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 coordinates to the vertices that satisfy the given constraints up to a constant factor away from the best possible. We obtain the first subexponential-time (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 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.

Categories and Subject Descriptors: F.2.2 [Analysis of Algorithms and Problem Complexity]: Nonnumerical Algorithms and Problems—*geometrical problems and computations*; G.2.2 [Discrete Mathematics]: Graph Theory—*graph algorithms*

General Terms: Algorithms, Theory

Keywords: Graph embedding, metrics, angles, order type, distribution, range graphs, approximation algorithms

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 such measurements are approximate. In many applications, we would like to use this 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 [6, 8, 25, 20], rigidity theory [11, 5, 14], sensor networks [4, 19], and structural analysis of molecules [2, 7, 12]. The reconstruction problem arises frequently in several distributed physical structures such as the atoms in a protein [2, 7, 12] or the nodes in an ad-hoc wireless network [4, 19, 17].

A reconstruction is always unique 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 noncollinear located neighbors. More interesting is that such graphs include visibility graphs [6] and segment visibility graphs [8]. In general, however, the reconstruction problem is NP-hard [25], even in the strong sense [20]. It is also NP-hard to test whether a graph has a unique reconstruction [21, Sect. 6]. The uniqueness of a reconstruction in the "generic" case (in 2D) was recently shown to be testable in polynomial time by a simple characterization related to generic infinitesimal rigidity [11, 14], but this result has not yet lead 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 [22, 23], Kruskal [15, 16], and others. The area is usually called *multi-dimensional scaling* and is a subject of extensive research with several applications [24]. 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 the running time and/or embedding quality.

Recently, several papers [10, 13, 3, 1, 9] have presented algorithms for various versions of the embedding problem. These algorithms offer *provable* guarantees on the distortion of the computed embeddings. Unfortunately, the results suffer from two important limitations:

- None of the algorithms support embedding into the Euclidean plane. For example, the polynomial-time algorithm of [3] works only for additive-error embeddings into $\Re^2$ under the ℓ_1 norm.
- The algorithms assume that approximate distances between all pairs of points are specified. In some contexts, we have

*Supported in part by NSF under grant number ITR ANI-0205445.

SCG'04, June 8–11, 2004, Brooklyn, New York, USA.

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 reliable.

In this paper we attempt to overcome these difficulties. 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 quasipolynomial-time algorithm that uses no extra information. This algorithm is in fact the first subexponential-time algorithm for additive-distortion embedding into low-dimensional Euclidean space.

We consider the following types of extra information:

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

Extremum oracle: Suppose the x coordinates of the embedding are known (fixed). Let f be the 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 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 $p = \operatorname{argmin}_{p' : f_x(p') \in [x_l, x_r]} f_y(p')$ and $f(p)$, and symmetrically with argmax. Guessing this extra information is exactly what causes one of our algorithms to use 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 the metric is induced by random points in a square plus random noise.

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 [18] location system. In this system, several beacons are placed in a physical environment, and beacons can measure approximate distances (using a combination of ultrasonic and radio pulses) and, in some cases, approximate angles (using multiple ultrasound receivers). In this practical scenario, order type, distribution information, range constraints, and especially angle information are all reasonable assumptions to consider.

We show that any of the types of extra information described above often allow us to construct good embeddings with efficient algorithms. More precisely, we consider the problem of embedding a graph $G = (P, E)$ with specified lengths $D[p, q]$ for all edges $\{p, q\} \in E$, plus some additional information. Our goal is to embed G into ℓ_s^k space via a mapping $f : P \to \ell_s^k$ to either approximately minimize additive distortion $\max_{(p,q)\in E} |\, \|f(p) - f(q)\|_s - D[p, q]\, |$, or approximately minimize multiplicative distortion $\max_{(p,q)\in E} \|f(p) - f(q)\|_s / D[p, q]$ subject to $\|f(p) - f(q)\|_s \geq D[p, q]$ (non-contractiveness). In ℓ_s^k space, distances and lengths are measured according to the ℓ_s norm $\|(x_1, \ldots, x_k)\| = \sqrt[s]{x_1^s + \cdots + x_k^s}$.

We develop polynomial-time algorithms for the following versions of this embedding problem:

1. Embedding a general graph with approximate angle information into ℓ_s^2, $s \in \{1, 2, \infty\}$, with approximately optimal multiplicative error. If we are given the angle of each edge with respect to a fixed axis, or we are given angles between incident pairs of edges in the complete graph, our approximation factor is $O(1)$. If we are given angles between pairs of incident edges in a general graph, our approximation factor is $O(D)$ where D 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 the multiplicative error.

2. Embedding a complete graph into ℓ_2^2 with $O(1)$-approximate additive distortion in weakly 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 weakly quasipolynomial time.

   This algorithm is the first subexponential-time algorithm for minimizing the additive error of an embedding into a low-dimensional Euclidean space.

