Metric Dimension for Gabriel Unit Disk Graphs is NP-Complete

Stefan Hoffmann*, Egon Wanke

Institute of Computer Science, Heinrich-Heine-Universität Düsseldorf, D-40225 Düsseldorf, Germany

Abstract

We show that finding a minimal number of landmark nodes for a unique virtual addressing by hop-distances in wireless ad-hoc sensor networks is NP-complete even if the networks are unit disk graphs that contain only Gabriel edges. This problem is equivalent to Metric Dimension for Gabriel unit disk graphs. The Gabriel edges of a unit disc graph induce a planar $O(\sqrt{n})$ distance and an optimal energy spanner. This is one of the most interesting restrictions of Metric Dimension in the context of wireless multi-hop networks.

Keywords: metric dimension, unit disk graph, gabriel graph, virtual address, wireless multi-hop network, sensor network

1. Introduction

Wireless radio networks in which all nodes have the same radio range are often modeled as unit disc graphs. An undirected graph is a unit disk graph (UDG) if its vertices can be embedded in the Euclidean plane $\mathbb{R}^2$ by an embedding $\rho$ such that two vertices $u, v$ are connected by an edge if and only if their Euclidean distance $d^e(u,v) := \|\rho(u) - \rho(v)\|_2$ is at most 1. The vertices represent the sensor nodes and the undirected edges the symmetric communication channels between them. Two sensor nodes can communicate with each other, if they are close enough.

A widely used idea to reduce the complexity of distributed algorithms for wireless ad-hoc sensor networks is to consider only some of the available connections. The strongest restriction that preserves connectivity is a spanning tree. A spanning tree unfortunately does not allow efficient routing through the network. A slightly weaker restriction is the Gabriel graph $G$. An edge $\{u, v\}$ of an embedded graph $G$ is a Gabriel edge, if there is no vertex $w$ such that $d^e(u,w)^2 + d^e(w,v)^2 \leq d^e(u,v)^2$. The Gabriel edges induce a planar

*Corresponding author, phone number +492118110684

Email addresses: stefan.hoffmann@hhu.de (Stefan Hoffmann), E.Wanke@hhu.de (Egon Wanke)
$O(\sqrt{n})$ distance and optimal energy spanner $G'$ of $G$, see [3]. That is, the ratio between the length of a shortest path in $G'$ and $G$ is of $O(\sqrt{n})$ if the costs of the edges $\{u, v\}$ are $d^*(u, v)$, and 1 if the costs of the edges are $d^*(u, v)\alpha$ for some $\alpha \geq 2$.

Many routing algorithms for wireless ad-hoc sensor networks are based on virtual coordinates, see for example VCap [3], JUMPS [2], GLIDER [11], VCost [10], BVR [12], and HBR [15]. The virtual coordinates are computed from the distances to specific nodes called landmarks, anchors, or beacons [12, 13]. The setup of the virtual addressing starts by selecting some landmark nodes that flood large parts of the network. After that every node defines its virtual coordinates depending on the distances to the landmark nodes.

The virtual coordinates can be used to route a message through the network. Most routing algorithms require the virtual addresses to be unique. Packet delivery is also only guaranteed if different nodes have different virtual coordinates. The length of the virtual addresses increases with the number of landmark nodes. Since every landmark node has to flood large parts of the network during initialization, the number of landmark nodes should be as small as possible to reduce the amount of consumed energy. These conditions cause the question of how to determine a minimum set of landmark nodes that provides a unique virtual addressing.

From a graph theoretical point of view, the decision version of the problem above is called Metric Dimension. A set of vertices $S \subseteq V$ of a graph $G = (V, E)$ is called a resolving set, if for every vertex pair $u, v \in V$ there is at least one vertex $s \in S$ such that the distance between $s$ and $u$ differs from the distance between $s$ and $v$. A graph has metric dimension at most $k$ if there is a resolving set of size at most $k$. Metric Dimension is known to be NP-complete for general graphs as well as for planar graphs [9, 14, 18]. For a fixed number of landmarks $k$ it is decidable in polynomial time. It is also decidable in polynomial time for special classes of graphs like trees, wheels, complete graphs, $k$-regular bipartite graphs and outerplanar graphs [7, 9, 18, 19, 20]. The approximability of Metric Dimension by centralized algorithms has been studied for bounded degree, dense, and general graphs in [16]. There are also boundaries for the metric dimension of some special classes of graphs, see [6, 8].

In this paper, we prove that Metric Dimension is NP-complete even for unit disk graphs that contain only Gabriel edges. This is the most interesting restriction of the problem in the context of computing unique virtual addresses in wireless ad-hoc sensor networks. We show how to construct in polynomial time for an arbitrary instance $(X, C)$ of a special satisfiability problem a Gabriel unit disk graph $H_\psi$ and an integer $k$ such that there is a satisfying truth assignment for $X$ if and only if there is a resolving set $S$ for $H_\psi$ of size at most $k$.

2. Definitions and terminology

Let $G = (V, E)$ be a directed or undirected graph. That is, $V$ is the set of vertices and $E$ is the set of directed edges $E \subseteq \{(u, v) \mid u, v \in V, u \neq v\}$ or undirected edges $E \subseteq \{(u, v) \mid u, v \in V, u \neq v\}$, respectively. A graph
A graph $G = (V, E)$ is a subgraph of $G = (V', E')$ if $V' \subseteq V$ and $E' \subseteq E$. It is an induced subgraph of $G$ if $E' = E \cap \{(u, v) \mid u, v \in V'\}$ or $E' = E \cap \{(u, v) \mid u, v \in V'\}$, respectively.

A sequence of $k$ vertices $(v_1, \ldots, v_k)$ of $G$ is called a directed path if $G$ is directed and $(v_i, v_{i+1}) \in E$ for $1 \leq i < k$. It is called an undirected path if $G$ is undirected and $(v_i, v_{i+1}) \in E$ for $1 \leq i < k$.

### 2.1. Gabriel unit disk graphs

For a vector $(x, y) \in \mathbb{R}^2$, let $\|(x, y)\|_2 := \sqrt{x^2 + y^2}$ be its Euclidean norm.

**Definition 1.** An undirected graph $G = (V, E)$ is a unit disk graph (UDG) if there is a UD embedding $\rho : V \to \mathbb{R}^2$ such that $\forall u, v \in V$, $(u \neq v) \Rightarrow (\rho(u) \neq \rho(v))$ and $\{u, v\} \in E \Rightarrow \|\rho(u) - \rho(v)\|_2 \leq 1$.

**Definition 2.** For an undirected graph $G = (V, E)$ and an embedding $\rho : V \to \mathbb{R}^2$ of its vertices an edge $\{u, v\}$ is a Gabriel edge if there is no vertex $w \in V$ with

$$\|\rho(u) - \rho(w)\|_2 + \|\rho(w) - \rho(v)\|_2 \leq \|\rho(u) - \rho(v)\|_2.$$ 

An undirected graph $G = (V, E)$ is a Gabriel UDG (GUDG) if there is a UD embedding $\rho : V \to \mathbb{R}^2$ for which all edges in $E$ are Gabriel edges. Such an embedding is called a GUDG embedding.

**Definition 3.** A graph $G = (V, E)$ is planar if there is an embedding $\rho : V \to \mathbb{R}^2$ with the following two properties:

1. $\forall u, v \in V$, $(u \neq v) \Rightarrow (\rho(u) \neq \rho(v))$ and

2. for every edge between two vertices $u$ and $v$ there is a line in $\mathbb{R}^2$ connecting the points $\rho(u)$ and $\rho(v)$ such that no two lines for two different edges intersect except at a common endpoint.

