

Reconstruction in one dimension from unlabeled Euclidean lengths

Robert Connelly *

Steven J. Gortler[†]

Louis Theran[‡]

Abstract

Let G be a 3-connected ordered graph with n vertices and m edges. Let \mathbf{p} be a randomly chosen mapping of these n vertices to the integer range $\{1, 2, 3, \dots, 2^b\}$ for $b \geq m^2$. Let ℓ be the vector of m Euclidean lengths of G 's edges under \mathbf{p} . In this paper, we show that, with high probability over \mathbf{p} , we can efficiently reconstruct both G and \mathbf{p} from ℓ . This reconstruction problem is NP-HARD in the worst case, even if both G and ℓ are given. We also show that our results stand in the presence of small amounts of error in ℓ , and in the real setting, with sufficiently accurate length measurements.

Our method combines lattice reduction, which has previously been used to solve random subset sum problems, with an algorithm of Seymour that can efficiently reconstruct an ordered graph given an independence oracle for its matroid.

1. Introduction

Let \mathbf{p} be a *configuration* of n points p_i on the real line. Let G be an ordered graph with n vertices, numbered $\{1, \dots, n\}$ (briefly, $[n]$) and m edges, indexed by $[m]$. Having edge indices is very useful when applying linear algebraic methods to graphs. We denote the edge with endpoints i and j as ij or $\{i, j\}$. For each edge ij in G , we denote its Euclidean length as $\ell_{ij}(\mathbf{p}) := |p_i - p_j|$. We denote by $\ell = \ell(\mathbf{p})$, the m -vector of all of these measured lengths. In a *labeled* reconstruction problem, we are given G and ℓ , and we are asked to reconstruct \mathbf{p} , up to an unknowable translation and reflection. Since the graph G and the vector ℓ are ordered, this means we are given the bijection between lengths and edges. In an *unlabeled* reconstruction problem, the input is the vector ℓ , and the task is to reconstruct G and \mathbf{p} . As in the unlabeled setting, \mathbf{p} can be reconstructed only up to a translation and reflection. Since G is ordered, reconstructing it requires determining a bijection between the edges of G and the entries of ℓ . We note that the number of entries of ℓ determines m , but not n , and that n is not in the input. That problems of this type are difficult to solve in the worst case is classical. Saxe [30] showed in 1979 that the labeled version of the problem in one dimension is strongly NP-hard.

Two and three-dimensional labeled and unlabeled reconstruction problems often arise in applications such as sensor network localization and molecular shape determination. These problems are even harder than the one-dimensional setting studied here. But due to the practical

*Department of Mathematics, Cornell University. rc46@cornell.edu. Partially supported by NSF grant DMS-1564493.

[†]School of Engineering and Applied Sciences, Harvard University. sjg@cs.harvard.edu. Partially supported by NSF grant DMS-1564473.

[‡]School of Mathematics and Statistics, University of St Andrews. lst6@st-and.ac.uk

importance of these applications, many heuristics have been studied, especially in the labeled setting. See [8, 9] for some surveys of this topic. There do exist polynomial time algorithms that have guarantees of success, but only on restricted classes of graphs and configurations. For example, labeled problems can be solved with semidefinite programming if the unknown configuration \mathbf{p} and graph G correspond to a “universally rigid framework” (G, \mathbf{p}) [33]. Labeled problems in dimension d can be solved with greedy algorithms if the underlying graph happens to be “ d -trilateral”. This means that the graph has an edge set that allows one to localize one or more K_{d+2} subgraphs, and then iteratively glue vertices onto already-localized subgraphs using $d+1$ edges, (and unambiguously glue together already localized subgraphs) in a way to cover the entire graph. There is also a huge literature on the related problem of low-rank matrix completion; see [27] for a survey of this topic. Typically, those results assume randomness in the choice of matrix sample locations, roughly corresponding to the graph G in our setting.

Unlabeled problems are much harder, and there are fewer approaches known to be viable. In [4], the authors prove that for all dimensions d and configurations \mathbf{p} outside of a Zariski-closed subset of d -dimensional configurations, there is a unique way to associate the values in ℓ to the edges of the complete graph K_n . Thus, when all pairwise distances are given, an exhaustive combinatorial search (exponential time) must correctly reconstruct such configurations. For d -trilateral graphs, a backtracking approach called LIGA is presented in [21] and a greedy approach called TRIBOND is presented in [12]. TRIBOND is shown to run in $O(m^k)$ time, where k is a fixed exponent for each dimension. The work described by these groups has spearheaded the research in unlabeled reconstruction. In [2], the authors define a property for a measured length vector ℓ they call “strongly generic”. They then say that the TRIBOND approach must succeed if ℓ is strongly generic. Notably, no sufficient condition is provided for any graph G to have even a single strongly generic measured length vector. In [15] it is shown that for $d \geq 2$, when the underlying graph is d -trilateral then the TRIBOND algorithm, given exact measurements of a d -dimensional configuration \mathbf{p} , must correctly reconstruct \mathbf{p} , provided that \mathbf{p} is not in an exceptional, Zariski-closed subset of all d -dimensional configurations. When $d = 1$ the authors show that the graph must be 2-trilateral (instead of 1-trilateral) in order for a greedy trilateration method to work.

The unlabeled reconstruction problem in the one-dimensional integer setting also goes by names such as “the turnpike problem” and the “partial digest problem”. Theoretical work on these problems typically uses worst case analysis and often only applies to the complete graph (e.g. [5, 23, 32]).

In this paper we present an algorithm that can provably solve the one-dimensional unlabeled (and thus, also the labeled) reconstruction problems in polynomial time, for each fixed ordered graph, with high probability over randomly chosen \mathbf{p} , as long as we have sufficiently accurate length measurements. We only require that G is 2-connected for the labeled case and 3-connected for the unlabeled case. As we will see below, these conditions are necessary for any method to work.

The algorithm is based on combining the LLL lattice reduction algorithm [13, 22, 24] together with the graphic matroid reconstruction algorithm of Seymour [31]. The limiting factor of our algorithm in practice is its reliance on very precise measurements of the edge lengths ℓ . But the existence of an efficient algorithm with these theoretical guarantees was surprising to us. Even though the required precision grows with the number of edges, we give an explicit bound. This is in contrast with theoretical results about generic configurations often found in rigidity theory, which do not apply to configurations with integer coordinates or inexactly-measured lengths.

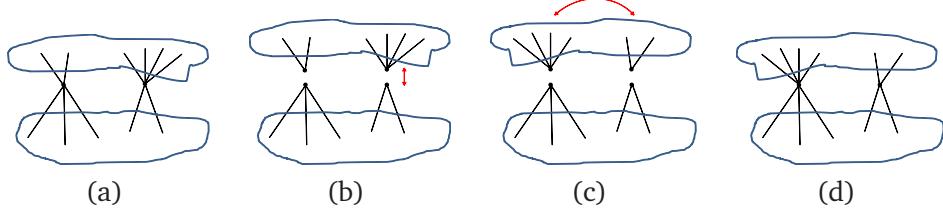


Figure 1: The graphs, (a) and (d) are related by a 2-swap. Note that the edge lengths in one dimension are unchanged under a 2-swap.

1.1. Connectivity assumption

Clearly we can never detect a translation or reflection of \mathbf{p} in the line, so we will only be interested in recovery of the configuration up to such *congruence*. In the labeled setting, if i is a cut vertex of G , then G is the union of two graphs H_1 and H_2 with at least two vertices each, so that the vertex set of H_1 and H_2 intersect on exactly $\{i\}$. Hence, we can reflect the positions of the vertices in H_1 across p_i without changing ℓ . Such a flip is invisible to our input data. To avoid such partial reflections, we must assume that G is 2-connected.

The question of whether it is possible to recover \mathbf{p} , a configuration in \mathbb{R}^d , uniquely up to isometry from the the labeled length measurements is known in rigidity theory as the question of whether the *framework* (G, \mathbf{p}) is *globally rigid* in d -dimensions. If, for generic \mathbf{p} , (G, \mathbf{p}) is always globally rigid in dimension d , then G is called *generically globally rigid* in dimension d [7, 17]. It is a folklore result that G is generically globally rigid in dimension one if and only if it is 2-connected.

In higher dimensions, $(d + 1)$ -connectivity is necessary for a graph to be generically globally rigid [19], but it is not sufficient [6]. In dimension two, there is a combinatorial characterization of generically globally rigid graphs [20]. In dimensions $d \geq 3$, there is no known combinatorial characterization of generically globally rigid graphs, but there is a characterization that can be checked with a randomized algorithm that uses linear algebra [17].

