Graph Realization of Distance Sets

Amotz Bar-Noy
City University of New York (CUNY), USA. Email: amotz@sci.brooklyn.cuny.edu

David Peleg
Weizmann Institute of Science, Rehovot, Israel. Email: david.peleg@weizmann.ac.il

Mor Perry
The Academic College of Tel-Aviv-Yaffo, Israel. Email: morpy@mta.ac.il

Dror Rawitz
Bar Ilan University, Ramat-Gan, Israel. Email: dror.rawitz@biu.ac.il

Abstract

The Distance Realization problem is defined as follows. Given an $n \times n$ matrix D of nonnegative integers, interpreted as inter-vertex distances, find an n-vertex weighted or unweighted graph G realizing D, i.e., whose inter-vertex distances satisfy $dist_G(i,j) = D_{i,j}$ for every $1 \leq i < j \leq n$, or decide that no such realizing graph exists. The problem was studied for general weighted and unweighted graphs, as well as for cases where the realizing graph is restricted to a specific family of graphs (e.g., trees or bipartite graphs). An extension of Distance Realization that was studied in the past is where each entry in the matrix D may contain a *range* of consecutive permissible values. We refer to this extension as Range Distance Realization (or Range-DR). Restricting each range to at most k values yields the problem k-Range Distance Realization (or k-Range-DR). The current paper introduces a new extension of Distance Realization, in which each entry $D_{i,j}$ of the matrix may contain an arbitrary set of acceptable values for the distance between i and j. We refer to this extension as Set Distance Realization (Set-DR), and to the restricted problem where each entry may contain at most k values as k-Set Distance Realization (or k-Set-DR).

We first show that 2-Range-DR is NP-hard for unweighted graphs (implying the same for 2-Set-DR). Next we prove that 2-Set-DR is NP-hard for unweighted and weighted trees. We then explore Set-DR where the realization is restricted to the families of stars, paths, or cycles. For the weighted case, our positive results are that for each of these families there exists a polynomial time algorithm for 2-Set-DR. On the hardness side, we prove that 6-Set-DR is NP-hard for stars and 5-Set-DR is NP-hard for paths and cycles. For the unweighted case, our results are the same, except for the case of unweighted stars, for which k-Set-DR is polynomially solvable for any k.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases Graph Realization, distance realization, network design

Funding Supported in part by a US-Israel BSF grant (2018043).

1 Introduction

Background. *Network realization* problems are fundamental graph-algorithmic questions in which one is asked to construct a network conforming to some predefined requirements. Given a *specification* (or *information profile*) that consists of constraints on some network parameters, such as the vertex degrees, distances, or connectivity, one is required to construct a network conforming to the given specification, i.e., satisfying the requirements, or to determine that no such network exists. The motivation for network realization problems stems from both "exploratory" contexts where one attempts to reconstruct an *existing* network of unknown structure based on the outcomes of experimental measurements, and engineering contexts related to network design.

In the Distance Realization problem, the given profile is an $n \times n$ matrix D such that each entry $D_{i,j} \in \mathbb{N} \cup \{\infty\}$, for $1 \leq i < j \leq n$, and $D_{i,i} = \{0\}$, for every $1 \leq i \leq n$. We view $D_{i,j}$ as specifying the required distance between the vertices i and j in the network. A graph $G = (V, E)$ is a *realization* of D if $dist_G(i, j) = D_{i,j}$, for every $1 \leq i < j \leq n$, where $dist_G(i, j)$ denotes the distance between i and j in G. Generally, we may be interested in two types of realizing graphs. In unweighted Distance Realization it is assumed that each edge of the realizing graph is of length 1. In weighted Distance Realization the edges of the realizing graph may have arbitrary integral lengths, where for normalization purposes it is assumed that the minimum edge length is 1.

Observe that an unweighted realizing graph is fully determined by D: the edge (i, j) exists in the graph if and only if $D_{i,j} = 1$. It follows that there is only one graph G_D that may serve as a candidate realizing graph. This was observed by Hakimi and Yau [12], who provided a characterization for distance realization by unweighted graphs, implying also a polynomial-time algorithm for unweighted Distance Realization. Notice also that in the case where the realization is required to be a specific graph H, one can solve unweighted Distance Realization by deciding whether H and G_D are isomorphic. This Graph Isomorphism problem is computationally easy when H belongs to certain graph types, such as stars, paths, and cycles, and therefore the problem of distance realization by such graphs can be solved in polynomial time.

Hakimi and Yau [12] also studied weighted Distance Realization. They proved that the necessary and sufficient condition for the realizability of a given martix D is that D is a metric. Furthermore, they gave a polynomial-time algorithm that computes a realization for any given metric distance matrix. More specifically, their algorithm constructs a minimum-edge realizing graph whose edges are necessary in every realization of D.

Patrinos and Hakimi [14] considered the case where weights can be negative. They showed that any symmetric matrix (with zero diagonal) is a distance matrix of some graph G. They gave necessary and sufficient conditions for realizing such a matrix by a tree, and they showed that if a tree realization exists it is unique. Distance Realization in *weighted trees* was considered in [2], which presented a characterization for realizability. For unweighted trees, there is a straightforward realization algorithm, based on the algorithm of [12] for general unweighted graphs, and on the fact that the realization, if exists, is unique. Distance realization restricted to bipartite graphs was studied in [4], where it was observed that it is sufficient to check the unique realization in the unweighted case or the minimal realization in the weighted case.

A natural extension of Distance Realization is when each entry in the distance matrix may contain a *range* of consecutive values instead of a single value. Range specifications may arise, for example, when D reflects the properties of an unknown network, and its values are obtained by imprecise measurements, or alternatively, when D represents a design specification for a planned network in a setting where distance constraints are not rigid and allow some flexibility. Formally, we are given two values $D^-_{i,j}$ and $D^+_{i,j}$ for every i, j and the realizing G must satisfy $D^-_{i,j} \leq dist_G(i, j) \leq D^+_{i,j}$. We refer to this extended version of the problem as Range Distance Realization (or Range-DR).

Tamura et al. [18] obtained necessary and sufficient conditions for the realizability of a range distance matrix by weighted graphs, generalizing the result of [12] from precise to range specifications. A polynomial-time algorithm for weighted Range-DR was given in [15]. The unweighted version of Range-DR was shown to be NP-hard in [4], where it was also shown that if the realizing graph is required to be a tree, then both the unweighted and weighted versions of Range-DR are NP-hard.

Realization with Distance Sets. In this paper we introduce a novel extension of Range Distance Realization, called Set Distance Realization (Set-DR). Instead of a *range*, we assume that each entry $D_{i,j}$ in the distance matrix specifies a *set* of acceptable values for the distance between i and j. More formally, consider an $n \times n$ matrix D, such that each entry $D_{i,j} \subseteq \mathbb{N} \cup \{\infty\}$, for $1 \leq i < j \leq n$, is a non-empty set, and $D_{i,i} = \{0\}$, for every $1 \leq i \leq n$. We view $D_{i,j}$ as specifying a list of acceptable values of the distance between i and j, where i and j are vertices in some network. A graph $G = (V, E)$ is a *realization* of the D if $dist_G(i, j) \in D_{i,j}$, for every $1 \leq i < j \leq n$.

