Algorithmica (2019) 81:2963–2990
https://doi.org/10.1007/s00453-019-00561-0

The Homogeneous Broadcast Problem in Narrow and Wide Strips II: Lower Bounds

Mark de Berg[1] · **Hans L. Bodlaender**[2] · **Sándor Kisfaludi-Bak**[1]

Received: 12 December 2017 / Accepted: 28 February 2019 / Published online: 19 March 2019

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—where we also consider how the complexity depends on the width w of the strip. We prove the following two lower bounds. First, we show that the regular version of the problem is W[1]-complete when parameterized by the solution size k. More precisely, we show that the problem does not admit an algorithm with running time $f(k)n^{o(\sqrt{k})}$, unless ETH fails. The construction can also be used to show an $f(w)n^{\Omega(w)}$ lower bound when we parameterize by the strip width w. Second, we prove that the hop-bounded version of the problem is NP-hard in strips of width 40. These results complement the algorithmic results in a companion paper (de Berg et al. in Algorithmica, submitted).

Keywords Broadcast · Dominating set · Unit disk graph · Range assignment

This research was supported by The Netherlands Organization for Scientific Research (NWO) under Project No. 024.002.003.

✉ Sándor Kisfaludi-Bak
s.kisfaludi.bak@tue.nl

Mark de Berg
m.t.d.berg@tue.nl

Hans L. Bodlaender
h.l.bodlaender@uu.nl

[1] Department of Mathematics and Computer Science, TU Eindhoven, P.O. Box 513, 5600 MB Eindhoven, The Netherlands

[2] Department of Computer Science, Utrecht University, Utrecht, The Netherlands

1 Introduction

Inspired by wireless network applications, we study two variants of the *homogeneous broadcast problem*. Let P be a set of n points in $\mathbb{R}^d$ and let $s \in P$ be a source node. Each point can either be *active*, in which case its transmission range is 1, or it can be *inactive*. Thus a point p can transmit to point q if and only if p is active, and $|pq| \leqslant 1$. We can assign a directed graph to a set of active points $\Delta \subseteq P$: Let $\mathcal{G}_\Delta = (P, E_\Delta)$ be the directed graph where $(p, q) \in E_\Delta$ iff $p \in \Delta$ and $|pq| \leqslant 1$. We say that Δ is a *homogeneous broadcast set* (or simply a *broadcast set*) if every point $p \in P$ is reachable from s in $\mathcal{G}_\Delta$. If every $p \in P$ is reachable within h hops, for a given parameter h, then Δ is an *h-hop broadcast set*. The (h-hop) broadcast problem is to find an (h-hop) broadcast set of minimum size. Observe that if $p \in \Delta$ 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 $\mathcal{G}_\Delta$ is the intersection graph of a set of congruent disks or, in other words, a *unit disk graph*. Because of their relation to wireless networks, unit disk graphs have been studied extensively [2,3].

For more information on the broadcast problem's relation to wireless networks, please see the introduction of the companion paper [1].

Let $\mathcal{D}$ be a set of congruent disks in the plane, and let $\mathcal{G}_\mathcal{D}$ be the unit disk graph 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 $\Delta \subset \mathcal{D}$ such that the subgraph induced by Δ is connected and each node in $\mathcal{G}_\mathcal{D}$ is either in Δ or a neighbor of a node in Δ. The broadcast problem is thus equivalent to the following: given a unit disk graph $\mathcal{G}_\mathcal{D}$ with a designated source node s, compute a minimum-size connected dominating set $\Delta \subset \mathcal{D}$ such that $s \in \Delta$. The Connected Dominating Set problem is recognized as a fundamental problem for wireless network design, see the survey [4].

Given an algorithm for the broadcast problem, one can solve Connected Dominating Set in Unit Disk Graphs by running the algorithm n times, once for each possible source point. (In fact, we only need to run the algorithm $d_{\min} + 1$ times, where $d_{\min}$ is the minimum degree of any vertex in the graph, since it suffices to try v and each of its neighbors as the source.) Consequently, hardness results for Connected Dominating Set in Unit Disk Graphs can be transferred to the broadcast problem, and algorithms for the broadcast problem can be transferred to Connected Dominating Set in Unit Disk Graphs at the cost of an extra linear factor in the running time. It is well known that Dominating Set and Connected Dominating Set are NP-hard, even for planar graphs [5], and they remain NP-hard in unit disk graphs [6,7]. For any fixed d, both problems can be solved in $2^{O(n^{1-1/d})}$ time in unit balls graphs of $\mathbb{R}^d$, and even in more general intersection graphs [8]; this running time is tight under ETH. The parameterized complexity of Dominating Set in Unit Disk Graphs has also been investigated: Marx [9] proved that it is W[1]-hard when parameterized by the size of the dominating set, and De Berg et al.[10] showed that for most natural geometric intersection graphs (including unit disk graphs), Dominating Set is contained in W[1]. (The definition of W[1] and other parameterized complexity classes can be found in the book by Flum and Grohe [11].)

1.1 Our Contributions

We give lower bounds for the broadcast problem inside a strip of width w. Together with the companion paper [1], we get an almost complete dichotomy for the width parameter, both for the general and for the hop-bounded version of the problem.

Our first lower bound result investigates the parameterized complexity of Connected Dominating Set in unit disk graphs, where the parameter k is the size of the solution set. We show that the problem is W[1]-complete. The proof is discussed in Sect. 2. It is a reduction from Grid Tiling based on ideas by Marx [9]. The running time lower bound is based on the Exponential Time Hypothesis (ETH) [12].

Theorem 1 *The broadcast problem and* Connected Dominating Set *in Unit Disk Graphs are* W[1]*-complete when parameterized by the solution size. Moreover, there is no $f(k)n^{o(\sqrt{k})}$ algorithm for these problems, where n is the number of input disks and k is the size of the solution, unless ETH fails.*

As seen in Remark 2 of [1], this lower bound is tight: there is a Connected Dominating Set algorithm for unit disk graphs with running time $n^{O(\sqrt{k})}$.

If we consider the same construction inside a strip, we immediately get that the $n^{O(w)}$ dynamic-programming algorithm for broadcasting in strips of width w (given in Part I of this paper) is likely best possible. The obtained lower bound is $f(w)n^{\Omega(w)}$, conditional on ETH.

Interestingly, the h-hop broadcast problem has no such algorithm (unless $\mathsf{P} = \mathsf{NP}$) by our second main theorem.

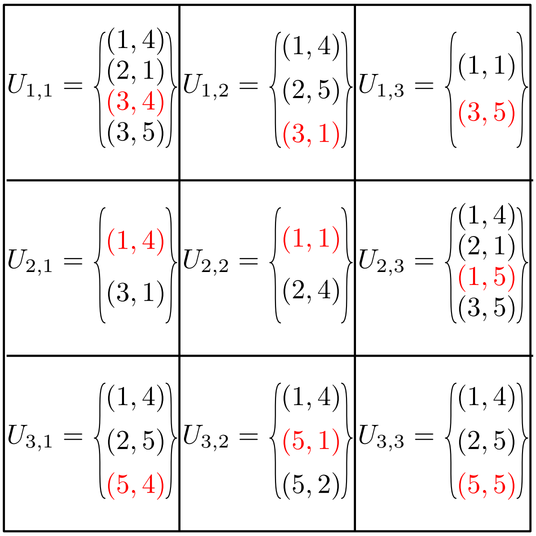
Theorem 2 *The h-hop broadcast problem is* NP*-complete in strips of width 40.*

Some of the gadgets in this intricate construction are simple variations of the gadget we needed for the W[1]-hardness proof in Theorem 1.

Our reduction is a polynomial reduction from 3-SAT. Given a formula with n variables and m clauses, it generates an instance with $O(n^3m)$ disks, where the minimum distance between neighboring disk centers is $\Omega(1/n)$, but the centers need to be defined up to a precision of $\Theta(n^{-2})$. The running time of the reduction is proportional to the number of generated disks, $O(n^3m)$, and the size of the minimum broadcast set is also $k = \Theta(n^3m)$. The value of h in our construction is $O(n^2m)$.

1.2 Related Work

The Grid Tiling problem has been a useful basis for geometric reductions in the past decade. It was used for W[1]-hardness results of Independent Set and Dominating Set in unit disk graphs [9,13]. A variant has been useful in getting lower bounds for approximation schemes [14], but mostly it has been used to get lower bounds on



$U_{1,1}=\{(1,4),(2,1),{\color{red}(3,4)},(3,5)\}$	$U_{1,2}=\{(1,4),(2,5),{\color{red}(3,1)}\}$	$U_{1,3}=\{(1,1),{\color{red}(3,5)}\}$
$U_{2,1}=\{{\color{red}(1,4)},(3,1)\}$	$U_{2,2}=\{{\color{red}(1,1)},(2,4)\}$	$U_{2,3}=\{(1,4),(2,1),{\color{red}(1,5)},(3,5)\}$
$U_{3,1}=\{(1,4),(2,5),{\color{red}(5,4)}\}$	$U_{3,2}=\{(1,4),{\color{red}(5,1)},(5,2)\}$	$U_{3,3}=\{(1,4),(2,5),{\color{red}(5,5)}\}$

Fig. 1 GRID TILING example with $k = 3$; a solution is highlighted. The goal is to find a $u_{a,b} \in U_{a,b}$ for each $(a, b) \in [k] \times [k]$ so that horizontal/(vertical) neighbors share their first/(second) coordinate

exact computation. GRID TILING has also been extended to higher dimensions [15], and successfully used to establish lower bounds for coloring unit disk and unit ball graphs [16]. Most recently, it has been used to study the parameterized complexity of STEINER TREE in planar graphs [17,18]. The problem is also starting to be applied outside the strictly geometric setting: in computational topology [19] and for studying H-free graphs [20].

Our Theorem 2 is using gadgetry similar to other reductions in geometric intersection graphs, but it is ultimately a more standard reduction from the 3-SAT problem. It can be regarded as a strengthening of [21], where the authors show NP-hardness for a version of h-hop broadcast where there is no restriction to a strip, and instead of a single source vertex s there can be a set S of sources given with the input.

2 A Parameterized Look at CONNECTED DOMINATING SET in Unit Disk Graphs

In this section we prove that CONNECTED DOMINATING SET in Unit Disk Graphs is W[1]-hard parameterized by the solution size; our proof heavily relies on the proof of the W[1]-hardness of DOMINATING SET in Unit Disk Graphs by Marx [9].