In the unlabeled setting, we can never detect a renumbering of the points in \mathbf{p} along with a corresponding vertex renumbering in G , so we will be unconcerned with such isomorphic outputs. More formally, let $G = (V, E)$ be an ordered graph with vertex set $[n]$ and with an ordering of the edges given by a bijection $\sigma : E \rightarrow [m]$. If $H = (V, E')$ is another ordered graph with the same vertex set, such that there is a bijection $\pi : V \rightarrow V$ such that $ij \in E$ if and only if $\pi(i)\pi(j) \in E'$ and the edges of H are ordered via a bijection $\tau : E' \rightarrow [m]$ so that $\sigma(ij) = \tau(\pi(i)\pi(j))$, then G and H are isomorphic as ordered graphs, and it is impossible to detect whether a measurement vector ℓ arose from G or H .

Without labeling information, if G has a cut vertex i , then, decomposing G into H_1 and H_2 as above, we cannot hope to distinguish measurements from G and those from a graph made from identifying H_1 and H_2 along some other vertex besides i , because besides reflecting the positions of vertices in H_2 across i , we can also translate them. Even if G is 2-connected but not 3-connected, the unlabeled problem does not have a unique solution. Let $\{i, j\}$ be a cut pair and decompose G into graphs H_1 and H_2 so that their vertex sets intersect exactly on $\{i, j\}$. Any set of length measurements of G also arises from the graph obtained by identifying the copy of i in H_1 with the copy of j in H_2 and vice versa. This operation, illustrated in Figure 1, is called a *2-swap*. Graphs that are related by a 2-swap are not isomorphic, but they have the same graphic matroids.

To avoid ambiguities such swapping, we must assume, at least, that G is 3-connected. There are a number of subtleties in how to even define an appropriate notion of generic global rigidity in the unlabeled setting. These are discussed in [15, 18].

1.2. Integer setting

Moving to an algorithmic setting, we will now assume that the points are placed at integer locations in the set $[B]$, with $B = 2^b$ for some positive integer b . For the labeled problem, we will want an algorithm that takes as input G and an integer vector $\ell \in [B]^m$, and outputs \mathbf{p} (up to congruence). For the unlabeled problem, we will want an algorithm that takes as input the integer vector ℓ , and outputs G (up to vertex relabeling) and \mathbf{p} (up to congruence).

Even the labeled problem is strongly NP-HARD by a result of Saxe [30]. Interestingly, one of his reductions is from SUBSET-SUM. This is a problem that, while hard in the worst case, can be efficiently solved, using the Lenstra-Lenstra-Lovász (LLL) lattice reduction algorithm [24], for most instances that have sufficiently large integers [22]. Indeed, we will take cue from this insight and use LLL to solve both the labeled and unlabeled reconstruction problems.

We say that a sequence, A_m , of events in a probability space occurs *with high probability as $m \rightarrow \infty$* (WHP) if $\Pr[A_m] = 1 - o(1)$. Here is our basic result.

Theorem 1.1. *Let G be a 3-connected (resp. 2-connected) graph with m edges. Let $b \geq m^2$. Let each p_i be chosen independently and uniformly at random from $[2^b]$. Then there is a polynomial time algorithm that succeeds, WHP, on the unlabeled (resp. labeled) reconstruction problems.*

The success probability, the convergence rate and the minimal required b can be estimated more carefully (see Proposition 3.9 and Remark 3.10 below). In any case, what we can prove seems to be pessimistic. Experiments (see below) show that our algorithm succeeds with very high probability on randomly chosen \mathbf{p} even with significantly smaller b . In part, this is because the LLL algorithm very often outperforms its worst case behavior [28] on random input lattices. On the other hand, lattice reduction is a brittle operation, and, for fixed m , it will fail ungracefully once b becomes too small.

1.3. Errors in input

To move towards a numerical setting, we need to differentiate between the geometric data in \mathbf{p} and the combinatorial data that records the order of the points of \mathbf{p} along the line. Since we only have access to the measurements ℓ , the ordering data will be represented by an orientation $\sigma_{\mathbf{p}}$ of a graph G produced by the algorithm; the orientation of an edge indicates the ordering of its endpoints on the line using the configuration \mathbf{p} . To model error or noise, we consider the case where there is $\{-1, 0, 1\}$ error, added adversarially, to each of the integer measurements ℓ_{ij} . Relative to our scale of 2^{m^2} , a $\{-1, 0, 1\}$ error is very small.

Theorem 1.2. *Let G be an ordered graph with m edges. For the labeled setting, assume G is 2-connected. For the unlabeled setting, assume G is 3-connected. Let $b \geq m^2$. Let each p_i be chosen independently and uniformly at random from $[2^b]$. Assume that $\{-1, 0, 1\}$ noise has been added adversarially to the ℓ_{ij} . Then, in both the labeled and unlabeled settings, there is a polynomial time algorithm that, WHP, correctly outputs G and $\sigma_{\mathbf{p}}$, up to reversing the orientation of every edge in $\sigma_{\mathbf{p}}$.*

The bound on b given here can be improved slightly, at the cost of a more complicated statement. Remark 3.10 describes such an improvement. In Section 4 we discuss ways in which our algorithm can fail.

Once we have the combinatorial data of G and $\sigma_{\mathbf{p}}$, it is easy to reconstruct \mathbf{p} , up to congruence, when ℓ has no error. We can simply use a spanning tree of G to lay out the p_i one at a time. If there are errors, then it is unlikely that there is any configuration corresponding to the observed measurements. However, we can apply a linear least squares algorithm to get an estimate of \mathbf{p} with small error.

Theorem 1.3. *Let G be a graph with m edges, and suppose that we are given measurements ℓ of an unknown configuration \mathbf{p} , with adversarial noise in the model of Theorem 1.2, and an orientation $\sigma_{\mathbf{p}}$ of G that is consistent with the ordering of the points in \mathbf{p} on the line.*

Then we can efficiently compute a vector $\mathbf{p}^ \in \mathbb{Q}^n$ so that, there is a vector $\mathbf{q} = \pm \mathbf{p} + t\mathbf{1}$ with*

$$\|\mathbf{p}^* - \mathbf{q}\| \leq n^{2.5} \sqrt{m}$$

where $\pm \mathbf{p}$ represents either \mathbf{p} or the configuration $-\mathbf{p}$ obtained from negating each of the points p_i in \mathbf{p} and $t\mathbf{1}$ represents translation of \mathbf{p} by an unknown rational number t .

Since we are working with signals at the scale of $B = 2^{m^2}$, this is a very small error indeed. We show in Remark 3.11 shows that when we have noise of larger magnitude than 1, the best approach is to round and truncate the input in such a way that the errors on each length measurement are in the set $\{-1, 0, 1\}$, albeit with a smaller b . As long as this new, smaller, b is sufficiently large, then our theorem will still apply. Informally, the important thing is not the number of error bits, but the number of signal bits. Additionally, Remark 3.12 shows that with sufficiently large m , we can tolerate any constant error maintained in our representation, or even a number of error bits that is bounded by a function that is $o(m)$.

1.4. Real setting

Theorem 1.2 can be used in the setting where \mathbf{p} is real valued. We suppose that each p_i is chosen independently from the uniform distribution on the closed interval $[1, B]$, which determines the underlying ℓ_{ij} . We also suppose that each ℓ_{ij} is corrupted by an adversarial error ε_{ij} with $-\frac{1}{2} \leq \varepsilon_{ij} \leq \frac{1}{2}$, so that the input data is

$$\ell_{ij} + \varepsilon_{ij}.$$

Let $\ell'_{ij} := |\text{nint}(p_i) - \text{nint}(p_j)|$, where nint is the nearest integer. Under these assumptions we have

$$|\ell'_{ij} - \text{nint}(\ell_{ij} + \varepsilon_{ij})| \leq 1.$$

As a result, the rounded data, $\text{nint}(\ell_{ij} + \varepsilon_{ij})$, will be a $\{-1, 0, 1\}$ noisy version of the lengths obtained from the rounded configuration, $\text{nint}(\mathbf{p})$. Hence, our Theorem 1.2 applies to this integer input. If we instead start with a real configuration \mathbf{p} with points lying in the closed interval $[0, 1]$, scaling by B and replaying the above construction shows that we can tolerate adversarial error on the order of $\frac{1}{2^{m^2}}$.

If we are given data with higher ε values, we can simply divide by an appropriate constant so that, afterwards, $-\frac{1}{2} \leq \varepsilon_{ij} \leq \frac{1}{2}$, and then proceed with a smaller effective b and B . As long as

b is large enough, Theorem 1.2 will still apply. In light of Remark 3.11, this approach is strictly better than keeping around additional error bits. Though in light of Remark 3.12, our theorem can tolerate $o(m)$ extra error bits.

1.5. Basic ideas

Consider traversing a cycle C in G , which is an ordered list of distinct vertices (except for the first and last, which are equal)

$$v_1, v_2, \dots, v_k = v_1$$

so that the edges $v_i v_{i+1}$ and $v_{k-1} v_1$ are in G . As we traverse C , we consider the configuration \mathbf{p} . Some of the edges go to the right ($p_{v_{i+1}} > p_{v_i}$) and some to the left ($p_{v_{i+1}} < p_{v_i}$). If we sum up the Euclidean lengths of these edges, with a +1 coefficient for edges that go to the right and a -1 coefficient for edges that go to the left, the sum is zero, since the cycle traversal starts and ends at p_{v_1} . Thus, the cycle C determines an integer relation on the coordinates of the length vector ℓ . The integer coefficients in this relation are all “small”, (the actual bounds we use will become apparent in Section 3.2) in fact only 0, -1, and 1, and, hence, so is the norm of the coefficient vector for such a cycle relation.