One of the main questions studied in this paper involves the effect of limiting the number of values in each entry of the matrix D. This question is equally interesting for Set-DR and Range-DR. Given an integer k, we say that the matrix D is a *k-set* distance matrix if $|D_{i,j}| \leq k$ for every $1 \leq i < j \leq n$. A distance matrix D is a *k-range* distance matrix, if $D_{i,j}$ is a range that contains at most k consecutive values for every $1 \leq i < j \leq n$. A 1-set distance profile is called *precise*. Restricting the Set Distance Realization problem to k-set distance matrices yields the problem k-Set Distance Realization (or k-Set-DR). Similarly, restricting the Range Distance Realization problem to k-range distance matrices yields the problem k-Range Distance Realization (or k-Range-DR).

Henceforth, we assume that entry $D_{i,j}$ in a 2-set distance matrix D consists of two integers, $D_{i,j} = \left\{d^0_{i,j}, d^1_{i,j}\right\}$, which need not be distinct.

Our Results. In this paper we study the computational complexity of k-Set-DR and k-Range-DR, as a function of k, in various graph families.

Inspecting the proof given in [4] for the hardness results for Range-DR by trees and unweighted graphs reveals that 3-Range-DR is already NP-hard over these graph families, implying that 3-Set-DR is NP-hard as well. We modify the reductions from [4] to show that already 2-Range-DR is NP-hard for general unweighted graphs, where precise realization is known to be polynomial [12]. For general weighted graphs, it is known that Range-DR is computationally easy [15]. We note that the algorithm from [15] does not work for Set-DR, since it relies on the continuity of the given ranges. In fact, Set-DR for general weighted graphs remains an open problem. We show that both unweighted 2-Set-DR and weighted 2-Set-DR are NP-hard for trees. Thus, we obtain a dichotomy between 2-set distance realization and precise realization for trees, since precise realization is known to be solvable in polynomial time [12, 2].

Next, we show that 2-Set-DR is polynomial time solvable for stars, paths, and cycles. Our realization algorithms are based on a reduction to the 2-SAT problem (satisfiability of a 2-CNF formula), which can be solved in linear time [10]. The idea is to use one vertex i_0 as a point of reference for all other vertices. Thus, a Boolean variable b_j is associated with each vertex j and determines which of the two values of $D_{i_0,j}$ should be used. The 2-CNF formula is constructed according to the rest of the entries of D. Applying this approach for stars is rather straightforward, but it becomes more complicated for paths, and especially for cycles. In addition, we prove that there exists a polynomial time algorithm for k-Set-DR on unweighted stars, for any k. For weighted stars, we present a polynomial time realization algorithm for Range-DR when the range values are polynomially bounded. This algorithm is based on a reduction to the feasibility problem of linear integer programs with at most two variables per constraint, which can be solved efficiently [5].

On the hardness side, we show that Set-DR is NP-hard for weighted stars already with 6-set distance profiles. We obtain slightly tighter results for paths and cycles, for which both unweighted Set-DR and weighted Set-DR are already NP-hard already with 5-set

Graph family	Range-DR	Set-DR
General	2-Range-DR is NP-hard (Thm 1)	2-Set-DR is NP-hard (Thm 1)
	1-Range-DR is polynomial [12]	1-Range-DR is polynomial [12]
Tree	3-Range-DR is NP-hard [4]	2-Set-DR is NP-hard (Thm 2)
	1-Range-DR is polynomial [12]	1-Range-DR is polynomial [12]
Star	Range-DR is polynomial (Thm 4)	Set-DR is polynomial (Thm 4)
Path	2-Range-DR is polynomial (Thm 8)	2-Set-DR is polynomial (Thm 8)
	Range-DR is NP-hard (Thm 9)	5-Set-DR is NP-hard (Thm 10)
Cycle	2-Range-DR is polynomial (Thm 12)	2-Set-DR is polynomial (Thm 12)
	Range-DR is NP-hard (Thm 14)	5-Set-DR is NP-hard (Thm 15)

Table 1 Results for realization with unweighted graphs.

Graph family	Range-DR	Set-DR
General	Range-DR is polynomial [15]	Open problem
Tree	3-Range-DR is NP-hard [4]	2-Set-DR is NP-hard (Thm 2)
	1-Range-DR is polynomial [2]	1-Range-DR is polynomial [2]
Star	Range-DR is polynomial[1] (Thm 6)	2-Set-DR is polynomial (Thm 3)
		6-Set-DR is NP-hard (Thm 5)
Path	2-Range-DR is polynomial (Thm 7)	2-Set-DR is polynomial (Thm 7)
	Range-DR is NP-hard (Thm 9)	5-Set-DR is NP-hard (Thm 10)
Cycle	2-Range-DR is polynomial (Thm 13)	2-Set-DR is polynomial (Thm 13)
	Range-DR is NP-hard (Thm 14)	5-Set-DR is NP-hard (Thm 15)

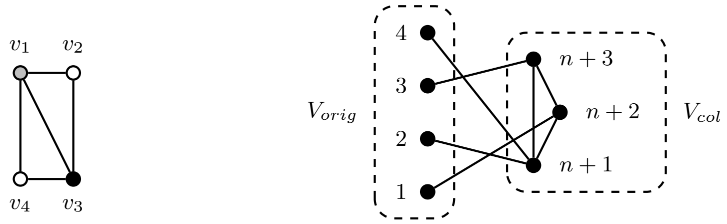
Table 2 Results for realization with weighted graphs.

distance profiles. Our hardness results are based on reductions from the 3-colorability problem. However, the reductions are not similar. Specifically, in the case of weighted stars, the possible colors of a vertex are encoded in the distance matrix by possible edge weights, while in the case of weighted and unweighted paths, the colors are encoded by vertices and their location on the path. The hardness result for 5-Set-DR on the cycle is obtained by a reduction from 5-Set-DR on the path.

Tables 1 and 2 summarize our results.

Related Work. An optimization variant of Distance Realization problem was introduced in [12]. In this problem, a distance matrix D is given over a set S of n vertices, and the goal is to find a graph G including S, with possibly *auxiliary* vertices, that realizes the given distance matrix for S. Necessary and sufficient conditions are given for a matrix to be realizable by a weighted or an unweighted graph. It is shown in [9] that an optimal realization can have at most n^4 vertices, and therefore, there is a finite (but exponential) time algorithm to find an optimal realization. In [1] it is shown that finding optimal realizations of distance matrices with integral entries is NP-complete, and evidence to the difficulties in approximating the optimal realization is provided in [6]. Over the years, various heuristics for optimal realizations were considered [13, 16, 17, 19]. Since optimal realization seems hard even to approximate, special cases and other variations have been studied [6, 11].

[1] This result requires that the entries of D are polynomially bounded



(a) A 3-colorable graph. **(b)** A realization of D.

Figure 1 An example of the reduction in the proofs of Theorem 1 for $n = 4$. White, gray, and black correspond to nodes $n+1$, $n+2$, and $n+3$, respectively.