3. Embedding a complete graph into ℓ_2^2 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 ℓ_2^2 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 from the "designed distortion". 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 ℓ_p^2 for $p \in \{1, 2, \infty\}$ is NP-hard (Section 7). This problem was known to be NP-hard without range constraints in ℓ_2^d for all d [20].

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 that we know the angle between every edge and one fixed edge.

2.1 Different Types of Angle Information

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

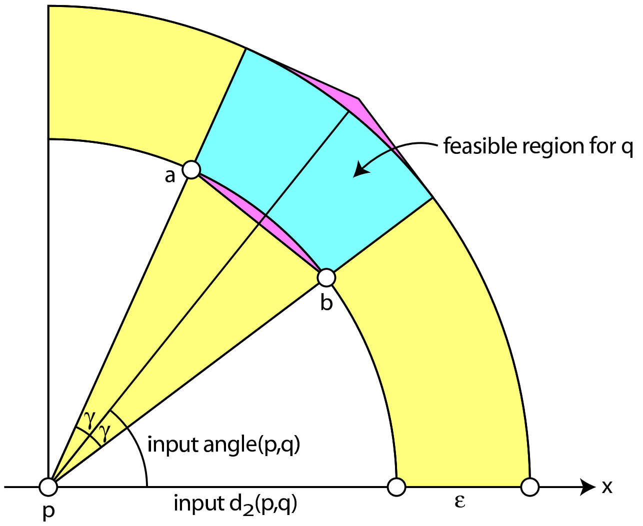


Figure 1. Feasible region of a point q with respect to p given the ℓ_2 distance within a multiplicative ε and given the angle to the x axis within an additive γ.

Proof: Fix one edge (p, q) and call it the x axis. To estimate the angle of an edge (v, w) with respect to the x axis, we consider the angles $\theta_1 = \angle pqv$ and $\theta_2 = \angle qvw$. If the angles were exact, the 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 angles between pairs of incident edges each with (one-sided or two-sided) additive error at most γ, we can compute the angle of every edge with respect to a particular edge with additive error at most $(D + 1)\gamma$ where D is the diameter of the graph.*

Proof: Similar to Lemma 1, except now we must combine angles along a path $p, q, \ldots, v, w$, which might have length up to $D + 2$, and therefore involves at most $D + 1$ 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 0, the error is $O(\sqrt{D})$ with high probability.

2.2 ℓ_2 Algorithm

Our algorithm sets up a constraint program for finding the coordinates of each vertex. The straightforward setup is nonconvex and difficult to compute. We relax the problem to be a convex program at the cost of some error. We further relax the problem to be a linear program at the cost of additional error.

The basic optimization problem has the following constraints. The distance and angle information of an edge (p, q) specifies a (nonconvex) feasible region for q given a proposed location for p. See Figure 1. If we know both ε and γ, our goal is to find a feasible solution. If only one of these error parameters (e.g., γ) is known, our objective function is to minimize the other error parameter (e.g., ε). If neither error parameter is known, we can solve the bicriterion version of the problem by minimizing one error parameter and binary searching on the other.

We can relax the feasible region to be convex by taking the convex hull. More precisely, we add one edge (a, b) to cut off the inner arc of the region; see Figure 1. This relaxation 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 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 relative error $(1+\varepsilon)/\cos\gamma - 1$.*

We can further relax the feasible region to be piecewise-linear by approximating the unique arc of the region with a polygonal chain. 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 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 relative 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 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.

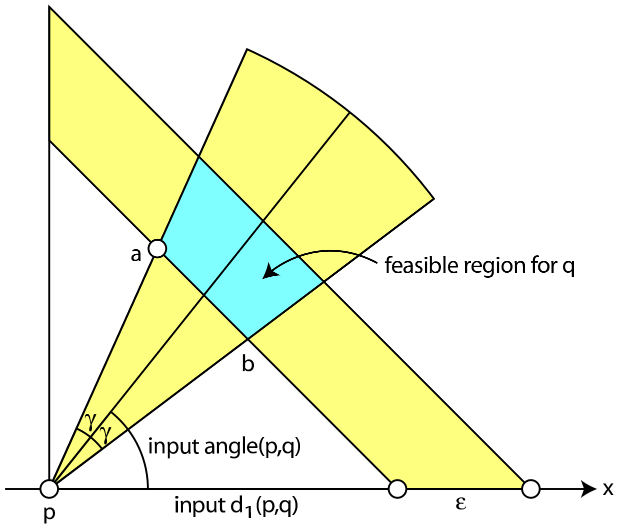
If our input specifies angles of edges with respect to the x axis, we are done. For other types of input, we can apply the following reductions:

