



HAL
open science

A Linear Program for Points of Interest Relocation in Adaptive Maps

Simon Hengeveld, Frank Plastria, Antonio Mucherino, David Pelta

► **To cite this version:**

Simon Hengeveld, Frank Plastria, Antonio Mucherino, David Pelta. A Linear Program for Points of Interest Relocation in Adaptive Maps. 6th International Conference on Geometric Science of Information (GSI'23), Aug 2023, Saint Malo, France. pp.551-559, 10.1007/978-3-031-38299-4_57. hal-04183466

HAL Id: hal-04183466

<https://hal.science/hal-04183466>

Submitted on 19 Aug 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

A Linear Program for Points of Interest Relocation in Adaptive Maps

S.B. Hengeveld¹, F. Plastra², A. Mucherino¹, D.A. Pelta³

¹ IRISA, Université de Rennes, Rennes, France.

Email: {simon.hengeveld,antonio.mucherino}@irisa.fr

² BUTO, Vrije Universiteit Brussel, Brussel, Belgium.

Email: Frank.Plastra@vub.be

³ Dept. of Comp. Science and A.I., Universidad de Granada, Granada, Spain.

Email: dpelta@decsai.ugr.es

Abstract. The Point-Of-Interest (POI) relocation problem is a challenge encountered during the construction of personalized maps for given groups of users. This kind of maps was already studied and is known in the scientific literature under the name of “adaptive maps”. In this work, we formulate this problem as a subclass of the widely studied Distance Geometry Problem (DGP), where some extra constraints are included for taking into account the local orientation of the POIs in the map. These very same constraints allow us to linearize the problem, and hence to propose a novel linear program for the POI relocation problem. Our initial computational experiments indicate that our approach is promising for further investigations.

1 Introduction

Geographical maps are widely used in our everyday life. In a symbolic fashion, they are designed to depict the main relationships among the various objects represented therein. A classical example is given by city touristic maps, where the main points of touristic attraction are put in evidence in the map, and the connections between them are emphasized, for a potential tourist to easily navigate from one of such points to another. We refer to such *Points-Of-Interest* with the acronym POIs.

In classical geographical maps, the distances between POIs are generally proportional to their actual relative Euclidean distances. In some particular cases, such as in the touristic maps mentioned above, this general rule may be slightly relaxed in order to give more emphasis, for example, to a very important POI. The Euclidean distance, however, remains satisfied in most of the cases, for the final user to more easily interpret the map, and navigate through it.

The maps we are interested in are special maps where the Euclidean distance does not predominate. These maps are meant to give to the user a sense of proximity in some particular conditions impacting the user’s mobility conditions. Take for example a person in wheelchair approaching a cathedral from a classical route (the route one can derive from a standard map) and discovering that a few small steps need to be climbed in order to have access to the square where the cathedral stands. While many other

people with no mobility restrictions are likely to not even notice the presence of those steps, each step represents instead an important barrier for a wheelchair. This user in wheelchair would have much preferred to have an alternative map where the best route in this particular condition (to seat on a wheelchair) is depicted as the shortest one. We refer to this kind of maps as *adaptive maps*.

The very first reference to this kind of maps seems to have appeared in 1983 in [9]. The idea was then much later employed in [8] for example for the conception of metro maps. More recently, that original idea was extended to the concept of adaptive maps in [11] and [12]. These works indicate that three main steps can be identified for the construction of an adaptive map from a classical geographic map: (i) the identification of the POIs and of the main criteria for defining POI proximity; (ii) the relocation of the POIs on the basis of the defined proximity criteria; and (iii) the modification (basically the *distortion*) of the original map so that the new POI locations are taken into consideration in the final representation of the adaptive map.

We particularly focus our attention in this paper on the step (ii) for the construction of an adaptive map. In the following, this step will be referred to as the “POI relocation problem”. Since we position ourselves immediately after the step (i) above, we can suppose that a set of POIs is already given, and that a position in the two-dimensional Euclidean space (from the original map) is already associated to each POI. In addition, we can suppose that proximity information is associated to pairs of POIs, where such a proximity is appreciated by using a method that does not simply evaluate the classical Euclidean norm between the original locations.