Special attention has been given to the optimal distance realization problem where the realizing graph is a tree. In [12], a procedure is given for finding a tree realization of D if exists. It is also shown therein that a tree realization, if exists, is unique and is the optimum realization of D. Necessary and sufficient conditions for a distance matrix to be realizable by a tree were given in several papers [3, 8, 17]. Finally, an $O(n^2)$ time algorithm for optimal tree-realization is described in [7].

2 Realizations by Trees and Unweighted Graphs

In this section we consider unweighted realizations in general graphs and both unweighted and weighted realizations in trees.

For general graphs, recall that the range realization problem in weighted graphs and the precise realization problem in unweighted graphs both have a polynomial time algorithm. We provide an NP-hardness result for Range-DR by unweighted graphs, even for 2-range distance profiles.

▶ **Theorem 1.** *2-Range-DR is NP-hard in unweighted graphs.*

Proof. We prove the lemma using a reduction from the 3-coloring problem.

Consider an instance G of the 3-coloring problem. We construct a 2-range distance matrix D for $n+3$ vertices, i.e., for $\{1,\ldots,n+3\}$. Intuitively, we think of the first n vertices, $V_{orig} = \{1,\ldots,n\}$, as representing the *original* vertices of the given graph G, and of additional 3 vertices of D, $V_{col} = \{n+1, n+2, n+3\}$, as representing the three *colors*. Let

$$D_{i,j} = \begin{cases} \{1\} & i = n+1,\ldots,n+3, \quad j = n+1,\ldots,n+3, \\ \{1,2\} & i = 1,\ldots,n, \quad j = n+1,\ldots,n+3, \\ \{2,3\} & 1 \le i < j \le n, \quad (v_i,v_j) \notin E(G), \\ \{3\} & 1 \le i < j \le n, \quad (v_i,v_j) \in E(G). \end{cases}$$

We now prove that the input G is 3-colorable if and only if D is realizable by an unweighted graph. (See Figure 1 for an illustration.)

Suppose G is 3-colorable. Let $\chi : V(G) \mapsto \{1,\ldots,3\}$ be the coloring function. For the matrix D defined from G, construct a realizing graph $\mathcal{G}$ as follows. Start with a triangle containing the color vertices $n+1, n+2, n+3$. Connect each original vertex i to the color vertex $n+\chi(i)$. It is easy to verify that $\mathcal{G}$ realizes D (see Figure 1b for an example).

Suppose there exists an unweighted graph $\mathcal{G}$ which realizes D. Consider two original vertices i and j. Since $1 \notin D_{i,j}$, it follows that i and j are not connected by an edge. Therefore, every original vertex i must be connected to at least one of the color vertices.

Define a coloring function for G as follows. For every original vertex i, let $n+c$ be some color vertex connected to i, and let $\chi(v_i) = c$. Since $1, 2 \notin D_{i,j}$, if two vertices v_i and v_j are connected by an edge in G, then their distance in $\mathcal{G}$ must be at least 3. This ensures that none of the color vertices are connected to both i and j (as this would make their distance 2). It follows that if $(v_i, v_j) \in E(G)$, then i and j are assigned different colors. ◀

Next, we show that distance realization in unweighted or weighted trees is hard even for 2-set distance profiles. The reduction we use is almost identical to a reduction from [4] that (implicitly) shows hardness for 3-range distance profiles. It is given in Appendix A for completeness.

▶ **Theorem 2.** *2-*Set*-*DR *is NP-hard for both unweighted trees and weighted trees.*

3 Star Realizations

In this section we study Set-DR in stars. We show that the exists a polynomial time algorithm that solves weighted 2-Set-DR in stars. On the other hand, we show that 5-Set-DR on weighted stars is NP-hard. Furthermore, we present an polynomial time algorithm that solves k-Range-DR on weighted stars, for any k, provided that matrix entries are polynomially bounded.

To put these results in context, it may be useful to observe that the *unweighted* case in stars is easier: unweighted k-Set-DR in stars can be solved in polynomial time for any k.

3.1 2-Set-DR on Stars is Easy

We show that the 2-Set-DR problem in stars can be solved efficiently.

▶ **Theorem 3.** *There exists a polynomial time algorithm for 2-*Set*-*DR *on stars.*

Proof. Assume that i is the center of the star. It follows that the weight of any edge (i, j), for $j \neq i$ can be either $d^0_{i,j}$ or $d^1_{i,j}$. Define a Boolean variable x_j, where $x_j = \text{FALSE}$, if the weight of the edge (i, j) is $d^0_{i,j}$, and $x_j = \text{TRUE}$, if the weight of the edge (i, j) is $d^1_{i,j}$. The rest of the entries of D are used to create a 2-CNF formula that is satisfiable if and only if there exists a star realization of D in which i is the center.

Consider two vertices $j, k \neq i$. Since there are two possible weights for the edges (i, j) and (i, k), it follows that there are four possible distances from j to k: (i) $d^0_{i,j} + d^0_{i,k}$, (ii) $d^0_{i,j} + d^1_{i,k}$, (iii) $d^1_{i,j} + d^0_{i,k}$, and (iv) $d^1_{i,j} + d^1_{i,k}$. For each one of the above options, check whether it equals $d^0_{j,k}$ or $d^1_{j,k}$. This induces a truth table on the variable x_j and x_k that can be represented by at most two 2-CNF clauses.[2] Doing this for all pairs of vertices creates a 2-CNF formula that contains at most $O(n^2)$ clauses by concatenating all the above mentioned clauses.

Suppose that there exists a star realization P of D with i as a center. This induces an assignment to the variables Boolean variables that satisfies the 2-CNF formula. On the other hand, assume that the 2-CNF formula that is obtained by assuming that i is the center is satisfiable. A satisfying assignment induces a star, which complies with the profile.

Since there are n candidates for the center vertex, we need to run the above process n times. It follows that the total running time of the algorithm is $O(n^3)$. ◀

[2] For example, let $D_{i,j} = \{2, 3\}$, $D_{i,k} = \{3, 4\}$, and $D_{j,k} = \{5, 7\}$. There are two possible weight assignments: either $w(i, j) = d^0_{i,j}$ and $w(i, k) = d^0_{i,k}$ or $w(i, j) = d^1_{i,j}$ and $w(i, k) = d^1_{i,k}$ This can be represented by the clause $(\neg x_j \vee \neg x_k) \wedge (x_j \vee x_k)$.

We remark that the running time for unweighted case can be improved to $O(n^2)$.

▶ **Theorem 4.** *There exists a polynomial time algorithm for* k-SET-DR *on unweighted stars, for any* k.

Proof. Since all distances in an unweighted star are either 1 or 2, one may assume that $D_{i,j} \subseteq \{1,2\}$, for every $i \neq j$. The theorem follows due to Theorem 3. ◀

3.2 6-Set-DR on Weighted Stars is Hard

We show that the star realization problem is NP-hard even on 6-set distance profiles.

▶ **Theorem 5.** *6*-SET-DR *is NP-hard in weighted stars.*

Proof. We prove the lemma using a reduction from the 3-COLORING problem.