2.1 Sketch of the Construction by Marx for DOMINATING SET in Unit Disk Graphs

Marx uses a reduction from GRID TILING, see the book [13] (note that in [9] the GRID TILING problem is not stated explicitly). In a GRID TILING instance we are given an integer k, an integer n, and a collection $\mathcal{S}$ of k^2 non-empty sets $U_{a,b} \subseteq [n] \times [n]$ for $1 \leqslant a, b \leqslant k$. The goal is to select an element $u_{a,b} \in U_{a,b}$ for each $1 \leqslant a, b \leqslant k$ such that

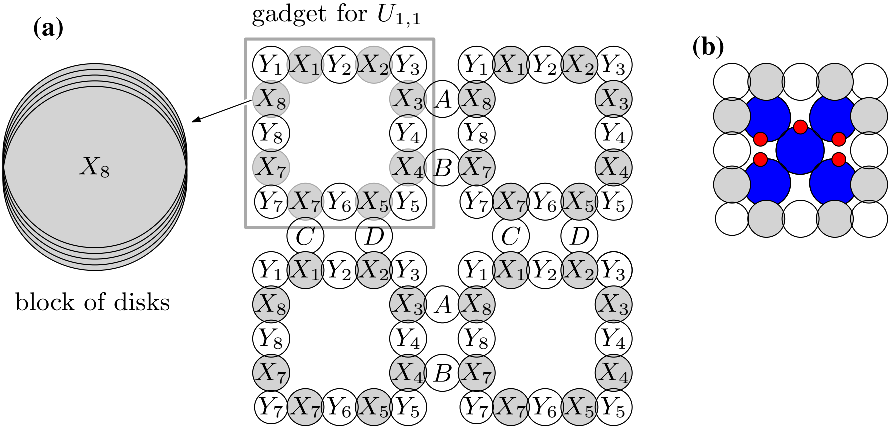


Fig. 2 **a** The construction by Marx. **b** The idea behind our construction

– If $u_{a,b} = (x, y)$ and $u_{a+1,b} = (x', y')$, then $x = x'$.
– If $u_{a,b} = (x, y)$ and $u_{a,b+1} = (x', x')$, then $y = y'$.

One can picture these sets in a $k \times k$ matrix: in each cell (a, b), we need to select a representative from the set $U_{a,b}$ so that the representatives selected from horizontally neighboring cells agree in the first coordinate, and representatives from vertically neighboring sets agree in the second coordinate (see Fig. 1).

Marx's reduction places k^2 gadgets, one for each $U_{a,b}$. A gadget contains 16 blocks of disks, labeled $X_1, Y_1, X_2, Y_2, \dots, X_8, Y_8$, that are arranged along the edges of a square—see Fig. 2a. Initially, each block X_ℓ contains n^2 disks, denoted by $X_\ell(1), \dots, X_\ell(n^2)$ and each block Y_ℓ contains $n^2 + 1$ disks denoted by $Y_\ell(0), \dots, Y_\ell(n^2)$. The argument j of $X_\ell(j)$ can be thought of as a pair (x, y) with $1 \leqslant x, y \leqslant n$ for which $f(x, y) := (x-1)n + y = j$. Let $f^{-1}(j) = (\iota_1(j), \iota_2(j)) = (1 + \lfloor j/n \rfloor, 1 + (j \mod n))$. For the final construction, in each gadget at position (a, b), delete all disks $X_\ell(j)$ for each $\ell = 1, \dots, 8$ and $(\iota_1(j), \iota_2(j)) \notin U_{a,b}$. This deletion ensures that the gadgets represent the corresponding set $U_{a,b}$. The construction is such that a minimum dominating set uses only disks in the X-blocks, and that for each gadget (a, b) the same disk $X_\ell(j)$ is chosen for each $1 \leqslant \ell \leqslant 8$. This choice signifies a specific choice $u_{a,b} = (x, y)$. To ensure that the choice for $u_{a,b}$ in the same row and column agrees on their first and second coordinate, respectively, there are special connector blocks between neighboring gadgets. The connector blocks are denoted by A, B, C and D in Fig. 2a, and they each contain $n + 1$ disks—see Sect. 2.3 for further details.

2.2 Overview of Our Construction for Connected Dominating Set in Unit Disk Graphs

To extend the construction to Connected Dominating Set in Unit Disk Graphs, we have to make sure there is a minimum-size dominating set that is connected. This requires two things. First, we must add new disks inside the gadgets—that is, in the empty space surrounded by the X- and Y-blocks—to guarantee a connection between all chosen $X_\ell(j)$ disks without interfering with the disks in the Y-blocks. Second, we need to connect all the different gadgets. This time, in addition to avoiding the Y-blocks, we also need to avoid interfering with the connector blocks.

The idea is as follows. Inside each gadget we add several pairs of disks, consisting of a *parent* disk and a *leaf* disk. The parent disks are placed such that, for any choice of one disk from each of the X-blocks, the parent disks together with the eight chosen disks from the X-blocks form a connected set. Moreover, the parent disks do not intersect any disk in a Y-block. See Fig. 2b for an illustration; the parent disks are blue in the figure. For each parent disk we add a leaf disk—the red disks in the figure—that only intersects its parent disk. The following is a key observation.

Observation 3 *There is a minimum dominating set which contains all parent disks.*

Proof Since the leaf needs to be dominated, either it or its parent needs to be in any given dominating set D. Note that if the leaf is in D but its parent is not, then they can be exchanged; the resulting set is not larger than D and it is dominating. □

This observation can be used to show that any canonical minimum dominating set in our construction is connected.

In Fig. 2b we used disks of different sizes. Unfortunately this is not allowed, which makes the construction significantly more tricky. To be able to place the pairs in a suitable way, we need to create more space inside the gadget. To this end we use a gadget consisting of 16 (instead of eight) X- and Y-blocks. This will also give us sufficient space to put parent–leaf pairs in between the gadgets, so the dominating sets from adjacent gadgets are connected through the parent disks. Thus the size of a minimum connected dominating set in the new construction is equal to the size of a minimum dominating set in the old construction plus the number of parent disks. Hence, we can decide if the Grid Tiling instance has a solution by checking the size of the minimum connected dominating set in our construction. Thus Connected Dominating Set in Unit Disk Graphs is W[1]-hard, and the reduction together with the the Exponential Time Hypothesis (ETH) yields the desired lower bound.

To review our construction, we need to delve into some of the details of the construction in [9].

2.3 Some Details of the Construction in [9]

In every block, the place of each disk center is defined with regard to the midpoint of the block, $(x(z), y(z))$. The center of each circle is of the form $(x(z) + \alpha\epsilon, y(z) + \beta\epsilon)$ where $x(z)$, $y(z)$, α and β are integers, and $\epsilon > 0$ a small constant. We say that the *offset* of the disk centered at $(x(z)+\alpha\epsilon, y(z)+\beta\epsilon)$ is (α, β). Note that $|\alpha|, |\beta| \leqslant n^2+1$, and $\epsilon < n^{-3}$, so the disks in a block all intersect each other. The offsets of X and Y-blocks are defined as follows.

$$\begin{array}{ll}
\mathrm{offset}(X_1(j)) = (j, -\iota_2(j)) & \mathrm{offset}(Y_1(j)) = (j + 0.5, j + 0.5) \\
\mathrm{offset}(X_2(j)) = (j, \iota_2(j)) & \mathrm{offset}(Y_2(j)) = (j + 0.5, -n) \\
\mathrm{offset}(X_3(j)) = (-\iota_1(j), -j) & \mathrm{offset}(Y_3(j)) = (j + 0.5, -j - 0.5) \\
\mathrm{offset}(X_4(j)) = (\iota_1(j), -j) & \mathrm{offset}(Y_4(j)) = (-n, -j - 0.5) \\
\mathrm{offset}(X_5(j)) = (-j, \iota_2(j)) & \mathrm{offset}(Y_5(j)) = (-j - 0.5, -j - 0.5) \\
\mathrm{offset}(X_6(j)) = (-j, -\iota_2(j)) & \mathrm{offset}(Y_6(j)) = (-j - 0.5, n) \\
\mathrm{offset}(X_7(j)) = (\iota_1(j), j) & \mathrm{offset}(Y_7(j)) = (-j - 0.5, j + 0.5) \\
\mathrm{offset}(X_8(j)) = (-\iota_1(j), j) & \mathrm{offset}(Y_8(j)) = (n, j + 0.5)
\end{array}$$

We remark some important properties. First, two disks can intersect only if they are in the same or in neighboring blocks. Consequently, one needs at least eight disks to dominate a gadget. The second important property is that disk $X_\ell(j)$ dominates exactly $Y_\ell(j), \ldots, Y_\ell(n^2)$ from the "previous" block Y_ℓ, and $Y_{\ell+1}(0), \ldots, Y_{\ell+1}(j-1)$ from the "next" block $Y_{\ell+1}$. This property can be used to prove the following key lemma.

Lemma 4 (Lemma 1 of [9]) *Assume that a gadget is part of an instance such that none of the blocks Y_i are intersected by disks outside the gadget. If there is a dominating set Δ of the instance that contains exactly $8k^2$ disks, then there is a* canonical *dominating set Δ' with $|\Delta'| = |\Delta|$, such that for each gadget $\mathcal{G}$, there is an integer $1 \leqslant j^G \leqslant n$ such that Δ' contains exactly the disks $X_1(j^G), \ldots, X_8(j^G)$ from $\mathcal{G}$.*

In the gadget $G_{a,b}$, the value j defined in the above lemma represents the choice of $s_{a,b} = (\iota_1(j), \iota_2(j))$ in the grid tiling problem. Our deletion of certain disks in X-blocks ensures that $(\iota_1(j), \iota_2(j)) \in U_{a,b}$. Finally, in order to get a feasible grid tiling, gadgets in the same row must agree on the first coordinate, and gadgets in the same column must agree on the second coordinate. These blocks have $n+1$ disks each, with indices $0, 1, \ldots, n$. We define the offsets in the connector gadgets the following way.

$$\begin{array}{ll}
\mathrm{offset}(A_j) = (-j - 0.5, -n^2 - 1) & \mathrm{offset}(B_j) = (j + 0.5, n^2 + 1) \\
\mathrm{offset}(C_j) = (n^2 + 1, -\iota_2(j)) & \mathrm{offset}(D_j) = (-n^2 - 1, \iota_2(j))
\end{array}$$

Using this definition, it is easy to prove the following lemma.

Lemma 5 *Let Δ be a canonical dominating set. For horizontally neighboring gadgets $\mathcal{G}$ and H representing j_G and j_H, the disks of the connector block A are dominated if and only if $\iota_1(j_G) \leqslant \iota_1(j_H)$; the disks of B are dominated if and only if $\iota_1(j_G) \geqslant \iota_1(j_H)$. Similarly, for vertically neighboring blocks G' and H', the disks of block C are dominated if and only if $\iota_2(j_{G'}) \leqslant \iota_2(j_{H'})$; the disks of D are dominated if and only if $\iota_2(j_{G'}) \geqslant \iota_2(j_{H'})$.*

With the above lemmas, the correctness of the reduction follows. A feasible grid tiling defines a dominating set of size $8k^2$: in gadget $G_{a,b}$, the dominating disks are $X_\ell\left(f(u_{a,b})\right),\ \ell = 1, \ldots, 8$. On the other hand, if there is a dominating set of size $8k^2$, then there is a canonical dominating set of the same size that defines a feasible grid tiling.