Let us call σ_p the ordering of the vertices along the line. Repeating the argument for each cycle in G defines more small integer relations on the coordinates of ℓ . We define S^{σ_p} to be the linear span, in \mathbb{R}^m , of all the coefficient vectors of integer relations defined by cycles in G . The space S^{σ_p} contains all the small integer relations that are forced on ℓ because it arises from measuring \mathbf{p} . Heuristically, if \mathbf{p} is selected uniformly at random, then ℓ should not satisfy *any other* small integer relation, because ℓ still behaves otherwise randomly.

In Section 3.2, we make this heuristic reasoning precise. What we show is that, if \mathbf{p} is selected uniformly at random from configurations of n integer points in $[0, B]$, then, WHP any integer relation that is satisfied by the coordinates of ℓ and has norm below a cutoff value is in S^{σ_p} . This gives us a way to find a basis for S^{σ_p} , provided that we can find a basis for an appropriate lattice with small enough basis vectors.

Finding a lattice basis consisting of vectors that are as short as possible is an NP-hard problem. However, the cutoff we need is large enough so that any LLL reduced basis for the lattice we construct will, WHP, allow us to find a basis for S^{σ_p} . Our analysis builds on work of Lagarias and Odlyzko [22] and Frieze [13] on random SUBSET-SUM problems. The main technical difference between our problem and random SUBSET-SUM instances is that ℓ is obtained from \mathbf{p} via a non-linear mapping. We handle this issue using a counting argument that extends those of Frieze [13]. Because the LLL algorithm [24] can find a reduced basis in polynomial time, this step of the algorithm is polynomial.

Once we recover S^{σ_p} , our next task is to use it to recover the ordered graph G . For this, we will use a polynomial time algorithm of Seymour [31] that constructs an unknown graph using an independence oracle for its graphic matroid. We can implement the oracle in polynomial time using the basis for S^{σ_p} found in the previous step, so we can recover G in polynomial time. To complete the combinatorial steps, we need to recover the orientation σ_p . Using S^{σ_p} , we can compute a consistent orientation for the edges of any cycle in G . We then find a globally consistent orientation, which must be $\pm\sigma_p$, using an iterative procedure based on an ear decomposition of G . In this step, we use 2-connectivity of G , because otherwise an ear decomposition will not exist.

Finally, we can use G , σ_p and ℓ to determine the positions of each of the p_i . In the noiseless

case, this can be done by traversing a spanning tree of G and laying out the points. If there is noise, we solve a least squares problem to approximate \mathbf{p} .

```

input :  $\ell$ 
output:  $(G, \mathbf{p})$ 

1  $S^{\sigma_p} \leftarrow \text{ComputeRelations}(\ell);$ 
2  $G \leftarrow \text{Seymour}(S^{\sigma_p});$ 
3  $\sigma_p \leftarrow \text{ComputeOrientation}(G, S^{\sigma_p});$ 
4  $\mathbf{p} \leftarrow \text{Layout}(G, \sigma_p, \ell);$ 
5 return  $(G, \mathbf{p});$ 
```

Algorithm 1: The full Algorithm

The methods in this paper do not directly generalize to higher dimensions. In fact, each of our steps poses a challenging problem once the dimension becomes at least two. There is no direct analogue to the ± 1 cycle relations, and instead, we would have to find integer polynomial relations on the measured lengths. To this end, we have no known non-trivial bound on either the degree or coefficient-sizes of these polynomials. Even if it were possible to recover the polynomials, and then, somehow, using a (so-far unknown) algorithmic version of [18], the unknown graph G , the remaining labeled graph realization problem is difficult in practice.

1.6. Related LLL Work

Recently in [14], a similar approach has been carefully applied to recover an unknown discrete signal from multiple randomly chosen large integer linear measurements. As described in [14], an algorithm based on LLL lattice reduction can even work in the presence of adversarially chosen error on the input, provided that the error is sufficiently small. Theorem 2.1 of [14] provides a detailed analysis of the relationship between the various parameters.

In other related recent work, [1] has shown that the LLL method can be used to efficiently solve the correspondence retrieval problem and the phase retrieval problem with no noise. Song et al. [34] have shown how LLL can be used to efficiently learn single periodic neurons in the low noise regime (this problem also generalizes phase retrieval). Work in [10, 38] shows how a number of other problems in machine learning, such as learning mixtures of Gaussians, can be solved by LLL, provided that the noise is sufficiently small. The main difference between our setting and those of [1, 10, 14, 22, 34, 38], is that in our case, we are looking for a number of linearly independent small relations in one random data vector, instead of a single small relation in one or more random data vectors.

In [14, 22], the data that satisfies a small integer relation (ℓ in our setting) is directly generated by a random process. In our work, as in the work of [1, 10, 34, 38], the input data is obtained indirectly from the random process. And in particular, in this paper, the process going from the randomly chosen configuration \mathbf{p} to the length measurements ℓ involves an absolute value operation.

In [14] and [10], the authors show that when one has $O(m)$ random data vectors that satisfy a common small integer relation, then LLL can be successfully applied with only around m bits of reliable data, instead of the m^2 bits that we need here. For our reconstruction problem, we only have one underlying random configuration \mathbf{p} , so we cannot exploit any additional linear relations to get a quantitative improvement.

In our random model, each p_i is chosen independently from a uniform distribution. In rigidity

applications, this is a natural assumption to make. Presumably, similar results could hold under some other sampling assumptions for the points p_i . In particular, the works in [1, 10, 34, 38] consider a Gaussian random process. What appears to be key is that the sampling distribution possesses some sufficient anti-concentration properties [10, 38].

2. Graph theoretic preliminaries

First we set up some graph-theoretic background.

2.1. Graphs, orientations, and measurements

In this paper, $G = (V, E)$ will denote a simple, undirected ordered graph with n ordered vertices and m ordered edges. These orderings will allow us to index various matrix rows and columns associated with edges and vertices.

Definition 2.1. Let $G = (V, E)$ be an undirected simple graph with n vertices and m edges. We denote undirected edges by $\{i, j\}$, with $i, j \in V$.

An *orientation* σ of G is a map that assigns each undirected edge $\{i, j\}$ of G to exactly one of the directed edges (i, j) or (j, i) ; i.e., $\sigma(\{i, j\}) \in \{(i, j), (j, i)\}$. Informally, σ assigns one endpoint of each edge to be the tail and the other to be the head.

An *oriented graph* G^σ is a directed graph obtained from an undirected graph G and an orientation σ of G by replacing each undirected edge $\{i, j\}$ of G with $\sigma(\{i, j\})$. ■

We are usually interested in a pair (G, \mathbf{p}) where G has n vertices and \mathbf{p} is a configuration of n points (p_1, \dots, p_n) on the line. Such a pair has a natural orientation of G associated with it.

Definition 2.2. Let (G, \mathbf{p}) denote a graph with n vertices and \mathbf{p} a configuration of n points. Define the *configuration orientation* $\sigma_{\mathbf{p}}$ by

$$\sigma_{\mathbf{p}}(\{i, j\}) = \begin{cases} (i, j) & \text{if } p_i < p_j \\ (j, i) & \text{if } p_i \geq p_j \end{cases} \quad (\text{for all } \{i, j\} \in E)$$

We say that an orientation σ of G is *vertex consistent* if $\sigma = \sigma_{\mathbf{p}}$ for some configuration \mathbf{p} . ■

There are 2^m possible orientations, but, when $m > n$, many fewer are vertex consistent.

Lemma 2.3. Let G be a graph with n vertices. There are at most $n!$ vertex consistent orientations of G .

Proof. The orientation $\sigma_{\mathbf{p}}$ is determined by the permutation of $\{1, \dots, n\}$ induced by the ordering of the p_i on the line. □

We now introduce some specialized facts about a standard linear representation of a graphic matroid over \mathbb{R} . We start by associating two vector configurations with a graph G .

Definition 2.4. Let G be a graph with n vertices and m edges. For $i \neq j$, both in V , a *signed edge incidence vector* $e_{(i,j)}$ has n coordinates indexed by V ; all the entries are zero except for the i th, which is -1 and the j th, which is $+1$. (So that $e_{(i,j)} = -e_{(j,i)}$.)

Given an ordered graph G and an orientation σ of G , the *signed edge incidence matrix*, $M^\sigma(G)$, has as its rows the vectors $e_{\sigma(\{i,j\})}$, over all of the edges $\{i,j\}$ of G .

