Discrete & Computational Geometry manuscript No.
(will be inserted by the editor)

Low-Dimensional Embedding with Extra Information*

Mihai Bădoiu, Erik D. Demaine, MohammadTaghi Hajiaghayi, Piotr Indyk

MIT Computer Science and Artificial Intelligence Laboratory
32 Vassar Street, Cambridge, MA 02139, USA
{mihai,edemaine,hajiagha,indyk}@mit.edu

Abstract A frequently arising problem in computational geometry is when a physical structure, such as an ad-hoc wireless sensor network or a protein backbone, can measure local information about its geometry (e.g., distances, angles, and/or orientations), and the goal is to reconstruct the global geometry from this partial information. More precisely, we are given a graph, the approximate lengths of the edges, and possibly extra information, and our goal is to assign two-dimensional coordinates to the vertices such that the (multiplicative or additive) error on the resulting distances and other information is within a constant factor of the best possible. We obtain the first pseudo-quasipolynomial-time algorithm for this problem given a complete graph of Euclidean distances with additive error and no extra information. For general graphs, the analogous problem is NP-hard even with exact distances. Thus, for general graphs, we consider natural types of extra information that make the problem more tractable, including approximate angles between edges, the order type of vertices, a model of coordinate noise, or knowledge about the range of distance measurements. Our pseudo-quasipolynomial-time algorithm for no extra information can also be viewed as a polynomial-time algorithm given an "extremum oracle" as extra information. We give several approximation algorithms and contrasting hardness results for these scenarios.

* A preliminary version of this paper appeared in *Proceedings of the 20th Annual ACM Symposium on Computational Geometry*, June 2004, pages 320–329. The second and third authors were supported in part by NSF under grant number ITR ANI-0205445.

1 Introduction

Suppose we have a geometric structure (a graph realized in Euclidean space), but we can only measure local properties in this structure, such as distances between pairs of vertices, and the measurements are just approximate. In many applications we would like to use this approximate local information to reconstruct the entire geometric structure, that is, the realization of the graph. Two interesting questions arise in this context: when is such a reconstruction unique, and can it be computed efficiently? These problems have been studied extensively in the fields of computational geometry [CL92, EHKN99, Yem79, Sax79], rigidity theory [Hen92, Con91, JJ05], sensor networks [ČHH01, SRB01], and structural analysis of molecules [BKL99, ABC+05, CH88, Hen95]. The reconstruction problem arises frequently in several distributed physical structures such as the atoms in a protein [BKL99, CH88, Hen95] or the nodes in an ad-hoc wireless network [ČHH01, SRB01, PCB00].

A reconstruction is always unique (up to isometry) and easy to compute for a complete graph of exact distances, or any graph that can be "shelled" by incrementally locating nodes according to the distances to three non-collinear located neighbors. More interesting is that such graphs include visibility graphs [CL92] and segment visibility graphs [EHKN99]. In general, however, the reconstruction problem is NP-hard [Yem79], even in the strong sense [Sax79]. It is also NP-hard to test whether a graph has a unique reconstruction [Sax80, Section 6]. The uniqueness of a reconstruction in the *generic* case[1] was recently shown to be testable in polynomial time in two dimensions by a simple characterization related to generic infinitesimal rigidity [Hen92, JJ05], but this result has not yet led to efficient algorithms for actual reconstruction in the generic case.

This reconstruction problem can also be cast in the context of embedding arbitrary distance matrices into (low-dimensional) geometric spaces. Methods for computing such embeddings have their roots in work going back to the first half of the 20th century, and in the more recent work of Shepard [She62a, She62b], Kruskal [Kru64a, Kru64b], and others. The area is usually called *multi-dimensional scaling* and is a subject of extensive research with several applications [Wor]. However, despite significant practical interest, very few theoretical results exist in this area. The most commonly used algorithms are heuristic (e.g., gradient-based method or simulated annealing) and are often not satisfactory in terms of running time and/or embedding quality.

Recently, several papers [HIL03, Iva00, Băd03, ABF+99, FK99] have presented algorithms for various versions of the embedding problem. These

[1] In the generic case [JJ05], we are given the promise that the goal embedding is "generic". An embedding of a graph into d-dimensional Euclidean space is *generic* if the coordinates of the vertices are algebraically independent over the rationals, i.e., no polynomial over the vertex coordinates with rational coefficients evaluates to zero, except for the zero polynomial.

algorithms offer *provable* guarantees on the distortion of the computed embeddings, for particular families of input metrics and target spaces. More precisely, these papers consider the problem of embedding the complete graph $G = (V, E)$, with specified lengths $D[v, w]$ for all edges $\{v, w\} \in E$, chosen from some (restricted) family of metrics. Their goal is to embed G into d-dimensional ℓ_s space, for particular values of d and s, via a mapping $f : V \to \mathbb{R}^d$ either to approximately minimize *additive distortion* $\max_{\{v,w\}\in E} |\, \|f(v) - f(w)\|_s - D[v, w]\, |$, or to approximately minimize *multiplicative distortion* $\max_{\{v,w\}\in E} \|f(v) - f(w)\|_s / D[v, w]$ subject to $\|f(v) - f(w)\|_s \geq D[v, w]$ (noncontractiveness). (In d-dimensional ℓ_s space, distances and lengths are measured according to the ℓ_s norm $\|(x_1, \ldots, x_d)\|_s = \sqrt[s]{x_1^s + \cdots + x_d^s}$.)

Unfortunately, these results suffer from two important limitations:

- Few algorithms support embedding into constant-dimensional space (other than one dimension). Among the few such algorithms, none support Euclidean distances, which are the most common in the applications mentioned above. For example, the polynomial-time algorithm of [Băd03] works only for embeddings into the plane under the ℓ_1 norm, and its guarantee of approximately minimum additive distortion in ℓ_1 does not extend to the ℓ_2 norm.
- The algorithms assume that approximate distances between all pairs of points are specified. In some contexts, we have only partial distance information, for example, because an obstacle between two objects prevents estimating their distance or because the objects are too far for the estimation to be possible/reliable.[2]

The goal of this paper is to overcome these difficulties by obtaining efficient algorithms for approximate embedding of metrics into the plane. Our approach is to explore possible additional types of local information and study their influence on the complexity of the problem. In many practical scenarios such information is readily available. In other cases the amount of extra information needed is so small that it can be "guessed" via exhaustive enumeration, which leads to a pseudo-quasipolynomial-time algorithm that uses no extra information.[3] This algorithm is in fact the first such algorithm for embedding into low-dimensional Euclidean space with approximately optimal additive distortion.

We consider the following types of extra information:

Angle information: Between every pair of incident edges, we are given the approximate counterclockwise angle.

[2] More generally, we might consider the situation in which each distance has an explicit error bound, some of which might be infinite (in which case nothing is known about the distance). However, as we will see, the problem is hard even in the case of equal error bounds except for some error bounds which are infinite.

[3] An algorithm's running time is *quasipolynomial* if it is $2^{\log^{O(1)} n}$, *pseudopolynomial* if it is $N^{O(1)}$ where N is the maximum value of any number in the problem instance, and *pseudo-quasipolynomial* if it is $2^{\log^{O(1)} n \,\cdot\, \log^{O(1)} N}$.

Extremum oracle: Suppose that the x coordinates of the embedding are known (fixed). Let f be an optimal (minimum-distortion) embedding subject to these and all other constraints. The extremum oracle reports, in any specified vertical slab of the optimal embedding, the minimum y coordinate of a point and a point achieving that coordinate, and symmetrically for the maximum y coordinate. More precisely, given a range $[x_l, x_r]$, the oracle reports the data point $p = \text{argmin}_{p': f_x(p') \in [x_l, x_r]} f_y(p')$ and $f(p)$, and symmetrically with argmax. In addition, we require that the answers returned by the oracle to different queries are consistent, that is, based on the same embedding f.
Guessing this extra information is exactly what causes one of our algorithms to use pseudo-quasipolynomial time when given no extra information.

Order type: For some point p and all pairs of points q, r, we are given the clockwise/counterclockwise orientation of $\triangle pqr$.

Distribution information: We know that the metric is induced by random points in a square (as in, e.g., [GRK04]) plus adversarial noise added to their pairwise distances.

Range constraints: Each point p has a range r_p such that we know the (approximate) distance between p and a point q precisely when this distance is at most r_p.

One of our motivations for studying these problems is "autoconfiguration" in the Cricket Compass [PMBT01,MIT] location system. In this system several beacons are placed in a physical environment, and the goal is to find the global geometry of these beacons in order to enable private localization of mobile devices such as PDAs (personal digital assistants). In general, the beacons live in three-dimensional space, but a common scenario is that they all live in a common plane (the ceiling). Beacons can measure approximate pairwise distances, with subcentimeter accuracy and a range of up to several meters, using a combination of ultrasonic and radio pulses (measuring the difference in travel time between the sound-speed pulse and the light-speed pulse). Using two or more ultrasonic transceivers to measure distances from two or more points on a beacon, beacons can also measure approximate counterclockwise angles of other beacons within range, relative to a local coordinate system. In this practical scenario, distribution information, range constraints, order type, and especially angle information are all reasonable assumptions to consider.