The aim of this work is two-fold. Firstly, we will propose a formulation of the POI relocation problem as a distance geometry problem (see Section 2). The interest in such a formulation is given by the large body of work in distance geometry that can as a consequence be exploited in the context of adaptive maps. Secondly, we will focus on some recent research on linearization in distance geometry and we will propose a novel linear program that is particularly tailored to the POI relocation problem (Section 3). Finally, computational experiments will be presented in Section 4, and Section 5 will conclude the paper.

2 Distance Geometry for adaptive maps

Given a simple weighted undirected graph $G = (V, E, d)$ and a positive integer K , the Distance Geometry Problem (DGP) [6] asks whether a realization $X : V \rightarrow \mathbb{R}^K$ exists such that, every time an edge connects two vertices u and $v \in V$ (i.e. $\{u, v\} \in E$), the Euclidean distance between $X(u)$ and $X(v)$ corresponds to the given weight $d(u, v)$. The DGP belongs to the NP-hard class of problems [10], and several methods and algorithms have been proposed in recent years for its solution [7].

In the context of adaptive maps, we can naturally fix the dimension K to 2. The vertex set V represents the set of selected POIs, while the proximity information is encoded by the edge set E and the corresponding weights d . Since we suppose that an original map containing these POIs is already available, we can notice that a possible realization X_p for our graph G is already available. However, this realization represents the original map, where the distances between POIs reflect the traditional Euclidean distance.

The main interest in the POI relocation problem is to find an alternative location for all these POIs, on the basis of the proximity measure encoded by the edge set E , and of the weight function d .

We can remark that, while the proximity measure is not meant to be Euclidean, its representation in the new realization X is going to be Euclidean. This raises a main issue concerning the compatibility of the proximity measure with the properties of the Euclidean norm. One example is given by the triangular inequality, which needs to be satisfied in Euclidean space, but may not be satisfied by the employed proximity measure. As a consequence, a realization X where *all* weights d are *exactly* satisfied is unlikely to exist, and hence the constraints on the distances $d(u, v)$ need to be relaxed. In other words, the DGP is in this case not considered as a decision problem, but rather as an optimization problem where the deviations from the given weights is minimized [3].

We can additionally remark that the realization related to the original map (the realization we referred to with the symbol X_p) allows us to include some useful additional information to our problem instances. This is explained in the paragraph below: before we do this, we warn the reader that, in order to consider this extra information, it is necessary to enrich our graph representation with an orientation for each of its edges. In other words, our graphs G are *directed* graphs.

Suppose an important POI in the map we wish to adapt is originally placed at its North pole. Since the DGP is only based on distance information, there is no way to predict in advance whether the given distances will force this POI to move to completely different places in the map, and be positioned for example at the South pole, whereas the other POIs basically remain in the same locations. It is evident that this kind of result would be extremely confusing for the final user. Therefore, we include in our model the information about the local orientations of the POIs, and we enforce, in the resulting adaptive maps, that these orientations are preserved.

The local orientations are included as follows. From the original map, we can define two orthogonal axes, \hat{x} indicating the North-South direction, as well as \hat{y} indicating the West-East direction, that we can use to define specific Cartesian systems. For the generic edge $(u, v) \in E$, we can in fact define a Cartesian system centered in a given position $X_p(u)$ for u and having as axes \hat{x} and \hat{y} , so that it can be easily verified in which of the four standard quadrants (NW, NE, SW, SE) of this Cartesian system the position $X_p(v)$ of the second POI is contained. Then, we can simply partition the edge set E in four subsets, to which we assign the same names of the Cartesian quadrants: NW, NE, SW and SE. The fact that $(u, v) \in \text{NW}$ indicates, for example, that the positions for v need to be in *North-West* quadrant of the Cartesian system defined above and centered in the available position for u . Since our graph contains directed edges, it is necessary to pay attention to the fact that the information about the orientations remains symmetric for each possible realization, i.e. $(u, v) \in \text{NW}$ always needs to imply that $(v, u) \in \text{SE}$.