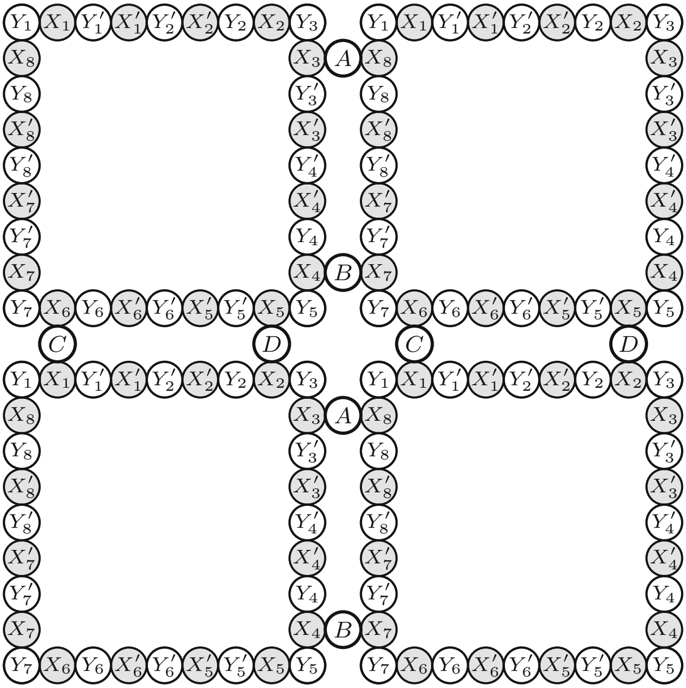

Y_1 X_1 Y_1' X_1' Y_2' X_2' Y_2 X_2 Y_3 Y_1 X_1 Y_1' X_1' Y_2' X_2' Y_2 X_2 Y_3
X_8 X_3 A X_8 X_3
Y_8 Y_3' Y_8 Y_3'
X_8' X_3' X_8' X_3'
Y_8' Y_4' Y_8' Y_4'
X_7' X_4' X_7' X_4'
Y_7' Y_4 Y_7' Y_4
X_7 X_4 B X_7 X_4
Y_7 X_6 Y_6 X_6' Y_6' X_5' Y_5' X_5 Y_5 Y_7 X_6 Y_6 X_6' Y_6' X_5' Y_5' X_5 Y_5
C D C D
Y_1 X_1 Y_1' X_1' Y_2' X_2' Y_2 X_2 Y_3 Y_1 X_1 Y_1' X_1' Y_2' X_2' Y_2 X_2 Y_3
X_8 X_3 A X_8 X_3
Y_8 Y_3' Y_8 Y_3'
X_8' X_3' X_8' X_3'
Y_8' Y_4' Y_8' Y_4'
X_7' X_4' X_7' X_4'
Y_7' Y_4 Y_7' Y_4
X_7 X_4 B X_7 X_4
Y_7 X_6 Y_6 X_6' Y_6' X_5' Y_5' X_5 Y_5 Y_7 X_6 Y_6 X_6' Y_6' X_5' Y_5' X_5 Y_5

Fig. 3 Connecting neighboring gadgets

2.4 Gadgetry of Our Connected Dominating Set Construction

To extend the construction to Connected Dominating Set in Unit Disk Graphs, we want to make sure that minimum-size dominating set is connected. This requires two things. First, we must add new disks "inside" the gadgets—that is, in the empty space surrounded by the X and Y-blocks—such that a canonical minimum dominating set includes some new disks that connect the chosen $X_\ell(j)$ disks without interfering with disks in the Y-blocks. Second, we need to connect all the different gadgets. This time in addition to avoiding the Y-blocks, we also need to avoid interfering with the connector blocks.

In order to have enough space, our gadgets contain 16 X-blocks and 16 Y-blocks instead of eight. The offsets of disks inside the blocks are not modified: we use the same building blocks. Figure 3 shows how we arrange these blocks, and depicts the connector block placement.

The analogue of Lemmas 4 and 5 are true here; we have a construction that could be used to prove the W[1]-hardness of Dominating Set in Unit Disk Graphs, with canonical sets of size $16k^2$, that contain one disk from each X-block and X'-block. We extend this construction with parent–leaf pairs so that we have canonical dominating sets that span a connected subgraph.

We are going to add 72 extra disks to every gadget, and 4 "connector" disks between every pair of horizontally or vertically neighboring gadgets, resulting in canonical

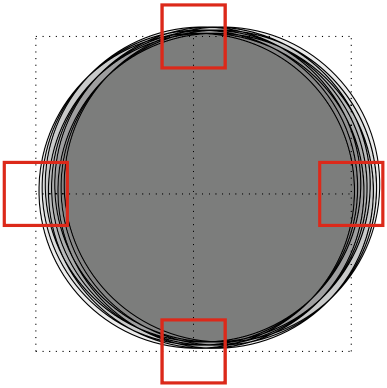


Fig. 4 Circles in a block. The squares intersect every disk in the block

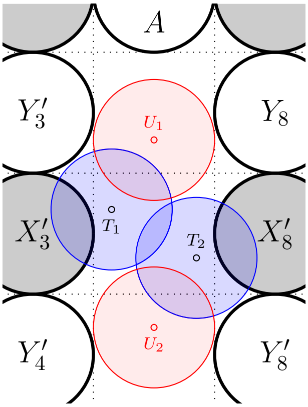


Fig. 5 Connecting horizontally neighboring gadgets

dominating sets of size $16k^2 + 36k^2 + 4k(k-1) = 56k^2 - 4k$ (Note that only the parent disks are included in the canonical set). In other words, the new construction has a connected dominating set of size $56k^2 - 4k$ if and only if there is a feasible grid tiling.

An important property of the blocks that we use is that for a small enough value ϵ, the boundaries of the disks in a block all lie inside a small width annulus - for this reason, the blocks in our pictures are depicted with thick boundary disks. In order for a parent disk p to intersect every disk in a block it is sufficient if the boundary of p crosses this annulus.

Inside any of the blocks, all offsets are in the rectangle with bottom left $(-n^2 - 1, -n^2 - 1)$ and top right (n^2+1, n^2+1). Consequently, every disk in the block with center offset (α, β) where $|\alpha|, |\beta| \in \{0, \ldots, n^2+1\}$ intersects the square with bottom left $\big((-n^2-1)\epsilon, 1-(n^2+1)\epsilon\big)$ and top right $\big((n^2+1)\epsilon, 1+(n^2+1)\epsilon\big)$. There are three similar squares that also have this property, which we can get by rotating the square around the midpoint of the block by 90, 180 and 270 degrees. Consequently, a unit disk that contains such a square intersects all the disks in the given block. For an example with $n = 3$ and $\epsilon = 0.02$ for the block X_2, see Fig. 4.

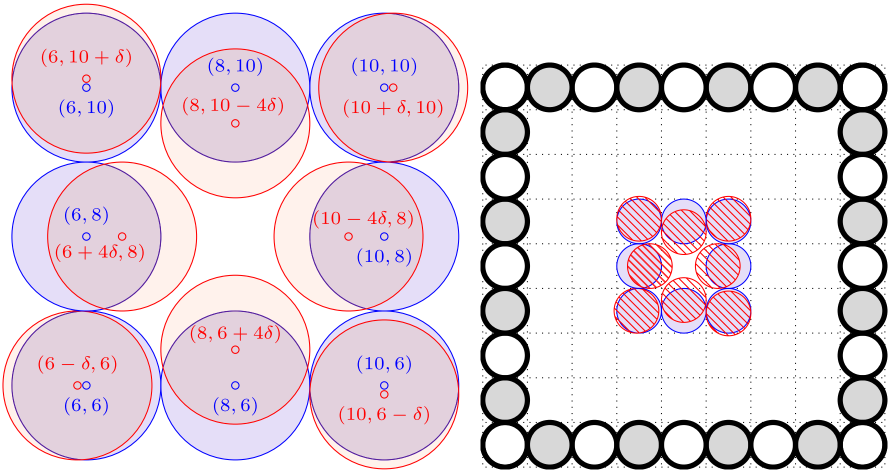


Fig. 6 Left: The circles in the center of every gadget; right: placement inside a gadget

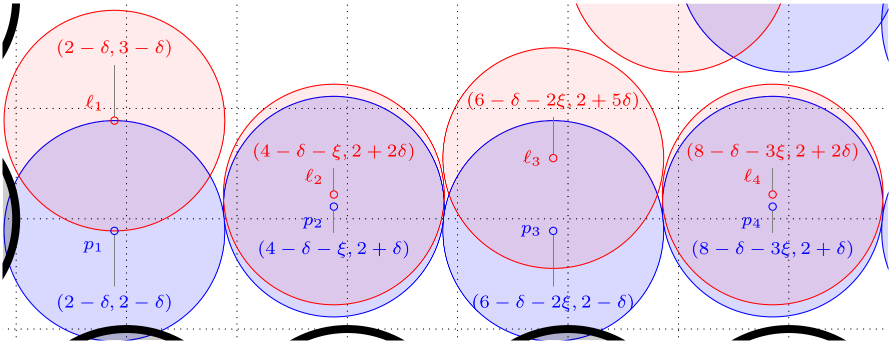

$(2-\delta, 3-\delta)$
ℓ_1
p_1
$(2-\delta, 2-\delta)$
$(4-\delta-\xi, 2+2\delta)$
ℓ_2
p_2
$(4-\delta-\xi, 2+\delta)$
$(6-\delta-2\xi, 2+5\delta)$
ℓ_3
p_3
$(6-\delta-2\xi, 2-\delta)$
$(8-\delta-3\xi, 2+2\delta)$
ℓ_4
p_4
$(8-\delta-3\xi, 2+\delta)$

Fig. 7 The zig-zag arrangement: the first four parent–leaf pairs that connect a side to the center of the gadget

2.4.1 Connecting Neighboring Gadgets

For a pair of horizontally neighboring gadgets, we add two pairs of disks that connect X'_3 from the left gadget to X'_8 in the right gadget. This arrangement is depicted in Fig. 5. The parent disk with center T_1 intersects every disk in the block X'_3 of the left gadget, and the other parent intersects every disk in the block X'_8. The two leaf disks (red disks in the figure) only intersect their parent. Let the origin be the center of the block X'_3 in the left gadget. The coordinates for the disk centers are:

$$T_1 = (1.3, 0.4) \quad U_1 = (2, 1.55)$$
$$T_2 = (2.7, -0.4) \quad U_2 = (2, -1.55)$$

We use a rotated version of these four disks for vertical connections, where the parents connect X'_5 from the upper gadget and X'_2 from the lower gadget.

