Solving Distance Geometry Problem with Inexact Distances in Integer Plane

Piyush K. Bhunre(✉), Partha Bhowmick, and Jayanta Mukhopadhyay

Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, India
kbpiyush@gmail.com, {pb,jay}@cse.iitkgp.ernet.in

Abstract. Given the pairwise distances for a set of unknown points in a known metric space, the distance geometry problem (DGP) is to compute the point coordinates in conformation with the distance constraints. It is a well-known problem in the Euclidean space, has several variations, finds many applications, and so has been attempted by different researchers from time to time. However, to the best of our knowledge, it is not yet fully addressed to its merit, especially in the discrete space. Hence, in this paper we introduce a novel variant of DGP where the pairwise distance between every two unknown points is given a tolerance zone with the objective of finding the solution as a collection of integer points. The solution is based on characterization of different types of annulus intersection, their equivalence, and cardinality bounds of integer points. Necessary implementation details and useful heuristics make it attractive for practical applications in the discrete space.

1 Introduction

Given the inter-point distances for a finite set of points with unknown coordinates in a particular metric space, the *distance geometry problem* (DGP) is to embed the points in that space so as to satisfy the distance constraints. The basic problem in the Euclidean space was first introduced by Menger in 1920s, and later formally established by Blumenthal in 1950s [2,14,17]. The general version of the problem is as follows. Given a positive integer k and a simple undirected weighted graph $G = (V, E)$ with weight function $w : E \mapsto \mathbb{R}^+$, determine if there is a function $f : V \mapsto \mathbb{R}^k$ such that for all $(u, v) \in E$, $\|f(u) - f(v)\| = w(u, v)$, where $\|\cdot\|$ denotes the Euclidean norm in k-dimensional space. The problem is NP-complete for $k = 1$ (i.e., embedding in real line) and NP-hard for $k > 1$ [19]. Following are the variations of DGP as per the input distance set.

1. *Exact Distances:* The point-set embedding should be such that the corresponding distance set exactly matches the input distance set. Here all the distances are given, and so the distance set is a complete set.
2. *Inexact or Bounded Distances:* Each pairwise distance in the solution need not be exact, but lies in an interval.
3. *Sparse Distances:* The distance set is incomplete; in addition, the distances could be of exact or of bounded type.

A. Bac and J.-L. Mari (Eds.): CTIC 2016, LNCS 9667, pp. 277–289, 2016.
DOI: 10.1007/978-3-319-39441-1_25

The second and the third variations of the problem are more realistic than the first one, since the exact distances are often not available due to limitations of the measurement device. The DGP is related to many important research problems, such as Molecular Distance Geometry Problem and Molecular Conformation [8,11–13,15,16,20], Sensor Network Localization [4,5,18], and Graph Drawing [1,3,6,9]. In all these problems, the DGP is dealt in the real space. When the pairwise distances are exact, it can be solved in polynomial time [7,20]. For bounded and sparse distances, the scenario becomes complex and difficult to solve [14,20].

In our work, we have addressed the DGP for inexact or bounded distances in the integer plane and have devised an efficient technique for solving it whenever a given input instance admits a feasible solution. Although all the solution points have integer coordinates, the proposed technique can be extended for finding solutions in finer grids. To start with, in Sect. 2, we formulate the problem in the integer plane. In Sect. 3, we present a characterization of digital annulus intersection for efficient computation of the solution in the integer plane. The proposed heuristics and algorithm are discussed in Sect. 4. Further research directions and concluding notes are put in Sect. 5.

2 Preliminaries and Problem Definition in $\mathbb{Z}^2$

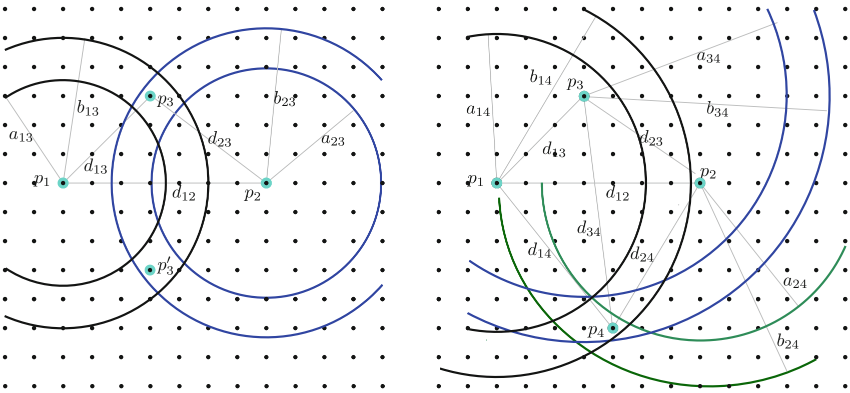
We fix here few definitions and notations used in the sequel. A real circle with center at $c \in \mathbb{R}^2$ and radius r is denoted by $\mathcal{C}^{\mathbb{R}}(c,r)$. A real annulus is defined as $\mathcal{A}^{\mathbb{R}}(c,a,b) = \{p \in \mathbb{R}^2 : a \leqslant \|c-p\| \leqslant b\}$, where $c \in \mathbb{R}^2$ is the center of the annulus, and a and b are the respective inner and outer radii. A digital annulus centered at $c \in \mathbb{R}^2$, and with inner and outer radii a and b respectively, is defined as the set of all integer points in $\mathcal{A}^{\mathbb{R}}(c,a,b)$, and so given by $\mathcal{A}^{\mathbb{Z}}(c,a,b)=\{p \in \mathbb{Z}^2 : a \leqslant \|c-p\| \leqslant b\}$.

Let $P = \{p_1, p_2, \ldots, p_n\}$ be a set of points with unknown coordinates in $\mathbb{Z}^2$ such that the distance between every two points p_i and p_j is known to lie in a given interval $[a_{ij}, b_{ij}]$. The objective is to determine the integer coordinates of all points in P. We consider an alternative form of the distance bounds where the distance between p_i and p_j lies in a known interval $[d_{ij}-\epsilon, d_{ij}+\epsilon]$ for some $\epsilon > 0$. We also assume that the given intervals are such that there is at least one solution to the problem.