Let $\rho = v_1, v_2, \dots, v_k = v_1$ be a cycle in G . The *signed cycle vector* $\mathbf{w}_{\rho,\sigma}$ of ρ in G^σ has m coordinates indexed by E . Coordinates corresponding to edges not traversed by ρ are zero. The coordinate corresponding to the edge $\{v_i, v_{i+1}\}$ is 1 if $\sigma(\{v_i, v_{i+1}\}) = (v_i, v_{i+1})$ and otherwise it is -1 .

If we have fixed some \mathbf{p} , we can shorten $\mathbf{w}_{\rho,\sigma_p}$ to \mathbf{w}_ρ (with no explicit orientation written). ■

It is easy to see that a signed cycle vector $\mathbf{w}_{\rho,\sigma_p}$ must be in the kernel of the transpose of $M^\sigma(G)$. Now we shift focus to an associated linear space.

Definition 2.5. Let G be a graph with n vertices and m edges and σ an orientation. An *ℝ-cycle* is a vector in the kernel of the transpose of $M^\sigma(G)$. The *ℝ-cycle space (cycle space*, for short) $S^\sigma \subseteq \mathbb{R}^m$ is the subspace of all ℝ-cycles. A basis of S^σ is *induced by graph cycles* if it contains only signed cycle vectors. ■

For our intended application, it is important that S^σ have a basis comprising integer vectors of small norm. It is a classical fact that this is the case.

Proposition 2.6 (See e.g, [29, Chapter 5]). *Let G be a connected ordered graph with n vertices and m edges and σ an orientation of G . The ℝ-cycle space S^σ , has dimension $c := m - n + 1$. Moreover, it is spanned by the signed cycle vectors in G^σ . In particular, S^σ has a basis induced by graph cycles.*

If we denote by $-\sigma$ the orientation of G obtained by reversing every edge in G^σ , it is easy to see that $S^\sigma = S^{-\sigma}$. We now describe a family of linear maps related to edge length measurements.

Definition 2.7. Let G be an ordered graph with n vertices and σ an orientation of G . We define the *oriented measurement map* $\ell_\sigma : \mathbb{R}^n \rightarrow \mathbb{R}^m$ by

$$\ell_\sigma(\mathbf{p})_{\{i,j\}} := \begin{cases} p_j - p_i & \text{if } \sigma(\{i,j\}) = (i,j) \\ p_i - p_j & \text{if } \sigma(\{i,j\}) = (j,i) \end{cases}$$

For any choice of σ , ℓ_σ is a linear map. We also define the *measurement map* ℓ by

$$\ell(\mathbf{p})_{\{i,j\}} = |p_j - p_i|$$

The measurement map is only piecewise linear. ■

Lemma 2.8. *Let (G, \mathbf{p}) be an ordered graph with n vertices and \mathbf{p} a configuration of n points. Let ρ be a cycle of G . Let \mathbf{w}_ρ be its signed cycle vector in G^{σ_p} . Then*

$$\mathbf{w}_\rho^\top \ell(\mathbf{p}) = 0$$

Proof. Unraveling definitions, we see that $\ell(\mathbf{p}) = \ell_{\sigma_p}(\mathbf{p})$, and that, in the standard bases for the domain \mathbb{R}^n and codomain \mathbb{R}^m , respectively, the matrix of the linear transformation ℓ_{σ_p} is $M^{\sigma_p}(G)$. We then compute

$$\mathbf{w}_\rho^\top \ell(\mathbf{p}) = \mathbf{w}_\rho^\top \ell_{\sigma_p}(\mathbf{p}) = \mathbf{w}_\rho^\top M^{\sigma_p}(G)\mathbf{p} = 0$$

because \mathbf{w}_ρ^\top is in the cokernel of $M^{\sigma_p}(G)$ by definition. □

3. The algorithm

In this section we describe the algorithm in detail and develop the tools required for its analysis. We will focus on the harder, unlabeled setting, and afterwards describe the various simplifications that can be applied in the labeled case.

3.1. The sampling model

The algorithm has the following parameters. We have a precision $b \in \mathbb{N}$. There is fixed ordered graph G with n vertices and m edges. Our algorithm takes as input $\ell = \ell(\mathbf{p}) + \varepsilon$, where \mathbf{p} is selected uniformly at random among configurations of n points with coordinates in $[2^b]$ and ε is an error vector in $\{-1, 0, 1\}^m$.

The algorithm does not have access to n , only m (via the number of coordinates in ℓ). All the probability statements are taken with respect to \mathbf{p} , which is not part of the input. This complicates the probabilistic analysis a little bit relative to [13, 22], but much remains similar.

3.2. Reconstruct S^{σ_p}

In this section, we will show how, WHP to reconstruct S^{σ_p} from ℓ . Our main tool will be the LLL algorithm [24], as used by [22], and analyzed by [13].

Definition 3.1. For any vector \mathbf{r} in \mathbb{Z}^m define the *lattice* $\text{Lat}(\mathbf{r})$ in \mathbb{Z}^{m+1} as the integer span of the columns of the following $(m+1)$ -by- m *lattice generating matrix*

$$L(\mathbf{r}) := \begin{pmatrix} I_m \\ \mathbf{r}^\top \end{pmatrix}$$

where I_m is the $m \times m$ identity matrix. ■

Remark 3.2. Frieze [13] multiplies the last row of $L(\mathbf{r})$ by a large constant. We do not do that, so that we can easily deal with noise. In [16], we elaborate on this issue and show how, in the original context of SUBSET-SUM, Frieze's analysis goes through without the use of the large constant. ◇

In our algorithm we will work over the lattice $\text{Lat}(\ell)$, where ℓ is the edge length measurements of the unknown, configuration \mathbf{p} , plus adversarially chosen noise. Figure 2 summarizes the algorithm in pseudo-code.

```

input :  $\ell$ 
output:  $S^{\sigma_p}$ 

1  $m \leftarrow |\ell|$ ;
2  $\Lambda \leftarrow \text{LLL}(\text{Lat}(\ell))$ ;
3  $\text{basis} \leftarrow \text{threshold}(\Lambda, \sqrt{2m}2^{m/2})$ ;
4  $S^{\sigma_p} \leftarrow \text{truncate}(\text{basis})$ ;
5 return ( $S^{\sigma_p}$ );

```

Algorithm 2: The ComputeRelations Algorithm

If \mathbf{x} is a vector in \mathbb{R}^m and $a \in \mathbb{R}$ a real number, we denote by $\begin{pmatrix} \mathbf{x} \\ a \end{pmatrix}$ the vector in \mathbb{R}^{m+1} that has \mathbf{x} as its first m coordinates and a as its $(m+1)$ st. We call \mathbf{x} the *truncation* of $\begin{pmatrix} \mathbf{x} \\ a \end{pmatrix}$. Now we can describe the algorithm to reconstruct S^{σ_p} in more detail.

- Given, ℓ , we let m be the number of coordinates in ℓ . This is the number of edges of G . The parameter m determines a threshold value $\sqrt{2m}2^{m/2}$.
- We then use the LLL algorithm to compute a reduced basis Λ for the lattice $\text{Lat}(\ell)$.
- Finally, we truncate every vector in Λ of norm at most the threshold value $\sqrt{2m}2^{m/2}$ and return these as the basis for S^{σ_p} .

The rest of the section contains the analysis of this algorithm. Fix an ordered graph G , a configuration \mathbf{p} and an error vector ε , which determine the observed measurements $\ell = \ell(\mathbf{p}) + \varepsilon$. Because ℓ arises from measuring an integer configuration on the line, $\text{Lat}(\ell)$ will deterministically contain some short vectors associated with generators of the \mathbb{R} -cycle space S^{σ_p} .

Definition 3.3. Let G be an ordered graph with n vertices and m edges, $\varepsilon \in \mathbb{Z}^m$, and an orientation σ of G be fixed. Let S^σ be the \mathbb{R} -cycle space of G^σ . We define

$$\hat{S}^\sigma := \left\{ \begin{pmatrix} \mathbf{v} \\ \varepsilon^\top \mathbf{v} \end{pmatrix} : \mathbf{v} \in S^\sigma \right\}$$

The subspace $\hat{S}^\sigma \subseteq \mathbb{R}^{m+1}$ is the image of S^σ under an injective linear transformation. ■

Lemma 3.4. Let G be a connected ordered graph with n vertices and m edges, σ an orientation of G , \mathbf{p} a configuration of n points, and $\varepsilon \in \{-1, 0, 1\}^m$ an error vector. Then \hat{S}^σ is contained in the linear span of a set of vectors in $\text{Lat}(\ell)$, each of norm at most $\sqrt{2m}$.