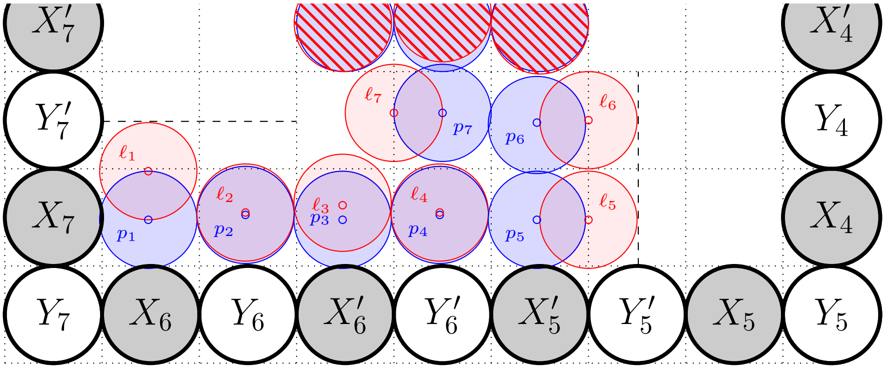


Fig. 8 Connecting one side to the middle

2.4.2 Disks Inside Gadgets

We begin by adding eight disk pairs to the center. The parents are arranged in the vertices and edge midpoints of a square, touching the neighbors. There are four parent disks whose leaf is placed towards the center of the gadget, leaving the portion of their boundary that faces the X- and Y-blocks available for connecting to further parent disks. See Fig. 6 for a picture: the corresponding leaf disks are displayed using a fill pattern of parallel lines.

Let $\delta > 0$ be a small constant to be specified later. From now on, we fix the origin in the center of the bottom left block, Y_7. The disks that are placed in the middle of the gadget have their centers defined below; in each pair we specify the coordinates of a parent and its leaf.

$(6, 6), (6-\delta, 6)$	$(8, 6), (8, 6+4\delta)$	$(10, 6), (10, 6-\delta)$	$(10, 8), (10-4\delta, 8)$
$(10, 10), (10, 10+\delta)$	$(8, 10), (8, 10-4\delta)$	$(6, 10), (6, 10+\delta)$	$(6, 8), (6+4\delta, 8)$

In order to connect the X-blocks, we add parent disks that together connect X_7, X_6, X_6', X_5' to the center; rotated versions of these parent disks will allow us to connect all X-blocks to the center. For this purpose, we are going to use a zigzag pattern of disks. The first parent disk intersects all disks in X_6 and X_7 (i.e., it contains the small squares of X_6 and X_7 that are facing the inside of the gadget). The second parent is above the block Y_6, but it is disjoint from it. The next with center p_3 intersects all disks in X_6', and the disk around p_4 is disjoint from the disks in Y_6'. Finally, the disk around p_5 intersects all disks in X_5'. See Fig. 7 for an example. The leafs follow a more complicated pattern. In our zigzag pattern, two neighboring parents touch each other. We need the centers to have distance 2δ along the y-axis, so the distance along the x-axis is $\sqrt{4-4\delta^2}$. Let $\xi = 2 - \sqrt{4-4\delta^2}$. Note that

$$2 - \delta^2 - \delta^4 < \sqrt{4-4\delta^2} < 2 - \delta^2,$$

so $\delta^2 < \xi < \delta^2+\delta^4$. We add two more disk pairs to this pattern, and some modifications to the leafs. These seven disk pairs are depicted in Fig. 8. Each pair consists of a parent centered at p_i and a leaf centered at ℓ_i; their coordinates are defined as follows.

$$\begin{aligned}
p_1 &= (2-\delta, 2-\delta) & \ell_1 &= (2-\delta, 3-\delta)\\
p_2 &= (4-\delta-\xi, 2+\delta) & \ell_2 &= (4-\delta-\xi, 2+2\delta)\\
p_3 &= (6-\delta-2\xi, 2-\delta) & \ell_3 &= (6-\delta-2\xi, 2+5\delta)\\
p_4 &= (8-\delta-3\xi, 2+\delta) & \ell_4 &= (8-\delta-3\xi, 2+2\delta)\\
p_5 &= (10-\delta-4\xi, 2-\delta) & \ell_5 &= (11, 2-\delta)\\
p_6 &= (10-\delta-4\xi, 4-\delta) & \ell_6 &= (11, 4)\\
p_7 &= (8, 4+3\delta) & \ell_7 &= (7, 4+3\delta)
\end{aligned}$$

By analyzing the coordinates carefully, it can be verified that only the intended intersections arise among these seven disk pairs and the central eight disk pairs. Also not that the seven disk pairs are disjoint from the segment $(12+\delta/2, 1)(12+\delta/2, 5)$ and also from the segment on egets by rotating this by 90 degrees around $(8, 8)$, namely $(1, 4-\delta/2)(5, 4-\delta/2)$.

Our final gadget can be attained by rotating the above seven disk pairs around the center $(8, 8)$ by 90, 180 and 270 degrees: see Fig. 9. We added the spanned edges of a canonical dominating set to this picture.

We can now finish the proof of Theorem 1.

A feasible grid tiling defines k^2 values $u_{a,b}$ for $(a, b) \in [k]\times[k]$. We can use this to define $16k^2$ disks in our blocks. Recall that f is the function assigning pairs of integers to a single index: for $1 \leqslant x, y \leqslant n$, $f(x, y) = (x-1)n + y$. In the gadget with index (a, b), we include the disks $X_\ell\left(f(u_{a,b})\right)$ and $X'_\ell\left(f(u_{a,b})\right)$ for all $\ell = 1, \ldots, 8$. We add all parent disks of the construction, this results in a connected dominating set of size $56k^2 - 4k$. In the other direction, if there is a connected dominating set of size $56k^2 - 4k$, then there is a canonical dominating set of the same size, whose disks inside X-blocks and X'-blocks define a feasible grid tiling. Thus, it is sufficient to prove that the intersection patterns are as described.

It can be verified using the coordinates that our final leaf disks only intersect their parent disk, and also that the parent disks form a connected subgraph both inside gadgets and at every connection. We need to show that the parents inside the gadget connect all the X-blocks of the gadget, and that the horizontal and vertical connectors intersect the two X-blocks that they need to connect. In all of these cases, it is sufficient to show that the parent disk contains one of the four squares that we associated with each block. For connector disks, it is easy to see that the center of one of the four squares is covered by the interior of the corresponding parent disk (i.e., the square around $(1, 0)$ is contained in the interior of disk(T_1)). By choosing a small enough value for ϵ, the square is contained in the parent disk.

For the inner connections of gadgets, it is sufficient to show that the inner squares of X_7, X_6, X'_6 and X'_5 are contained in disk(p_1), disk(p_1), disk(p_3) and disk(p_5) respectively: the other sides have the same containments since the rotation around $(8, 8)$ by 90, 180 and 270 degrees are automorphisms on the small squares.

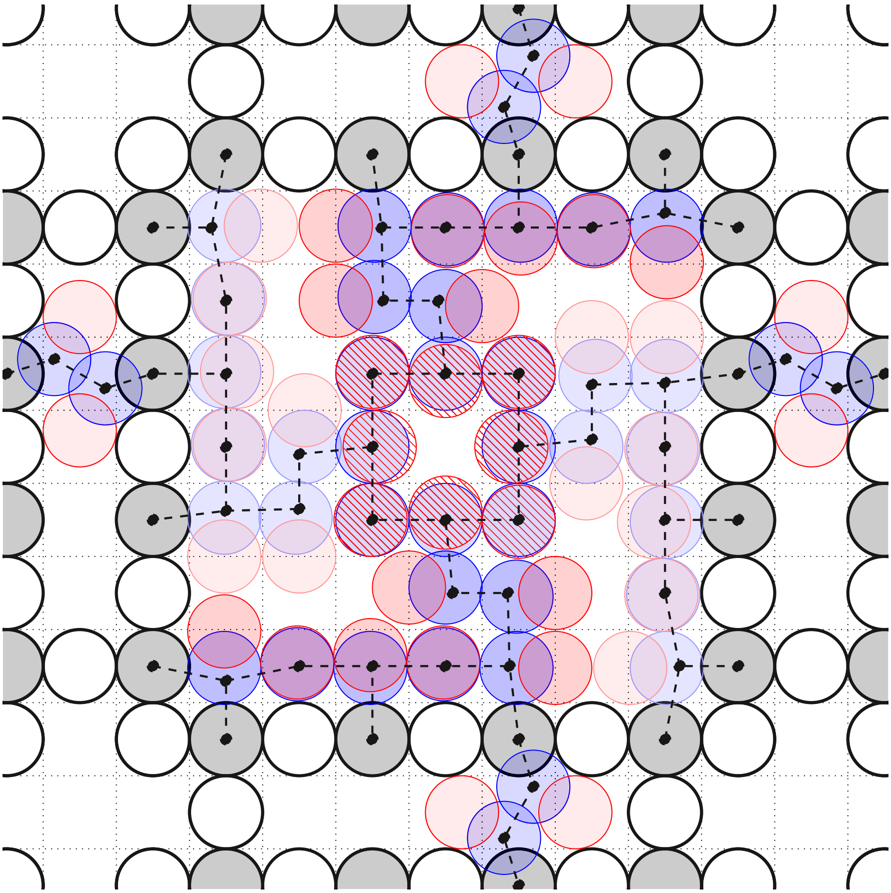


Fig. 9 A gadget in the final construction. The dashed lines are spanned edges of a canonical dominating set

Concentrating on disk(p_1) now, notice that $p_1 = (2-\delta, 2-\delta)$ contains the right hand side square of X_7 if and only if it contains the top square of X_6 by the symmetry on the line $x = y$. Furthermore, observe that p_1 and the center of the top square of X_6 are closer to each other than p_3 and the center of the top square of X'_6, which are in turn closer than p_5 and the center of the top square of X'_5. (This follows from the fact that the differences in y-coordinates are the same but are increasing in x-coordinates.) Therefore, if we can show that disk(p_5) contains the top square of X'_5, than all the other desired intersection must also be present. The farthest corner of this square from p_5 is $\big(10 + (n^2+1)\epsilon, 1-(n^2+1)\epsilon\big)$. Let $\epsilon < \frac{1}{2n^3}$ and $\delta < 1$. The distance squared from p_5 has to be at most 1:

$$\Big(10 - \delta - 4\xi - (10 + (n^2+1)\epsilon)\Big)^2 + \Big(2 - \delta - (1 - (n^2+1)\epsilon)\Big)^2$$
$$< \left(\delta + 4\delta^2 + \delta^4 + \frac{1}{n}\right)^2 + \left(1 - \delta + \frac{1}{n}\right)^2$$

$$= 1 - 2\delta + \frac{4}{n} + O\left(\frac{\delta}{n}\right) + O(\delta^2)$$

Let $\delta = \frac{1}{\sqrt{n}}$. For n large enough,

$$1 - 2\delta + \frac{4}{n} + O\left(\frac{\delta}{n}\right) + O(\delta^2) \;=\; 1 - \frac{2}{\sqrt{n}} + \frac{4}{n} + O\left(\frac{1}{n\sqrt{n}}\right) + O\left(\frac{1}{n}\right) < 1.$$

Since $\delta > 1/n > (n^2+1)\epsilon$, we can also observe that p_2 and p_4 are disjoint from the y-blocks, since their projection on the y-axis is disjoint from the projection of the top squares of these blocks.