Since we focus on instances in dimension 2, a pair $(x, y) \in \mathbb{R}^2$ will indicate in the following the location of a POI through its x and y components in the Euclidean 2-dimensional space. For example, for the POI $v \in V$, we will write $X(v) = (x_v, y_v)$. On the basis of the comments above, we propose the following definition of POI relocation problem in dimension 2.

Definition 2.1. Given a simple weighted directed graph $G = (V, E, d)$ and an initial realization X_p of G , where

- V is a set of POIs,
- E represents the presence of proximity information and it is partitioned in the four subsets NW , NE , SW and SE in order to encode the local orientations in X_p ,
- the weight function d provides the numerical values for the proximity measures,

find a realization $X : V \rightarrow \mathbb{R}^2$ that is solution to the following optimization problem:

$$\begin{aligned} \min_X \sum_{(u,v) \in E} & \left(\sqrt{(x_v - x_u)^2 + (y_v - y_u)^2} - d(u, v) \right)^2 \\ \text{s.t. :} & \begin{cases} \forall (u, v) \in NW, & x_u \geq x_v \text{ and } y_u \leq y_v, \\ \forall (u, v) \in NE, & x_u \leq x_v \text{ and } y_u \leq y_v, \\ \forall (u, v) \in SW, & x_u \geq x_v \text{ and } y_u \geq y_v, \\ \forall (u, v) \in SE, & x_u \leq x_v \text{ and } y_u \geq y_v. \end{cases} \end{aligned} \quad (1)$$

Notice the use of the Euclidean norm in dimension 2, and that the function we minimize corresponds to the standard *stress* function already used in the context of the DGP [5], but other penalty functions may also be used. Since the partition of E into NW , NE , SW and SE reflects the relative realworld positions of the POI, these realworld coordinates give a feasible solution to the constraints in (1), which are thereby shown not to be contradictory.

The POI relocation problem can therefore be seen as a particular DGP subclass of instances, where the additional constraints on the relative orientations are enforced. In general, adding constraints to a known problem is likely to increase its complexity. In our case, instead, these new orientation constraints will allow us to formulate the POI relocation problem as a linear program consisting of only real variables. This implies that the DGP subclass consisting of our reformulated POI relocation problem (see next section) is a polynomial case. Moreover, the reduction of the problem to a pure linear problem implies that extremely rapid, well-tested and robust solvers can be used for its solution.

3 A linear program for POI relocation

Let $G = (V, E, d)$ be our simple weighted and directed graph representing an instance of the POI relocation problem. In this section, we propose a “purely” linear programming model for this important problem in the context of adaptive maps.

The optimization problem in Def. 2.1 is based on the Euclidean norm, where the square of the difference between desired and computed distances is summed up to give a global estimate of the error on the distances. In order to avoid the nonlinear terms, we replace the Euclidean norm with the L_1 norm and replace the squared differences by their absolute value.

We point out that we are not the first ones to consider combining these two substitutions: they were already considered for example in [4] (this article moreover, as well

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
P1	0	1553	2783	3614	4240	5674	6972	6064	7408	5387
P2	1553	0	1674	3867	2975	5936	7234	6326	6807	4693
P3	2783	1674	0	3427	2118	5098	6396	5409	5950	3837
P4	3614	3867	3427	0	2471	2571	4609	3745	5222	3292
P5	4240	2975	2118	2471	0	3600	4684	3172	3840	1727
P6	5674	5936	5098	2571	3600	0	2569	3076	5083	3416
P7	6972	7234	6396	4609	4684	2569	0	2692	4966	3903
P8	6064	6326	5409	3745	3172	3076	2692	0	2415	1698
P9	7408	6807	5950	5222	3840	5083	4966	2415	0	2303
P10	5387	4693	3837	3292	1727	3416	3903	1698	2303	0

Table 1. The walking distances between 10 pairs of POIs, that we use in our computational experiment. Reproduced from [12].

as [2], also exploited similar substitutions with the L_∞ norm), at the price, however, of adding binary variables and nonlinear constraints, with the consequent loss of polynomiality. It is precisely the additional wish to conserve the relative orientations of the POIs that allows us to use continuous variables only.

