An Exact Algorithm for Finite Metric Space Embedding into a Euclidean Space when the Dimension of the Space is not Known.

Ewa Skubalska-Rafajłowicz[1][0000−0002−2795−3835] and
Wojciech Rafajłowicz[2][0000−0003−4347−1358]

[1] Dept. of Computer Engineering, ewa.rafajlowicz@pwr.edu.pl
[2] Dept. of Control Systems and Mechatronics, wojciech.rafajlowicz@pwr.edu.pl
Wroclaw University of Science and Technology, Wyb. Wyspianskiego 27, 50 370 Wrocław, Poland

Abstract. We present a $O(n^3)$ algorithm for solving the Distance Geometry Problem for a complete graph (a simple undirected graph in which every pair of distinct vertices is connected by a unique edge) consisting of $n+1$ vertices and non-negatively weighted edges. It is known that when the solution of the problem exists, the dimension of the Euclidean embedding is at most n. The algorithm provides the smallest possible dimension of the Euclidean space for which the exact embedding of the graph exists. Alternatively, when the distance matrix under consideration is non-Euclidean, the algorithm determines a subset of graph vertices whose mutual distances form the Euclidean matrix. The proposed algorithm is an exact algorithm. If the distance matrix is a Euclidean matrix, the algorithm provides a geometrically unambiguous solution for the location of the graph vertices.
The presented embedding method was illustrated using examples of the metric traveling salesman problem that allowed us in some cases to obtain high dimensional partial immersions.

Keywords: isometric embedding, Euclidean distance matrix, Euclidean distance geometry problem, rigidity of graphs

1 Introduction

In this paper, we concentrate on providing a fast algorithm for solving the Euclidean geometry problem which aims at deciding whether it is possible to find the configuration of points in the Euclidean space, such that the Euclidean distances between each pair of points match a given distance matrix. The dimension of the Euclidean space is not known in advance. Our goal, if any solution exists, is to find the point configuration in the Euclidean space of the smallest dimensionality.
The existence of a solution is directly related to the Euclidean distance matrix problem (EDM) [5], [13]. In this paper, we assume that only distance matrix $D = [d_{ij}],\ i, j \in V$ is given. Our goal is to check if D is EDM and if the answer is positive, to determine the location of vertices V in the Euclidean space of the smallest dimensionality $m \leq |V| - 1$.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_42

The problem can also be formulated equivalently in the terms of graphs. Namely, given a non-negatively weighted complete graph (defined by the distance matrix), decide whether the smallest dimension m exists and the configuration of points in R^m corresponding to the graph vertices such that Euclidean distances between all pairs of the points are equal to the edge weights.
There are known conditions that guarantee that the real solution of the Euclidean embedding problem exists [5], [18], [19], [15], [23], but verifying these conditions has a similar computational complexity as the direct solving of the problem proposed here.
For many years, other problems related to the Euclidean geometry problem have also been considered, such as Euclidean matrix completion problems [1], molecular reconstruction of chemical structures [12], [14], [15], [16] [17], sensor localization in sensor networks [4], [6], [10], [26], machine learning and dimensionality reduction [7], [25], and signal processing [11] among others.
These problems are outside the scope of this paper, but we are convinced that the algorithm proposed in this paper can also be adapted to solve many of the problems mentioned. However, this will be the subject of further research and experimentation.
The outline of the paper is as follows. Section 2 introduces the main ideas, definitions, and properties used throughout this paper. Section 3 describes the proposed embedding algorithm and discuss its computational complexity. Section 4 provides the experimental framework used to illustrate the performance of the algorithm and its possible results when the distance matrix is a non-Euclidean one. Finally, Section 5 presents our conclusions as well as some propositions for further research.