Note that the coordinates of each point can be represented with $O(\log n)$ bits, since a precision of c/n^4 is sufficient for the construction. Since Grid Tiling has no $n^{o(k)}$ algorithm under ETH and the above is a parameterized reduction leading to a parameter $16k^2$, there can be no $n^{o(\sqrt{k})}$ algorithm for Connected Dominating Set in Unit Disk Graphs unless ETH fails. Finally, the containment in W[1] is a simple consequence of the proof for Dominating Set in Unit Disk Graphs in [10]. The key point of that proof is that a dominating set can be verified with a tail-deterministic machine; in our case, we only need to add a connectivity check on the solution set to the end of the Dominating Set verifier program. This concludes the proof for Connected Dominating Set. To prove W[1]-hardness and the same lower bound for the broadcast problem, we can let one of the blue parent disks be the source disk: in this way, the minimum broadcast sets equal the minimum connected dominating sets. Finally, the containment in W[1] needs an extra check that the source s is in the set. This concludes the proof of Theorem 1. □

Since the whole construction fits in a strip of width $w = O(k)$, we also get the following corollary.

Corollary 6 *The broadcast problem in strips is* W[1]*-hard parameterized by the strip width w. Moreover, there is no $f(w)n^{o(w)}$ algorithm for it, unless the Exponential Time Hypothesis fails.*

3 The Hardness of *h*-Hop Broadcast in Wide Strips

The goal of this section is to prove Theorem 2.

Our reduction is from 3-SAT. Let $x_1, x_2, \ldots x_n$ be the variables and let $C_1, \ldots, C_m$ be the clauses of a 3-CNF formula. Note that in the construction we are using two types of wires to transport information called tapes and strings, as we will define later formally.

3.1 Proof Overview

Figure 10 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

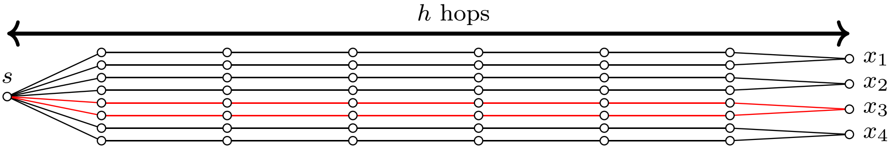
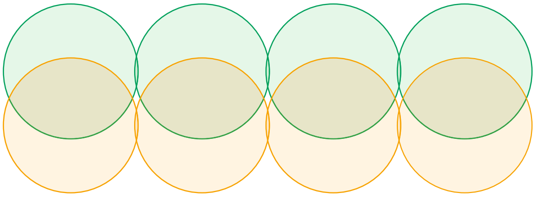


Fig. 10 The gadget representing the variables. The red paths form the x_3-string (Color figure online)

Fig. 11 Disk pairs of a string



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. (The level of a point is its distance from the source in the unit disk graph corresponding to the complete construction.) 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* (see Fig. 11). 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*.

The clause gadgets all start and end in the base bundle, as shown in Fig. 12. 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. 12, where the strings for x_i, x_j, and x_k are drawn in red, blue, and green respectively.

First we split off the top $i - 1$ strings from the base bundle, by letting the top $i - 1$ strings of the base bundle turn left (in Fig. 12 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 first $i - 1$ strings. 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 path) of the x_i-string was selected to be part of the broadcast set initially,

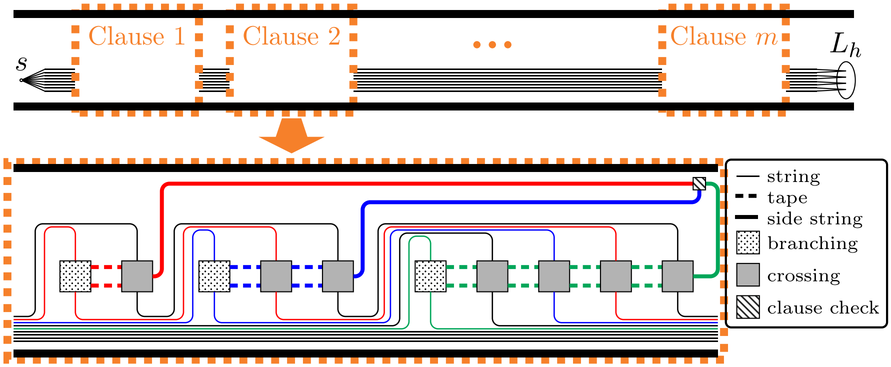


Fig. 12 The overall construction, and the way a single clause is checked. Note that in this figure each string (which actually consists of two paths) is shown as a single curve. The figure corresponds to a clause with $i = 2$ (red), $j = 3$ (blue) and $k = 5$ (green) (Color figure online)

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 (See Fig. 12). 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.

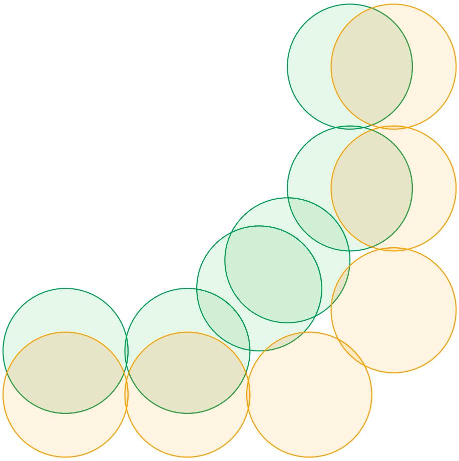
The final construction contains $\Theta(n^3 m)$ points that all fit into a strip of width 40.

In order to simplify our discussion and figures, we scale the input such that a can broadcast to b if their unit disks intersect (or equivalently, if their distance is at most 2).

3.2 Handling Strings and Bundles

We start the initial bundle directly from the source, and end each string with a disk that intersects the last true and false disk of the given variable, as already seen in Fig. 10. (A TRUE disk is a disk on a TRUE path, a FALSE disk is a disk on a FALSE path.) A minimum-size solution of this bundle contains the source disk and true or false disks from each of the true–false disk pairs in all the strings. In the final construction, once all the clause checks are done and the strings have returned to the bottom bundle, we are going to add some extra levels so that the h-hop restriction does not interfere with the last side strings. (This can be done by for example doubling the maximum distance of a side string ending from s.) The disks of a given level in a bundle lie on the same vertical line, at distance $\frac{1}{2n}$ from each other, so for a bundle containing all the variables, the disk centers on a given level fit on a vertical segment of length 1, and the whole bundle fits within width 3.

Fig. 13 A turn of 90 deg in a side string or string outside a bundle using constant horizontal space



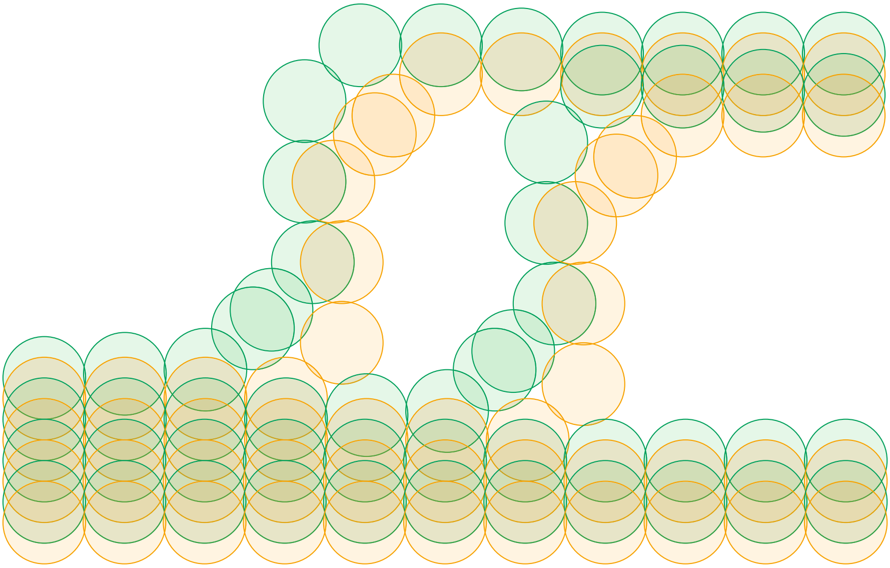


Fig. 14 Splitting the top 2 strings off a bundle of 4 strings: we peel the top layers one by one

Bundled strings are in lockstep, i.e., a pair of intersecting disks in the bundle that are not in the same string and truth value are on the same level. We call this the *lockstep condition*.

Next, we describe some important aspects of handling strings, bundles and side strings. First, we show that we can do turns with strings in constant horizontal space, and do turns in bundles in polynomial horizontal space. An example of a string turn can be seen in Fig. 13.

This turning operation can be used on the top string of a bundle to "peel" off strings one by one and unify them later in a new bundle, see Fig. 14. This is how we can split and turn a bundle: we peel and turn the strings one by one. Notice that the lockstep condition is upheld both in the bottom and top bundle. It requires $O(n)$ extra horizontal space and $O(n^2)$ disks to split a bundle with this method.

If we were to return the strings one by one to the bottom bundle without correction as depicted in Fig. 12, the returning strings would be at a larger hop distance from the source compared to the strings of the bottom bundle with the same x-coordinates, so the new unified bundle would violate the lockstep condition. To avoid this issue, we use a correction mechanism. We have some room to squeeze bundle levels horizontally. The largest horizontal distance between neighboring levels is 2; for the smallest distance, we need to make sure that a disk does not intersect other disks from neighboring levels other than the disks in the same string with the same truth value. So the horizontal distance has to be at least $2\sqrt{1-\left(\frac{1}{4n}\right)^2} < 2-\frac{1}{15n^2}$. Thus, if we have $15n^2$ compressed levels in a bundle, then they take up the same horizontal space as $15n^2-1$ maximum distance levels.

Note that for each extracted side string, it is sufficient to apply the correction mechanism $O(1)$ times (i.e., to gain $O(1)$ steps) since the strings returning to the bundle are shifted by the same number of levels. Therefore the correction mechanism adds $\Theta(n^2)$ levels to the construction for each side string. There are $\Theta(m)$ side strings, so we get $\Theta(n^2m)$ levels from this source. Also note that for the correction mechanism to function, we require that the coordinates are specified up to an error of $O(n^{-2})$.

A detour of a string (peeling off, going through a gadget, returning to the bottom bundle) requires a constant number of extra levels to achieve, we can compensate for this with the addition of a polynomial number of extra disks. Before a string peels off from the top bundle downward to rejoin the bottom bundle, we add $15n^2k$ compressed levels to the top bundle and $(15n^2-1)k$ maximum distance levels to the bottom bundle, if the total number of extra levels added by turning up, going through the gadget and turning down is k. This ensures that the lockstep condition is upheld in the bottom bundle after the return of this string. For each string that leaves the bottom bundle and later returns, we use this correction mechanism. Overall, this correction mechanism is invoked a polynomial number of times, so requires a polynomial number of disks.