To simplify our strategy, we first fix the reference frame with p_1 as the origin, p_2 as an integer point on the $+x$-axis, and p_3 lying left of $\overrightarrow{p_1p_2}$. With $a_{12} \leqslant \|p_1 - p_2\| \leqslant b_{12}$, we get $p_2 \in \mathcal{A}^{\mathbb{Z}}(p_1, a_{12}, b_{12})$, and $p_3 \in \mathcal{A}^{\mathbb{Z}}(p_1, a_{13}, b_{13}) \cap \mathcal{A}^{\mathbb{Z}}(p_2, a_{23}, b_{23})$ (Fig. 1a). Now, for $i \geqslant 4$, if we try to embed p_i based on its distance intervals from the previous $i-1$ points, then we need to compute $\bigcap_{j=1}^{i-1} \mathcal{A}^{\mathbb{Z}}(p_j, a_{ji}, b_{ji})$, which is expensive. Hence, as a faster solution, we use $\bigcap_{j=1}^{3} \mathcal{A}^{\mathbb{Z}}(p_j, a_{ji}, b_{ji})$ for each $p_i (i \geqslant 4)$ (Fig. 1b). For this, we have the following lemma.

Lemma 1. *All the solution points corresponding to p_i for $i = 4, 5, \ldots, n$ belong to $\bigcap_{j=1}^{3} \mathcal{A}^{\mathbb{Z}}(p_j, a_{ji}, b_{ji})$, which, if empty, does not yield any solution for P.*

Proof. Follows from the problem statement and our consideration. □



(a) p_3 (or, p'_3 as an alternative) (b) p_4

Fig. 1. Embedding of p_3 followed by p_4 in $\mathbb{Z}^2$.

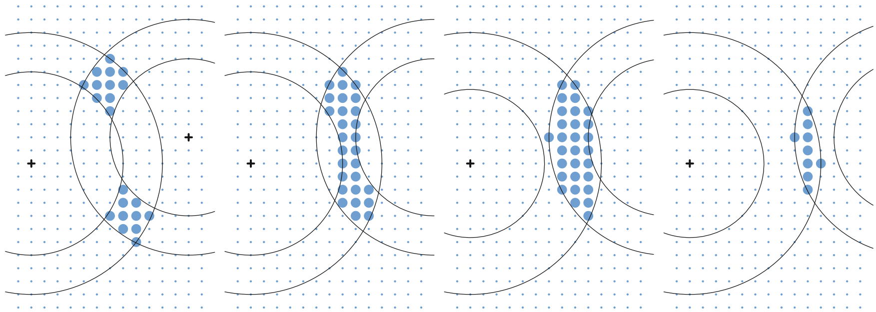
3 Characteristics of Annulus Intersection

We first provide some basic concepts of discrete geometry in the integer plane. For more details we refer to [10]. Let $p = (i, j) \in \mathbb{Z}^2$. The 4-*neighbors* of p in $\mathbb{Z}^2$ is defined as $N_4(p) = \{(i', j') \in \mathbb{Z}^2 : |i - i'| + |j - j'| = 1\}$, and the 8-*neighbors* as $N_8(p) = \{(i', j') \in \mathbb{Z}^2 : \max(|i - i'|, |j - j'|) = 1\}$. A subset A of $\mathbb{Z}^2$ is said to be 4-*connected* (resp., 8-*connected*) if either A is singleton or there exists a sequence of points in A between every two points of A such that every two consecutive points in that sequence are 4-neighbors (resp., 8-neighbors) of each other. When A is not connected, its maximally connected subsets are called *connected components*.

Let, without loss of generality, p_1 and p_2 be the farthest pair, and the distance d_{12} between them be denoted by d for simplicity. We examine the possible locations of an unknown point p, which is at a distance d_1 and d_2 from p_1 and p_2 respectively, where $a_1 \leqslant d_1 \leqslant b_1$ and $a_2 \leqslant d_2 \leqslant b_2$. So, by our supposition, $\max(b_1, b_2) \leqslant d$. Also, $b_1 + b_2 \geqslant d$, failing which there will be no such point p. Then p can be chosen as some point common to the digital annuli $\mathcal{A}^{\mathbb{Z}}(p_1, a_1, b_1)$ and $\mathcal{A}^{\mathbb{Z}}(p_2, a_2, b_2)$ if their intersection is non-empty. When a solution exists, i.e., the width of the annulus is sufficiently large, the intersection will be either a single or a pair of 8-connected components (Fig. 2). We define $\mathcal{I}^{\mathbb{Z}}$ as the set of integer points belonging to both $\mathcal{A}^{\mathbb{Z}}(p_1, a_1, b_1)$ and $\mathcal{A}^{\mathbb{Z}}(p_2, a_2, b_2)$. The non-empty intersection $\mathcal{I}^{\mathbb{Z}}$ is classified as follows.

- *Type 1* ($a_1 + a_2 > d$): $\mathcal{I}^{\mathbb{Z}}$ comprises two connected components lying in two different sides of the line p_1p_2. (As a degeneracy, it may have a single component when $a_1 + a_2$ tends to d.)
- *Type 2* ($(a_1 + a_2 \leqslant d) \wedge (a_1 + b_2 \geqslant d) \wedge (b_1 + a_2 \geqslant d)$): $\mathcal{I}^{\mathbb{Z}}$ is a single connected component. (As a degeneracy, it may have two connected components as in Type 1.)

- *Type 3* $(((a_1 + b_2 < d) \wedge (b_1 + a_2 > d)) \vee ((b_1 + a_2 < d) \wedge (a_1 + b_2 > d)))$: $\mathcal{I}^{\mathbb{Z}}$ is a single connected component.
- *Type 4* $((a_1 + a_2 < d) \wedge (a_1 + b_2 < d) \wedge (b_1 + a_2 < d))$: $\mathcal{I}^{\mathbb{Z}}$ is a single connected component bounded by the outer circles of the real annuli.



(a) Type 1 (b) Type 2 (c) Type 3 (d) Type 4

Fig. 2. Possible types of intersection between two annuli such that no radius exceeds the distance between the annulus centers.