Consider a graph G, where $V(G) = \{v_1, \ldots, v_n\}$. We construct a distance matrix D on $n+3$ vertices, denoted by $\{u_1, \ldots, u_{n+3}\}$. Informally, the distance matrix is defined to force u_{n+1} to be the center of the star while u_{n+2} and u_{n+3} must be two of the leaves whose distance from the center is 1. The rest of the vertices, $u_1, \ldots, u_n$ that are associated with the n vertices $v_1, \ldots, v_n$ of G, are leaves whose distance from the center is either 1, 2, or 4 with the idea is that distance 2^c is associated with the color c (0, 1, or 2). Finally, the distances between these n leaves is defined to guarantee that the two endpoints of any edge of G are associated with different colors, if a realization exists. See More formally, the 6-set distance matrix D is defined as follows for any two indices $0 \leq k < \ell \leq n+3$:

$$D_{k,\ell} = \begin{cases} \{3,5,6\} & k,\ell \leq n, (v_k, v_\ell) \in E(G), \\ \{2,3,4,5,6,8\} & k,\ell \leq n, (v_k, v_\ell) \notin E(G), \\ \{1,2,4\} & k \leq n, \ell = n+1, \\ \{2,3,5\} & k \leq n, \ell = n+2, n+3, \\ \{1\} & k = n+1, \ell = n+2, n+3, \\ \{2\} & k = n+2, \ell = n+3 \ . \end{cases}$$

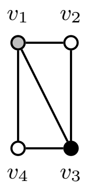
We show that G is 3-colorable if and only if D is realizable by weighted star.

Assume that G is 3-colorable and $\chi : V \mapsto \{0,1,2\}$ is a 3-coloring of G. We describe a star realization S. First, let u_{n+1} be the center of the star. Vertices u_{n+2} and u_{n+3} are leaves such that $w(u_{n+1}, w_{n+2}) = w(u_{n+1}, w_{n+2}) = 1$. Next, for every $i \in \{1, \ldots, n\}$, if $\chi(v_i) = c$, then $w(u_{n+1}, u_i) = 2^{\chi(v_i)}$. It is straightforward to verify that S realizes D. In particular, we observe that the first requirement of D is satisfied, since χ is a 3-coloring.

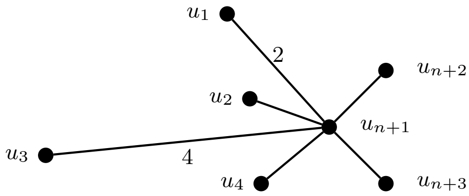
For the other direction, suppose that S is a star realization of D. First, notice that the above distance matrix makes sure that $u_{n+2}, u_{n+1}, u_{n+3}$ form a path with two edges of weight 1. Hence, u_{n+1} must be the center of the star. We define a coloring χ of $V(G)$ according to the weights of the edges to the center: $\chi(v_i) = \log_2 w(u_{n+1}, u_i)$. χ is a proper 3-coloring, since the first requirement of D ensures that $w(u_{n+1}, u_i) \neq w(u_{n+1}, u_j)$ if $(v_i, v_j) \in E$. This is because $2 = 1+1$, $4 = 2+2$, and $8 = 4+4$ are not members of $\{3,5,6\}$ which are the possible distances between u_i and u_j. (See Figure 2.) ◀

3.3 Range-DR on Weighted Stars

In [4] it was shown that there exists a polynomial time algorithm that solves the RANGE-DR problem on a given fixed weighted tree, assuming that non-integral edges weights are allowed.



(a) A 3-colorable graph.



(b) A realization of D.

Figure 2 An example of the reduction in the proof of Theorem 5 for $n = 4$. White, gray, and black correspond to the weights of the edges of the star. Edges without a label are of weight 1.

Let D be a range distance matrix, where $D_{ij} = \{D^-_{ij}, \ldots, D^+_{ij}\}$. Also, let $T = (V, E_T)$ be a known tree, and let $P_{k,\ell} = (k = i_0, i_1, \ldots, i_{t_{k,\ell}} = \ell)$ be the unique path from vertex k to vertex ℓ in T. For each edge $(i, j) \in E_T$, let $w_{i,j}$ be the variable that denotes the weight of this edge. Consider the following linear program:

$$\sum_{j=0}^{j=t_{k,\ell}-1} w_{i_j,i_{j+1}} \geq D^-_{k,\ell} \qquad \sum_{j=0}^{j=t_{k,\ell}-1} w_{i_j,i_{j+1}} \leq D^+_{k,\ell} \qquad \forall P_{k,\ell}$$

The algorithm from [4] finds a realization by obtaining a feasible solution to the above LP.

As mentioned above this approach may obtain a realization with non-integral edge weights and distances. Moreover, it may be the case that there exists a realization with non-integral edge weights, while a realization with with integral edge lengths does not exist. For example, consider the following distance matrix:

$$D = \begin{pmatrix} \{0\} & \{3\} & \{3\} & \{1,2\} \\ \{3\} & \{0\} & \{3\} & \{1,2\} \\ \{3\} & \{3\} & \{0\} & \{1,2\} \\ \{1,2\} & \{1,2\} & \{1,2\} & \{0\} \end{pmatrix}$$

D admits no star realization with integral weights, but if one allows edge lengths and distances of 1.5, then a realization exists.

We show that the realization problem with integral edge weights is solvable on stars, assuming that the entries of D are polynomially bounded.

▶ **Theorem 6.** *Assume that the entries of D are polynomially bounded. Then there exists a polynomial time algorithm for* Range-DR *on weighted stars.*

Proof. Fix the center of the star, and consider the above LP. Since all paths are of length one or two, the resulting integer linear program contains at most two variables per inequality. Since we requires integral edges, we add the following integrality constraints: $w_i \in \mathbb{N}$, for every i. The feasibility problem of integer programs with at most two variables per constraint is solvable in $O(mU)$, where m is the number of constraints and U is the range of the variables [5]. In this case $U = \max_{i,j} \max(D_{ij})$. ◀

Notice that the above mentioned NP-hardness for 6-Set-DR applies to profiles in which each entry is composed of at most 6 constants, which means that it is polynomially bounded.

4 Path Realizations

In this section we study the realization of distance profiles by paths. We first show that if each entry of the distance matrix consists of at most two different values than a realization by a path (if exists) can be found by a polynomial time algorithm. On the other hand, we show that Set-DR on paths is NP-hard, even on 5-set distance profiles. Both results hold both for weighted and unweighted paths.

4.1 Algorithm for 2-Set-DR on Paths

A path P realizes the 2-set distance profile D if either $dist_P(i,j) = d^1_{i,j}$ or $dist_P(i,j) = d^2_{i,j}$ holds for every i and j.

We show that 2-Set-DR on weighted paths can be solved by a reduction to 2-CNF satisfiability, which can be solved in $O(m)$ time [10], where m is the number of clauses.

▶ **Theorem 7.** *There exists a polynomial time algorithm for 2-*Set-DR *on weighted paths.*

Proof. Assume that the weighted path is embedded on the real line where the vertices are located at integers and the distance between any two vertices is their distance on the line. Furthermore, assume that the leftmost vertex is vertex i and it is located at 0.