2 The Euclidean Distance Matrix Problem and Euclidean Distance Geometry Problem.

The Euclidean Distance Matrix (EDM) problem is formulated as follows. Determine whether a given matrix is Euclidean and a specific Distance Geometry Problem (DGP): given a nonnegatively weighted complete graph defined by the distance matrix, decide whether the smallest dimension m exists such that Euclidean distances between pairs of points in R^m are equal to the edge weights.
Let us denote d_{ij}^2 - a squared distance between nodes $i \in V$ and $j \in V$, by a_{ij} Thus, $A = [a_{ij}]$ is the matrix of squared distances $[d_{ij}^2]_{ij,0:n}$.
Matrix A is a squared distance matrix if and only if all elements on the diagonal of A are zero, the matrix is symmetric, i.e., $a_{ij} = a_{ji}$, $a_{ij} \geq 0$ and (by the triangle inequality)

$$\sqrt{a_{ij}} \leq \sqrt{a_{ik}} + \sqrt{a_{kj}}.$$

Any function $\psi : V \to R^m$ is an embedding of V in R^m.

Theorem 1 ([15], [22]) *A necessary and sufficient condition for the isometric embeddability of a finite metric set (V, d) of $n+1$ elements in*

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_42

an Euclidean space R^n is that the following statement be true:
The matrix $[\frac{1}{2}(a_{0i} + a_{0j} - a_{ij})]_{i,j=1:n}$ is positive definite.

There are also known conditions of embeddability based on the Cayley-Menger determinants [15], [23], but discussion of that approach is outside the scope of our paper.
An embedding is locally unique [21], i.e., we say that $p : V \to R^m$ and $q : V \to R^m$ are congruent, if

$$||p_i - p_j|| = ||q_i - q_j||$$

for all pairs $i, j \in V$.
As a consequence, if Euclidean distance matrix $D =$ is given, then any solution $\psi : V \to R^m$ such that $||\psi_i - \psi_j|| = d_{ij},\ i, j \in V$ is locally unique.
Obviously, any rigid transformation such as a translation, a rotation or a reflection or their composition does not destroy the distance structure of (V, D). More precisely, two complete graphs in the same Euclidean space are congruent if they are related by an isometry which is either a rigid motion (translation and/or rotation), or a composition of a rigid motion and a reflection [8].
A rigid graph is an embedding of a graph in a Euclidean space which is structurally rigid [9], [18]. A graph in Euclidean space R^m is said to be rigid if and only if a continuous motion of the vertices in R^m with keeping the distances between adjacent vertices unchanged preserves the distances between all pairs of graph vertices.
It is well known that every complete graph embedded in R^m is rigid in R^m [2], [3], [9].

3 An Exact Incremental Algorithm for Embedding a Complete Weighted Graph in the Euclidean Space

In this section, we will deal with the algorithmic side of the problem.
A set of $n + 1$ points in the Euclidean space spans a subspace of at most n dimensions, hence the embedding dimension considered should not exceed this number. We assume that metric distance matrix $D = [d_{ij}]$, $ij, = 0 : n$ is given. The naive approach leads to the need to solve the system of $n(n + 1)/2$ non-linear equations with a very large number of variables (at least $2n$, at most n^2):

$$||x(i) - x(j)|| = d_{ij},\ i, j = 0 : n$$

$x(i) \in R^n,\ i = 0 : n$.
It is clear that without loss of generality we can set $x(0) = (0, \ldots, 0)$.
If the system of equations is contradictory, the problem is a non-Euclidean one and an exact embedding in the Euclidean space is not possible.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_42

Analyzing the equivalent form of the previous system of equations, i.e.,

$$\sum_{l=1}^{n}(x_l(i) - x_l(j))^2 = a_{ij},\ i,j = 0:n$$

(recall that $a_{ij} = d_{ij}^2$) we can provide a computationally efficient algorithm for embedding a complete graph (a set of objects) in the Euclidean space when such exact immersion exists. Otherwise, the algorithm will stop indicating that it is impossible to preserve the currently considered set of equality constraints. The proposed approach consists of joining another vertex to an already existing partial immersion (as a new point in the Euclidean space). Next, it is necessary to check the possibility of maintaining the distance of the vertex in question, let us say k_{th}, from other $k-1$ vertices already located in the Euclidean space, so as not to violate the values contained in matrix D. It may possibly require increasing the current dimension of the Euclidean space by one.
In the end, the algorithm provides the dimension number of the embedding and the locally unique embedding of V , or stops when exact embedding is impossible. The exact embedding does not depend on the vertex chosen as a starting point.
We begin analysis with a set of three vertices, let say, vertices labeled by $0, 1$ and 2. Due to the triangle inequality, it is obvious that these vertices can be embedded in R^2 and this embedding is locally unique. Vertex-representing points form a triangle and the shape of this triangle is unique. So, without loss of the generality we can locate vertex v_0 in $x(0) = (0,0)$, v_1 in $x(1) = (d_{01}, 0)$, and v_2 in $x(2) = (x_1(2), x_2(2))$, where $x(2)$ coordinates are obtained by solving the following system of equations:

