Computational Complexity of Combinatorial Distance Matrix Realisation

David L. Fairbairn[1][0000−0002−5377−9797]⋆,
George B. Mertzios[2][0000−0001−7182−585X]⋆⋆, and
Norbert Peyerimhoff[1][0000−0001−9630−7901]

[1] Department of Mathematical Sciences, Durham University, UK
[2] Department of Computer Science, Durham University, UK

Abstract. The k-CombDMR problem is that of determining whether an $n \times n$ distance matrix can be realised as a sub-matrix by n vertices in some unweighted undirected graph with $n + k$ vertices. This problem has a simple solution in the case $k = 0$. In this paper we show that this problem is polynomial-time solvable for $k = 1$ and $k = 2$, and we provide algorithms to construct such graph realisations by solving appropriate 2-SAT instances. For the case where $k \geq 3$, we prove that the problem becomes NP-complete. We show this by a reduction from the k-colourability problem, where $k \geq 3$. Finally, we present how the simpler problem of tree realisability can be solved in polynomial time for all $k \geq 0$.

Keywords: Distance matrix · Graph realisation · NP-completeness · Polynomial-time algorithm · Graph colourability.

1 Introduction

The ability to realise graphs from a partially known distance function (typically given as a distance matrix) is a fundamental problem with wide-ranging practical applications, including network tomography, phylogenetics, and computational network design. Network tomography, for example, relies on understanding the internal structure of a network based on incomplete information about its distances or connectivity [4, 10]. Phylogenetics, on the other hand, uses distance matrix realisation to infer evolutionary relationships among species, often by reconstructing evolutionary trees from genetic data. A tree realisation with the fewest vertices often corresponds to the most parsimonious evolutionary scenario [16]. Generating graph realisations is crucial for both synthetic data generation and data analysis (for example clustering), allowing the inference of possible structures of underlying systems from observed distances [2, 4, 6]. In computational network design, minimising the number of vertices (e.g. servers)

⋆ Supported by Tharsus Ltd.
⋆⋆ Supported by the Engineering & Physical Sciences Research Council (EPSRC grant EP/P020372/1).

can significantly reduce network cost and complexity where the cost of edges (e.g. server connections) is negligible compared to the cost of vertices [8, 13].

Various graph realisation problems have been studied in the literature, most of them are concerned with weighted graphs with *minimising the sum of the edge weights* as its optimisation criterion [9]. Amongst the results of Hakimi and Yau are a set of necessary and sufficient conditions for graph realisability of a given matrix and a proof of uniqueness of shortest length tree realisations. Dress showed the existence of a weighted graph realisation with minimum total edge weight for any given distance matrix [7]. Moreover, he proved the existence of an optimum solution with at most n^4 vertices for any $n \times n$ distance matrix. In his result, the vertices are only the branch points (vertices with ≥ 3 incident edges) or leaves (vertices with just one incident edges), since all vertices with precisely 2 incident edges can be condensed. Such vertices are called *essential* vertices. Finding weighted graph realisations having the smallest sum of edge weights is NP-hard. More specifically, Althöfer proved that this problem remains NP-hard even in the case where the input distance matrix has integer values (while the edge weights are still real valued) [1]. Althöfer also showed that in the case of integer valued distance matrices, there is always an optimum realisation with rational edge weights. Chung, Garrett and Graham considered a weak version of the weighted graph realisation problem, namely, finding optimum graph realisations for which the distance matrix provides a lower bound on the distances of the corresponding n vertices [4]. They showed that even this weak version of the problem is NP-hard.

In contrast, this paper's focus is the problem of finding combinatorial graph realisations for a prescribed integer valued distance matrix with a prescribed number of additional vertices. Our motivation stems from the field of Multi-Agent Path-Finding (MAPF) [3, 12, 17], where the problem is to find a set of collision-free paths for a group of agents from their unique start locations to their respective unique goal locations within some graph, or within some temporal graph (i.e. a graph whose structure changes over time). Klobas et al. 2022 [12] studied the computational complexity of the problem of finding temporally disjoint paths and walks in temporal graphs, i.e. paths and walks which do never visit the same vertex at the same time. The recent paper by Atzmon et al. 2023 [3] studied the problem of computing solutions to the MAPF problem, only by utilising the pairwise distances among specific vertices (the "terminals"), while the computed paths are allowed to use any number of non-terminal vertices of the graph. Similarly to [3], in this paper we are again given the pairwise distances among terminal vertices, but we are not given the input graph, and the goal is to generate a graph that respects the given terminal distances by adding the smallest number of additional, non-terminal, vertices.

Throughout this paper we use the notation $[n] = \{1, 2, \ldots, n\}$ for any $n \in \mathbb{N}$ and denote the set of all non-negative integers by $\mathbb{N}_0$ (that is $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$). First we introduce the following problem for every integer $k \in \mathbb{N}_0$.

Problem 1. k-COMBINATORIAL DISTANCE MATRIX REALISATION PROBLEM (k-COMBDMR)

Input: An $n \times n$ matrix D with non-negative integer values.
Question: Does there exist a simple (unweighted) graph $G = (V, E)$ with $|V| \leq n + k$ and an injective mapping $\Phi : [n] \to V$ such that the shortest-path distance function d in G satisfies

$$d(\Phi(i), \Phi(j)) = D_{ij}$$

for all $i, j \in [n]$?

We call such a pair (G, Φ) a *graph realisation* of D. Given an $n \times n$ matrix D, any graph realisation (G, Φ) which has the smallest number of vertices is called a *minimum graph realisation* of D.

Example 2. Two possible graph realisations of the following matrix D, are given in Figure 1 with $n = 3$, while $k = 3$ and $k = 1$, respectively.

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

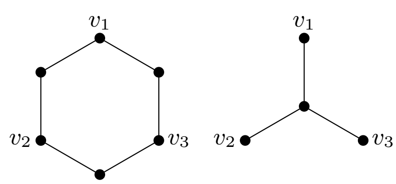

Fig. 1. Two graph realisations of the above matrix D, where $\Phi(i) = v_i$ for $i \in [3]$, while $k = 3$ and $k = 1$ in the left and the right realisation, respectively. The right realisation is *minimum*.

