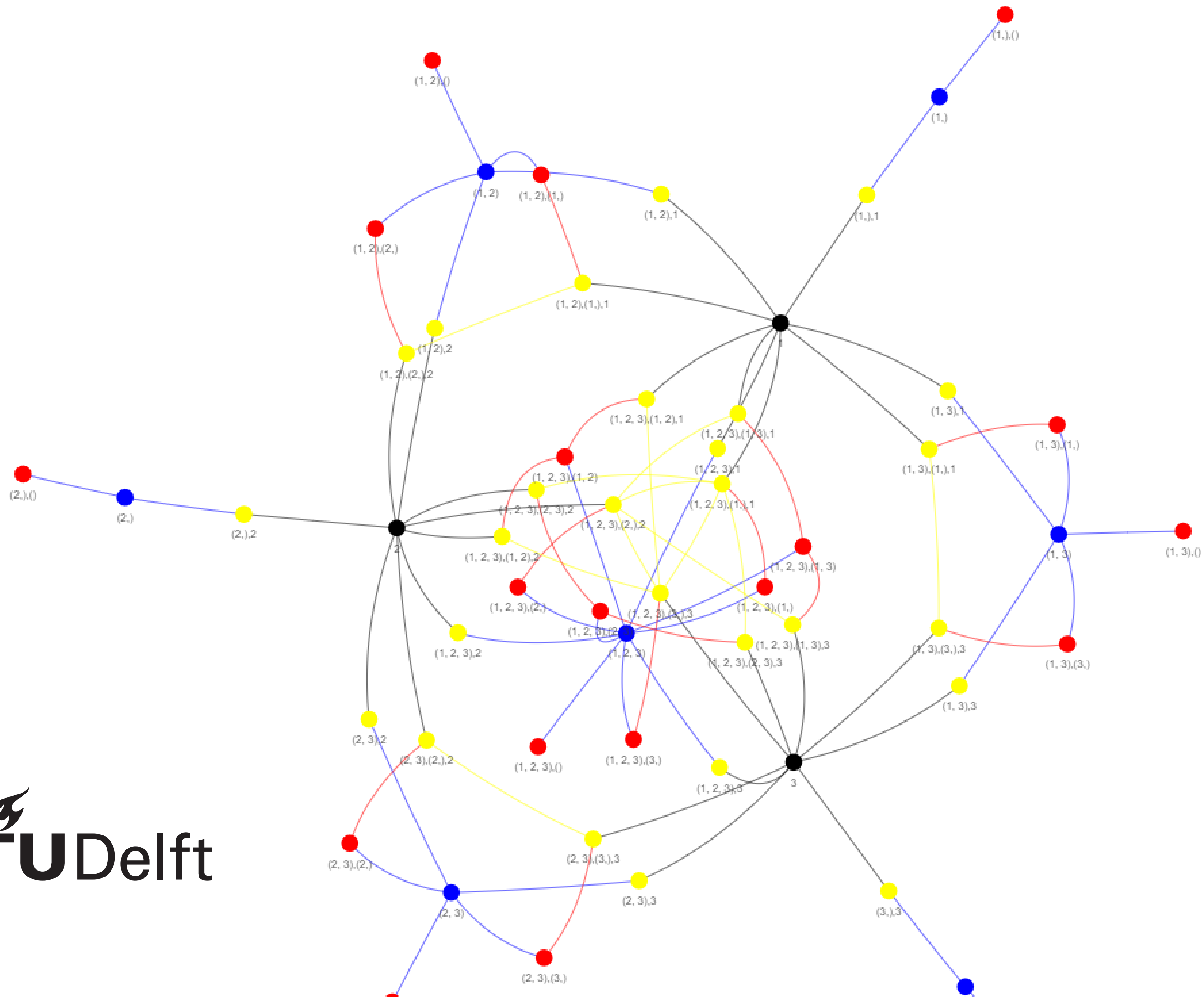
Extremal graphs for threshold metric dimension

T. Datema

Finding the extremal size of a graph with threshold-k metric dimension m



TU Delft

Extremal graphs for threshold metric dimension

by

T. Datema

to obtain the degree of Bachelor of Mathematics
at the Delft University of Technology,
to be defended publicly on Thursday July 14, 2022 at 13:00 AM.

Student number:	4499263	
Project duration:	April 19, 2022 – July 7, 2022	
Thesis committee:	Dr. J. Komjáthy,	TU Delft, supervisor
	Dr. A. Bishnoi,	TU Delft

This thesis is confidential and cannot be made public until March 31, 2023.

TU Delft

Preface

This thesis was written for the bachelor project of the study "Applied Mathematics" at the TU Delft. In this thesis we consider the problem of finding extremal graphs with a threshold-k metric dimension m, that is: for a set of sensors of size m that can measure up to a range of k, we will construct a graph that contains these sensors and as many vertices as possible, such that every vertex is measured uniquely by this set of sensors. This is an open problem, for as far as we know.

We will prove optimality of the constructions for up to $k = 3$ and arbitrary m and we shall also introduce a construction that works for general k and m with an incomplete proof of optimality.

I very much wish to thank my supervisor Júlia Komyáthy for steering this project in such a way that I was able to achieve these results and I would also like to thank Anurag Bishnoi for taking his time to be in the thesis committee.

T. Datema
Delft, July 2022

Abstract

In this thesis, we consider the threshold metric dimension problem of graphs, related to and motivated by source detection.

We construct a graph $G = (V, E)$ for a given set of sensors of size m: $\{s_1, s_2, ..., s_m\}$ and a range $k > 0$. We want that each node $v \in V$ has a unique combination of distances $(d_k(s_1, v), d_k(s_2, v), ..., d_k(s_m, v))$, where d_k is the distance function in a graph limited by the range k (the distance is denoted as being ∞ if the distance is larger than k). Our aim in this thesis is to construct such a graph that is extremal in size, that is: the vertex set V is as large as possible. We shall give such constructions with proof for optimality up to $k = 3$ and general m and a different construction with incomplete proof for optimality for general k and m. For any construction we will prove that each vertex is uniquely identified.

Furthermore, we will compare our results to another paper [6] with a similar conclusion about the extremal size of graphs with metric dimension m and a given diameter D.

Abstract for Layman

In this thesis we construct a network based on a given set of sensors that have a certain range. Dependent on the amount of sensors we are given and the range of these sensors, we want to construct our network, such that any possible source can be uniquely detected by the given set of sensors. That is: if each sensor gives out a signal that tells us its distance to a source in the network, we want that the total combination of all signals the sensors give, is unique for every possible source. This unique combination allows us to precisely determine the location of the source in our network. The aim of this thesis is to construct such a network that is as large as possible, in which we have succeeded.

Contents

1

Introduction

1.1. Motivation and description of the problem

The source detection problem considers a network consisting of nodes connected by links, a graph in mathematical terms, in which there is a source of some kind. This source can be many different things, for example an infected person spreading a disease through the network, in which case the nodes in the network represent a group of people and the links connecting them represent people who come into contact with one another. Another interpretation could be a fire in a building as suggested in a paper by Slater [10], in which case the nodes in the network represent the rooms of the building and the links in between the nodes represent rooms that are connected to one another, see Figure 1.1 and Figure 1.2.

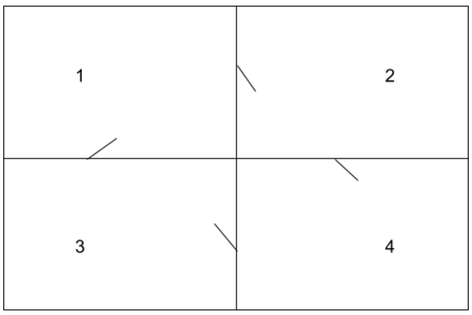


Figure 1.1: A possible floor plan of a building with four rooms, connected to each other by doors.

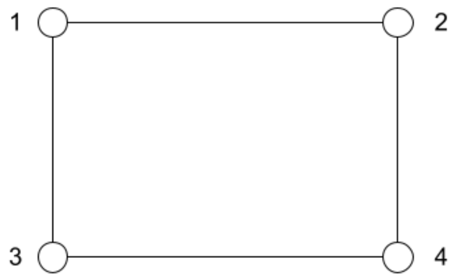


Figure 1.2: A graph representing the floor plan of Figure 1.1.

Now consider the example of a building with a fire as a source and suppose we want to be able to detect such a fire in a building, that is: we want to be able to detect a source in such a network. To do this, we can place smoke detectors in certain rooms. These give out a signal, one signal (say 0) if there is a fire in the room the detector is placed in, a different signal (say 1) if there is a fire in a room adjacent to the room the detector is placed in, or no signal (say ∞) if there is no fire in the same room as the detector or in an adjacent room, see Figure 1.3.

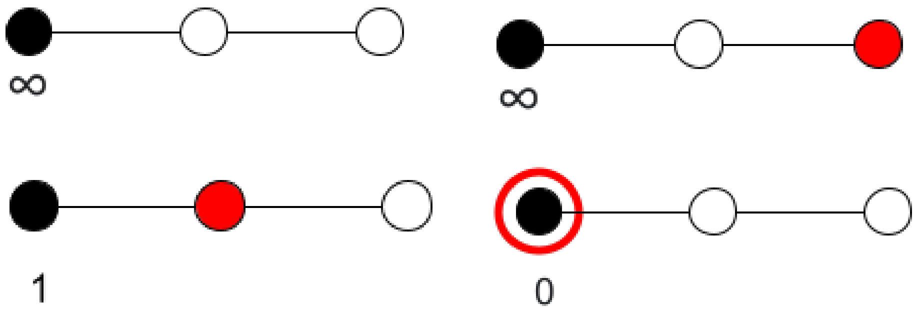


Figure 1.3: Four examples of different signals that a sensor (black) can send out when detecting (or not detecting as in the top left case) a source (red). The red circle represents a source that is on the sensor itself.

We want to place these detectors in such a way that we get an unique combination of signals from all the detectors, for each possible source (room with a fire), that way we can detect each possible fire: it does not matter in which room the fire is, we can always deduce in which room the fire is according to the combination of signals. An example of a graph that suffices this condition is given in figure

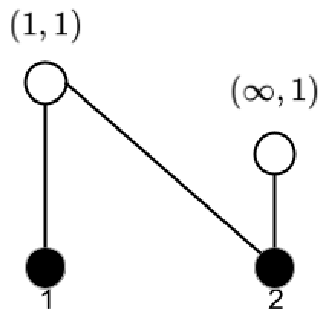


Figure 1.4: Example of a graph where all the non-sensor nodes (white) can be uniquely identified by the two sensors (black), each non-sensor node has a double assigned to it with the signals the sensors would give if there was a source there. The node with $(\infty, 1)$ for example, would not be detected by sensor 1 if it were a source, but would be detected by sensor 2, with signal 1 as it has distance 1 to sensor 2.

So in a more general case: for a certain network we want to assign certain nodes as detectors, such that any possible source in the network has a unique signal according to the detectors, so that it can be found uniquely.

The example above considers $k = 1$, that is: the range of the detectors (which we call k) has a distance of 1. That means that the detectors can give out a signal for sources that have a distance of at most 1 to the detector. We can also consider detectors that have a range of a larger distance, say $k = 2$ for example. In this case the detector can give out 4 different signals: 0 if the source is in the same location as the detector, 1 if it is adjacent to the detector, 2 if it has distance 2 to the detector or ∞ if it has distance 3 or higher to the detector. By distance we mean the amount of links one has to cross from the source to the detector.

The value of k can also be infinity, that means that a detector can measure a source up to any distance. In this case, we call the amount of sensors needed to uniquely determine each source, the “metric dimension” of the given graph. Should we work with a limited range k (a threshold on the range), we call the amount of sensors needed the “threshold-k metric dimension”.

We shall now give some more context to this problem, both in specific variations and in the historical sense.

1.2. Related problems and historical overview

The problem for optimally placing fire detectors and in a broader sense, source detection in a graph, was first introduced by Slater [9] in 1975 and a year later independently by Harary and Melter [5], though the term "dimension" of a graph has been used before in 1965 by Erdös [4] in a different, though related, context. In [10], Slater called a set of detectors with range $k = 1$ that can identify any source a "Locating dominating set", often also referred to as a "Locating dominating code" by other authors [2]. In general these locating dominating codes do not necessarily need that $k = 1$, but they do need that every non-sensor vertex in the graph is a neighbor to at least one sensor.

A closely related concept to this is the "r-Locating dominating code" introduced by Karpovsky [7], in which case the sensors may have a larger range than 1, but they will only distinguish between whether a source is within their range or not; they do give information about the distance to the source, merely if they measure it or not.

Another related problem is the "Fault tolerant locating dominating set", introduced by Slater [11]. This is the same situation as in his problem of finding a locating dominating set in [9] where the sensors have range 1, except that for any source, at most one sensor may give out no signal (or ∞ in our notation) when it actually should measure the source (thus being faulty).

In this thesis we shall discuss the truncated metric dimension of a graph, or "threshold-k metric dimension" as introduced by Tillquist [13], in Section 2.1 we shall formally define what this means. This thesis is closely related to a paper about extremal tree graphs of threshold-k metric dimension, by Bartha [1]. The research in this paper is what inspired this thesis.

One special variation that is of specific interest to us, is a paper dedicated to finding a graph with metric dimension m, that is as large as possible while retaining a diameter not larger than D, posed by Hernando et al. [6]. This problem is interesting to us, because its results turn out to be very similar to ours. We will elaborate more on this in Section 4.2.1.

One of the reasons there has been so much research in this branch of graph theory and combinatorics, is because it turns out to be quite difficult to find the metric dimension of an arbitrary graph, as we will explain in the next subsection.

1.3. Algorithms for finding the metric dimension of a graph

For a general graph, a brute force search algorithm can be used to find a resolving set of sensors, this is very computationally intensive and it turns out that this is a NP-complete problem [3].

However, for certain subsets of graphs, such as trees, wheels, d-dimensional grids etc... the metric dimension is known [12]. In [9], Slater also determined the metric dimension of trees: it is equal to the number of leaves in the tree, minus the amount of vertices in the tree that have degree at least 3 and that have at least a single path of degree two vertices that end in a leaf. For wheel graphs (graphs that form a cycle of nodes of a certain length with one central node that is connected to all the nodes in the cycle) the metric dimension is equal to $\lfloor \frac{2n+2}{5} \rfloor$, for $n > 6$, where n is the length of the cycle. And for d-dimensional grids (graphs representing a bounded, d-dimensional grid of nodes where nodes are connected if they differ by 1 at a single coordinate) the metric dimension is d.

There also exist algorithms for finding the metric dimension of certain graphs, such as an algorithm to find a threshold-1 resolving set for trees, as given by Slater [10]. Some reasons as to why it is interesting to develop these algorithms are given in the next subsection.

