The Homogeneous Broadcast Problem in Narrow and Wide Strips*

Mark de Berg[1], Hans L. Bodlaender[1,2], and Sándor Kisfaludi-Bak[1]

[1] TU Eindhoven, The Netherlands
m.t.d.berg@tue.nl; h.l.bodlaender@tue.nl; s.kisfaludi.bak@tue.nl
[2] Utrecht University, The Netherlands

Abstract. Let P be a set of nodes in a wireless network, where each node is modeled as a point in the plane, and let $s \in P$ be a given source node. Each node p can transmit information to all other nodes within unit distance, provided p is activated. The (homogeneous) broadcast problem is to activate a minimum number of nodes such that in the resulting directed communication graph, the source s can reach any other node. We study the complexity of the regular and the hop-bounded version of the problem (in the latter, s must be able to reach every node within a specified number of hops), with the restriction that all points lie inside a strip of width w. We almost completely characterize the complexity of both the regular and the hop-bounded versions as a function of the strip width w.

1 Introduction

Wireless networks give rise to a host of interesting algorithmic problems. In the traditional model of a wireless network each node is modeled as a point $p \in \mathbb{R}^2$, which is the center of a disk $\delta(p)$ whose radius equals the transmission range of p. Thus p can send a message to another node q if and only if $q \in \delta(p)$. Using a larger transmission radius may allow a node to transmit to more nodes, but it requires more power and is more expensive. This leads to so-called range-assignment problems, where the goal is to assign a transmission range to each node such that the resulting communication graph has desirable properties, while minimizing the cost of the assignment. We are interested in broadcast problems, where the desired property is that a given source node can reach any other node in the communication graph. Next, we define the problem more formally.

Let P be a set of n points in $\mathbb{R}^d$ and let $s \in P$ be a source node. A *range assignment* is a function $\rho : P \to \mathbb{R}_{\geqslant 0}$ that assigns a transmission range $\rho(p)$ to each point $p \in P$. Let $\mathcal{G}_\rho = (P, E_\rho)$ be the directed graph where $(p, q) \in E_\rho$ iff $|pq| \leqslant \rho(p)$. The function ρ is a *broadcast assignment* if every point $p \in P$ is reachable from s in $\mathcal{G}_\rho$. If every $p \in P$ is reachable within h hops, for a given parameter h, then ρ is an *h-hop broadcast assignment*. The (h-hop) broadcast

* This research was supported by the Netherlands Organization for Scientific Research (NWO) under project no. 024.002.003.

F. Ellen et al. (Eds.): WADS 2017, LNCS 10389, pp. 289–300, 2017.
DOI: 10.1007/978-3-319-62127-2_25

problem is to find an (h-hop) broadcast assignment whose cost $\sum_{p\in P} \mathrm{cost}(\rho(p))$ is minimized. Often the cost of assigning transmission radius x is defined as $\mathrm{cost}(x) = x^\alpha$ for some constant α. In $\mathbb{R}^1$, both the basic broadcast problem and the h-hop version are solvable in $O(n^2)$ time [7]. In $\mathbb{R}^2$ the problem is NP-hard for any $\alpha > 1$ [6,10], and in $\mathbb{R}^3$ it is even APX-hard [10]. There are also several approximation algorithms [1,6]. For the 2-hop broadcast problem in $\mathbb{R}^2$ an $O(n^7)$ algorithm is known [2] and for any constant h there is a PTAS [2]. Interestingly, the complexity of the 3-hop broadcast problem is unknown.

An important special case of the broadcast problem is where we allow only two possible transmission ranges for the points, $\rho(p) = 1$ or $\rho(p) = 0$. In this case the exact cost function is irrelevant and the problem becomes to minimize the number of active points. This is called the *homogeneous broadcast problem* and it is the version we focus on. From now on, all mentions of broadcast and h-hop broadcast refer to the homogeneous setting. Observe that if $\rho(p) = 1$ then (p, q) is an edge in $\mathcal{G}_\rho$ if and only if the disks of radius $1/2$ centered at p and q intersect. Hence, if all points are active then $\mathcal{G}_\rho$ in the intersection graph of a set of congruent disks or, in other words, a *unit-disk graph (UDG)*. Because of their relation to wireless networks, UDGs have been studied extensively.