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

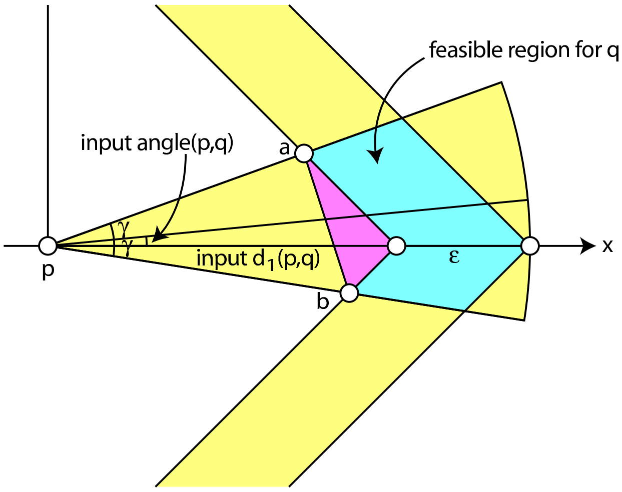
Proof: Apply Lemma 2 relative to the edge for which we know the angle with respect to the x axis, and translate using this angle. □

LEMMA 4. *Given a graph, and given 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 angles relative to the x axis with additive error at most $\gamma + \gamma'$.*

Proof: Apply Lemma 2 to obtain angles relative to an edge e, and then "guess" the angle of the x axis with respect to e among the $2\pi\lceil 1/\gamma' \rceil$ angles of the form $0, \lceil 1/\gamma' \rceil, 2\lceil 1/\gamma' \rceil, \ldots$. □



(a) Convex case



(b) Nonconvex case

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

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 feasible 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 Figure 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 Figure 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 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 relative error*

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

If we are given the approximate angles between incident pairs of edges, and the approximate 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 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.

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.

2.6 Extension to Higher Dimensions

Our embedding algorithms extend to d dimensions for fixed d. The input now becomes the approximate length of every edge and the approximate angle of every edge with respect to every coordinate axis. (Lemmas 3 and 4 can no longer obtain this information from just the angles between pairs of edges.) These angles determine, for each edge (p, q), a cone with origin at p specifying the region of q that satisfies the angular constraints. The approximate length information intersects this cone with a ball and the complement of a ball (a higher-dimensional annulus); the resulting feasible region for q with respect to p is nonconvex as before. The rest of the algorithm proceeds as before; the error bounds from convexifying and polyhedralifying the feasible regions grow with d. In the end, we obtain a constant-factor approximation, where the approximation factor depends on ε, γ, and d.

3. ADDITIVE EMBEDDING INTO THE PLANE

In this section we describe an $O(1)$-approximate algorithm for embedding a metric into ℓ_2^2 that minimizes (approximately) the additive error. The algorithm assumes that the minimum additive error t is known. In fact, knowing a constant-factor approximation to

t is good enough. This assumption can be satisfied in a standard way by trying out several values of t of the form diameter$(P)/2^i$ for $i = 0, 1, \ldots$. We also assume that the ratio of the diameter of $f(P)$ to t is bounded by Δ, called the *spread* of the metric. For simplicity, we assume that we know t exactly, and without loss of generality, we assume that $t = 1$.

The algorithm runs in polynomial time given the extremum oracle. By exhaustive enumeration of the possible oracle answers, it can be converted to a standard algorithm with running time $2^{O(\log n \cdot \log^2 \Delta)}$.

We use the following notation. $B(p, r)$ denotes a ball of radius r around point p. $R(p, r, s)$ denotes the ring (or annulus) $B(p, r + s) - B(p, r - s)$.

In the first step, the algorithm guesses the diameter pair (i.e., we run $\Theta(n^2)$ copies of the algorithm for each pair of points). Denote the diameter pairs by o and o'. By rotation, we can assume $f(o) = (0, 0)$ and $f(o') = (\Delta, 0)$. Consider any other point q. Let $S(q) = R(f(o), D(o, q), 1) \cap R(f(o'), D(o', q), 1)$ (in the following, we will simply refer to this set as S). Clearly, $f(q)$ must lie in the set S. At the same time, S is "narrow", i.e., there is a value x_q so that S is contained in the strip $[x_q - c_1, x_q + c_1] \times \Re$, for some constant c_1. Thus, if we fix the x-coordinates of the images of q to x_q, then there exists an embedding f' that $f'_x(q) = x_q$, whose additive error is at most $2c_1$. In addition, we require that $f'_y(q)$ is a multiple of t (i.e., an integer if $t = 1$); this increases the additive error by at most t. The goal of the remaining part of the algorithm is to find a mapping g with at most the error of f', assuming we know $g_x(q) = f'_x(q)$ for all $q \in P$. For simplicity, we will assume that all x-coordinates are different; this can be easily achieved by a small perturbation.