$$(x_1(2) - x_1(0))^2 + (x_2(2) - x_2(0))^2 = a_{02}, \tag{1}$$

$$(x_1(2) - x_1(1))^2 + (x_2(2) - x_2(1))^2 = a_{12}. \tag{2}$$

Since $x(0) = (0,0)$, (1) simplifies to

$$x_2^2(1) + x_2^2(2) = a_{02} \tag{3}$$

and (2) takes the form

$$(x_2(1) - d_{01})^2 + x_2^2(2) = a_{12}.$$

Substracting the first equation from the second one, we obtain a linear equation:

$$-2d_{01}x_2(1) + a_{01} = a_{12}.$$

Thus, $x_{21} = -0.5(a_{12} - a_{01})/d_{01}$, and consecutively

$$x_2^2(2) = a_{02} - \frac{(a_{12} - a_{01})^2}{4a_{01}}.$$

Due to the triangle inequality,

$$a_{02} - \frac{(a_{12} - a_{01})^2}{4a_{01}} \leq 0.$$

There exist at most two solutions of the system with $x_2(2)$ being a positive or a negative real number. When points $x(0)$, $x(1)$, $x(2)$ are collinear, $x_2(2) = 0$ and the embedding dimension is $m = 1$. As a consequence, all coordinates $x_2(\cdot)$ can be neglected and may be removed.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_42

3.1 Algorithm of the Euclidean embedding of a 3-vertex structure

Algorithm 1.

1. Step B1. Set $x(0) = 0.$, $x(1) = d_{01}$ and dimension number $m(1) = 1$.
2. Step B2. Compute $\Delta = a_{02} - \frac{(a_{12}-a_{01})^2}{4a_{01}}$ If $\Delta = 0.$ set $x(2) = (0.5(a_{12} - a_{01})/d_{01})$ and $m(2) = 1$. Otherwise set $m(2) = 2$, $x(0) = (0., 0.)$, $x(1) = (d_{01}, 0.)$, and $x(2) = (-0.5(a_{12} - a_{01})/d_{01}, \sqrt{\Delta})$.

The proposed approach can generalized for larger graphs and larger dimensions. Expanding the set of the immersed vertices one by one leads to the incremental embedding algorithm.

3.2 General Location Algorithm

Locating a new vertex from V in the Euclidean space leads to the following problem:

Problem 1 *We assume that we have given coordinates of $r + 1$ points representing an exact embedding of $V_r \subset V$, $|V_r| = r + 1$ in a Euclidean space, i.e., $(x(0), x(1), \ldots x(r)) \in R^{m(r)}$, where $m(r)$ is a dimension of the Euclidean space, and $m(r) <= r$. Thus, that we have:*

$$||x(i) - x(j)||^2 = a_{ij},\ i,\ j \in 0 : r.$$

Find a vector of dimensionality at most $m(r) + 1$ representing a new selected vertex, let us say, $v_{r+1} \in V - V_r$ that does not violate distances from D between all vertices in $V_r \cup v_{r+1}$.

