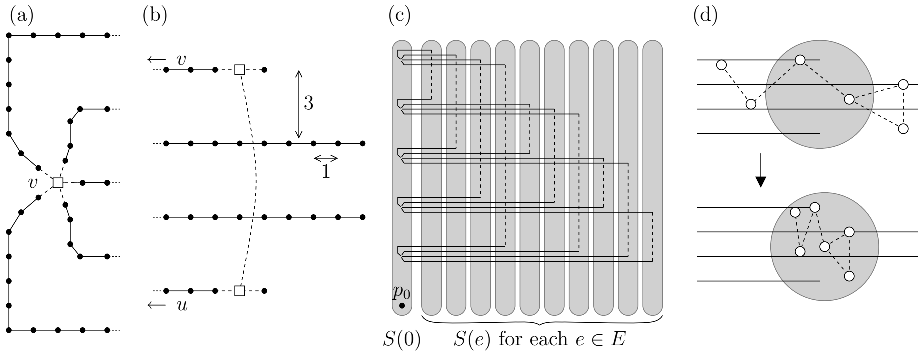

Fig. 2. (a) Vertex gadget for $v \in V$. (b) Crossover gadget for $\{v, u\} \in E$. (c) Reduction for K_5. (d) Normalising a solution, step 1.

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 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.

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.

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.*

Proof. 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. □

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.*

Proof. 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 $0 < \varepsilon < 1/2$: given a graph $\mathcal{G}$ with An nodes and maximum degree Δ, decide whether the minimum vertex cover of $\mathcal{G}$ is smaller than $(B+\varepsilon)n$ or larger than $(C - \varepsilon)n$.

If $n < 2$, the claim is trivial. Otherwise we can choose a positive constant ε such that $\alpha - 1 < (C - B - 2\varepsilon)/(B + \varepsilon + \Delta A + 1/n)$ 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 $(B + \varepsilon)n$, then by Lemma 5, the algorithm $\mathcal{A}$ returns a solution with at most $b = \alpha((B + \varepsilon)n + 2|E| + 1)$ relays. If minimum vertex cover of $\mathcal{G}$ is larger than $(C - \varepsilon)n$, then by Lemma 6, the algorithm $\mathcal{A}$ returns a solution with at least $c = (C - \varepsilon)n + 2|E| + 1$ relays. As $2|E| \leq \Delta An$, we have $c - b \geq (C - \varepsilon)n + 2|E| + 1 - \alpha((B + \varepsilon)n + 2|E| + 1) \geq (C - B - 2\varepsilon - (\alpha - 1)(B + \varepsilon + \Delta A + 1/n))n > 0$, 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 $0 < \varepsilon < 1/2$, the gap-vertex-cover problem is NP-hard [13, Thm. 3]. □

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.

Acknowledgments

We thank Guoliang Xue for suggesting the problem to us and for fruitful discussions, and Marja Hassinen for comments and discussions. We thank the anonymous referees for their helpful suggestions. Parts of this research were conducted at the Dagstuhl research center. AE is supported by NSF CAREER Grant 0348000. JM is partially supported by grants from the National Science Foundation (CCF-0431030, CCF-0528209, CCF-0729019), NASA Ames, and Metron Aviation. JS is supported in part by the Academy of Finland grant 116547, and Helsinki Graduate School in Computer Science and Engineering (Hecse). VP is supported in part by the Academy of Finland grant 118653 (ALGODAN).

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), 17–33 (2000)
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), 83–99 (2001)
3. Cheng, X., Du, D.Z., Wang, L., Xu, B.: Relay sensor placement in wireless sensor networks. Wireless Networks (to appear, 2007)
4. Liu, H., Wan, P.J., Jia, X.: Fault-tolerant relay node placement in wireless sensor networks. In: Wang, L. (ed.) COCOON 2005. LNCS, vol. 3595, pp. 230–239. Springer, Heidelberg (2005)
5. Lloyd, E.L., Xue, G.: Relay node placement in wireless sensor networks. IEEE Transactions on Computers 56(1), 134–138 (2007)
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, pp. 166–177. ACM Press, New York (2006)
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, pp. 1649–1657. IEEE, Piscataway (2007)
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, pp. 309–319. ACM Press, New York (2005)
9. Cerdeira, J.O., Pinto, L.S.: Requiring connectivity in the set covering problem. Journal of Combinatorial Optimization 9(1), 35–47 (2005)
10. Shuai, T.P., Hu, X.D.: Connected set cover problem and its applications. In: Cheng, S.-W., Poon, C.K. (eds.) AAIM 2006. LNCS, vol. 4041, pp. 243–254. Springer, Heidelberg (2006)
11. Yang, Y., Lin, M., Xu, J., Xie, Y.: Minimum spanning tree with neighborhoods. In: Kao, M.-Y., Li, X.-Y. (eds.) AAIM 2007. LNCS, vol. 4508, pp. 306–316. Springer, Heidelberg (2007)
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, pp. 76–85. IEEE, Piscataway (1990)
13. Berman, P., Karpinski, M.: On some tighter inapproximability results. In: Wiedermann, J., Van Emde Boas, P., Nielsen, M. (eds.) ICALP 1999. LNCS, vol. 1644, pp. 200–209. Springer, Heidelberg (1999)