### 2.2. Metric Dimension

The distance $d(u, v)$ between two nodes $u, v \in V$ in a graph $G = (V, E)$ is the smallest integer $k$ for which there is a path between $u$ and $v$ with $k$ edges.

**Definition 4.** Let $G = (V, E)$ be an undirected graph. A set of vertices $S \subseteq V$ is a resolving set for $G$, see for example $[1, 2, 3]$, if for every pair $u, v \in V$ of two distinct vertices there is a vertex $s \in S$ such that $d(u, s) \neq d(v, s)$. The minimum size of a resolving set for an undirected graph $G$ is called the metric dimension of $G$.

In this paper, we consider the complexity of the following decision problem.

| GUDG Metric Dimension |
|------------------------|
| Given: A GUDG $G = (V, E)$ and a positive integer $k$. |
| Question: Is there a resolving set $S \subseteq V$ for $G$ of size at most $k$? |
For a given undirected graph \( G \) and a positive integer \( k \), deciding whether there is a resolving set for \( G \) of size at most \( k \) is NP-complete for general graphs \( [14, 18] \) and even for planar graphs \( [9] \). We extend this and show that it remains NP-complete for GUDGs with vertex degree \( \leq 6 \).

### 2.3. Satisfiability

Let \( X = \{x_1, \ldots, x_n\} \) be a set of boolean variables. A truth assignment for \( X \) is a function \( t : X \to \{\text{true}, \text{false}\} \). If \( t(x_i) = \text{true} \) then we say variable \( x_i \) is true under \( t \); if \( t(x_i) = \text{false} \) then we say variable \( x_i \) is false under \( t \). If \( x_i \) is a variable of \( X \) then \( x_i \) and \( \overline{x_i} \) are literals over \( X \); \( x_i \) is called a positive literal, \( \overline{x_i} \) is called a negative literal. The positive literal \( x_i \) is true under \( t \) if and only if variable \( x_i \) is true under \( t \); negative literal \( \overline{x_i} \) is true under \( t \) if and only if variable \( x_i \) is false under \( t \). A clause over \( X \) is a set of literals over \( X \). It represents the disjunction of literals which is satisfied by a truth assignment \( t \) if and only if at least one of its literals is true under \( t \). A collection \( C \) of clauses over \( X \) is satisfiable if and only if there is a truth assignment \( t \) that simultaneously satisfies all clauses of \( C \).

**Definition 5.** Let \( X \) be a set of boolean variables and \( C \) be a collection of clauses over \( X \). The directed clause variable graph \( G_\psi = (X, C) \) has a vertex \( x \) for every variable \( x \in X \) and a vertex \( c \) for every clause \( c \in C \). There is a directed edge \((x, c)\) from variable vertex \( x \) to clause vertex \( c \) if and only if \( c \) contains literal \( x \) or \( \overline{x} \).

The following problem is NP-complete as shown in Corollary 3 of \( [9] \).

**1-Negative Planar 3-Sat**

**Given:** A set \( X \) of boolean variables and a collection \( C \) of clauses over \( X \) such that
- every variable \( x \in X \) occurs in exactly two or three clauses, once as a negative literal, and once or twice as a positive literal,
- the clause variable graph \( G_\psi \) for \( \psi = (X, C) \) is planar,
- every clause contains two or three literals and
g- every clause with three literals contains at least one negative literal.

**Question:** Is there a satisfying truth assignment for \( C \)?

**Definition 6.** Let \( G = (V, E) \) be a directed planar graph and \( \rho : V \to \mathbb{Z}^2 \) be an embedding of the vertices of \( G \). An edge path of length \( k - 1 \) for an edge \((u, v) \in E\) is a sequence of \( k \) points \((u, v) := (p_1, \ldots, p_k)\), \( p_i \in \mathbb{Z}^2 \), \( 1 \leq i \leq k \), for which

1. \((i \neq j) \Rightarrow (p_i \neq p_j)\),
2. \( p_1 = \rho(u), p_k = \rho(v), \) and

3. for \( 1 \leq i < k, \) if \( p_i = (x_i, y_i) \) then
\[
p_{i+1} \in \{(x_i + 1, y_i), (x_i - 1, y_i), (x_i, y_i + 1), (x_i, y_i - 1)\}.
\]

A planar orthogonal grid drawing of a directed graph \( G = (V, E) \) is a pair \( (\rho, E) \) where

- \( \rho : V \rightarrow \mathbb{Z}^2 \) is an embedding of the vertices of \( G \),
- \( E \) is a collection of edge paths, one for every edge of \( G \), and
- no two distinct edge paths \((u, v)\) and \((u', v')\) of \( E \) have a common point \( p \) unless the corresponding two edges \((u, v)\) and \((u', v')\) have a common vertex \( w \in \{u, v\} \cap \{u', v'\} \) and \( p = \rho(w) \).

3. Main result

**Theorem 1.** GUDG Metric Dimension is NP-complete.

The membership to NP is obvious, because it is easy to verify in polynomial time whether a given set of vertices \( S \) is a resolving set. The NP-hardness is shown by a reduction from 1-Negative Planar 3-Sat. Every directed planar graph with vertex degree at most four has a planar orthogonal grid drawing, whose size (determined by the number of grid points used for the edge paths) is polynomially bounded in the number of vertices. Such a drawing can be computed in polynomial time using the algorithms described in [21, 22]. Let \( G_\psi \) be a planar clause variable graph for \( \psi \). Since the vertices of \( G_\psi \) have a degree of at most three, we can assume that we have a planar orthogonal grid drawing \((\rho_0, E_0)\) for the clause variable graph \( G_\psi \).

The following preprocessing phase modifies \( \psi = (X, C) \) into an equivalent instance \( \psi' = (X', C') \) for which there is a planar orthogonal grid drawing \((\rho', E')\) for the clause variable graph \( G_{\psi'} \) whose edge paths have a length of at most two. This upper bound on the length of the edge paths is necessary, because the gadget that is going to represent the edge paths cannot be embedded across arbitrarily large areas.

Let \((x, c)\) be an edge of the clause variable graph \( G_\psi \) whose edge path in \( E \) is \((x, c) = (p_1, \ldots, p_k)\) with length \( l = k - 1 \geq 3 \). Assume clause \( c \) contains a positive literal \( x \) (the case for a negative literal runs analogously). Then a new variable \( h \) and a new clause \( c' = \{x, h\} \) are inserted and the positive literal \( x \) in clause \( c \) is replaced by the positive literal \( h \). The new sets of variables and clauses again define an instance for 1-Negative Planar 3-Sat that has a satisfying truth assignment if and only if the original instance has a satisfying truth assignment. The original planar orthogonal grid drawing for \( G_\psi \) is modified for the new clause variable graph as follows. Define \( \rho(c') := p_2, \rho(h) := p_3, (x, c') := (p_1, p_2), (h, c') := (p_2, p_3), \) and \((h, c) := (p_1, \ldots, p_k)\). The old edge path of length \( l \) is now replaced by two edge paths of length 1 and one edge path of length \( l - 2 \). This splitting step is repeated until all edge paths have length at most two. See Figure 1 for an example.
3.1. The construction of $H_\psi$

The GUDG $H_\psi$ is assembled using copies of the following 6 graphs called gadgets. Some of these gadgets are similar to the gadgets used in [9] for the proof that Planar Metric Dimension is NP-complete. The modifications are done to obtain GUDGs. Every gadget has two or three special path pairs called $(t,f)$-path pairs. The $(t,f)$-path pairs are used to represent the edges of the clause variable graph $G_{\psi'}$. The end-vertices of degree one of a $(t,f)$-path pair are called $(t,f)$-vertex pairs.