Thus, our goal is to find

$$x(r + 1) = (x_1(r + 1), x_2(r + 1), \ldots , x_{m(r)+1}(r + 1))$$

such that

$$||x(i) - x(r + 1)||^2 = a_{i,r+1},\ i = 0, 1, \ldots , r, \tag{4}$$

where the dimension of all points in $X_r = [x(i)]_{i=0:r}$ is expanded to $m(r) + 1$, and the $(m(r) + 1)$-th coordinates of all vectors in X_r are set to zero. If $x(r + 1) \in R^{m(r)+1}$ exists and $x_{m(r)+1}(r + 1) = 0$ the system of $r + 2$ points lays in the m dimensional quotient space, i.e., $x(0), x(2), \ldots x(r), x(r+1) \in R^s$. All $m(r)+1$ zero coordinates should be removed and $m(r + 1) = m(r)$. Otherwise, when $x_{m(r)+1}(r + 1) \neq 0$, the new dimension of the embedding Euclidean space is $m(r+1) = m(r)+1$. The lack of any real solution means that the metric d is not a Euclidean metric and an exact immersion in the Euclidean space does not exist.
Taking into account the fact that the number of non-zero coordinates successively considered vertices v_i gradually increase and is equal to the previously determined values of $m(i)$, we can rewrite (4) as a system of linear equations:

$$x(r + 1)x^T(i) = \sum_{j=1}^{m(i)} x_j(i)x_j(r + 1) = \tag{5}$$

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_42

$$\frac{1}{2}[a_{0,i} + a_{0,r+1} - a_{i,r+1}],\ i = 1 : r$$

supplemented by a nonlinear equation of the form $||x(r+1)||^2 = a_{0,r+1}$. Notice, that in (5) we have replaced $||x(r+1)||^2$ by $a_{0,r+1}$ and $||x(i)||^2$ by $a_{0,i}$.
If the solution of the linear system (2) exists, but it is such that $||x(r+1)||^2 \neq a_{0,r+1}$ there is no real solution of (4) and the exact embedding of $V(r+1)$ in the Euclidean space is not possible.
To better illustrate the essence of the proposed algorithm, let us first show a simple example.

Example Let $r = 2$, $m(1) = 1$, and $m(2) = 2$ then (2) is of the form

$$x_1(3)x_1(1) = [a_{0,1} + a_{0,3} - a_{1,3}]/2,$$

$$x_1(3)x_1(2) + x_2(3)x_2(2) = [a_{0,2} + a_{0,3} - a_{2,3}]/2.$$

Using results of Step B2, i.e., $x(1) = (d_{01}, 0.)$, and $x(2) = (-0.5(a_{12} - a_{01})/d_{01}, \sqrt{\Delta})$, we can obtain values of $x_1(3),\ x_2(3)$ and check if $x_1^2(3) + x_2^2(3) = a_{0,3}$. Alternatively, after computing $x_1(3)$ from the first equation, one can obtain $x_2^2(3) = a_{0,3} - x_1^2(3)$. It is obvious that when $a_{0,3} - x_1^2(3) < 0$ the real solution of the problem does not exist.
The system of equations (5) can be written as:

$$\mathcal{X} x^T(r+1) = b, \tag{6}$$

where $x(r+1) = (x_1(r+1), \ldots, x_{m(r)}(r+1))$, $b = \frac{1}{2}[a_{0,i} + a_{0,r+1} - a_{i,r+1}]_{i=1:r}$ and $\mathcal{X} = [x(i)]_{i=1:r} \in R^{r \times m(r)}$ is a triangle-like matrix.

Algorithm

1. Start with Algorithm 1. We assume that previously computed values of $m(1), \ldots, m(r)$ are known, and available. Additionally $m(0) = 0$.
2. **Step 1. Solving (5):**
3. **for** $i \leftarrow 1$ **to** r **do**
   set $m = m(i)$.
4. **if** $m = m(i-1) + 1$ compute

$$x_m(r+1) = \frac{(a_{0,i} + a_{0,r+1} - a_{i,r+1})}{2x_{m(i)}(i)} x - \frac{\sum_{j=1}^{m(i)-1} x_j(i) x_j(r+1)}{x_{m(i)}(i)}$$

   and go to Step 2.
5. **else** (i.e., when $m = m(i-1)$) recalculate $x_m(r+1)$ as

$$x_m^{new}(r+1) = \frac{(a_{0,i} + a_{0,r+1} - a_{i,r+1})}{2x_{m(i)}(i)} - \frac{\sum_{j=1}^{m(i)-1} x_j(i) x_j(r+1)}{x_{m(i)}(i)}.$$

   if $|x_m(r+1) - x_m^{new}(r+1)| > 0$
   then system of equations (5) is not consistent. Stop and provide adequate information.
6. **else** (i.e., when $|x_m(r+1) - x_m^{new}(r+1)| = 0$) **end**