Let $\mathcal{D}$ be a set of congruent disks in the plane, and let $\mathcal{G}_\mathcal{D}$ be the UDG induced by $\mathcal{D}$. A *broadcast tree* on $\mathcal{G}_\mathcal{D}$ is a rooted spanning tree of $\mathcal{G}_\mathcal{D}$. To send a message from the root to all other nodes, each internal node of the tree has to send the message to its children. Hence, the cost of broadcasting is related to the internal nodes in the broadcast tree. A cheapest broadcast tree corresponds to a minimum-size *connected dominating set* on $\mathcal{G}_\mathcal{D}$, that is, a minimum-size subset $\varDelta \subset \mathcal{D}$ such that the subgraph induced by $\varDelta$ is connected and each node in $\mathcal{G}_\mathcal{D}$ is either in $\varDelta$ or a neighbor of a node in $\varDelta$. The broadcast problem is thus equivalent to the following: given a UDG $\mathcal{G}_\mathcal{D}$ with a designated source node s, compute a minimum-size connected dominated set $\varDelta \subset \mathcal{D}$ such that $s \in \varDelta$.

In the following we denote the dominating set problem by DS, the connected dominating set problem by CDS, and we denote these problems on UDGs by DS-UDG and CDS-UDG, respectively. Given an algorithm for the broadcast problem, one can solve CDS-UDG by running the algorithm n times, once for each possible source point. Consequently, hardness results for CDS-UDG can be transferred to the broadcast problem, and algorithms for the broadcast problem can be transferred to CDS-UDG at the cost of an extra linear factor in the running time. It is well known that DS and CDS are NP-hard, even for planar graphs [11]. DS-UDG and CDS-UDG are also NP-hard [13,15]. The parameterized complexity of DS-UDG has also been investigated: Marx [14] proved that DS-UDG is W[1]-hard when parameterized by the size of the dominating set. (The definition of W[1] and other parameterized complexity classes can be found in the book by Flum and Grohe [9].)

Our contributions. Knowing the existing hardness results for the broadcast problem, we set out to investigate the following questions. Is there a natural special case or parameterization admitting an efficient algorithm? Since the broadcast problem is polynomially solvable in $\mathbb{R}^1$, we study how the complexity of the problem changes as we go from the 1-dimensional problem to the 2-dimensional problem. To do this, we assume the points (that is, the disk centers) lie in a strip of width w, and we study how the problem complexity changes as we increase w. We give an almost complete characterization of the complexity, both for the general and for the hop-bounded version of the problem. More precisely, our results are as follows.

We first study strips of width at most $\sqrt{3}/2$. Unit disk graphs restricted to such *narrow strips* are a subclass of co-comparability graphs [16], for which an $O(nm)$ time CDS algorithm is known [12,3]. (Here m denotes the number of edges in the graph.) The broadcast problem is slightly different because it requires s to be in the dominating set; still, one would expect better running times in this restricted graph class. Indeed, we show that for narrow strips the broadcast problem can be solved in $O(n \log n)$ time. The hop condition in the h-hop broadcast problem has not been studied yet for co-comparability graphs to our knowledge. This condition complicates the problem considerably. Nevertheless, we show that the h-hop broadcast problem in narrow strips is solvable in polynomial time. Our algorithm runs in $O(n^6)$ and uses a subroutine for 2-hop broadcast, which may be of independent interest: we show that the 2-hop broadcast problem is solvable in $O(n^4)$ time. Our subroutine is based on an algorithm by Ambühl et al. [2] for the non-homogeneous case, which runs in $O(n^7)$ time. This result is can be found in the full version of this paper.

Second, we investigate what happens for wider strips. We show that the broadcast problem has an $n^{O(w)}$ dynamic-programming algorithm for strips of width w. We prove a matching lower bound of $n^{\Omega(w)}$, conditional on the Exponential Time Hypothesis (ETH). Interestingly, the h-hop broadcast problem has no such algorithm (unless P = NP): we show this problem is already NP-hard on a strip of width 40. One of the gadgets in this intricate construction can also be used to prove that a CDS-UDG and the broadcast problem are W[1]-hard parameterized by the solution size k. The W[1]-hardness proof is discussed only in the full version. It is a reduction from GRID TILING based on ideas by Marx [14], and it implies that there is no $f(k)n^{o(\sqrt{k})}$ algorithm for CDS-UDG unless ETH fails.

2 Algorithms for broadcasting inside a narrow strip

In this section we present polynomial algorithms (both for broadcast and for h-hop broadcast) for inputs that lie inside a strip $\mathcal{S} := \mathbb{R} \times [0, w]$, where $0 < w \leqslant \sqrt{3}/2$ is the width of the strip. Without loss of generality, we assume that the source lies on the y-axis. Define $\mathcal{S}_{\geqslant 0} := [0, \infty) \times [0, w]$ and $\mathcal{S}_{\leqslant 0} := (-\infty, 0] \times [0, w]$.