1.4. Applications

An important application of these graphs is source detection. The two cases discussed in the motivation, the spreading of a disease and placing smoke detectors, are cases of source detection. One particular example is detecting the source of a covid infection, as given by Ódor [8]. In this paper, Ódor discusses how it is possible to determine the source of the spread of a covid epidemic, even if there are asymptomatic patients. If we know the social network of the patients, we only need a special subset of all patients to have symptoms (the detecting set in the graph) to be able to guess where the source came from.

In short, this branch of mathematics can be applied to any discrete system in which there is a source of some kind and in which this source 'flows' through a network of sorts, to finally be detected by sensors.

We could also imagine, for example, a gossip-network of people in which someone starts a rumour. Assuming one knows who gossips with whom, the source of the rumour could be localised based on when certain people (the sensors) start spreading the rumour.

More applications other than source localization as described above are given in [12] by Tillquist et al., one example considers detecting network motifs. Motifs are subgraphs that appear in a graph with a higher than expected frequency. Finding these subgraphs is a very computationally expensive task. It is possible, however, to use algorithms to find subgraphs of a certain size. These algorithms then rely on resolving sets in these graphs. More on this can be read in [12] in Chapter 7.2.

2

Mathematical Formulation and content of thesis

2.1. Formulation of problem

Now a more formal description of the problem will be given. We consider a simple, undirected graph $G = (V, E)$ and we shall refer to $|G| = |V|$ as the size of a graph. On this graph, we first define a path $P_{u,v}$ from vertex $u \in V$ to vertex $v \in V$ as a collection of distinct vertices $P_{u,v} = (v_0, v_1, ..., v_l)$, where $v_0 = u$ and $v_l = v$ and $(v_i, v_{i+1}) \in E$ for all $0 \le i \le l-1$.

The length of a path is defined as the number of edges in that path (equal to the number of vertices in the path minus 1). We define the distance $d(u, v)$ as the length of the shortest possible path from u to v and $d(u, u) = 0$ for $u, v \in V$. We also define $d_k(u, v) := \min\{d(u, v), k+1\}$ as the distance bounded by the range k of the detectors. If $d_k(u, v) = k+1$, we may choose to write $d_k(u, v) = \infty$.

We define $S \subseteq V$ as the set of vertices that function as sensors in G and we say that a sensor vertex s measures another vertex v if $d_k(s, v) \le k$. We also define:

Definition 1 (Threshold-k resolving set)**.** *We call $S \subseteq V$ a "threshold-k resolving set" if for each $v \in V$ and the sensors $s_1, s_2, ..., s_n \in S$, we have that $d_k(v, s_i) \le k$ for at least one $i \in \{1, 2, ..., n\}$ (thus v is measured by at least one sensor) and that the collection of distances $d_k(v, s_i)$ for $i \in \{1, 2, ..., n\}$ is unique.*

This means that for a threshold-k resolving set of sensors, the specific combinations of signals that all the sensors send out are unique for every possible source in V. This ensures that we can uniquely deduce the location of some source in V depending on the signals sent out by the sensors.

Now we define:

Definition 2 (Metric dimension)**.** *The smallest amount of sensors needed to uniquely detect any source in V (that is: the smallest possible cardinality for S for it to be a threshold-k resolving set) is defined as the "threshold-k metric dimension" of G and written as $Tmd_k(G)$.*

2.2. Contribution of this thesis to the problem

The aim of this thesis is to find, for given values k and m, a graph $G(k, m)$ that has a threshold-k resolving set of vertices $\{s_1, s_2, ..., s_m\}$ that is extremal in size. This is an open problem, for as far as we know.

To this end, multiple constructions will be discussed that suffice this condition, one for general k and three for some specific values of k. For the general case, some proofs still need to be written out to show that it is actually extremal in size.

2.3. Layout of thesis

Finally, the thesis is organised as follows:

In Section 3 we consider a specific set of constructions that create an extremal graph for general m and for $1 \le k \le 3$. We will introduce what conditions a graph must suffice in order to be extremal and we shall give a construction and a proof for every value k for $1 \le k \le 3$. We will also intrudoce the problem of sensors being too close together and how we will deal with that in the case of $k = 3$ and how we will still end up with a extremal graph.

In Section 4 we shall consider a construction that will create a graph of threshold-k metric dimension m for general k and m. We are still left to finalize the proof to show that these constructions are extremal, though we are able to show that these graphs do indeed have threshold-k metric dimension m. We will also expand on the problem posed in Section 3 where the sensors may be too close together for certain values of k.

3

Optimal constructions

The aim of this thesis is to construct, for as many values for k as possible and an arbitrary integer value $m \geq 1$, a graph $G = (V, E)$ that is as large as possible that contains a threshold-k resolving set S with cardinality $|S| = m$. That is, we want to construct a graph with as many vertices as possible that contains m sensors and furthermore satisfies the property that each vertex $v \in V$ has a unique combination of distances to the m sensors.

To check whether each vertex has a unique combination of distances to the m sensors, which are m distinct vertices labeled $s_1, s_2, ..., s_m$, we assign a vector $\underline{d}^{(v)} \in \mathbb{Z}^m$ to each vertex v, called the distance vector $\underline{d}^{(v)}$ of v. In this vector $\underline{d}^{(v)}$, the i-th value $d_i^{(v)}$ is defined to be the distance $d_k(v, s_i)$ between the vertex v and the i-th sensor s_i, where $1 \leq i \leq m$, $i \in \mathbb{N}$. So:

$$\underline{d}^{(v)} = \left(d_1^{(v)}, d_2^{(v)}, ..., d_m^{(v)}\right) \tag{3.1}$$

$$= (d_k(v, s_1), d_k(v, s_2), ..., d_k(v, s_m)). \tag{3.2}$$

An example of what distance vectors look like in a graph is given in Figure 3.1.

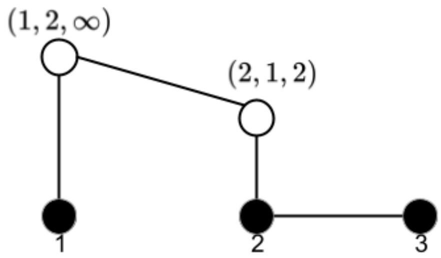


Figure 3.1: A graph with 3 sensors and 2 non-sensor vertices, where each non-sensor vertex has their distance vector next to it: a vector containing the distances (d_1, d_2, d_3) to sensors 1, 2 and 3.

Now given this notation, we want to construct a graph containing at least the m sensor vertices $s_1, s_2, ..., s_m$; and then containing as many vertices v_i as possible so that each $\underline{d}^{(v_i)}$ is a distinct vector. In the next claim we shall give an upper bound on the amount of vertices in a graph with m sensors with range k such that each vertex is measured uniquely.

Claim 1. *The number of possible vertices in a graph G with m sensors that form a threshold-k resolving set, is at most* $(k+1)^m + m - 1$.

Proof. Since only the sensors themselves can have distance 0 to any sensor (that is, themselves), the distances $d_k(v, s)$ between any other non-sensor vertex v and a sensor vertex s, takes values in $1, 2, ..., k+1$, by the definition of $d_k(\cdot, \cdot)$. Knowing this, we can establish a sharp bound for the amount of vertices we can fit in this graph, namely: $(k+1)^m + m - 1$, since we have m sensors and then $(k+1)^m$ possibilities for the non-sensor vertices, since they all must be assigned to a vector of length m containing $k+1$ different values for each element in the vector. Furthermore, we require that all vertices must be measured, so we do not allow the distance vector $(\infty, \infty, ..., \infty)$, hence the '-1'. □

If possible, for some integer value $k \geq 1$, we would like to construct a graph containing $(k+1)^m + m - 1$ different vertices. This is only possible whenever $k = 1$ or $k = 2$, as we will soon see. We will now show constructions for three different values of k.

3.1. Construction for an extremal graph with threshold-1 metric dimension

We start off with a construction for $k = 1$, so we will construct a graph $G = (V, E)$ with sensors $s_1, s_2, ..., s_m$ where all the non-sensor vertices $v \in V$ must be assigned to all possible vectors $\underline{d}^{(v)} = (d_1^{(v)}, d_2^{(v)}, ..., d_m^{(v)})$, where each $d_i^{(v)}$ takes values in $\{1, \infty\}$ (where ∞ means that the distance between a vertex and sensor is 2 or larger).

So given the m sensors $s_1, s_2, ..., s_m$, we want to construct a graph where each non-sensor vertex v has all possible combinations of 1 and ∞ over all m sensors. We can construct this graph as follows:

Definition 3 (Construction of an extremal graph $G_1(m)$ for $k = 1$)**.** *First we define $S = \{1, 2, ..., m\}$. Now let $G_1(m) = (V, E)$ be a graph where the vertex set V is given by:*

$$V := \{s_1, s_2, ..., s_m\} \cup \bigcup_{\substack{A \subseteq S \\ A \neq \emptyset}} \{v_A\}. \tag{3.3}$$

The edge set E is given by:

$$E := \bigcup_{\substack{A \subseteq S \\ A \neq \emptyset}} \bigcup_{i \in A} \{(v_A, s_i)\}. \tag{3.4}$$

In words, this construction is as follows: First create the m sensors as m distinct vertices $s_1, s_2, ..., s_m$, now for every nonempty subset $A \subseteq \{1, 2, ..., m\}$ of the m sensor vertices, we add a vertex v_A to the graph. For every v_A, we connect v_A to precisely all sensors that are in A. That is: we connect them to all sensors in $\{s_i : i \in A\}$ and we do not connect them to $\{s_i : i \notin A\}$. If $m = 3$ (so that $S = \{1, 2, 3\}$), for example, we connect $v_{\{1,3\}}$ to s_1 and s_3, but not to s_2.
A construction for $m = 3$ is presented in Figure 3.2.

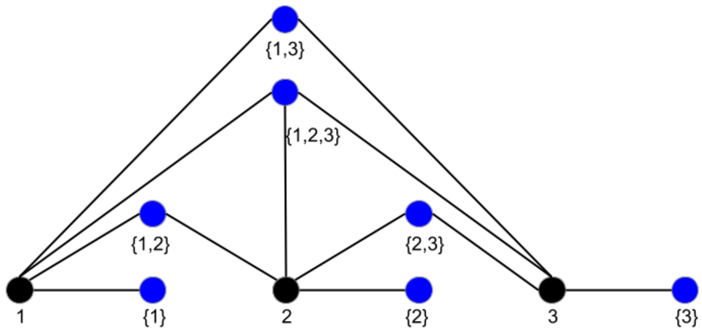


Figure 3.2: An extremal graph constructed as in Definition 3 with threshold-1 metric dimension 3. The sets below the blue vertices denote the set A for which the blue vertex v_A is created. Thus v_A connects to all sensors $s_i \in A$.

Theorem 1. *The construction as given for $G_1(m)$ in Definition 3 yields a graph with threshold-1 resolving set $\{s_1, s_2, ..., s_m\}$ for every $m \in \mathbb{N}$. Furthermore, $G_1(m)$ is extremal in size, that is: for any graph G with threshold-1 metric dimension m, we have $|G| \leq |G_1(m)|$.*

Proof. With this construction we obtain a graph, where no two sensors are connected to one another and no two non-sensor vertices v_{A_1} and v_{A_2}, where $A_1, A_2 \subseteq \{1, 2, ..., m\}, A_1 \neq A_2$, are connected to one another (see (3.4) in Definition 3). This means that all sensors have at least distance 2 to each other, and all non-sensor vertices v_A have at least distance 2 to all sensors that they are not directly connected to, and thus are not in $\{s_j : j \in A\}$.

Thus all sensors s_j have distance vectors $(\infty, \infty, ..., 0, ..., \infty)$ where the 0 is at position j, since they have distance 0 to themselves and a distance of 2 or larger (thus ∞) to the other sensors. Any non-sensor vertex v_A corresponding to $A \subseteq \{1, 2, ..., m\}$ has a distance vector which has value 1 for any position $j \in A$, since they are directly connected to $\{s_j : j \in A\}$, and value ∞ for any position $j \notin A$, since they have distance 2 or larger to the sensors in $\{s_j : j \notin A\}$.

Now each vertex has its own unique distance vector and we can construct any distance vector $\underline{d}^{(v)} \in \{1,\infty\}^m$, since for every distance vector $\underline{d}^{(v)}$ in $\{1,\infty\}$ we can create a set A that suffices $\{j \in A\}$ for all positions j that have value 1 in $\underline{d}^{(v)}$. So, after accounting for the m sensors and excluding the vertex not measured by any sensor, we have obtained the sharp bound of $(k+1)^m + m - 1 = 2^m + m - 1$ for $k = 1$, so that we indeed obtain extremality. □

3.2. Construction for an extremal graph with threshold-2 metric dimension

Now for $k = 2$ the construction requires some more steps, though it is similar in nature. This time we want to create vertices, such that they all uniquely correspond to all vectors $\underline{d}^{(v)} \in \{1,2,\infty\}^m$. The construction is given as follows:

Definition 4 (Construction of an extremal graph $G_2(m)$ for $k = 2$)**.** *First we define:* $S = \{1,2,...,m\}$. *Now let* $G_2(m) = (V,E)$ *be a graph where the vertex set* V *is given by:*

$$V := \{s_1, s_2, ..., s_m\} \cup \bigcup_{\substack{A \subseteq S \\ A \neq \emptyset}} \left[\{v_A\} \cup \bigcup_{B \subset A} \{v_{A,B}\} \right]. \tag{3.5}$$

The edge set E *is given by:*

$$E := \bigcup_{\substack{A \subseteq S \\ A \neq \emptyset}} \left[\left(\bigcup_{i \in A} \{(v_A, s_i)\} \right) \cup \bigcup_{B \subset A} \left(\{(v_A, v_{A,B})\} \cup \bigcup_{j \in B} \{(v_{A,B}, s_j)\} \right) \right]. \tag{3.6}$$

And B *is allowed to be the empty set in this definition.*

In words, this construction works as follows: First we create all the sensors $s_1, s_2, ..., s_m$ as our starting vertices. Then, as is the case with $k = 1$, we create all nonempty subsets $A \subseteq \{1,2,...,m\}$ and again for each subset A, we add a vertex v_A to the graph and connect it to each sensor $\{s_i : i \in A\}$ and not to $\{s_i : i \notin A\}$. After having created all possible subsets A, we create further subsets for every A; we create all possible, strictly smaller, subsets $B \subset A \subseteq \{1,2,...,m\}$. And now for each subset $B \subset A \subseteq \{1,2,...,m\}$, we create a vertex $v_{A,B}$ and connect it to v_A and then we also connect $v_{A,B}$ to each sensor $\{s_i : i \in B\}$ (and again, not to $\{s_i : i \notin B\}$).

If $m = 3$, for example, we would connect $v_{\{1,3\},\{1\}}$ to $v_{\{1,3\}}$ and to its sensors s_1 and s_3.