It is important to note that k-CombDMR is distinct from the weighted graph realisation problem. Within the weighted graph realisation problem the edges are equipped with positive real valued weights (their lengths) with the aim to minimise the sum of the edge weights of the graph realisation, whereas in k-CombDMR we are only concerned with minimising the number of vertices in the graph realisation. Take for instance the distance matrix D below and optimum solutions within these two problems as shown in Figure 2.

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

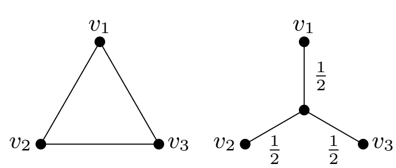

Fig. 2. Minimum graph realisations of D above, for k-CombDMR (Left) and for the weighted graph realisation problem (Right).

Figure 2 shows that optimum solutions for k-CombDMR and for the weighted graph realisation problem can differ significantly. An optimum solution for k-CombDMR may not be trivially transformed into an optimum solution for the weighted graph realisation problem, and vice versa. The reader may ask what happens if we consider the weighted graph realisation problem with the additional constraint that the weights must be integers – could we transform any optimum solution for this problem into an optimum solution for k-CombDMR by replacing weighted edges with paths of length equal to the weight? Figure 3 shows that this is not always the case, as the weighted graph realisation problem with integer weights may have multiple solutions, some of which do not have a corresponding optimum solution for k-CombDMR under this transformation. Therefore, k-CombDMR is also a distinct problem from the weighted graph realisation problem with integer weights.

$$D = \begin{pmatrix} 0&2&2&2&1&3&3&1 \\ 2&0&2&2&3&1&3&1 \\ 2&2&0&2&3&1&1&3 \\ 2&2&2&0&1&3&1&3 \\ 1&3&3&1&0&4&2&2 \\ 3&1&1&3&4&0&2&2 \\ 3&3&1&1&2&2&0&4 \\ 1&1&3&3&2&2&4&0 \end{pmatrix} \tag{1}$$

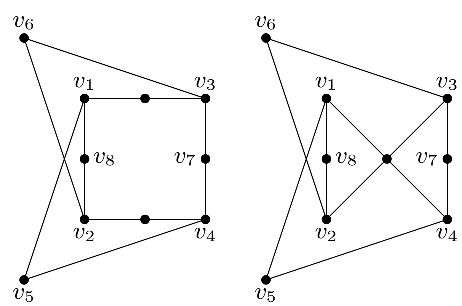

Fig. 3. Optimum graph realisations of D as in (1), with $\Phi(i) = v_i, i \in [8]$ for the weighted graph realisation problem with integer weights $w(e) = 1$ (Left and Right). Only the right graph realisation is optimum for the combinatorial distance realisation problem.

Our results. In this paper we prove that k-CombDMR can be solved in polynomial time for $k \leq 2$, while it is NP-complete for $k \geq 3$. In Section 3 we provide a polynomial-time algorithm for $k \leq 2$ which solves appropriate 2-SAT instances (see Algorithm 20 and Theorem 21). In Section 4 we provide, as our main result, an NP-hardness reduction from the k-colourability problem for every fixed $k \geq 3$

(see Theorem 27). In Section 5, we consider the simpler polynomial-time solvable problem of tree realisability for a given distance matrix (see Corollary 30).

In Section 2 we introduce our main notions and foundational results. In particular, we introduce the notion of the *q-skeleton* and its associated distance matrix (see Definition 6), which are crucial for the polynomial-time results in Section 3. In a nutshell, given the input $n \times n$ distance matrix D, the q-skeleton is the weighted graph of n vertices, where every edge connects vertices of distance at most q in D, and the weight of such an edge is equal to the corresponding distance in D. Although the notion of a q-skeleton is simple and natural, it turns out to be quite powerful, as it allows us to deduce useful upper and lower bounds for the additional vertices needed in a minimum graph realisation of D (see Propositions 8 and 10).

2 Notions and Foundational Results

We begin by identifying the necessary and sufficient conditions for an input matrix D to admit at least one graph realisation.

Definition 3 (Distance matrix). *Let D be an $n \times n$ matrix with non-negative integer valued entries. We call D a* distance matrix *if it satisfies the following properties:*

(i) All diagonal entries of D are zero and all non-diagonal entries are strictly positive.
(ii) D is a symmetric matrix.
(iii) For all $i, j, w \in [n]$, we have

$$D_{iw} + D_{wj} \geq D_{ij}.$$

This definition gives rise to the following result.

Proposition 4. *Let D be an $n \times n$ matrix with non-negative integer valued entries. D is a distance matrix if and only if D admits at least one graph realisation $(G = (V, E), \Phi)$. Furthermore, a graph realisation is obtained by connecting vertices v_i, v_j by a path of length D_{ij} for all $i < j$ such that no two such paths have common interior vertices and $\Phi(i) = v_i$ for all $i \in [n]$. We call such paths* elementary paths *of the graph G.*

Therefore, we will assume that D is a distance matrix with integer valued entries for all instances of k-CombDMR. As Proposition 4 shows, we can always find a graph realisation of a distance matrix D with some number of additional vertices. Note that, in the weighted case, a graph realisation without any additional vertices can be constructed by replacing the elementary paths in Proposition 4 by single edges with appropriate weights. Another immediate consequence of the above construction is the following upper bound on the number of vertices for the existence of a graph realisation of D.

Proposition 5. *Let D be an $n \times n$ distance matrix. Then there exists a graph realisation $(G = (V, E), \Phi)$ of D with*

$$|V| \leq n + \sum_{1 \leq i < j \leq n} (D_{ij} - 1).$$

We now seek to improve the result of Proposition 5, and in doing so, we introduce the following weighted graph, whose distance matrix will be of fundamental importance.

Definition 6 (q-skeleton). *Let D be an $n \times n$ distance matrix and $q \in \mathbb{N}$. The* q-skeleton of D *is the weighted graph $G^q = (V^q, E^q, w)$ with vertices $V^q = [n]$ and edges*

$$E^q = \{\{i, j\} \in [n] \times [n] \mid (i < j) \wedge (D_{ij} \leq q)\},$$

that is, G^q has an edge between i and j if and only if $D_{ij} \leq q$. Additionally, let the edge weights $w : E^q \to \mathbb{N}$ be given by

$$w(i, j) = D_{ij}, \quad \{i, j\} \in E^q.$$

Let $d_{G^q} : V \times V \to \mathbb{N}_0 \cup \{\infty\}$ be the associated distance function of G^q, that is, $d_{G^q}(i, j)$ is the length of the shortest path between i and j in G^q and equal to ∞ if no such path exists. The $n \times n$ matrix $D^{(q)}$, given by $D^{(q)}_{ij} = d_{G^q}(i, j)$, is called the distance matrix of the q-skeleton of D.