Let P be the set of input points. We define $x(p)$ and $y(p)$ to be the x- and y-coordinate of a point $p \in P$, respectively, and $\delta(p)$ to be the unit-radius disk centered at p. Let $\mathcal{G} = (P, E)$ be the graph with $(p, q) \in E$ iff $q \in \delta(p)$, and let $P' := P \setminus \delta(s)$ be the set of input points outside the source disk. We say that a point $p \in P$ is *left-covering* if $pp' \in E$ for all $p' \in P'$ with $x(p') < x(p)$; p is *right-covering* if $p'p \in E$ for all $p' \in P'$ with $x(p') > x(p)$. We denote the set of left-covering and right-covering points by Q^- and Q^+ respectively. Finally, the *core area* of a point p, denoted by core(p), is $[x(p) - \frac{1}{2}, x(p) + \frac{1}{2}] \times [0, w]$. Note that core$(p) \subset \delta(p)$ because $w \leqslant \sqrt{3}/2$, i.e., the disk of p covers a part of the strip that has horizontal length at least one. This is a key property of strips of width at most $\sqrt{3}/2$, and will be used repeatedly.

We partition P into levels $L_0, L_1, \dots L_t$, based on hop distance from s in $\mathcal{G}$. Thus $L_i := \{p \in P : d_{\mathcal{G}}(s,p) = i\}$, where $d_{\mathcal{G}}(s,p)$ denotes the hop-distance. Let L_i^- and L_i^+ denote the points of L_i with negative and nonnegative coordinates, respectively. We will use the following observation multiple times.

Observation 1. *Let $\mathcal{G} = (P, E)$ be a unit disk graph on a narrow strip $\mathcal{S}$.*

(i) Let π be a path in $\mathcal{G}$ from a point $p \in P$ to a point $q \in P$. Then the region $[x(p) - \frac{1}{2}, x(q) + \frac{1}{2}] \times [0, w]$ is fully covered by the disks of the points in π.

(ii) The overlap of neighboring levels is at most $\frac{1}{2}$ in x-coordinates: $\max\{x(p) | p \in L_{i-1}^+\} \leqslant \min\{x(q) | q \in L_i^+\} + \frac{1}{2}$ *for any $i > 0$ with $L_i^+ \neq \emptyset$; similarly,* $\min\{x(p) | p \in L_{i-1}^-\} \geqslant \max\{x(q) | q \in L_i^-\} - \frac{1}{2}$ *for any $i > 0$ with $L_i^- \neq \emptyset$.*

(iii) Let p be an arbitrary point in L_i^+ for some $i > 0$. Then the disks of any path $\pi(s,p)$ cover all points in all levels $L_0 \cup L_1 \cup L_2^+ \cup \dots \cup L_{i-1}^+$. A similar statement holds for points in L_i^-.

2.1 Minimum broadcast set in a narrow strip

A *broadcast set* is a point set $D \subseteq P$ that gives a feasible broadcast, i.e., a connected dominating set of $\mathcal{G}$ that contains s. Our task is to find a minimum broadcast set inside a narrow strip. Let $p, p' \in P$ be points with maximum and minimum x-coordinate, respectively. Obviously there must be paths from s to p and p' in $\mathcal{G}$ such that all points on these paths are active, except possibly p and p'. If p and p' are also active, then these paths alone give us a feasible broadcast set: by Observation 1(i), these paths cover all our input points. Instead of activating p and p', it is also enough to activate the points of a path that reaches Q^- and a path that reaches Q^+. In most cases it is sufficient to look for broadcast sets with this structure.

Lemma 1. *If there is a minimum broadcast set with an active point on L_2, then there is a minimum broadcast set consisting of the disks of a shortest path π^- from s to Q^- and a shortest path π^+ from s to Q^+. These two paths share s and they may or may not share their first point after s.*

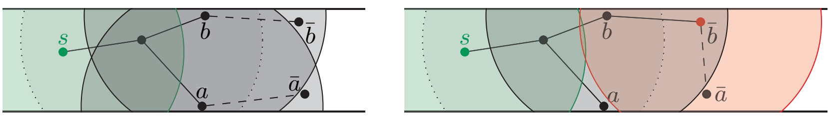

Fig. 1. A swap operation. The edges of the broadcast tree are solid lines.

Proof sketch. If a minimum broadcast set does not contain a point from Q^+, then we can find two active points a and b whose disks uniquely cover two non-active points $\bar{a}$ and $\bar{b}$, respectively; see Fig. 1. By deactivating a and activating $\bar{b}$ we get a new feasible solution, since $\delta(\bar{b})$ covers all points previously only covered by $\delta(a)$. By using such operations repeatedly, we can find a solution containing a point from Q^+. Using similar arguments, we can find a solution also containing a point from Q^-. Finally, using Observation 1(iii), we can show that a shortest path π^+ from s to Q^+ and a shortest path π^- from s to Q^- together form a feasible and minimum-size solution. $\square$

Lemma 2 below fully characterizes optimal broadcast sets. To deal with the case where Lemma 1 does not apply, we need some more terminology. We say that the disk $\delta(q)$ of an active point q in a feasible broadcast set is *bidirectional* if there are two input points $p^- \in L_2^-$ and $p^+ \in L_2^+$ that are covered only by $\delta(q)$. See points p and p' in Fig. 2 for an example. Note that $q \in \mathrm{core}(s)$, because $\mathrm{core}(s) = [-\frac{1}{2}, \frac{1}{2}] \times [0, w]$ is covered by $\delta(s)$, and our bidirectional disk has to cover points both in $(-\infty, -\frac{1}{2}] \times [0, w]$ and $[\frac{1}{2}, \infty) \times [0, w]$. Active disks that are not the source disk and not bidirectional are called *monodirectional*.