Any realization of D in which i is the left-most node, implies that vertex $j \neq i$ is located either at $d^0_{i,j}$ or at $d^1_{i,j}$. Define a Boolean variable x_j, where $x_j = \text{FALSE}$ represents that j is located at $d^0_{i,j}$ and $x_j = \text{TRUE}$ represents that j is located at $d^1_{i,j}$. The rest of the entries of D will create a 2-CNF formula that is satisfiable if and only if there exists a realization of D on the path in which i is the left-most vertex.

Consider the possible placements of two vertices $j, k \neq i$. Since each one of them has two possible placements, it follows that there are four possible placements of j and k: (i) j at $d^0_{i,j}$ and k at $d^0_{i,k}$, (ii) j at $d^0_{i,j}$ and k at $d^1_{i,k}$, (iii) j at $d^1_{i,j}$ and k at $d^0_{i,k}$, and (iv) j at $d^1_{i,j}$ and k at $d^1_{i,k}$. For each one of the above four options, check whether it complies with $d^0_{j,k}$ or with $d^1_{j,k}$. This induces a truth table on the variable x_j and x_k that can be represented by at most two 2-CNF clauses. Doing this for all pairs of vertices creates a 2-CNF formula that contains at most $O(n^2)$ clauses by concatenating all the above mentioned clauses.

If there exists a path realization P of the profile, then embed it on the real line such that the left-most vertex i of P is located at 0. Then, consider the formula that we get by assuming that i is the left-most vertex of P, and assign values to the Boolean variables according to the distances from i. Since P realizes the profile, the 2-CNF formula must be satisfied. On the other hand, assume that the 2-CNF formula that is obtained by assuming that i is placed on 0 is satisfiable. A satisfying assignment induces a placement of the vertices on the real line, and this implies a realization of the profile.

Since there are n candidates for the left-most vertex we need to run the above process n times. It follows that the total running time of the algorithm is $O(n^3)$. ◀

The same proof works for the unweighted case. One only need to notice that in this case all the distances are integers in the range $\{1, \ldots, n-1\}$ and therefore the placement of the vertices is a bijection from $\{1, \ldots, n\}$ to $\{0, \ldots, n-1\}$.

▶ **Theorem 8.** *There exists a polynomial time algorithm for 2-*Set-DR *on unweighted paths.*

4.2 Hardness Result

Range-DR in weighted paths was shown to be NP-hard in [4] using a reduction from the Linear Arrangement problem. This result also applies to unweighted paths. It is important to note that the reduction constructs a matrix with unlimited ranges.

We start the section with an alternative and simpler proof that also requires unlimited ranges.

▶ **Theorem 9.** RANGE-DR *is NP-hard in both weighted and unweighted paths.*

Proof. We prove the theorem using a reduction from HAMILTONIAN PATH. Given a graph G, construct the following distance matrix:

$$D_{i,j} = \begin{cases} \{1,\ldots,n-1\} & (v_i,v_j) \in E(G) \ , \\ \{2,\ldots,n-1\} & (v_i,v_j) \notin E(G) \ . \end{cases}$$

If G has a Hamiltonian path, then this path induces a realization of D. On the other hand, a realization of D corresponds to a Hamiltonian path in G. ◀

Next we show that the SET-DR problem on paths is NP-hard even on 5-set distance profiles.

▶ **Theorem 10.** 5-SET-DR *is NP-hard in unweighted and weighted paths.*

Proof. We prove the lemma using a reduction from the 3-COLORING problem.

Consider a graph G, where $V(G) = \{v_1,\ldots,v_n\}$. We construct a distance matrix D on $3n+2$ vertices, denoted $\{u_0,\ldots,u_{3n+1}\}$. Intuitively, the vertices u_{3i-2}, u_{3i-1} and u_{3i} represent vertex v_i in the original graph, and the location of u_{3i-2} encoded the color of v_i. The vertices u_0 and u_{3n+1} are the end-point of the path, and they serve as preference points. More formally, define $\bar{k} \triangleq \lceil k/3 \rceil$. The matrix is defined as follows for any two indices $0 \le k < \ell \le 3n+1$:

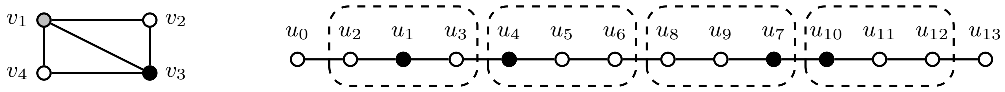
$$D_{k,\ell} = \begin{cases} \{3n+1\} & k=0, \ell=3n+1, \\ \left\{3\bar{\ell}-2, 3\bar{\ell}-1, 3\bar{\ell}\right\} & k=0, \bar{\ell} \in \{1,\ldots,n\}\,, \\ \left\{3n-3\bar{k}+1, 3n-3\bar{k}+2, 3n-3\bar{k}+3\right\} & \bar{k} \in \{1,\ldots,n\}\,, \ell = 3n+1, \\ \{1,2\} & \bar{k} = \bar{\ell}, \\ \left\{3(\bar{\ell}-\bar{k}) + \Delta : \Delta \in \{-2,-1,0,1,2\}\right\} & \bar{k} < \bar{\ell}, \\ & k \bmod 3 \ne 1 \text{ or } \ell \bmod 3 \ne 1 \\ & \text{ or } (v_{\bar{k}}, v_{\bar{\ell}}) \notin E(G), \\ \left\{3(\bar{\ell}-\bar{k}) + \Delta : \Delta \in \{-2,-1,1,2\}\right\} & \bar{k} < \bar{\ell}, \\ & k,\ell \bmod 3 = 1, (v_{\bar{k}}, v_{\bar{\ell}}) \in E(G) \ . \end{cases}$$

Observe that D is a 5-set distance matrix. Also, notice that only the last requirement is not a range, and it consists of a union of two ranges of size 2.

We show that G is 3-colorable if and only if D is realizable using an unweighted path.

Assume that G is 3-colorable and $\chi : V \mapsto \{0,1,2\}$ is a 3-coloring of G. We describe a path realization as a placement of the vertices on integral points from 0 to $3n+1$. First, u_0 is placed on 0 and u_{3n+1} is placed on $3n+1$. Next, for every $i \in \{1,\ldots,n\}$, if $\chi(v_i) = c$, then u_{3i-2} is placed at location $3i-2+c$. The vertices u_{3i-1} and u_{3i} are placed at the two remaining free locations from $\{3i-2, 3i-1, 3i\}$. It is straightforward to verify that P realizes D. We observe that the last requirement of D is satisfied, since χ is a 3-coloring. See example in Figure 3.

Now suppose that P is a path realization of D. First, notice that the above distance matrix makes sure that the distance between u_0 and u_{3n+1} must be $3n+1$. Moreover, all the other distances are strictly less than $3n+1$. Hence, if a path realization exists, then we may assume without loss of generality that u_0 is placed on 0 and u_{3n+1} is placed on $3n+1$. In the weighted case, it follows that all other vertices are located at $\{1,\ldots,3n\}$, which means that all edges are of unit length. Since the distances between u_{3i-2}, u_{3i-1} and u_{3i} are 1 or 2, for



(a) A 3-colorable graph G. **(b)** A realization of D.