Note that there can be other types of intersection if $\max(b_1, b_2) > d$. Hence, to avoid this, we consider p_1 and p_2 as the farthest pair. we have the following lemma related to intersection types.

Lemma 2. *Type 3 and Type 4 intersections are computationally equivalent to Type 2.*

Proof. For Type 3 (Fig. 3(a-b)), we can replace $\mathcal{A}^{\mathbb{Z}}(p_1, a_1, b_1)$ by $\mathcal{A}^{\mathbb{Z}}(p_1, a_1', b_1)$ with $a_1' = d - b_2$, whence it gets converted to Type 2, thereby equalizing $\mathcal{A}^{\mathbb{Z}}(p_1, a_1, b_1) \cap \mathcal{A}^{\mathbb{Z}}(p_2, a_2, b_2)$ to $\mathcal{A}^{\mathbb{Z}}(p_1, a_1', b_1) \cap \mathcal{A}^{\mathbb{Z}}(p_2, a_2, b_2)$; similarly, we can replace $\mathcal{A}^{\mathbb{Z}}(p_2, a_2, b_2)$ by $\mathcal{A}^{\mathbb{Z}}(p_2, a_2', b_2)$ with $a_2' = d - b_1$, which also results in Type 2. In case of Type 4 (Fig. 3c), $\mathcal{A}^{\mathbb{Z}}(p_1, a_1, b_1)$ and $\mathcal{A}^{\mathbb{Z}}(p_2, a_2, b_2)$ can be replaced by $\mathcal{A}^{\mathbb{Z}}(p_1, a_1', b_1)$ and $\mathcal{A}^{\mathbb{Z}}(p_2, a_2', b_2)$ respectively, with $a_1' = d - b_2$ and $a_2' = d - b_1$, thereby again resulting in Type 2. □

Thus, Type 3 and Type 4 intersections can be converted to their equivalent Type 2 intersection, which simplifies the associated computation.

3.1 Computing Annulus Intersection in $\mathbb{Z}^2$

We first ensure that an annulus intersection is of Type 1 or of Type 2 or its equivalent. For computing the integer points in $\mathcal{I}^{\mathbb{Z}} := \mathcal{A}^{\mathbb{Z}}(p_1, a_1, b_1) \cap \mathcal{A}^{\mathbb{Z}}(p_2, a_2, b_2)$, we first find a seed point $s \in \mathcal{I}^{\mathbb{R}} := \mathcal{A}^{\mathbb{R}}(p_1, a_1, b_1) \cap \mathcal{A}^{\mathbb{R}}(p_2, a_2, b_2)$. The seed point s is taken as an/the intersection point of the circles $\mathcal{C}^{\mathbb{R}}(p_1, \frac{a_1+b_1}{2})$

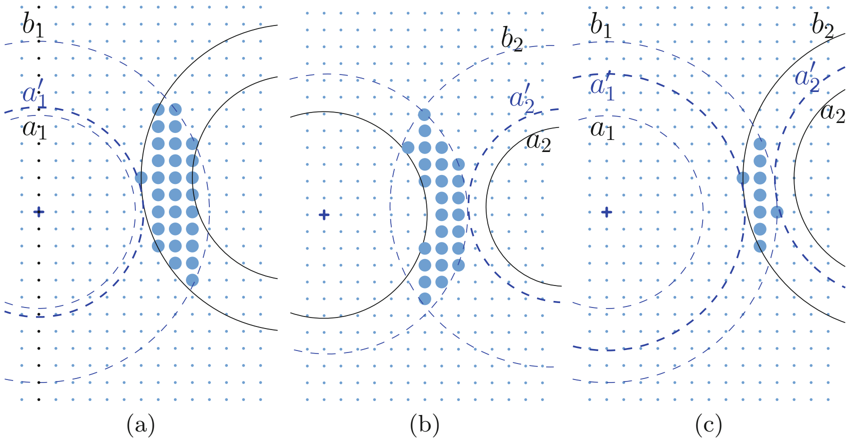


Fig. 3. Converting Type 3 and Type 4 intersections into equivalent Type 2 intersection by removing the empty sub-annulus incident on the inner circle (shown dashed and blue). (Color figure online)

and $\mathcal{C}^{\mathbb{R}}(p_2, \frac{a_2+b_2}{2})$. Note that $\mathcal{C}^{\mathbb{R}}(p_1, \frac{a_1+b_1}{2})$ and $\mathcal{C}^{\mathbb{R}}(p_2, \frac{a_2+b_2}{2})$ intersect at two points if $d' = \frac{a_1+b_1}{2} + \frac{a_2+b_2}{2} > d$, where d is the distance between p_1 and p_2. If $d' = d$, then they touch at a single point, and they do not intersect or touch at all when $d' < d$. As the intersection is of Type 1 or Type 2, there always exists a seed point with minimum distance $\frac{b_1-a_1}{2}$ and $\frac{b_2-a_2}{2}$ from the boundaries of the annuli (Fig. 4). This gives us the following lemma.

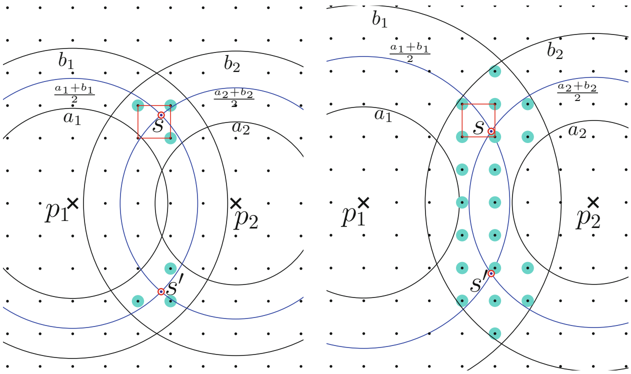
Lemma 3. *If* $\min(b_1 - a_1, b_2 - a_2) \geqslant \sqrt{2}$ *and the intersection of* $\mathcal{A}^{\mathbb{Z}}(p_1, a_1, b_1)$ *and* $\mathcal{A}^{\mathbb{Z}}(p_2, a_2, b_2)$ *is of Type 1 or of Type 2, then there exists an integer point* $q_0 \in \mathcal{A}^{\mathbb{Z}}(p_2, a_1, b_1) \cap \mathcal{A}^{\mathbb{Z}}(p_2, a_2, b_2)$ *and it can be determined in constant time.*