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_42

7. **Step 2. Checking feasibility:**
8. Compute

$$a = \sum_{j=1}^{m(r)} x_j^2(r+1).$$

9. **if** $a > a_{0,r+1}$,
   then full embedding does not exist. Stop and give adequate information.
   else
   if $a < a_{0,r+1}$
   then set $m(r+1) = m(r)+1$ and $x_{m(r+1)}(r+1) = \sqrt{a_{0,r+1} - a}$,
   else (when $a = a_{0,r+1}$) **then** set $m(r+1) = m(r)$.
10. **end**

System (5) can be over-determined (when $m(r) < r$) but it is consistent when exact embedding exists.

Notice that if two subsequent, previously embedded vertices, let us say v_i and v_{i+1}, have the same embedding dimension (i.e., $m(i) = m(i+1)$) then two subsequent equations from the system (5) are linearly dependent or inconsistent. In the second case the solution does not exist. The presented algorithm exploits the triangle-like structure of $\mathcal{X}$.

Thus, it allows us to solve (5) performing at most $O(r^2)$ arithmetic operations. Using the ordinary least square method to solve (5) leads to $O(r^3)$ local complexity.

Summarizing, the algorithm of embedding V is as follows:

Given a $n+1 \times n+1$ matrix of squared distances A the algorithm provides the dimension of the embedding $m(n)$ and vector's coordinates $\mathcal{X} = [x(i)]_{i=1:n}$ (recall that $x(0)$ is the zero vector) or information that exact embedding does not exist. Additionally, a partial embedding of the first k vertices $(3 < k < n+1)$ is given.

It should be emphasized that the computational complexity of embedding of the whole set of $n+1$ vertices is in the worst case $O(n^3)$. Using the least square method to solve a local system of equations (5) increases the total complexity to $O(n^4)$ [24].

4 Computational Experiments

The proposed algorithm was extensively tested with problems from the well-know TSPLIB. The TSPLIB proposed by [20] (`http://elib.zib.de/pub/mp-testdata/tsp/`) is a typical set of benchmarks containing 111 different symmetric traveling salesman problems (TSP) problems.

We are investigating two groups of problems:

- symmetric problems with Euclidean 2D distances (`EUC_2D`),
- symmetric problems with explicit weights in the form of distance matrices (`MATRIX`).

As a method of verification of the concept and its accuracy some of the 61 problems of type `EUC_2D` (two-dimensional problems where coordinates of the cities are known and the distance is calculated as a simple Euclidean norm) were converted to distance matrices. We know that the exact

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_42

dimension is 2. By this reversed method we can easily verify the accuracy of the result.

Annotation MATRIX indicates that the distances are supplied in the form of matrices. No other information is available. As a rule, such matrices are not Euclidean distance matrices. Such data allow us to obtain some partial Euclidean immersions of the TSP vertices.

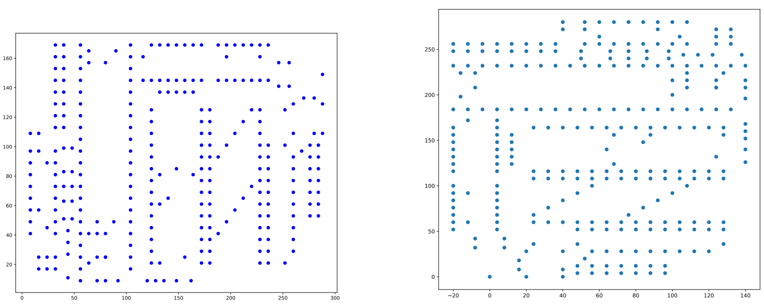

Fig. 1. Original (first) and reconstructed (second) layout of the `a280` problem from tsplib.