Of particular relevance is the following fact:

$$D^{(q)}_{ij} \begin{cases} = D_{ij} & \text{if } D_{ij} \leq q, \\ \geq D_{ij} & \text{if } D_{ij} > q. \end{cases}$$

Notice, when we have $D^{(q)} = D$ for some q, then we can replace each edge $\{i, j\} \in E^q$ with an elementary path of length D_{ij} in G^q to obtain a graph realisation of D. Furthermore, $D^{(q)}$ can be computed by any polynomial-time weighted all-pairs shortest-paths (APSP) algorithm. In fact, we have the following ordering of the matrices $D^{(q)}$.

Lemma 7. *Let D be an $n \times n$ distance matrix and $m = \max\{D_{ij} : 1 \leq i < j \leq n\}$. Let $D^{(q)}$ be the distance matrix of the q-skeleton of D. Then we have,*

$$D^{(1)}_{ij} \geq D^{(2)}_{ij} \geq \cdots \geq D^{(m)}_{ij} = D_{ij} \quad \textit{for all } i, j \in [n].$$

Now let q_0 denote the smallest $q \in \mathbb{N}$ such that $D^{(q)} = D$. Then we have the following improvement on Proposition 5, utilising the q-skeleton of D.

Proposition 8. *Let D be an $n \times n$ distance matrix and $q_0 \in \mathbb{N}$ be the smallest $q \in \mathbb{N}$ such that $D^{(q)} = D$, where $D^{(q)}$ the distance matrix of the q-skeleton of D. Then there exists a graph realisation $(G = (V, E), \Phi)$ of D with*

$$|V| \leq n + \sum_{(1 \leq i < j \leq n) \wedge (2 \leq D_{ij} \leq q_0)} (D_{ij} - 1). \tag{2}$$

Using again the q-skeleton of D we can now provide a lower bound on the number of vertices required for a graph realisation of D (see Proposition 10). To do so, we generalise the earlier notion of elementary paths as follows: for a graph realisation (G, Φ) of D, an *elementary path* is a path of length D_{ij} between vertices v_i and v_j with no interior vertices in $\Phi([n])$. Proposition 10 then follows from the following result.

Proposition 9. *Let $s \in \mathbb{N}$, D be an $n \times n$ distance matrix, $D^{(q)}$ be the distance matrix of the q-skeleton of D and (G, Φ) be a graph realisation of D. If there are no elementary paths of length greater than s in G, then $D^{(s)} = D$.*

Proposition 10. *Let D be an $n \times n$ distance matrix and $q_0 \in \mathbb{N}$ be the smallest $q \in \mathbb{N}$ such that $D^{(q)} = D$, with $D^{(q)}$ the distance matrix of the q-skeleton of D. Any graph realisation $(G = (V, E), \Phi)$ of D must satisfy $|V| \geq n + (q_0 - 1)$.*

Combining Proposition 8 and Proposition 10 we obtain the following Proposition.

Proposition 11. *Let D be an $n \times n$ distance matrix and $D^{(q)}$ be the distance matrix of the q-skeleton of D. Then D has a graph realisation (G, Φ) with $|V| = n$ if and only if $D^{(1)} = D$.*

Due to its general significance throughout this paper, we introduce for any distance matrix D the unweighted graph G_D, which is simply the 1-skeleton G^1 of D as in Definition 6 without the edge weights. Note that Proposition 11 is equivalent to the following theorem.

Theorem 12 ([9]). *For any $n \times n$ distance matrix D, the following statements are equivalent:*

- *A graph realisation $(G = (V, E), \Phi)$ of D with $|V| = n$ exists.*
- *A graph realisation of D is $(G_D = (V_D, E_D), \Phi)$ with $V_D = v_1, \ldots, v_n$ and $\Phi(i) = v_i$ for $i \in [n]$.*

The graph realisation of a distance matrix D with $|V| = n$ is also unique up to isomorphism (See [9]).
0-CombDMR is solved by Theorem 12 as follows:

1. Construct the graph G_D.
2. Check if the distance function of G_D coincides with D.

If D has a graph realisation with $|V| = n$, then we call D a *self-realising distance matrix*. Note then the importance of the graph G_D in the following proposition.

Proposition 13. *Let D be an $n \times n$ distance matrix and $G_D = (V_D, E_D)$ be the associated unweighted graph, with $V_D = \{v_1, \ldots, v_n\}$. Then any graph realisation (G, Φ) of D with $\Phi(i) = v_i$ for $i \in [n]$ has G_D as the induced subgraph on the vertices $v_1, \ldots, v_n$.*

3 Polynomial solutions of 1-CombDMR and 2-CombDMR

In this section, we consider the cases where $k = 1$ and $k = 2$. If D is not self-realising, we know that $|V| \geq n+1$ for any graph realisation $(G = (V,E), \Phi)$ of D. Specifically for $k = 1$ and $k = 2$ there are three possibilities for a graph realisation, namely, the graph has a single additional vertex (•), two additional vertices which are not adjacent (⁚), or two additional vertices which are adjacent (⦙). We now aim to develop polynomial-time algorithms for each of these cases (•,⁚,⦙) separately. Our algorithms will also provide such graph realisations if they exists.

The approach we take to solve these problems involves constructing a particular 2-Satisfiability (2-SAT) instance [15] for each case, which can be solved in polynomial-time. We seek to show that a satisfying assignment of the 2-SAT instance gives rise to a graph with each respective property which can subsequently be checked if it is a realisation of D, and if not then we will prove that no graph realisation of D with the respective property exists.

Formally, a 2-SAT instance is expressed as a 2-CNF formula ϕ, which is the conjunction of a set of clauses, that is

$$\phi = \bigwedge_{i=1}^{m} c_i = c_1 \wedge \ldots \wedge c_m$$

for some finite $m \in \mathbb{N}_0$, where each clause c_i is a disjunction of two literals:

$$c_i = (\ell_i^{(1)} \vee \ell_i^{(2)}).$$

Here, each of $\ell_i^{(1)}, \ell_i^{(2)}$ is a literal, i.e., either a variable x or its negation $\bar{x}$.