Proof. The seed point s belongs to a unit square U whose vertices are integer points. If the width of each annulus is at least $\sqrt{2}$, then the minimum distance of s from each annulus boundary is at least $\frac{\sqrt{2}}{2} = \frac{1}{\sqrt{2}}$. Hence, the real circle $\mathcal{C}^{\mathbb{R}}(s, \frac{1}{\sqrt{2}})$ is completely enclosed by both the annuli. Since s lies on or inside U, at least one vertex of U is within a distance of $\frac{1}{\sqrt{2}}$ from s, and so lies on or inside $\mathcal{C}^{\mathbb{R}}(s, \frac{1}{\sqrt{2}})$, and hence inside the annulus intersection. $\square$

By Lemma 3, we get an integer point $q_0 \in \mathcal{I}^{\mathbb{Z}}$. Starting from q_0, we get all other points in $\mathcal{I}^{\mathbb{Z}}$ as a connected component, using breadth first search (BFS) in the integer plane.

Henceforth in this paper, for notional and notational simplicity, we consider 2ϵ as the width of each annulus. So, the digital annulus centered at a point p_i is denoted by $\mathcal{A}^{\mathbb{Z}}(p_i, d_i - \epsilon, d_i + \epsilon)$, where d_i is its mean radius.

Theorem 1. *If* $d := ||p_1 - p_2|| \geqslant \max(d_1, d_2) + \epsilon$, *then the cardinality of* $\mathcal{I}^{\mathbb{Z}}$ *is* $O(\epsilon^2)$ *for Type 1 intersection and* $O(\epsilon\sqrt{\epsilon d})$ *for Type 2 intersection.*



(a) Type 1 (b) Type 2

Fig. 4. Finding a seed points s (red) for computing the annulus intersection. (Color figure online)

Proof. We have $d > d_1 + \epsilon$ and $d > d_2 + \epsilon$. If $d > d_1 + d_2 + 2\epsilon$, then $\mathcal{I}^{\mathbb{Z}}$ is empty. So, $d \leqslant d_1 + d_2 + 2\epsilon$. Without loss of generality, we take $p_1 = (0, 0)$ and $p_2 = (d, 0)$, as illustrated in Fig. 5.

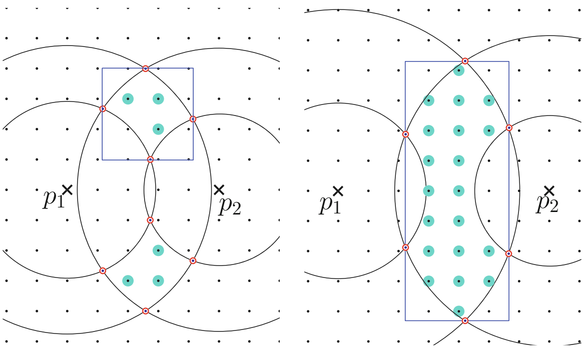
Let $\mathcal{I}^+ = \{(x, y) \in \mathcal{I}^{\mathbb{R}} : y \geqslant 0\}$, $x_{\min} = \min\{x : (x, y) \in \mathcal{I}^+\}$, $x_{\max} = \max\{x : (x, y) \in \mathcal{I}^+\}$, $y_{\min} = \min\{y : (x, y) \in \mathcal{I}^+\}$, and $y_{\max} = \max\{y : (x, y) \in \mathcal{I}^+\}$. Then there exists a rectangle with edge-lengths $w = (x_{\max} - x_{\min})$ and $h = (y_{\max} - y_{\min})$ such that it encloses $\mathcal{I}^+$. Hence, the number of integer points in $\mathcal{I}^+$ is $O(wh)$, and from the symmetry of the problem, the cardinality of $\mathcal{I}^{\mathbb{Z}}$ is $O(wh)$. We determine w and h by case analysis.

Case I ($\mathcal{I}^{\mathbb{R}}$ is Type 1): Here, $d \leqslant d_1 + d_2 - 2\epsilon$, and $\mathcal{I}^{\mathbb{R}}$ consists of two connected components: one lies above and the other lies below the line p_1p_2 (Fig. 5(a)). Then $x_{\min}$ is the abscissa of the point of intersection of $\mathcal{C}^{\mathbb{R}}(p_1, d_1 - \epsilon)$ and $\mathcal{C}^{\mathbb{R}}(p_2, d_2 + \epsilon)$. By solving the equations $x^2 + y^2 = (d_1 - \epsilon)^2$ and $(x - d)^2 + y^2 = (d_2 + \epsilon)^2$, we get $x_{\min} = \frac{d^2 + (d_1 - \epsilon)^2 - (d_2 + \epsilon)^2}{2d}$. Similarly, the abscissa of the point of intersection of $\mathcal{C}^{\mathbb{R}}(p_1, d_1 + \epsilon)$ and $\mathcal{C}^{\mathbb{R}}(p_2, d_2 - \epsilon)$ gives $x_{\max} = \frac{d^2 + (d_1 + \epsilon)^2 - (d_2 - \epsilon)^2}{2d}$. So by using $d \geqslant \max(d_1, d_2) + \epsilon$, we get

$$w = x_{\max} - x_{\min} = \frac{4\epsilon(d_1 + d_2)}{2d} \leqslant 4\epsilon\left(1 - \frac{\epsilon}{d}\right) \leqslant 4\epsilon. \tag{1}$$

By symmetry of the problem, we can interchange x and y to obtain a similar bound for h as 4ϵ. Hence, for each of the two components of the annulus intersection, there exists an enclosing square of length 4ϵ. Hence, the cardinality of $\mathcal{I}^{\mathbb{Z}}$ becomes $O(\epsilon^2)$.

Case II ($\mathcal{I}^{\mathbb{R}}$ is Type 2): Here, $d_1 + d_2 - 2\epsilon < d$, because the inner circles $\mathcal{C}^{\mathbb{R}}(p_1, d_1 - \epsilon)$ and $\mathcal{C}^{\mathbb{R}}(p_2, d_2 - \epsilon)$ do not intersect or touch each other. By following the procedure used in Case I, it can be shown that $x_{max} - x_{min} \leqslant 4\epsilon$, or, $w = O(\epsilon)$.