The algorithm is based on the divide and conquer paradigm. Firstly, we compute the median x of the x-coordinates of points in $g(P)$. Let P^+ be the set of all points $p \in P$ such that $g_x(p) > x$, and let $P^- = P - P^+$. The algorithm proceeds by creating the set of *constraints* on $g(P^+)$ and $g(P^-)$. The constraints have two properties:

- They are feasible, i.e., f' satisfies them
- For any mapping g satisfying those constraints, we have $|g(p) - g(q)| = D(p, q) \pm c$, for all $p \in P^+$, $q \in P^-$; here c is a certain fixed constant.

This allows us to compute $g(P^+)$ and $g(P^-)$ (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 $I_i = (x + 2^{i-1} - 1, x + 2^i - 1]$; if $i < 0$, define $I_i = -I_{-i}$. For each I_i, the algorithm queries the extremum oracle to obtain the point $p^i_{up} \in P$, $f'_x(p^i_{up}) \in I_i$, such that $f'_y(p_{up})$ is maximal. 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.

The oracle's answers can be implemented by exploting all choices of the guessed variables. The total number of such choices is bounded by $2^{O(\log^2 \Delta)}$, since there are at most $O(\Delta)$ different potential values of y coordinates of $f'(\cdot)$.

After making the guesses, the algorithm imposes the following new constraints, for each i, $d \in \{up, down\}$, and $p \in P$:

- "$g_y(p^i_d) = f'_y(p^i_d)$"
- if $f'_x(p) \in I_i$, then "$g_y(p) \in [f'_y(p^i_{down}), f'_y(p^i_{up})]$"
- "$g(p) \in R(f'(p^i_d), D(p^i_d, p), 2c_1)$"; note that the latter condition can be expressed as a restriction on $g_y(p)$

As mentioned above, after imposing the constraints, the algorithm recurses to find $g(P^+)$ and $g(P^-)$ 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).

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 $g(P^+)$ and $g(P^-)$ satisfying the constraints are found, then we have $|g(p) - g(q)| = D(p, q) \pm c$, for all $p \in P^+$, $q \in P^-$. This is done via the following two claims.

CLAIM 1. *Consider any two points $a = (x, y)$ and $b = (x', y')$, such that $x' \geq x/2$. Define $b' = (x', y)$ and $I = \{0\} \times \Re$. Then, for any r there exists r' such that $I \cap R(a, r, 2c_1) \subset R(b', r', c)$ for a fixed constant c.*

The interpretation (and usage) of this claim 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 it is not the case that $i = -1, j = 1$ (we will take care of this case later). In the procedure described above, we impose constraints on $g(p)$ of the form $g(p) \in R(a, r, 2c_1)$, for $d \in \{down, up\}$, $r = D(p^j_d, p)$, $a = f'(p^j_d)$. However, it will be more convenient to consider a different constraint, namely $g(p) \in R(b', r', c)$, where $b' = (f'_x(q), f'_y(p^j_d))$, since in this way $f'(q)$ and b' have the same x-coordinate, which will be used in the next claim. However, we do not know $f'(q)$, so we cannot impose the second constraint explicitely. Fortunately, the above claim 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, I_j.

Proof (of Claim 1): Without loss of generality we can assume that $I \cap R(a, r, 2c_1)$ is nonempty. In addition, we assume that it consists of two disconnected components (if it consists of only one component, the proof is similar). Denote the upper component (with higher values of y-coordinates) by $Y = \{0\} \times [y_d, y_u]$. Let $q_d = (0, y_d)$, $q_u = (0, y_u)$. Note that $y_u^2 + x^2 = (r + 2c_1)^2$, and $y_d^2 + x^2 = (r - 2c_1)^2$. 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

$$x'^2 + (r - 2c_1)^2 - x^2 \geq [(r + 2c_1)^2 - x^2] + x'^2 - 2r'c + c^2$$

which simplifies to $2r'c - c^2 \geq 4c_1 r$.

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 $c \geq 8c_1$. This proves the claim. □

Now we need the second claim.

Consider the following configuration of points $a = (0, y_a)$, $b = (0, y_b)$, $c = (x, y_c)$, $d = (x, y_d)$.

For any r_a, r_b, r_c, r_d, define two sets:

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

CLAIM 2. *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* $4s$.

Note that this implies that for any points $p \in P^-$, $q \in P^+$ that satisfy the imposed constraints, we have $|g(p) - g(q)| = |f'(p) - f'(q)| \pm O(1)$. To show that, consider two cases. Let $f'_x(p) \in I_i$, $f'_x(q) \in I_j$.

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 $y_{up} - y_{down} \le c_2$ for c_2 larger than, say, $10c_1$, then the statement follows. Otherwise, if $y_{up} - y_{down} > 10c_1$, then for any $u \in \{p, q\}$, the set

$$([-1,1] \times \Re) \cap_{i\{-1,1\}, d\in\{up,down\}} R(f'(p^i_d), D(p^i_d, u), 2c_1)$$

has constant diameter. Thus the statement again follows.

