Published in European Journal of Operational Research, vol. 174, no. 1, 2006, p. 102-111
http://dx.doi.org/10.1016/j.ejor.2005.02.071

A constructive algorithm for realizing a distance matrix

Sacha C. Varone [1]

LARIM - Department of Computer Engineering, École Polytechnique de Montréal, (Canada)

Abstract

The natural metric of a weighted graph is the length of the shortest paths between all pairs of vertices. The investigated problem consists in a representation of a given metric by a graph, such that the total length of the graph is minimized. For that purpose, we give a constructive algorithm based on a technique of reduction, fusion and deletion. We then show some results on a set of various distance matrices whose optimal realization is known.

Key words: metrics, reduction, heuristics
1991 MSC: 68R05, 05C10, 05C12, 05C50, 05C62

1 Introduction

The problem of realizing a distance matrix by a graph of minimal total length is a difficult problem. A lot of authors have considered the special case where the distance matrix is a tree metric, also called an additive metric (see for example [1–14]). Efficient algorithms that construct such trees were published in [15] or [16]. But real data describe merely a tree metric because they arise from a similarity measure that includes errors. Such a measure appears in various fields such as the study of evolution [17–20], the synthesis of certain electrical circuits [21], the seismic tomography, the traffic modelling [22,23] or the analysis of memory [10]. Therefore the interest was focus on approximating a metric by a tree metric [24–33] or finding the greatest sub-metric that is realizable by a tree [34].

Email address: sacha.varone@polymtl.ca (Sacha C. Varone).
[1] This work has been partially funded by grant PA002-104974/1 from the Swiss National Science Foundation.

In this paper we consider the problem of realizing a symmetric distance matrix by a weighted graph. We recall that a weighted graph $G = \langle V, E, w \rangle$ realizes the distance matrix D of order n if and only if $\{1, ..., n\} \subseteq V$ and $d_{ij}^G = d_{ij}$ $\forall i, j = 1, ..., n$, where d_{ij}^G is the length of a shortest path in G between the vertices i and j. The vertices in the set $\{1, ..., n\}$ are called external and the vertices in $V \setminus \{1, ..., n\}$ are called internal.

We use a heuristic to solve the problem since it has been proven to be NP-hard [35,36]. In the next section we describe the algorithm and in the last section we give some numerical experiments and make some remarks on the results.

2 Algorithm

We give below the algorithm that we have developed. Each step will be explained in the remaining of this section.

Algorithm
(1) Repeat
    Compactify
    Reduct
  Until no compaction is possible
(2) Greedy algorithm
(3) Triangles'reduction
(4) Fusion
(5) Remove useless edges/vertices
(6) Decompaction/augmentation phase

The complexity of this algorithm is polynomial in $O(n^4)$ (in the worst case) where n is the number of entries in the distance matrix.

2.1 Compaction and reduction phase

This phase will be applied on the distance matrix itself. The aim is to simplify as much as possible the data. It is a preprocessing step.

Let D be a distance matrix of order $n \times n$ and let E_i be the matrix where

$$(E_i)_{jk} = \begin{cases} 1 & : \quad j = i \neq k \\ 1 & : \quad j \neq i = k \\ 0 & : \quad \text{elsewhere} \end{cases}$$

Let $D_i(a) = D - a * E_i$ as in [1].

$$D_i(a) = \begin{pmatrix} d_{11} & \cdots & d_{1n} \\ \vdots & & \vdots \\ d_{n1} & \cdots & d_{nn} \end{pmatrix} - a * \begin{array}{c} \\ \\ \\ i^{th}\,row\{ \\ \\ \\ \end{array} \begin{array}{c} \\ \\ \\ \\ \\ \\ \end{array} \begin{array}{c} \overbrace{\begin{matrix} 1 \\ \vdots \\ 1 \\ 0 \\ 1 \\ \vdots \\ 1 \end{matrix}}^{i^{th}column} \end{array} \begin{pmatrix} & 0 & & 0 \\ & & & \\ 1 \cdots 1 & & 1 \cdots 1 \\ & & & \\ & 0 & & 0 \end{pmatrix}$$

The question that arises then is : "How to choose the number a such that $D_i(a)$ is still a distance matrix ?" The answer is in the following theorem.