(a) (b)

Fig. 5. Cardinality of intersection for annulus width 2ϵ. (a) Type 1 annulus intersection is bounded by a rectangle with (axis-parallel) edges of length $O(\epsilon)$ and cardinality $O(\epsilon^2)$. (b) Type 2 annulus intersection is bounded by a rectangle with edges of length $O(\epsilon)$ and $O(\sqrt{\epsilon d})$, and cardinality $O(\epsilon\sqrt{\epsilon d})$.

Let $c = (c_x, c_y)$ be the topmost point of the annulus intersection, and hence it is the intersection point of the outer real circles of the two annuli. By symmetry, $c' = (c_x, -c_y)$ is the bottommost point, which is the reflection of c with respect to x-axis. Then solving the equations $c_x^2 + c_y^2 = (d_1+\epsilon)^2$ and $(d-c_x)^2 + c_y^2 = (d_2+\epsilon)^2$, we get $c_x = \frac{d^2+(d_1+\epsilon)^2-(d_2+\epsilon)^2}{2d}$ and c_y as follows.

$$\begin{aligned} c_y^2 &= (d_1+\epsilon)^2 - \left(\frac{d^2+(d_1+\epsilon)^2-(d_2+\epsilon)^2}{2d}\right)^2 = \frac{\left((d_1+d_2+2\epsilon)^2-d^2\right)\left(d^2-(d_1-d_2)^2\right)}{4d^2} \\ &\leqslant \frac{\left((d+4\epsilon)^2-d^2\right)\left(d^2-(d_1-d_2)^2\right)}{4d^2}, \text{ since } d_1+d_2-2\epsilon < d \\ &= 4\epsilon^2\left(1+\frac{d}{2\epsilon}\right)\left(1-\left(\frac{d_1-d_2}{d}\right)^2\right) \\ &\leqslant 4\epsilon^2\left(\frac{d}{2\epsilon}+\frac{d}{2\epsilon}\right), \text{ since } d \geqslant \max(d_1,d_2)+\epsilon, d > d_1+d_2-2\epsilon, \frac{d}{2\epsilon} > 1 \\ &= 4\epsilon d \implies c_y = O(\sqrt{\epsilon d}). \end{aligned} \tag{2}$$

By Eq. 2, $h = O(\sqrt{\epsilon d})$; as $w = O(\epsilon)$, the number of points in $\mathcal{I}^{\mathbb{Z}}$ is $O(\epsilon\sqrt{\epsilon d})$. □

By Lemma 3 and Theorem 1, we have the following corollary.

Corollary 1. *$\mathcal{I}^{\mathbb{Z}}$ is computable in $O(\epsilon^2)$ time for Type 1 and in $O(\epsilon\sqrt{\epsilon d})$ time for Type 2 intersections.*

4 Proposed Algorithm and Implementation Issues

The first three points p_1, p_2, and p_3 are known, or are fixed with an appropriate coordinate system, as explained in Sect. 2. That is, for $i = 1, 2, 3$, the candidate solutions are $\mathcal{I}_i^{\mathbb{Z}} = \{p_i\}$. For each other point p_i, a set of candidate solutions is obtained as $\mathcal{I}_i^{\mathbb{Z}} := \bigcap_{j=1}^{3} \mathcal{A}^{\mathbb{Z}}(p_j, a_{ji}, b_{ji})$. One or more points from $\mathcal{I}_i^{\mathbb{Z}}$ would belong to a/the global solution with all n points. In particular, we have the following theorem on the collection of these sets of candidate solutions.

Theorem 2. *If a given set of distance constraints admits a global solution, then it is always contained in the collection* $\mathcal{K} := \{\mathcal{I}_i^{\mathbb{Z}} : i = 1, 2, \ldots, n\}$ *whose computational time is* $O(\epsilon^2 n)$ *in the best case and* $O(\epsilon^{\frac{3}{2}} d^{\frac{1}{2}} n)$ *in the worst case.*

Proof. The containment of global solution in $\mathcal{K}$ follows from Lemma 1. For proving the computational part, we use Corollary 1. For each p_i with $i = 4, \ldots, n$, computation of $\mathcal{A}^{\mathbb{Z}}(p_1, a_{1i}, b_{1i}) \cap \mathcal{A}^{\mathbb{Z}}(p_2, a_{2i}, b_{2i})$ takes $O(\epsilon^2)$ time for Type 1 intersection and $O(\epsilon\sqrt{\epsilon d})$ time for Type 2 intersection. The additional time needed for validation against $\mathcal{A}^{\mathbb{Z}}(p_3, a_{3i}, b_{3i})$ is subsumed within the aforesaid time complexity. Summing up this over i from 4 to n, we get the result. □

By Lemma 3, we always have an integer point $q_{i,0}$ in $\mathcal{I}_i^{\mathbb{Z}}$ if we set $\epsilon \geqslant \frac{1}{\sqrt{2}}$ and $\mathcal{I}_i^{\mathbb{Z}}$ is of Type 1 or of Type 2, for $i = 1, 2, \ldots, n$. Further, by Theorem 1, if $d \geqslant \max(d_1, d_2) + \epsilon$, then with $\epsilon = \frac{1}{\sqrt{2}}$, we get $|\mathcal{I}_i^{\mathbb{Z}}| = O(1)$ for Type 1 intersection and $O(\sqrt{d})$ for Type 2 intersection. So by Corollary 1, $\mathcal{I}_i^{\mathbb{Z}}$ can be computed in $O(1)$ time in the best case and in $O(\sqrt{d})$ time in the worst case. This gives the following corollary.