We show that any of the types of extra information described above, in addition to the approximate distance information given by D, often allow us to design efficient algorithms to construct embeddings into two dimensions with approximately optimal distortion. Specifically, we develop polynomial-time algorithms for the following versions of this embedding-with-extra-information problem:

1. Embedding a general graph with approximate angle information into two-dimensional ℓ_s space, $s \in \{1, 2, \infty\}$, with approximately optimal

multiplicative distortion. If we are given the counterclockwise angle of each edge with respect to a fixed axis, or we are given counterclockwise angles between incident pairs of edges in the complete graph, our approximation factor is $O(1)$. If we are given counterclockwise angles between pairs of incident edges in a general graph, our approximation factor is $O(\text{diam})$ where diam is the diameter of the graph. The approximation factors depend on the additive error on the angles; see Section 2 for details.
These algorithms are the first subexponential-time algorithms for embedding an arbitrary metric into a low-dimensional space (even in the one-dimensional case) to approximately minimize multiplicative distortion. Without angles, even embedding tree metrics into the line with approximately minimum multiplicative distortion is hard to approximate better than a factor of $\Theta(n^{1/12})$, by a recent result of Bădoiu et al. [BCIS05].

2. Embedding a complete graph into the Euclidean plane with $O(1)$-approximate additive distortion in pseudo-quasipolynomial time of $2^{O(\log n \cdot \log^2 \Delta)}$ where Δ is the "spread" of the input point set. We obtain this result in Section 3 using a polynomial number of calls to an extremum oracle, which can be simulated in pseudo-quasipolynomial time.
This algorithm is the first algorithm for minimizing the additive distortion of an embedding into a low-dimensional Euclidean space, other than trivial exponential-time algorithms.
3. Embedding a complete graph into the Euclidean plane with $O(1)$-approximate additive distortion given the orientation of all triples of points involving a common point (Section 4).
4. Embedding a complete graph into the Euclidean plane with $O(1)$-approximate additive distortion given the prior that the distances D are approximately induced by a random set of points in a unit square. In this case our algorithm returns an embedding with additive distortion that is within a constant factor of the maximum noise added to any distance. See Section 5 for the detailed formulation.
5. Embedding a general graph that satisfies the range constraints into the line with $O(1)$-approximate additive distortion (Section 6).
6. In contrast, we show that embedding a general graph that satisfies the range constraints into two-dimensional ℓ_p space, for $p \in \{1, 2, \infty\}$, is NP-hard (Section 7). This problem was known to be NP-hard without range constraints in d-dimensional Euclidean space for all d [Sax79].

Several of these algorithms are practical; often they are based on simple linear programs.

2 Embedding with Angle Information

This section considers embedding a graph with given edge lengths up to multiplicative error and given angles with additive error, in ℓ_1, ℓ_2, and ℓ_∞.

We consider several possible angle specifications in the next section, and reduce to the case where we know the counterclockwise angle between every edge and one fixed edge.

2.1 Different Types of Angle Information

Lemma 1 *Given a complete graph, and given counterclockwise angles between pairs of incident edges each with (one-sided or two-sided) additive error at most γ, we can compute the counterclockwise angle of every edge with respect to a particular edge with additive error at most 2γ.*

Proof Fix one edge (p, q) and call it the x axis. To estimate the counterclockwise angle of an edge (v, w) with respect to the x axis, we use the known counterclockwise angles $\theta_1 = \angle pqv$ and $\theta_2 = \angle qvw$. If the angles were exact, the counterclockwise angle of (v, w) with respect to (p, q) would be $\theta_1 + \theta_2 - 180°$ (modulo $360°$). With additive error, the errors in $\theta_1 + \theta_2$ accumulate to at most double in the worst case. □

Lemma 2 *Given a general graph, and given counterclockwise angles between pairs of incident edges each with (one-sided or two-sided) additive error at most γ, we can compute the counterclockwise angle of every edge with respect to a particular edge with additive error at most* $(\text{diam} + 1)\gamma$ *where* diam *is the diameter of the graph.*

Proof Similar to Lemma 1, except now we must combine counterclockwise angles along a path $p, q, \ldots, v, w$, which might have length up to $\text{diam} + 2$, and therefore involves at most $\text{diam} + 1$ angles. □

This lemma is the best we can obtain in the worst case. We can of course improve the angles estimates by, e.g., choosing (p, q) to be maximally central, computing shortest paths, etc. If the errors are known to be independent and randomly distributed with mean zero, the error is $O(\sqrt{\text{diam}})$ with high probability, where diam is the diameter of the graph.

2.2 ℓ_2 Algorithm

Our algorithm for embedding into the Euclidean plane assumes, possibly using the reductions of the previous section, that we are given the approximate counterclockwise angle of every edge with respect to one edge (which we view as the x axis). The algorithm sets up a constraint program for finding the coordinates of each vertex. The straightforward setup has nonconvex constraints and is difficult to solve. We relax the program to a convex program at the cost of some error. We further relax the program to a linear program at the cost of additional error.

The basic optimization problem has the following constraints. For every edge (p, q), the distance and angle information of that edge specifies

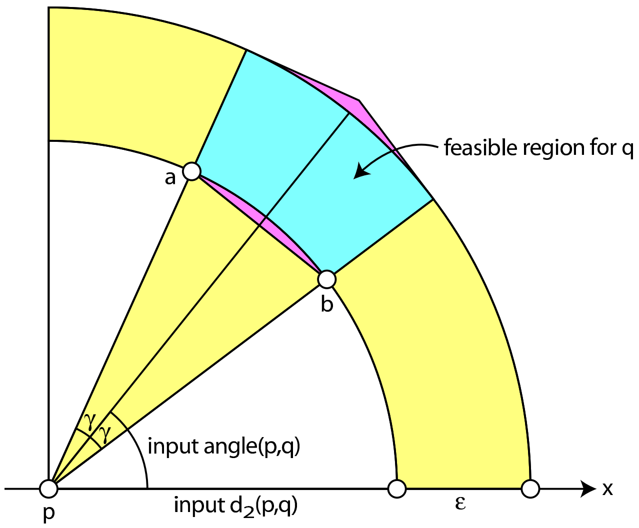


Fig. 1 Feasible region of a point q with respect to p given the ℓ_2 distance within a multiplicative ε and given the counterclockwise angle to the x axis within an additive γ. (Measuring the counterclockwise angle, instead of just the angle, distinguishes between q being "above" or "below" the x axis.)

a (nonconvex) *constraint region*, relative to the location of p, that must contain q. (See Fig. 1.) Conditioning on that there is an embedding of the graph achieving multiplicative error ε on the distances and additive error γ on the angles, we can find such an embedding by finding a feasible solution satisifying all constraints. If only one of these error parameters (e.g., γ) is known, we can still find such an embedding by setting the objective function to minimize the other error parameter (e.g., ε). If neither error parameter is known, we obtain a family of solutions by minimizing one error parameter subject to various choices for the other parameter; alternatively, we can minimize any desired linear combination of the error parameters by setting the objective function accordingly.

We can relax each constraint region to be convex by taking its convex hull. More precisely, we add one edge (a, b) to cut off the inner arc of the region; see Fig. 1. This relaxation, applied to every constraint region defined by an edge (p, q), produces a convex program. The maximum possible error is obtained when q is placed at the midpoint between a and b. Then the distance between p and q is $\cos\gamma$ times the input distance between p and q. We can transform this contraction into an expansion by multiplying all distances by $1/\cos\gamma$. Thus, the maximum expansion is at most $(1+\varepsilon)/\cos\gamma$, proving the following theorem:

Theorem 1 *Given a graph, given the length of each edge with multiplicative error ε, and given the counterclockwise angle of every edge with respect to a particular edge with additive error γ, we can compute in polynomial time* *an ℓ_2 embedding with angles of maximum additive error γ and distances of maximum multiplicative error $(1+\varepsilon)/\cos\gamma - 1$.*

We can further relax the constraint region to be piecewise-linear by approximating the unique arc of the region with a polygonal chain. Then we obtain a linear program from combining the relaxed constraint for each edge (p, q). If we use $k + 1 \geq 2$ segments in a regular chain, the maximum expansion factor of a distance is $(1 + \varepsilon)/\cos(\gamma/k)$. By incorporating the expansion factor from the previous theorem as well, we obtain the following theorem:

Theorem 2 *Given a graph, given the length of each edge with multiplicative error ε, and given the counterclockwise angle of every edge with respect to a particular edge with additive error γ, we can compute in polynomial time an ℓ_2 embedding with angles of maximum additive error γ and distances of maximum multiplicative error*

$$\frac{1+\varepsilon}{\cos\gamma\cos(\gamma/k)} - 1 = \frac{1+\varepsilon}{\cos\gamma} - 1 + O\left(\frac{\gamma^2}{k^2}\right).$$

2.3 More Types of Angle Information

For embedding into ℓ_1, we need additional information about the global rotation of the graph. More precisely, we need to know, for each edge (p, q), the quadrant of q with respect to p. In other words, we need to know the two high-order bits of the counterclockwise angle of each edge (p, q) with respect to the x axis, i.e., whether this angle is in $[0, 90°]$, $[90°, 180°]$, $[180°, 270°]$, or $[270°, 360°]$. Because of our additive angle errors, we may not know to which quadrant an edge belongs; in this case we would like to know that the edge is borderline between two particular quadrants.