Theorem 1 *[1]$D_i(a)$ is a distance matrix $\Leftrightarrow a \leq \frac{1}{2}(d_{pi} + d_{ir} - d_{pr})\ \forall p, r \neq i$*

The new metric $D_i(a)$ obtained from D is called a *compaction* [37]. The compaction of an index i of D leads to a new matrix with a possible pair of equal rows (and by symmetry of equal columns). By deleting one of these equal rows and columns we obtain a new distance matrix whose order is one unit lower. This new matrix is called a *reduction* of D.

Define $a_0(i)$ as the maximal value such that $D_i(a_0(i))$ is a metric and let a be smaller or equal to $a_0(i)$. $G_i(a)$ is a realization of $D_i(a)$ with its external vertices denoted by $v_1^{'}, \cdots, v_n^{'}$. To construct a realization G of D from $G_i(a)$ it is sufficient to add a vertex v_i to $G_i(a)$ and connect v_i to $v_i^{'}$ with an edge whose weight is equal to a. This corresponds to step 6 of our algorithm.

Theorem 2 *[1]If $0 \leq a \leq a_0(i)$ and if $G_i(a)$ is an optimum realization of $D_i(a)$ then G obtained from $G_i(a)$ is an optimum realization of D.*

The first step towards searching for a realization of a distance matrix D is to compact and to reduce it (if possible) iteratively so that the new matrix is still a metric. Note that, as mentioned in [37], if a distance matrix is tree-realizable then the iteration of the above process leads to a matrix of order 1.

This phase requires for each entry of the distance matrix to find two other entries p and r that minimize the quantity $d_{pi} + d_{ir} - d_{pr}$. The latter quantity is $a_0(i)$. This is done in a time complexity of $\Theta(n^2)$. Moreover the number of time that this compaction process has to be done is at most $\lfloor \frac{n}{2} \rfloor$ since it

is the number required by a simple path on n vertices. So the overall time complexity for this phase is $\Theta(n^4)$

From now on we will consider as a data a distance matrix of order $N \times N$.

2.2 Greedy algorithm

A distance matrix D of order $N \times N$ is trivially realized by the complete graph with for each edge a weight equals to the distance in D. A first realization is obtained as follow :

Greedy algorithm
(1) Start from a graph $G = (V, \emptyset)$
(2) Sort the distances by a non decreasing order in a list L
(3) For each $d_{i,j} \in L$
    if[a] $d^G_{i,j} > d_{i,j}$ add edge (i, j) of weight $d_{i,j}$

[a] if no path exists between i and j, set $d^G_{i,j} = \infty$

In the case that the graph so far obtained is a polygon, it is also an optimal realization. This is formalized by the underlying theorem :

Theorem 3 *[38]Let D be realized by a polygon on at least four vertices denoted $1, 2, \cdots, n$. This realization is unique and optimal if and only if $\forall i$ mod(n) $d(i-1, i) + d(i, i+1) = d(i-1, i+1)$*

The greedy algorithm works in a time complexity of $O(e \log(e))$ where e is the number of distances, i.e. $O(N^2 \log(N))$

We observe that if a distance matrix can be realized by an unicyclic graph with (pendant) trees attached to the vertices of the cycle, then the following algorithm leads to the optimal realization :

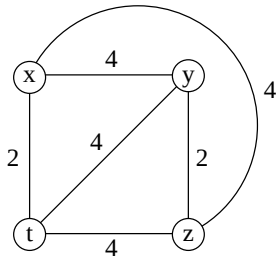
Algorithm
(1) Repeat
        Compactify
        Reduct
    Until no compaction is possible
(2) Greedy algorithm
(3) Decompaction/Augmentation phase

In this last algorithm step 1 ensures to get a reduced matrix whose realization is an unicyclic graph and step 2 constructs this graph.

Example 4 *The greedy algorithm applied to the below distance matrix gives*

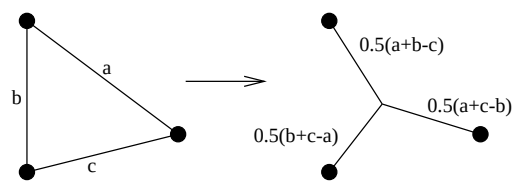
the complete graph.

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



2.3 Triangles'reductions

The purpose of this step is to simplify all the triangles produced by the greedy algorithm. The general following transformation is applied.