Our linear model is a minimization problem:

$$\min \sum_{(u,v) \in E} z_{uv}, \tag{2}$$

where

$$z_{uv} = ||x_v - x_u| + |y_v - y_v| - d(u, v)|, \tag{3}$$

where the symbol $|\cdot|$ simply represents the absolute value of a real number. The information about the relative orientation for every edge $(u, v) \in E$ (see equ. (1)) allows us to compute z_{uv} by performing only sums of coordinates and distances (or of their opposite values):

$$\left\{ \begin{array}{l} \forall (u, v) \in NW, z_{uv} \geq y_v - y_u + x_u - x_v - d(u, v), \\ \forall (u, v) \in NW, z_{uv} \geq d(u, v) - y_v + y_u - x_u + x_v, \\ \forall (u, v) \in NE, z_{uv} \geq y_v - y_u + x_v - x_u - d(u, v), \\ \forall (u, v) \in NE, z_{uv} \geq d(u, v) - y_v + y_u - x_v + x_u, \\ \forall (u, v) \in SW, z_{uv} \geq y_u - y_v + x_u - x_v - d(u, v), \\ \forall (u, v) \in SW, z_{uv} \geq d(u, v) - y_u + y_v - x_u + x_v, \\ \forall (u, v) \in SE, z_{uv} \geq y_u - y_v + x_v - x_u - d(u, v), \\ \forall (u, v) \in SE, z_{uv} \geq d(u, v) - y_u + y_v - x_v + x_u. \end{array} \right. \tag{4}$$

Notice that equ. (3) is therefore only satisfied at optimality. Some computational experiments on our linear model (2)-(1)-(4) are presented in the next section.

We remark that, in our formalism, the graph G allows us to have weights $d(u, v)$ that are different from the weights $d(v, u)$, for some $(u, v) \in E$, because G is a directed graph. However, in this work, we will suppose for simplicity that the distance information, even if not Euclidean, is still symmetric, and we will perform our computational experiments for this specific case.

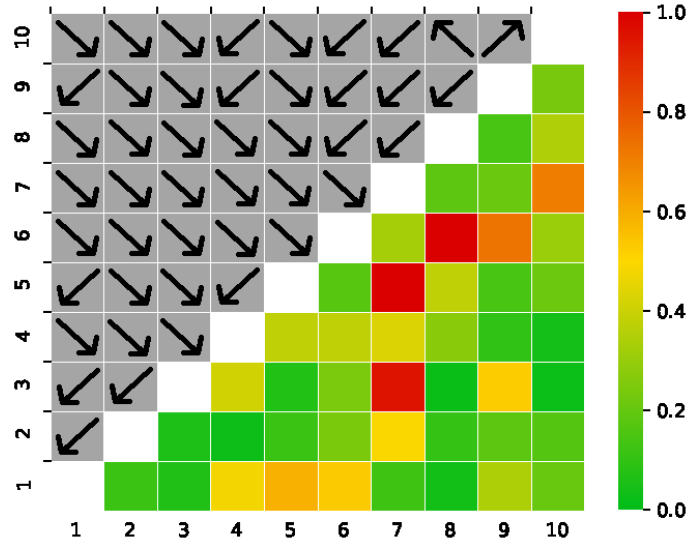


Fig. 1. Some information related to the 10-point instance in [12]. At the upper triangle, a visual representation of the local orientations of the POIs in the original realization, which our linear model is supposed to preserve in the found solution. At the lower triangle, a heatmap where the colors encode the difference in values between the Euclidean distances (in the original realization X_p) and the given proximity distances. The lower the difference, the smaller the value attributed to the corresponding square in the heatmap; all difference values are normalized between 0 and 1.

4 Computational experiments

We present an initial experiment where we use the 10-point POI relocation instance available from [12], reproduced in our Table 1. The graph G is defined so that the vertex set V corresponds to this set of POIs, and its edge set E makes it fully connected. The weight function d associated to the graph maps the edge set to a set of proximity measures where the walking distance between each pair of POIs is estimated. Notice that the walking distance is much likely to differ from the Euclidean distance, while paths along streets resemble more L_1 norm distances. In order to present the data in a visual way, we show in Fig. 1 a colored map depicting the POI distance variations (when comparing the original Euclidean distances and the given proximity distances) in the instance, as well as the local orientations of the POIs.