If our input specifies counterclockwise angles of edges with respect to the x axis, we are done. For other types of input, we can apply the following reductions:

Lemma 3 *Given a graph, given counterclockwise angles between pairs of incident edges each with (one-sided or two-sided) additive error at most γ, and given the counterclockwise angle of one edge relative to the x axis with the same additive error, we can compute the counterclockwise angle of every edge with respect to the x axis with additive error at most* $(\mathrm{diam} + 2)\gamma$.

Proof Apply Lemma 2 relative to the edge for which we know the counterclockwise angle with respect to the x axis, and translate using this angle. □

Lemma 4 *Given a graph, and given counterclockwise angles between pairs of incident edges each with (one-sided or two-sided) additive error at most γ, we can compute a family of $O(1/\gamma')$ possible assignments of counterclockwise angles relative to the x axis with additive error at most $\gamma + \gamma'$.*

Proof Apply Lemma 2 to obtain counterclockwise angles relative to an edge e, and then "guess" the counterclockwise angle of the x axis with respect to e among the $\lceil 360°/\gamma' \rceil$ angles of the form $0, \gamma', 2\gamma', \ldots$. □

2.4 ℓ_1 Algorithm

We can adapt the ℓ_2 algorithm to an ℓ_1 algorithm as follows. The convex program and linear program are the same as before; the only difference is the shape of the constraint region of q with respect to p. For an edge (p, q) that is known to be in a particular quadrant, the region is a trapezoid as shown in Fig. 2(a). In this case the region is already convex and polygonal, and we find an embedding with no error beyond the optimal distortion.

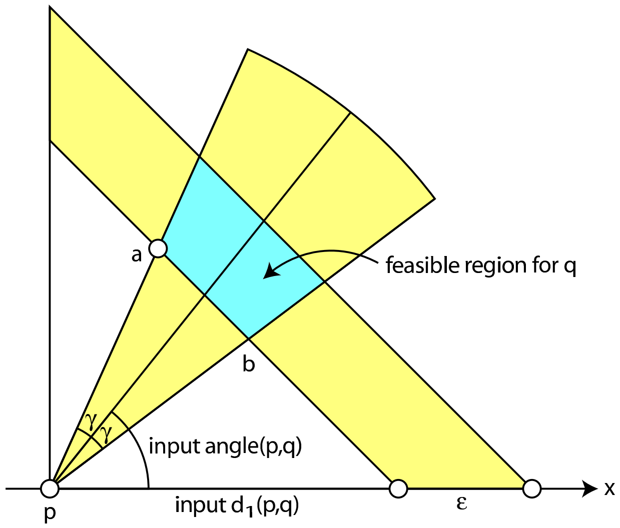
The difficult case is when the edge (p, q) straddles two quadrants, i.e., is almost parallel to a coordinate axis. See Fig. 2(b). In this case the angular wedge intersects two sides of the ℓ_1 circle around p, and the region becomes a nonconvex "V". As before, we convexify this region by closing the mouth of the "V". The resulting region is also polygonal, so we can apply linear programming.

The worst-case error arises when (p, q) is exactly parallel to a coordinate axis. Then the smallest distance between p and a relaxed position for q is $1 - (\tan\gamma)/(1 + \tan\gamma)$ times the input distance between p and q. Again we can transform this contraction into an expansion by multiplying all distances by $1/[1 - (\tan\gamma)/(1 + \tan\gamma)]$, and the maximum expansion is at most $(1 + \varepsilon)/[1 - (\tan\gamma)/(1 + \tan\gamma)]$:

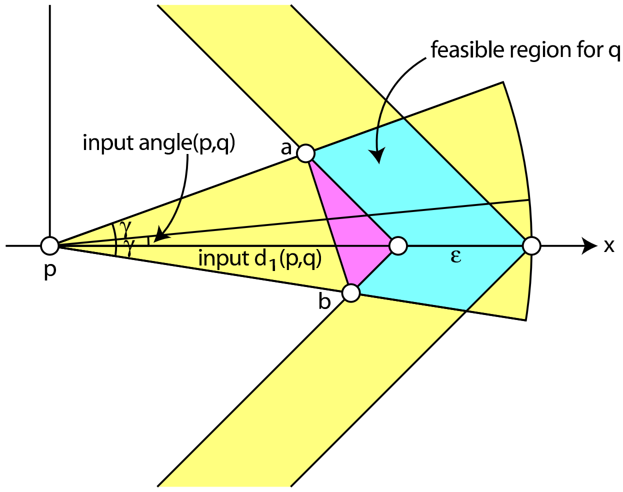
Theorem 3 *Given a complete graph, given the length of each edge with multiplicative error ε, and given the counterclockwise angle of every edge with respect to the x axis with additive error γ, we can compute in polynomial time an ℓ_1 embedding with angles of maximum additive error γ and distances of maximum multiplicative error*

$$\frac{1 + \varepsilon}{1 - (\tan\gamma)/(1 + \tan\gamma)} - 1 = (1 + \varepsilon)(\gamma + O(\gamma^3)) + \varepsilon.$$

If we are given the approximate counterclockwise angles between incident pairs of edges, and the approximate counterclockwise angle between one edge and the x axis, then we can apply this theorem in combination with Lemma 3. If we are just given the approximate counterclockwise angles between incident pairs of edges, we can consider all "combinatorial rotations" with respect to the x axis, and extract whether each edge is roughly horizontal, roughly vertical, or substantially within one of the four quadrants. This partial information increases the region error for near-horizontal and near-vertical edges, and does not preserve the angle for all other edges, but will approximately preserve distances in the resulting embedding.



(a) Convex case



(b) Nonconvex case

Fig. 2 Feasible region of a point q with respect to p given the ℓ_1 distance within a multiplicative ε and given the counterclockwise angle to the x axis within an additive γ.

2.5 Extension to ℓ_∞

We can directly adapt the ℓ_1 algorithm to an ℓ_∞ algorithm. If we rotate an ℓ_∞ input by 45°, and scale by a factor of $1/\sqrt{2}$ in each dimension, then we obtain an "identical" ℓ_1 input. The two inputs are identical in the sense that the ℓ_∞ distance between any pair of points in the ℓ_∞ input is equal to the ℓ_1 distance between that pair in the ℓ_1 input. Thus, we can apply the ℓ_1 embedding algorithm to the ℓ_1 input, and then undo the transformation, and we obtain an ℓ_∞ embedding of an ℓ_∞ input.

3 Embedding with an Extremum Oracle

In this section we describe an $O(1)$-approximation algorithm for minimizing the additive distortion in an embedding of a complete graph with distances specified by D into the Euclidean plane. Define the *spread* Δ of the metric by $\Delta = \operatorname{diam}(D)/\varepsilon$, where ε is the minimum additive distortion possible and $\operatorname{diam}(D)$ is the diameter of D, i.e., the maximum distance in D. The algorithm runs in polynomial time, multiplied by a factor of $O(\lg \Delta)$ if ε is not approximately known, given an extremum oracle for a promised embedding f attaining minimum additive distortion ε. By exhaustive enumeration of the possible oracle answers, this algorithm can be converted into an algorithm without extra information having pseudo-quasipolynomial running time $2^{O(\log n \cdot \log^2 \Delta)}$.

We view the algorithm as being given D and $\varepsilon > 0$, and the goal is either to find an embedding of D into the plane with additive distortion $O(\varepsilon)$ or to report that no embedding with additive distortion at most ε exists. Here we assume that $\varepsilon > 0$ (and thus Δ is finite) because it is easy to test whether a complete graph of distances can be embedded without distortion. If ε is unknown, we can guess ε up to a constant factor in a standard way by trying values of the form $\operatorname{diam}/2^i$ for $i = 0, 1, 2, \ldots$. This guessing multiplies the running time by $O(\lg \Delta)$, which is absorbed in the pseudo-quasipolynomial time bound.

We use a geometric annulus (the difference between two disks of the same radii) to represent approximately known distances. Define $R(p, r, \delta)$ to be the annulus centered at point p with inner radius $r - \delta$ and outer radius $r + \delta$. The next lemma shows how two such annuli can help isolate a point.

Lemma 5 *Consider two points a and b at a distance r on the x axis, and two radii r_a and r_b such that $\max\{r_a, r_b\} \le 2r$. Then, for any $\varepsilon \le r$, the intersection $R = R(a, r_a, \varepsilon) \cap R(b, r_b, \varepsilon)$ is enclosed in a vertical slab $[x_0 - 4\varepsilon, x_0 + 4\varepsilon]$, where $x_0 = (r^2 + r_a^2 - r_b^2)/2r$.*

Proof By a suitable translation, we may assume without loss of generality that $a = (0, 0)$ and $b = (r, 0)$. Any point $(x, y) \in R$ must satisfy $(r_a - \varepsilon)^2 \le x^2 + y^2 \le (r_a + \varepsilon)^2$ and $(r_b - \varepsilon)^2 \le (x - r)^2 + y^2 \le (r_b + \varepsilon)^2$. Subtracting these two bounds, the terms quadratic in x and y cancel out, and we obtain that any point $(x, y) \in R$ must satisfy $|x - (r^2 + r_a^2 - r_b^2)/2r| \le \varepsilon |r_a - r_b|/r \le \varepsilon(r_a + r_b)/r \le 4\varepsilon$. Thus (x, y) is in the vertical slab $[x_0 - 4\varepsilon, x_0 + 4\varepsilon]$ where $x_0 = r^2 + r_a^2 - r_b^2$. □

Using this tool, we show how to guess approximate x coordinates; the following lemma is also useful in Section 4.