We give a full example for $m = 2$ in Figure 3.3

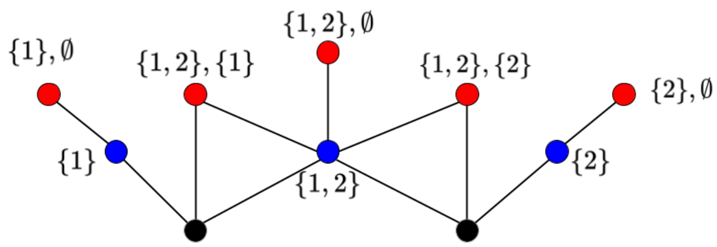


Figure 3.3: An extremal graph constructed as in Definition 4 with threshold-2 metric dimension 2. The sets next to the blue vertices denote the set A for which the blue vertex v_A is created. The sets next to the red vertices denote the sets A, B for which the red vertex $v_{A,B}$ is created.

We also give a python generated graph for $k = 2$ and $m = 3$ in Figure 3.4, with a code similar to that of Appendix A.

Theorem 2. *The construction as given for* $G_2(m)$ *in Definition 4 yields a graph with threshold-2 resolving set* $\{s_1, s_2, ..., s_m\}$ *for every* $m \in \mathbb{N}$ *Furthermore,* $G_2(m)$ *is extremal in size, that is: for any graph* G *with threshold-2 metric dimension* m*, we have* $|G| \leq |G_2(m)|$.

Before we begin with the proof, we first introduce the notation $N(v)$ of the set of vertices directly connected to a vertex v in a graph $G = (V,E)$:

$$N(v) = \{u \in V : (u,v) \in E\}. \tag{3.7}$$

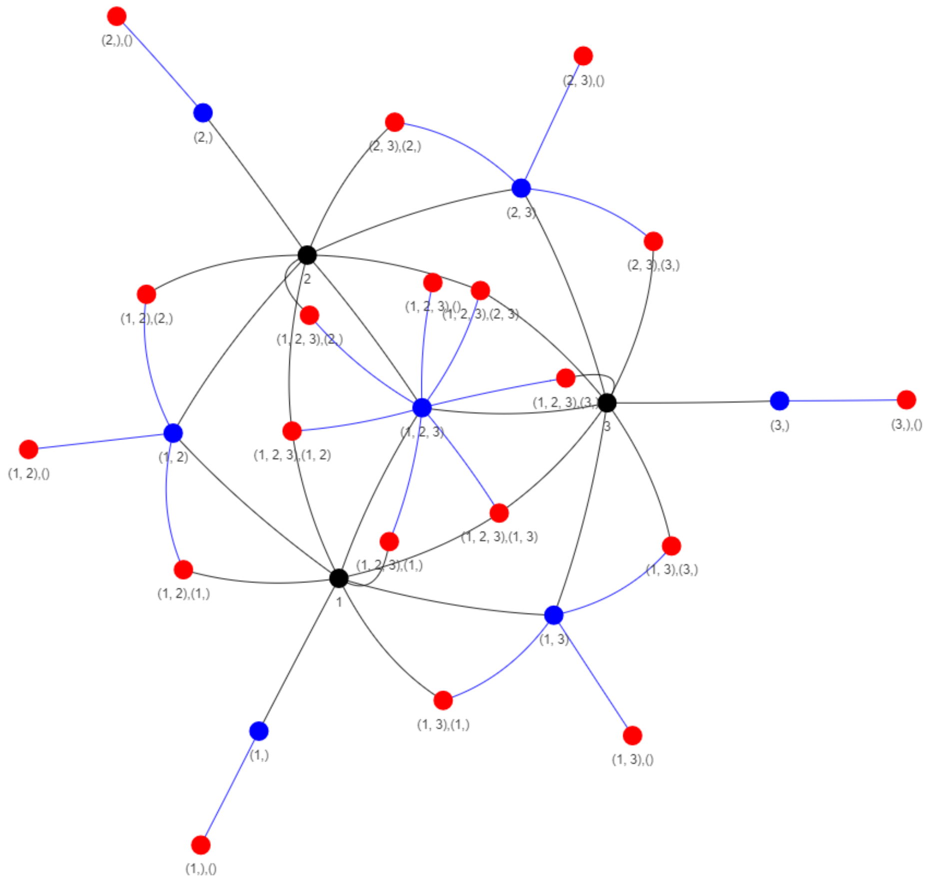


Figure 3.4: An extremal graph constructed as in Definition 4 with threshold-2 metric dimension 3. The sets next to the blue vertices denote the set A for which the blue vertex v_A is created. The sets next to the red vertices denote the sets A, B for which the red vertex $v_{A,B}$ is created.

Proof. From (3.6) in Definition 4 it follows that the vertices v_A for $A \subseteq \{1,2,\ldots,m\}, A \neq \emptyset$ have the following neighbors:

$$N(v_A) = \{s_i : i \in A\} \cup \{v_{A,B} : B \subset A\}. \tag{3.8}$$

For $v_{A,B}$ where $B \subset A$ we see that:

$$N(v_{A,B}) = \{s_i : i \in B\} \cup \{v_A\}. \tag{3.9}$$

And for s_i where $i \in \{1,2,\ldots,m\}$ we see that:

$$N(s_i) = \{v_A : i \in A\} \cup \{v_{A,B} : i \in B\}. \tag{3.10}$$

The vertices v_A.
From (3.8) it follows that every vertex v_A created for a nonempty subset $A \subseteq \{1,2,\ldots,m\}$ has distance 1 to all sensors $\{s_i : i \in A\}$, since they are directly connected to those sensors. These are the shortest paths possible from v_A to $\{s_i : i \in A\}$, so they have distance 1 to the sensors $\{s_i : i \in A\}$.

Furthermore, we see that v_A is only connected to sensors s_i where $i \in A$ and $v_{A,B}$ where $B \subset A$. Since the sensors are not connected to any other sensors and $v_{A,B}$ is not connected to any sensor s_i where $i \notin A$ (since $B \subset A$ and they are thus only connected to sensors that v_A is connected to), we see that v_A has no paths of length 1 or 2 to sensors $\{s_i : i \notin A\}$. Thus they have only paths of length 3 or more to the sensors $\{s_i : i \notin A\}$ and are thus not measured by those sensors and denoted as having distance ∞ to them.
From this we can conclude that the vertices v_A have distance vectors with value 1 on positions $j \in A$ and value ∞ on positions $j \notin A$. For example, if $m = 5$: $v_{\{1,2,4\}}$ has distance vector $(1,1,\infty,1,\infty)$.

The vertices $v_{A,B}$.

Now for the vertices $v_{A,B}$ for $B \subset A \subseteq \{1,2,...,m\}$. In (3.9) we see that these vertices $v_{A,B}$ are only directly connected to v_A and the sensors $\{s_i : i \in B\}$. So we directly see that they have distance 1 to the sensors $\{s_i : i \in B\}$. They furthermore have distance 2 to the sensors that are in $\{s_i : i \in A \setminus B\}$, since these vertices are not directly connected to $v_{A,B}$, but they are connected to v_A, to which $v_{A,B}$ is connected, thus a path $P_{v_{A,B},s_i} = (v_{A,B}, v_A, s_i)$ of length 2 exists from $v_{A,B}$ to any s_i where $i \in A \setminus B$.

And finally, they have distance 3 or larger (thus written as ∞) to all sensors in $\{s_i : i \notin A\}$, since they are only connected to v_A or directly to a subset of the sensors that v_A is connected to, and the sensors $v_{A,B}$ are directly connected to, are not directly connected to any other sensors.

Thus we summarize that $v_{A,B}$ has distance 1 to the sensors in $\{s_i : i \in B\}$, distance 2 to the sensors in $\{s_i : i \in A \setminus B\}$ and distance ∞ to the sensors in $\{s_i : i \notin A\}$. So they have distance vectors with 1 on positions $j \in B$, 2 on positions $j \in A \setminus B$ and ∞ on positions $j \notin A$. For example, if $m = 5$: $v_{\{1,2,4\},\{1,4\}}$ has corresponding distance vector $(1,2,\infty,1,\infty)$. And because B is a proper subset of A, we will always have at least a single value of 2 in a distance vector of $v_{A,B}$.

Conclusion.
Now we see that every vertex v corresponds to a unique distance vector $\underline{d}^{(v)} \in \{1,2,\infty\}^m$ and furthermore, each distance vector $\underline{d}^{(v)} \in \{1,2,\infty\}^m$ can be constructed: Any vertex whose distance vector contains only 1's and ∞'s can be constructed by all possible subsets $A \subseteq \{1,2,...,m\}$ and any vertex whose vector contains at least one 2, can be constructed with all possible subsets $B \subset A \subseteq \{1,2,...,m\}$ of all possible subsets A.

Thus we conclude that this construction yields all possible distance vectors $\underline{d}^{(v)} \in \{1,2,\infty\}^m$ and thus we attain the $(k+1)^m + m = 3^m + m - 1$ bound when we include the m sensors and exclude the vertex not measured by any sensor with this construction, so that we obtain extremality. □

3.3. Construction for an extremal graph with threshold-3 metric dimension

Now it turns out that it is impossible to attain the $(k+1)^m + m$ bound for $k = 3$ when $m \geq 2$, this is due to the following:

Suppose we construct a graph that contains the vertices that correspond to all distance vectors $\underline{d}^{(v)} \in \{1,2,3,\infty\}^m$ and suppose that $m = 2$ (the same argument holds for $m > 2$), then we must have vertices, say $v_{(1,1)}$ and $v_{(1,\infty)}$, corresponding to distance vectors $(1,1)$ and $(1,\infty)$ present in the graph. This however means that $v_{(1,1)}$ must be directly connected to both sensors s_1 and s_2 (since that is the only way to have distance 1 to both sensors) and since $v_{(1,\infty)}$ must be directly connected to sensor s_1 (since that is the only way to have distance 1 to sensor s_1), there exists a path $v_{(1,\infty)}, s_1, v_{(1,1)}, s_2$ from $v_{(1,\infty)}$ to s_2 of length 3. And this means that $v_{(1,\infty)}$ will be measured by s_2, since $k = 3$ and thus $v_{(1,\infty)}$ will actually have distance vector $(1,3)$ instead of $(1,\infty)$ as we wanted. This is a contradiction and thus it turns out we can not construct vertices such that all distance vectors $\underline{d}^{(v)} \in \{1,2,3,\infty\}^m$ are accounted for simultaneously.

Hence, not all possible vectors can be realised in one graph G and so we must some subset of distance vectors to construct a graph with metric dimension 3. The choice of the vertices we exclude is inspired by the example above: we shall only allow vertices that will not cause sensors to measure 'through' other sensors. That is: we will not allow paths between sensors of length $k - 1 = 2$, since if these paths exist, we obtain problems as above. We shall later prove that this restriction will lead to an extremal construction in the case of $k = 3$.

To formalize this restriction, we introduce the $\text{Min}_2\text{sum}(v)$ value of a vertex v, which works as follows:

Definition 5 ($\text{Min}_2\text{sum}(v)$)**.** *Consider $G = (V,E)$, $v \in V$ and let $k > 0$. We take the distance vector of v as $\underline{d}^{(v)} = \left(d_1^{(v)}, d_2^{(v)}, ..., d_m^{(v)}\right)$. We introduce the reordered distance vector $\underline{r}^{(v)} = \left(r_1^{(v)}, r_2^{(v)}, ..., r_m^{(v)}\right)$, where $\left(r_1^{(v)}, r_2^{(v)}, ..., r_m^{(v)}\right)$ is a permutation of $\left(d_1^{(v)}, d_2^{(v)}, ..., d_m^{(v)}\right)$ such that $r_1^{(v)} \leq r_2^{(v)} \leq ... \leq r_m^{(v)}$, we then finally define:*

$$Min_2sum(v) = r_1^{(v)} + r_2^{(v)}, \tag{3.11}$$

on vertices $v \in V$ and:

$$Min_2sum(\underline{d}^{(v)}) = r_1^{(v)} + r_2^{(v)}, \tag{3.12}$$

on vectors in $\{k+1\}^m$.

So $\mathrm{Min_2sum}(v)$ is the sum of the lowest two elements of the distance vector of v. And with this notation, we shall only allow vertices that have:

$$\mathrm{Min_2sum}(v) \geq k. \tag{3.13}$$

So in this case of $k = 3$, we will allow all vertices v with distance vectors $\underline{d}^{(v)} \in \{1,2,3,\infty\}^m$ that have no more than one element equal to 1. So for example with $m = 5$: the distance vector $(1,2,3,\infty,2)$ is allowed, but $(1,2,1,\infty,2)$ is not allowed.

In what follows we will show that a graph containing only the vertices v satisfying $\mathrm{Min_2sum}(v) \geq k$ is the best possible construction for extremality and we will give a construction of a graph that contains all possible vertices satisfying this condition, thus being an extremal graph for $k = 3$. Before we give the theorem, we will calculate the amount of different vertices such a construction should have:

We have vectors with m digits and we can freely construct vectors consisting solely of the elements $2,3,\infty$, so that leaves us with 3^m possibilities to start with. Now we consider the vectors that contain a 1, since we can have at most a single 1 in any vector we know that its position is unique in the vector, so we have m choices to place this 1. Now we can fill in the other $m-1$ free places in this vector, with again $2,3,\infty$ as possible elements, so that makes $m \cdot 3^{m-1}$ possible vectors containing a 1. This means that if we do not allow sensors to measure through other sensors, we can have at most $3^m + m \cdot 3^{m-1}$ non-sensor vertices, so this way we can construct a graph of at most $3^m + m \cdot 3^{m-1} + m - 1$ vertices after adding the sensors and subtracting the vertex that is not measured by any sensor.

We shall now present a theorem that states how only allowing vertices that suffice (3.13) will lead to an extremal graph with threshold-3 metric dimension m.

Theorem 3. *Let G^* be a graph with threshold-3 metric metric dimension $m > 0$ and let G^* be extremal in size (that is, there exists no other graph G with threshold-3 metric dimension m so that $|G| > |G^*|$). Then $|G^*| \leq 3^m + m \cdot 3^{m-1} + m - 1$.*

We shall present a rough proof for Theorem 3, due to time constraints it has not been written down as properly as possible, but the argument still holds. This is also because in Section 4 we discuss a proof (though incomplete) for extremality of a graph with threshold-k metric dimension m for general k and m.

Proof. Suppose we wish to construct a graph G with m sensors that form a threshold-3 resolving set, with as many vertices as possible and with this in mind suppose we have to choose between adding a vertex v with a distance vector $\underline{d}^{(v)}$ that has the value 1 on positions $j \in J \subseteq \{1,2,...,m\}$ and arbitrary values $x_j \neq 1$ on the remaining positions $j \in X = \{1,2,...,m\} \setminus J$.