3.3 Tapes

Our tapes consist of *switches*: a tape switch is a collection of three disks, the centers of which lie on a line at distance ϵ apart—so it is isometric to the old connector blocks A, B, C and D for the case "n"$= 2$ (see Sect. 2). Denote the three disks inside a tape switch T^k by δ_1^k, δ_2^k and δ_3^k. We can place multiple such switches next to each other to form a tape. An example is depicted in Fig. 15.

The tapes always connect blocks in which disks have truth values assigned, e.g., the end of a string or disks of a gadget block. Denote the starting true and false disks by F and the ending true and false disks by G. We say that a set of tape switches $T^1, T^2, \ldots, T^p$ forms a tape from F to G if it satisfies the following conditions.

- In the first switch, δ_1^1 intersects both the true and false disk(s) of F, δ_2^1 intersects the true disk(s) of F, and δ_3^1 is disjoint from both the true and false disk(s).
- δ_i^k intersects the disk δ_j^{k+1} if and only if $j \leqslant i$ $(k = 1, \ldots, p-1)$.

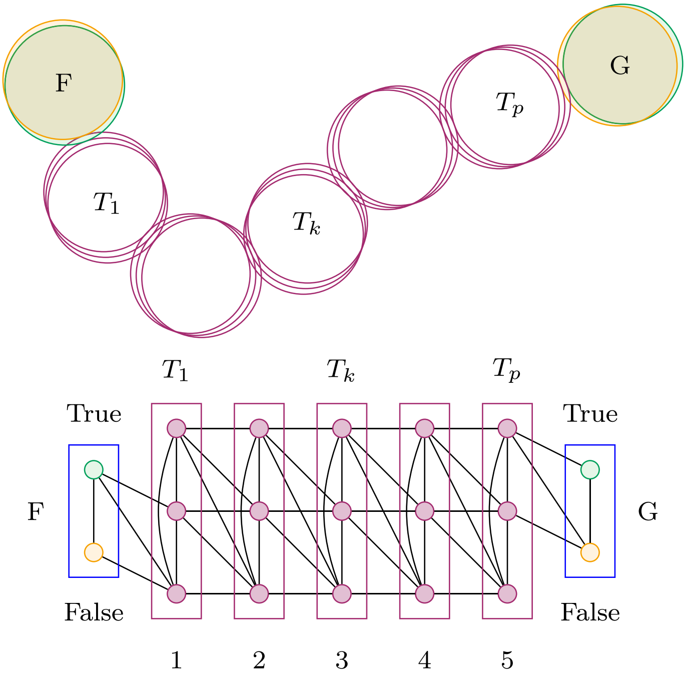


Fig. 15 Tape switches connecting two true–false disk pairs and the corresponding subgraph

- In the last switch, δ_1^p is disjoint from G, δ_2^p intersects the false disk(s) of G, and δ_3^p intersects both the true and false disk(s).
- Non-neighboring tape switches are disjoint, F is disjoint from all switches except the first, and G is disjoint from all switches except the last.

We would like to examine the active disks of a tape within a minimum broadcast set. We say that a switch is *empty* if it has no active disks.

Lemma 7 *Let T be a tape from F to G that has p tape switches. Every h-hop broadcast set contains at least $p-1$ disks from the tape. If a broadcast set contains exactly $p-1$ disks, then it cannot happen that the active disk in F is a false disk and the active disk in G is a true disk.*

Proof Let the tape switches be $T^1, T^2, \ldots, T^p$. If there are at most $p-2$ active disks, then there are at least two empty switches. These switches have to be neighboring, otherwise a disk in between the two switches is impossible to reach from the source. Let these switches be T^k and T^{k+1}. All disks in T_k must be reached through the block F and the switches $T^1, \ldots, T^{k-1}$. Specifically, δ_3^k has to be reached. The shortest path to this point from any F-disk requires at least k tape disks. Similarly, the shortest path from any G-disk to δ_1^{k+1} requires at least $p-k$ disks. Overall, at least p active disks of the tape are required to reach these disks—this is a contradiction.

If the tape contains $p-1$ active disks, then both F and G must contain an active disk, otherwise there would be a component inside the tape that is not connected to the source. Suppose for the purpose of contradiction that the active disks of F are false and the active disks of G are true. There is at least one tape switch that has no active disk; let T^k be such a switch, where k is as small as possible. Since δ_2^k has to be covered, it has to be reached either from F or G.

Suppose that δ_2^k is reached through F; this requires k active disks from the tape switches. We have only $p - 1 - k$ active disks for the rest of the $p - k$ switches $T^{k+1}, \ldots, T^p$, so there has to be another empty tape switch, T^ℓ $(\ell > k)$. As previously mentioned, we cannot have non-neighboring empty switches, so the other empty switch is T^{k+1}. This means that δ_3^k also has to be dominated from the left side, the shortest path to which requires $k + 1$ active tape disks from any false disk of F. This leads to an additional empty switch among $T^{k+1}, \ldots, T^p$. But as shown above, there can be at most one such switch (T^{k+1})—we arrived at a contradiction. A similar argument (actually, it is even slightly easier) works for the case when δ_2^k is reached from G. □

3.4 Gadgets and their Connection to Tapes and Strings

3.4.1 Crossing and Branching Gadgets

Our crossing gadget and our branching gadget are almost identical to the one used in the W[1]-hardness proof of Connected Dominating Set in Unit Disk Graphs. This gadget can be used to transmit information both horizontally and vertically simultaneously—this is exactly what we need. Since we only need to transmit truth values, we take the gadget for "n"= 2, resulting in X blocks with $2 \cdot 2$ and Y-blocks with $2 \cdot 2 + 1$ disks. The only change we make in the crossing gadget is that we swap the X_1 and X_2 blocks. That is, the new $\overline{X}_1$ has the same center as X_1 had but the offsets are defined as in X_2, and the new $\overline{X}_2$ has the same center as X_2 had but the offsets are defined as in X_1. The modification is necessary for proper connection with vertical strings.

For the branching gadget, we modify some offsets so that we can transmit the vertical truth value on the right side of our gadget. For this purpose, we change the offsets in X_3 and X_4 the following way.

$$\text{offset}(\overline{X}_3(j)) = (-\iota_2(j), -j) \qquad \text{offset}(\overline{X}_4(j)) = (\iota_2(j), -j)$$

In case of these horizontal connections, we say that a disk $X_k(j)$ from the block X_k is a true disk if $\iota_1(j) = 2$ and a false disk if $\iota_1(j) = 1$. Similarly, for vertical connections, a disk $X_\ell(j)$ is a true disk if $\iota_2(j) = 2$ and a false disk if $\iota_2(j) = 1$.

3.4.2 Connecting Gadgets with Tapes and Strings

When connecting branching and crossing gadgets or two crossing gadgets with tapes horizontally, we are going to add a tape that goes from the X_4 (or $\overline{X_4}$) block of the left gadget to the X_7 block of the right gadget, and a tape that goes from the X_8 block of the right gadget to the X_3 (or $\overline{X}_3$) block of the left gadget. Note that in the W[1]-hardness proofs, we used the same strategy with tapes consisting of only one switch.

In this case, we place the first and last switch of each tape at the same location as the connector block in the proof of Theorem 1, and use some tape switches in between these, the number of which will be specified later. Note that this placement gives us a tape that is consistent with the definition of true and false disks in the X-blocks.

In order to connect strings and side strings to the gadgets, we use both tapes and parent–leaf pairs. Figure 16 depicts a connection to a crossing gadget from the top and bottom.

We need to connect both “sides” of the string: on the top, we use a tape from X_2 to the last string block, and a tape from the last string block to X_1. Moreover, in order to make sure that all the disk pairs of the strings are in use, and the connection is not maintained through the tapes, we create a short path to the gadget with some disks that are guaranteed to be in the solution. This path consists of parent and leaf disk pairs, where all the parents will be inside a canonical solution—we used this technique before inside the gadgets to ensure gadget connectivity. The string exits the gadget similarly. Note that the shortest path through the gadget from the string end on the top to the string end on the bottom has length 18, and its internal vertices are all parent disks, a disk from X_1' and a disk from X_5'; the paths using any of these tapes are longer.

We use the same type of connection to connect side strings to the right side of the last crossing gadget (or to the branching gadget, if the current clause contains the first variable). The complete gadget together with the connections and string turns fits in 50 units of vertical space. (Recall that all distances have been scaled by a factor of two, so that we have unit radius disks.)

We briefly return to the tape pairs that connect neighboring gadgets. We need to make sure that the tapes do not provide a shortcut—we want the shortest path from source to the last level h to be through string blocks, and to go through gadgets as discussed above. When choosing a tape length, we also need to bridge the distance between neighboring gadgets. Note that this amount can be polynomial in n because of the correction mechanism for strings. We add a small detour to make sure that the shortest path to a gadget that uses a tape is longer than the shortest path that uses only the string that enters the gadget. It is easy to see that there is enough place for such a detour: taking twice the amount of switches that would be necessary to cover the distance is enough. A tape connection between neighboring gadgets is depicted in Fig. 17. (Note that these tapes need no additional vertical space: they fit easily in the 18 units of vertical space between the gadgets.)