Lemma 2. *For any input P that has a feasible broadcast set, there is a minimum broadcast set D that has one of the following structures.*

(i) Small*: $|D| \leqslant 2$.*
(ii) Path-like*: $|D| \geqslant 3$, and D consists of a shortest path π^- from s to Q^- and a shortest path π^+ from s to Q^+; π^+ and π^- share s and may or may not share their first point after s.*
(iii) Bidirectional*: $|D| = 3$, and D contains two bidirectional disk centers and s.*

As it turns out, the bidirectional case is the most difficult one to compute efficiently. (It is similar to CDS-UDG in co-comparability graphs, where the case of a connected dominating set of size at most 3 dominates the running time.)

Lemma 3. *In $O(n \log n)$ time we can find a bidirectional broadcast if it exists.*

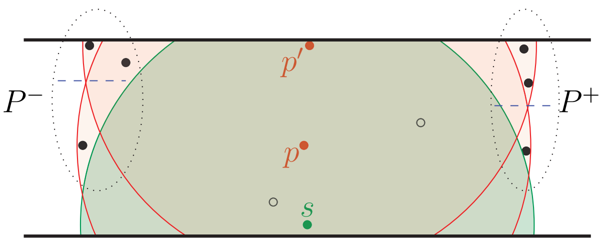

Fig. 2. A bidirectional broadcast.

Proof. Let $P^- := \{u_1, u_2, \ldots, u_k\}$ be the set of points to the left of the source disk $\delta(s)$, where the points are sorted in increasing y-order with ties broken arbitrarily. Similarly, let $P^+ := \{v_1, v_2, \ldots, v_l\}$ be the set of points to the right of $\delta(s)$, again sorted in order of increasing y-coordinate. Define $P^-_{\leqslant i} := \{u_1, \ldots, u_i\}$, and define $P^-_{>i}$, and $P^+_{\leqslant i}$ and $P^+_{>i}$ analogously. Our algorithm is based on the following observation: There is a bidirectional solution if and only if there are indices i, j and points $p, p' \in \text{core}(s)$ such that $\delta(p)$ covers $P^-_{\leqslant i} \cup P^+_{\leqslant j}$ and $\delta(p')$ covers $P^-_{>i} \cup P^+_{>j}$; see Fig. 2.

Now for a point $p \in \text{core}(s)$, define $Z^-_{\leqslant}(p) := \max\{i : P^-_{\leqslant i} \subset \delta(p)\}$ and $Z^-_{>}(p) := \min\{i : P^-_{>i} \subset \delta(p)\}$, and $Z^+_{\leqslant}(p) := \max\{i : P^+_{\leqslant i} \subset \delta(p)\}$, and $Z^+_{>}(p) := \min\{i : P^+_{>i} \subset \delta(p)\}$. Then the observation above can be restated as:

> There is a bidirectional solution if and only if there are points $p, p' \in \text{core}(s)$ such that $Z^-_{\leqslant}(p) \geqslant Z^-_{>}(p')$ and $Z^+_{\leqslant}(p) \geqslant Z^+_{>}(p')$.

It is easy to find such a pair—if it exists—in $O(n \log n)$ time once we have computed the values $Z^-_{\leqslant}(p)$, $Z^-_{>}(p)$, $Z^+_{\leqslant}(p)$, and $Z^+_{>}(p)$ for all points $p \in \delta(s)$. It remains to show that these values can be computed in $O(n \log n)$ time.