Adding this vertex means that we can freely add any vertex that has any combination of values for $x_j \neq 1$ for $j \in X$, but we can not add any vertices that have both at least a single value ∞ in a position $j \in J$ and at least a single value 1 in any other position $i \in J, i \neq j$ and then having any combination of values for $x_j \neq 1$ for $j \in X$.

This we gain a single class of vertices that have arbitrary values for $x_j \neq 1, j \in X$, but we lose the ability to add $2^n - 1 - 1$, where $n = |J|$, classes of vertices, that have arbitrary values for $x_j \neq 1, j \in X$, since there are 2^n possibilities to choose between 1 and ∞ on n locations in the vector, but we do not count the vector that only has the value 1 on all positions $j \in J$ (since that is the vector we gain) and we do not count the vector that only has the value ∞ on all positions $j \in J$ (since we need at least one value to be 1).

This way we end up with not being able to have $2^n - 2$ classes of vertices in our construction, while only gaining a single class of the same size of vertices, by adding a vertex v whose distance vector $\underline{d}^{(v)}$ constains more than a single value 1. Since every class of vertices whose vectors contain more than a single 1, uniquely corresponds to $2^n - 2$ classes of vertices, and since $2^n - 2 \geq 1$ for $n \geq 2$, we conclude that adding vertices v such that $\mathrm{Min_2sum}(v) < 3$ can not lead to an extremal construction for a graph with a threshold-3 resolving set, *if* we can find a construction that only contains vertices v wuch that $\mathrm{Min_2sum}(v) \geq 3$. □

This means that if we can find a graph with a threshold-3 resolving set of size m that contains all vertices $\{v_{\underline{d}} : \underline{d} \in \{1,2,3,\infty\}, \mathrm{Min_2sum}(\underline{d}) \geq 3\}$, it must be extremal in size.

Now we will show that there does indeed exists a construction that attains this bound for $k = 3$ and arbitrary many sensors m and suffices the condition that $\mathrm{Min_2sum}(v) \geq 3$ for all v. This construction is similar to the construction for $k = 2$, though with some modifications to include the $\mathrm{Min_2sum}(v) \geq 3$ limitation.

Definition 6 (Construction of an extremal graph $G_3(m)$ for $k = 3$)**.** *First we define:* $S = \{1, 2, ..., m\}$. *Now let* $G_3(m) = (V, E)$ *be a graph where the vertex set* V *is given as:*

$$V = \{s_1, s_2, ..., s_m\} \cup \bigcup_{\substack{A \subseteq S \\ A \neq \emptyset}} \left[\{v_A\} \cup \left(\bigcup_{i \in A} \{v_{A,i}\} \right) \cup \bigcup_{B \subset A} \left(\{v_{A,B}\} \cup \bigcup_{i \in B} \{v_{A,B,i}\} \right) \right].$$

The edge set E *is given as:*

$$E = \bigcup_{\substack{A \subseteq S \\ A \neq \emptyset}} \left[\left(\bigcup_{i \in A} \{(v_A, v_{A,i}), (v_{A,i}, s_i)\} \right) \cup \bigcup_{B \subset A} \left(\{(v_A, v_{A,B})\} \right. \right.$$
$$\left. \left. \cup \bigcup_{i \in B} \left[\{(v_{A,B}, v_{A,B,i}), (v_{A,B,i}, s_i)\} \cup \bigcup_{j \in A \setminus B} \{(v_{A,B,i}, v_{A,\{j\},j})\} \right] \right) \right].$$

In words, this construction is as follows: Again we first create the sensor vertices $s_1, s_2, ..., s_m$. Then, as with the other cases, we start off with creating all possible nonempty subsets $A \subseteq \{1, 2, ..., m\}$ of the sensor numbers. For each $A \subseteq \{1, 2, ..., m\}$ we create a vertex v_A. Though instead of connecting it directly to the sensors $\{s_i : i \in A\}$ as we did in the other cases, we add a connecting vertex $v_{A,i}$ inbetween v_A and the sensor s_i, so we connect this vertex $v_{A,i}$ to v_A and to s_i, for each $i \in A$. So for example, if $m = 5$ we connect $v_{\{1,2,4\}}$ to $v_{\{1,2,4\},1}, v_{\{1,2,4\},2}$ and to $v_{\{1,2,4\},4}$ and we connect $v_{\{1,2,4\},1}$ to s_1, we connect $v_{\{1,2,4\},2}$ to s_2 and we connect $v_{\{1,2,4\},4}$ to s_4.

Now for every set A as above, we again create its strictly smaller subsets $B \subset A$ and for each B, we create a vertex $v_{A,B}$. As before, we connect this vertex $v_{A,B}$ to v_A, however we again do not connect $v_{A,B}$ directly to the sensors $\{s_i : i \in A\}$. As with the vertices v_A, we add connecting vertices $v_{A,B,i}$ which we connect to $v_{A,B}$ and to s_i for every $j \in B$.

Now for the final step: for each subset $A \subseteq \{1, 2, ..., m\}$ we identify its subsets $B \subset A$ that have size $|B| = 1$. These subsets B of size 1 have only one element j in them, so $B = \{j\}$. Thus they have connecting vertex $v_{A,\{j\},j}$ between them and s_j. Now for each A we consider all its subsets $B \subset A$ and for each B we take all its connecting vertices $v_{A,B,i}$ for $i \in B$. Now for each $j \in A \setminus B$, we connect the vertex $v_{A,\{j\},j}$ to all vertices $v_{A,B,i}$ for $i \in B$. We do this for every $B \subset A$, for every A.

For example, if $m = 5$ and we have $A = \{1, 2, 4, 5\}$ and $B = \{1, 4\}$, we connect $v_{\{1,2,4,5\},\{1,4\},1}$ to $v_{\{1,2,4,5\},\{2\},2}$ and to $v_{\{1,2,4,5\},\{5\},5}$ and we connect $v_{\{1,2,4,5\},\{1,4\},1}$ to $v_{\{1,2,4,5\},\{2\},2}$ and to $v_{\{1,2,4,5\},\{5\},5}$. An example of this whole conscruction for $m = 2$ is given in Figure 3.5.

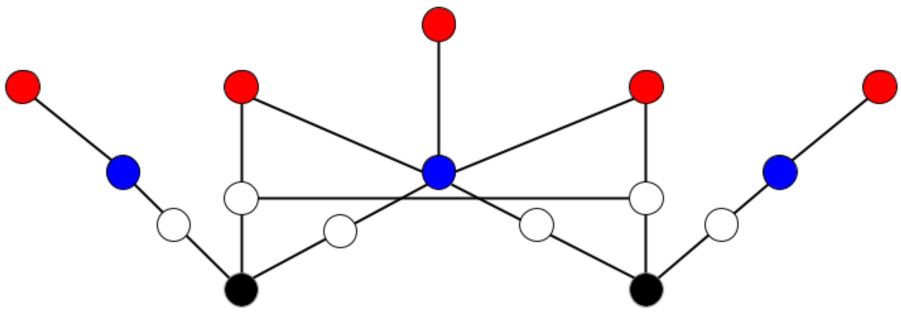


Figure 3.5: An extremal graph constructed as in Definition 6 with threshold-3 metric dimension 2. The sets next to the blue vertices denote the set A for which the blue vertex v_A is created. The vertices v_A are given in blue, the vertices $v_{A,B}$ are in red, and the connecting vertices are white.

An example of this whole contscruction for $m = 3$ generated by the code in Appendix A is given in Figure 3.6.

Now we are left to show that we can indeed construct every distance vector uniquely according to the rules described earlier. To do this, we show that each vertex we created for this graph, has its own distinct distance vector pattern.

Theorem 4. *The construction as given for* $G_3(m)$ *in Definition 6 yields a graph with threshold-3 resolving set* $\{s_1, s_2, ..., s_m\}$ *for every* $m \in \mathbb{N}$. *Furthermore,* $G_3(m)$ *is extremal in size, that is: for any graph* G *with threshold-3 metric dimension* m, *we have* $|G| \leq |G_3(m)|$.

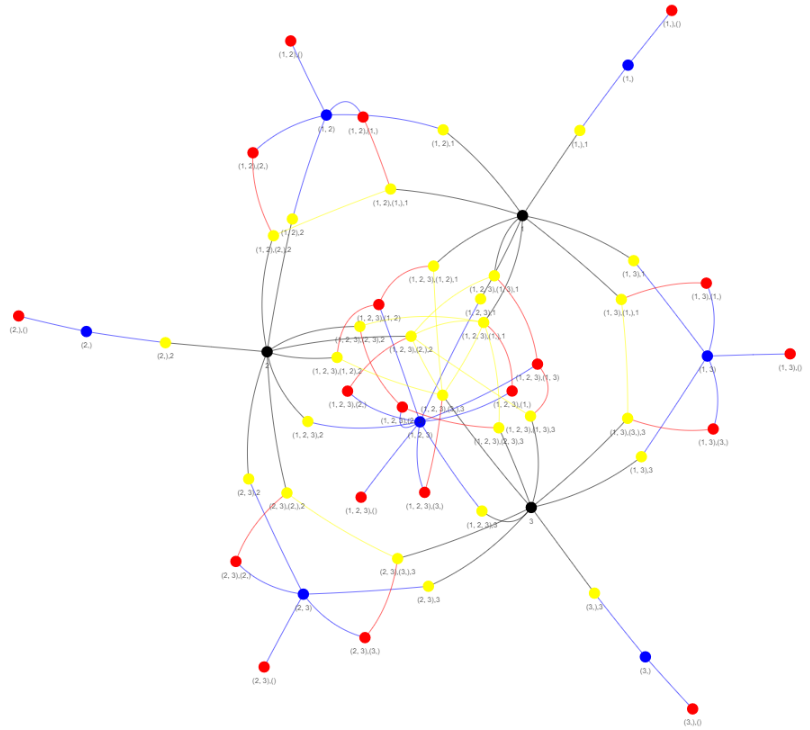


Figure 3.6: An extremal graph constructed almost the same as in Definition 6 (some edges can be left out) with threshold-3 metric dimension 3. The sets next to the blue vertices denote the set A for which the blue vertex v_A is created. The sets next to the red vertices denote the sets A, B for which the red vertex $v_{A,B}$ is created. The sets and elements next to the yellow connecting vertices denote what sets A and B they are connected to.

Proof. For simplicity, we first consider the construction *without* the edges:

$$\bigcup_{\substack{A\subseteq S\\ A\neq\emptyset}} \bigcup_{B\subset A} \bigcup_{i\in B} \bigcup_{j\in A\setminus B} \{(v_{A,B,i}, v_{A,\{j\},j})\}.$$

That is, we do not yet consider the edges that are added in the final step: the edges between the connecting edges of $v_{A,B,i}$ and the connecting edges $v_{A,\{j\},j}$ where $j \in A\setminus B$. We will consider these edges later on in this proof.

1.1: The vertices v_A.
First we calculate the distance vectors of the vertices v_A for $A \subset \{1,2,...,m\}$, due to the connecting vertices $v_{A,i}$ between v_A and s_i for $i \in A$, these vertices no longer have a direct connection to the sensors $\{s_i : i \in A\}$ and so no longer distance to these sensors. They instead have distance 2 to these sensors, through the connecting vertices.

They furthermore have distance ∞ to the sensors $\{s_i : i \not\in A\}$, this is because the vertices are only connected to the sensors $\{s_i : i \in A\}$ through their connecting vertices and through the vertices $v_{A,B}$, that are connected to the same sensors, again through connecting vertices. The only paths between v_A and $\{s_i : i \not\in A\}$ are through the sensors, and since the shortest paths between the sensors are through the vertices v_A or $v_{A,B}$ connecting multiple sensors through connecting vertices, we see that the shortest paths between the sensors are of at least length 3, and thus v_A can not be measured by a sensor in $s_i : i \not\in A$.

We thus see that the vertices v_A have distance vectors with value 2 on the positions $j \in A$ and value ∞ on the positions $j \not\in A$.

1.2: The vertices $v_{A,B}$.
These vertices are not directly connected to any sensors $\{s_i : i \in \{1,2,...,m\}\}$, they do have distance 2 to the sensors $\{s_i : i \in B\}$ through the connecting vertices $v_{A,B,i}, i \in B$. They furthermore have paths of length 3 to the sensors that are in $\{s_i : i \in A\setminus B\}$ through v_A and $v_{A,i}$ for $i \in A\setminus B$. Since the vertices $v_{A,B}$ are only connected to v_A and $v_{A,B,i}$ for $i \in B$, none of which are themselves connected to any sensors in $\{s_i : i \in A\setminus B\}$, we can see

that these paths of lenght 3 are the shortest paths to the sensors $\{s_i : i \in A \setminus B\}$. We can also determine that the vertices $v_{A,B}$ have length ∞ to the sensors $\{s_i : i \not\in A\}$, since the only paths possible to sensors in $\{s_i : \not\in A\}$ are again through the sensors $\{s_i : i \in A\}$ and since we already determined the distance between the sensors to be at least 3, we can see that $v_{A,B}$ will not be measured by sensors in $\{s_i : i \not\in A\}$.

We thus see that the distance vectors of $v_{A,B}$ have value 2 on the positions $j \in B$, distance 3 on the positions $j \in A \setminus B$ and distance ∞ on the positions $j \not\in A$. And again, since B is a proper subset of A, we must have at least one value to be distance 3.

1.3: The vertices $v_{A,i}$.
These vertices are directly connected to the sensor s_i, so they have distance 1 to sensor s_i. Since this is the only sensor they are directly connected to, the $\text{Min}_2\text{sum}(v_{A,i}) \geq 3$ condition is satisfied.

In all, these vertices $v_{A,B,i}$ are only connected to the sensor s_i, and to v_A, to the only other sensors they can have connections with, must be the sensors that measure v_A, since the paths between the sensors themselves are of length at least 3.

Now since v_A has distance 2 to the sensors that measure it, we can conclude that $v_{A,i}$ has distance 3 to the sensors $\{s_j : j \in A\}$, except for the sensor s_i, since that sensor is connected directly to $v_{A,i}$.

And so we see that the distance vector of $v_{A,i}$ has value 1 on position i, value 3 on positions $j \in A, j \neq i$ and value ∞ on positions $j \not\in A$.

1.4: The vertices $v_{A,B,i}$.
These vertices are directly connected to the sensor s_i, so they have distance 1 to the sensor s_i. The only other connection they have is to $v_{A,B}$, which has distance 2 to all sensors $\{s_j : j \in B\}$ and distance 3 or ∞ to any other sensors. So the only sensors that will measure $v_{A,B,i}$ through $v_{A,B}$ are the sensors that have distance 2 to $v_{A,B}$, which will measure $v_{A,B,i}$ with distance 3. So $v_{A,B,i}$ has distance 3 to all sensors $\{s_j : j \in B\}$.