Within any graph realisation $(G = (V,E), \Phi)$ for these cases with $\Phi(i) = v_i$ we know that the induced subgraph on vertices $v_1, \ldots, v_n$ agrees with G_D by Proposition 13. Therefore, each case must start with the construction of the graph G_D as in Proposition 13.

We now construct the three 2-SAT instances ϕ_1, ϕ_2 and ϕ_3 for the cases (•,⁚,⦙), respectively. In the below construction each step will be suffixed with the cases it is relevant to and ϕ represents ϕ_1, ϕ_2 or ϕ_3, respectively. Furthermore, in the case (•) let $K = \{n+1\}$, and in the cases (⁚,⦙) let $K = \{n+1, n+2\}$.

1. [•,⁚,⦙] Let $G_D = (V_D, E_D)$ be the graph as described in Proposition 13 with vertices $V_D = \{v_1, \ldots, v_n\}$ and $\Phi(i) = v_i$ for $i \in [n]$.
2. [•,⁚,⦙] Let $\{x_{i,k} : i \in [n], k \in K\}$ be the set of boolean variables representing the existence of an additional edge $\{v_i, v_k\}$ to those already in G_D. That is, $x_{i,k}$ is true if and only if v_i is adjacent to v_k in G.
3. [•,⁚,⦙] For all $i, j \in [n], k \in K$ with $D_{ij} > 2$, we know that the vertices v_i and v_k must not both be adjacent to v_j in any graph realisation of D, as otherwise this would result in a distance of 2 between v_i and v_j. This condition is equivalent to the following clause being satisfied:

$$(\bar{x}_{i,k} \vee \bar{x}_{j,k}). \tag{3}$$

We therefore add the clause (3) to ϕ for all $i, j \in [n], k \in K$ with $D_{ij} > 2$.

4.[•]For all $i, j \in [n]$, with, $D_{ij} = 2$ and $d_{G_D}(v_i, v_j) > 2$, the following boolean expression must be satisfied:

$$(x_{i,n+1} \wedge x_{j,n+1}), \tag{4}$$

meaning that the distance between v_i and v_j must be 2 and realised by a path of length 2 via v_{n+1}. The boolean expression (4) is equivalent to the following two clauses both being satisfied:

$$(x_{i,n+1} \vee x_{i,n+1}), (x_{j,n+1} \vee x_{j,n+1}). \tag{5}$$

Therefore, we add the clauses (5) to ϕ for all $i, j \in [n]$ with $D_{ij} = 2$ and $d_{G_D}(v_i, v_j) > 2$ as there is no other way to realise a distance of 2 between v_i and v_j in G.

5.[⁝, ⁝]For all $i, j \in [n]$, such that, $D_{ij} = 2$ and $d_{G_D}(v_i, v_j) > 2$, we know the following boolean expression must be satisfied:

$$(x_{i,n+1} \wedge x_{j,n+1}) \vee (x_{i,n+2} \wedge x_{j,n+2}), \tag{6}$$

meaning that the distance between v_i and v_j must be 2 and realised via a path of length 2 via v_{n+1} or v_{n+2}. The boolean expression (6), by distributivity, is equivalent to the following 4 clauses being satisfied:

$$\begin{gathered}(x_{i,n+1} \vee x_{i,n+2}), (x_{j,n+1} \vee x_{j,n+2}),\\ (x_{i,n+1} \vee x_{j,n+2}), (x_{j,n+1} \vee x_{i,n+2}).\end{gathered} \tag{7}$$

Therefore, we add the clauses (7) to ϕ for all $i, j \in [n]$ such that $D_{ij} = 2$ and $d_{G_D}(v_i, v_j) > 2$.

6.[⁝]Compute $D^{(2)}$ the distance matrix of the 2-skeleton of D.

7.[⁝]For all $i, j \in [n]$ with $D_{ij} > 3$, we know that, if v_i is adjacent to v_{n+1} then v_j cannot be adjacent to v_{n+2}, as otherwise this would result in a distance of 3 between v_i and v_j. Similarly, if v_i is adjacent to v_{n+2} then v_j cannot be adjacent to v_{n+1}. This condition is equivalent to the following clauses both being satisfied:

$$(\bar{x}_{i,n+1} \vee \bar{x}_{j,n+2}), (\bar{x}_{i,n+2} \vee \bar{x}_{j,n+1}). \tag{8}$$

Therefore, add the clauses (8) to ϕ for all $i, j \in [n]$ with $D_{ij} > 3$.

8.[⁝]For all $i, j \in [n]$ with $D_{ij} = 3$ and $D^{(2)}_{ij} > 3$, the following boolean expression must be satisfied (by Lemma 14 below):

$$(x_{i,n+1} \wedge x_{j,n+2}) \vee (x_{i,n+2} \wedge x_{j,n+1}), \tag{9}$$

meaning that the distance between v_i and v_j must be 3 and it must be realised via a path of length 3 through v_{n+1} and v_{n+2}. The boolean expression (9), is equivalent to the following four clauses being satisfied:

$$\begin{gathered}(x_{i,n+1} \vee x_{i,n+2}), (x_{j,n+1} \vee x_{j,n+2}),\\ (x_{i,n+1} \vee x_{j,n+1}), (x_{i,n+2} \vee x_{j,n+2}).\end{gathered} \tag{10}$$

Therefore, we add the clauses (10) to ϕ for all $i, j \in [n]$ with $D_{ij} = 3$ and $D^{(2)}_{ij} > 3$.

This concludes the construction of the 2-SAT instances ϕ_1, ϕ_2 and ϕ_3 for the cases (•, ⁚, ⁞), respectively. As noted in the construction we have the following lemma in the case (⁞).