Proof. Let \mathbf{w}_ρ be the signed cycle vector of a cycle ρ in the oriented graph G^{σ_p} . Define $e_\rho = \mathbf{w}_\rho^\top \ell$. From Lemma 2.8, we have

$$e_\rho = \mathbf{w}_\rho^\top \ell = \mathbf{w}_\rho^\top (\ell(\mathbf{p}) + \varepsilon) = \mathbf{w}_\rho^\top \varepsilon$$

We then define $\hat{\mathbf{w}}_\rho := \begin{pmatrix} \mathbf{w}_\rho \\ e_\rho \end{pmatrix}$. For any $\mathbf{r} \in \mathbb{Z}^m$, the column span of the lattice generating matrix $L(\mathbf{r})$ is exactly vectors of the form

$$\hat{\mathbf{x}} = \begin{pmatrix} \mathbf{x} \\ \mathbf{r}^\top \mathbf{x} \end{pmatrix}. \quad (1)$$

The vector $\hat{\mathbf{w}}_\rho$ is in $\text{Lat}(\ell)$, because it is integral and is in the column span of $L(\ell)$.

From Proposition 2.6, there is a basis $\mathbf{w}_{\rho_1}, \dots, \mathbf{w}_{\rho_c}$ of signed cycle vectors for S^{σ_p} , where $c := m - n + 1$. As the images under the injective linear map of Definition 3.3, the vectors $\hat{\mathbf{w}}_{\rho_1}, \dots, \hat{\mathbf{w}}_{\rho_c}$ are linearly independent, and, by the discussion above, in $\text{Lat}(\ell)$. Because the vectors \mathbf{w}_{ρ_i} and ε are ± 1 vectors, we get, for each $1 \leq i \leq c$, the estimate

$$\|\hat{\mathbf{w}}_\rho\|_2^2 = \|\mathbf{w}_\rho\|_2^2 + |e_\rho|^2 \leq 2m$$

It now follows that $\text{Lat}(\ell)$ contains c linearly independent vectors of norm at most $\sqrt{2m}$ that are contained in \hat{S}^σ . As \hat{S}^σ is c -dimensional, these must be a spanning set. □

Now we proceed with the probabilistic analysis of our algorithm. The idea, which is already present in Frieze's work [13], is that if ℓ (instead of \mathbf{p}) has been selected uniformly at random, then $\text{Lat}(\ell)$ would, WHP, not contain any vector of length $\sqrt{2m}2^{m/2}$. The point of Lemma 3.4 is that, because ℓ is a (corrupted) measurement of a configuration \mathbf{p} , $\text{Lat}(\ell)$ must contain some "planted" short vectors of norm at most $\sqrt{2m}$ that span the linear space \hat{S}^σ . We won't be able to recover these efficiently, so the next step of the analysis is to show that, WHP (in \mathbf{p}), every vector of norm at most $\sqrt{2m}2^{m/2}$ in $\text{Lat}(\ell)$ lies in \hat{S}^σ . Informally, $\text{Lat}(\ell)$ is still "random enough" looking that we can spot the vectors in it that arise from cycle relations.

Definition 3.5. We call any vector of norm at most $\sqrt{2m}2^{m/2}$ *medium sized*. ■

We defer the proof of the following key estimate to Section 3.3.1.

Proposition 3.6. Let G be a connected graph with m edges and n vertices, $\varepsilon \in \{-1, 0, 1\}^m$, and $\hat{\mathbf{x}} \in \mathbb{Z}^{m+1}$. If \mathbf{p} is selected uniformly at random from $[B]^n$, then the probability that $\hat{\mathbf{x}}$ is in $\text{Lat}(\ell(\mathbf{p}) + \varepsilon)$ but $\hat{\mathbf{x}} \notin \hat{S}^{\sigma_{\mathbf{p}}}$ is at most $\frac{2^n n!}{B}$.

Using the key estimate, we prove the result mentioned in the informal discussion above.

Lemma 3.7. Let G be a connected graph with m edges and n vertices. If \mathbf{p} is selected uniformly at random from $[B]^n$, then, with probability at least

$$1 - \frac{2^{m^2/2} 2^{O(m \log m)}}{B}$$

for every $\varepsilon \in \{-1, 0, 1\}^m$, every medium sized vector in $\text{Lat}(\ell(\mathbf{p}) + \varepsilon)$ is in $\hat{S}^{\sigma_{\mathbf{p}}}$.

Proof. For the moment, fix $\varepsilon \in \{-1, 0, 1\}$. Set $K = (2m)^{1/2}2^{m/2}$. If $\hat{\mathbf{x}} \in \mathbb{Z}^{m+1}$ is a medium sized integer vector, then certainly all its coordinates have magnitude at most K . Hence, there are at most $(2K + 1)^{m+1} \leq (3K)^{m+1}$ medium sized integer vectors. We bound this latter quantity as follows

$$(3(2m)^{1/2}2^{m/2})^{m+1} \leq 2^{\frac{m(m+1)}{2}} 2^{\frac{m+1}{2}} 3^{(m+1)} m^{\frac{m+1}{2}} = 2^{m^2/2} 2^{O(m \log m)}$$

With ε fixed, Proposition 3.6 then implies that, if \mathbf{p} is selected uniformly at random from $[B]^n$, the probability that there is some medium sized $\hat{\mathbf{x}}$ so that $\hat{\mathbf{x}} \in \text{Lat}(\ell(\mathbf{p}) + \varepsilon)$ and $\hat{\mathbf{x}} \notin \hat{S}^{\sigma_{\mathbf{p}}}$ is at most

$$\frac{2^n n!}{B} 2^{m^2/2} 2^{O(m \log m)}$$

Finally, we let ε vary over 3^m choices to conclude that the statement fails with probability at most

$$3^m \frac{2^n n!}{B} 2^{m^2/2} 2^{O(m \log m)}$$

Because G is connected it has at least $n - 1$ edges, so we can absorb the term $3^m 2^n n!$ into the $2^{O(m \log m)}$. The final failure probability is

$$\frac{2^{m^2/2} 2^{O(m \log m)}}{B}$$

as claimed. □

Now picking B to make the probability of success high is routine.

Lemma 3.8. *Let $b \geq (\frac{1}{2} + \delta)m^2$ for fixed $\delta > 0$. For m sufficiently large (depending on δ), if \mathbf{p} is selected uniformly at random from $[2^b]^n$, with probability at least $1 - \frac{1}{2^{\delta m^2/2}}$, for every $\varepsilon \in \{-1, 0, 1\}^n$, if $\hat{\mathbf{x}}$ is medium sized and $\hat{\mathbf{x}} \in \text{Lat}(\ell(\mathbf{p}) + \varepsilon)$, then $\hat{\mathbf{x}} \in \hat{S}^{\sigma_{\mathbf{p}}}$.*

Proof. For sufficiently large m , the quantity $2^{m^2/2} 2^{O(m \log m)}$ is bounded by $2^{(\frac{1}{2} + \frac{\delta}{2})m^2}$. The probability of the event in the statement fails is, by Lemma 3.7, is then at most

$$\frac{2^{(\frac{1}{2} + \frac{\delta}{2})m^2}}{2^{(\frac{1}{2} + \delta)m^2}} = \frac{1}{2^{\delta m^2/2}}$$

□

At this point, we can analyze the algorithm.

Proposition 3.9. *For all $\delta > 0$, there is an $M \in \mathbb{N}$ and a polynomial time algorithm that correctly computes, with probability at least $1 - 2^{-\delta m^2/2}$, for any connected ordered graph G with n vertices and $m \geq M$ edges, and any noise vector $\varepsilon \in \{-1, 0, 1\}^m$, a basis for the space $S^{\sigma_{\mathbf{p}}}$ from $\ell = \ell(\mathbf{p}) + \varepsilon$. The probability is with respect to \mathbf{p} chosen uniformly at random among configurations of n integer points in $[B]$, where $B = 2^b$ and $b \geq (\frac{1}{2} + \delta)m^2$.*

Proof. Let us recall that, given ℓ , the algorithm first computes an LLL reduced basis Λ , of $\text{Lat}(\ell)$ and then returns the truncation of every medium sized vector in Λ .

By Lemma 3.4, $\text{Lat}(\ell)$ contains at least c linearly independent vectors of norm at most $\sqrt{2m}$. As these vectors can be extended to a lattice basis, the approximation guarantee of an LLL reduced basis [24, Proposition 1.12] implies that Λ must contain at least c vectors of norm at most $\sqrt{2m}2^{(m-1)/2}$. Let us call these $\hat{\mathbf{v}}_1, \dots, \hat{\mathbf{v}}_c$.

Since $b \geq (\frac{1}{2} + \delta)m^2$, by Lemma 3.8, there is an $M > 0$ so that, if $m > M$, with probability at least $1 - 2^{-\delta m^2/2}$, all the vectors $\hat{\mathbf{v}}_i$ are in the c -dimensional linear space $\hat{S}^{\sigma_{\mathbf{p}}}$. As the $\hat{\mathbf{v}}_i$ are linearly independent, they are a basis for $\hat{S}^{\sigma_{\mathbf{p}}}$. The algorithm returns the truncated vectors $\mathbf{v}_1, \dots, \mathbf{v}_c$. By construction, the \mathbf{v}_i are in $S^{\sigma_{\mathbf{p}}}$. They must also be linearly independent. Otherwise the $\hat{\mathbf{v}}_i$ would be linearly dependent as the image of a dependent set under the linear transformation of Definition 3.3. Hence, the algorithm returns a basis for $S^{\sigma_{\mathbf{p}}}$ with probability at least $1 - 2^{-\delta m^2/2}$.