Consider the computation of $Z^-_{\leqslant}(p)$; the other values can be computed similarly. Let $\mathcal{T}$ be a balanced binary tree whose leaves store the points from P^- in order of their y-coordinate. For a node ν in $\mathcal{T}$, let $F(\nu) := \{\delta(u_i) : u_i \text{ is stored in the subtree rooted at } \nu\}$. We start by computing at each node ν the intersection of the disks in $F(\nu)$. More precisely, for each ν we compute the region $I(\nu) := \text{core}(s) \cap \bigcap F(\nu)$. Notice that $I(\nu)$ is y-monotone and convex, and each disk $\delta(u_i)$ contributes at most one arc to $\partial I(\nu)$. (Here $\partial I(\nu)$ refers to the boundary of $I(\nu)$ that falls inside $\mathcal{S}$.) Moreover, $I(\nu) = I(\text{left-child}(\nu)) \cap I(\text{right-child}(\nu))$. Hence, we can compute the regions $I(\nu)$ of all nodes ν in $\mathcal{T}$ in $O(n \log n)$ time in total, in a bottom-up manner. Using the tree $\mathcal{T}$ we can now compute $Z^-_{\leqslant}(p)$ for any given $p \in \text{core}(s)$ by searching in $\mathcal{T}$, as follows. Suppose we arrive at a node ν. If $p \in I(\text{left-child}(\nu))$, then descend to right-child(ν), otherwise descend to left-child(ν). The search stops when we reach a leaf, storing a point u_i. One easily verifies that if $p \in \delta(u_i)$ then $Z^-_{\leqslant}(p) = i$, otherwise $Z^-_{\leqslant}(p) = i - 1$.

Since $I(\nu)$ is a convex region, we can check if $p \in I(\nu)$ in $O(1)$ time if we can locate the position of p_y in the sorted list of y-coordinates of the vertices of $\partial I(\nu)$. We can locate p_y in this list in $O(\log n)$ time, leading to an overall query time of $O(\log^2 n)$. This can be improved to $O(\log n)$ using fractional cascading [5]. Note that the application of fractional cascading does not increase the preprocessing time of the data structure. We conclude that we can compute all values $Z^-_{\leqslant}(p)$ in $O(n \log n)$ time in total. □

In order to compute a minimum broadcast, we can first check for small and bidirectional solutions. To find path-like solutions, we first compute the sets Q^- and Q^+, and compute shortest paths starting from these sets back to the source disk. The path computation is very similar to the shortest path algorithm in UDGs by Cabello and Jejčič [4].

Theorem 2. *The broadcast problem inside a strip of width at most $\sqrt{3}/2$ can be solved in $O(n \log n)$ time.*

Remark 1. If we apply this algorithm to every disk as source, we get an $O(n^2 \log n)$ algorithm for CDS in narrow strip UDGs. We can compare this to $O(mn)$, the running time that we get by applying the algorithm for co-comparability graphs [3]. Note that in the most difficult case, when the size of the minimum connected dominating set is at most 3, the unit disk graph has constant diameter, which implies that the graph is dense, i.e., the number of edges is $m = \Omega(n^2)$. Hence, we get an (almost) linear speedup for the worst-case running time.

2.2 Minimum-size h-hop broadcast in a narrow strip

In the hop-bounded version of the problem we are given P and a parameter h, and we want to compute a broadcast set D such that every point $p \in P$ can be reached in at most h hops from s. In other words, for any $p \in P$, there must be a path in $\mathcal{G}$ from s to p of length at most h, all of whose vertices, except possibly p itself, are in D. We start by investigating the structure of optimal solutions in this setting, which can be very different from the non-hop-bounded setting.

As before, we partition P into levels L_i according to the hop distance from s in the graph $\mathcal{G}$, and we define L_i^+ and L_i^- to be the subsets of points at level i with positive and nonnegative x-coordinates, respectively. Let L_t be the highest non-empty level. If $t > h$ then clearly there is no feasible solution.

If $t < h$ then we can safely use our solution for the non-hop-bounded case, because the non-hop-bounded algorithm gives a solution which contains a path with at most $t+1$ hops to any point in P. This follows from the structure of the solution; see Lemma 2. (Note that it is possible that the solution given by this algorithm requires $t+1$ hops to some point, namely, if $Q^+ \cup Q^- \subseteq L_t$.) With the $t < h$ case handled by the non-hop-bounded algorithm, we are only concerned with the case $t = h$.

We deal with *one-sided* inputs first, where the source is the leftmost input point. Let $\mathcal{G}^*$ be the directed graph obtained by deleting edges connecting points inside the same level of $\mathcal{G}$, and orienting all remaining edges from lower to higher levels. A *Steiner arborescence* of $\mathcal{G}^*$ for the terminal set L_h is a directed tree rooted at s that contains a (directed) path π_p from s to p for each $p \in L_h$. From now on, whenever we speak of *arborescence* we refer to a Steiner arborescence in $\mathcal{G}^*$ for terminal set L_h. We define the *size* of an arborescence to be the number of internal nodes of the arborescence. Note that the leaves in a minimum-size arborescence are exactly the points in L_h: these points must be in the arborescence by definition, they must be leaves since they have out-degree zero in $\mathcal{G}^*$, and leaves that are not in L_h can be removed.