Case 2: By Claim 1 we can assume that the points p^i_{up}, p^i_{down}, p (as well as p^j_{up}, p^j_{down}, q) have the same x coordinates. Then we apply Claim 2.

Proof (of Claim 2): Let $z_1 \in S_1$, $z_2 \in S_2$ be any two points such that $|z_1 - z_2| = \max\{|u - v| : u \in S_1, v \in S_2\}$. Similarly, let $t_1 \in S_1, t_2 \in S_2$ be any two points such that $|t_1 - t_2| = \min\{|u - v| : u \in S_1, v \in S_2\}$.

Without loss of generality we can assume that $y_{z1} < y_{z2} < y_d$.

Note that if $y_{t2} < y_a$ then by considering $t_3 = (x, 2y_a - y_d)$ we get $|t_1 - t_3| < |t_1 - t_2|$ and we can consider only the case when $y_{t2} \ge y_a$. Also, if $y_{t2} < y_{z1}$ by increasing y_{t2} we decrease $|t_1 - t_2|$ so we can assume $y_{t2} > y_{z1}$.

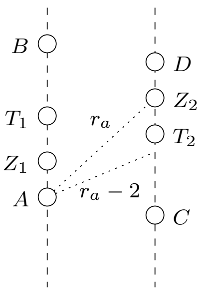


Figure 3. Illustration of the proof.

It is easy to see that as long as $y_a < y_{z1}$ we can increase y_a and decrease r_a such that t_2 and z_2 will continue to belong to S_2. Therefore, without loss of generality we can assume $a = z_1$ and $r_a + s = |z_1 - z_2|$.

Similarly, we apply the same idea to d and t_1: we note that $y_{t1} < y_{z2}$ 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 Figure 3) we have $|t_1 - t_2| \ge r_a - 3s = |z_1 - z_2| - 3s$. □

4. EMBEDDING WITH ORDER TYPE

In this section, we consider the model in which we are given all distances in ℓ_2, and in addition *order type* of the points or in other words *orientations* of all triples. Here *orientations* of three points p, q, and r is a triple (p, q, r) which says we see these points in this order in triangle pqr, when we start from p and turn clockwise. In fact, we present the algorithm when we have orientations of all triples, however we relax the condition such that knowing only orientations of all triples which have a point p in common is sufficient (i.e., knowing $\binom{n-1}{2}$ orientations instead of $\binom{n}{3}$ ones.) In fact, knowing this information is equivalent to knowing the order of points around point p according to their angles with a fixed axis. In addition, we assume that the given orientations are *robust*.

DEFINITION 1. *A set of orientations is called* δ-totally robust *if perturbing the x and y coordinates of all points by at most δ does not change the given orientations. A set of orientations is called* δ-robust *if perturbing the x and y coordinates of a single point by at most δ does not change the orientations.*

We note that robustness is weaker than totally robustness. However, Lemma 4 shows totally robustness and robustness are equivalent up to a constant factor.

LEMMA 5. *If a set of orientations is 3δ-robust, then it is δ-totally robust.*

Proof: One can easily observe that if the set of orientations are not totally robust, there should exist a situation in which a point p passes a line segment between q and r. Since the total movement in such a situation is at most 3δ, one can fix q and r and just perturb p by 3δ and still changes the orientations. Thus we reach to a contradiction. □

In the rest of the proof, we mainly use totally robustness. However since by Lemma 5 totally robustness can be further relaxed to robustness by losing a constant factor, we use these words interchangeably. The main theorem of this section is as follows:

THEOREM 4. *Given a complete graph, given the length of each edge with additive error ε, and given a set of $3c\varepsilon$-robust orientations of every set of three points, we can compute in polynomial time an ℓ_2 embedding with additive distortion $c\varepsilon$ for some constant c.*

By setting the robustness high enough, we can guarantee that our embedding satisfies all specified orientation constraints. Also, our proof uses the orientations of only one point with respect to all other points. Thus, knowing only the orientations of all triples involving a common point is sufficient.

Proof: We extend the algorithm of Bădoiu [3] for embedding in ℓ_1. Similar to [3], we guess ε by binary search and thus we can assume that we know ε. Next we show how to guess the x coordinates within a constant factor of ε, which is enough for this result. The main idea of this step is that we first guess p and q denoting the diameter and without loss of generality we take $p = (0,0)$ and $q = (D[p,q], 0)$. For any other point v, let R_v be the intersection of the annulus between balls of radius $D[p,v] - \varepsilon$ and $D[p,v] + \varepsilon$ around p and the annulus between balls of radius $D[v,q] - \varepsilon$ and $D[v,q] + \varepsilon$ around q. One can easily observe that v should lie in R_v and the x-coordinates of points in R_v are contained in the interval of $[x_v - c'\varepsilon, x_v + c'\varepsilon]$ for some constant $c' \le 4$. (Intuitively, region R is narrow since p and q form a diameter and the worst case happens when q and v and p and v also form diameters.) This step is similar to a step in Section 3 and in an algorithm of Bădoiu [3]. Now by fixing $f_x(v) = x_v$, we have the additive error at most $2c'\varepsilon$. In the rest of the algorithm, we fix these x-coordinates and assume that they are all distinct (otherwise perturb them a bit), and then we try to obtain the y coordinates of the points. Also, since orientations are robust, fixing the x-coordinates dose not prevent us to use orientations to obtain the y-coordinates.