The LLL algorithm runs in polynomial time [24], and all the other steps are evidently polynomial, so the algorithm to compute a basis of $S^{\sigma_{\mathbf{p}}}$ runs in polynomial time. □

We note, for later, than the polynomial running time of the algorithm to find a basis for $S^{\sigma_{\mathbf{p}}}$ implies that the coordinates of the vectors it returns have a number of bits that is polynomially bounded in m .

3.3. More Error/Fewer Bits

In the following remarks, we touch on a number of noise related issues.

Remark 3.10. Lemma 3.8, Proposition 3.9, and our main Theorems 1.1 and 1.2 can be generalized somewhat. In particular we can set $b \geq (\frac{1}{2} + \delta)m^2 - f(m)$ where $f(m) = o(m^2)$. In this case, the result can still be shown to hold for sufficiently large values of m (which now depends on f as well as δ). ◇

Remark 3.11. Suppose that instead of errors in $\{-1, 0, 1\}$, we had integer errors in $\{-(2^k - 1), \dots, 2^k - 1\}$, for some k . The estimate on the size of the small vectors in Lemma 3.4 increases somewhat to $\sqrt{2(2^k - 1)m}$. This means, that in the proof of Lemma 3.7, we need to increase the quantity K to $\sqrt{2(2^k - 1)m}2^{m/2}$ and replace the 3^m choices of ε with $(2^{k+1} - 1)^m$. This second change raises the failure probability in the conclusion of Lemma 3.7 by a factor of $(\frac{2^{k+1}-1}{3})^m$.

Alternatively, by dividing by 2^{k+1} , we could model the resulting rational numbers as a real valued input, as in Section 1.4, with errors bounded in magnitude by $1/2$. Then we can round the data, putting us back to the $\{-1, 0, 1\}$ error model, but on integers in the interval $\{1, \dots, 2^{b-k-1}\}$. Then in the proof of Lemma 3.7, the $B = 2^b$ term in the denominator would become 2^{b-k-1} . This would raise the probability of failure instead by only 2^{k+1} .

Since $(\frac{2^{k+1}-1}{3})^m$ is bigger than 2^{k+1} , we see, that if we have integer errors in $\{-(2^k - 1), \dots, 2^k - 1\}$, for some k , we are much better off rounding off the lowest $k + 1$ bits and treating this as $\{-1, 0, 1\}$ error, than keeping around the noise! This observation is in line with the conclusions of [14]. \diamond

Remark 3.12. If we do have errors of magnitude bounded by some constant K , the proof of Lemma 3.7 can be modified to accommodate the larger error. Likewise Lemma 3.8 and its proof still hold but the sufficiently large values required for m will need to increase. Indeed, as long as $\log(K) = o(m)$ Lemma 3.8 can still be shown to hold. For this generalization, the statement of Lemma 3.7 needs to have the $O(m \log m)$ term replaced by $o(m^2)$. This is in line with [14, Theorem 2.1], where the role of error is tracked more carefully than we are doing here. \diamond

3.3.1. Proof of Proposition 3.6 Now we develop the proof of Proposition 3.6. In this section, we fix a connected ordered graph G with n vertices and m edges, as well as an error vector $\varepsilon \in \{-1, 0, 1\}^m$ and any vector $\hat{\mathbf{x}} \in \mathbb{Z}^{m+1}$. Let $B \in \mathbb{N}$ be given, and define

$$\mathcal{S}(\hat{\mathbf{x}}, \varepsilon) = \{\mathbf{p} \in [B]^n : \hat{\mathbf{x}} \in \text{Lat}(\ell(\mathbf{p})) + \varepsilon \text{ and } \hat{\mathbf{x}} \notin \hat{S}^{\sigma_{\mathbf{p}}}\}$$

The Proposition asserts that

$$|\mathcal{S}(\hat{\mathbf{x}}, \varepsilon)| \leq 2^n n! B^{n-1} \tag{2}$$

Establishing (2) is complicated by the fact that the measurement map ℓ is non-linear. To deal with this, we first define, for each orientation σ of G , the set

$$\mathcal{S}(\hat{\mathbf{x}}, \varepsilon, \sigma) = \{\mathbf{p} \in \mathcal{S}(\hat{\mathbf{x}}, \varepsilon) : \sigma_{\mathbf{p}} = \sigma\}$$

And we have

$$|\mathcal{S}(\hat{\mathbf{x}}, \varepsilon)| = \sum_{\sigma} |\mathcal{S}(\hat{\mathbf{x}}, \varepsilon, \sigma)| \tag{3}$$

where the sum ranges over all vertex consistent orientations of G . By Lemma 2.3, there are at most $n!$ vertex consistent orientations.

Next we fix a σ so that $\mathcal{S}(\hat{\mathbf{x}}, \varepsilon, \sigma)$ is non-empty. For any \mathbf{p} in $\mathcal{S}(\hat{\mathbf{x}}, \varepsilon, \sigma)$, we know that $\ell(\mathbf{p}) = \ell_{\sigma}(\mathbf{p})$. The map $\ell_{\sigma}(\mathbf{p})$ is linear. Since G is connected, Proposition 2.6 implies that the rank of ℓ_{σ} is $n - 1$, so it has a one-dimensional kernel. It follows that the fiber $\ell_{\sigma}^{-1}(\ell_{\sigma}(\mathbf{p}))$ consists only of translations of \mathbf{p} . We observe that there are at most B of these where the points p_i remain in the range $[B]$.

Meanwhile, because $\mathbf{p} \in \mathcal{S}(\hat{\mathbf{x}}, \varepsilon, \sigma)$, the measurement $\ell(\mathbf{p}) + \varepsilon$ must be in the set

$$\mathcal{B}(\hat{\mathbf{x}}, \sigma) = \{\mathbf{r} \in \{-1, \dots, B\}^m : \hat{\mathbf{x}} \in \text{Lat}(\mathbf{r}), \hat{\mathbf{x}} \notin \hat{S}^\sigma, \text{ and } \hat{S}^\sigma \text{ is in the linear span of } \text{Lat}(\mathbf{r})\}$$

where we have used Lemma 3.4 to ensure that $\text{Lat}(\ell(\mathbf{p}) + \varepsilon)$ contains a basis for \hat{S}^σ . From the observations above, we obtain

$$|\mathcal{S}(\hat{\mathbf{x}}, \varepsilon, \sigma)| \leq B |\mathcal{B}(\hat{\mathbf{x}}, \sigma)| \quad (4)$$

The next task is to bound the size of a non-empty $\mathcal{B}(\hat{\mathbf{x}}, \sigma)$. We do this by showing that such a $\mathcal{B}(\hat{\mathbf{x}}, \sigma)$ is contained in an $(n - 2)$ -dimensional affine subspace of \mathbb{R}^m . Our main tool will be the following simple fact.

Lemma 3.13. *Let A be a d -dimensional affine subset of \mathbb{R}^m . Then*

$$|\{-1, 0, \dots, B\}^m \cap A| \leq (B + 2)^d \leq (2B)^d$$

Proof. For convenience, set $C^m = \{-1, 0, \dots, B\}^m$ and define $C^d = \{-1, 0, \dots, B\}^d$. Without loss of generality, the projection of A onto the first d coordinates of \mathbb{R}^m by forgetting the last $m - d$ coordinates is bijective (if not, one can pick a different coordinate subspace), and in particular injective. This affine map sends points in $A \cap C^m$ to points in C^d . It follows that $|A \cap C^m| \leq |C^d| = (B + 2)^d$. Since $B \geq 2$, then $B + 2 \leq 2B$. \square

Returning to the proof of the Proposition, we let $\{\hat{\mathbf{v}}_1, \dots, \hat{\mathbf{v}}_c\}$ be some fixed basis for \hat{S}^σ ; recall that, because G is connected the dimension of \hat{S}^σ is $c = m - n + 1$. Each of these basis vectors is of the form $\hat{\mathbf{v}}_i = \begin{pmatrix} \mathbf{v}_i \\ f_i \end{pmatrix}$ for some \mathbf{v}_i and f_i . Let us record for later the following linear system with variable $\mathbf{r} \in \mathbb{R}^m$:

$$\begin{pmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_c^\top \end{pmatrix} \mathbf{r} = \begin{pmatrix} f_1 \\ \vdots \\ f_c \end{pmatrix} \quad (5)$$

Since the \mathbf{v}_i are linearly independent, the rank of this system is c . (For linear independence recall that if the \mathbf{v}_i were dependent, then the $\hat{\mathbf{v}}_i$ would have to be linearly dependent as the image of a dependent set under the linear transformation of Definition 3.3.)