Lemma 8 *In a canonical solution, the crossing gadget has the following properties. The last pair of the vertical string on the top carries the same truth value as the first pair of the vertical string on the bottom, and there is a path of 18 hops between them consisting of parent disks, a disk from X_1' and a disk from X_5'. Moreover, the pair of tapes on the left hand side carry the same value as the pair of tapes on the right hand side. If the right hand side connects to a side string instead of a pair of tapes, then the truth value selected in the first pair of the side string is the same as the truth value carried by the pair of tapes on the left side.*

Proof *(sketch)* A canonical solution contains exactly one active disk from each X-block, it contains all the parent disks, and has exactly one active disk from each true–false disk pair of a string. The path of 18 hops can indeed be found through

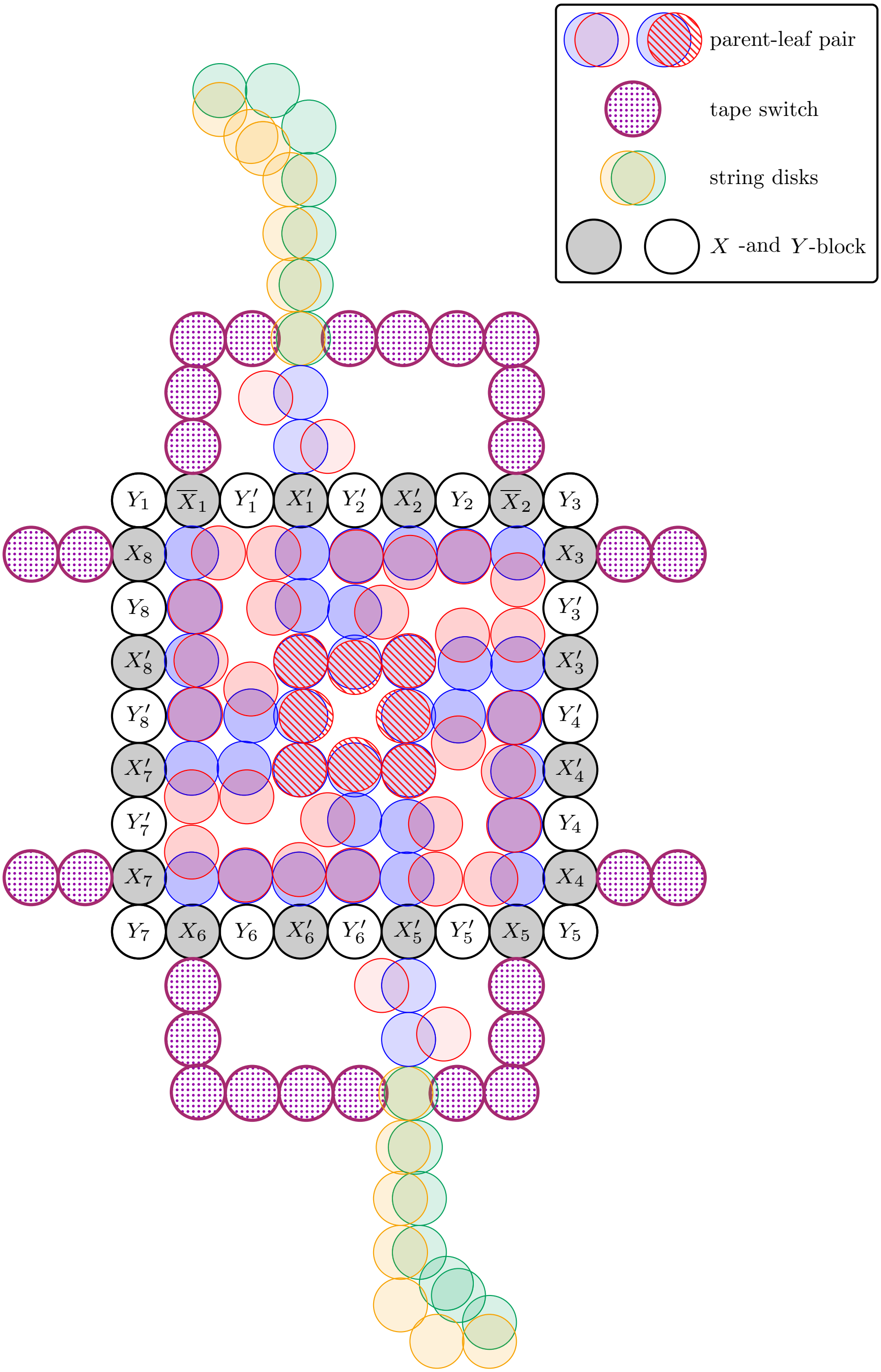


Fig. 16 Connecting to a crossing gadget from the top and bottom

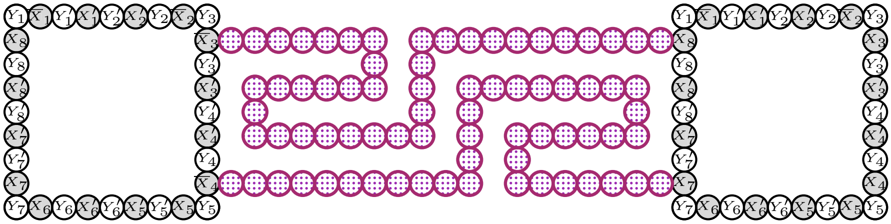

Y_1 $\overline{X}_1$ Y_1' X_1' Y_2' X_2' Y_2 $\overline{X}_2$ Y_3
X_8 Y_8 X_8' Y_8' X_7' Y_7' X_7
$\overline{X}_3$ Y_3' X_3' Y_4' X_4' Y_4 $\overline{X}_4$
Y_7 X_6 Y_6 X_6' Y_6' X_5' Y_5' X_5 Y_5
Y_1 $\overline{X}_1$ Y_1' X_1' Y_2' X_2' Y_2 $\overline{X}_2$ Y_3
X_8 Y_8 X_8' Y_8' X_7' Y_7' X_7
X_3 Y_3' X_3' Y_4' X_4' Y_4 X_4
Y_7 X_6 Y_6 X_6' Y_6' X_5' Y_5' X_5 Y_5

Fig. 17 Connecting neighboring gadgets with tapes

these disks. Regarding the truth values, in a canonical broadcast, we have the disks $\overline{X}_1(j), \overline{X}_2(j), X_5(j), X_6(j)$ selected for some $j \in [n^2]$. If the last true–false disk pair on the top is true, then the disk in $\overline{X_1}$ cannot be false by the properties of tapes, so $\iota_2(j) = 2$. Consequently, X_5 is "true", and therefore the first true–false pair on the bottom has to be set to true. The pairs of tapes on the right and left hand side carry the truth value encoded by ι_1 just as the connector blocks A and B did in the Connected Dominating Set construction. If the right hand side connects to a side string, then notice that in order to get canonical tapes, at least one of the disks in the first pair of the side string has to be active (as otherwise both of the tapes on the right hand side would need an extra disk). The truth value carried in this first pair has to equal true if and only if $\iota_1 = 2$ in the gadget. □

A similar argument yields the following lemma.

Lemma 9 *In a canonical solution, the branching gadget has the following properties. The last pair of the vertical string on the top carries the same truth value as the first pair of the vertical string on the bottom, and the same as the value carried by the pair of tapes on the right. If the right hand side connects to a side string instead of a pair of tapes, then the truth value selected in the first pair of the side string is the same as the truth value carried by the vertical string. There is a path of 18 hops between the last pair of the top and the first pair of the bottom string consisting of parent disks, a disk from X_1' and a disk from X_5'.*

3.4.3 The Clause Check Gadget

The clause check gadget is very simple, it contains four well-placed disks: one at the end of each of the three side strings, and one disk that only intersects the three other clause check disks. We turn the three side strings towards their corresponding disks so that the side strings do not interfere with each other. Among the six last disks at the end of the three side strings only the ones corresponding to the literals of this clause intersect the gadget. The rest of the side string disks are disjoint from the gadget. See Figs. 18 and 19 for an example of checking $(x_2 \vee x_3 \vee \bar{x}_5)$. The vertical space required is less than 20 units.

Our complete construction can fit in 80 units of vertical space. Ten units can accommodate the lower bundle and turning strings up and down from it; 50 units of vertical

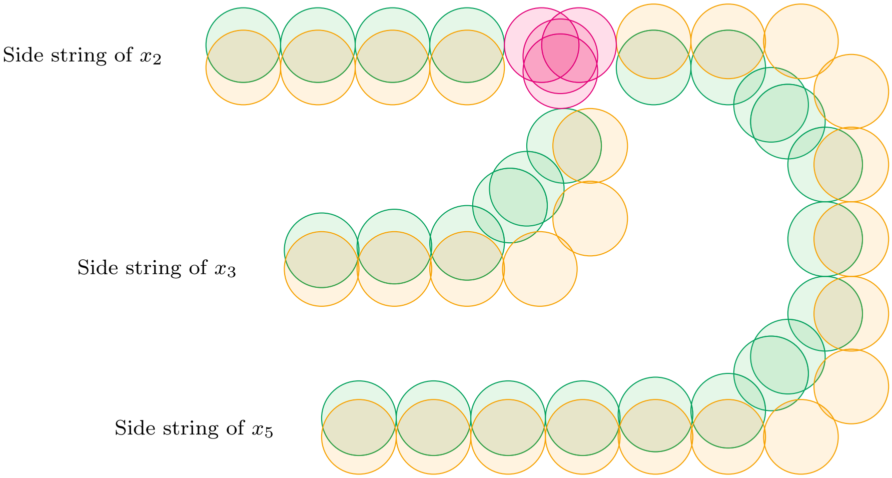


Fig. 18 Clause check gadget for the clause $(x_2 \vee x_3 \vee \bar{x}_5)$

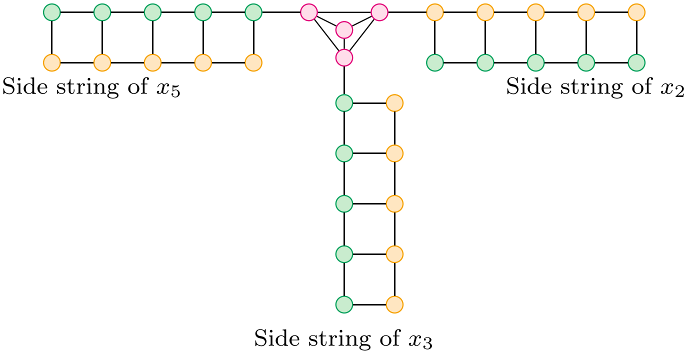


Fig. 19 The subgraph spanned by the disks of a clause check gadget and its surroundings

space can accommodate the branching and crossing gadgets, along with their connections and tapes. We need ten units for the bundle that goes above the gadgets (along with the string turns), and finally 20 more for the side strings and the clause check gadget. Recall that we did a scaling by two to switch to the intersection model of broadcasting. In the original model of broadcasting, the construction occupies 40 units of vertical space.

In case of a satisfiable formula, we can choose the disks in each side string that correspond to the value of the variable, and choose a disk from the clause check gadget that intersects a true literal (at least one of the literals is true in the clause).

This lemma describes the usage of the clause check gadgets and the side strings. We say that a true–false disk pair is an empty pair if none of its disks are active.

Lemma 10 *Let ϱ be the number of true–false disk pairs in the three side strings that correspond to a particular clause checking gadget. An h-hop broadcast set contains at least $\varrho + 1$ disks from the three side strings and the clause check gadget. Moreover,* *if an h-hop broadcast set has exactly $\varrho + 1$ actives among these disks, then the truth values chosen at the beginning of the side strings satisfy the clause.*

Proof We prove the following claim first.

Claim There is an optimal h-hop broadcast set where each side string contains at most one empty pair.