1. There are three variable gadgets $G_{v}^3$, $G_{v,a}^2$, and $G_{v,b}^2$ for the variables of $X'$, see Figure 2. The first variable gadget $G_{v}^3$ is used for the case that the corresponding variable is contained in three clauses, once as a negative literal and twice as a positive literal. The other two are used for the case that the corresponding variable is contained in two clauses, once as a negative literal and once as a positive literal.

These three gadgets have two different types of $(t,f)$-path pairs marked by $\ominus$ for the negative literal and $\oplus$ for the positive literals. The $\ominus$ $(t,f)$-vertex pair is $(t_{1,14},f_{1,14})$, the $\oplus$ $(t,f)$-vertex pairs are $(t_{2,14},f_{2,14})$ and $(t_{3,14},f_{3,14})$.

2. There are two clause gadgets $G_{c}^3$, $G_{c}^2$ for the clauses of $C'$, see Figure 3 to the left and in the middle. The first clause gadget $G_{c}^3$ is used for clauses with three literals. The second clause gadget $G_{c}^2$ is an induced subgraph of $G_{c}^3$. It is used for clauses with two literals.

The $(t,f)$-vertex pairs are $(t_{1,15},f_{1,15})$, $(t_{2,15},f_{2,15})$ and $(t_{3,15},f_{3,15})$.

3. There is an edge gadget $G_e$ for the edges of $G_{\psi'}$, see Figure 3 to the right. It is used to connect variable gadgets with clause gadgets and consists of
two disjoint paths with 37 vertices each.

These two paths represent one \((t, f)\)-path pair with two \((t, f)\)-vertex pairs \((t_1, f_1)\) and \((t_{37}, f_{37})\).

![Figure 2: The variable gadgets](image)

Graph \(H_\psi\) contains for every variable \(x \in X'\) with two positive literals and one negative literal in the clauses of \(C'\) a copy of \(G^3_x\). It contains for every variable \(x \in X'\) with one positive and one negative literal in the clauses of \(C'\) a copy of \(G^2_{v,a}\) or \(G^2_{v,b}\). Then \(H_\psi\) contains for every clause of \(C'\) with three literals one copy of \(G^3_c\) and for every clause of \(C'\) with two literals one copy of \(G^2_c\). Finally it contains for every edge of \(G_{\psi'}\) one copy of \(G_e\).

The gadgets are connected to each other by identifying \((t, f)\)-vertex pairs as follows. Let \(G_x\) be a variable gadget for variable \(x\), \(G_c\) a clause gadget for clause \(c\), and \(G_e\) the edge gadget for the edge \((x, c) \in E_{G_{\psi'}}\). One \((t, f)\)-vertex pair of \(G_x\) is identified with one \((t, f)\)-vertex pair of \(G_c\) and the other \((t, f)\)-vertex pair of \(G_e\) is identified with one \((t, f)\)-vertex pair of \(G_x\). If clause \(c\) contains the positive literal \(x\), then a ⊕-\((t, f)\)-vertex pair is used, otherwise, a ⊖-\((t, f)\)-vertex pair is used.

Identifying two \((t, f)\)-vertex pairs \((t_1, f_1)\) and \((t_2, f_2)\) means that all vertices adjacent to \(t_2\) are connected to \(t_1\), all vertices adjacent to \(f_2\) are connected to \(f_1\), and then the two vertices \(t_2, f_2\) and their incident edges are removed.

### 3.2. A GUDG embedding for \(H_\psi\)

In this section, it is shown that \(H_\psi\) can be assembled such that there is a GUDG embedding \(\rho_{H_\psi}\) for \(H_\psi = (V_\psi, E_\psi)\). For each gadget \(G\) we present several different GUDG embeddings. In every of these embeddings the vertices of the gadget are mapped to positions inside a polygon. The \((t, f)\)-vertices are
placed at the border of the polygon. The non-\((t, f)\)-vertices, i.e. all vertices not belonging to a \((t, f)\)-vertex pair, are placed at positions with a distance > 1 to all positions outside the polygon.

**Definition 7.** A tile is a pair \((G, \rho_G)\), where \(G = (V, E)\) is one of the gadgets defined above and \(\rho_G : V \rightarrow \mathbb{R}^2\) is a GUDG embedding for \(G\), called tile embedding, such that for every \((t, f)\)-vertex pair \((t, f)\),

\[
\{\rho_G(t), \rho_G(f)\} \in \left\{\left\{(−6, −1), (−6, 1)\right\}, \left\{(6, −1), (6, 1)\right\}, \left\{(-1, -6), (1, -6)\right\}, \left\{(-1, 6), (1, 6)\right\}\right\}.
\]

The \((t, f)\)-vertices are placed at the border of the square \([-6, 6] \times [-6, 6]\) with side length 12 and center \((0, 0)\).

All other vertices of the gadget are mapped to positions of the square \([-6, 6] \times [-6, 6]\) that have a distance > 1 to all positions of \(\mathbb{R}^2 \setminus [-6, 6] \times [-6, 6]\) outside the square \([-6, 6] \times [-6, 6]\).

We consider several different tile embeddings for gadget \(G^3_c\). These are the three tile embeddings shown in Figure 4 and all tile embeddings obtained by 90, 180, or 270 degree rotations and/or a horizontal or vertical mirroring. The rotations by 90, 180, and 270 degree and the horizontal and vertical mirroring can be performed by the mappings

\[
\begin{align*}
f_{90} &: (x, y) \mapsto (-y, x), \\
f_{180} &: (x, y) \mapsto (-x, -y), \\
f_{270} &: (x, y) \mapsto (y, -x),
\end{align*}
\]

\[
\begin{align*}
f_{\text{horizontal}} &: (x, y) \mapsto (-x, y), \\
f_{\text{vertical}} &: (x, y) \mapsto (x, -y),
\end{align*}
\]
We also consider several different tile embeddings for every gadget $G^2_{v,a}$, $G^3_{v,a}$, $G^3_{v,b}$, and $G_e$, see Figures 4, 5, and 6.

The tile embeddings resulting from the last embedding of Figure 7 for edge gadget $G_e$ are only used for edges whose edge paths have length 1. For edge paths of length 2, the following definition is used.

Let $S_{x,y} := \{(x', x + y') \in [-6, 6] \times [-6, 6]\}$ be the square of side length 12 and center (0, 0) translated by $(x, y) \in \mathbb{Z}^2$. The vertices of $G_e$ are mapped to positions of one of the following polygons $P_i$, $1 \leq i \leq 6$, formed by the union of 3 squares.
\[ P_1 := S_{-12,0} \cup S_{0,0} \cup S_{12,0} \]
\[ P_3 := S_{-12,0} \cup S_{0,0} \cup S_{0,-12} \]
\[ P_5 := S_{0,-12} \cup S_{0,0} \cup S_{12,0} \]

\[ P_2 := S_{0,12} \cup S_{0,0} \cup S_{0,-12} \]
\[ P_4 := S_{-12,0} \cup S_{0,0} \cup S_{0,12} \]
\[ P_6 := S_{0,12} \cup S_{0,0} \cup S_{12,0} \]

Figure 6: Three tile embeddings for clause gadget \( G^3_e \). Rotating and mirroring yields further tile embeddings.

Figure 7: Five tile embeddings for clause gadget \( G^2_e \) and one tile embedding for the edge gadget \( G_e \). Rotating and mirroring yields further tile embeddings.