Having the x coordinates we will find $f_y(v)$ for each point v, such that $|D[p,q] - ||f(p) - f(q)||_2| \le 3\varepsilon$. We note that here since x coordinates are fixed, $f_y(p) - f_y(q)$ can take at most two intervals (which depends whether $f_y(p) \ge f_y(q)$ or not.) Our final goal is to obtain the y coordinates by setting up a linear program.

First, we define a graph whose vertex set is the set of points as follows. We connect p and q by a *strong edge* if $D[p,q] \ge$

$\sqrt{(f_x(p)-f_x(q))^2+3\varepsilon^2}$. Also, we connect two points v and w by a *weak edge* if there exists p and q in the same connected component as w of strong edges and v is not in the same connected component as w and $D[v,w] > \sqrt{(f_x(v)-f_x(w))^2+\varepsilon^2}$ and $f_x(p) \leq f_x(v) \leq f_x(q)$. The set of edges of G is the union of all strong and weak edges. We say an edge $\{p,q\}$ is *oriented up* if $f_x(p) \leq f_x(q)$ and $f_y(p) \leq f_y(q)$. Similarly, we say an edge $\{p,q\}$ is *oriented down* if $f_x(p) \leq f_x(q)$ and $f_y(p) \geq f_y(q)$. The proof of the following lemmas is very similar to Claims 4.1 and 4.2 of Badoiu [3] and hence omitted.

LEMMA 6. *No two connected components of G overlap, i.e., there is a vertical line l not passing through any given points that separates the vertices of the first component from the vertices of the second component.*

LEMMA 7. *If we fix the orientation of an edge of G, we can uniquely determine the orientation of all other edges in the same connected components.*

One can easily observe that if there is no strong edge between two points p and q, the distance already given by $f_x(p)-f_x(q)$ is good enough for a 3ε-approximation. Then we can add the constraint $D[p,q]+\varepsilon \geq ||f(p)-f(q)||$, which is equivalent to

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

to make sure that the error is not too much. In addition, for an edge $(v,w) \in E(G)$ which is oriented up and $f_x(v) \leq f_x(w)$, we have this linear constraint on f_y.

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

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 if $(v,w) \in E(G)$ is oriented down.

Now, using Lemmas 6 and 7 and the description above we can obtain a $c\varepsilon$-approximation solution for the problem, if G' has only one connected component. However if there are several connected components, each connected component can be oriented up or down and then the number of cases can be exponential. For this case, we use the given orientations. We note that since the orientations are $3c\varepsilon$-robust (and thus $c\varepsilon$-totally robust by Lemma 5), and we guessed the x and y coordinates within some $2c'\varepsilon$ error for $2c' \leq c$, still we can use the orientations. Without loss of generality, we can assume the left-most component is oriented up. Now consider a point v in this component. We show that for each other component C, using orientations, we can determine whether C is oriented up or down. Consider a strong edge $(u,w) \in C$ (there should exist such a strong edge, since otherwise C has only one point and its orientation is trivial.). Since there is no strong edge between v and u, it means segment v and u is almost horizontal (see the definition of strong edges). Using this property and the fact that (u,w) is a strong edge and also, we know the orientations of three points v,u,w, we can determine the orientation of (u,w) and thus 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 LP, we obtain the desired approximation embedding.

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

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

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

□

5. EMBEDDING WITH DISTRIBUTION INFORMATION

In this section we consider embedding methods for complete graphs with given edge lengths that are chosen at random from certain distribution. In particular, we show a polynomial-time algorithm that finds an embedding f into $\Re^2$ under ℓ_2 such that the additive distortion of f is within a constant factor away from the "designed" distortion.

The given edge lengths D is chosen via the following random process. Let P be a set, $|P|=n$, and t be the "designed distortion". We first choose the "designed optimal embedding" $f^* : P \to [0,1]^2$. This is done by choosing $f^*(p), p \in P$, to be a point chosen at random from a uniform distribution over $S=[0,1]^2$. Then, each $D[p,q]$ is chosen such that $|D[p,q]-||f^*(p)-f^*(q)||| \leq t$. Note that f^* provides an embedding of (P,D) into $[0,1]^2$ that has an additive distortion t. The goal of the algorithm is: given (P,D), find an embedding f such that the additive distortion of f is $O(t)$.