Corollary 2. *For an appropriately small value of* ϵ *(*$= \frac{1}{\sqrt{2}}$ *for definiteness), the time complexity of the algorithm varies from* $O(n)$ *to* $O(n\sqrt{d})$*, where* d *is the maximum point-pair distance.*

When positions of p_1, p_2, and p_3 are not known, we can choose p_1 as the origin and p_2 lying in $\mathcal{A}^{\mathbb{Z}}(p_1, a_{12}, b_{12})$. There are $O(\epsilon d)$ possible choices for p_2. Next, the third point p_3 can be chosen from the intersection of $\mathcal{A}^{\mathbb{Z}}(p_1, a_{13}, b_{13})$ and $\mathcal{A}^{\mathbb{Z}}(p_2, a_{23}, b_{23})$. There are $O(\epsilon^{\frac{3}{2}}\sqrt{d})$ ways to choose p_3. The rest of the points can be computed using the proposed technique. For certain choices of p_2 and p_3, a global solution (satisfying all the distance bounds) may not exist. So, we may need to consider all possible choices of p_2 and p_3 in order to compute a global solution. There are $O(\epsilon^{\frac{5}{2}} d^{\frac{3}{2}})$ possible pairs of possible solutions for p_2 and p_3 in the worst case. Hence, total time required for computing a global solution is $O(\epsilon^4 d^2 n)$ in the worst case. We also need $O(n^2)$ time to verify the distance bounds for a solution.

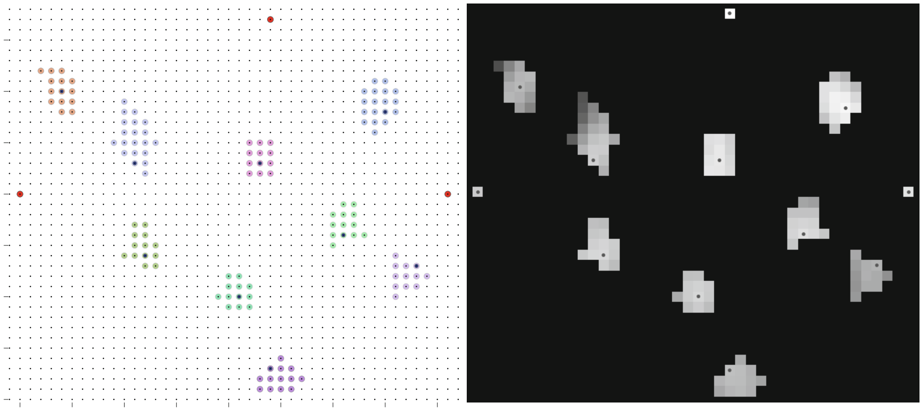
4.1 Search for Feasible Solution

By Theorem 2, the collection $\mathcal{K}$ contains all feasible solutions. However, it may contain some infeasible combinations too, alongside. To distinguish the former from the latter, one has to search at least $O(\epsilon^{2n})$ possible combinations. These combinations can be represented in a rooted tree with the root node containing the solution for p_1 and a node at the ith level representing a set of solutions for p_i that satisfies the distance constraints with its predecessor points from p_{i-1} to p_1. The tree traversal is similar to the branch-and-prune technique proposed

Algorithm 1. DDGA(n, a, b, P_1, P_2, P_3)

Create empty list $\mathcal{I}_i^{\mathbb{Z}}$ of candidate solutions for ith point, for $i = 1, 2, \cdots, n$.
$\mathcal{I}_1^{\mathbb{Z}} \leftarrow \{P_1\}$, $\mathcal{I}_2^{\mathbb{Z}} \leftarrow \{P_2\}$, $\mathcal{I}_3^{\mathbb{Z}} \leftarrow \{P_3\}$
for $i = 4, 2, \cdots, n$ **do** ▷ set of candidate solutions for ith point
    $\mathcal{I}_i^{\mathbb{Z}} \leftarrow \mathcal{A}^{\mathbb{Z}}(P_1, a_{1i}, b_{1i}) \cap \mathcal{A}^{\mathbb{Z}}(P_2, a_{2i}, b_{2i}) \cap \mathcal{A}^{\mathbb{Z}}(P_3, a_{3i}, b_{3i})$.
    $n_i \leftarrow |\mathcal{I}_i^{\mathbb{Z}}|$ ▷ number of candidate solutions for ith point
$N \leftarrow$ **ShiftOrigin**$(\bigcup_{i=1}^{n} \mathcal{I}_i^{\mathbb{Z}})$ ▷ make sure all points are in $[1, N] \times [1, N]$
Create a matrix $B[1..N; 1..N]$, initialized to 0s ▷ to accumulate votes
for $i = 1, 2, \cdots n$ **do** ▷ casting mutual votes of $\mathcal{I}_i^{\mathbb{Z}}$ and $\mathcal{I}_j^{\mathbb{Z}}$
    for $j = i + 1, i + 2, \cdots, n$ **do**
        for $k = 1, 2, \cdots, n_i$ **do**
            $(\alpha_i, \beta_i) \leftarrow \mathcal{I}_i^{\mathbb{Z}}[k]$ ▷ get the kth point of $\mathcal{I}_i^{\mathbb{Z}}$
            for $l = 1, 2, \cdots, n_j$ **do**
                $(\alpha_j, \beta_j) \leftarrow \mathcal{I}_j^{\mathbb{Z}}[l]$ ▷ get the lth point of $\mathcal{I}_j^{\mathbb{Z}}$
                $d \leftarrow ||(\alpha_i, \beta_i) - (\alpha_j, \beta_j)||$ ▷ distance between the points
                if $a_{ij} \leqslant d \leqslant b_{ij}$ **then**
                    $B[\alpha_i, \beta_i] \leftarrow B[\alpha_i, \beta_i] + 1$ ▷ casting a vote for (α_i, β_i)
                    $B[\alpha_j, \beta_j] \leftarrow B[\alpha_j, \beta_j] + 1$ ▷ casting a vote for (α_j, β_j)
for $i = 4, 5, \cdots, n$ **do** ▷ find a point in $\mathcal{I}_i^{\mathbb{Z}}$ having maximum vote
    $(\alpha, \beta) \leftarrow \mathcal{I}_i^{\mathbb{Z}}[1]$ ▷ first point in the list $\mathcal{I}_i^{\mathbb{Z}}$
    for $k = 2, 3, \cdots, n_i$ **do** ▷ search in $\mathcal{I}_i^{\mathbb{Z}}$
        $(\alpha_i, \beta_i) \leftarrow \mathcal{I}_i^{\mathbb{Z}}[k]$
        if $B[\alpha, \beta] < B[\alpha_i, \beta_i]$ **then**
            $(\alpha, \beta) \leftarrow (\alpha_i, \beta_i)$
    $P_i \leftarrow (\alpha, \beta)$ ▷ final solution point with maximum vote
return $\{P_i : i = 1, 2, \cdots, n\}$