**Definition 8.** A tile triple is a pair \((G_e, \rho)\), where \( G_e = (V, E) \) is the edge gadget defined above and \( \rho : V \rightarrow P_i \) for some \( i \in \{1, \ldots, 6\} \) is a GUDG embedding for \( G_e \), called a tile triple embedding, such that \( \{\rho(t_1), \rho(f_3)\}, \{\rho(t_{37}), \rho(f_{37})\} \) is either \( \{((-18, -1), (-18, 1)), ((18, -1), (18, 1))\}, \)
\( \{((-1, -18), (1, -18)), ((-1, 18), (1, 18))\}, \)
\( \{((-18, -1), (-18, 1)), ((-1, -18), (1, -18))\} \).
Every non-\((t,f)\)-vertex is mapped to a position of \(P_i\) with a distance > 1 to all positions \(\mathbb{R}^2 \setminus P_i\) outside of \(P_i\).

\[
\{(\{-18, -1\}, \{-18, 1\}), (\{-18, -1\}, \{18, -1\}), (\{-18, 1\}, \{18, 1\})\},
\]
or \[
\{(\{-1, 18\}, \{1, 18\}), (\{-1, -1\}, \{1, -1\})\},
\] respectively.

Figure 8: Three tile triple embeddings for the edge gadget \(G_e\), rotating and mirroring yields further tile triple embeddings for edge paths of length 2.

**Definition 9.** The orientation of a \((t,f)\)-vertex pair \((t,f)\) inside a tile (triple) \((G,\rho)\) is TF, if \(\rho(t)\) is encounter before \(\rho(f)\) on a clockwise traversal of the corresponding polygon and FT otherwise.

**Definition 10.** The connection vector of a tile \((G,\rho)\) is a 4-tuple \((a_1, a_2, a_3, a_4)\) consisting of components \(a_i = (b_i, c_i)\), where \(b_i\) is the orientation of the \((t,f)\)-vertex pair \((t_i, f_i)\) on the \(i\)-th side of the square \([-6, 6] \times [-6, 6]\) or \(\epsilon\), if none exists. The sides of the square are enumerated clockwise, starting from the top. If \(G\) is a variable gadget and \(b_i \neq \epsilon\), then \(c_i \in \{\oplus, \ominus\}\) denotes whether \((t_i, f_i)\) is an \(\oplus\)-(\(t,f\))-vertex pair or an \(\ominus\)-(\(t,f\))-vertex pair. For all other cases set \(c_i := \emptyset\).

Now define the equivalence relation \(\sim\) on the space of connection vectors as follows:
\((a_1, a_2, a_3, a_4) \sim_r (b_1, b_2, b_3, b_4)\) if and only if there exists a \(\delta \in \mathbb{N}\), such that for all \(1 \leq i \leq 4\), \(a_i = b_{(i+\delta) \mod 4}\). The type of a tile is the equivalence class of its connection vector, denoted by one representative in square brackets. Table II lists the types of all available tile embeddings.

Next we place the tiles and tile triples according to the planar orthogonal grid drawing \((\rho_{G'}, \mathcal{E})\) of the clause variable graph \(G'\).

Let \((v, c)\) be an edge of \(G'\) with edge path \(\{(x_1, y_1), \ldots, (x_n, y_n)\}\). Edge \((v, c)\) is from-left, from-right, from-top, or from-bottom if \(x_2 > x_1, x_2 < x_1, y_2 < y_1, \) or \(y_2 > y_1\), respectively. Analogously it is to-left, to-right, to-top, or to-bottom if \(x_n < x_{n-1}, x_n > x_{n-1}, y_n > y_{n-1}, \) or \(y_n < y_{n-1}\). Edge \((v, c)\) is a \(\ominus\)-edge or a \(\ominus\)-edge if variable \(v\) occurs in clause \(c\) as a positive or negative literal.

A tile or tile triple \((G, \rho_G)\) has a left-, right-, top-, or bottom-\((t, f)\)-vertex pair if gadget \(G\) has a \((t, f)\)-vertex pair mapped to \(\{(x_1, y_1), (x_2, y_2)\}\) such that \(x_1, x_2 < 0, x_1, x_2 > 0, y_1, y_2 > 0, \) or \(y_1, y_2 < 0\), respectively.

Let \(v\) be a variable vertex of \(G'\) and \(G\) be the copy of the gadget for \(v\). Then a tile \((G, \rho_G)\) for \(G\) is chosen whose \((t, f)\)-vertices are arranged according to the directions of the edges \(e = (v, c)\) from \(v\) in the following sense. If, for example, \(e\) is a \(\ominus\)-from-left edge then tile \((G, \rho_G)\) has a right-\(\ominus\)-(\(t, f)\)-vertex pair. If \(e\) is a \(\ominus\)-from-top edge then tile \((G, \rho_G)\) has a bottom-\(\ominus\)-(\(t, f)\)-vertex pair, etc.

Let \(c\) be a clause vertex of \(G'\) and \(G\) be the copy of the gadget for \(c\). Then a tile \((G, \rho_G)\) for \(G\) is chosen whose \((t, f)\)-vertices are arranged according to the directions of the edges \(e = (v, c)\) to \(c\) in the following sense. If, for example, \(e\) is a to-left edge then tile \((G, \rho_G)\) has a right-\((t, f)\)-vertex pair. If \(e\) is a to-top edge then tile \((G, \rho_G)\) has a bottom-\((t, f)\)-vertex pair, etc.

Let \(e\) be an edge of \(G'\) and \(G\) be the copy of the gadget for \(e\). Then a tile or tile triple \((G, \rho_G)\) for \(G\) is chosen whose \((t, f)\)-vertices are arranged according to the direction of edge \(e\) in the following sense. If, for example, \(e\) is a from-left-to-right edge then \((G, \rho_G)\) has a left- and a right-\((t, f)\)-vertex pair. If \(e\) is a from-top-to-bottom edge then \((G, \rho_G)\) has a top- and bottom-\((t, f)\)-vertex pair, etc.

To construct a GUDG embedding \(\rho_{H_\psi}\) for \(H_\psi\) we place the vertices of the gadgets as follows.

1. Let \(G\) be a gadget inserted into \(H_\psi\) for a variable or clause vertex \(u\) of the clause variable graph \(G'\), and let \((G, \rho_G)\) be the selected tile. Then the GUDG embedding \(\rho_{H_\psi}\) for the vertices \(w\) from gadget \(G\) is defined by \(\rho_{H_\psi}(w) := 24 \cdot \rho_{G'}(u) + \rho_G(w)\).

2. Let \(G\) be a gadget inserted into \(H_\psi\) for an edge path \((u, v) = ((x_1, y_1), (x_2, y_2)) \in \mathcal{E}\) of length 1, and let \((G, \rho_G)\) be the selected tile for gadget \(G\). Then the GUDG embedding \(\rho_{H_\psi}\) for the vertices \(w\) from gadget \(G\) is defined by \(\rho_{H_\psi}(w) := 12 \cdot ((x_1, y_1) + (x_2, y_2)) + \rho_G(w)\).