Figure 3 An example of the reduction in the proof of Theorem 10. White, gray, and black are colors 0, 1, and 2, respectively. The location of full nodes correspond to the chosen colors.

every $i \in \{1, \ldots, n\}$, these three nodes are forced to appear as a sub-path of P consisting of two edges. Moreover, the required distances from u_0 and u_{3n+1} forces u_{3i-2}, u_{3i-1} and u_{3i} to be assigned to the three consecutive positions $3i-2, 3i-1, 3i$ on the path. We define a coloring χ according to the positions of $\{u_{3i-2} : i \in \{1, \ldots, n\}\}$. More specifically, $\chi(v_i) = c$ if u_{3i-2} is located at $3i - 2 + c$. χ is a 3-coloring, since the last requirement of D ensures that $\chi(v_i) \neq \chi(v_j)$ if $(v_i, v_j) \in E$. ◀

The above proof implies an even stronger result, that we will need in the sequel for the hardness of 5-Set-DR in cycles.

▶ **Theorem 11.** *5-Set-DR is NP-hard in unweighted and weighted paths, even when the required end-points of the path are given in the input.*

5 Cycle Realizations

As was the case with the path, we first show that for 2-set distance profiles a realization by a cycle, if exists, can be found with a polynomial time algorithm. On the other hand, we show that the Set-DR problem on cycles is NP-hard even on 5-set distance profiles, by a reduction from the path realization problem.

5.1 Realization of 2-set distance Profiles by Cycles

A cycle C realizes the 2-set distance profile D if either $dist_C(i,j) = d^1_{i,j}$ or $dist_C(i,j) = d^2_{i,j}$ holds for every i and j.

The proofs of theorems 12 and 13 are given in the appendix.

▶ **Theorem 12.** *There exists a polynomial time algorithm for 2-Set-DR on unweighted cycles.*

▶ **Theorem 13.** *There exists a polynomial time algorithm for 2-Set-DR on weighted cycles.*

5.2 Hardness Result

We start the section with a hardness proof that requires unlimited ranges.

▶ **Theorem 14.** *Range-DR is NP-hard in both weighted and unweighted cycles.*

Proof. We prove the theorem using a reduction from Hamiltonian Cycle. Given a graph G, construct the following distance matrix:

$$D_{i,j} = \begin{cases} \{1, \ldots, \lfloor n/2 \rfloor\} & (v_i, v_j) \in E(G) , \\ \{2, \ldots, \lfloor n/2 \rfloor\} & (v_i, v_j) \notin E(G) . \end{cases}$$

If G has a Hamiltonian cycle, then this cycle induces a realization of D. On the other hand, a realization of D corresponds to a Hamiltonian cycle in G. ◀

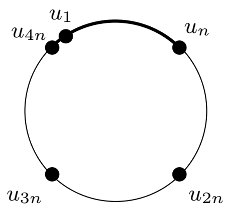


Figure 4 Depiction of the reduction in the proof of Theorem 15, from path realization to cycle realization. The thick line between u_1 and u_n corresponds to the location of the original vertices.

Next we show that SET-DR on cycles is NP-hard even on 5-set distance profiles using reductions from the problem on paths, where the required end-points of the paths are given in the input.

▶ **Theorem 15.** *5-*SET-DR *is NP-hard in unweighted and weighted cycles.*

Proof. We use a reduction from the SET-DR in unweighted paths, where D is a 5-set distance matrix and the required end-points of the path are given in the input, which was shown to be NP-hard in Theorem 11.

Intuitively, we add $3n$ vertices $n+1, \ldots, 4n$, unit weight edges between i and $i+1$, for $i \in \{n, \ldots, 4n-1\}$, and the unit weight edge $(4n, 1)$. Formally, given a matrix $D \in \mathbb{N}^{n \times n}$ we construct a matrix $D' \in \mathbb{N}^{4n \times 4n}$ as follows:

$$D'_{k,\ell} = \begin{cases} D_{k,\ell} & 1 \leq k, \ell \leq n, \\ \min\{(\ell - k), (4n - \ell + k)\} & n < k < \ell \leq 4n, \\ \{\min\{(\ell - \delta - 1), (4n + 1 - \ell + \delta)\} : \delta \in D_{k,1}\} & 1 \leq k \leq n, n < \ell \leq 4n, \end{cases}$$

Note that we assume without loss of generality that $\delta \in D_{k,1}$ if and only if $n - 1 - \delta \in D_{k,n}$.

Suppose that D is realizable. In this case, D' is realizable by using a cycle of total length $4n$, where the vertices $u_1, \ldots, u_n$ are placed at positions $1, .., n$ as they are placed in the path realization. Vertex u_i, for $i > n$ is placed at i. (See Figure 4.)

On the other hand, assume that D' is realizable, and assume that u_1 and u_n are placed at locations 1 and n on the cycle. It follows that u_i is located at i, for every $i > n$. In this case, D can be realized by the arc from 1 to n. ◀

6 Summary and Open Problems

This paper introduces the parametric SET DISTANCE REALIZATION (SET-DR) problem, which is an extension of the RANGE DISTANCE REALIZATION (RANGE-DR) problem. We study the computational complexity of k-SET-DR and k-RANGE-DR, as a function of k, in various graph families

Several questions remain open, including the following.

- RANGE-DR in weighted general graphs can be solved in polynomial time, but the status of SET-DR is currently unclear.
- For trees, 3-RANGE-DR and 2-SET-DR are NP-hard, but the status of the 2-RANGE-DR problem remains unsettled.
- For stars, the hardness of the k-SET-DR problem is unsettled for $k = 3, 4, 5$.
- For paths and cycles the k-SET-DR problem is unsettled for $k = 3, 4$.
- The status of RANGE-DR for paths and cycles is an open problem.