Lemma 6 *Given a complete graph $G = (V, E)$ with distances specified by D, and given $0 < \varepsilon < \operatorname{diam}(D)/2$, we can compute in polynomial time a set of guesses of the form $x : V \to \mathbb{R}$ such that, if there is an embedding f of G into the Euclidean plane of minimum additive distortion ε, at least one guess satisfies, for a suitable translation and rotation $\tilde{f}$ of f, $|\tilde{f}_x(v) - x(v)| \le 5\varepsilon$ for all $v \in V$. We can also ensure that the x coordinates are distinct in each guess.*

Proof First we guess the diameter pair (a, b) in the embedding f, that is, the pair that maximizes $\|f(p)-f(q)\|$, by trying all pairs such that $D[a, b] \ge \operatorname{diam}(D) - 2\varepsilon$. (The diameter pair must satisfy this property because f has additive distortion ε.) By suitable translation and rotation $\tilde{f}$ of f, we can assume that $\tilde{f}(a) = (0, 0)$ and $\tilde{f}_y(b) = 0$. Therefore we can assign $x(a) = 0$ and $x(b) = D[a, b]$, and we have that $x(a) = \tilde{f}_x(a)$ and $|x(b) - \tilde{f}_x(b)| \le \varepsilon$.

To guess the remaining x coordinates $\tilde{f}_x(v)$ for vertices $v \notin \{a, b\}$, we proceed as in Bădoiu's algorithm [Băd03]. For any such vertex v, define the region $R_v = R(a, D[a, v], \varepsilon) \cap R(b, D[b, v], \varepsilon)$. Because $D[a, v] \le \operatorname{diam}(D) \le D[a, b] + 2\varepsilon < 2D[a, b]$, we can apply Lemma 5 and set $x(v)$ to the center x_0 of the vertical slab. Because $|x(b) - \tilde{f}_x(b)| \le \varepsilon$ and at worst the errors add, we have that $|x(v) - \tilde{f}_x(v)| < 5\varepsilon$.

If two x coordinates are equal, we perturb them slightly, to guarantee that all x coordinates are distinct. By a sufficiently small perturbation, we preserve that $|x(v) - \tilde{f}_x(v)| < 5\varepsilon$ for all vertices v. Therefore we obtain a suitable guess x. □

We assume in the rest of this section that $\varepsilon = 1$, by scaling the entries in D by $1/\varepsilon$. Thus $\Delta = \operatorname{diam}(D)$.

We claim that it suffices to consider embeddings g with x coordinates given by a suitable guess of Lemma 6. Consider the translated and rotated optimal embedding $\tilde{f}$. Construct f' by setting $f'_x(v) = x(v)$ and $f'_y(v) = \tilde{f}_y(v)$ for all vertices v. By Lemma 6, $\|\tilde{f}(v) - f'(v)\| < 5\varepsilon$ (for a suitable guess). By the triangle inequality, $|\,\|f'(v) - f'(w)\| - \|\tilde{f}(v) - \tilde{f}(w)\|\,| < 10\varepsilon$, so the additive distortion of f' is at most $\varepsilon + 10\varepsilon = 11\varepsilon$.

In addition, we require that each y coordinate in the embeddings we construct is a multiple of ε. By a similar argument as above, this assumption increases the additive error by at most ε, to $c'\varepsilon = 12\varepsilon$.

The algorithm uses the divide-and-conquer paradigm to compute the y coordinates in an embedding g (using the x coordinates given by the guess of Lemma 6). First, we compute the median x_m of the x coordinates of the vertices as mapped by g. Let V^+ be the set of all points $p \in V$ such that $g(p)$ has x coordinate larger than the median x_m, and let $V^- = V - V^+$. The algorithm proceeds by creating the set of *constraints* on $g(V^+)$ and $g(V^-)$. The constraints have two properties:

1. The constraints are feasible; namely, f' satisfies them.
2. For any mapping g satisfying the constraints, we have $|\,\|g(p) - g(q)\| - D[p, q]| \le c$, for all $p \in V^+$ and $q \in V^-$; here c is a certain global constant.

These properties allow us to compute $g(V^+)$ and $g(V^-)$ (while enforcing the constraints) recursively and independently from each other.

The constraints are of the form "$g_y(p) \in Y(p)$", where $Y(p)$ is a finite set of intervals. They are constructed as follows. For $i \geq 1$, define $I_i = (x_m + 2^{i-1} - 1, x_m + 2^i - 1]$; for $i \leq -1$, define $I_i = -I_{-i}$. For each I_i, the algorithm queries the extremum oracle to obtain a point $p^i_{up} \in V$, $f'_x(p^i_{up}) \in I_i$, such that $f'_y(p_{up})$ is maximum. Similarly, the algorithm obtains p^i_{down}. In addition, the algorithm obtains the values $f'_y(p^i_{up})$ and $f'_y(p^i_{down})$ for each i.

With the oracle's answers in hand, the algorithm imposes the following new constraints, for each i, $d \in \{up, down\}$, and $p \in V$:

1. "$g_y(p^i_d) = f'_y(p^i_d)$";
2. if $f'_x(p) \in I_i$, then "$g_y(p) \in [f'_y(p^i_{down}), f'_y(p^i_{up})]$"; and
3. "$g(p) \in R(f'(p^i_d), D[p^i_d, p], c')$". (This latter condition can be expressed as a restriction on $g_y(p)$.)

As mentioned above, after imposing the constraints, the algorithm recurses to find $g(V^+)$ and $g(V^-)$ independently. At the leaf level of recursion (i.e., when we are given only one point p), the algorithm sets $g_y(p)$ to be an arbitrary y coordinate satisfying all constraints (if it exists). If no such y coordinate exists, the algorithm concludes that there is no acceptable embedding for the guess of Lemma 6 and this set of oracle answers.

The oracle's answers can be implemented by trying all possible choices of the guessed variables. Each combination of a guess from Lemma 6 and the oracle answers leads to a different execution of the algorithm, ending with either a failure or a final embedding g whose additive distortion can be checked to be at most $c'\varepsilon$. The total number of such choices is bounded by $2^{O(\log^2 \Delta)}$, because there are at most $O(\Delta)$ different potential values for the y coordinates of f'. The claimed bound for the running time $T(n)$ follows from the recursion $T(n) = 2^{O(\log^2 \Delta)}[T(n/2) + n^{O(1)}]$. Note that, if we could compute the oracle's answers in polynomial time, our algorithm would have polynomial running time as well.

It is easy to see that the constraints imposed at all stages are consistent with f'. It remains to show that, after $g(V^+)$ and $g(V^-)$ satisfying the constraints are found, we then have $|\,\|g(p) - g(q)\| - D[p,q]| \leq c$, for all $p \in V^+$, $q \in V^-$, and some global constant $c > 0$. This is done via the following two lemmas.

Lemma 7 *Consider any two points $a = (x, y)$ and $b = (x', y')$, such that $x' \geq x/2$. Define $b' = (x', y)$ and $I = \{0\} \times \mathbb{R}$. Then for any r there exists r' such that $I \cap R(a, r, c') \subset R(b', r', c)$ for a fixed constant c.*

The interpretation (and usage) of this lemma is as follows. Consider the points $g(p)$ and $g(q)$ as above, and assume that $g_x(p) \in I_i$, $i < 0$, and $g_x(q) \in I_j$, $j > 0$, such that $(i, j) \neq (-1, 1)$. (We will take care of the case $(i, j) = (-1, 1)$ later.) In the procedure described above, we impose constraints on $g(p)$ of the form "$g(p) \in R(a, r, c')$", for $d \in \{down, up\}$, $r = D[p_d^j, p]$, and $a = f'(p_d^j)$. However, it will be more convenient to consider a different constraint, namely "$g(p) \in R(b', r', c)$", where $b' = (f'_x(q), f'_y(p_d^j))$, because in this way $f'(q)$ and b' have the same x coordinate, a property used in the next lemma. However, we do not know $f'(q)$, so we cannot impose the second constraint explicitly. Fortunately, Lemma 7 guarantees that the latter constraint is implied by the former. Note that the assumption $x' \geq x/2$ is satisfied by the construction of the intervals I_i and I_j.

Proof (of Lemma 7) Without loss of generality, we can assume that $I \cap R(a, r, c')$ is nonempty. In addition, we assume that $I \cap R(a, r, c')$ consists of two disconnected components. (If it consists of only one component, the proof is similar.) Finally, without loss of generality, we can assume that $y = 0$. Denote the upper component (with larger y coordinates) by $Y = \{0\} \times [y_d, y_u]$. Let $q_d = (0, y_d)$, $q_u = (0, y_u)$. Note that $y_u^2 + x^2 = (r + c')^2$, and $y_d^2 + x^2 = (r - c')^2$. By symmetry, it suffices to ensure that $Y \subset R(b', r', c)$.