Remark 2. In the minimum Steiner Set problem, we are given a graph G and a vertex subset T of terminals, and the goal is to find a minimum-size vertex subset S such that $T \cup S$ induces a connected subgraph. This problem has a polynomial algorithm in co-comparability graphs [3], and therefore in narrow

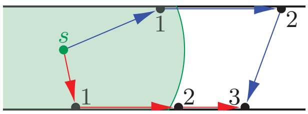

Fig. 3. Two different arborescences, with vertices labeled with their level. The arborescence made of the bottom path does not define a feasible broadcast for $h = 3$, since it would take four hops to reach the top right node.

strip unit disk graphs. However, the broadcast set given by a solution does not fit our hop bound requirements. Hence, we have to work with a different graph (e.g. the edges within each level L_i have been removed), and this modified graph is not necessarily a co-comparability graph.

Lemma 4 below states that either we have a path-like solution—for the one-sided case a path-like solution is a shortest $s \to Q^+$ path— or any minimum-size arborescence defines a minimum-size broadcast set. The latter solution is obtained by activating all non-leaf nodes of the arborescence. We denote the broadcast set obtained from an arborescence A by D_A.

Lemma 4. *Any minimum-size Steiner arborescence for the terminal set L_h defines a minimum broadcast set, or there is a path-like minimum broadcast set.*

Notice that a path-like solution also corresponds to an arborescence. However, it can happen that there are minimum-size arborescences that do not define a feasible broadcast; see Fig. 3. Lemma 4 implies that if this happens, then there must be an optimal path-like solution. The lemma also implies that for non-path-like solutions we can use the Dreyfus-Wagner dynamic-programming algorithm to compute a minimum Steiner tree [8], and obtain an optimal solution from this tree.[3] Unfortunately the running time is exponential in the number of terminals, which is $|L_h|$ in our case. However, our setup has some special properties that we can use to get a polynomial algorithm.

We define an arborescence A to be *nice* if the following holds. For any two arcs uu' and vv' of A that go between the same two levels, with $u \neq v$, we have: $y(u') < y(v') \Rightarrow y(u) < y(v)$. Intuitively, a nice arborescence is one consisting of paths that can be ordered vertically in a consistent manner, see the left of Fig. 4. We define an arborescence A to be *compatible* with a broadcast set D if $D = D_A$. Note that there can be multiple arborescences—that is, arborescences with the same node set but different edge sets—compatible with a given broadcast set D.

Lemma 5. *Every optimal broadcast set D has a nice compatible arborescence.*

Proof sketch. To find a nice compatible arborescence we will associate a unique arborescence with D. To this end, we define for each $p \in (D \cup L_h) \setminus \{s\}$ a unique

[3] The Dreyfus-Wagner algorithm minimizes the number of edges in the arborescence. In our setting the number of edges equals the number of internal nodes plus $|L_h| - 1$, so this also minimizes the number of internal nodes.

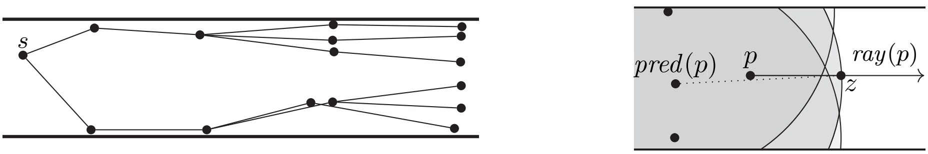

Fig. 4. Left: A nice Steiner arborescence. Note that arc crossings are possible. Right: Defining the *pred* function.

predecessor $pred(p)$, as follows. Let ∂_i^* be the boundary of $\bigcup\{\delta(p)|p \in L_i \cap D\}$. The two lines bounding the strip $\mathcal{S}$ cut ∂_i^* into four parts: a top and a bottom part that lie outside the strip, and a left and a right part that lie inside the strip. Let ∂_i be the part on the right inside the strip. We then define the function $pred : (D \cup L_h) \setminus \{s\} \to D$ the following way. Consider a point $p \in (D \cup L_h) \setminus \{s\}$ and let i be its level. Let $ray(q)$ be the horizontal ray emanating from q to the right; see the right of Fig. 4. It follows from Observation 1(iii) that $ray(q)$ cannot enter any disk from level $i-1$. We can prove that any point $p \in D \cap L_h$ is contained in a disk from p's previous level, so $pred(p)$ is well defined for these points. The edges $pred(p)p$ for $p \in D \cap L_h$ thus define an arborescence. We can prove that it is nice by showing that the y-order of the points in a level L_i corresponds to the vertical order in which the boundaries of their disks appear on $\bigcup\{\delta(p) : p \in L_i \cap D\}$. $\square$

Let $q_1, q_2, \ldots, q_m$ be the points of L_h in increasing y-order. The crucial property of a nice arborescence is that the descendant leaves of a point p in the arborescence form an interval of $q_1, q_2, \ldots, q_m$. Using the above lemmas, we can adapt the Dreyfus-Wagner algorithm and get the following theorem.