References

1. Ingo Althöfer. On optimal realizations of finite metric spaces by graphs. *Discret. Comput. Geom.*, 3:103–122, 1988.
2. Agnese Baldisserri. Buneman's theorem for trees with exactly n vertices. *CoRR*, 2014. URL: http://arxiv.org/abs/1407.0048.
3. Hans-Jürgen Bandelt. Recognition of tree metrics. *SIAM J. Discret. Math.*, 3(1):1–6, 1990.
4. Amotz Bar-Noy, David Peleg, Mor Perry, and Dror Rawitz. Composed degree-distance realizations of graphs. In *32nd IWOCA*, volume 12757 of *LNCS*, pages 63–77. Springer, 2021.
5. Reuven Bar-Yehuda and Dror Rawitz. Efficient algorithms for integer programs with two variables per constraint. *Algorithmica*, 29(4):595–609, 2001.
6. Fan R. K. Chung, Mark W. Garrett, Ronald L. Graham, and David Shallcross. Distance realization problems with applications to internet tomography. *J. Comput. Syst. Sci.*, 63(3):432–448, 2001. doi:10.1006/jcss.2001.1785.
7. Joseph C. Culberson and Piotr Rudnicki. A fast algorithm for constructing trees from distance matrices. *Inf. Process. Lett.*, 30(4):215–220, 1989.
8. Elias Dahlhaus. Fast parallel recognition of ultrametrics and tree metrics. *SIAM J. Discret. Math.*, 6(4):523–532, 1993.
9. Andreas W. M. Dress. Trees, tight extensions of metric spaces, and the cohomological dimension of certain groups: a note on combinatorial properties of metric spaces. *Adv. in Math.*, 53:321–402, 1984.
10. Shimon Even, Alon Itai, and Adi Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM J. Comput.*, 5(4):691–703, 1976.
11. Tomás Feder, Adam Meyerson, Rajeev Motwani, Liadan O'Callaghan, and Rina Panigrahy. Representing graph metrics with fewest edges. In *20th STACS*, pages 355–366. Springer, 2003.
12. S. Louis Hakimi and S. S. Yau. Distance matrix of a graph and its realizability. *Quart. Appl. Math.*, 22:305–317, 1965.
13. Juhani Nieminen. Realizing the distance matrix of a graph. *J. Inf. Process. Cybern.*, 12(1/2):29–31, 1976.
14. A. N. Patrinos and S. L. Hakimi. The distance matrix of a graph nand its tree realizability. *Quarterly of Applied Math.*, 255, 1972.
15. Elena Rubei. Weighted graphs with distances in given ranges. *J. Classif.*, 33:282–297, 2016.
16. J. M. S. Simões-Pereira. An optimality criterion for graph embeddings of metrics. *SIAM J. Discret. Math.*, 1(2):223–229, 1988.
17. J. M. S. Simões-Pereira. An algorithm and its role in the study of optimal graph realizations of distance matrices. *Discret. Math.*, 79(3):299–312, 1990.
18. Hiroshi Tamura, Masakazu Sengoku, Shoji Shinoda, and Takeo Abe. Realization of a network from the upper and lower bounds of the distances (or capacities) between vertices. In *Proc. IEEE Int. Symp. on Circuits and Systems (ISCAS)*, pages 2545–2548. IEEE, 1993.
19. Sacha C. Varone. A constructive algorithm for realizing a distance matrix. *Eur. J. Oper. Res.*, 174:102–111, 2006.

Appendix

A Omitted Proofs

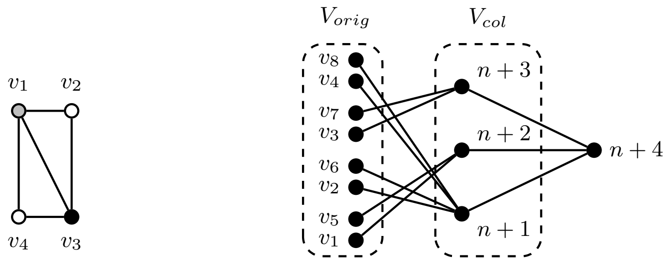
Proof of Theorem 2. We prove the lemma using a reduction from the 3-COLORING problem.

Consider an instance G of the 3-COLORING problem, We construct the following 2-set distance matrix D. D is a distance matrix for $2n+4$ vertices, $\{u_1, \ldots, u_{2n+4}\}$. Intuitively, we think of the first n vertices, $U_{orig} = \{u_1, \ldots, u_n\}$, as representing the *original* vertices of the given graph G. The next n vertices $U'_{orig} = \{u_{n+1}, \ldots, u_{2n}\}$ as *duplicates* of the original vertices. Additional three vertices of D, $U_{col} = \{u_{2n+1}, u_{2n+2}, u_{2n+3}\}$, as representing the three *colors*, and the last vertex u_{n+4} represents a *coordinator*. Let

$$D_{i,j} = \begin{cases} \{1\} & i = 2n+1, \ldots, 2n+3, \quad j = 2n+4, \\ \{2\} & i = 1, \ldots, 2n, \quad j = n+4, \\ \{2\} & 2n+1 \le i < j \le 2n+3, \\ \{2\} & 1 \le i \le n, \quad j = n+i, \\ \{4\} & 1 \le i < j \le n, \quad (v_i, v_j) \in E(G), \\ \{2,4\} & 1 \le i < j \le n, \quad (v_i, v_j) \notin E(G), \\ \{2,4\} & \text{otherwise.} \end{cases}$$

We now prove that the input G is k-colorable if and only if D is realizable using a weighted tree.

Suppose G is 3-colorable, and let $\chi : V \mapsto \{1,2,3\}$ be the coloring function. For the matrix D defined from G, construct a realizing tree T as follows. Start with a star rooted at $2n+4$ with the color vertices $2n+1, \ldots, 2n+3$ as leaves. Connect each original vertex u_i and its duplicate u_{n+i} to the color vertex u_{2n+c}, where $c = \chi(v_i)$. It is easy to verify that T is a tree that realizes D. (See Figure 5 for an example).



(a) A 3-colorable graph. **(b)** A realization of D.

Figure 5 An example of the reduction in the proof of Theorem 2 for $n = 4$. White, gray, and black correspond to nodes $n+1$, $n+2$, and $n+3$, respectively.

On the other hand, suppose that there exists an weighted tree T that realizes D. Let T_i be the minimal subtree of T that contains u_i, u_{n+i}, and u_{2n+4}. Observe that T_i cannot be a path, because the three vertices need to be at distance 2 from each other. Hence, T_i must contain at least one more internal vertex. It follows that r is not directly connected to u_i. The vertex u_j, for $j \neq i$, cannot be the internal vertex in T_i, since it needs to be at distance 2 from u_{2n+4}. The same goes for v_{n+j}. Hence the internal vertex of T_i has to be a color vertex. If there are two color vertices in T_i, then the distance from v_i and v_{n+i} will be more than 2, since both color vertices must be at distance 1 from v_{n+4}. Hence, there is a single color vertex u_{2n+c} in T_i. Since u_{2n+c} is at distance 1 from u_{2n+4}, it follows that it must be at distance 1 from both u_i and u_{n+i}. Define $\chi(v_i) = c$. The distance constraints defined for the original vertices specify that if $(v_i, v_j) \in E(T)$, then their distance must be at least 4. This ensures that none of the color vertices are connected to both u_i and u_j (as this would make their distance 2). It follows that if $(v_i, v_j) \in E(G)$, then v_i and v_j are assigned different colors. ◀

We note that NP-hardness for 3-Range-DR is obtained by setting $D_{i,j} = \{2, 3, 4\}$, for every i and j such that of $D_{i,j} = \{2, 4\}$.

Proof of Theorem 12. Assume that the unweighted cycle is embedded on a circle of length n. We fix a point on the circle and call it 0 and place vertices on locations whose distance from 0 is integral. Clearly, we have n such locations. Denote the above vertices by $0, 1, 2, \ldots, n-1$ going clockwise and also $0, -1, -2, \ldots, -(n-1)$ going counter-clockwise. Note that the maximum distance in this cycle is $\lfloor n/2 \rfloor$.

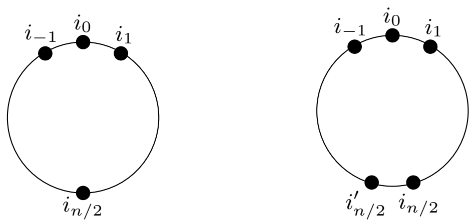
We choose three special vertices, denoted by i_0, i_1, and i_{-1}, place vertex i_0 at location 0 and place i_1 and i_{-1} at locations 1 and at -1, respectively. (See depiction in Figure 6.) If this placement does not comply with D, then the choice of these three vertices is considered a failure.