Define $r' = \|b' - q_u\| = {x'}^2 + y_u^2$. Consider any $(0, z) \in Y$. We need to show (1) $\|b' - (0, z)\|^2 \leq (r' + c)^2$ and (2) $\|b' - (0, z)\|^2 \geq (r' - c)^2$ or $r' < c$. First, $\|b' - (0, z)\|^2 = {x'}^2 + z^2 \leq {x'}^2 + y_u^2 = {r'}^2$. Second, $\|b' - (0, z)\|^2 \geq {x'}^2 + y_d^2 \geq {r'}^2 - 2r'c + c^2$.

By plugging in the expressions for y_d^2, ${r'}^2$, and then y_u^2, we obtain equivalently that

$$ {x'}^2 + (r - c')^2 - x^2 \geq [(r + c')^2 - x^2] + {x'}^2 - 2r'c + c^2, $$

which simplifies to $2r'c - c^2 \geq 2c'r$.

Because $r' \geq \max\{x', y_u\}$, $r' \leq x + y_u$, and (by the assumption) $x' \geq x/2$ and $r' \geq c$, it follows that the last expression is satisfied if $c \geq 4c'$. This proves the lemma. □

The next lemma is about the following configuration of points: $a = (0, y_a)$, $b = (0, y_b)$, $c = (x, y_c)$, and $d = (x, y_d)$. For any r_a, r_b, r_c, r_d, and s, define two sets:

$$ S_1 = \{(0, y) : y_a < y < y_b\} \cap R(c, r_c, s) \cap R(d, r_d, s), $$
$$ S_2 = \{(x, y) : y_c < y < y_d\} \cap R(a, r_a, s) \cap R(b, r_b, s). $$

Lemma 8 *The difference* $\max_{u \in S_1, v \in S_2} \|u - v\| - \min_{u \in S_1, v \in S_2} \|u - v\|$ *is at most* $3s$.

Before we prove this lemma, we show how the two lemmas together imply that, for any two points $p \in V^-$ and $q \in V^+$ satisfying the imposed constraints, we have $\|g(p) - g(q)\| = \|f'(p) - f'(q)\| \pm O(1)$ as desired. To show this implication, we consider two cases. Let $f'_x(p) \in I_i$ and let $f'_x(q) \in I_j$.

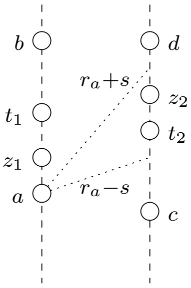


Fig. 3 Proof illustration of Lemma 8.

Case 1: $i = -1, j = 1$. Let $y_{up} = \max[f'_y(p^{-1}_{up}), f'_y(p^1_{up})]$ and $y_{down} = \max[f'_y(p^{-1}_{down}), f'_y(p^1_{down})]$. If $y_{up} - y_{down} \leq c_2$ for c_2 larger than, say, $10c'$, then the statement follows. Otherwise, if $y_{up} - y_{down} > 10c'$, then for any $u \in \{p, q\}$, the set

$$([-1, 1] \times \mathbb{R}) \cap_{i\{-1,1\}, d\in\{up,down\}} R(f'(p^i_d), D[p^i_d, u], c')$$

has constant diameter. Thus the statement again follows.

Case 2: By Lemma 7 we can assume that the points p^i_{up}, p^i_{down}, and p (as well as p^j_{up}, p^j_{down}, and q) have the same x coordinates. Then we apply Lemma 8.

It remains only to prove Lemma 8.

Proof (of Lemma 8) Let $z_1 \in S_1$ and $z_2 \in S_2$ be any two points such that $\|z_1 - z_2\| = \max\{\|u - v\| : u \in S_1, v \in S_2\}$. Similarly, let $t_1 \in S_1$ and $t_2 \in S_2$ be any two points such that $\|t_1 - t_2\| = \min\{\|u - v\| : u \in S_1, v \in S_2\}$. Let y_p denote the y coordinate of point p. Without loss of generality, we can assume that $y_{z_1} < y_{z_2} < y_d$.

We claim that, if $y_{t_2} \leq y_{z_1}$, then $z_1 = t_1$. If $y_{z_1} < y_{t_1}$, then by decreasing y_{t_1}, we decrease $\|t_1 - t_2\|$. If $y_{z_1} > y_{t_1}$, then by decreasing y_{z_1}, we increase $\|z_1 - z_2\|$. Thus, $z_1 = t_1$, and in this case $\|z_1 - z_2\| - \|t_1 - t_2\| \leq 2s$.

It remains to analyze the case that $y_{t_2} > y_{z_1}$. In this case it is easy to see that, as long as $y_a < y_{z_1}$, we can increase y_a and decrease r_a such that t_2 and z_2 continue to belong to S_2. Therefore, without loss of generality, we can assume that $a = z_1$ and $r_a + s = \|z_1 - z_2\|$.

Similarly, we apply the same idea to d and t_1: we note that $y_{t_1} < y_{z_2}$ and, by decreasing y_d, we can assume that $d = z_2$ and $r_d + s = r_a + s = \|z_1 - z_2\|$. It is easy to see that, in this case (see Fig. 3), we have $\|t_1 - t_2\| \geq r_a - 3s = \|z_1 - z_2\| - 3s$. We conclude that $\|z_1 - z_2\| - \|t_1 - t_2\| \leq 3s$. □

4 Embedding with Order Type

In this section we consider the situation in which we are given all pairwise Euclidean distances between points in the plane as well as the “order type” of the points. The *order type* of a set of (labeled) points in the plane specifies, for each triple (p, q, r) of points, the *orientation* of that triple, i.e., whether visiting those points in order (forming a triangle) proceeds clockwise or counterclockwise (or, in degenerate cases, collinear).

While we present the initial algorithm assuming that we know the entire order type, we later relax the assumption to knowing only the orientations of all triples including a fixed point p. This relaxation reduces the amount of required extra information from $\binom{n}{3}$ orientations to $\binom{n-1}{2}$ orientations. In fact, this information is equivalent to knowing the counterclockwise order of points around point p.

Orientations can be very sensitive to small perturbations, and we are told only approximate information about the pairwise distances between points, so for orientations to be useful we need to know a range in which they apply. For a set of points in the plane, we call a set of triples of points *totally δ-robust* if perturbing the x and y coordinates of every point by at most $\pm\delta$ does not change the orientations of any of the triples in the set. A set of orientations is *δ-robust* if perturbing the x and y coordinates of any single point by at most $\pm\delta$ does not change the orientation of any of the triples in the set. Obviously, total δ-robustness implies δ-robustness, but in fact, the two notions are equivalent up to constant factors:

Lemma 9 *If a set of triples is 3δ-robust, then it is totally δ-robust.*

Proof If the set of triples is not totally δ-robust, there must be a perturbation of the points such that some triple (p, q, r) in the set changes orientation, i.e., p crosses the line segment between q and r. Because the total movement of p, q, and r in such a situation is at most 3δ, we can instead change the orientation of (p, q, r) by fixing q and r (and all other points except p) and just perturbing p by 3δ. But this contradicts the assumption that the set of triples is 3δ-robust. □

Our embedding algorithm assumes that the given orientations are totally δ-robust, for a particular choice of δ related to the distortion of the desired embedding. By Lemma 9, it suffices to assume that they are 3δ-robust. More precisely, the main theorem of this section is as follows:

Theorem 4 *Suppose that we are given a complete graph with specified edge lengths, and we are given an orientation for each triple of points involving one common point. Suppose we are promised that there is an embedding into the Euclidean plane with additive distortion ε in which these triples involving one common point have the specified orientations and are totally $c\varepsilon$-robust (or $3c\varepsilon$-robust). Then in polynomial time, multiplied by a factor of $O(\lg \Delta)$ if ε is not approximately known, we can compute an embedding f of the graph into the Euclidean plane with additive distortion at most $c\varepsilon$, for a global constant c.*

Proof First we guess ε up to a constant factor as in Section 3 by trying values of the form $\operatorname{diam}(D)/2^i$ for $i = 0, 1, 2, \ldots$, where $\operatorname{diam}(D)$ is the maximum distance in the given metric D. Then we apply Lemma 6 to guess the x coordinates of the vertices up to an additive $\pm 5\varepsilon$. By setting $c \geq 5$, robustness tells us that the orientations remain valid within this fixing of x coordinates. Also, changing the x coordinates of the promised embedding f according to this assignment increases the additive distortion by at most 10ε.

Next we show how to assign the y coordinates of f such that, for every pair (v, w) of vertices, $|D[v, w] - \|f(v) - f(w)\|_2| \leq 3\varepsilon$ (not counting the distortion introduced by fixing the x coordinates). Because the x coordinates are fixed, this constraint forces $f_y(v) - f_y(w)$ to lie within the union of (at most) two intervals, one interval for when $f_y(v) \geq f_y(w)$ and the other for when $f_y(v) \leq f_y(w)$. We show how to obtain the y coordinates by setting up a linear program, using the orientations to disambiguate between the two intervals.