We start from a realization $G = (V, E, w)$ obtained with the greedy algorithm, with V as vertex set, E as edge set, and w a weight function on the edges. Since there is no optimal realization containing as a subgraph a complete graph on three vertices [1] , denoted by K_3 and called a triangle, we can look for existing K_3 in G and transform them into a 3-star (a star with 3 branches).

<u>K_3 'reduction</u>
(1) Find the list L of all K_3 in G
(2) For each $K_3 \in L$
    If K_3 still exists in the current realization
        Transform it into a 3-star

This algorithm works in the worst case in $O(N^3)$.

Let K_3^1 and K_3^2 be two complete graphs with three vertices in the realization G such that $K_3^1 \cap K_3^2 \neq \emptyset$. The transformation of K_3^1 into a 3-star deletes one edge of K_3^2 and we have two triangles less in the realization G.

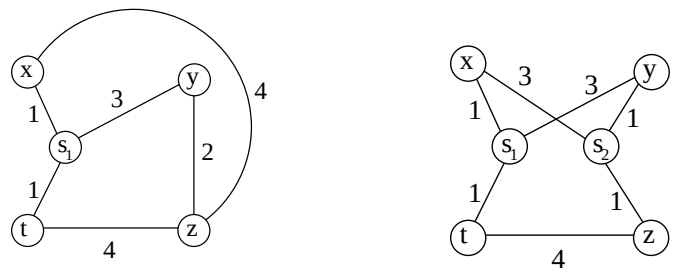
Nevertheless we can say that the deleted edge is a virtual edge in the transformed graph since there exists a shortest paths in G of length given by the distance matrix. If the transformation of this "virtual K_3", namely K_3^2, leads to a better realization, then we make it effective. We could therefore try to transform all paths formed on triplets of external vertices of G. The time

complexity is again in $O(N^3)$.

The first triangle reduction applied to the previous example gives:

Considering the "virtual K_3" on the vertices x, y and z, we get:

Example 5



2.4 Fusion

When applying the above mentioned transformation, a special topology can occur : two internal vertices can be linked to two common vertices and some other vertices. In Figure 1 below, the internal vertices s_1 and s_2 are linked to the external vertices x_1 and x_2. The vertices v_1 and v_2 are assumed to be external.

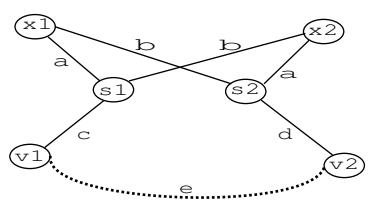


Fig. 1. Topology T_1

We assume that if the graph G contains such a topology T_1, there exists another path linking v1 to v2 whose length e is such that $e \leq c + (b - a) + d$. Let also assume (wlog) that a $\leq$ b.

We consider the transformation of this topology T_1 into T_2 described below:

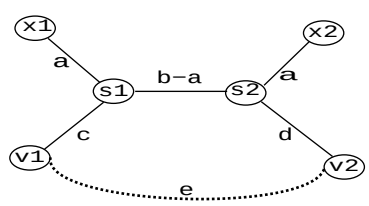


Fig. 2. Topology T_2

The result is a fusion between two edges into another one. In the case that

$a = b$ then s_1 and s_2 are reduced into one single vertex (fusion of vertices). We give a justification of that transformation :

Lemma 6 *If G is a realization of a metric D then G' obtained by the above transformation is also a realization of D.*

Proof

We have to show that the length of a shortest path in T_1 between two external vertices is not modified by this transformation.

We denote by $P^{T_i}(x, y)$ the length of the shortest path between x and y in the topology T_i. We have

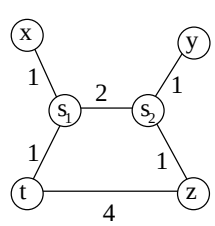
- $P^{T_2}(x_1, x_2) = a + (b - a) + a = a + b = P^{T_1}(x_1, x_2) \geq D(x_1, x_2)$.
- $P^{T_2}(x_1, v_1) = a + c = P^{T_1}(x_1, v_1) \geq D(x_1, v_1)$.
- $P^{T_2}(x_1, v_2) = a + (b - a) + d = b + d = P^{T_1}(x_1, v_2) \geq D(x_1, v_2)$.
- By symmetry, equalities hold when x_1 is replaced by x_2.
- $P^{T_2}(v_1, v_2) = c+(b-a)+d \geq e = D(v_1, v_2)$ by our assumption. This means that between v_1 and v_2 there exists a path whose length is $e = D(v_1, v_2)$.