Now suppose that for some fixed $\mathbf{r} \in \mathbb{R}^m$, $\mathbf{r} \in \mathcal{B}(\hat{\mathbf{x}}, \sigma)$. This implies that $\text{Lat}(\mathbf{r})$ contains a basis $\hat{\mathbf{w}}_1, \dots, \hat{\mathbf{w}}_c$ of \hat{S}^σ . Each of these basis vectors is of the form $\hat{\mathbf{w}}_i = \begin{pmatrix} \mathbf{w}_i \\ e_i \end{pmatrix}$, and as each $\hat{\mathbf{w}}_i$ is in $\text{Lat}(\mathbf{r})$, \mathbf{r} satisfies (recalling Equation (1)) the following linear system:

$$\begin{pmatrix} \mathbf{w}_1^\top \\ \vdots \\ \mathbf{w}_c^\top \end{pmatrix} \mathbf{r} = \begin{pmatrix} e_1 \\ \vdots \\ e_c \end{pmatrix} \quad (6)$$

Meanwhile, putting the $\hat{\mathbf{v}}_i$ and $\hat{\mathbf{w}}_j$ as the columns of $(m + 1) \times c$ matrices, we can find an invertible $c \times c$ matrix W so that

$$(\hat{\mathbf{v}}_1 \quad \cdots \quad \hat{\mathbf{v}}_c) = (\hat{\mathbf{w}}_1 \quad \cdots \quad \hat{\mathbf{w}}_c) W$$

it now follows that \mathbf{r} must also satisfy this next linear system, which which is equivalent to (6)

$$W^\top \begin{pmatrix} \mathbf{w}_1^\top \\ \vdots \\ \mathbf{w}_c^\top \end{pmatrix} \mathbf{r} = W^\top \begin{pmatrix} e_1 \\ \vdots \\ e_c \end{pmatrix}$$

But this system is exactly (5). So we see that \mathbf{r} satisfies (5) whether or not the $\hat{\mathbf{v}}_i$ are in $\text{Lat}(\mathbf{r})$.

Up to this point, we have not yet used the other hypothesis coming from our assumption that $\mathbf{r} \in \mathcal{B}(\hat{\mathbf{x}}, \sigma)$: that $\hat{\mathbf{x}} \in \text{Lat}(\mathbf{r})$ and $\hat{\mathbf{x}} \notin \hat{S}^\sigma$. Now we will. Let $\hat{\mathbf{x}} = \begin{pmatrix} \mathbf{x} \\ g \end{pmatrix}$. Because $\hat{\mathbf{x}} \in \text{Lat}(\mathbf{r})$, \mathbf{r} satisfies the linear equation

$$\mathbf{x}^\top \mathbf{r} = g$$

and, because $\mathcal{B}(\hat{\mathbf{x}}, \sigma)$ is not empty, we have $\hat{\mathbf{x}} \notin \hat{S}^\sigma$, so the rank of the the linear system that adds this equation to (5), namely

$$\begin{pmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_c^\top \\ \mathbf{x}^\top \end{pmatrix} \mathbf{r} = \begin{pmatrix} f_1 \\ \vdots \\ f_c \\ g \end{pmatrix} \quad (7)$$

must be $c + 1$. Because we assume that $\mathcal{B}(\hat{\mathbf{x}}, \sigma)$ is non-empty, this final system (7) is consistent, and has, as its solution space, an affine subspace A of dimension $m - (c - 1) = n - 2$. Since (7) depends only on the data defining $\mathcal{B}(\hat{\mathbf{x}}, \sigma)$, we conclude that

$$\mathcal{B}(\hat{\mathbf{x}}, \sigma) \subseteq A \cap \{-1, \dots, B\}^m$$

And now using Lemma 3.13 we get

$$|\mathcal{B}(\hat{\mathbf{x}}, \sigma)| \leq (2B)^{n-2} \quad (8)$$

Finally, plugging (8) into (4), we get

$$|\mathcal{S}(\hat{\mathbf{x}}, \varepsilon, \sigma)| \leq 2^n B^{n-1}$$

whenever the l.h.s. is non-zero. Plugging this estimate into (3), and summing over the at most $n!$ non-zero terms, we obtain (2), and the proposition is proved. \square

3.4. Reconstruct G

Next we show how to reconstruct G from S^{σ_p} . Our main tool will be an algorithm by Seymour [31]. First we recall some standard facts about graphic matroids. A standard reference for matroids is [29]; the material in this section can be found in [29, Chapter 5].

Definition 3.14. Let $G = (V, E)$ be an ordered graph. The *graphic matroid* of G is the matroid on the ground set E that has, as its independent sets, the subsets of E corresponding to acyclic subgraphs of G . \blacksquare

If G is connected and has n vertices, the rank of its graphic matroid is $n - 1$. The following lemma is due to Whitney [37].

Lemma 3.15. Let G^σ be any orientation of an ordered graph $G = (V, E)$. Then a subset $E' \subseteq E$ is independent in the graphic matroid of G if and only if $S^\sigma \subseteq \mathbb{R}^m$ contains no non-zero vector with support contained in E' .

Definition 3.16. Let \mathcal{M} be a matroid defined on a ground set E . An *independence oracle* for \mathcal{M} takes as its input a subset E' of E and outputs “yes” or “no” depending on whether E' is independent in \mathcal{M} . ■

Since G has n vertices and is connected, the dimension of S^σ is $m - n + 1$. Given an $m \times (m - n + 1)$ matrix A that has as its columns a basis of S^σ , we can check whether a set $E' \subseteq E$ is independent follows. Let A' be the matrix obtained from A by discarding the rows corresponding to E' . If A' has a non-zero vector \mathbf{v} in its null space then $A\mathbf{v}$ has zeros in coordinates corresponding to edges outside of E' . Since \mathbf{v} is non-zero and A has independent columns, $A\mathbf{v}$ is non-zero, so, by Lemma 3.15, E' is dependent. On the other hand, if S^σ contains a non-zero vector \mathbf{w} with support in E' , there is a \mathbf{v} so that $\mathbf{w} = A\mathbf{v}$, and this \mathbf{v} must be in the null space of A' .

It now follows that, if every entry of A has a number of bits polynomially bounded in n and m , we can check independence of E' in polynomial time by forming A' and computing its rank with Gaussian elimination. This gives us a polynomial time independence oracle for the graphic matroid of G .

A striking result of Seymour [31], which builds on work of Tutte [35] (see also [3]) is that an independence oracle can be used to efficiently determine a graph G from its matroid.

Theorem 3.17 ([31]). *Let \mathcal{M} be a matroid over an ordered ground set. There is an algorithm that uses an independence oracle to determine whether \mathcal{M} is the graphic matroid of some ordered graph G . If so, the algorithm outputs some such G . If the independence oracle is polynomial time, so is the whole algorithm.*

Since the matroid of an ordered graph G depends only on whether each subset of edges is a cycle or not, if G and H are isomorphic as ordered graphs, they will have the same matroid and so Seymour’s algorithm can only return an ordered graph H that is isomorphic to G . A foundational result of Whitney [36] says that if an (ordered) graph G is 3-connected, then G is determined up to (ordered) graph isomorphism by its graphic matroid.

Corollary 3.18. *If G is 3-connected, then Seymour’s algorithm outputs an ordered graph isomorphic to G , given an independence oracle for the graphic matroid of G .*

Which for our purposes, together with Lemma 3.15 gives us:

Proposition 3.19. *Let G be a 3-connected ordered graph with n vertices and m edges. Let σ be an orientation and A be a $m \times c$ matrix with column span S^σ . Suppose that every entry in A has a bit complexity that is polynomial in n and m . Then there is an algorithm, with running time polynomial in n and m that correctly computes G from A , up to isomorphism.*

3.5. Reconstructing σ_p

Next we show how to reconstruct σ_p from G and S^{σ_p} .

Lemma 3.20. *Let G^σ be any orientation of an ordered graph $G = (V, E)$. Let E' be the edge set of a cycle ρ in G . Then any non-zero vector in S^σ with support contained in coordinates corresponding to E' is a scale of the signed cycle vector $\mathbf{w}_{\rho, \sigma}$.*

Proof. As a cycle, every proper subset of E' must be independent in the graphic matroid of G . Hence, by Lemma 3.15, any non-zero vector in S^σ that is supported on coordinates corresponding to E' must have no zero entries. If there are two such vectors that are not scalar multiples of each other, then an appropriate linear combination would be non-zero and have smaller support. The resulting contradiction says that, up to scale there is a single vector in S^σ supported on E' ; since $\mathbf{w}_{\rho, \sigma}$ is such a vector, we are done. \square

If we have an $m \times (m - n + 1)$ matrix A with column span S^σ , we can find the vector $\pm \mathbf{w}_{\rho, \sigma}$ from the edge set E' of ρ by finding a non-zero vector \mathbf{v} in the null space of the matrix A' obtained by discarding the rows of A corresponding to E' . We then have $\alpha \mathbf{w}_{\rho, \sigma} = A\mathbf{v}$, for some scalar $\alpha \in \mathbb{R}$. Lemma 3.20 implies that \mathbf{v} is unique up to scale, and then we can scale $A\mathbf{v}$ to obtain a vector with only entries ± 1 , which must be $\pm \mathbf{w}_{\rho, \sigma}$.