From now on, we do include the edges:

$$\bigcup_{\substack{A\subseteq S\\ A\neq\emptyset}} \bigcup_{B\subset A} \bigcup_{i\in B} \bigcup_{j\in A\setminus B} \{(v_{A,B,i}, v_{A,\{j\},j})\}.$$

This means, that for every $B \subset A$, we connect all $\{v_{A,B,i} : i \in B\}$ to all $\{v_{A,\{j\},j} : j \in A \setminus B\}$, so we connect the connecting vertices of every $B \subset A$ to the connecting vertices of the subsets containing only the element j, for every j that is in A but not in B.

First we will show that the addition of these new edges does not change the distance vectors of the vertices we discussed earlier on in this proof. We note first that the distance between the sensors still remains at least 3, since these edges are only between the connecting vertices of any $v_{A,B}$. So they only create a path from a sensor, say s_a, to a connecting vertex $v_{A,B,a}$, to the connecting vertex of a set B that only contains one element, say $v_{A,\{b\},b}$, and then finally to the sensor s_b. This path is a path of length 3, and since paths like these are the only paths created by these new edges, we can conclude that the distance between the sensors remains at least 3, so any argument that relied on the distance of the sensors being at least 3 still holds.

2.1: The vertices v_A.
Now since these new edges are only between connecting vertices of sensors that are in the same set $A \subseteq \{1,2,...,m\}$, we can also see that the distance vectors of v_A remain the same, since they still only connect to the sensors in $\{s_i : i \in A\}$.

2.2: The vertices $v_{A,B}$.
The new edges do form new paths from $v_{A,B}$ to sensors in $\{s_j : j \in A \setminus B\}$, since they make connections between the connecting vertices of $v_{A,B}$ and the connecting vertices $v_{A,\{j\},j}$ where $j \in A \setminus B$. These paths are of length 3, however and since we already had that $v_{A,B}$ has a distance of 3 to the sensors $\{s_j : j \in A \setminus B\}$, the distance 3 does not change.

2.3: The vertices $v_{A,i}$.
The vertices $v_{A,i}$ have only connections to the sensor s_i and to v_A, and since the distance to any connecting vertex $v_{A,B,j}$ is 3 (through v_A and $v_{A,B}$, the only path possible through v_A), nothing changes for $v_{A,i}$.

2.4: The vertices $v_{A,B,i}$.

Now for $v_{A,B,i}$, their distance of 1 to s_i remains. For each A we now make two distinctions: the subsets $B \subset A$ with multiple elements, and the 'singular' subsets of $\{j\} \subset A$ containing only j.
First we consider the subsets B with multiple elements. The connecting vertices $v_{A,B,i}$ are now connected to all connecting vertices $v_{A,\{j\},j}$, where j is in A but not in B, this means that there now exists a path from $v_{A,B,i}$ to all $\{s_j : j \in A \setminus B\}$ of length 2, namely $v_{A,B,i}, v_{A,\{j\},j}, s_j$, this means that all $v_{A,B,i}$ have distance 2 to the sensors $\{s_j : j \in A \setminus B\}$, at least when B contains multiple elements.

$v_{A,B,i}$ also retains it's distance of 3 to $\{s_j : j \in B, j \neq i\}$, since we do not add an edge between $v_{A,B,i}$ and $v_{A,\{j\},j}$ if $j \in B$, as per the definition of the construction. So for B containing multiple elements, $v_{A,B,i}$ has distance 1 to s_i, distance 2 to $\{s_j : j \in A \setminus B\}$, distance 3 to $\{s_j : j \in B, j \neq i\}$ and distance ∞ to $\{s_j :\not\in A\}$ (since the only possible paths to sensors $\{s_j : j \not\in A\}$ are still through the sensors $\{s_j : j \in A\}$ themselves).

Now for $B = \{j\}$, the case when B consists of only a single element j. Not only do we connect this vertex $v_{A,\{j\},j}$ to all $v_{A,B,i}$ that have $j \in A \setminus B$, we also connect it to all $v_{A,\{k\},k}$ where $k \in A \setminus \{j\}$. In both cases do these connections add paths of length 2 from $v_{A,\{j\},j}$ to sensors $\{s_i : i \in A \setminus \{j\}\}$. So we get distance 2 to all sensors $\{s_i : i \in A \setminus \{j\}\}$. So these vertices $v_{A,\{j\},j}$ also have distance 1 to s_j, distance 2 to $\{s_i : i \in A \setminus B\}$ and again distance ∞ to $\{s_i : i \not\in A\}$ (since $\{s_i : i \in B, i \neq j\}$ is empty we need not consider these cases).

We thus conclude that $v_{A,B,i}$ has a distance vector with 1 on position i, 2 on positions $j \in A \setminus B\}$, 3 on positions $j \in B, j \neq i$ and ∞ on positions $j \not\in A$.

Conclusion.
We now finally summarize that for $A \subseteq \{1,2,...,m\}, A \neq \emptyset$, we have that v_A has distance vector with 2 on positions $j \in A$ and ∞ on positions $j \not\in A$, these are all distance vectors containing all permutations of only at least a single 2 and any amount of value ∞.

For $B \subset A$ we have that $v_{A,B}$ has a distance vector with 2 on positions $j \in B$, 3 on positions $j \in A \setminus B$ and ∞ on positions $j \not\in A$, these are all the distance vectors containing all permutations of only at least a single 3 and any amount of values 2 and ∞.

For $i \in \{1,2,...,m\}$ we have that $v_{A,i}$ has a distance vector with 1 on position i, 3 on positions $j \in A, j \neq i$ and ∞ on positions $j \not\in A$, these are all the distance vectors containing all the permutations of precisely a single value 1, and any amount of values 3 and ∞.

For $i \in \{1,2,...,m\}$ we have that $v_{A,B,i}$ has a distance vector with 1 on position i, 2 on positions $j \in A \setminus B$, 3 on positions $j \in B, j \neq i$ and ∞ on positions $j \not\in A$, these are all the distance vectors containing all the permutations of precisely a single value 1, at least a single value 2 and any amount of values 3 and ∞.

From this we see that we uniquely represent all possible distance vectors, whose vertices v suffice the $\text{Min}_2\text{sum}(v_{A,B,i}) \geq 3$ condition. We have thus constructed a graph that has $\{s_1, s_2, ..., s_m\}$ as a threshold-3 resolving set of size m and is extremal in size. □

4

Extremal construction for general k and m

The three constructions given in Section 3 are designed to create an extremal graph of metric dimension 1,2 and 3 respectively. Though they do indeed yield extremal graphs, they only work for $k = 1, 2$ or 3.

To this end, we want to find a construction that generalizes for all $k > 0$, such a construction has been found and will be discussed in this section. First we shall introduce a general notion on what makes such a construction extremal.

4.1. Extremality of a graph with threshold-k metric dimension m

Before we state the main theorem of this section, we shall introduce the concept of overflow, denoted as f. If sensors are sufficiently close to each other, it may occur that one sensor can measure 'through' another sensor. What we mean by this is the following: Suppose sensors s_i and s_j have range k and distance $k - f$ to each other, then if $f > 0$, we get an overflow, in the sense that sensor s_j can measure vertices that are within range f from sensor s_i, and vice versa. Lemma 1 will elaborate on this.

For now we shall state the main theorem of this section:

Theorem 5. *For given values m and k, an upper bound for a graph with threshold-k metric dimension m is given by the cardinality of the set of vectors* $\left\{\underline{d} \in \{k+1\}^m : \text{Min}_2\text{sum}\left(\underline{d}\right) \geq k - f(k), \max\left(\underline{d}\right) - \min\left(\underline{d}\right) \leq k - f(k)\right\}$. *Where* $f(k)$ *can either be* $f(k) = \left\lfloor \frac{k-2}{3} \right\rfloor$ *or* $f(k) = \left\lfloor \frac{k-1}{3} \right\rfloor$.

We shall now present a series of lemmas and corollaries to make this theorem assumable, due to time constraints, the proof of Theorem 5 will remain partially incomplete. We will also make use of Lemma 3.2 of the paper by Hernando [6] as a part of our proof.

Lemma 1. *Let* $G = (V, E)$ *be a graph with threshold-k metric dimension m and threshold-k resolving set* $\{s_1, s_2, ..., s_m\}$. *Suppose that* $v \in V$ *and that* $\text{Min}_2\text{sum}(v) = k - f$, *where* $f \in \mathbb{N}, f \geq 0$. *By the definition of* Min_2sum, *there exist elements* $d_i^{(v)}$ *and* $d_j^{(v)}$ *in the distance vector* $\underline{d}^{(v)} \in \{k+1\}^m$ *of* v, *such that* $d_i^{(v)} + d_j^{(j)} = k - f$. *Then for all* $w \in V$ *that have distance vector* $\underline{d}^{(w)} \in \{k+1\}^m$ *we have:*

$$\max\left(d_i^{(w)}, d_j^{(w)}\right) - \min\left(d_i^{(w)}, d_j^{(w)}\right) \leq k - f. \tag{4.1}$$

Proof. Let $G = (V, E)$ be a graph with threshold-k metric dimension m with sensor set $S = \{s_1, s_2, ..., s_m\}$ and suppose that $v \in V$ and that $\text{Min}_2\text{sum}(v) = k - f$, where $f \in \mathbb{N}, f \geq 0$.

Now per the definition of Min_2sum, there exist elements $d_i^{(v)}$ and $d_j^{(v)}$ in the distance vector $\underline{d}^{(v)} \in \{k+1\}^m$ of v, such that $d_i^{(v)} + d_j^{(j)} = k - f$ and thus we have sensors $s_i, s_j \in S$ so that $d(s_i, v) + d(s_j, v) = k - f$. This means that the distance $d(s_i, s_j)$ between s_i and s_j is at most $k - f$, since there exists a path from s_i to v of length $d(s_i, v)$ and a path from v to s_j of length $d(s_j, v)$. Combining the two paths yields a path of length $d(s_i, v) + d(s_j, v) = k - f$.

This means that any vertex $w \in V$ that has distance $d(s_i, w) = r_i \leq f$ to sensor s_i, also has a distance of at most $r_i + k - f$ to sensor s_j, since there exists a path from w to s_i of length r_i and a path from s_i to s_j of length $k - f$, combining these paths yields a path of length $r_i + k - f$, and since $r_i \leq f$, we have that $r_i + k - f \leq k$ and thus w is measured by s_j.

Thus we have shown that if we have that if $d(s_i, w) \le f$, we get that $d(s_j, w) \le d(s_i, w) + k - f$, or:

$$d(s_j, w) - d(s_i, w) \le k - f, \tag{4.2}$$

and by symmetry of the previous argument we also get:

$$d(s_i, w) - d(s_j, w) \le k - f. \tag{4.3}$$

This yields the system:

$$\begin{cases} d(s_j, w) - d(s_i, w) \le k - f \\ d(s_i, w) - d(s_j, w) \le k - f \end{cases}. \tag{4.4}$$

This means that the difference between $d(s_i, w)$ and $d(s_j, w)$ may be no more than $k - f$, which indeed implies:

$$\max\left(d_i^{(w)}, d_j^{(w)}\right) - \min\left(d_i^{(w)}, d_j^{(w)}\right) \le k - f,$$

as required. □

We can generalize this over all pairs i, j within a distance vector:

Corollary 1. *Let $G = (V, E)$ be a graph with threshold-k metric dimension m and threshold-k resolving set $\{s_1, s_2, ..., s_m\}$. If for all pairs $(i, j) \in \binom{m}{2}$ there exists at least one $v = v_{i,j} \in V$ so that $Min_2sum(v) = d_i^{(v)} + d_j^{(v)} = k - f$, then for all $w \in G$:*

$$\max\left(d_1^{(w)}, d_2^{(w)}, ..., d_m^{(w)}\right) - \min\left(d_1^{(w)}, d_2^{(w)}, ..., d_m^{(w)}\right) \le k - f. \tag{4.5}$$

Or equivalently:

$$\max\left(\underline{d}^{(w)}\right) - \min\left(\underline{d}^{(w)}\right) \le k - f. \tag{4.6}$$

Proof. We can apply Lemma 1 pairwise to every $(i, j) \in \binom{m}{2}$ to obtain:

$$\max\left(d_i^{(w)}, d_j^{(w)}\right) - \min\left(d_i^{(w)}, d_j^{(w)}\right) \le k - f, \tag{4.7}$$

for every $(i, j) \in \binom{m}{2}$.
Since $\max\left(\underline{d}^{(w)}\right)$ and $\min\left(\underline{d}^{(w)}\right)$ must suffice (4.7) for the exact pair (i, j) that contains the maximum and minimum of $\underline{d}^{(w)}$, we immediately obtain that indeed:

$$\max\left(\underline{d}^{(w)}\right) - \min\left(\underline{d}^{(w)}\right) \le k - f,$$

as required. □

4.1.1. Extremality and optimal f

This corollary gives us a way to determine what possible combinations can fit in a graph with threshold-k metric dimension m. There is one assumption that we make here that we shall not prove. This assumption is that an extremal combination of vertices that suffice the condition in Lemma 1 has a form of symmetry to it, that is: that the condition of Lemma 1 must apply pairwise to all sensors in such a graph for the same amount of overflow f, thus making the distance between all sensors equal.

Assuming that the amount of overflow should be the same between all pairs of sensors, finding an extremal graph of threshold-k metric dimension m corresponds with finding the value f such that the amount different distance vectors $\underline{d}$ such that the cardinality of the set
$\left\{\underline{d} \in \{k+1\}^m : \text{Min}_2\text{sum}(\underline{d}) \ge k - f, \max(\underline{d}) - \min(\underline{d}) \le k - f\right\}$ is maximal.

We can calculate this optimal f for $m = 2$ and general $k > 0$, but first we shall introduce some notation. We let $\mathscr{A}_{m,k}(f)$ be the set of all vectors $\underline{d} \in \{k+1\}^m$ that suffice the equations:

$$\begin{cases} \text{Min}_2\text{sum}(\underline{d}) \ge k - f \\ \max(\underline{d}) - \min(\underline{d}) \le k - f \end{cases}. \tag{4.8}$$

We shall also let $A_{m,k}(f)$ be the cardinality $|\mathscr{A}_{m,k}(f)|$. We give the optimal value that we will calculate in the form of a claim:

Claim 2. *The value f for which the value* $A_{2,k}(f)$ *is optimal for arbitrary k, is given as:* $f = \left\lfloor \frac{k-2}{3} \right\rfloor$ *or* $f = \left\lfloor \frac{k-1}{3} \right\rfloor$.