$\sharp$

Remark 1 *The transformation can be generalized to the case of internal vertices s_1 and s_2 of degree greater or equal to three. Again, the neighbors of s_1 and s_2 are assumed to be external. The condition under which this transformation remains consistent is : $P^{G'}(u, v) \geq P^G(u, v)$ $\forall u \in N(s_1)$ and $\forall v \in N(s_2)$ where $N(s_i)$ denotes the set of neighbors of s_i, $i = 1, 2$ and $P^G(u, v)$ is the length of a shortest path in G between u and v.*

The cost in time of such a transformation is a constant. But the search for such a topology could be very expansive if there exists a lot of internal vertices to be checked. The triangles'reductions gives at most $\binom{N}{3}$ internal vertices. So the number of occurrences of topology T_1 is bounded by $\binom{\binom{N}{3}}{2}$ and hence the fusion step is polynomial in $O(N^6)$. Since this could be too much time consuming in practice, this phase has been applied a number of time proportional to the number of internal vertices. Therefore this fusion step runs in $O(N^3)$.

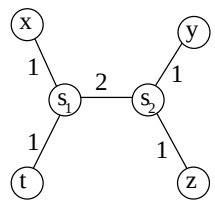
Example 7 *We go on with our example and apply the fusion step. The realization obtained so far is :*



2.5 Deletion

Once the fusion step has been applied the current realization can contain useless edges and/or vertices. These edges have to be deleted in order to reduce the total length of the realization. This could be done by an updating rule which runs in the worst case in $O(N^3)$. As each edge has to be checked, the total running time for this step would be in $O(N^9)$. Since some experiments of our algorithm on random distance matrices have shown that the deleted edges (u, v) are (almost) always such that u belongs to the neighbors of s_1 and v belongs to the neighbors of s_2, we have restricted the deletion phase to the set of edges $\{(u, v) \mid u \in N(s_1),\ v \in N(s_2)\}$. Hence the deletion phase runs in $O(N^2)$.

Example 8 *Finally, in our example, the edge linking z and t could be deleted. We get :*



2.6 Decompactication/Augmentation

The last step of our algorithm is to get a realization of the original distance matrix. This can easily be done in at most $O(N)$ iterations with the method described in 2.1.

3 Numerical experiments

There is a lack of numerical results in the literature for the problem we have solved. Therefore we have produced our own data. We tested our algorithm on several distance matrices induced by a given topology. The appendix contains optimal realizations whose induced metrics were applied to our algorithm.

We indicate in the first column the name of the data file. Columns 2 to 4 shows the results without the compaction phase. Column 2 indicates the weight of the realization constructed with the greedy algorithm. Then we applied the K_3 reduction (column 3) and the "virtual K_3" reduction (column 4). We present in column 5 the result obtained with the compaction phase. The last column

shows the weight of an optimal realization.

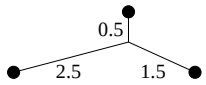
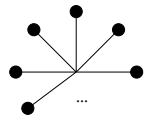
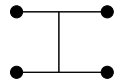
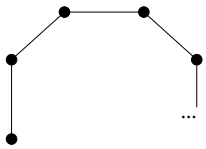
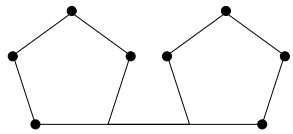
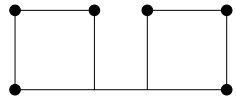
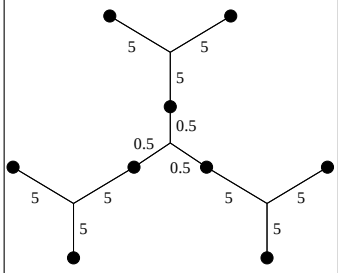
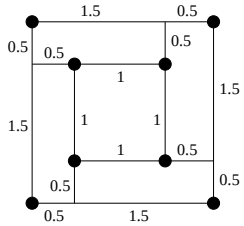
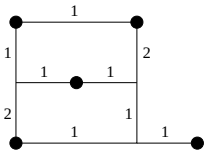
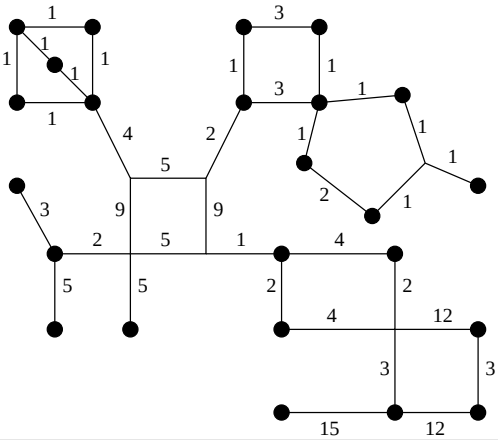
Topology	Greedy	K_3	"Virtual K_3"	Compact	Optimal
triangle	9	4.5	4.5	4.5	4.5
star4	12	8	4	4	4
star5	20	12	10	5	5
star6	30	20	14	6	6
H	5	5	5	5	5
4triangles	93	46.5	46.5	46.5	46.5
4nontree	20	13	10	8	8
PolygonN	n	n	n	n	n
2pentagons	22	16	11	11	11
2squares	16	16	9	9	9
Cube2	16	16	14	14	14
Hakimi64	18	13	13	11	11
Imrich84	250	192	156	130	128