3. Let \(G\) be a gadget inserted into \(H_\psi\) for an edge path \((u, v) = ((x_1, y_1), (x_2, y_2), (x_3, y_3)) \in \mathcal{E}\) of length 2, and let \((G, \rho_G)\) be the
| type | gadget | tile embedding |
|------|--------|----------------|
| $([TF,\oplus],(\epsilon,\emptyset),(FT,\ominus),(FT,\oplus))$ | \(G_v^3\) | fig 4 (1) |
| $([TF,\oplus],(\epsilon,\emptyset),(TF,\ominus),(FT,\oplus))$ | \(G_v^3\) | fig 4 (3), mirrored |
| $([\epsilon,\emptyset),(TF,\ominus),(FT,\ominus),(FT,\oplus))$ | \(G_v^3\) | fig 4 (2) |
| $([TF,\ominus),(TF,\ominus),(FT,\ominus),(\epsilon,\emptyset))$ | \(G_v^3\) | fig 4 (2), mirrored |
| $([FT,\oplus),(TF,\ominus),(\epsilon,\emptyset))$ | \(G_v^3\) | fig 4 (3) |
| $([FT,\ominus],(TF,\ominus),(\epsilon,\emptyset))$ | \(G_v^3\) | fig 4 (1), mirrored |
| $([\epsilon,\emptyset],(\epsilon,\emptyset),(FT,\ominus),(FT,\oplus))$ | \(G_{v,a}\) | fig 5 (1) |
| $([FT,\ominus],(\epsilon,\emptyset),(FT,\ominus),(\epsilon,\emptyset))$ | \(G_{v,a}\) | fig 5 (6), mirrored |
| $([FT,\ominus],(\epsilon,\emptyset),(FT,\ominus),(\epsilon,\emptyset))$ | \(G_{v,a}\) | fig 5 (2) |
| $([FT,\ominus],(\epsilon,\emptyset),(TF,\ominus),(\epsilon,\emptyset))$ | \(G_{v,b}\) | fig 5 (5), mirrored |
| $([\epsilon,\emptyset](FT,\ominus),(FT,\ominus),(\epsilon,\emptyset))$ | \(G_{v,a}\) | fig 5 (3) |
| $([TF,\emptyset],(TF,\emptyset),(\epsilon,\emptyset))$ | \(G_c^3\) | fig 6 (1) |
| $([TF,\emptyset],(TF,\emptyset),(FT,\emptyset),(\epsilon,\emptyset))$ | \(G_c^3\) | fig 6 (1), mirrored |
| $([TF,\emptyset],(TF,\emptyset),(TF,\emptyset),(\epsilon,\emptyset))$ | \(G_c^3\) | fig 6 (2) |
| $([TF,\emptyset],(TF,\emptyset),(TF,\emptyset),(\epsilon,\emptyset))$ | \(G_c^3\) | fig 6 (3) |
| $([TF,\emptyset],(\epsilon,\emptyset),(TF,\emptyset),(\epsilon,\emptyset))$ | \(G_c^3\) | fig 6 (3), mirrored |
| $([TF,\emptyset],(\epsilon,\emptyset),(TF,\emptyset),(\epsilon,\emptyset))$ | \(G_c^2\) | fig 7 (1) |
| $([TF,\emptyset],(\epsilon,\emptyset),(FT,\emptyset),(\epsilon,\emptyset))$ | \(G_c^2\) | fig 7 (1), mirrored |
| $([TF,\emptyset],(\epsilon,\emptyset),(FT,\emptyset),(\epsilon,\emptyset))$ | \(G_c^2\) | fig 7 (2) |
| $([TF,\emptyset],(\epsilon,\emptyset),(FT,\emptyset),(\epsilon,\emptyset))$ | \(G_c^2\) | fig 7 (3) |
| $([TF,\emptyset],(\epsilon,\emptyset),(TF,\emptyset),(\epsilon,\emptyset))$ | \(G_c^2\) | fig 7 (3), mirrored |
| $([TF,\emptyset],(\epsilon,\emptyset),(TF,\emptyset),(\epsilon,\emptyset))$ | \(G_c^2\) | fig 7 (4) |
| $([TF,\emptyset],(\epsilon,\emptyset),(FT,\emptyset),(\epsilon,\emptyset))$ | \(G_c^2\) | fig 7 (4), mirrored |
| $([FT,\emptyset],(\epsilon,\emptyset),(TF,\emptyset),(\epsilon,\emptyset))$ | \(G_c^2\) | fig 7 (5) |

Table 1: The available tile types and the corresponding embeddings
selected tile triple for gadget $G$. Then the GUDG embedding $\rho_{H_\psi}$ for the vertices $w$ from gadget $G$ is defined by $\rho_{H_\psi}(w) := 24 \cdot (x_2, y_2) + \rho_G(w)$.

Figure 9 shows an example of the placed tiles for the clause variable graph $G_{\psi'}$ of Figure 1.

Graph $H_\psi$ is defined by the union of gadgets through identification of their $(t, f)$-vertices. Thus we have to show that it is always possible to select a tile (or tile triple) for every inserted copy of a gadget such that identified vertices are placed at the same position.

Every tile and tile triple embedding is a GUDG embedding. A GUDG embedding for the inner part used in the variable gadgets is shown in Figure 9. The distance between non-$(t, f)$-vertices from different tiles is always greater than 1. Thus, the resulting embedding $\rho_{H_\psi}$ is a GUDG embedding for $H_\psi$.

Figure 9: Left: A tile placement for the clause variable graph $G_{\psi'}$ of Figure 1. Clause and variable gadgets are labeled $C$ and $V$, respectively. Right: A GUDG embedding for the inner part of the variable gadgets.

If $t$ is identified with $t'$ and $f$ is identified with $f'$ then there only are the following two cases left to be considered due to the restrictions for the placement of $(t, f)$-vertex pairs inside tiles and tile triples. Either $(\rho_{H_\psi}(t), \rho_{H_\psi}(f)) = (\rho_{H_\psi}(t'), \rho_{H_\psi}(f'))$ or $(\rho_{H_\psi}(t), \rho_{H_\psi}(f)) = (\rho_{H_\psi}(f'), \rho_{H_\psi}(t'))$. In the first case there is no problem, the second case is called an orientation conflict. $\rho_{H_\psi}$ is a valid embedding if it has no orientation conflicts.

**Lemma 1.** Let $(G, \rho)$ be a variable tile. Then there also is a variable tile $(G', \rho')$ with:

1. $(\rho(t_{1,14}), \rho(f_{1,14})) = (\rho'(f'_{1,14}), \rho'(t'_{1,14}))$
Theorem 2. Let \((G, \rho)\) be a clause tile and \((t_i, f_i), (t_j, f_j)\) two distinct \((t, f)\)-vertex pairs in \(G\). Then there also are three other tiles \((G, \rho_1), (G, \rho_2)\) and \((G, \rho_3)\) with:

\[
\begin{align*}
\rho(t_i) &= \rho_1(t_i), \quad \rho(f_i) = \rho_1(f_i), \\
\rho(t_j) &= \rho_1(t_j), \quad \rho(f_j) = \rho_1(f_j) \\
\rho(t_i) &= \rho_2(t_i), \quad \rho(f_i) = \rho_2(f_i), \\
\rho(t_j) &= \rho_2(t_j), \quad \rho(f_j) = \rho_2(f_j) \\
\rho(t_i) &= \rho_3(t_i), \quad \rho(f_i) = \rho_3(f_i), \\
\rho(t_j) &= \rho_3(t_j), \quad \rho(f_j) = \rho_3(f_j)
\end{align*}
\]

Lemmas 1 and 2 can easily be verified using Table 1. Note that the tile in Lemma 1 may contain a different variable gadget. This is the case for a variable that occurs only once as a positive literal.