We shall prove this by means of determining the difference $A_{2,k}(f) - A_{2,k}(f-1)$, that is, we determine if increasing the value of f yields a positive or negative difference of allowed vectors. To this end, we calculate the amount of vectors that we gain by increasing f, thus the size of $\mathscr{A}_{2,k}(f) \setminus \mathscr{A}_{2,k}(f-1)$. And we calculate the amount of vectors that we lose by increasing f, thus the size of $\mathscr{A}_{2,k}(f-1) \setminus \mathscr{A}_{2,k}(f)$. We then finally determine: $A_{2,k}(f) - A_{2,k}(f-1) = |\mathscr{A}_{2,k}(f) \setminus \mathscr{A}_{2,k}(f-1)| - |\mathscr{A}_{2,k}(f-1) \setminus \mathscr{A}_{2,k}(f)|$.

Proof. Suppose we have overflow $f-1$, where f is some integer. Then we allow all distance vectors $\underline{d} \in \{k+1\}^2$ that suffice $\text{Min}_2\text{sum}(\underline{d}) \geq k - f$ and $\max(\underline{d}) - \min(\underline{d}) \leq k - f$, that is: all vectors (d_1, d_2) where d_1 and d_2 take values in $\{1, 2, ..., k+1\}$ and they all must have that $d_1 + d_2 \geq k - (f-1) = k - f + 1$ and $\max(d_1, d_2) - \min(d_1, d_2) \leq k - (f-1) = k - f + 1$.

Now suppose we increase the overflow by 1, that is: we now have overflow f instead of $f-1$, we can consider the distance vectors that we gain and lose by this change.

First the vectors we gain: we now have that $d_1 + d_2 \geq k - f$ instead of $d_1 + d_2 \geq k - f + 1$, so we gain all vectors with $d_1 + d_2 = k - f$, since both d_1 and d_2 are larger than 0, we obtain $k - f - 1$ different pairs that suffice. Now since these d_1 and d_2 can differ at most $k - f - 2$ (for example if $d_1 = 1$ and $d_2 = k - f - 1$ so that $d_1 + d_2 = k - f$), we always have that $\max(d_1, d_2) - \min(d_1, d_2) \leq k - f$.

Now the vectors we lose: we now have that $\max(d_1, d_2) - \min(d_1, d_2) \leq k - f$ instead of $\max(d_1, d_2) - \min(d_1, d_2) \leq k - f + 1$, this means we lose the vectors that suffice $\max(d_1, d_2) - \min(d_1, d_2) = k - f$, these vectors either have $d_1 > d_2$ or $d_2 > d_1$ (since vectors with $d_1 = d_2$ will never break this condition). Let us consider the case $d_1 > d_2$, then we must count the combinations d_1 and d_2 such that $d_1 - d_2 = k - f$. Since we must have, per formulation of the problem, that $d_1 \leq k$ and $d_2 \geq 1$, we count from $d_1 = k$ and $d_2 = f$ so that $k - f = k - f$ to $d_1 = k - f + 1$ and $d_2 = 1$, so that $k - f + 1 - 1 = k - f$. These are f different combinations. Now since the case $d_2 > d_1$ is symmetrical, we end up with $2 \cdot f$ different vectors that we lose.

To conclude: if we increase the overflow from $f-1$ to f, we gain $k - f - 1$ vectors and we lose $2f$ vectors, thus increasing our overflow to f is only a net gain if: $k - f - 1 > 2f \Rightarrow 3f < k - 1$ or at least not a loss if $k - f - 1 \geq 2f \Rightarrow 3f \leq k - 1$, so we want to find the maximum f such that still $3f < k - 1$ or $3f \leq k - 1$ (if we don't increasing the overflow when it does not change the total amount of vectors). The functions $f(k) = \left\lfloor \frac{k-2}{3} \right\rfloor$ and $f(k) = \left\lfloor \frac{k-1}{3} \right\rfloor$ suffice these cases, respectively. □

Now from a numerical analysis in python, it turns out that the optimal value for f is independent from the amount of sensors m, we do not prove this, but in the last week of research we did find a paper that confirms this [6]. In this paper, a proof of this is given in Lemma 3.2, though with different notation.

4.2. The construction

We define our construction as follows:

Definition 7 (Construction of an extremal graph $G(k,m)$ for general k)**.** *First we define* $S = \{1, 2, ..., m\}$. *Now let* $G(k,m) = (V,E)$ *be a graph where the vertex set V is given by:*

$$V := \{s_1, s_2, ..., s_m\} \cup \{v_{\underline{d}} : \underline{d} \in \{k+1\}^m, \text{Min}_2\text{sum}(\underline{d}) \geq k - f(k), \max(\underline{d}) + f(k) \leq \min(\underline{d}) + k\}. \tag{4.9}$$

Where $f(k)$ *is defined as* $f(k) = \left\lfloor \frac{k-2}{3} \right\rfloor$.
The edge set E is given by:

$$E := \left(\bigcup_{i \in V} \bigcup_{\substack{j \in V \\ ||\underline{d}^{(i)} - \underline{d}^{(j)}||_\infty = 1}} \{(i,j)\} \right) \cup \bigcup_{\substack{v_{\underline{d}} \in V \\ d_i = 1}} \{(v_{\underline{d}}, s_i)\}. \tag{4.10}$$

In words, this construction first creates a set of as many different distance vectors we believe can exist together in a graph with threshold-k metric dimension m. This selection is based on the results of Corollary 1 as discussed in Section 4.1.1.

We have determined that the maximum amount of different distance vectors we can attain for the vertices in a graph is when, for given k and m, we consider precisely all the distance vectors:

$$\{\underline{d} \in \{k+1\}^m : \text{Min}_2\text{sum}(\underline{d}) \geq k - f(k), \max(\underline{d}) + f(k) \leq \min(\underline{d}) + k\}$$

Where $f(k)$ can be either $f(k) = \left\lfloor \frac{k-1}{3} \right\rfloor$ or $f(k) = \left\lfloor \frac{k-2}{3} \right\rfloor$. Both versions of $f(k)$ lead to an equal, extremal amount of different possible distance vectors for k and m, though for consistency, we shall define: $f(k) = \left\lfloor \frac{k-2}{3} \right\rfloor$.

Now we create an m-dimensional grid in which we place the vertices as if their distance vectores were the coordinates. We then connect all the vertices if their distance vectors have distance 1 to one another in the infinity metric, in other words: we connect vertices u and v if and only if: the individual coordinates of their distance vectors $\underline{d}^{(u)}$ and $\underline{d}^{(v)}$ have difference at most 1 and the distance vectors $\underline{d}^{(u)}$ and $\underline{d}^{(v)}$ are not equal to each other.

For example, if we have $k = 3$ and $m = 3$: we would connect (2,2,2) and (2,3,3), but not (2,2,2) and (2,4,4).

An example of such a graph with $k = 3$ and $m = 2$ (withouth the diagonal edges in the 'internal' vertices) is given in Figure 4.1 Another example for $k = 5$ and $m = 2$ is given in Figure 4.2. Note how we now allow

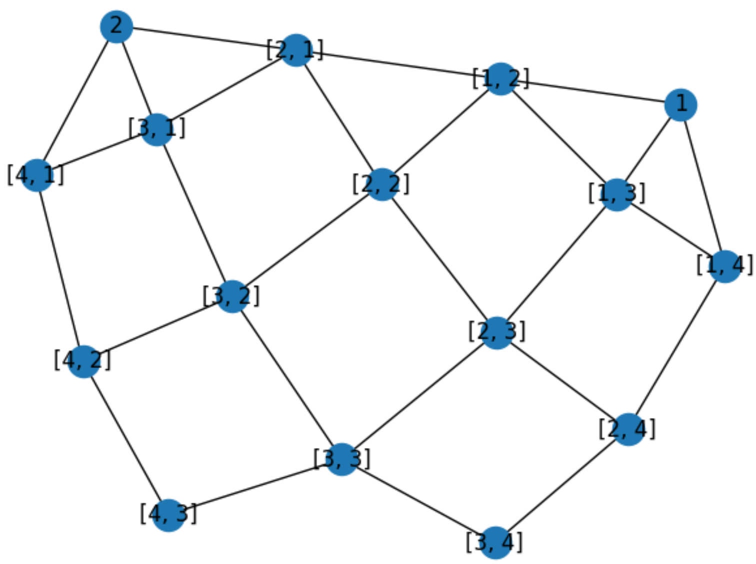


Figure 4.1: A graph with threshold-3 metric dimension 2 as constructed with Definition 7, so with $k = 3$ and $m = 2$. We did not include the diagonal edges on the 'internal' vertices.

distance vectors (1,3),(2,2) and (3,1) and not (1,6) and (6,1).

As stated before, this construction can be interpreted as an m-dimensional grid, or an m-dimensional hypercube with some 'parts' missing.

We shall now show that this construction yields a graph that has $\{s_1, s_2, ..., s_m\}$ as a threshold-k resolving subset.

Theorem 6. *The graph $G(k, m)$ as defined in Definition 7 has a threshold-k resolving set $\{s_1, s_2, ..., s_m\}$ of size m.*

Proof. To show this, we have to show that each vertex $v_{\underline{d}}$ created for a distance vector $\underline{d}$, does indeed have that distance vector, that is: for every element $d_i \in \underline{d}$, we have to show that $d(s_i, v) = d_i$.

Now let $G = (V, E)$ be a graph as constructed in Definition 7 with given k and m, let f be some integer such that $f < k$ and let $v_{\underline{d}} \in V$ and consider some element $d_i \in \underline{d}$, we are to show that $d(v_{\underline{d}}, s_i) = d_i$
Now if $d_i = 1$, we have per Definition 7, that $v_{\underline{d}}$ is connected to s_i and so $d(v_{\underline{d}}, s_i) = 1$, as required.

Now suppose $d_i \geq 2$, we need to show that $d(v_{\underline{d}}, s_i) = d_i$, to this end, we need to show that there exists a path from s_i to $v_{\underline{d}}$ of length d_i, and that no shorter path exists.

Existence of a path

First we show that there exists a path from s_i to $v_{\underline{d}}$ of length d_i. Since $v_{\underline{d}} \in V$, we know that $\underline{d}$ suffices both $\text{Min}_2\text{sum}(\underline{d}) \geq k - f$ and $\max(\underline{d}) - \min(\underline{d}) \leq k - f$. If we write $\text{Min}_2\text{sum} = \min(\underline{d}) + \min_2(\underline{d})$, where $\min_2(\underline{d})$

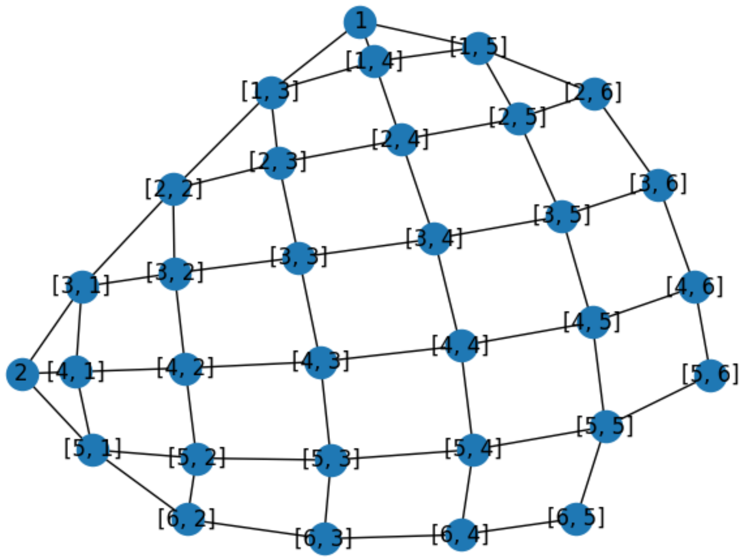


Figure 4.2: A graph with threshold-5 metric dimension 2 as constructed with Definition 7, so with $k = 5$ and $m = 2$. We did not include the diagonal edges on the 'internal' vertices.

represents the second lowest element in $\underline{d}$, which is possibly equal to $\min(\underline{d})$, we can rephrase this as:

$$\begin{cases} \min_2(\underline{d}) + \min(\underline{d}) \geq k - f \\ \max(\underline{d}) - \min(\underline{d}) \leq k - f \end{cases} . \tag{4.11}$$

Now we wish to find a neighbor $v_{\underline{d}'}$ of $v_{\underline{d}}$ that has the property that $d'_i = d_i - 1$. Since we also must have that $||\underline{d} - \underline{d}'||_\infty = 1$, all elements other than the i-th element may also differ by at most 1. We also must have that $v_{\underline{d}'} \in V$, so that $\underline{d}'$ must suffice both equations from System 4.11:

$$\begin{cases} \min_2(\underline{d}') + \min(\underline{d}') \geq k - f \\ \max(\underline{d}') - \min(\underline{d}') \leq k - f \end{cases} . \tag{4.12}$$

Case 1: $d_i > \min_2(\underline{d})$

First consider the case that $d_i > \min_2(\underline{d})$, then surely $d'_i = d_i - 1 \geq \min_2(\underline{d}) \geq \min(\underline{d})$, so that $d'_i \geq \min_2(\underline{d})$, this means that if we take $\underline{d}'$ such that $d'_j = d_j$ for $j \in \{1, 2, ..., m\}, j \neq i$ and $d'_i = d_i - 1$, we will find that $\min(\underline{d}') = \min(\underline{d})$ (since d'_i is the only element that changes and it can't become smaller than $\min(\underline{d})$) and $\max(\underline{d}') \leq \max(\underline{d})$ (since the only change of the vector consists of lowering an element) and $\min_2(\underline{d}') = \min_2(\underline{d})$ (again, since d'_i is the only element that changes and it can't become smaller than $\min_2(\underline{d})$.

So since $\min_2(\underline{d}') = \min_2(\underline{d})$, $\min(\underline{d}') = \min(\underline{d})$ and $\min_2(\underline{d}) + \min(\underline{d}) \geq k - f$, we get that:

$$\min_2(\underline{d}') + \min(\underline{d}') \geq k - f.$$

As required.

In the same way, since $\max(\underline{d}') \leq \max(\underline{d})$, $\min(\underline{d}') = \min(\underline{d})$ and $\max(\underline{d}) \leq k - f + \min(\underline{d})$ we find:

$$\max(\underline{d}') - \min(\underline{d}') \leq k - f.$$

As required.

And since the only element in $\underline{d}'$ that differs from $\underline{d}$ is d'_i, and it differs by only one, we suffice

$$||\underline{d} - \underline{d}'||_\infty = 1.$$

Thus we conclude that there exists a neighbor $\underline{d}'$ of $\underline{d}$ that has $d'_i = d_i - 1$ for $d_i > \min_2(\underline{d})$.