Next, consider an index $j \neq i_0, i_1, i_{-1}$ and assume that j is not placed at location $n/2$. We claim that only two locations are possible for j after examining the distances in $D_{i_0,j}$, $D_{i_1,j}$, and $D_{i_{-1},j}$. Let $D_{i_0,j} = \{x, y\}$ for $2 \leq x < y < n/2$. Hence, there are four potential locations for j, namely $\{-y, -x, x, y\}$. If j can be placed at x and at y, then it must be the case that $D_{i_1,j} = \{x-1, y-1\}$ and $D_{i_{-1},j} = \{x+1, y+1\}$. It follows that j cannot be placed at $-x$, since $x - 1 \notin D_{i_{-1},j}$, and it cannot be placed at $-y$, since $y + 1 \notin D_{i_1,j}$. A symmetric argument can be used for the case where j can be placed at $-x$ and at $-y$. The pigeonhole principle implies that if j can be placed at more than two locations from $\{-y, -x, x, y\}$ then two of them must be either x and y or $-x$ and $-y$. This contradicts the above arguments. Therefore j can be placed in at most two locations.[3] Denote the two possible locations of j by δ_j^0 and δ_j^1.

Now choose one or two additional special vertices to be placed at $\lfloor \frac{n-1}{2} \rfloor$ and $\lceil \frac{n-1}{2} \rceil$. More specifically, if n is even we choose a vertex $i_{n/2}$ placed at $n/2$ and if n is odd we choose two vertices $i_{n/2}$ and $i'_{n/2}$ placed at $(n-1)/2$ and $(n+1)/2$. (See depiction in Figure 6.) For the other vertices define a Boolean variable x_j, where $x_j =$ false represents that j is located at δ_j^0 and $x_j =$ true represents that j is located at δ_j^1. The entries of D are used to create a 2-CNF formula that is satisfiable if and only if there exists a realization of D on the cycle which corresponds to the above mentioned special vertices.

We assume that n is even. The same arguments can be used for odd n. Consider the possible placements of two vertices $j, k \neq \{i_0, i_1, i_{-1}, i_{n/2}\}$. Since each one has two possible placements, it follows that there are four possible placements of j and k: (i) j at δ_j^0 and k at δ_k^0, (ii) j at δ_j^0 and k at δ_k^1, (iii) j at δ_j^1 and k at δ_k^0, (iv) j at δ_j^1 and k at δ_k^1. For each one of the above four options, check whether it complies with $d_{j,k}^0$ or with $d_{j,k}^1$. This induces a truth table on the variable x_j and x_k that can be represented by at most two 2-CNF clauses.Doing

[3] The following example shows that i_0 and i_1 are not enough to obtain at most two possible locations. Let $D_{i_0,j} = \{h, h+2\}$ and $D_{i_1,j} = \{h-1, h+1\}$, where $h+2 < n/2$. Then the possible locations for j are h, $h+2$, and $-h$.



(a) n is even. **(b)** n is odd.

Figure 6 An example of the algorithm in the proof of Theorem 12, for unweighted cycle realization.

this for all pairs of vertices creates a 2-CNF formula that contains at most $O(n^2)$ clauses by concatenating all the above mentioned clauses.

If there exists a cycle realization C of the profile, where i_0, i_1, i_{-1} and $i_{n/2}$ are placed at $0, 1, -1$ and at $n/2$, then the above formula is satifiable by by the assignment that is induced by the distances of the vertices from i. On the other hand, assume that the 2-CNF formula that is obtained by assuming that i_0, i_1, i_{-1} and $i_{n/2}$ are placed at $0, 1, -1$ and at $n/2$, is satisfiable. A satisfying assignment induces a placement of the vertices on the circle, and this implies a realization of the profile assuming that the placement complies with the matrix entries of the form $D_{j,i_{n/2}}$.

Without loss of generality, assume that $i_0 = 0$. When n is even, there are $\binom{n-1}{3}$ candidates i_1, i_{-1} and $i_{n/2}$, and the total running time of the algorithm is $O(n^5)$. When n is odd, there are $\binom{n-1}{4}$ candidates $i_1, i_{-1}, i_{n/2}$, and $i'_{n/2}$, and the total running time is $O(n^6)$. ◀

Proof of Theorem 13. Assume that the unweighted cycle is embedded on a circle. First, choose a special vertex i_0 and place it at 0. Choose two additional special vertices denoted by i_a and i_b. We assume that i_a is the furthest vertex from i_0 whose distance is measured clockwise, and we assume that i_b is the furthest vertex from i_0 whose distance is measured counter-clockwise. Hence, we choose $\delta_a \in D_{i,i_a}$ and $\delta_b \in D_{i,i_b}$, and place i_a at δ_a and i_b at $-\delta_b$. In addition, we choose $\delta_{ab} \in D_{i_a,i_b}$. If δ_a, δ_b, and δ_{ab} do not satisfy the three possible triangle inequalities, then we considered this a failed attempt. Finally, we assume that the circumference is $\delta_a + \delta_b + \delta_{ab}$. Observe that if $\delta_{ab} = \delta_a + \delta_b$, then if there is a realization whose clockwise distance from i_a to i_b is larger than $2(\delta_a + \delta_b)$, then there is a realization in which this distance is $2(\delta_a + \delta_b)$.

Choose two more special vertices, denoted by i_1, and i_{-1}. Choose a value from $\delta_1 \in D_{i_0,i_1}$, and place i_i at δ_1. Similarly, choose a value from $\delta_{-1} \in D_{i_0,i_{-1}}$, and place i_{-1} at δ_{-1}. If $\delta_1 > \delta_a - 1$, $\delta_{-1} > \delta_b - 1$, of if this placement does not comply with D, then the choice of these vertices is considered a failure.

Now, consider a non-special index j and assume that j. We claim that only two locations are possible for j after examining the distances in $D_{i_0,j}$, $D_{i_1,j}$, and $D_{i_{-1},j}$. Let $D_{i_0,j} = \{x, y\}$ for $1 \leq x < y$. Hence, there are four potential locations for j, namely $\{-y, -x, x, y\}$. If j can be placed at x and at y, then it must be the case that $D_{i_1,j} = \{x - \delta_1, y - \delta_1\}$ and $D_{i_{-1},j} = \{x + \delta_{-1}, y + \delta_{-1}\}$. It follows that j cannot be placed at $-x$, since $x - \delta_{-1} \notin D_{i_{-1},j}$, and it cannot be placed at $-y$, since $y_{\delta_1} \notin D_{i_1,j}$. A symmetric argument can be used for the case where j can be placed at $-x$ and at $-y$. The pigeonhole principle implies that if j can be placed at more than two locations from $\{-y, -x, x, y\}$ then two of them must be either x and y or $-x$ and $-y$. This contradicts the above arguments. Therefore j can be placed in at most two locations. Denote the two possible locations of j by δ_j^0 and δ_j^1.

The next step is to use a reduction to 2-SAT feasibility as was done in the proof of Theorem 12.

Since we have five special vertices, the total running time of the algorithm is $O(n^6)$. ◀