THEOREM 5. *There is a polynomial time algorithm that, given a complete graph with edge lengths constructed as above, finds an embedding f that has an additive distortion of $O(t)$ with probability $1-o(1)$, as long as $t=\omega(1/\sqrt{n})$. The algorithm is deterministic; the probability is taken over the space of mappings f^*.*

Let r be such that $r=\omega(1/\sqrt{n})$ and $r=O(t)$. The algorithm uses the "triangulation" approach:

1. For each sequence $p_1,p_2,p_3,p_4 \in P$ do:
   (a) Assign $f(p_1)=(0,0), f(p_2)=(0,1), f(p_3)=(1,1), f(p_4)=(1,0)$
   (b) For each $p \in P-\{p_1,p_2,p_3,p_4\}$, choose $f(p)$ to be an arbitrary point in the region

   $$R_p = \cap_{i=1\ldots4} R(f(p_i), D[p,p_i], t+2\sqrt{2}r)$$

   where $R(p,r,w)$ is an annulus centered at p, with inner radius $r-w$ and outer radius $r+w$. If $R_p \cap S$ is empty, ignore the (incomplete) embedding.

2. Report the embedding with the smallest additive distortion

The following claim follows from basic calculations.

CLAIM 3. *With probability $1 - o(1)$, each of the four $r \times r$ sub-squares of S that each touches a corner of S contain $f^*(p)$ for some $p \in P$.*

From the claim it follows that, with probability $1 - o(1)$, there exist $p_1 \dots p_4$ with the above property. Consider the case when the algorithm chooses that set of points (step (1) of the code). It follows that, if we modify f^* into f by performing the assignment as in step (a) of the algorithm, then f has distortion at most $t + 2\sqrt{2}r$. This implies that, in this case, all regions $R_p \cap S$ are non-empty.

It suffices to show that the diameter of each set $R_p \cap S$ is $O(t+r)$. Consider any $p \in P - \{p_1, p_2, p_3, p_4\}$.

The following claim follows from the argument as in the proof of Theorem 4.

CLAIM 4. *Consider any $p, q \in S$ and $r_1, r_2, w > 0$ such that $r_1, r_2 = O(|p-q|)$. The set $R(p, r_1, w) \cap R(q, r_2, w)$ is contained in a strip of width $O(w)$.*

Recall that R_p is an intersection of four annuli. By applying Claim 4 to annuli around points $(0, 0)$ and $(0, 1)$, we conclude that R_p is contained in a vertical strip of width $O(t + r)$. In the same way, we conclude that R_p is contained in a in horizontal strip of the same width. It follows that that the diameter of R_p is $O(t + r)$ as claimed.

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 $f : P \rightarrow \Re$ of a graph $G = (P, E)$ with edge lengths specified by D satisfies the *range condition* if, for every three points $p, q, r \in P$, (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 8. *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 $|P|$ 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 $O(|P| \cdot |E|)$. □

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 9. *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 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 9 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:

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

□

In Section 7, we show that embedding a graph with given edge lengths in ℓ_2^2 and ℓ_1^2, even using exact distances and a more restricted form of range-condition, is NP-hard.

7. HARDNESS RESULTS

Saxe [20] proved that deciding 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 [25] 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 2D, 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 1D, and thus our proofs require us to use the structure of 2D. In contrast, previous hardness proofs start with 1D, and then trivially extend to higher dimensions.

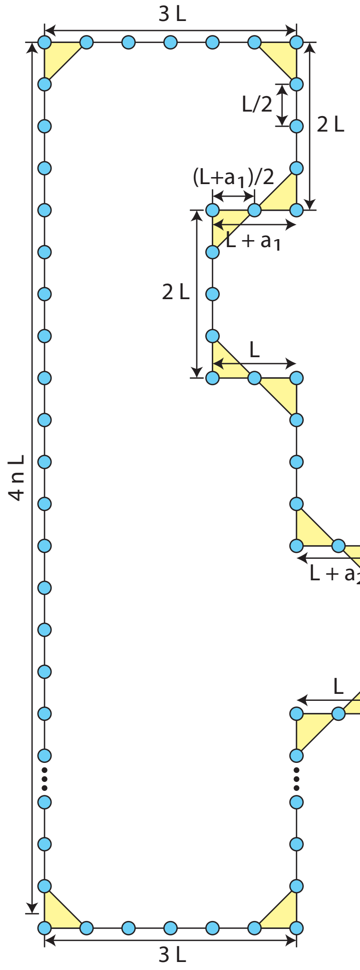


Figure 4. Our reduction from Partition to ℓ_2 embedding of a graph satisfying the constant-range condition. In the reduction, the a_i's are much smaller than L, and in this drawing, the a_i's are drawn as 0.