The IBM's CPLEX solver [1] was used to solve the POI relocation problem for this 10-point instance. The result was obtained in 0.072 seconds on a MacOS (ventura), 32GB RAM, and with CPU Apple M1 Max (10 cores). In order to superimpose the obtained coordinates with the original map, we first center both point sets at the origin by subtracting the centroid from each point. Then, we scale the both point sets so that their distances are within the same range. The obtained result, which has objective function value equal to 10273, is shown in Fig. 2.

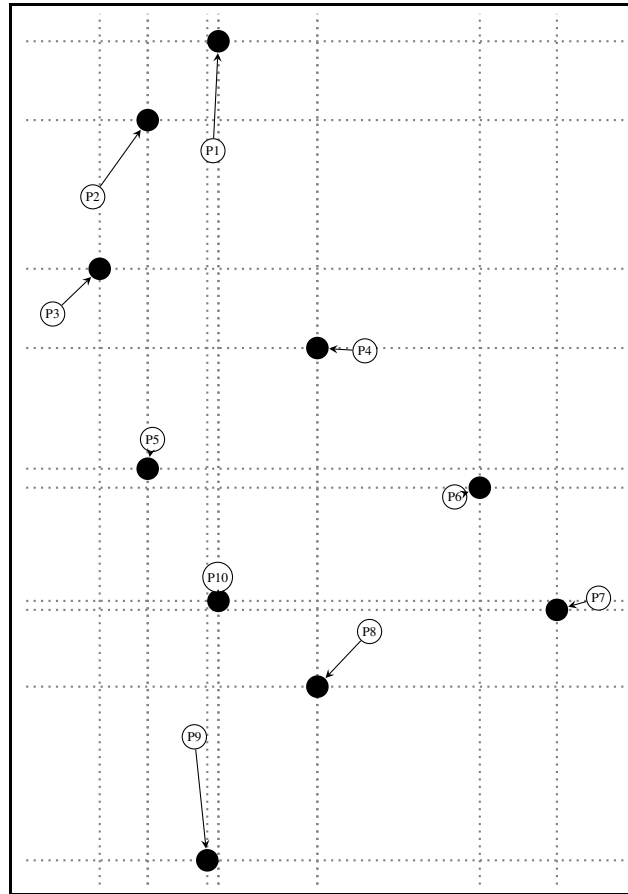


Fig. 2. The solution found by our linear model to the POI relocation problem related to data in Table 1. The points P^* indicate the original locations of the POI in the initial map. In black, the new positions for the POIs.

In order to validate our result, we focus our attention on the “red” edges (appearing in darker gray if in black-and-white) between P_6 and P_8 in Fig. 1. The distance between these two POIs is supposed to get larger in the adaptive map, and we can notice that this actually happens in our solution (see Fig. 2). We can remark, however, that the locations of some other POIs are subject to even larger modifications, apparently contradicting our input data (because these POIs are mostly related to “green” edges, see for example P_9). But a more attentive look at our adaptive map can reveal that these POIs were subject to a change in order to maintain their relative distances to other POIs which were involved in some “red” edges. In the particular case mentioned above, P_8 is subject to a substantial relocation because of the edge between itself and P_6 . As a consequence, P_9 is pushed down in the South direction, because otherwise its relative distance from P_8

would not be satisfied anymore (Fig. 1 shows that the distance between these two POIs remains approximately the same).

Let us now consider the point P3. Two of the edges related to P7 are “red”, indicating a quite important variation on the corresponding relative distances. However, the location of P3 does not seem to have changed much in our adaptive map. This behavior is most likely due to the introduced orientation constraints. In fact, P7 belongs to the SE quadrant of P10, and the little movement it is allowed to take seems to be the consequence of the fact that P10 moved a little in the South direction. Any other movement for P7 in the allowed quadrant would have most likely caused larger distance penalties with its closest neighbors. In general, we can observe that all local orientations are well respected, as imposed by the constraints in equ. (1) in Def. 2.1, that we use in our linear model.