Lemma 14. *Let D be an $n \times n$ distance matrix and $i, j \in [n]$. Assume $D_{ij} = 3$ and $D^{(2)}_{ij} > 3$, where $D^{(2)}$ is the distance matrix of the 2-skeleton of D. In any graph realisation $(G = (V, E), \Phi)$ of D with $V = \{v_i = \Phi(i) : i \in [n]\} \cup \{v_{n+1}, v_{n+2}\}$, with v_{n+1} adjacent to v_{n+2}, any shortest path from v_i to v_j in G must be of the following form:*

$$v_i \to v_{n+1} \to v_{n+2} \to v_j \quad \text{or} \quad v_i \to v_{n+2} \to v_{n+1} \to v_j.$$

Proof. Let $(G = (V, E), \Phi)$ be a graph realisation of D with $V = \{v_i = \Phi(i) : i \in [n]\} \cup \{v_{n+1}, v_{n+2}\}$. Since $D_{ij} = 3$, there exists a shortest path of the form $v_i \to v_s \to v_t \to v_j$ for some $s, t \in [n+2]$. If $\{s, t\} \neq \{n+1, n+2\}$ then this shortest path is a concatenation of elementary paths of length 1 or 2 and therefore $D_{ij} = D^{(2)}_{ij}$, which is a contradiction to the assumption that $D^{(2)}_{ij} > 3$.

We now introduce the following definition. A truth assignment of boolean variables $\{x_{i,k}\}$ is said to be *consistent* with a graph $G = (V, E)$ with $V \supset \{v_1, \ldots, v_m\}$ when

$$x_{i,k} = \begin{cases} \text{True} & \text{if } \{v_i, v_k\} \in E, \\ \text{False} & \text{if } \{v_i, v_k\} \notin E. \end{cases}$$

Since all clauses in ϕ_1, ϕ_2 and ϕ_3 are nessessary conditions for a graph realisation of D, we have the following observation.

Observation 15. *Let D be an $n \times n$ distance matrix. If ϕ_1, ϕ_2 or ϕ_3 is not satisfiable then no graph realisation $(G = (V, E), \Phi)$ of D in the respective case (•, ⁚, ⁞) exists.*

Note that any graph realisation $(G = (V, E), \Phi)$ of D for the cases (•, ⁚, ⁞) gives rise to a satisfying assignment $\mathbf{X}$ of ϕ_1, ϕ_2 or ϕ_3, respectively, which is consistent with G.

We now seek to show that we require only a single satisfying assignment to determine whether such a graph realisation exists. Let $G_{\phi_i,\mathbf{X}}$ be the unique graph with vertex set $V = \{v_1, \ldots, v_n\}$, and additional vertices v_{n+1}, v_{n+2} (if required), having G_D as the induced subgraph on the vertices $v_1, \ldots, v_n$ and consistent with a satisfying assignment $\mathbf{X}$ of ϕ_i. For such a graph, let $D(\phi_i, \mathbf{X})$ denote the $n \times n$ distance matrix of $G_{\phi_i,\mathbf{X}}$ over the vertices $\{v_1, \ldots, v_n\}$. Computing $D(\phi_i, \mathbf{X})$ can be done in polynomial-time via an all-pairs shortest-paths (APSP) algorithm.

Lemma 16. *Let $\mathbf{X}$ be a satisfying assignment of $\phi \in \{\phi_1, \phi_2\}$ and $D^{(2)}$ be the distance matrix of the 2-skeleton of D. Then*

$$D(\phi, \mathbf{X}) = D^{(2)}.$$

Lemma 17. *Let* $\mathbf{X}$ *be a satisfying assignment of* ϕ_3 *and* $D^{(3)}$ *be the distance matrix of the 3-skeleton of* D*. Then*

$$D(\phi_3, \mathbf{X}) = D^{(3)}.$$

An immediate consequence of Lemma 16 and Lemma 17 is the following corollary.

Corollary 18. *Let* $\mathbf{X}$ *and* $\mathbf{X}'$ *be two distinct satisfying assignments of* $\phi \in \{\phi_1, \phi_2, \phi_3\}$*. Then*

$$D(\phi, \mathbf{X}) = D(\phi, \mathbf{X}').$$

Corollary 18 tells us that the distance matrix $D(\phi_i, \mathbf{X})$ of $G_{\phi_i,\mathbf{X}}$ is invariant over all satisfying assignments of ϕ_i for $i \in \{1, 2, 3\}$. Therefore, if we can find a single satisfying assignment $\mathbf{X}$ of ϕ_i, then we can construct $G_{\phi_i,\mathbf{X}}$, and if $D(\phi_i, \mathbf{X}) = D$ then we have found a graph realisation $(G = (V, E), \Phi)$ of D for the respective case (•, ⁖, ⁚). It remains to show that, in the case $D(\phi_i, \mathbf{X}) \neq D$, no graph realisation of D with the respective property exists.

Proposition 19. *Let* D *be an* $n \times n$ *distance matrix. If* $D(\phi_i, \mathbf{X}) \neq D$ *for some satisfying assignment* $\mathbf{X}$ *of* ϕ_i*,* $i \in \{1, 2, 3\}$*, then no graph realisation* $(G = (V, E), \Phi)$ *of* D *with the respective property* (•, ⁖, ⁚) *exists.*

Proof. Assume that such a graph realisation $(G = (V, E), \Phi)$ of D exists with the respective property (•, ⁖, ⁚). By Observation 15, we know that there exists a satisfying assignment $\mathbf{X}'$ of ϕ_i, consistent with the graph G, such that $D(\phi_i, \mathbf{X}') = D$. By Corollary 18, we know that $D(\phi_i, \mathbf{X}) = D(\phi_i, \mathbf{X}')$ which is a contradiction to $D(\phi_i, \mathbf{X}) \neq D$.

Therefore, we have the following polynomial-time algorithm to determine whether there exists a graph realisation $(G = (V, E), \Phi)$ of D with the respective property (•, ⁖, ⁚). Moreover, the algorithm produces such a graph realisation if it exists.