We define a graph G whose vertex set is the same as the input graph. The edges of G are of two types: strong and weak. We connect vertices v and w by a *strong edge* in G if $D[v, w] \geq \sqrt{(f_x(v) - f_x(w))^2 + 3\varepsilon^2}$. We connect two points v and w by a *weak edge* in G if there are two points u_1 and u_2, connected via paths of strong edges to w but not to v, such that $D[v, w] > \sqrt{(f_x(v) - f_x(w))^2 + \varepsilon^2}$ and $f_x(u_1) \leq f_x(v) \leq f_x(u_2)$. The proofs of the following lemmas are very similar to Claims 4.1 and 4.2 of Bădoiu [Băd03] and hence are omitted.

Lemma 10 *No two connected components of G overlap in x extent; that is, there is a vertical line (not passing through any vertices) that separates the vertices of the first component from the vertices of the second component.*

Call an edge $\{v, w\}$ *oriented up* if $f_x(v) \leq f_x(w)$ and $f_y(v) \leq f_y(w)$, and call an edge $\{v, w\}$ *oriented down* if $f_x(v) \leq f_x(w)$ and $f_y(v) \geq f_y(w)$.

Lemma 11 *If we fix the orientation of an edge of G, we can uniquely determine the orientation of all other edges in the same connected component.*

By the definition of a strong edge, if there is no strong edge between two points v and w, the horizontal distance already fixed as $f_x(v) - f_x(w)$ is "good enough" for a 3ε-approximation. To ensure that the distortion remains sufficiently small, we form the constraint $D[v, w] + \varepsilon \geq \|f(v) - f(w)\|$, which is equivalent to the pair of linear constraints

$$\begin{aligned} & -\sqrt{(D[v, w] + \varepsilon)^2 - (f_x(v) - f_x(w))^2} \\ & \leq f_y(v) - f_y(w) \\ & \leq \sqrt{(D[v, w] + \varepsilon)^2 - (f_x(v) - f_x(w))^2}. \end{aligned}$$

For any edge $\{v, w\} \in E(G)$ that is oriented up and for which $f_x(v) \leq f_x(w)$, we form this linear constraint on f_y:

$$\begin{aligned}&\sqrt{(f_x(w) - f_x(v))^2 + (f_y(w) - f_y(v))^2} - \varepsilon \\ &\leq D[v, w] \\ &\leq \sqrt{(f_x(w) - f_x(v))^2 + (f_y(w) - f_y(v))^2} + \varepsilon,\end{aligned}$$

or equivalently,

$$\begin{aligned}&\sqrt{D[v, w]^2 - 2\varepsilon D[v, w] + \varepsilon^2 - (f_x(w) - f_x(v))^2} \\ &\leq f_y(w) - f_y(v) \\ &\leq \sqrt{D[v, w]^2 + 2\varepsilon D[v, w] + \varepsilon^2 - (f_x(w) - f_x(v))^2}.\end{aligned}$$

We have a similar relation for edges $\{v, w\} \in E(G)$ that are oriented down.

Now, using Lemmas 10 and 11 and the description above, we can obtain a $c\varepsilon$-approximation solution for the problem provided that G has only one connected component. However, if there are several connected components, each connected component can be oriented up or down, and the total number of cases can be exponential. Instead, we use the given orientations of triples to disambiguate the orientations of components. Because the orientations are totally $c\varepsilon$-robust, and we guess the x and y coordinates within $\varepsilon + 10\varepsilon + 3\varepsilon = 14\varepsilon$ total additive distortion (counting the ε distortion in f), the orientations remain correct if we set $c \geq 14$. Without loss of generality, we assume that the leftmost component is oriented up. Now consider a point v in this component. We show that, for each other component C, we can use orientations of triples involving v to determine whether C is oriented up or down. Consider a strong edge $(u, w) \in C$. (Such a strong edge should exist, because otherwise C has only one point and its orientation is trivial.) Because there is no strong edge between u and v, the segment connecting u to v is almost horizontal (see the definition of a strong edge). Using this property, using that (u, w) is a strong edge, and using the orientation of the triple (v, u, w), we can determine the orientation of edge (u, w) and thus by Lemma 11 the orientation of the whole component C. Thus, fixing the orientation of the leftmost component, we can determine the orientation of all edges of other components. Finally, by setting up the following linear program, we obtain the desired approximation embedding:

$$\begin{aligned}
&\sqrt{D[v,w]^2 - 2\varepsilon D[v,w] + \varepsilon^2 - (f_x(w) - f_x(v))^2} \\
&\le f_y(w) - f_y(v) \\
&\le \sqrt{D[v,w]^2 + 2\varepsilon D[v,w] + \varepsilon^2 - (f_x(w) - f_x(v))^2} \\
&\qquad\qquad\qquad \text{if } \{v,w\} \in E \text{ is oriented up and } f_x(w) \ge f_x(v),
\end{aligned}$$

$$\begin{aligned}
&\sqrt{D[v,w]^2 - 2\varepsilon D[v,w] + \varepsilon^2 - (f_x(w) - f_x(v))^2} \\
&\le f_y(v) - f_y(w) \\
&\le \sqrt{D[v,w]^2 + 2\varepsilon D[v,w] + \varepsilon^2 - (f_x(w) - f_x(v))^2} \\
&\qquad\qquad\qquad \text{if } \{v,w\} \in E \text{ is oriented down and } f_x(w) \ge f_x(v),
\end{aligned}$$

$$\begin{aligned}
&-\sqrt{(M[v,w] + \varepsilon)^2 - (f_x(v) - f_x(w))^2} \\
&\le f_y(v) - f_y(w) \\
&\le \sqrt{(D[v,w] + \varepsilon)^2 - (f_x(v) - f_x(w))^2} \\
&\qquad\qquad\qquad \text{if } \{v,w\} \notin E.
\end{aligned}$$

□

5 Embedding with Distribution Information

In this section we consider embedding the complete graph on n vertices into the Euclidean plane while approximately minimizing additive distortion of specified edge lengths that come from a kind of adversarial distribution. Roughly speaking, we are given the promise that the distances satisfy that, after perturbing each distance within $\pm\varepsilon$, the resulting distances are exactly the pairwise distances between n points sampled uniformly from the unit square $[0,1]^2$. More precisely, the specified distances come from first randomly sampling n points uniformly from the unit square, then exactly measuring their Euclidean distances, and then *adversarially* perturbing each distance within $\pm\varepsilon$. Our goal is to construct an embedding with additive distortion $O(\varepsilon)$.

Theorem 5 *There is a polynomial-time algorithm that, given a complete graph with edge lengths arising from the adversarial distribution described above, finds an embedding that has additive distortion $O(\varepsilon)$ with probability $1 - o(1)$, as long as $\varepsilon = \omega(1/\sqrt{n})$. The algorithm is deterministic; the probability is taken over the uniform sample of points in the unit square.*

Proof Let r be any value such that $r = \omega(1/\sqrt{n})$ and $r = O(\varepsilon)$. The algorithm first guesses a "frame" for the square, and then uses a "triangulation" approach relative to this frame:

1. For every quadruple (v_1, v_2, v_3, v_4) of vertices (the *frame*), construct the following embedding f:

(a) Embed v_i, $i \in \{1,2,3,4\}$, as follows: $f(v_1) = (0,0)$, $f(v_2) = (0,1)$, $f(v_3) = (1,1)$, and $f(v_4) = (1,0)$.

(b) Embed every other vertex w to an arbitrary point $f(w)$ in the region

$$R_w = [0,1]^2 \cap \bigcap_{i=1,2,3,4} R\left(f(v_i), D[v_i,w], \varepsilon + 2\sqrt{2}r\right),$$

where $R(p,r,\delta)$ is the annulus centered at point p with inner radius $r-\delta$ and outer radius $r+\delta$. If R_w is empty, we ignore this (incomplete) embedding and skip this iteration of the loop.

2. Report the embedding f with the smallest additive distortion.

This algorithm has the feature that every constructed embedding maps the vertices into the unit square. It remains to analyze the quality of the best embedding f found. Let f^* denote the uniformly random embedding into the unit square that we assume exists, and which has additive distortion ε.

We start by showing that there is a good choice of the frame. The following claim follows from basic calculations:

Claim With probability $1 - o(1)$, each of the four $r \times r$ subsquares of the unit square that each share a corner with the unit square contain $f^*(v)$ for some $v \in V$.

We condition on the event that there is at least one vertex v_1, v_2, v_3, and v_4 mapped via f^* to the lower-left, lower-right, upper-right, and upper-left corner subsquares, respectively. (By Claim 5, this event happens with probability $1-o(1)$.) Consider the iteration of Step 1 of the algorithm that chooses this quadruple of points for the frame. If we modify f^* by performing the assignment as in Step 1(a) of the algorithm, then the resulting embedding has additive distortion at most $\varepsilon + 2\sqrt{2}r$. Therefore, in this iteration, every region R_w includes $f^*(w)$ and is thus nonempty.