Theorem 3. *The one-sided h-hop broadcast problem inside a strip of width at most $\sqrt{3}/2$ can be solved in $O(n^4)$ time.*

In the general (two-sided) case, we can have path-like solutions and arborescence-based solutions on both sides, and the two side solutions may or may not share points in L_1. We also need to handle "small" solutions—now these are 2-hop solutions—separately.

Theorem 4. *The h-hop broadcast problem inside a strip of width at most $\sqrt{3}/2$ can be solved in $O(n^6)$ time.*

3 Broadcasting in a wide strip

Theorem 5. *The broadcast problem and* CDS-UDG *can be solved in $n^{O(w)}$ time on a strip of width w. Moreover, there is no algorithm for* CDS-UDG *or the broadcast problem with runtime $f(w)n^{o(w)}$ unless ETH fails.*

Surprisingly, the h-hop version has no $n^{O(w)}$ algorithm (unless P $=$NP).

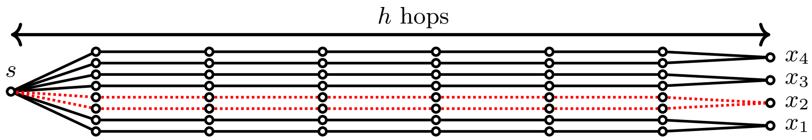

Fig. 5. The gadget representing the variables. The dotted paths form the x_2-string.

Theorem 6. *The h-hop broadcast problem is* NP*-complete in strips of width* 40*.*

(The theorem of course refers to the decision version of the problem: given a point set P, a hop bound h, and a value K, does P admit an h-hop broadcast set of size at most K?) Our reduction is from 3-SAT. Let $x_1, x_2, \ldots x_n$ be the variables and $C_1, \ldots, C_m$ be the clauses of a 3-CNF.

Fig. 5 shows the structural idea for representing the variables, which we call the *base bundle*. It consists of $(2h-1)n+1$ points arranged as shown in the figure, where h is an appropriate value. The distances between the points are chosen such that the graph $\mathcal{G}$, which connects two points if they are within distance 1, consists of the edges in the figure plus all edges between points in the same level. Thus (except for the intra-level edges, which we can ignore) $\mathcal{G}$ consists of n pairs of paths, one path pair for each variable x_i. The i-th pair of paths represents the variable x_i, and we call it the *x_i-string*. By setting the target size, K, of the problem appropriately, we can ensure the following for each x_i: any feasible solution must use either the top path of the x_i-string or the bottom path, but it cannot use points from both paths. Thus we can use the top path of the x_i-path to represent a TRUE setting of the variable x_i, and the bottom path to represent a FALSE setting. A group of consecutive strings is called a *bundle*. We denote the bundle containing all x_t-strings with $t = i, i+1, \ldots, j$ by *bundle*(i, j).

The clause gadgets all start and end in the base bundle, as shown in Fig. 6. The gadget to check a clause involving variables x_i, x_j, x_k, with $i < j < k$, roughly works as follows; see also the lower part of Fig. 6, where the strings for x_i, x_j, and x_k are drawn with dotted lines.

First we split off *bundle*$(1, i-1)$ from the base bundle, by letting the top $i-1$ strings of the base bundle turn left. (In Fig. 6 this bundle consists of two strings.) We then separate the x_i-string from the base bundle, and route the x_i-string into a *branching gadget*. The branching gadget creates a branch consisting of two *tapes*—this branch will eventually be routed to the *clause-checking gadget*—and a branch that returns to the base bundle. Before the tapes can be routed to the clause-checking gadget, they have to cross each of the strings in *bundle*$(1, i-1)$. For each string that must be crossed we introduce a *crossing gadget*. A crossing gadget lets the tapes continue to the right, while the string being crossed can return to the base bundle. The final crossing gadget turns the tapes into a *side string* that can now be routed to the clause-checking gadget. The construction guarantees that the side string for x_i still carries the truth value that was selected for the x_i-string in the base bundle. Moreover, if the TRUE path (resp. FALSE

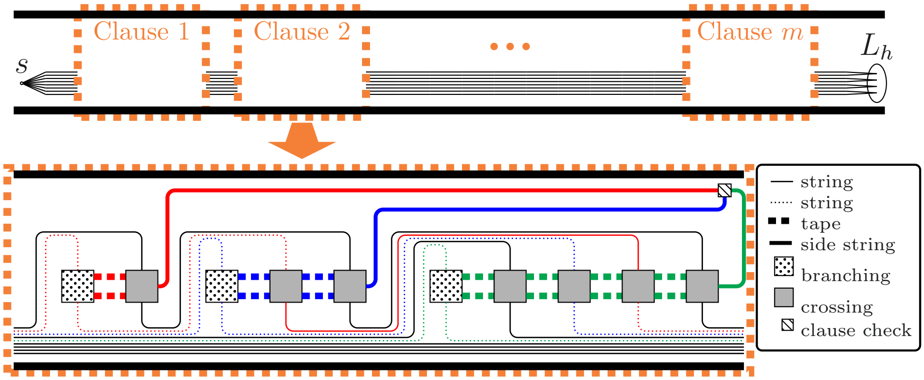

Fig. 6. The overall construction, and the way a single clause is checked. Note that in this figure each string (which actually consists of two paths) is shown as a single curve.

path) of the x_i-string was selected to be part of the broadcast set initially, then the TRUE path (resp. FALSE path) of the rest of the x_i-string that return to the base bundle must be in the minimum broadcast set as well.