Data used in these experiments are not extensive but focus on different topologies. We can observed that the constructive algorithm we have presented in this paper works very well with various kinds of topologies. Our algorithm can be improved, especially at step 4 where one could find other types of topologies.

Our goal was to propose a method to tackle the problem of realizing a distance matrix by a weighted graph. As far as we know, this method belongs to the very few ones which solve the problem of realizing a (non-tree) distance matrix by a graph.

A Appendix

Unless explicitely given, all edges are assumed to have a unit length. The metric is induced by the black circles.

triangle 	starN (N=4,5,6) 	H 
PolygonN : polygon on N vertices 	2pentagons 	2squares : an example presented by Simoões-Pereira in [16] 
4triangles 	cube2 	Hak64 : an example presented by Hakimi and Yau in [1] [2] 
Imr64 : an example presented by Imrich, Simoões-Pereira and Zamfirescu in [39] 		

[2] In [1] the graph is slightly different since it is not optimal

References

[1] S. L. Hakimi, S. S. Yau, Distance matrix of a graph and its realizability, Quart. Appl. Math. 22 (1964) 305–317.

[2] K. A. Zareskii, Constructing a tree on the basis of a set of distances between the hanging vertices, Uspekki Mat. Nauk. 20 (1965) 90–92.

[3] F. T. Boesch, Properties of the distance matrix of a tree, Quat. Appl. Math. 26 (1969) 607–609.

[4] J. M. S. Simões-Pereira, A note on the tree realizability of a distance matrix, J. Combin. Theory 6 (1969) 303–310.

[5] A. N. Patrinos, S. L. Hakimi, The distance matrix of a graph and its tree realization, Quart. Appl. Math. 30 (1972) 255–269.

[6] P. Buneman, A note on metric properties of trees, J. Combin. Theory Ser. B 17 (1974) 48–50.

[7] S. Sattath, A. Tversky, Additive similarity trees, Psychometrika 42 (1977) 319–345.

[8] W. Imrich, On metric properties of tree like spaces, in: Contributions to Graph Theory and its Applications, Technische Hochschule Illmenau, Illmenau, 1977, pp. 129–156.

[9] L. Graham, R. L.and Lovasz, Distance matrix polynomials of trees, Advances in Mathematics 29 (1978) 60–88.

[10] J. A. Cunningham, Free trees and bidirectional trees as representations of psychological distance, J. Math. Psychol 17 (1978) 165–188.

[11] W. Imrich, G. Schwarz, Trees and length functions in groups, Ann. Discrete Math. 17 (1983) 347–359.

[12] A. 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 (1984) 321–402.

[13] F. Murtagh, Counting dendrograms: a survey, Discrete Applied Mathematics 7 (1984) 191–199.

[14] J. E. Corter, A. Tversky, Extended similarity trees, Psychometrika 51 (1986) 429–451.

[15] J. C. Culberson, P. Rudnicki, A fast algorithm for constructing tree from distance matrices, in: Inf. Process. Lett., Vol. 30, 1989, pp. 215–220.