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 Figure 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 0. Thus, a distortion-free embedding corresponds to a solution to Partition and vice versa.

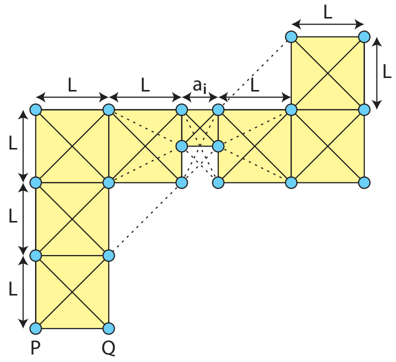


Figure 5. Analogous gadgets for use in Figure 4 for the ℓ_∞ case. Here a_i is drawn larger than reality. Dotted edges are present, but not necessary for rigidity.

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 Figure 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 Figure 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 Figure 4 and we have the theorem.

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 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 error [3].

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 [14].

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 [14] 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.

References

[1] R. Agarwala, V. Bafna, M. Farach-Colton, B. Narayanan, M. Paterson, and M. Thorup, *On the approximability of numerical taxonomy: (fitting distances by tree metrics)*, 7th Symposium on Discrete Algorithms, (1996).

[2] B. Berger, J. Kleinberg, and T. Leighton, *Reconstructing a three-dimensional model with arbitrary errors*, in Proc. 28th Annu. ACM Sympos. Theory Comput., May 1996, pp. 449–458.

[3] M. Bădoiu, *Approximation algorithm for embedding metrics into a two-dimensional space*, 14th Annual ACM-SIAM Symposium on Discrete Algorithms, (2003).

[4] 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, January 2001, pp. 3481–3490.

[5] R. Connelly, *On generic global rigidity*, in Applied Geometry and Discrete Mathematics: The Victor Klee Festschrift, P. Gritzman and B. Sturmfels, eds., vol. 4 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science, AMS Press, 1991, pp. 147–155.

[6] C. Coullard and A. Lubiw, *Distance visibility graphs*, Internat. J. Comput. Geom. Appl., 2 (1992), pp. 349–362.

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

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

[9] M. Farach-Colton and S. Kannan, *Efficient algorithms for inverting evolution*, 28th Symposium on Theory of Computing, (1996).

[10] J. Hastad, L. Ivansson, and J. Lagergren, *Fitting points on the real line and its application to rh mapping*, Lecture Notes in Computer Science, 1461 (1998), pp. 465–467.

[11] B. Hendrickson, *Conditions for unique graph realizations*, SIAM J. Comput., 21 (1992), pp. 65–84.

[12] ———, *The molecule problem: Exploiting structure in global optimization*, SIAM J. on Optimization, 5 (1995), pp. 835–857.

[13] L. Ivansson, *Computational aspects of radiation hybrid*, Doctoral Dissertation, Department of Numerical Analysis and Computer Science, Royal Institute of Technology, (2000).

[14] B. Jackson and T. Jordán, *Connected rigidity matroids and unique realizations of graphs*. Manuscript, March 2003.

[15] J. Kruskal, *Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis*, Psychometrika, 29 (1964), pp. 1–27.

[16] ———, *Nonmetric multidimensional scaling: A numerical method*, Psychometrika, 29 (1964), pp. 115–129.

[17] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, *The Cricket location-support system*, in Proceedings of 6th Annual International Conference on Mobile Computing and Networking, Boston, MA, August 2000, pp. 32–43.

[18] 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, Rome, Italy, July 2001, pp. 1–14.

[19] C. Savarese, J. Rabaey, and J. Beutel, *Locationing in distributed ad-hoc wireless sensor networks*, in Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, Salt Lake City, UT, May 2001, pp. 2037–2040.

[20] J. B. Saxe, *Embeddability of weighted graphs in k-space is strongly NP-hard*, in Proc. 17th Allerton Conf. Commun. Control Comput., 1979, pp. 480–489.

[21] J. B. Saxe, *Two papers on graph embedding problems*, Tech. Rep. CMU-CS-80-102, Department of Computer Science, Carnegie-Mellon University, Jan. 1980.

[22] R. N. Shepard, *The analysis of proximities: Multidimensional scaling with an unknown distance function 1*, Psychometrika, 27 (1962), pp. 125–140.

[23] ———, *The analysis of proximities: Multidimensional scaling with an unknown distance function 2*, Psychometrika, 27 (1962), pp. 216–246.

[24] Working Group on Algorithms for Multidimensional Scaling, *Algorithms for multidimensional scaling*. http://dimacs.rutgers.edu/SpecialYears/2001_Data/Algorithms/MDSdescription.html.

[25] Y. Yemini, *Some theoretical aspects of position-location problems*, in Proc. 20th Annu. IEEE Sympos. Found. Comput. Sci., 1979, pp. 1–8.