After we have created a side string for x_i, we create side strings for x_j and x_k in a similar way. The three side strings are then fed into the clause-checking gadget. The clause-checking gadget is a simple construction of four points. Intuitively, if at least one side string carries the correct truth value—TRUE if the clause contains the positive variable, FALSE if it contains the negated variable—, then we activate a single disk in the clause check gadget that corresponds to a true literal. Otherwise we need to change truth value in at least one of the side strings, which requires an extra disk.

The final construction contains $\Theta(n^4 m)$ points that all fit into a strip of width 40. The details are given in the full version.

4 Conclusion

We studied the complexity of the broadcast problem in narrow and wider strips. For narrow strips we obtained efficient polynomial algorithms, both for the non-hop-bounded and for the h-hop version, thanks to the special structure of the problem inside such strips. On wider strips, the broadcast problem has an $n^{O(w)}$ algorithm, while the h-hop broadcast becomes NP-complete on strips of width 40. With the exception of a constant width range (between $\sqrt{3}/2$ and 40) we characterized the complexity when parameterized by strip width. We have also proved that the planar problem (and, similarly, CDS-UDG) is W[1]-hard when parameterized by the solution size. The problem of finding a planar h-hop broadcast set seems even harder: we can solve it in polynomial time for $h = 2$ (see full version) but already for $h = 3$ we know no better algorithm than brute force. Interesting open problems include:

- What is the complexity of planar 3-hop broadcast? In particular, is there a constant value t such that t-hop broadcast is NP-complete?
- What is the complexity of h-hop broadcast in planar graphs?

References

1. Ambühl, C.: An optimal bound for the MST algorithm to compute energy efficient broadcast trees in wireless networks. In: ICALP, Proceedings. pp. 1139–1150 (2005)
2. Ambühl, C., Clementi, A.E.F., Ianni, M.D., Lev-Tov, N., Monti, A., Peleg, D., Rossi, G., Silvestri, R.: Efficient algorithms for low-energy bounded-hop broadcast in ad-hoc wireless networks. In: STACS, Proceedings. pp. 418–427 (2004)
3. Breu, H.: Algorithmic aspects of constrained unit disk graphs. Ph.D. thesis, University of British Columbia (1996)
4. Cabello, S., Jejčič, M.: Shortest paths in intersection graphs of unit disks. Comput. Geom. 48(4), 360–367 (2015)
5. Chazelle, B., Guibas, L.J.: Fractional cascading: I. A data structuring technique. Algorithmica 1(1), 133–162 (1986)
6. Clementi, A.E., Crescenzi, P., Penna, P., Rossi, G., Vocca, P.: A worst-case analysis of an mst-based heuristic to construct energy-efficient broadcast trees in wireless networks. In: STACS, Proceedings. pp. 121–131 (2001)
7. Das, G.K., Das, S., Nandy, S.C.: Range assignment for energy efficient broadcasting in linear radio networks. Theor. Comput. Sci. 352(1-3), 332–341 (2006)
8. Dreyfus, S.E., Wagner, R.A.: The Steiner problem in graphs. Networks 1(3), 195–207 (1971)
9. Flum, J., Grohe, M.: Parameterized Complexity Theory. Texts in Theoretical Computer Science. An EATCS Series, Springer (2006)
10. Fuchs, B.: On the hardness of range assignment problems. Networks 52(4), 183–195 (2008)
11. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman (1979)
12. Kratsch, D., Stewart, L.: Domination on cocomparability graphs. SIAM J. Discrete Math. 6(3), 400–417 (1993)
13. Lichtenstein, D.: Planar formulae and their uses. SIAM J. Comput. 11(2), 329–343 (1982)
14. Marx, D.: Parameterized complexity of independence and domination on geometric graphs. In: Parameterized and Exact Computation, Second International Workshop, IWPEC, Proceedings. pp. 154–165 (2006)
15. Masuyama, S., Ibaraki, T., Hasegawa, T.: The computational complexity of the m-center problems on the plane. IEICE Transactions 64(2), 57–64 (1981)
16. Matsui, T.: Approximation algorithms for maximum independent set problems and fractional coloring problems on unit disk graphs. In: JCDCG, Proceedings. pp. 194–200. Springer (1998)