Proof of Claim Suppose that U_k and U_ℓ are two empty pairs in a side string. If they are not neighboring, then a disk between them is unreachable from the source. So $\ell = k + 1$. Consequently, both disks of U_k are dominated from the start of the side string, and both disks of U_{k+1} are dominated from the end. Since the side string has length more than four, either $k > 2$ or $k < p - 1$. Suppose $k > 2$, the other case is similar. The only way to reach both disks in U_k is to have both of the disks in U_{k-1} active. Since U_{k-1} is also reached from the left, there is an active disk in U_{k-2}; let its truth value be v. So we can deactivate the disk in U_{k-1} of value $\neg v$ and activate the disk in U_k of value v. This way every disk that has been dominated remains dominated, and the number of active disks does not increase. (Note that we do not need to worry about exceeding h hops since h will be chosen large enough to not interfere with side strings.) □

Now we show that every h-hop broadcast set includes at least $\varrho + 1$ disks from these side strings and the clause check gadget. Suppose there is a side string of p pairs that contains an empty pair U_k, and let v be the truth value of the disk in the last pair U_p. Suppose $2 \leqslant k \leqslant p - 1$; a small variation of the argument applies to the cases $k = 1$ and $k = p$. In U_{k-1}, the disk of value $\neg v$ has to be reached from the beginning of the string—the shortest path requires at least $k - 2$ active disks in $U_1, \ldots, U_{k-2}$. In U_{k+1}, the disk of value $\neg v$ has to be reached through the clause check gadget; this requires that the clause check disk corresponding to this side string is active, and there are at least $p - k$ side string actives from $U_{k+1}, \ldots, U_p$, since we also need to change truth value along the way. Additionally, the disk of value $\neg v$ in U_k has to be reached from one of the neighboring pairs, requiring $U_{k-1}(\neg v)$ or $U_{k+1}(\neg v)$ to be active. Overall, either a side string does not contain an empty pair (so it has at least p disks), or we needed $k - 2 + (p - k) + 1 + 1 = p$ active disks from the side string and the corresponding clause check disk. Moreover, at least one of the three side strings needs to connect the middle point of the clause check gadget to the source: the shortest path through a side string of p pairs has $p + 1$ inner vertices, since it has to include one disk from each pair of the side string and the clause check disk corresponding to this side string. Consequently, we need at least $\varrho + 1$ active disks.

Finally, we need to show that if the assignment at the beginning of the side strings does not satisfy the clause (all literals are false), then we need at least $\varrho + 2$ active disks. A similar argument shows that the side string reaching the middle disk of the clause check gadget must have one extra active disk. □

3.5 Reduction from 3-SAT

We examine the disks that are necessarily part of a minimum broadcast set if the formula is satisfiable. It will be apparent that a solution of the same size cannot exist if the formula is not satisfiable.

We include all the disks from the strings and side strings that correspond to the value given to the variable. Add the disks from the gadgets: the blue parent disks inside and one disk from each X-block. The branching and crossing gadget connections require four blue parent disks outside the gadget at the top and bottom connection, and two more disks on the last gadget (where the side string begins). In each tape we include one disk from all of its switches except one. Finally, we use one disk to cover each clause check gadget, and we include the source disk. Let k be the number of disks listed here.

Lemma 11 *There is a minimum h-hop broadcast set containing k disks if and only if the original 3-CNF formula is satisfiable.*

Proof As we demonstrated previously, if the formula is satisfiable, then there is an h-hop broadcast set of the given size. We need to show that if there is an h-hop broadcast set of this size, then the formula is satisfiable. Take a minimum h-hop broadcast set. First, we know that the shortest path to the string ending disks requires exactly h hops, and the only path of this length includes all pairs of the string in question, plus the shortest way through the gadgets in which this string is involved. By Lemmas 8 and 9, the shortest way through a gadget from the string end on the top to the string end on the bottom uses only blue disks, and one disk from X'_1 and X'_5 each. Without loss of generality we can suppose that the h-hop broadcast set restricted to each gadget is canonical by the analogue of Lemma 4. A minimum h-hop broadcast set must include an active disk for all switches except one per tape, and for each clause i, at least $\varrho_i + 1$ active disks as shown by Lemmas 7 and 10, where ϱ_i is the number of true–false disk pairs in the three side strings that correspond to a clause. So a minimum h-hop broadcast set does indeed require at least k disks. An h-hop broadcast set of this size that is canonical when restricted to each gadget also means that the truth value carried by a string before entering a gadget is the same as the truth value carried after exiting the gadget by Lemmas 8 and 9. Similarly, the truth values are correctly transferred between neighboring gadgets connected by a tape pair: this can be seen by applying Lemma 7 for both tapes. And finally, all clauses must have a true literal at the beginning of at least one of the corresponding side strings by Lemma 10. Since the disk choice at the beginning of a side string is forced to comply with the corresponding string, it follows that the truth values defined by the strings satisfy the formula. □

Our construction can be built in polynomial time—note that the coordinates of each point can be represented with $O(\log n)$ bits, since a precision of c/n^2 is sufficient for some small constant c. It is easy to see that the eventual hop bound h is dominated by the correction mechanism that we have introduced to preserve the lockstep condition of the bundles, therefore $h = O(n^2m)$. Since neighboring disks have distance $\Omega(1/n)$ and the strip has constant width, it follows that the number of disks used is $\Theta(nh) =$

$\Theta(n^3m)$, and this is also asymptotically equal to the number k of disks in a minimum broadcast set. Finally, note that the reduction takes $O(n^3m)$ time.

We have successfully reduced 3-SAT to the h-hop broadcast problem in a strip of width 40. Since the problem is trivially in NP, this concludes the proof of Theorem 2.

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 have a dichotomy for the complexity when parameterized by strip width. We have also proved that the problem (and, similarly, Connected Dominating Set in Unit Disk Graphs) is W[1]-complete when parameterized by the solution size. The problem of finding an h-hop broadcast set seems even harder: we can solve it in polynomial time for $h = 2$ (see Part I) but already for $h = 3$ we know no better algorithm than brute force. Interesting open problems include:

- What is the complexity of 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?
- Is the broadcast problem NP-complete already in a strip of width $\sqrt{3}/2 + \epsilon$?
- Is there an efficient approximation or perhaps approximation scheme for h-hop broadcast?
- What is the computational complexity of the weighted version of the problem, where each disk has some nonnegative cost associated with it, and we are looking for the cheapest (h-hop) broadcast? Are the problems still solvable in polynomial time if we stay in a strip of width at most $\sqrt{3}/2$?

References

1. de Berg, M., Bodlaender, H.L., Kisfaludi-Bak, S.: The homogeneous broadcast problem in narrow and wide strips I: algorithms. Algorithmica (submitted)
2. Alzoubi, K.M., Wan, P., Frieder, O.: Message-optimal connected dominating sets in mobile ad hoc networks. In: Proceedings of the 3rd ACM Interational Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc 2002, June 9–11, 2002, pp. 157–164. ACM, Lausanne (2002). https://doi.org/10.1145/513800.513820
3. Kuhn, F., Wattenhofer, R., Zhang, Y., Zollinger, A.: Geometric ad-hoc routing: of theory and practice. In: Borowsky, E., Rajsbaum, S., (eds.) Proceedings of the 22nd ACM Symposium on Principles of

Distributed Computing, PODC 2003, Boston, Massachusetts, USA, July 13–16, 2003, pp. 63–72. ACM, New York (2003). https://doi.org/10.1145/872035.872044

4. Yu, J., Wang, N., Wang, G., Yu, D.: Connected dominating sets in wireless ad hoc and sensor networks—a comprehensive survey. Comput. Commun. **36**(2), 121–134 (2013). https://doi.org/10.1016/j.comcom.2012.10.005
5. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman, San Francisco (1979)
6. Lichtenstein, D.: Planar formulae and their uses. SIAM J. Comput. **11**(2), 329–343 (1982). https://doi.org/10.1137/0211025
7. Masuyama, S., Ibaraki, T., Hasegawa, T.: The computational complexity of the m-center problems on the plane. IEICE Trans. **64**(2), 57–64 (1981)
8. de Berg, M., Bodlaender, H.L., Kisfaludi-Bak, S., Marx, D., van der Zanden, T.C.: A framework for ETH-tight algorithms and lower bounds in geometric intersection graphs. In: Proceedings of STOC 2018, pp. 574–586. ACM, New York (2018). https://doi.org/10.1145/3188745.3188854
9. Marx, D.: Parameterized complexity of independence and domination on geometric graphs. In: Parameterized and Exact Computation, 2nd International Workshop, IWPEC, Proceedings, pp. 154–165 (2006). https://doi.org/10.1007/11847250_14
10. de Berg, M., Kisfaludi-Bak, S., Woeginger, G.: The complexity of dominating set in geometric intersection graphs. Theor. Comput. Sci. (2018). https://doi.org/10.1016/j.tcs.2018.10.007
11. Flum, J., Grohe, M.: Parameterized Complexity Theory. Texts in Theoretical Computer Science. An EATCS Series. Springer, Berlin (2006). https://doi.org/10.1007/3-540-29953-X
12. Impagliazzo, R., Paturi, R.: On the complexity of k-SAT. J. Comput. Syst. Sci. **62**(2), 367–375 (2001). https://doi.org/10.1006/jcss.2000.1727
13. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer, Berlin (2015). https://doi.org/10.1007/978-3-319-21275-3
14. Marx, D.: On the optimality of planar and geometric approximation schemes. In: 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20–23, 2007, Providence, RI, USA, Proceedings, pp. 338–348. IEEE Computer Society, New York (2007). https://doi.org/10.1109/FOCS.2007.50
15. Marx, D., Sidiropoulos, A.: The limited blessing of low dimensionality: when $1-1/d$ is the best possible exponent for d-dimensional geometric problems. In: Proceedings of the 30th Annual Symposium on Computational Geometry, SOCG 2014, pp. 67–76. ACM, New York (2014). https://doi.org/10.1145/2582112.2582124
16. Biró, C., Bonnet, É., Marx, D., Miltzow, T., Rzążewski, P.: Fine-grained complexity of coloring unit disks and balls. In: Proceedings of SoCG 2017, pp. 18:1–18:16 (2017). https://doi.org/10.4230/LIPIcs.SoCG.2017.18
17. Marx, D., Pilipczuk, M., Pilipczuk, M.: On subexponential parameterized algorithms for Steiner tree and directed subset TSP on planar graphs. In: 59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7–9, 2018, pp. 474–484. IEEE Computer Society, New York (2018). https://doi.org/10.1109/FOCS.2018.00052
18. Kisfaludi-Bak, S., Nederlof, J., van Leeuwen, E.J.: Nearly ETH-tight algorithms for planar Steiner tree with terminals on few faces (2019). arXiv:1811.06871
19. Burton, B., Cabello, S., Kratsch, S., Pettersson, W.: The parameterized complexity of finding a 2-sphere in a simplicial complex. CoRR arXiv:1802.07175 (2018)
20. Bonnet, É., Bousquet, N., Charbit, P., Thomassé, S., Watrigant, R.: Parameterized complexity of independent set in H-free graphs. e-prints arXiv:1810.04620 (2018)
21. Amis, A.D., Prakash, R., Huynh, D., Vuong, T.: Max–min d-cluster formation in wireless ad hoc networks. In: Proceedings IEEE INFOCOM 2000, the Conference on Computer Communications, 19th Annual Joint Conference of the IEEE Computer and Communications Societies, Reaching the Promised Land of Communications, Tel Aviv, Israel, March 26–30, 2000, pp. 32–41. IEEE, New York (2000). https://doi.org/10.1109/INFCOM.2000.832171