Theorem 2. \(H_0\) is a GUDG if the correct gadgets for all variables are chosen during the assembly. The assembly can be computed in polynomial time.
Proof. The only ambiguity during the assembly of $H_\psi$ is the selection of either $G_{v,a}^2$ or $G_{v,b}^2$ for a variable that occurs once as a positive literal. To decide which gadget has to be used, we start with any selection of tiles for the variable and clause vertices that fit the directions of the edges of $G'_\psi$ as described above. By selecting tiles and tile triples for the edge gadgets that do not cause an orientation conflict with the adjacent variable tile, we can assume w.l.o.g. that all orientation conflicts are directly between clause and variable tiles. Now resolve all existing orientation conflicts as follows: By Lemma 2 we can select a clause tile $(G_{c}^2, \rho)$ for every clause containing two literals that does not cause any orientation conflicts with the adjacent two variable tiles. For every clause containing three literals we can select a clause tile $(G_{c}^3, \rho)$ that causes at most one orientation conflict. Since every clause with three literals contains at least one negative literal $x$ we can restrict this orientation conflict to the variable tile $(G_{x}, \rho_x)$ for variable $x$. By Lemma 1, we can then select a variable tile $(G_{x}^\prime, \rho_x^\prime)$ that causes neither this orientation conflict nor one with another adjacent clause tile. $G_{x}^\prime$ is the variable gadget that has to be used for variable $x$ during the assembly of $H_\psi$. Since every orientation conflict is solved by at most three local replacements of tiles and tile triples that never cause new orientation conflicts to appear the tile selection for a valid GUDG embedding, and by extend the GUDG $H_\psi$, can be computed in polynomial time.

3.3. Correctness

In this section, we prove that there is a resolving set $S$ for $H_\psi$ of size at most $4 \cdot |X'|$ if and only if there is a satisfying truth assignment for $\psi$.

Definition 11. Let $G = (V,E)$ be an undirected graph. A vertex $u \in V$ resolves a vertex pair $v, w \in V$ if $d(u, v) \neq d(u, w)$. A vertex pair $v, w$ is called unsolved by set $U \subseteq V$, if no vertex of $U$ resolves $v, w$.

Lemma 3. Every resolving set $S$ for $H_\psi$ contains at least three vertices of each copy of a variable gadget $G$. $S$ contains vertex $a_i$ or $b_i$ for $i = 1, \ldots, 3$, see Figure 2.

Proof. Suppose neither $a_i$ nor $b_i$ is in $S$ for some $i$. Since all shortest paths from any vertex of $S$ to $a_i$ and $b_i$ contain $f_{i,0}$ and $d(f_{i,0}, a_i) = d(f_{i,0}, b_i) = 1$, there is no vertex $s \in S$ that resolves the pair $a_i, b_i$.

Let $\mathcal{F}$ be a set of vertices consisting of one vertex from $\{a_i, b_i\}$, $1 \leq i \leq 3$, for every copy of a variable gadget, i.e., $|\mathcal{F}| = 3 \cdot |X'|$. The vertices in $\mathcal{F}$ are called forced landmarks.

Lemma 4. Every vertex pair $T_1, T_2$ and $N_1, F$ from the same copy of a variable gadget and every vertex pair $w_1, w_2$ from the same copy of a clause gadget are unsolved by $\mathcal{F}$. For all other vertex pairs $u, v$, there is a forced landmark of $\mathcal{F}$ that resolves $u, v$.

Proof. Let $u, v$ be an arbitrary vertex pair of $H_\psi$. 

16
Case 1. $u$ and $v$ belong to the same gadget copy $G$.

Case 1.1. $G$ is a variable gadget copy. Vertex pair $u, v = T_1, T_2$ is unsolved by $F$, because every shortest path from a landmark of $F$ to $T_1$ and $T_2$ passes one of the $f_{i,0}$ vertices and $d(T_1, f_{i,0}) = d(T_2, f_{i,0})$ for $1 \leq i \leq 3$. Analogously, vertex pair $u, v = N_1, F$ is unsolved by $F$, since $d(N_1, f_{i,0}) = d(F, f_{i,0}) = 2$ for $1 \leq i \leq 3$. The vertex pairs $u, v = t_{i,j}, f_{i,j+1}, 1 \leq i \leq 3, 0 \leq j \leq 13$ are resolved by the forced landmarks from the variable gadgets connected to this $(t, f)$-path pair via one clause gadget and two edge gadgets. All other vertex pairs $u, v$ are resolved by one of the forced landmarks of $G$, see table $\text{2}$.

Case 1.2. $G$ is an edge gadget copy. All edge gadgets are basically extensions of the $(t, f)$-path pairs inside the clause and variable gadgets, so this case has already been covered in case $\text{1.1}$.

Case 1.3. $G$ is a clause gadget copy. The vertex pair $u, v = w_1, w_2$ is unsolved by $F$, because all shortest paths from the forced landmarks to $w_1$ and $w_2$ enter $G$ via an $f$ path and pass vertex $m$. All other vertex pairs $u, v$ are resolved by at least one forced landmark in an adjacent variable gadget copy, see table $\text{3}$.

Case 2. $u$ and $v$ belong to different gadget copies.

Case 2.1. $u$ and $v$ belong to variable gadget copies $G_1$ and $G_2$, respectively. All forced landmarks of $G_1$ resolve the pair $u, v$: The distance from a forced landmark of $G_1$ to any vertex of $G_1$ is at most 18 (e.g. $d(a_1, t_{2,14})$) while the distance to any vertex of $G_2$ is greater than 18, because any shortest path has to traverse at least two edge gadgets and one clause gadget.

Case 2.2. $u$ and $v$ both belong to edge gadget copies. Let $a$ and $b$ be the forced landmarks closest to $u$ and $v$, respectively. Also $d(a, u) < d(c, u)$ for all forced landmarks $c \neq a$, because all vertices at distance $d(a, u)$ from $u$ are either in a clause gadget, an edge gadget or the variable gadget that $a$ belongs to, hence none of them being another forced landmark. Now suppose that neither $a$ nor $b$ resolves the pair $u, v$, meaning $d(a, v) = d(a, u)$ and $d(b, u) = d(b, v)$. Together we get $d(a, v) = d(a, u) < d(b, u) = d(b, v)$, which contradicts to the assumption that $b$ is the landmark closest to $v$.

Case 2.3. $u$ and $v$ belong to the clause gadget copies $G_c$ and $G_d$ for the clauses $c$ and $d$, respectively.

Case 2.3.1. There is a variable $x$ such that $c$ contains literal $x$ or $\overline{x}$ and $d$ does contain neither $x$ nor $\overline{x}$. The forced landmarks of variable gadget $G_x$ for $x$ resolve the pair $u, v$: All shortest paths from forced landmarks of $G_x$ to vertices of the adjacent clause gadget have a length of at most 86 (e.g. $d(a_1, t_{1,15})$), while all shortest paths to vertices of a non-adjacent clause gadget have a length greater than 86, since they have to traverse three edge gadgets, one clause gadget and one variable gadget.
**Case 2.3.2.** All variables that occur as a literal in $c$ also occur as a literal in $d$ and vice versa. Since $c \neq d$ there is one variable $x$ such that $x \in c$ and $\overline{x} \in d$ or $\overline{x} \in c$ and $x \in d$. W.l.o.g. let $x \in c$, $\overline{x} \in d$. Furthermore let $a$ and $b$ be the forced landmarks in $G_x$ closest to $u$ and $v$, respectively. Then $d(u, a) < d(u, b)$ and $d(v, b) < d(v, a)$. Now assume $u, v$ is unsolved by $F$. Then $d(v, a) = d(u, a)$ and $d(v, b) = d(u, b)$. Together we get $d(v, a) = d(u, a) < d(u, b) = d(v, b) < d(v, a)$ which is a contradiction.

**Case 2.4.** $u$ belongs to variable gadget copy $G_x$ and $v$ to edge gadget copy $G_e$.

**Case 2.4.1.** $G_x$ is adjacent to $G_e$. See case [1.4]

**Case 2.4.2.** $G_x$ is not adjacent to $G_e$. Every forced landmarks $a$ in $G_x$ resolves $u, v$: The distance from $a$ to any vertex in $G_x$ is at most 18 while the distance to any vertex in $G_e$ is greater than 18, since the shortest path has to traverse at least one other edge gadget and one clause gadget.