5 Conclusions

We proposed a new solution method for the POI relocation problem, which is a problem encountered during the construction phase of an adaptive map. We showed that the POI relocation problem represents a particular class of DGP instances, where new constraints on the relative orientations between the POIs can be included. These new constraints allowed us to linearize the POI relocation problem, and thus to propose a linear program to find solutions for it where only real variables are involved.

We presented some initial experiments which show that this new approach is promising. We did not compare against other approaches for distance geometry or adaptive maps because we believe the comparison would have been unfair because we use the orientation constraints, which were, to the best of our knowledge, never used in previous studies.

Future works will be focusing on possible variants of the proposed linear model, to the possibility to use rotated versions of the L_1 norm, and to a large range of experiments aiming at finding the key points for improving our approach.

Acknowledgments

The authors thank the organizers of the 2021 Thematic Program on Geometric Constraint Systems, Framework Rigidity, and Distance Geometry (Fields Institute, Toronto, Canada) because they gave us the opportunity to participate to the Symposium on Sensor Network Localization and Dynamical Distance Geometry, where we set the basis of our collaboration.

AM and SH wish to thank the ANR French funding agency for support (MULTI-BIOSTRUCT project ANR-19-CE45-0019). DP acknowledges support from projects PID2020-112754GB-I0, MCIN/AEI /10.13039/501100011033 and FEDER/Junta de Andalucía-Consejería de Transformación Económica, Industria, Conocimiento y Universidades / Proyecto (B-TIC-640-UGR20).

References

1. IBM. ILOG CPLEX 12.6 User’s Manual. IBM, 2014.

2. G.M. Crippen, *An Alternative Approach to Distance Geometry using L^∞ Distances*, Discrete Applied Mathematics **197**, 20–26, 2015.
3. C. D’Ambrosio, V. Ky, C. Lavor, L. Liberti, N. Maculan, *New Error Measures and Methods for Realizing Protein Graphs from Distance Data*, Discrete & Computational Geometry **57**, 371–418, 2017.
4. C. D’Ambrosio, L. Liberti, *Distance Geometry in Linearizable Norms*, Lecture Notes in Computer Science **10589**, F. Nielsen, F. Barbaresco (Eds.), Proceedings of Geometric Side of Information (GSI17), Paris, 830–837, 2017.
5. W. Glunt, T.L. Hayden, M. Raydan, *Molecular Conformations from Distance Matrices*, Journal of Computational Chemistry **14**(1), 114–120, 1993.
6. L. Liberti, C. Lavor, N. Maculan, A. Mucherino, *Euclidean Distance Geometry and Applications*, SIAM Review **56**(1), 3–69, 2014.
7. A. Mucherino, C. Lavor, L. Liberti, N. Maculan (Eds.), *Distance Geometry: Theory, Methods and Applications*, 410 pages, Springer, 2013.
8. S. Raveau, J.C. Muñoz, L. de Grange, *A Topological Route Choice Model for Metro*, Transportation Research Part A: Policy and Practice **45**(2), 138–147, 2011.
9. C. Rolland-May, *A Valuation Model of Subjective Spaces*, IFAC Proceedings Volumes **16**(13), 375–380, 1983.
10. J. Saxe, *Embeddability of Weighted Graphs in k -Space is Strongly NP-hard*, Proceedings of 17th Allerton Conference in Communications, Control and Computing, 480–489, 1979.
11. M. Torres, D.A. Pelta, J.L. Verdegay, *A Proposal for Adaptive Maps*. In: J. Medina, M. Ojeda-Aciego, J. Verdegay, I. Perfilieva, B. Bouchon-Meunier, R. Yager (Eds.), Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU18). Communications in Computer and Information Science **855**, 657–666, 2018.
12. M. Torres, D.A. Pelta, J.L. Verdegay, C. Cruz, *Towards Adaptive Maps*, International Journal of Intelligent Systems **34**(3), 400–414, 2019.