Algorithm 20 (Solving 1-CombDMR and 2-CombDMR).
Input: *An* $n \times n$ *distance matrix* D *and property* (•, ⁖, ⁚).
Output: *A graph realisation* $(G = (V, E), \Phi)$ *of* D *with the respective property* (•, ⁖, ⁚) *if it exists, or a statement that no such graph realisation exists.*

1. *Let* $\Phi(i) = v_i$ *for* $i \in [n]$.
2. *Construct the 2-CNF formula* ϕ*, corresponding to the input property.*
3. *Compute a satisfying assignment* $\mathbf{X}$ *of* ϕ*, if it exists.*
4. *If* ϕ *is not satisfiable, then no graph realisation* $(G = (V, E), \Phi)$ *of* D *with the respective property* (•, ⁖, ⁚) *exists. (By Observation 15)*
5. *If* ϕ *is satisfiable, then construct the graph* $G_{\phi,\mathbf{X}}$ *consistent with the satisfying assignment* $\mathbf{X}$.
6. *Compute the distance matrix* $D(\phi, \mathbf{X})$ *of* $G_{\phi,\mathbf{X}}$ *(using any APSP algorithm).*
7. *If* $D(\phi, \mathbf{X}) = D$ *then* $(G_{\phi,\mathbf{X}}, \Phi)$ *is a graph realisation of* D *with the respective property* (•, ⁖, ⁚).

Otherwise, if $D(\phi, \mathbf{X}) \neq D$ then no graph realisation of D with the respective property (•, ⁚, ⁚) exists. (By Proposition 19)

The following theorem summarises of our results in this section.

Theorem 21. *The 1-*CombDMR *and 2-*CombDMR *problems are polynomial-time solvable.*

4 k-CombDMR is NP-complete for $k \geq 3$

In this section we prove that k-CombDMR is NP-complete for every $k \geq 3$, via a reduction from k-colourability, which is known to be NP-complete [11, 14, 18]. For the readers' convenience we restate the k-colourability problem as follows:

Problem 22 (k-colourability). Given a graph $G = (V, E)$. Does there exist a function $\chi : V \to [k]$ such that for all $\{i, j\} \in E$ we have $\chi(i) \neq \chi(j)$?

As we will prove, k-Colourability can be reduced to the k-CombDMR problem by the following reduction algorithm.

Algorithm 23 (Reduction of k-Colourability to k-CombDMR).
Input: *A connected simple undirected graph $G_c = (V_c, E_c)$ for which we want to determine if it is k-colourable.*
Output: *A distance matrix D such that G_c is k-colourable if and only if k-*CombDMR *for D is a YES-instance.*

1. *Enumerate the vertices of G_c such that $V_c = \{v_1, \dots, v_{n_c}\}$ where $n_c = |V_c|$.*
2. *Construct the gadget graph $G_g = (V_g, E_g)$, with $V_c \subseteq V_g$, as follows. We subdivide each edge in E_c twice, i.e., we replace each edge by a path of length 3 (containing two new vertices). For every pair of non-adjacent vertices in G_c, we add a path of length 2 between them (containing one new vertex). We enumerate the vertices of G_g such that $V_g = \{v_1, \dots, v_{n_c}, v_{n_c+1} \dots, v_{n_g}\}$ where $v_{n_c+1}, \dots, v_{n_g}$ are the new vertices and $n_g = |V_g|$, see Figure 4.*
3. *Let d_{G_g} denote the shortest path distance function of G_g. Construct the $n \times n$ distance matrix D where $n = n_g + 1$, with entries,*

$$\begin{aligned} D_{ij} &= d_{G_g}(v_i, v_j) && \text{for } i, j \in [n_g], \\ D_{i'n} &= D_{ni'} = 2 && \text{for } i' \in [n_c], \\ D_{i''n} &= D_{ni''} = 3 && \text{for } i'' \in [n_g] \setminus [n_c], \\ D_{nn} &= 0. \end{aligned}$$

This will result in a distance matrix of the form:

$$D = \left[\begin{array}{cc|c} \multicolumn{2}{c|}{\multirow{2}{*}{$\begin{array}{c} D_{ij} = d_{G_g}(v_i, v_j) \\ i, j \in [n_g] \end{array}$}} & \begin{array}{c} 2 \\ \vdots \\ 2 \end{array} \\ \cline{3-3} & & \begin{array}{c} 3 \\ \vdots \\ 3 \end{array} \\ \hline 2 \cdots\cdots 2 & 3 \cdots\cdots 3 & 0 \end{array}\right]$$

An example of the construction of a gadget graph as in Algorithm 23 is illustrated in Figure 4. Figure 5 illustrates an example graph realisation of D, where the n^{th} row and column correspond to the vertex v_n.

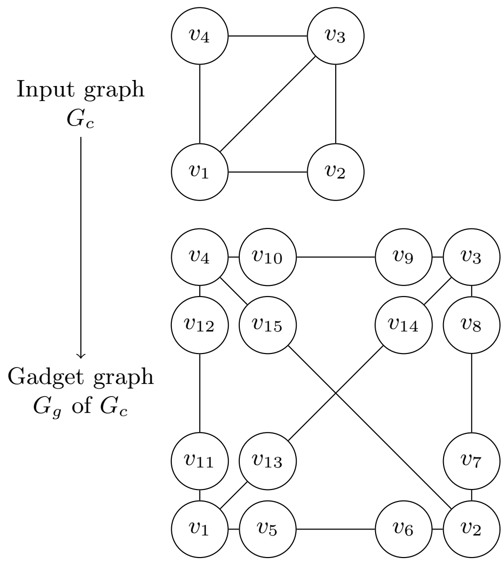

Fig. 4. Example of construction of a gadget graph G_g from an input graph G_c as in Algorithm 23 with old vertices $v_1, \ldots, v_4$ (i.e., $n_c = 4$) and new vertices $v_5, \ldots, v_{15}$ (i.e., $n_g = 15$).

Proposition 24. *The constructed matrix D of Algorithm 23 satisfies the conditions of Definition 3 and is therefore a distance matrix.*

Now that we have established that the constructed matrix D is a valid distance matrix, our next aim is to prove that, if D is a YES-instance of k-CombDMR then G_c is k-colourable (Proposition 26 below). We start with the following useful lemma.

Lemma 25. *Given an input graph $G_c = (V_c, E_c)$ and $k \in \mathbb{N}$. Let $(G = (V, E), \Phi)$ be any graph realisation of the constructed $n \times n$ distance matrix D by Algorithm 23, with $|V| = n + k$ vertices. Let $v_{n+1}, \ldots, v_{n+k} \in V \setminus \Phi([n])$. Any two vertices v_i and v_j adjacent in the input graph G_c cannot both be adjacent to the same vertex $v \in \{v_{n+1}, \ldots, v_{n+k}\}$ in G.*

Proof. Given a graph realisation $(G = (V, E), \Phi)$ of D with $|V| = n + k$ vertices and $\Phi(i) = v_i$ for $i \in [n]$, with D constructed by Algorithm 23 and let d_G denote the shortest path distance function of G. As v_i and v_j are adjacent in G_c, by construction $D_{ij} = 3$ and $d_G(v_i, v_j) = 3$. If v_i and v_j were both adjacent to the same vertex $v \in \{v_{n+1},\dots, v_{n+k}\}$ in G then $d_G(v_i, v_j) \leq 2$, which would be a contradiction to G being a graph realisation of D.