Case 2: $d_i = \min_2(\underline{d})$ and $d_i > \min(\underline{d})$
Now we consider what happens when $d_i = \min_2(\underline{d})$ and $d_i > \min(\underline{d})$. Now $d_i' = d_i - 1 \geq \min(\underline{d})$ and also $d_i' = d_i - 1 < \min_2(\underline{d})$. This means that if we take $\underline{d'}$ such that $d_j' = d_j$ for $j \in \{1,2,...,m\}, j \neq i$ and $d_i' = d_i - 1$, as we did earlier, we get that $\min(\underline{d'}) = \min(\underline{d})$ and $\min_2(\underline{d'}) < \min_2(\underline{d})$ and we might not suffice the condition $\min_2(\underline{d'}) + \min(\underline{d'}) \geq k - f$ since $\min_2(\underline{d'})$ has been lowered by 1. To circumvent this, we identify the unique $l \in \{1,2,...,m\}$ so that $d_l = \min(\underline{d})$. We know that this l is unique, since we assumed that $\min(\underline{d}) < \min_2(\underline{d})$ in this case ($d_i = \min_2(\underline{d})$ and $d_i > \min(\underline{d})$).

After we find the l so that $d_l = \min(\underline{d})$ and if $\min_2(\underline{d'}) + \min(\underline{d'}) < k - f$ we raise this value d_l by 1 (thus taking $d_l' = d_j + 1$), so that $\min(\underline{d})$ increases by 1 to compensate for the decreasing of $\min_2(\underline{d})$ by 1 and thus sufficing

$$\min_2(\underline{d'}) + \min(\underline{d'}) \geq k - f.$$

So if we now take $\underline{d'}$ such that $d_j' = d_j$ for $j \in \{1,2,...,m\}, j \neq i$ and $d_i' = d_i - 1$ and we change d_l' to: $d_l' = d_l + 1$ for $d_l = \min(\underline{d})$ if otherwise $\min_2(\underline{d'}) + \min(\underline{d'}) < k - f$, we get indeed

$$\min_2(\underline{d'}) + \min(\underline{d'}) \geq k - f,$$

as required.

Since the maximum did not increase (since we only increased the minimum of $\underline{d}$ which was unique and we lowered the value d_i), we have $\max(\underline{d'}) \leq \max(\underline{d})$. This in combination with the fact that $\min(\underline{d'}) = \min(\underline{d})$ and $\max(\underline{d}) - \min(\underline{d}) \leq k - f$. we obtain:

$$\max(\underline{d'}) - \min(\underline{d'}) \leq k - f,$$

as required.

And since again we only changed elements by value 1, we also suffice

$$||\underline{d} - \underline{d'}||_\infty = 1.$$

Thus we conclude that there exists a neighbor $\underline{d'}$ of $\underline{d}$ that has $d_i' = d_i - 1$ for $d_i = \min_2(\underline{d})$ and $d_i > \min(\underline{d})$.

Case 3: $d_i \leq \min_2(\underline{d})$ and $d_i = \min(\underline{d})$
Now we finally look at the case that $d_i \leq \min_2(\underline{d})$ and $d_i = \min(\underline{d})$. Now the lowest two values in $\underline{d}$ are $d_i = \min(\underline{d})$ and secondly $\min_2(\underline{d})$, which may or may not be equal to $\min(\underline{d})$. If we decrease d_i by one, $d_i' = d_i - 1$ will still be the lowest value and $\min_2(\underline{d'}) = \min_2(\underline{d})$ the second lowest.

By setting $d_i' = d_i - 1$, two things may happen: since $\min(\underline{d})$ decreases, it could happen that $\min_2(\underline{d'}) + \min(\underline{d'}) \geq k - f$ no longer holds, this is the case when $\min_2(\underline{d}) + \min(\underline{d}) = k - f$. It could also happen that $\max(\underline{d'}) - \min(\underline{d'}) \leq k - f$ no longer holds, this is the case when $\max(\underline{d}) - \min(\underline{d}) = k - f$. We consider these possible cases.

Case 3.1: $\min_2(\underline{d}) + \min(\underline{d}) = k - f$ and $\max(\underline{d}) - \min(\underline{d}) < k - f$
Now if $\min_2(\underline{d}) = k - f - \min(\underline{d})$ and $\max(\underline{d}) < k - f + \min(\underline{d})$, we can increase the values of all d_l, for which $d_l = \min_2(\underline{d})$. We then take $\underline{d'}$ such that $d_j' = d_j$ for $j \in \{1,2,...,m\}, j \neq i$ and $d_i' = d_i - 1$ and we change all d_l' to: $d_l' = d_l + 1$ for all $d_l = \min_2(\underline{d})$. Then through a very similar argument as in the case of $d_i = \min_2(\underline{d})$ and $d_i > \min(\underline{d})$, we get that $\min_2(\underline{d'}) + \min(\underline{d'}) \geq k - f$ as we require.

Now to show that $\max(\underline{d'}) - \min(\underline{d'}) \leq k - f$, even if we increase the second lowest values. Since the only values we increase are all d_l such that $d_l = \min_2(\underline{d})$ and we increase them by 1, it may happen that $\max(\underline{d'}) = \max(\underline{d}) + 1$, this is only the case if we have that $\min_2(\underline{d}) = \max(\underline{d})$, because then we increase some value(s) d_l that have $d_l = \min_2(\underline{d})$, and then $d_l' = d_l + 1$ becomes the new maximum.

Thus to summarize our situation, we have: $\max(\underline{d}) = \min_2(\underline{d})$, $\min_2(\underline{d'}) = \max(\underline{d'}) = \min_2(\underline{d}) + 1 = \max(\underline{d}) + 1$, $\min(\underline{d'}) = \min(\underline{d}) - 1$ and we have the system:

$$\begin{cases} \min(\underline{d}) + \min_2(\underline{d}) = k - f \\ \max(\underline{d}) - \min(\underline{d}) < k - f \end{cases} . \tag{4.13}$$

We wish to show that $\max(\underline{d'}) - \min(\underline{d'}) \leq k - f$, or equivalently: that assuming that $\max(\underline{d'}) - \min(\underline{d'}) > k - f$ will lead to a contradiction.

Thus we write: $\max(\underline{d}') - \min(\underline{d}') = \max(\underline{d}) - \min(\underline{d}) + 2 = \min_2(\underline{d}) - \min(\underline{d}) + 2$. Now if we make the assumption that $\max(\underline{d}') - \min(\underline{d}') > k - f$, we get that $\min_2(\underline{d}) - \min(\underline{d}) + 2 > k - f = \min(\underline{d}) + \min_2(\underline{d})$, thus: $\min_2(\underline{d}) - \min(\underline{d}) + 2 > \min(\underline{d}) + \min_2(\underline{d})$ which implies that $2 - \min(\underline{d}) > \min(\underline{d})$, so that $2 > 2\min(\underline{d})$ which would mean that $\min(\underline{d}) < 1$, however, we assumed that $d_i = \min(\underline{d})$ and we assumend that $d_i \geq 2$, so this is a contradiction and thus we have that $\max(\underline{d}') - \min(\underline{d}') \leq k - f$ as required.

If we don't have that $\min_2(\underline{d}) = \max(\underline{d})$, we will have that $\max(\underline{d}') = \max(\underline{d})$ and thus we will have that: $\max(\underline{d}') - \min(\underline{d}') = \max(\underline{d}) - \min(\underline{d}) + 1$ and since $\max(\underline{d}) - \min(\underline{d}) < k - f$, we have that $\max(\underline{d}) - \min(\underline{d}) + 1 \leq k - f$ and so indeed $\max(\underline{d}') - \min(\underline{d}') \leq k - f$.

Case 3.2: $\min_2(\underline{d}) + \min(\underline{d}) > k - f$ and $\max(\underline{d}) - \min(\underline{d}) = k - f$
If we now have that $\min_2(\underline{d}) + \min(\underline{d}) > k - f$ and $\max(\underline{d}) - \min(\underline{d}) = k - f$, we can lower the values of all d_l for which $d_l = \max \underline{d}$. We can then take $\underline{d}'$ such that $d'_j = d_j$ for $j \in \{1, 2, ..., m\}, j \neq i$ and $d'_i = d_i - 1$ and we change all d'_l to: $d'_l = d_l - 1$ for all $d_l = \max(\underline{d})$. Again through similar arguments as in the case of $d_i = \min_2(\underline{d})$ and $d_i > \min(\underline{d})$, we get that $\max(\underline{d}') - \min(\underline{d}') \leq k - f$. This time we wish to show that we also suffice $\min(\underline{d}') + \min_2(\underline{d}') \geq k - f$, even if we lower the maximum values. Now since the only values we decrease are all d_l such that $d_l = \max(\underline{d})$, it may happen that we get that $\min_2(\underline{d}') = \min_2(\underline{d}) - 1$. Similarly as in Case 3.1, this is only the case if $\max(\underline{d}) = \min_2(\underline{d})$.

Now we summarize that: $\min_2(\underline{d}) = \max(\underline{d})$, $\min_2(\underline{d}') = \max(\underline{d}') = \min_2(\underline{d}) - 1 = \max(\underline{d}) - 1$, $\min(\underline{d}') = \min(\underline{d}) - 1$ and we have the system:

$$\begin{cases} \min(\underline{d}) + \min_2(\underline{d}) > k - f \\ \max(\underline{d}) - \min(\underline{d}) = k - f \end{cases} . \tag{4.14}$$

Now we want to show that $\min(\underline{d}') + \min_2(\underline{d}') \geq k - f$ or equivalently: that assuming that $\min(\underline{d}') + \min_2(\underline{d}') < k - f$ will lead to a contradiction.

Thus we write: $\min(\underline{d}') + \min_2(\underline{d}') = \min(\underline{d}) + \min_2(\underline{d}) - 2 = \min(\underline{d}) + \max(\underline{d}) - 2$. If we now make the assumption that $\min(\underline{d}') + \min_2(\underline{d}' < k - f$, we get that $\min(\underline{d}) + \max(\underline{d}) - 2 < k - f = \max(\underline{d}) - \min(\underline{d})$ which would imply that $\min(\underline{d}) + \max(\underline{d}) - 2 < \max(\underline{d}) - \min(\underline{d})$ and so $\min(\underline{d}) - 2 < -\min(\underline{d})$ thus again that $2\min(\underline{d}) < 2$, which is indeed a contradiction since we assumed that $\min(\underline{d}) = d_i \geq 2$.

Now if we do not have that $\max(\underline{d}) = \min_2(\underline{d})$, we will have that $\min_2(\underline{d}') = \min_2(\underline{d})$ so we find: $\min(\underline{d}') + \min_2(\underline{d}') = \min(\underline{d}) + \min_2(\underline{d}) - 1$ and since we assumed that $\min(\underline{d}) + \min_2(\underline{d}) > k - f$, we know that $\min(\underline{d}) + \min_2(\underline{d}) - 1 \geq k - f$, as required.

Case 3.3: $\min_2(\underline{d}) + \min(\underline{d}) = k - f$ and $\max(\underline{d}) - \min(\underline{d}) = k - f$
We now finally consider the case that both $\min_2(\underline{d}) + \min(\underline{d}) = k - f$ and $\max(\underline{d}) - \min(\underline{d}) = k - f$, so that we both have to increase the second lowest value $\min_2(\underline{d})$ and lower the maximum value $\max(\underline{d})$ of $\underline{d}$ to attain the equations from (4.12).

We thus create $\underline{d}'$ as follows: we let $d'_j = d_j$ for $j \in \{1, 2, ..., m\}, j \neq i$ and then we replace all d'_l to $d_l + 1$ for all $d_l = \min_2(\underline{d})$ and we replace all d'_p to $d_p - 1$ for all $d_p = \max(\underline{d})$. If we do this, then by similar arguments as in Case 3.1, we should attain (4.12). This is, however, assuming that we don't lower the maximum past the second minimum and vice versa. Since we increase the second minimum and decrease the maximum, we want that they have at least distance 2 between each other, so that in the worst case they become equal after increasing and decreasing them.

Since we have the following system:

$$\begin{cases} \min(\underline{d}) + \min_2(\underline{d}) = k - f \\ \max(\underline{d}) - \min(\underline{d}) = k - f \end{cases} . \tag{4.15}$$

From this we see that $\min(\underline{d}) + \min_2(\underline{d}) = k - f = \max(\underline{d}) - \min(\underline{d})$ so that $\min(\underline{d}) + \min_2(\underline{d}) = k - f = \max(\underline{d}) - \min(\underline{d})$, this yields that $\max(\underline{d}) - \min_2(\underline{d}) = 2\min(\underline{d})$ and since we assumed that $\min(\underline{d}) = d_i \geq 2$, we see that: $\max(\underline{d}) - \min_2(\underline{d}) \geq 4$.

This means that we have sufficient distance between $\max(\underline{d})$ and $\min_2(\underline{d})$ and we will thus indeed suffice (4.12), since there is no possibility of the maximum decreasing below the second minimum and vice versa. Furthermore, since we again only change elements by at most 1, we attain

$$||\underline{d} - \underline{d}'||_\infty = 1,$$

as required.

Thus we conclude that if $d_i \leq \min_2(\underline{d})$ and $d_i = \min(\underline{d})$, we take $\underline{d'}$ such that $d'_j = d_j$ for $j \in \{1, 2, ..., m\}, j \neq i$ and $d'_i = d_i - 1$ and we change all d'_l to: $d'_l = d_l + 1$ for all $d_l = \min_2(\underline{d})$ if $\min_2(\underline{d}) = k - f - \min(\underline{d})$ and we change all d'_p to: $d'_p = d_p - 1$ for all $d_p = \max(\underline{d})$ if $\max(\underline{d}) = k - f + \min(\underline{d})$.

Thus we have shown that for $d_i \in \underline{d}$ for an arbitrary vertext $v_{\underline{d}} \in V$, there exists a neighboring vertext $v_{\underline{d'}} \in V$ that has $d'_i = d_i - 1$. We can thus apply this procedure iteratively to obtain a path of length $d_i - 1$ from $v_{\underline{d}}$ to a vertex $v_{\underline{d^*}} \in V$ that has $d^*_i = 1$. This vertex is a neighbor of s_i, per definition, thus we can add the edge between $v_{\underline{d^*}}$ and s_i to obtain a path from $v_{\underline{d}}$ to s_i of length d_i, as we wanted to show.

No shorter path exists
Now we wish to show that there exists no shorter path from $v_{\underline{d}}$ to s_i than d_i. We consider two types of paths: a path that merely passes through non-sensor vertices $v_{\underline{d}}$ and a path that takes a 'shortcut' through a sensor s_j.

Case 1: a path through non-sensor vertices
Since all non-sensor vertices are only neighbors if and only if their distance is 1 in the infinity metric (and thus if their elements differ at most 1), we can see that a path that does not pass through any sensors from vertext $v_{\underline{d_1}}$ to $v_{\underline{d_2}}$, where d_{1_i} and d_{2_i} differ by some value r, any path between d_1 and d_2 has at least length r, since the path must travel through all values between d_{1_i} and d_{2_i} to be valid.