It suffices to show that the diameter of each set R_w is $O(\varepsilon + r)$. Consider any vertex w other than v_1, v_2, v_3, and v_4. We need the following claim, which can be proved using the same type of argument as in the proof of Theorem 4:

Claim Consider any two points $p, q \in [0,1]^2$ and any $r_1, r_2, \delta > 0$ such that $r_1, r_2 = O(\|p - q\|)$. The set $R(p, r_1, \delta) \cap R(q, r_2, \delta)$ is contained in a strip of width $O(\delta)$ whose direction (i.e., an infinite line contained in the strip) is orthogonal to the line passing through p and q.

Recall that R_p is an intersection of four annuli (and the unit square). Applying Claim 5 to the annuli around points $(0,0)$ and $(1,0)$, we conclude that R_p is contained in a vertical strip of width $O(\varepsilon + r)$. Applying Claim 5 to the annuli around points $(0,0)$ and $(0,1)$, we conclude that R_p is contained in a horizontal strip of the same width. It follows that the diameter of R_p is $O(\varepsilon + r)$ as claimed, and therefore that the additive distortion of the embedding f computed by the algorithm is $O(\varepsilon + r)$. □

6 Embedding with Range Graphs

In this section we are interested in embedding a graph with specified edge lengths into the line subject to the following condition. An embedding $f : V \to \mathbb{R}$ of a graph $G = (V, E)$ with edge lengths specified by D satisfies the *range condition* if, for every three points $p, q, r \in V$, (a) if $\{p, q\} \in E$ and $\{p, r\} \notin E$, $|f(p) - f(q)| \leq |f(p) - f(r)|$, and (b) if $\{p, q\}, \{p, r\} \in E$, $|f(p) - f(q)| \leq |f(p) - f(r)|$ precisely if $D[p, q] \leq D[p, r]$. Among all such embeddings, we will find one that minimizes the additive distortion with respect to the specified edge lengths on G. Part (b) of this definition will be satisfied provided the difference between adjacent distances in a near-optimal embedding is at least the additive distortion.

6.1 The Exact Case

In this subsection we consider embedding with zero distortion:

Lemma 12 *Given a graph G with edge lengths specified by D, we can check in polynomial time whether there is an embedding f that satisfies the range condition and matches D exactly on the edges of f, and construct such an embedding if it exists.*

Proof Without loss of generality we assume that the graph G is connected. Let p be the leftmost point in an embedding f into the line that satisfies the conditions of the lemma. We guess p by enumerating all $|V|$ possibilities. Without loss of generality, p has coordinate 0. All neighbors of p in G lie to the right of p. Let q be such a neighbor. Let r be a neighbor of q but not a neighbor of p. By the range condition, we have $|f(p) - f(r)| > |f(p) - f(q)|$. Therefore, $f(r) > f(q)$ and thus $f(r) = f(q) + D[q, r]$. By traversing G in a breadth-first manner, we can reconstruct f. The running time of our algorithm is $O(|V| \cdot |E|)$. □

6.2 The Additive Error Case

In this subsection we consider the case when the optimal embedding has minimum additive distortion ε. We say an edge $(p, q) \in G$ is a *forward edge* if $f(p) \leq f(q)$ and a *backward edge* if $f(p) > f(q)$. We call this distinction the *orientation* of an edge. Note that if (p, q) is a forward edge then (q, p) is a backward edge.

Lemma 13 *Given a graph G with edge lengths specified by D for which there is an embedding f that satisfies the range condition, and for any two incident edges $\{p, q\}$ and $\{q, r\}$ in G, we can determine the orientation of (q, r) in f given the orientation of (p, q) in f using just D.*

Proof Without loss of generality (p, q) is a forward edge and $D[p, q] > D[q, r]$. By part (b) of the range condition, if $D[p, r] < D[p, q]$, then (q, r) must be a backward edge. By both parts of the range condition, if $D[p, r] > D[p, q]$ or $D[p, r]$ is unknown, then (q, r) must be a forward edge. □

Theorem 6 *Given a graph G with edge lengths specified by D, we can construct in polynomial time an embedding f that satisfies the range condition and matches D up to the minimum possible additive distortion subject to the range condition.*

Proof Let (p, q) be an edge in G. Without loss of generality we can assume (p, q) is a forward edge. Lemma 13 implies that we know the orientation of all the incident edges. By applying this argument multiple times, we can determine the orientation of all the edges within the connected component of G containing p. We cannot determine the relative orientation between different connected components, but this is not necessary. By placing the locally embedded connected components far away from each other, the resulting embedding satisfies the range condition. Knowing the orientations, we can construct the following linear program which minimizes additive distortion:

$$\begin{aligned}
&\text{minimize} \quad \varepsilon \\
&\text{subject to } f(p) + D[p,q] - \varepsilon < f(q) < f(p) + D[p,q] + \varepsilon \\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{if } (p,q) \text{ is a forward edge,} \\
&\qquad\quad\;\; f(p) - D[p,q] - \varepsilon < f(q) < f(p) - D[p,q] + \varepsilon \\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{if } (p,q) \text{ is a backward edge.}
\end{aligned}$$

□

In Section 7 we show that embedding a graph with given edge lengths in two-dimensional ℓ_1 and ℓ_2 space, even using exact distances and a more restricted form of range-condition, is NP-hard.

7 Hardness Results

Saxe [Sax79] proved that deciding the embeddability of a given graph with exact ℓ_2 edge lengths is strongly NP-hard in d dimensions, for any $d \geq 1$. Independently, Yemini [Yem79] proved weak NP-hardness of the same problem for $d = 2$ with a simple reduction from Partition. Here we prove weak NP-hardness for both ℓ_1 and ℓ_2 in two dimensions, even when the graph satisfies the *constant-range condition*: two vertices v, w are connected by an edge precisely when their distance is at most a fixed range r. This condition is a special case of the *(variable) range condition* defined in Section 6, and hence our hardness results apply under that restriction as well. One interesting feature of our restricted form of the problem is that the problem is not hard in one dimension, and thus our proofs require us to use the structure of two dimensions. In contrast, previous hardness proofs start with 1D, and then trivially extend to higher dimensions.

7.1 ℓ_2 Case

Theorem 7 *It is NP-hard to decide whether a given graph with exact ℓ_2 edge lengths and satisfying the constant-range condition has an embedding with zero distortion.*

Our reduction from Partition is sketched in Fig. 4. The range is $1.1L$, where L is a large number to be chosen later. In any embedding of our graph, all vertices lie roughly on a square grid with edge lengths $L/2$. We use strips of k vertices spaced every $L/2$ units to build rigid bars of length $kL/2$; the strips are rigid because each vertex can see the next two vertices in the strips. We use right isosceles triangles with edge lengths $L/2$, $L/2$, and $L/\sqrt{2}$ to force angles of 90°. All other pairs of vertices have distance at least $\sqrt{5}/2 > 1.1L$, so are not within range.

For a given instance $a_1, a_2, \ldots, a_n$ of Partition, we construct $2n$ edges, two with length $(L + a_i)/2$ for each i, and force them all to be parallel. We choose L large enough so that $\sum_{i=1}^{n} a_i < 0.1L$. For each pair of edges of length $(L + a_i)/2$, we also create a pair of edges of length $L/2$, so that the absolute horizontal shift caused by these four edges is $(L + a_i) - L = a_i$. Each such quadruple of edges can be independently flipped so that the shift is either a_i or $-a_i$. Finally, we add another connection between the two extreme edges which forces the total shift to be zero. Thus, a distortion-free embedding corresponds to a solution to Partition and vice versa.

7.2 ℓ_1 and ℓ_∞ Case

We prove the first hardness result about embedding with exact ℓ_1 or ℓ_∞ distances:

Theorem 8 *It is NP-hard to decide whether a given graph with exact ℓ_∞ edge lengths (or, equivalently, exact ℓ_1 edge lengths) and satisfying the constant-range condition has an embedding with zero distortion.*

The proof is similar to the ℓ_2, except that the gadgets are slightly more complicated; see Fig. 5. The radius r is now exactly L. We use a sequence of attached $L \times L$ boxes in place of a strip of vertices. As before, this construction acts as a rigid bar, except that it can be flipped. (In Fig. 5, vertices p and q can be swapped.) To perturb a length by a_i from a multiple of L, we add a small $a_i \times a_i$ box and attach it in the middle of the strip. This box is in fact rigid and cannot be flipped with respect to its neighbors. Thus, the construction can be plugged into Fig. 4 and we have the theorem.