in [13] and the interval branch-and-prune technique proposed in [12] for finding an approximate DGP solution in $\mathbb{R}^2$. In our approach, the search is performed in $\mathbb{Z}^2$, where we need to verify only a finitely many possible points for a valid solution. A possible solution to the DGP problem corresponds to a path of length $n-1$ from the root to a leaf node. In order to find a solution, in the worst case, we may need to explore and verify all paths, which would take exponential time when ϵ is not small. In such a case, a voting scheme may be adopted as an efficient heuristic to determine the best solution point in each $\mathcal{I}_i^{\mathbb{Z}}$ in the sense that it satisfies the maximum number of distance constraints. For any $p \in \mathcal{I}_i^{\mathbb{Z}}$ and $q \in \mathcal{I}_j^{\mathbb{Z}}$, if p and q satisfy the distance constraints, then each of them receives a mutual vote. In order to compute the vote for all candidate integer points, we use an accumulator B (implemented as a 2D array), such that for each digital point (α, β), $B(\alpha, \beta)$ stores the number of votes received by (α, β). The main steps are shown in Algorithm 1.



(a) (b)

Fig. 6. Visualization of a digital solution to DGP with inexact distances. See Appendix for the input. (a) p_1, p_2, and p_3 shown in red; the connected components making the candidate solutions of all other points shown in distinct colors; each point having the highest vote in a component shown as a blue dot. (b) The votes for all candidate solutions are treated as intensity values to visualize the accumulator matrix B as an image. Brighter pixels correspond to higher votes and pixels having highest votes are marked by colored dots. (Color figure online)

4.2 Improvement of Efficiency

If all the annulus intersections computed during the execution of the algorithm are of Type 1, then the algorithm has the speediest execution. So, in order to increase Type 1 intersections, we choose p_1 and p_2 as the farthest pair. Similarly, we choose p_3 such that $\min(b_{13}, b_{23}) \geqslant \min(b_{1i}, b_{2i})$, $\forall i \neq 1, 2$. Further, $\mathcal{I}_i^{\mathbb{Z}}$ can be computed by following an appropriate ordering of computation of the intersection of the three annuli $\mathcal{A}^{\mathbb{Z}}(p_1, a_{1i}, b_{1i})$, $\mathcal{A}^{\mathbb{Z}}(p_2, a_{2i}, b_{2i})$, and $\mathcal{A}^{\mathbb{Z}}(p_3, a_{3i}, b_{3i})$, so that $\mathcal{I}_i^{\mathbb{Z}}$ is Type 1 intersection; otherwise, we defer its computation. In particular, we first run the algorithm for the points that have Type 1 intersection with respect to p_1, p_2 and p_3, and find the solution by voting scheme. We take S_1 as the list of points whose solutions are found by this, and S_2 as the list of the remaining points, i.e., the ones having Type 2 intersection. The solutions for the points in S_2 are determined by using few suitable points from S_1.

4.3 Test Results

The proposed algorithm for solving inexact DGP is implemented and tested on randomly generated problem instances. Experimentation shows that the algorithm is able to solve inexact DGP efficiently. A solution to a small problem instance with $n = 12$ points is visualized in Fig. 6. Notice that only one solution is reported here; there are, in fact, multiple solutions when a connected component contains more than one point having the same maximum vote.

5 Concluding Notes

We have formulated the inexact DG problem in $\mathbb{Z}^2$ and have done a theoretical analysis in order to solve it efficiently. For a large value of ϵ, we have proposed an efficient voting scheme in order to bring down the exponential time complexity to a low-order polynomial. The idea can be extended for solving inexact DGP in 3D integer space also. Besides, many other issues and possibilities have opened up, which requires a deeper analysis of the problem in the discrete space. Some of these, which we foresee as potential research problems, are as follows.

1. The number of worst-case occurrences in the execution of the algorithm can possibly be minimized by changing the sequence in which the annuli are considered while computing their intersections. The rationale is that if the centers of three annuli are well-separated, then their intersection is well-formed, easy to compute, and lead to speedier execution.
2. In many applications, the distances may not be known for some pairs of points. We can then apply the algorithm to compute the candidate solutions for those points whose distances from p_1, p_2, and p_3 are known; then these partial solutions can be used in a suitable order to find the remaining points.
3. For a small ϵ, no integer solution may exist for certain points. To handle this, we can go for a higher resolution by subdividing $\mathbb{Z}^2$ and reporting the solution point coordinates as rational numbers.
4. We have not commented on the behavior of the heuristic used when ϵ is large. As ϵ increases, uncertainty also increases, and the solution space increases exponentially and depends on the distribution of the pairwise distances. We envisage a wide scope of research to analyze this aspect of the problem.

A Appendix

Top matrix: $(d)_{n\times n}$ contains actual distances among $n = 12$ points generated randomly. An instance of inexact DGP is made with $[a_{ij}, b_{ij}] = [(d_{ij} - \epsilon), (d_{ij} + \epsilon)$ for $\epsilon = 2$.
Bottom matrix: $(\hat{d})_{n\times n}$ (rounded off to two decimal places) of the embedded point set; the RMS error $\left(= \sqrt{\sum_{i<j}(d_{ij} - \hat{d}_{ij})^2 / \binom{n}{2}} \right)$ between $(d)_{n\times n}$ and $(\hat{d})_{n\times n}$ is 0.7654.