In Fig. 1 we can see the results of the reconstruction in the comparison to the original data `a280` problem from tsplib. It is easy to observe that the resulting image is accurate with respect to rotation and mirroring. In the Figure the reconstruction of the `bier127` problem is visualized. In both pictures the vertices (cities) are connected according to the optimal tour. Further, some metric problems, namely `si175`, `si535` and `si1032`, were examined. Distance matrices are not EDM, as shown by the proposed algorithm. Nevertheless, large parts of the corresponding graphs can be precisely (without any distortion) embedded in the separate Euclidean spaces of large dimensionality.

The experiments were performed in the following way. After the information about the inability to immerse the next vertex in the lastly indicated Euclidean space appeared, the procedure was interrupted and the next subset of vertices was generated. The last rejected vertex was taken as the starting point of the algorithm. A randomly generated vertex was selected as the starting node, and then the order of considering other vertices was established according to the nearest neighbor rule.

In the `si175` problem, 5 subsets of nodes immersed in separate Euclidean spaces of dimension 48, 58, 33, 2, and 29, respectively, and composed of 49, 59, 34, 3, and 30 vertices, was obtained. The `si535` problem has been divided into 8 subgraphs immersed in separate Euclidean spaces of dimension 14, 97, 116, 2, 133, 126, 2 and 37, respectively. Finally, the `si1032` problem was decomposed by the algorithm into 16 separate EDM

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_42

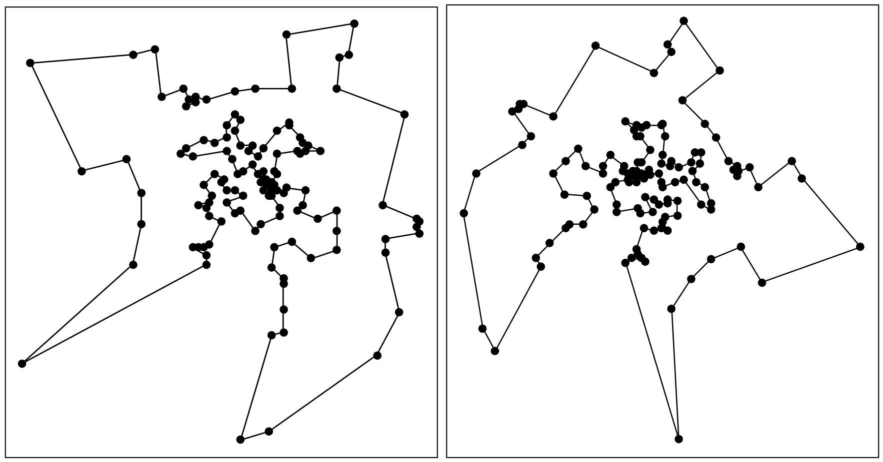

Fig. 2. Original (first) and reconstructed (second) layout of the `bier127` problem from tsplib.

cliques consisting of 109, 80, 80, 80, 69, 83, 39, 54, 50, 45, 127, 54, 53, 46, 54, and 9 nodes.

5 Concluding Remarks

However, the algorithm presented in this paper can also be used as a method of computing an approximate solution of the embedding problem, the global optimization approach based on minimization of the sum of squared distance distortions (see (7)), examined in [14], provides better approximations from the point of view of the mean squared error criterion minimization.
Unfortunately, the global distortion error defined as

$$\sum_{i=0}^{n-1} \sum_{j=i+1}^{n} \left(||x(i) - x(j)||^2 - a_{ij} \right)^2 ,$$

or as

$$\sum_{i=0}^{n-1} \sum_{j=i+1}^{n} |||x(i) - x(j)||^2 - a_{ij}|^2, \qquad (7)$$

is not easy to minimize, when exact embedding does not exist.
The algorithm presented in the paper allows us to easily impose upper bounds on subsequent distortion errors of individual distances (both absolute and relative errors). However, this will be the subject of further research.
Summarizing, the proposed algorithm provides the smallest dimension number of the embedding and the configuration of points in the Euclidean space, such that the Euclidean distances between each pair of points match a given distance matrix. Alternatively, if the matrix is a non-Euclidean one, only a partial solution is generated.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_42

In such a case the partial embedding depends on the vertex selected as a starting point. The problem of partial embedding of the finite metric set is combinatorial in nature and can be formulated in many different ways. In our opinion, it is a new class of open problems that is worthy of further research.