Proposition 26. *Given an input graph $G_c = (V_c, E_c)$ and D the $n \times n$ matrix as constructed in Algorithm 23. If D is a YES-instance of k-CombDMR then G_c is k-colourable.*

Proof. Assume a graph realisation $(G = (V, E), \Phi)$ of D as constructed by Algorithm 23 with $|V| = n + k$ exists. Without loss of generality let $\Phi(i) = v_i$ for $i \in [n]$ and let $\{v_{n+1}, \dots, v_{n+k}\} = V \setminus \Phi([n])$. Then, by Lemma 25 we know that any two adjacent vertices in G_c cannot both be adjacent to the same vertex $v \in \{v_{n+1}, \dots, v_{n+k}\}$ in G. Furthermore, we know that each vertex v_i for $i \in [n_c]$ must be adjacent to at least one of the vertices $v \in \{v_{n+1}, \dots, v_{n+k}\}$ in G to realise the $D_{in} = D_{ni} = 2$ distances in D. We construct a colouring of the vertices of G_c by assigning a colour to each of the vertices $v_{n+1}, \dots, v_{n+k}$ and then assign the same colour to any vertex v_i for $i \in [n_c]$ which is adjacent to that vertex (with arbitrary choice in the case of multiple adjacent vertices $v_{n+1}, \dots, v_{n+k}$). This is a valid k-colouring due to Lemma 25.

The colour assignment in the above proof is illustrated in Figure 5 as a continuation of the example in Figure 4.

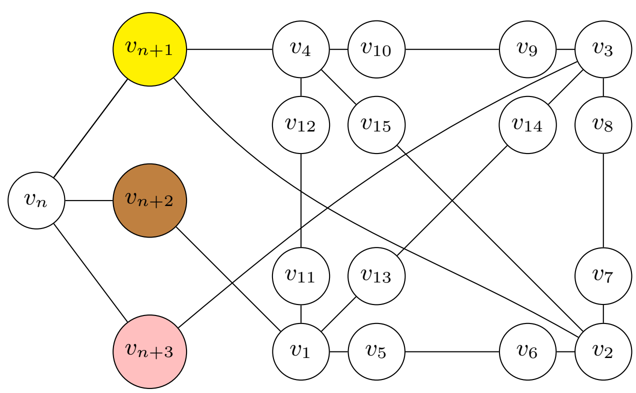

Fig. 5. A graph realisation of D constructed from the example in Figure 4 with $k = 3$. In accordance with the proof of Proposition 26, the vertex v_1 inherits the colour of vertex v_{n+2}, the vertices v_2 and v_4 inherit the colour of vertex v_{n+2}, and the vertex v_3 inherits the colour of vertex v_{n+3}.

The following theorem states that the implication in Proposition 26 is, in fact, an equivalence.

Theorem 27. *Let $k \in \mathbb{N}$, G_c be an input graph for Algorithm 23 and D be the constructed distance matrix. Then G_c is k-colourable if and only if D is a YES-instance of k-CombDMR.*

Proof. The forward direction is given by Proposition 26. It remains to prove that if G_c is k-colourable then D is a YES-instance of k-CombDMR. Let G_c have a k-colouring $\chi : V_c \to [k]$. We begin by constructing a graph realisation $(G = (V, E), \Phi)$ with $V = \{v_1, \ldots, v_{n+k}\}$ of the $n \times n$ distance matrix D. Let $\Phi(i) = v_i$ for $i \in [n]$. The edge set E of G is determined by the following requirements:

- The induced subgraph of G on the vertices $\{v_1, \ldots, v_{n_g}\}$ coincides with the gadget graph G_g.
- v_n is not adjacent to any of the vertices of the gadget graph G_g.
- For $j \in [k]$ the neighbours of v_{n+j} are precisely the following: v_n and all vertices v_i in $\{v_1, \ldots, v_{n_c}\}$ whose colour is j, that is, $\chi(v_i) = j$.

We now show that (G, Φ) is indeed a graph realisation of D. As each vertex v_i for $i \in [n_g] \setminus [n_c]$ is adjacent to some v_j for $j \in [n_c]$ in G and not adjacent to any vertex in $\{v_n, \ldots, v_{n+k}\}$, it suffices to verify the following equalities:

$$d_G(v_n, v_i) = 2 \qquad \text{for } i \in [n_c], \tag{11}$$
$$d_G(v_i, v_j) = D_{ij} \qquad \text{for } i, j \in [n_c]. \tag{12}$$

Let $i \in [n_c]$. By construction, v_i is adjacent to v_{n+j} with $j = \chi(v_i)$ and v_{n+j} is adjacent to v_n, therefore $d_G(v_n, v_i) \leq 2$ and (11) follows from the fact that v_n is not adjacent to v_i. For (12), we distinguish between two cases: if v_i and v_j are adjacent in G_c then $D_{ij} = 3$ and $d_{G_g}(v_i, v_j) = 3$ by construction. Moreover, $\chi(v_i) \neq \chi(v_j)$ implies that v_i and v_j are not adjacent to the same vertex in $\{v_{n+1}, \ldots, v_{n+k}\}$ in G. Therefore, there is no shortest path of length smaller than 3 between v_i and v_j in G. If v_i and v_j are not adjacent in G_c then $D_{ij} = 2$ and $d_{G_g}(v_i, v_j) = 2$ by construction. We do not add an edge between v_i and v_j in G, and therefore $d_G(v_i, v_j) = 2 = D_{ij}$. Hence, (G, Φ) is a graph realisation of D.

Note that k-CombDMR $\in$ NP since any distance matrix of a finite graph can be computed in polynomial-time. Therefore, Theorem 27 implies the next theorem.

Theorem 28. k-CombDMR *is NP-complete for all* $k \in \mathbb{N}, k \geq 3$.