To connect $v_{\underline{d}}$ to s_i, we first need to connect $v_{\underline{d}}$ to some $v_{\underline{d^*}}$ that has $d^*_i = 1$, since these are the only vertices that are connected directly to s_i. Now since the difference between d_i and 1 is $d_i - 1$, any path from $v_{\underline{d}}$ to $v_{\underline{d^*}}$ has length at least $d_i - 1$ and thus any path from $v_{\underline{d}}$ to s_i must have length at least d_i.

Case 2: a path through a sensor
Suppose we want to connect $v_{\underline{d}}$ to s_i and we take a path through s_j and then only through non-sensor vertices. We know that the sensors have distance $k - f$ between them, so we merely need to show that the distance from $v_{\underline{d}}$ to s_j is at least $d_i - (k - f)$, so we need that $d_j \geq d_i - (k - f)$, or equialently: $d_i - d_j \leq k - f$. This is precisely the condition that $\max(\underline{d}) - \min(\underline{d}) \leq k - f$, since that condition implies that two elements have at least a difference of $k - f$.

Thus we see that a path from $v_{\underline{d}}$ to s_i trough s_j must have distance at least d_i. Should we try to take a path that goes trough multiple sensors, we can repeat this above argument to still obtain that the distance is at least d_i.

Conslusion
Thus we have shown that every vertex $v_{\underline{d}}$ has a path of length d_i to s_i and the shortest path possible is of length d_i, so that $d(v_{\underline{d}}, s_i) = d_i$, so that all vertices have the intended distance vector. Since all these vertices are created for different distance vectors, we know that all the distance vectors are unique of all these vertices and so we conclude that $G(k, m)$ has a threshold-k resolving set $\{s_1, s_2, ..., s_m\}$ of size m. □

4.2.1. Connection to different research

The paper [6] we reffered to in (amongst other sections) Section 4.1.1, considers a very similar problem. In this paper they give a construction for an extremal graph with a metric dimension m, though instead of a threshold-k metric dimension m, they consider the metric dimension m of a graph with diameter D.

The construction given in [6] is given in Figure 4.3.

This construction looks exactly the same as the construction given in Definition 7, we shall explain as to why this is.

Having the condition that the graph with a resolving set $\{s_1, s_2, ..., s_m\}$ has a diameter of D essentially means that the graph has a threshold-$D - 1$ resolving set $\{s_1, s_2, ..., s_m\}$, since the distance a vertex has to any given sensor is at most D and thus the sensors measure up to $D - 1$, plus an additional distance D, that corresponds with the ∞ distance in our problem.

Now since the graph $G(k, m)$ we constructed in Definition 7 actually has a diameter of $k + 1$ and is designed to be extremal in size, we can explain why Definition 7 is seemingly equivalent to the construction given in [6].

This means that if we manage to finalise our proof of extremality in the future, we can show that the extremal graph with metric dimension m and diameter D is actually also an extremal graph with threshold-

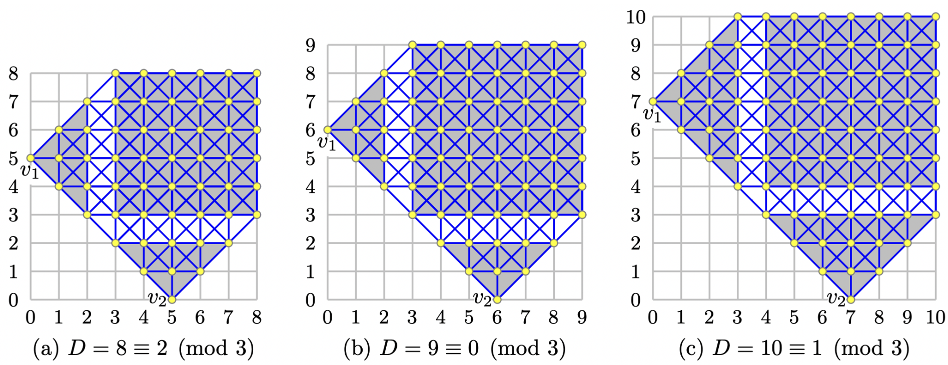


Figure 4.3: Extremal graphs with various diameters D and metric dimension 2, as given in [6].

$k+1$ metric dimension m.

We do note, that we found out about this paper in the last week of our research on extremal graphs with threshold-k metric dimension m, this is after we defined our construction in Definition 7.

5

Conclusion

In this thesis we considered a variation of the metric dimension problem, where each sensor vertex only measures up to radius k. The minimum amount of sensors needed to indentify each vertex uniquely by their measured distances to the sensors is called the threshold-k metric dimension of the graph.

We considered the question of extremality, that is: for a given range k and a number of sensors m, we aimed to find the maximal size of a graph that has threshold-k metric dimension m. For as far as we know, this is an open problem in general.

We managed to give two constructions for $k = 1$ and $k = 2$ that have $2^m + m - 1$ and $3^m + m - 1$ vertices, respectively. For $k = 3$ we gave a construction that had to consider the sensors measuring through each other, and constructed our graphs accordingly by including the condition that $\text{Min}_2\text{sum}(v) \geq 3$ for all vertices v. With this we gave a construction with $3^m + m \cdot 3^{m-1} + m - 1$ vertices. For these three cases, we also gave proofs of extremality. In the case of $k = 3$ we also proved why we need the condition that the lowest two coordinates of a distance vector should add up to a value greater than 2 to obtain an extremal graph.

Finally, we found a construction that constructs a graph with threshold-k metric dimension m for general k and m. We gave arguments that this construction is extremal, but due to time constraints, the proofs are incomplete. We did argue that patterns created by calculations on the computer and intuition seem to suggest that it is indeed optimal. We also compared these results to the construction of an extremal graph in a different paper [6] with a differently formulated problem, but with seemingly the same solution.

Future work that we would like to do on this subject would be to finalize the proof for general k and m. On a longer timescale it would be interesting to find extremal graphs for different variations, such as discussed in Section 1.2 for example. Based on applicability, fault tolereant versions would be attractive to consider.

Another optimizing problem could be to minimize the amount of edges needed for an extremal graph with threshold-k metric dimension m, or to construct extremal graphs G with a certain girth$(G) \geq g$. It would be interesting to show how the extremal size of such a graph might converge to the extremal size of a tree graph with threshold-k metric dimension, as described in [1], if we let $g \to \infty$.

A

Python code for creating a graph with threshold-3 metric dimension m

The following is python code for creating a graph very similar to $G_3(m)$ as defined in definition 3, though with some small modifications to lessen the amount of edges needed between the connecting vertices.

```
import networkx as nx
import matplotlib.pyplot as plt
import itertools as it
from pyvis.network import Network

m = 4 #number of sensors
G = nx.Graph() #initializing graph
slist = list(range(1,m+1))
lablist = []
color_map = []
for i in slist:
    lablist.append(str(i))
    color_map.append('gray')
for i in lablist: #adding the sensor vertices
    G.add_node(i,color='black')

for L in range(1, len(slist)+1):
    for A in it.combinations(slist, L): #creating all vertices vA
        G.add_node(str(A),color='blue')
        color_map.append('blue')
        for i in A: #creating and connecting the vertices vAi between vA and si
            v = str(A)+','+str(i)
            G.add_node(v,color='yellow')
            color_map.append('yellow')
            G.add_edge(v,str(i))
            G.add_edge(v,str(A))
        for LB in range(0,len(A)):
            for B in it.combinations(A,LB): #creating all vertices vAB and connecting them
                b = str(A)+','+str(B)
                G.add_node(b,color='red')
                color_map.append('red')
                G.add_edge(str(A),b)
                for i in B: #creating and connecting the vertices vABi between vAB and si
                    v = str(A)+','+str(B)+','+str(i)
                    G.add_node(v,color='yellow') #Bc
```

```
                    color_map.append('yellow')
                    G.add_edge(b,v)
                    G.add_edge(v,str(i))
                if len(B)>1 or len(A)<3: #connecting the vertices vABi amongst each other
                    for i in B:
                        v = str(A)+','+str(B)+','+str(i)
                        for j in A:
                            if not(j in B):
                                v2 = str(A)+',('+str(j)+',),'+str(j)
                                G.add_edge(v,v2)


print('afstanden meten')
coords = []
for i in G.nodes: #check the distances of al vertices to the sensors and check if each
distance is unique
    dis = []
    for j in slist:
        dis.append(0)
        d=nx.shortest_path_length(G,str(j),i)
        if d>3:
            d=4
        dis[j-1]=d
    print(dis)
    if dis in coords:
        print('dubbel')
    coords.append(dis)

print('plot maken') #visualization
nx.draw(G,node_color=color_map, with_labels=True)
plt.show()
nt = Network('700px', '1400px')
nt.from_nx(G)
nt.show('constructie2_'+str(m)+'.html')
```

B

Python code for creating a graph with threshold-k metric dimension m

The following is python code for creating a graph very similar to $G(k,m)$ as defined in definition 7, though with some small modifications to lessen the amount of edges needed between the vertices.

```
import networkx as nx
import matplotlib.pyplot as plt
from pyvis.network import Network

def min2(cor,k): #gives the sum of the lowest 2 elements of a vector
    min1 = k+1
    min2 = k+1
    for i in cor:
        if i < min1:
            min2 = min1
            min1 = i
        elif i < min2:
            min2 = i
    return min1,min2


k = 5 #range of the sensors
m = 3 #amount of sensors
f = 0 #optimal amount of overflow to be determined
for i in range(0,k):
    if 3*i >= k-1:
        print('beste f is '+str(i-1))
        f = i-1
        break

G = nx.Graph() #initialize graph

for i in range(1,m+1): #adding in the sensors
    G.add_node(str(i),color='black')


v = m*[1]
end = m*[k+1]

for t in range(0,(k+1)**m+2): #iterates over all possible distance vectors
```

```
    new = v.copy() #the next vector to be created
    if new == end: #stop if at last vector, this one wont be added in the graph
        break
    j = 0
    for j in range(0,m):
        if new[j]<k+1:
            new[j]+=1
            for r in range(0,j):
                new[r]=1
            break
    min2s = min2(v,k)
    mini = min(v)
    maxi = max(v)
    if sum(min2s) >= k-f and (maxi+f <= mini+k): #check if v is an allowed
    distance vector
        G.add_node(str(v))
        for i in range(0,m):
            if v[i] == 1: #connect it to sensor si if 1 at coordinate i
                G.add_edge(str(i+1),str(v))
            else: #connect it to the vertex next to it
                v2 = v.copy()
                v2[i] -= 1
                minc = 0
                maxc = 0
                if sum(min2(v2,k)) >= k-f: #check if it suffices both conditions
                    minc = 1
                if max(v2)+f <= min(v2)+k:
                    maxc = 1
                if minc + maxc == 2: #if both conditions are satisfied it will be
                added and connected
                    G.add_node(str(v2))
                    G.add_edge(str(v),str(v2))
                else:
                    if minc == 0: #fixing the min2sum condition
                        v2[i]+=1
                        min1 = v2[i]
                        for j in range(0,m):
                            if j!=i:
                                if min1 + v2[j] == k-f:
                                    v2[j] += 1
                        v2[i]-=1
                        if max(v2)+f <= min(v2)+k: #connect it if it suffices the
                        max-min condition
                            G.add_node(str(v2))
                            G.add_edge(str(v),str(v2))
                    if maxc == 0: #fixing the max-min condition
                        maxel = max(v2)
                        for j in range(0,m):
                            if v2[j] == maxel:
                                v2[j] -= 1
                        G.add_node(str(v2))
                        G.add_edge(str(v),str(v2))
    v = new


coords = [] #checking if the vertices have correct distance vectors
```

```
print('afstanden testen...')
if True:
    for i in G.nodes:
        dis = []
        for j in range(1,m+1):
            dis.append(0)
            d=nx.shortest_path_length(G,str(j),i)
            if d>k:
                d=k+1
            dis[j-1]=d
        #print(str(i)+' heeft vector '+str(dis))
        if str(i) != str(dis):
            print('fout: node '+str(i)+' heeft vector '+str(dis))
        if dis in coords:
            print('dubbel')
        coords.append(dis)
print(str(len(G.nodes))+' nodes')

if True: #plotting the graph
    if m < 3 and False:
        nx.draw_planar(G, with_labels = True)
    else:
        nx.draw(G, with_labels=True)
    plt.show()
```

Bibliography

[1] Zsolt Bartha, Júlia Komjáthy, and Järvi Raes. Sharp bound on the threshold metric dimension of trees. *arXiv preprint arXiv:2111.08813*, 2021.

[2] José Cáceres, Carmen Hernando, Mercè Mora, Ignacio M Pelayo, and María Luz Puertas. Locating–dominating codes: Bounds and extremal cardinalities. *Applied Mathematics and Computation*, 220: 38–45, 2013.

[3] Irene Charon, Olivier Hudry, and Antoine Lobstein. Minimizing the size of an identifying or locating-dominating code in a graph is np-hard. *Theoretical Computer Science*, 290(3):2109–2120, 2003.

[4] Paul Erdös, Frank Harary, and William T Tutte. On the dimension of a graph. *Mathematika*, 12(2):118–122, 1965.

[5] F Harary and RA Melter. On the metric dimension of a graph, combinatoria, vol. 2. 1976.

[6] Carmen Hernando, Merce Mora, Ignacio M Pelayo, Carlos Seara, and David R Wood. Extremal graph theory for metric dimension and diameter. *The Electronic Journal of Combinatorics [electronic only]*, 17 (1):Research–Paper, 2010.

[7] Mark G Karpovsky, Krishnendu Chakrabarty, and Lev B Levitin. On a new class of codes for identifying vertices in graphs. *IEEE transactions on information theory*, 44(2):599–611, 1998.

[8] Gergely Ódor, Jana Vuckovic, Miguel-Angel Sanchez Ndoye, and Patrick Thiran. Source detection via contact tracing in the presence of asymptomatic patients. *arXiv preprint arXiv:2112.14530*, 2021.

[9] Peter J Slater. Leaves of trees. *Congr. Numer*, 14(549-559):37, 1975.

[10] Peter J Slater. Domination and location in acyclic graphs. *Networks*, 17(1):55–64, 1987.

[11] Peter J Slater. Fault-tolerant locating-dominating sets. *Discrete Mathematics*, 249(1-3):179–189, 2002.

[12] Richard C Tillquist, Rafael M Frongillo, and Manuel E Lladser. Getting the lay of the land in discrete space: A survey of metric dimension and its applications. *arXiv preprint arXiv:2104.07201*, 2021.

[13] Richard C Tillquist, Rafael M Frongillo, and Manuel E Lladser. Truncated metric dimension for finite graphs. *arXiv preprint arXiv:2106.14314*, 2021.