**Case 2.5.** $u$ belongs to variable gadget copy $G_x$ and $v$ to clause gadget copy $G_c$. See case 2.4.2

**Case 2.6.** $u$ belongs to clause gadget copy $G_c$ and $v$ to edge gadget copy $G_e$.

**Case 2.6.1.** $G_c$ is adjacent to $G_e$. See case 1.3

**Case 2.6.2.** $G_c$ is not adjacent to $G_e$. If $v$ belongs to a $(t, f)$-vertex pair of $G_e$ then this case has already been covered in one of the previous cases, because $v$ has been identified with a vertex from a clause or variable gadget. So we can assume that $v \in \{t_2, \ldots, t_{36}, f_2, \ldots, f_{36}\}$. Let $a$ be the forced landmark closest to $v$ inside a variable gadget adjacent to $G_e$. Then $a$ resolves $u, v$, because $d(a, v) \leq 51$ while the distance from $a$ to any vertex in $G_c$ is at least 52.

**Lemma 5.** The vertex pair $T_1, T_2$ can only be resolved by $T_1, T_2, N_1, N_2$ or $F$ of the same gadget copy $G_x$.

**Proof.** Anyone of these vertices obviously resolves $T_1, T_2$ and except for $N_2$ they also resolve $N_1, F$. Furthermore there are shortest paths from $T_1$ and $T_2$ to any other vertex $v$ in $G_x$ that cross either $t_{2,0}$ or $t_{3,0}$, meaning that $v$ does not resolve $T_1, T_2$, because $d(T_1, v) = \min\{d(t_{2,0}, v), d(t_{3,0}, v)\} + 1 = d(T_2, v)$.

**Corollary 1.** The Lemmata 3, 4 and 5 imply that every resolving set for $H_\psi$ contains at least $4 \cdot |X'|$ vertices.

**Theorem 3.** GUDG Metric Dimension is NP-complete.
Table 3. Hopdistances \(d_x, d_y, d_z\) between vertices of a copy of \(G^3\) and their closest forced landmark inside adjacent variable gadget copies for variables \(x, y, z\).

| \(v\) | \(d_x, d_y, d_z\) | \(v\) | \(d_x, d_y, d_z\) | \(v\) | \(d_x, d_y, d_z\) | \(v\) | \(d_x, d_y, d_z\) |
|---|---|---|---|---|---|---|---|
| \(m\) | (66, 66, 66) | \(w_1\) | (67, 67, 67) | \(w_2\) | (67, 67, 67) | \(c_1\) | (67, 68, 68) |
| \(c_2\) | (68, 68, 68) | \(c_3\) | (68, 68, 68) | \(t_{1,1}\) | (66, 69, 69) | \(t_{1,2}\) | (65, 70, 70) |
| \(t_{1,3}\) | (64, 71, 71) | \(t_{1,4}\) | (63, 72, 72) | \(t_{1,5}\) | (62, 73, 63) | \(t_{1,6}\) | (61, 74, 74) |
| \(t_{1,7}\) | (60, 75, 75) | \(t_{1,8}\) | (59, 76, 76) | \(t_{1,9}\) | (58, 77, 77) | \(t_{1,10}\) | (57, 78, 78) |
| \(t_{1,11}\) | (56, 79, 79) | \(t_{1,12}\) | (55, 80, 80) | \(t_{1,13}\) | (54, 81, 81) | \(t_{1,14}\) | (53, 82, 82) |
| \(t_{1,15}\) | (52, 82, 82) | \(f_{1,1}\) | (65, 67, 67) | \(f_{1,2}\) | (64, 68, 68) | \(f_{1,3}\) | (63, 69, 69) |
| \(f_{1,4}\) | (62, 70, 70) | \(f_{1,5}\) | (61, 71, 71) | \(f_{1,6}\) | (60, 72, 72) | \(f_{1,7}\) | (59, 73, 73) |
| \(f_{1,8}\) | (58, 74, 74) | \(f_{1,9}\) | (57, 75, 75) | \(f_{1,10}\) | (56, 76, 76) | \(f_{1,11}\) | (55, 77, 77) |
| \(f_{1,12}\) | (54, 78, 78) | \(f_{1,13}\) | (53, 79, 79) | \(f_{1,14}\) | (52, 80, 80) | \(f_{1,15}\) | (51, 81, 81) |
| \(t_{2,1}\) | (69, 66, 69) | \(t_{2,2}\) | (70, 65, 70) | \(t_{2,3}\) | (71, 64, 71) | \(t_{2,4}\) | (72, 63, 72) |
| \(t_{2,5}\) | (73, 62, 73) | \(t_{2,6}\) | (74, 61, 74) | \(t_{2,7}\) | (75, 60, 75) | \(t_{2,8}\) | (76, 59, 76) |
| \(t_{2,9}\) | (77, 58, 77) | \(t_{2,10}\) | (78, 57, 78) | \(t_{2,11}\) | (79, 56, 79) | \(t_{2,12}\) | (80, 55, 80) |
| \(t_{2,13}\) | (81, 54, 81) | \(t_{2,14}\) | (82, 53, 82) | \(t_{2,15}\) | (82, 52, 82) | \(f_{2,1}\) | (67, 65, 67) |
| \(f_{2,2}\) | (68, 64, 68) | \(f_{2,3}\) | (69, 63, 69) | \(f_{2,4}\) | (70, 62, 70) | \(f_{2,5}\) | (71, 61, 71) |
| \(f_{2,6}\) | (72, 60, 72) | \(f_{2,7}\) | (73, 59, 73) | \(f_{2,8}\) | (74, 58, 74) | \(f_{2,9}\) | (75, 57, 75) |
| \(f_{2,10}\) | (76, 56, 76) | \(f_{2,11}\) | (77, 55, 77) | \(f_{2,12}\) | (78, 54, 78) | \(f_{2,13}\) | (79, 53, 79) |
| \(f_{2,14}\) | (80, 52, 80) | \(f_{2,15}\) | (81, 51, 81) | \(f_{3,1}\) | (69, 69, 69) | \(f_{3,2}\) | (70, 70, 70) |
| \(t_{3,3}\) | (71, 71, 64) | \(t_{3,4}\) | (72, 72, 63) | \(t_{3,5}\) | (73, 73, 62) | \(t_{3,6}\) | (74, 74, 61) |
| \(t_{3,7}\) | (75, 75, 60) | \(t_{3,8}\) | (76, 76, 59) | \(t_{3,9}\) | (77, 77, 58) | \(t_{3,10}\) | (78, 78, 57) |
| \(t_{3,11}\) | (79, 79, 56) | \(t_{3,12}\) | (80, 80, 55) | \(t_{3,13}\) | (81, 81, 54) | \(t_{3,14}\) | (82, 82, 53) |
| \(t_{3,15}\) | (82, 82, 52) | \(f_{3,1}\) | (67, 67, 65) | \(f_{3,2}\) | (68, 68, 64) | \(f_{3,3}\) | (69, 69, 63) |
| \(f_{3,4}\) | (70, 70, 62) | \(f_{3,5}\) | (71, 71, 61) | \(f_{3,6}\) | (72, 72, 60) | \(f_{3,7}\) | (73, 73, 59) |
| \(f_{3,8}\) | (74, 74, 58) | \(f_{3,9}\) | (75, 75, 57) | \(f_{3,10}\) | (76, 76, 56) | \(f_{3,11}\) | (77, 77, 55) |
| \(f_{3,12}\) | (78, 78, 54) | \(f_{3,13}\) | (79, 79, 53) | \(f_{3,14}\) | (80, 80, 52) | \(f_{3,15}\) | (81, 81, 51) |