5 Tree Realisations

In this section, we discuss the restricted case of combinatorial tree realisations of distance matrices. For a given $n \times n$ distance matrix D, we call a graph realisation (G, Φ) of D a *tree realisation* of D if G is a tree. In contrast to Proposition 5 for the general graph realisation problem, it is no longer true that any distance matrix always admits a combinatorial tree realisation with sufficiently many vertices. To see this, observe that the distance matrix of C_3 in Figure 2 can only be represented by a graph that contains a triangle, and thus is not a tree. While

C_3 admits a tree realisation in the weighted case, the distance matrix of C_4 does not have a weighted tree realisation.

Zareckiĭ shows that the tree realisation problem can be solved in $O(n^4)$ time [19] and provides a set of necessary and sufficient conditions for a distance matrix to have a tree realisation:

Theorem 29 ([19]). *Let D be an $n \times n$ matrix. Then D is a distance matrix and there exists a unique minimal combinatorial tree realisation $(T = (V, E), \Phi)$ of D if and only if*

(a) For all $i, j \in [n]$: $D_{ij} \in \mathbb{Z}, D_{ij} = D_{ji} > 0$ for all $i \neq j$, $D_{ii} = 0$.
(b) For all $i, j, k \in [n]$: $D_{ij} + D_{jk} - D_{ik}$ is even.
(c) For all $i, j, k, l \in [n]$: At least two of $D_{ij} + D_{kl}, D_{ik} + D_{jl}, D_{il} + D_{jk}$ are equal and at least the third.

For clarity, condition (c) of Theorem 29 is equivalent to

$$D_{ij} + D_{kl} \leq \max(D_{ik} + D_{jl}, D_{il} + D_{jk}),$$

for all $i, j, k, l \in [n]$, where minimal is with respect to the number of vertices in the tree.

Zareckiĭ also provides an $O(n^4)$ algorithm to construct the tree realisation of a distance matrix if it exists [19].

There exists an $O(n^2)$ algorithm to solve the minimum *weighted* tree realisation problem [5]. Provided the input distance matrix is integer valued, their algorithm can be adapted to solve the *combinatorial* tree realisation problem for integer valued distance matrices in $O(n^2)$ time.

Corollary 30. *If there is a minimal weighted tree realisation of D (containing only essential vertices) which has exclusively integer edge weights, then each edge can be replaced by an elementary path of the same length to obtain a combinatorial tree realisation of the original distance matrix. Otherwise, if this minimal weighted tree realisation (which is necessarily unique) requires a non-integer edge weight, then the distance matrix does not admit a combinatorial tree realisation.*

References

1. Althöfer, I.: On optimal realizations of finite metric spaces by graphs. Discrete Comput. Geom. **3**(2), 103–122 (1988)
2. Anirudh Sabnis, A., Sitaraman, R.K., Towsley, D.: Occam: An optimization based approach to network inference. SIGMETRICS Perform. Eval. Rev. **46**(2), 36–38 (2019)
3. Atzmon, D., Bernardini, S., Fagnani, F., Fairbairn, D.: Exploiting geometric constraints in multi-agent pathfinding. Proceedings of the International Conference on Automated Planning and Scheduling **33**(1), 17–25 (2023)
4. Chung, F., Garrett, M., Graham, R., Shallcross, D.: Distance realization problems with applications to internet tomography. Journal of Computer and System Sciences **63**(3), 432–448 (2001)

5. Culberson, J.C., Rudnicki, P.: A fast algorithm for constructing trees from distance matrices. Information Processing Letters **30**(4), 215–220 (1989)
6. Díaz, J., Öznur Yaşar Diner, Serna, M., Serra, O.: The multicolored graph realization problem. Discrete Applied Mathematics **354**, 146–159 (2024), 18th Cologne-Twente Workshop on Graphs and Combinatorial Optimization (CTW 2020)
7. Dress, A.W.M.: 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**(3), 321–402 (1984)
8. Feremans, C., Labbé, M., Laporte, G.: Generalized network design problems. European Journal of Operational Research **148**(1), 1–13 (2003)
9. Hakimi, S.L., Yau, S.S.: Distance matrix of a graph and its realizability. Quarterly of Applied Mathematics **22**, 305–317 (1965)
10. Herman, G.T., Kuba, A.: Discrete tomography: Foundations, algorithms, and applications. Springer Science & Business Media, New York City (2012)
11. Karp, R.M.: Reducibility among combinatorial problems. In: Complexity of computer computations (Proc. Sympos., IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y., 1972), pp. 85–103. The IBM Research Symposia Series, Plenum, New York-London (1972)
12. Klobas, N., Mertzios, G.B., Molter, H., Niedermeier, R., Zschoche, P.: Interference-free walks in time: temporally disjoint paths. Autonomous Agents and Multi-Agent Systems **37**(1), 1 (2023)
13. Kortsarz, G., Nutov, Z.: Approximating some network design problems with node costs. Theoretical Computer Science **412**(35), 4482–4492 (2011)
14. Lovász, L.: Coverings and coloring of hypergraphs. In: Proceedings of the Fourth Southeastern Conference on Combinatorics, Graph Theory and Computing (Florida Atlantic Univ., Boca Raton, Fla., 1973). Congress. Numer., vol. VIII, pp. 3–12. Utilitas Math., Winnipeg, MB, USA (1973)
15. Papadimitriou, C.H.: Computational Complexity. Addison-Wesley, Reading, Massachusetts (1994)
16. Semple, C., Steel, M.: Phylogenetics, Oxford Lecture Series in Mathematics and its Applications, vol. 24. Oxford University Press, Oxford (2003)
17. Stern, R., Sturtevant, N., Felner, A., Koenig, S., Ma, H., Walker, T., Li, J., Atzmon, D., Cohen, L., Kumar, S., Boyarski, E., Bartak, R.: Multi-agent pathfinding: Definitions, variants, and benchmarks. In: Proceedings of the Symposium on Combinatorial Search (SoCS) (2019)
18. Stockmeyer, L.: Planar 3-colorability is polynomial complete. SIGACT News **5**(3), 19–25 (1973)
19. Zareckiĭ, K.A.: Constructing a tree on the basis of a set of distances between the hanging vertices. Uspehi Mat. Nauk **20**(6), 90–92 (1965)