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}
p_1	0	41.77	28.92	30.89	38.19	31.69	23.49	12.27	10.71	23.37	35.63	13.35
p_2	41.77	0	24.01	24.42	9.33	10.85	18.57	30.39	38.81	22.94	11.08	30.30
p_3	28.92	24.01	0	34.07	27.74	21.04	13.72	17.40	20.75	26.31	13.11	25.00
p_4	30.89	24.42	34.07	0	15.54	15.72	20.43	25.68	34.76	8.88	28.00	17.87
p_5	38.19	9.33	27.74	15.54	0	7.45	17.54	28.38	37.67	16.28	16.87	25.39
p_6	31.69	10.85	21.04	15.72	7.45	0	10.09	21.21	30.36	12.24	12.37	19.61
p_7	23.49	18.57	13.72	20.43	17.54	10.09	0	11.82	20.51	12.65	12.76	14.02
p_8	12.27	30.39	17.40	25.68	28.38	21.21	11.82	0	9.48	16.85	23.43	9.72
p_9	10.71	38.81	20.75	34.76	37.67	30.36	20.51	9.48	0	26.06	30.41	17.48
p_{10}	23.37	22.94	26.31	8.88	16.28	12.24	12.65	16.85	26.06	0	22.86	10.02
p_{11}	35.63	11.08	13.11	28.00	16.87	12.37	12.76	23.43	30.41	22.86	0	26.71
p_{12}	13.35	30.30	25.00	17.87	25.39	19.61	14.02	9.72	17.48	10.02	26.71	0

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}
p_1	0	41.00	29.41	29.41	38.64	31.26	23.19	11.40	10.77	23.26	35.90	13.42
p_2	41.00	0	24.04	24.04	7.62	10.77	18.25	30.15	38.33	22.36	10.00	29.61
p_3	29.41	24.04	0	34.00	27.78	22.14	14.04	19.10	21.19	27.17	14.21	25.94
p_4	29.41	24.04	34.00	0	17.20	14.76	20.02	23.85	33.60	7.62	27.31	16.28
p_5	38.64	7.62	27.78	17.20	0	7.62	18.03	28.79	38.01	17.26	15.30	26.02
p_6	31.26	10.77	22.14	14.76	7.62	0	10.63	21.19	30.41	11.66	12.65	19.10
p_7	23.19	18.25	14.04	20.02	18.03	10.63	0	12.00	20.25	13.15	13.00	14.21
p_8	11.40	30.15	19.10	23.85	28.79	21.19	12.00	0	9.90	16.40	24.52	9.06
p_9	10.77	38.33	21.19	33.60	38.01	30.41	20.25	9.90	0	26.25	31.06	17.89
p_{10}	23.26	22.36	27.17	7.62	17.26	11.66	13.15	16.40	26.25	0	22.80	9.85
p_{11}	35.90	10.00	14.21	27.31	15.30	12.65	13.00	24.52	31.06	22.80	0	26.93
p_{12}	13.42	29.61	25.94	16.28	26.02	19.10	14.21	9.06	17.89	9.85	26.93	0

References

1. http://www.graphdrawing.org/. Accessed 27 Sept 2015
2. Blumenthal, L.M.: Theory and Application of Distance Geometry. Oxford University Press, Oxford (1953)
3. Brandes, U., Pich, C.: An experimental study on distance-based graph drawing. In: GD 2008, Revised Papers, pp. 218–229 (2008)
4. Bulusu, N., Estrin, D., Heidemann, J.: Scalable coordination for wireless sensor networks self-configuring localization systems. In: Proceedings of the ISCTA (2001)
5. Cheng, L., Wu, C., Zhang, Y., Wu, H., Li, M., Maple, C.: A survey of localization in wireless sensor network. Int. J. Distrib. Sens. Netw., 324–357 (2012)
6. Civril, A., Magdon-Ismail, M., Bocek-Rivele, E.: SDE: graph drawing using spectral distance embedding. In: Healy, P., Nikolov, N.S. (eds.) GD 2005. LNCS, vol. 3843, pp. 512–513. Springer, Heidelberg (2006)
7. Dong, Q., Wu, Z.: A linear-time algorithm for solving the molecular distance geometry problem with exact inter-atomic distance. J. Global Optim. **22**, 365–375 (2002)
8. Dong, Q., Wu, Z.: A geometric build-up algorithm for solving the molecular distance geometry problem with sparse distance data. J. Global Optim. **26**, 321–333 (2003)
9. Gibson, H., Faith, J., Vickers, P.: A survey of two-dimensional graph layout techniques for information visualization. Inf. Vis. **12**(3–4), 324–357 (2012)
10. Klette, R., Rosenfeld, A.: Digital Geometry: Geometric Methods for Digital Picture Analysis. Morgan Kaufmann, San Francisco (2004)
11. Lavor, C., Liberti, L., Maculan, N., Mucherino, A.: Recent advances on the discretizable molecular distance geometry problem. Computational Optimization and Applications (2012)
12. Lavor, C., Liberti, L., Mucherino, A.: The interval branch-and-prune algorithm for the discretizable molecular distance geometry problem with inexact distances. J. Global Optim. **56**, 855–871 (2013)
13. Liberti, L., Lavor, C., Maculan, N.: A branch-and-prune algorithm for the molecular distance geometry problem. Intl. Trans. Oper. Res. **15**, 1–17 (2008)
14. Liberti, L., Lavor, C., Mucherino, A.: Euclidean distance geometry and applications. SIAM Rev. **56**, 3–69 (2014)
15. Liberti, L., Lavor, C., Mucherino, A., Maculan, N.: Molecular distance geometry methods:from continuous to discrete. Intl. Trans. Oper. Res. **18**, 33–51 (2010)
16. Moré, J.J., Wu, Z.: Distance geometry optimization for protein structures. J. Global Optim. **15**, 219–234 (1999)
17. Mucherino, A., Lavor, C., Liberti, L., Maculan, N.: Distance Geometry: Theory, Methods, and Applications, 1st edn. Springer, New York (2013)

18. Savvides, A., Han, C.C., Strivastava, M.B.: Dynamic fine-grained localization in ad-hoc networks of sensors. In: Proceedings of the MobiCom 2001, pp. 166–179 (2001)
19. Saxe, J.B.: Embeddability of weighted graphs in k-space is strongly NP-hard. In: Proceedings of the 17th Allerton Conference in Communications, Control & Computing, pp. 480–489 (1979)
20. Sit, A.: Solving distance geometry problems for protein structure determination. P.h.D thesis, Iowa State University (2010)