**Proof.** Let \(\psi = (X, C)\) be an instance for 1-Negative Planar 3-Sat and \((H_\psi, A \cdot |X'|, H_\psi) = (V_H, E_H)\) the instance for GDG Metric Dimension as described above. Then \(\psi\) is satisfiable \(\Leftrightarrow\exists\) resolving set \(S \subseteq V_H : |S| \leq 4 \cdot |X'|\).

\[ \Rightarrow: \] Because \(\psi\) is satisfiable, the altered instance \(\psi'\) with variable set \(X' = \{x_1, \ldots, x_n\}\) and clause set \(C'\) is also satisfiable. Let \(A : X \rightarrow \{true, false\}\) be a satisfying truth assignment for \(\psi'\). For every variable \(x \in X'\) choose \(T_1\) of the corresponding variable gadget copy as a landmark if \(A(x) = true\) and \(F\) otherwise. Together with \(F\) this set of vertices is a resolving set: According to Lemmata 4 and 5 all pairs are resolved except for \(w_1, w_2\) of each clause gadget copy. Since \(A\) satisfies \(\psi'\) each clause gadget copy is connected to either the \(\oplus-(t, f)\)-path pair of a variable gadget copy in which \(F\) has been chosen as a landmark or to one of the \(\oplus-(t, f)\)-path pairs of a variable gadget copy in which \(T_1\) has been chosen as a landmark. In both cases, for some \(i \in \{0, 1\}\), the shortest path from this landmark to \(w_{1+i}\) enters the clause gadget through the \(t\) path while the shortest path to \(w_{2-i}\) enters the gadget through the \(f\) path,

19
thus resolving $w_1, w_2$.

$\Leftarrow$: Let $S$ be a resolving set for $H_\psi$ with $4 \cdot |X'|$ vertices. According to Lemmata 3 and 5 $S$ contains $3 \cdot |X'|$ forced landmarks and one of the vertices $T_1, T_2, N_1, N_2$ or $F$ for every variable in $X'$. Now construct a truth assignment $A$ for $\psi'$ by setting $A(x)$ to false if the fourth landmark in $x$'s variable gadget copy is $F$ and true otherwise. Then $A$ satisfies $\psi'$: $S$ resolves $w_1, w_2$ in every clause gadget copy, so there is one landmark $s \in S$ for each copy such that $d(s, w_1) \neq d(s, w_2)$. This implies that the shortest path $p$ from $s$ to either $w_1$ or $w_2$ does not contain $m$ and enters the clause gadget through the adjacent $t$ path. So $s$ has to be inside the variable gadget that is connected to the clause gadget through one edge gadget:

Assume that it is not. Then $p$ has to traverse an entire clause gadget $G_c$ and another variable gadget $G_v$ first. Independent of where $p$ enters $G_c$, on a shortest path to a vertex outside of $G_v$ it is never shorter to leave $G_c$ through a $t$ path than crossing $G_c$'s $m$ node and leaving through the $f$ path. So $p$ enters $G_v$ through an $f$ path, maybe contains the $N_2$ node of $G_v$ and leaves through another $f$ path, hence not resolving $w_1, w_2$.

Since $p$ enters the clause gadget through the $t$ path and $s$ also resolves $T_1, T_2$ in its variable gadget, $A(x)$ has been set to false if the clause gadget is connected to the $\ominus(t, f)$-path pair of the variable gadget and true otherwise. So every clause contains at least one positive literal whose variable is set to true or one negative literal whose variable is set to false, meaning that $\psi'$ is satisfied. And because $\psi'$ is satisfiable, the original instance $\psi$ is satisfiable.
References

[1] INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies, 13-17 March 2005, Miami, FL, USA. IEEE, 2005.

[2] F Benbadis, J.-J. Puig, M. Dias de Amorim, C. Chaudet, T. Friedman, and D. Simplot-Ryl. Jumps: Enhancing hop-count positioning in sensor networks using multiple coordinates. CoRR, abs/cs/0604105, 2006.

[3] P. Bose, L. Devroye, W.S. Evans, and D.G. Kirkpatrick. On the spanning ratio of gabriel graphs and $\beta$-skeletons. SIAM Journal on Discrete Mathematics, 20(2):412–427, 2006.

[4] D.L. Boutin. Determining sets, resolving sets, and the exchange property. Graphs and Combinatorics, 25(6):789–806, 2009.

[5] A. Caruso, S. Chessa, S. De, and A. Urpi. Gps free coordinate assignment and routing in wireless sensor networks. In INFOCOM [1], pages 150–160.

[6] G.G. Chappell, J.G. Gimbel, and C. Hartman. Bounds on the metric and partition dimensions of a graph. Ars Comb., 88, 2008.

[7] G. Chartrand, L. Eroh, M.A. Johnson, and O. Oellermann. Resolvability in graphs and the metric dimension of a graph. Discrete Applied Mathematics, 105(1-3):99–113, 2000.

[8] G. Chartrand, C. Poisson, and P. Zhang. Resolvability and the upper dimension of graphs. Computers and Mathematics with Applications, 39(12):19–28, 2000.

[9] J. Diaz, O.M. Pottonen, M. Serna, and E.J. van Leeuwen. On the complexity of metric dimension. In Algorithms ESA 2012, volume 7501 of Lecture Notes in Computer Science, pages 419–430. Springer Berlin Heidelberg, 2012.

[10] E. H. Elhafsi, N. Mitton, and D. Simplot-Ryl. Cost over progress based energy efficient routing over virtual coordinates in wireless sensor networks. In WOWMOM, pages 1–6. IEEE, 2007.

[11] Q. Fang, J. Gao, L.J. Guibas, V. de Silva, and L. Zhang. GLIDER: gradient landmark-based distributed routing for sensor networks. In INFOCOM [1], pages 339–350.

[12] R. Fonseca, S. Ratnasamy, J. Zhao, C.T. Ee, D.E. Culler, S. Shenker, and I. Stoica. Beacon vector routing: Scalable point-to-point routing in wireless sensornets. In NSDI. USENIX, 2005.

[13] R.K. Gabriel and R.R. Sokal. A new statistical approach to geographic variation analysis. Systematic Zoology, 18(3):259–278, 1969.
[14] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.

[15] D. Gaußmann, S. Hoffmann, and E. Wanke. Hierarchical bipartition routing for delivery guarantee in sparse wireless ad hoc sensor networks with obstacles. In *ICWN*, pages 3–9. CSREA Press, 2012.

[16] M. Hauptmann, R. Schmied, and C. Viehmann. On approximation complexity of metric dimension problem. In *Proceedings of the 21st international conference on Combinatorial algorithms*, IWOCA’10, pages 136–139, Berlin, Heidelberg, 2011. Springer-Verlag.

[17] M.C. Hernando, M. Mora, I.M. Pelayo, C. Seara, J. Cáceres, and M.L. Puertas. On the metric dimension of some families of graphs. *Electronic Notes in Discrete Mathematics*, 22:129–133, 2005.

[18] S. Khuller, B. Raghavachari, and A. Rosenfeld. Landmarks in graphs. *Discrete Applied Mathematics*, 70(3):217–229, 1996.

[19] S.W. Saputro, E.T. Baskoro, A.N.M. Salman, D. Suprijanto, and A.M. Baca. The metric dimension of regular bipartite graphs. *arXiv/1101.3624*, 2011.

[20] B. Shanmukha, B. Sooryanarayana, and K.S. Harinath. Metric dimension of wheels. *Far East Journal of Applied Mathematics*, 8(3):217–229, 2002.

[21] R. Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM Journal on Computing*, 16(3):421–444, 1987.

[22] R. Tamassia and I.G. Tollis. Planar grid embedding in linear time. *IEEE Transactions on Circuits and Systems*, 36(9):1230–1234, 1989.