[16] J. M. S. Simões-Pereira, An algorithm and its role in the study of optimal graph realizations of distance matrices, Discrete Mathematics 79 (1989/90) 299–312.

[17] B. Mau, M. A. Newton, Phylogenetic inference for binary data on dendograms using markov chain monte carlo, Journal of computational and graphical statistics 6 (1) (1997) 122–131.

[18] K. Atteson, The performance of neighbor-joining algorithms of phylogeny reconstruction, in: COCOON'97: Computing and Combinatorics, no. 1276 in Lecture Notes in Computer Science, 1997, pp. 101–110.

[19] V. Berry, O. Gascuel, Inferring evolutionary trees with strong combinatorial evidence, in: COCOON'97: Computing and Combinatorics, no. 1276 in Lecture Notes in Computer Science, 1997, pp. 111–123.

[20] L. Gąsieniec, J. Jansson, A. Lingas, A. Oestlin, T. Jiang, On the complexity of computing evolutionary trees, in: COCOON'97: Computing and Combinatorics, no. 1276 in Lecture Notes in Computer Science, 1997, pp. 134–145.

[21] S. L. Hakimi, S. S. Yau, Distance matrix and the synthesis of n-port resistance network, Technical Report 5, Network Theory Group (1963).

[22] D. Burton, P. L. Toint, On the use of an inverse shortest paths algorithm for recovering linearly correlated costs, Tech. rep., Facultés Universitaires ND de la Paix (1997).

[23] D. Burton, W. R. Pulleyblank, P. L. Toint, The inverse shortest paths problem with upper bounds on shortest paths costs, Tech. rep., Facultés Universitaires ND de la Paix (1997).

[24] J. P. Barthélemy, A. Guénoche, Les arbres et les représentations des proximités, Masson, Paris, 1988.

[25] M.-O. Delorme, A. Hénaut, Merging of distances matrices and classification by dynamic clustering, CABIOS 4 (4) (1988) 453–458.

[26] J. F. Caputo, K. J. Cook, A graph-theoretical approach t the prediction of the physical properties of alkanes based on the distances matrix, Pharmaceutical Research 6 (9) (1989) 809–812.

[27] L. Jin, M. Nei, Relative efficiencies of the maximum-parsimony and distance-matrix methods of phylogeny construction for restriction data, Molecular Biology and Evolution 8 (1991) 356–365.

[28] T. J. Warnow, Tree compatibility and inferring evolutionary history, J. of Algorithms 16 (1994) 388–407.

[29] M. Farach, S. Kannan, T. Warnow, A robust model for finding optimal evolutionary trees, Algorithmica 13 (1995) 155–179.

[30] R. Agarwala, V. Bafna, M. Farach, B. Narayanan, M. Paterson, M. Thorup, On the approximability of numerical taxonomy (fitting distances by tree metrics), in: Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, ACM, Atlanta, GA, 1996, pp. 365–372.

[31] L. A. Goldberg, P. W. Goldberg, C. A. Phillips, E. Sweedyk, T. Warnow, Minimizing phylogenetic number to find good evolutionary trees, Discrete Applied Mathematics 71 (1996) 111–136.

[32] S. K. Kannan, E. L. Lawler, T. J. Warnow, Determining the evolutionary tree using experiments, J. of Algorithms 21 (1996) 26–50.

[33] A. Amir, D. Keselman, Maximum agreement subtree in a set of evolutionary trees: metrics and efficient algorithms, SIAM J. Comput. 26 (6) (1997) 1656–1669.

[34] S. C. Varone, Trees related to realizations of distance matrices, Discrete Mathematics 192 (1998) 337–346.

[35] I. Althfer, On optimal realization of finite metric spaces by graphs, Discrete and Computational Geometry 3 (1988) 103–122.

[36] P. Winkler, The complexity of metric realisation, SIAM J. Disc. math. 1 (4) (1988) 552–559.

[37] J. M. S. Simões-Pereira, A note on optimal and suboptimal digraph realizations of quasidistance matrices, SIAM J. Algebraic Discrete Methods 5 (1984) 117–132.

[38] J. M. S. Simões-Pereira, A note on distance matrices with unicyclic graph realization, Discrete Mathematics 65 (1987) 277–287.

[39] W. Imrich, J. M. S. Simões-Pereira, C. M. Zamfirescu, On optimal embeddings of metrics in graphs, J. Combin. theory 36B (1984) 1–15.