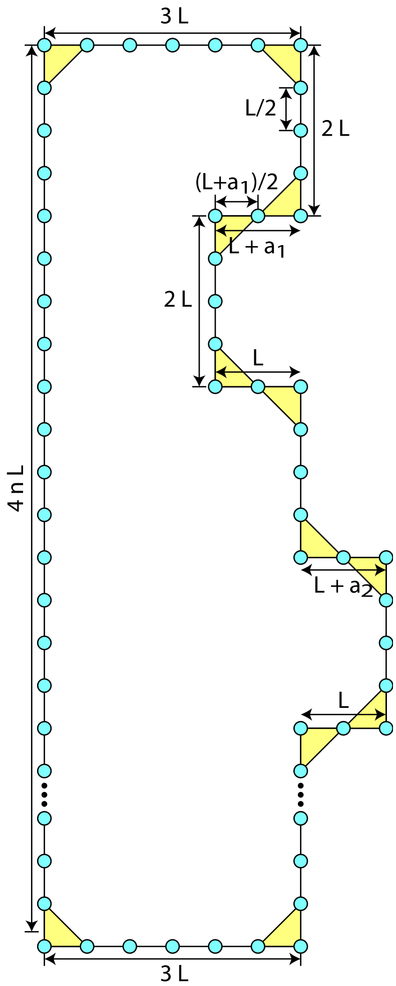


Fig. 4 Our reduction from Partition to ℓ_2 embedding of a graph satisfying the constant-range condition. In the reduction the a_i's are much smaller than L, and in this drawing the a_i's are drawn as zero.

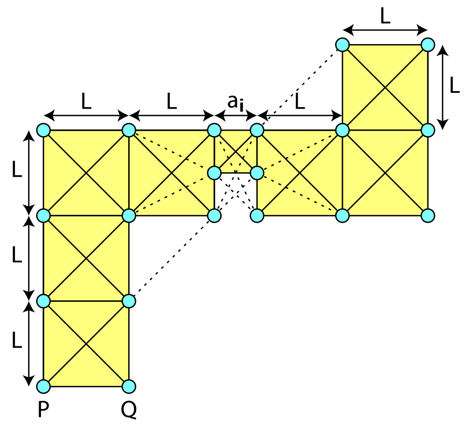


Fig. 5 Analogous gadgets for use in Fig. 4 for the ℓ_∞ case. Here a_i is drawn larger than reality. Dotted edges are present, but not necessary for rigidity.

8 Open Problems

An important open problem in this area is whether there is a polynomial-time algorithm for approximately minimizing additive distortion given all pairwise distance information and no extra information. Our pseudo-quasipolynomial-time algorithm is one step in this direction. The analogous problem for multiplicative distortion seems even harder.

A general theme of our work is to consider the case in which we do not know all distances. Another approach for making this case tractable is to constrain the connectivity to something less than $n - 1$ (for the complete graph). For example, what can we say about c-connected graphs for sufficiently large c, or cn-connected graphs for $c < 1$? These special cases will still likely require extra information, because even for the case where we know all pairwise distances, we do not know approximation algorithms without extra information except for ℓ_1 and additive distortion [Bǎd03].

It would seem natural to obtain angle estimates in a graph G "for free" using (approximate) distances in $G \cup G^2$, by analyzing triangles (p, q, r) in $G \cup G^2$. There are two problems with this approach. The first problem is that two vertices p, q in a triangle may be much closer to each other than to the third vertex r, and the multiplicative errors on distances allow p and q to spin around each other and allow p and q to have any angle. This problem can be surmounted by assuming that the ratio of lengths between any two incident edges is bounded. The second, more serious problem is that it is difficult to decode the orientations of triangles and hence the signs of the angles using purely distance information. We conjecture that this information can be decoded using distances in $G \cup G^2 \cup G^3 \cup G^4 \cup G^5 \cup G^6$, because 6-connected graphs have unique embeddings [JJ05].

Even with just distance information, the complexity of one interesting variation remains unresolved. Given a graph that is generically uniquely embeddable, in the sense that almost any assignment of edge lengths induces a unique embedding, can we construct the unique embedding for almost any assignment of edge lengths? Jackson and Jordán [JJ05] recently showed that, in polynomial time, we can test whether a graph has this property, but the proof is not entirely constructive. Another example of an NP-hard problem that can be solved in polynomial time almost always is Subset Sum. Our hardness reductions for embedding are based on Subset Sum, so there is hope that nongeneric examples are the only obstruction to polynomial-time algorithms.

In this paper we have focused on embedding metrics into the plane, but it would be natural to try to extend our work to slightly higher dimensions, in particular three dimensions which is important in some applications. Some of our results extend relatively easily. For example, in fixed dimension, given the approximate angle of every edge with respect to every coordinate axis (with additive error), and given distances with multiplicative error, we can apply the constant-factor approximation algorithms described in Sections 2.2 and 2.4.

Acknowledgments

We thank the anonymous referees for their helpful comments.

References

ABC⁺05. I. Abraham, Y. Bartal, T.-H. H. Chan, K. Dhamdhere, A. Gupta, J. Kleinberg, O. Neiman, and A. Slivkins. Metric embeddings with relaxed guarantees. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 83–100, 2005.

ABF⁺99. R. Agarwala, V. Bafna, M. Farach, M. Paterson, and M. Thorup. On the approximability of numerical taxonomy (fitting distances by tree metrics). *SIAM Journal on Computing*, 28(3):1073–1085, 1999.

Băd03. M. Bădoiu. Approximation algorithm for embedding metrics into a two-dimensional space. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 434–443, 2003.

BCIS05. M. Bădoiu, J. Chuzhoy, P. Indyk, and A. Sidiropoulos. Low-distortion embeddings of general metrics into the line. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 225–233, 2005.

BKL99. B. Berger, J. Kleinberg, and T. Leighton. Reconstructing a three-dimensional model with arbitrary errors. *Journal of the ACM*, 46(2):212–235, 1999.

CH88. G. M. Crippen and T. F. Havel. *Distance Geometry and Molecular Conformation.* John Wiley & Sons, 1988.

ČHH01. S. Čapkun, M. Hamdi, and J.-P. Hubaux. GPS-free positioning in mobile ad-hoc networks. In *Proceedings of the 34th Hawaii International Conference on System Sciences*, pages 3481–3490, January 2001.

CL92. C. Coullard and A. Lubiw. Distance visibility graphs. *Internatational Journal of Computational Geometry and Applications*, 2(4):349–362, 1992.

Con91. R. Connelly. On generic global rigidity. In P. Gritzman and B. Sturmfels, editors, *Applied Geometry and Discrete Mathematics: The Victor Klee Festschrift*, volume 4 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 147–155. AMS Press, 1991.

EHKN99. H. Everett, C. T. Hoàng, K. Kilakos, and M. Noy. Distance segment visibility graphs. Manuscript, 1999. http://www.loria.fr/~everett/publications/distance.html.

FK99. M. Farach and S. Kannan. Efficient algorithms for inverting evolution. *Journal of the ACM*, 46(4):437–449, 1999.

GRK04. A. Goel, S. Rai, and B. Krishnamachari. Sharp thresholds for monotone properties in random geometric graphs. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 580–586, 2004.

Hen92. B. Hendrickson. Conditions for unique graph realizations. *SIAM Journal on Computing*, 21(1):65–84, February 1992.

Hen95. B. Hendrickson. The molecule problem: Exploiting structure in global optimization. *SIAM Journal on Optimization*, 5:835–857, 1995.

HIL03. J. Håstad, L. Ivansson, and J. Lagergren. Fitting points on the real line and its application to RH mapping. *Journal of Algorithms*, 49(1):42–62, 2003.

Iva00. L. Ivansson. *Computational Aspects of Radiation Hybrid Mapping.* PhD thesis, Department of Numerical Analysis and Computer Science, Royal Institute of Technology, Stockholm, Sweden, 2000.

JJ05. B. Jackson and T. Jordán. Connected rigidity matroids and unique realizations of graphs. *Journal of Combinatorial Theory, Series B*, 94(1):1–29, 2005.

Kru64a. J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29:1–27, 1964.

Kru64b. J. B. Kruskal. Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, 29:115–129, 1964.

MIT. MIT CSAIL Networks and Mobile Systems group. The Cricket indoor location system. http://nms.csail.mit.edu/projects/cricket/.

PCB00. N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket location-support system. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pages 32–43, Boston, MA, August 2000.

PMBT01. N. B. Priyantha, A. K. L. Miu, H. Balakrishnan, and S. Teller. The Cricket compass for context-aware mobile applications. In *Proceedings of the 7th ACM International Conference on Mobile Computing and Networking*, pages 1–14, Rome, Italy, July 2001.

Sax79. J. B. Saxe. Embeddability of weighted graphs in k-space is strongly NP-hard. In *Proceedings of the 17th Allerton Conference on Communication, Control, and Computing*, pages 480–489, 1979.

Sax80. J. B. Saxe. Two papers on graph embedding problems. Technical Report CMU-CS-80-102, Department of Computer Science, Carnegie-Mellon University, Jan. 1980.

She62a. R. N. Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function 1. *Psychometrika*, 27:125–140, 1962.

She62b. R. N. Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function 2. *Psychometrika*, 27:216–246, 1962.

SRB01. C. Savarese, J. Rabaey, and J. Beutel. Locationing in distributed adhoc wireless sensor networks. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 2037–2040, Salt Lake City, UT, May 2001.

Wor. Working Group on Algorithms for Multidimensional Scaling. Algorithms for multidimensional scaling. http://dimacs.rutgers.edu/SpecialYears/2001_Data/Algorithms/MDSdescription.html.

Yem79. Y. Yemini. Some theoretical aspects of position-location problems. In *Proceedings of the 20th Annual IEEE Symposium on Foundations of Computer Science*, pages 1–8, 1979.