References

1. Alfakih A., Khandani A., Wolkowicz H.: Solving Euclidean distance matrix completion problems via semidefinite programming, Comput. Optim. Appl., 12, 13–30 (1999).
2. Asimow L., Roth B.: The rigidity of graphs. Trans. Amer. Math. Soc., 245, 279–289 (1978).
3. Barvinok A.: Problems of distance geometry and convex properties of quadratic maps. Discrete and Computational Geometry. 13, 189–202 (1995).
4. Biswas P., Lian T., Wang T., Ye Y.: Semidefinite programming based algorithms for sensor network localization. ACM Transactions on Sensor Networks. 2 (2): 188–220 (2006).
5. Blumenthal L.: Theory and Applications of Distance Geometry, Oxford University Press, Oxford, (1953).
6. Cao M., Anderson B. D.O., Morse S.: Sensor network localization with imprecise distances, Systems and Control Letters 55, 887–893 (2006).
7. Courrieu P.: Straight monotonic embedding of data sets in Euclidean spaces. Neural Networks 15, 1185–1196 (2002).
8. Coxeter, H. S. M.: Introduction to Geometry, Second edition. Wiley (1969).
9. Crapo H.: Structural rigidity, Structural Topology, 73 (1), 26–45, (1979).
10. Ding Y., Krislock N., Qian J., Wolkowicz H.: Sensor network localization, Euclidean distance matrix completions, and graph realization. Optim. Eng., 11(1) 45–66, (2010).
11. Dokmanic I., Parhizkar R., Ranieri J., Vetterli M.: Euclidean Distance Matrices: Essential theory, algorithms, and applications. IEEE Signal Processing Magazine 32 (6) 12– 30 (2015) https://doi.org/10.1109/MSP.2015.2398954.
12. Dong Q., Wu Z.: A linear-time algorithm for solving the molecular distance geometry problem with exact inter-atomic distances, Journal of Global Optimization 22, 365–375 (2002).
13. Gower J.: Euclidean distance geometry, Math. Sci., 7 , 1–14, (1982).
14. Lavor C.: On generating instances for the molecular distance geometry problem. In: Global Optimization: From Theory to Implementation, L. Liberti and N.Maculan, eds., Springer, Berlin, 405–414 (2006).
15. Liberti L., Lavor C., Nelson Maculan N., Mucherino A., Euclidean Distance Geometry and Applications. SIAM Review, 56(1), 3–69 (2014).

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_42

16. Liberti L., Lavor C.: Euclidean Distance Geometry. An Introduction. Springer International Publishing Switzerland (2017)
17. Liberti L.: Distance geometry and data science. TOP 28, 271–339 (2020).
18. Maehara H.: Euclidean embeddings of finite metric spaces. Discrete Mathematics 313, 2848–2856 (2013).
19. Menger K.: New foundation of Euclidean geometry. Amer. J. Math., 53, 721–745 (1931).
20. G. Reinelt.: TSPLIB – A Traveling Salesman Problem Library. ORSA Journal on Computing, 3(4), 1991, 376–384 (1991).
21. Saxe J.B.: Embeddability of weighted graphs in k-space is strongly NP-hard. Proc. of the 17th Allerton Conference in Communications, Control, and Computing 480–489 (1979).
22. Schoenberg, I. J.: Metric spaces and positive definite functions. Transactions of the American Mathematical Society, 44, 522–536 (1938).
23. Sippl M., Scheraga H.: Cayley-Menger coordinates. Proc. Natl. Acad. Sci. USA, 83, 2283–2287 (1986).
24. Stroeker, R. J.: On the sum of consecutive cubes being a perfect square. Compositio Mathematica. 97(1-2), 295–307 (1995).
25. Tenenbaum J.B., De Silva V., Langford J.C.:, A global geometric framework for nonlinear dimensionality reduction. Science 290(5500) 2319–2323 (2000).
26. Tasissa A., Lai R.: Exact Reconstruction of Euclidean Distance Geometry Problem Using Low-Rank Matrix Completion, IEEE Transactions On Information Theory, 65(5), 3124–3144 (2019).

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_42