In the algorithm to recover σ_p , we will use the following graph-theoretic notion.

Definition 3.21. Let G be a graph. An *ear decomposition* of G is a sequence of cycles

$$\rho_1, \dots, \rho_k$$

such that, for each $2 \leq j \leq k$

$$\rho_j \setminus (\rho_1 \cup \dots \cup \rho_{j-1})$$

is a path, and the edge set of G is the union of the cycles ρ_i . \blacksquare

A graph is 2-connected, if and only if it has an ear decomposition [11, Proposition 3.1.1], and an ear decomposition can be computed in polynomial time (e.g., [25]).

Proposition 3.22. Let $G = (V, E)$ be a 2-connected ordered graph, and A an $m \times (m - n + 1)$ matrix whose column span, S^σ , is the \mathbb{R} -cycle space of an unknown orientation of G . Also assume that the entries of A have a number of bits that is bounded polynomially in m . Then we can find σ from this data in polynomial time, up to a global flipping of the entire orientation.

Proof. Lemma 3.20 allows us to construct, an orientation of any cycle in G that is consistent with σ or the orientation $-\sigma$, in which all the edges are reversed. To reconstruct σ (or $-\sigma$), we will iteratively orient the whole graph, working one cycle at a time, in a way that makes sure that all the cycles are oriented consistently with each other, and thus globally correct up to a single sign.

We first compute an ear decomposition

$$\rho_1, \dots, \rho_k$$

Using the matrix A and Lemma 3.20, we find a signed cycle vector $\mathbf{w}_1 = \pm \mathbf{w}_{\rho_1, \sigma}$. Now, for each $j = 2, \dots, k$, we use A and Lemma 3.20 to find a vector $\mathbf{w}_j = \pm \mathbf{w}_{\rho_j, \sigma}$ so that the signs in coordinates corresponding to edges in $\rho_j \cap (\cup_{i=1}^{j-1} \rho_i)$ agree with the signs in $\mathbf{w}_1, \dots, \mathbf{w}_{j-1}$. The defining property of an ear decomposition is that

$$\rho_j \setminus (\cup_{i=1}^{j-1} \rho_i)$$

is a path. Because ρ_j is a cycle, this implies that

$$\rho_j \cap (\cup_{i=1}^{j-1} \rho_i)$$

is non-empty. Hence, exactly one of the two possible orientations of ρ_j consistent with σ or $-\sigma$ will also be consistent with the previous choices of the algorithm. At the end of the process, the coordinate corresponding to every edge is either ± 1 in at least one of the vectors \mathbf{w}_j , and the signs will be the same whenever non-zero. Hence, we may read off the orientation σ , up to an unknown sign.

Each of the steps is polynomial time. As discussed above, an ear decomposition can be computed in polynomial time. The number of cycles in the ear decomposition is at most m , so there are at most m calls to the polynomial time independence oracle obtained from A. All of the other steps are evidently polynomial, and so the algorithm is too. \square

Taken together, Propositions 3.9 (with δ set to $1/2$), 3.19, and 3.22, constitute a proof of our Theorem 1.2.

3.6. Reconstruct \mathbf{p}

Now armed with ℓ , G and $\sigma_{\mathbf{p}}$ (up to a global flip) we can reconstruct \mathbf{p} , up to translation and reflection.

In the noiseless setting, we can simply traverse a spanning tree of G and lay out each of the vertices in a greedy manner. This step completes the proof of Theorem 1.1.

In the setting with noise, we can set up a least-squares problem, finding the \mathbf{p} that minimizes the squared error in the length measurements, as follows. Suppose G and $\sigma_{\mathbf{p}}$ have been successfully reconstructed. Without loss of generality, assume that we have chosen the global sign of $\sigma_{\mathbf{p}}$ correctly as well. Then we can set up the linear system

$$M\mathbf{x} = \ell,$$

where M is the signed edge incidence matrix $M^{\sigma_{\mathbf{p}}}(G)$.

The matrix M has a kernel spanned by the $\mathbf{1}$ (all ones) vector, which we now want to mod out. To do this, let Q be an n -by- $(n-1)$ matrix whose columns form an orthonormal basis for $\mathbf{1}^\perp$. We now consider the linear system

$$MQ\mathbf{y} = \ell$$

To solve for the least squares solution, we set up the normal equations

$$Q^\top M^\top MQ\mathbf{y} = Q^\top M^\top \ell$$

with solution

$$\mathbf{y}^* = (Q^\top M^\top MQ)^{-1}(Q^\top M^\top \ell)$$

We then output the configuration $\mathbf{p}^* := Q\mathbf{y}^*$.

We now want to compare this output to $\mathbf{q} := QQ^\top \mathbf{p}$, which is the configuration \mathbf{p} translated to have its center of mass at the origin. Thus we want to bound

$$\|\mathbf{p}^* - \mathbf{q}\| = \|Q\mathbf{y}^* - QQ^\top \mathbf{p}\| = \|\mathbf{y}^* - Q^\top \mathbf{p}\|.$$

Let us define the correct lengths as $\ell' := M\mathbf{p} = (MQ)(Q^\top \mathbf{p})$ and the residual $\mathbf{r} := \ell - \ell'$. Then

$$\begin{aligned} \mathbf{y}^* &= (Q^\top M^\top MQ)^{-1}(Q^\top M^\top)[\ell' + \mathbf{r}] \\ &= Q^\top \mathbf{p} + (Q^\top M^\top MQ)^{-1}(Q^\top M^\top)\mathbf{r} \end{aligned}$$

and, hence,

$$\mathbf{y}^* - Q^\top \mathbf{p} = (Q^\top M^\top M Q)^{-1} (Q^\top M^\top) \mathbf{r}.$$

Under our noise assumption, the norm of \mathbf{r} is at most \sqrt{m} .

Recognizing $M^\top M$ as the unweighted Laplacian matrix of the graph G , we see its smallest non-zero eigenvalue is at least $\frac{1}{n^2}$ [26, Equation (6.10)] and the largest singular value of M is at most \sqrt{n} [26, Theorem 2.2(c)]. The only effect of Q on the spectral decomposition of $M^\top M$ is to remove the 0 eigenvalue. Likewise, its only effect on M is to remove the 0 singular value. Putting things together, we see that, up to translation and reflection on \mathbf{p} , the norm of $\mathbf{p}^* - \mathbf{q}$ is at most $n^{2.5} \sqrt{m}$. This completes the proof of Theorem 1.3.

3.7. Labeled Setting

In the labeled setting the algorithm is even easier. In particular, we can skip the step of reconstructing G from S^{σ_p} . (That is the only step that depends on 3-connectivity.)

Alternatively, since we have access to G , we can also apply a more expensive, but conceptually more simple algorithm. Instead of using LLL as in Section 3.2 to recover S^{σ_p} all at once, we can compute an ear decomposition of G (as described in Section 3.5), and then extract the orientation of the edges by solving a single SUBSET-SUM problem (using LLL) separately for each cycle. (These are not independent SUBSET-SUM problems, but there are at most n of them, so the failure probability is still exponentially small in m .) Then we continue with the computation as described in Section 3.5.

4. Detecting failures

Theorem 1.1 provides only a high probability guarantee of reconstructing G , in the unlabeled setting, and \mathbf{p} , in the labelled and unlabelled settings. We now discuss where the algorithm can fail, and some tests to detect failure.

The first observation is that the only step requiring a probabilistic analysis is the computation of S^{σ_p} using an LLL reduced basis for $\text{Lat}(\ell)$. After this, the rest of the steps are deterministic and, if S^{σ_p} is correctly computed, will compute G and then \mathbf{p} . Hence, every failure of the algorithm arises from failure of the LLL step to correctly compute S^{σ_p} . Moreover, since the truncations of the medium sized vectors in Λ span a subspace S of \mathbb{R}^m that contains S^{σ_p} , failure occurs iff $S \supsetneq S^{\sigma_p}$.

One example of how this can happen is when the pair (G, \mathbf{p}) is not “globally rigid”. This means there is another configuration \mathbf{q} so that $\ell = \ell(\mathbf{p}) = \ell(\mathbf{q})$, but \mathbf{q} is not related to \mathbf{p} by translation or reflection. Since we assume that G is 3-connected, there is some cycle in G that is oriented differently in σ_p and σ_q . Since ℓ satisfies the cycle relations from both G^{σ_p} and G^{σ_q} , the lattice reduction will find more than c medium-sized relations. We don’t know that failures of global rigidity are the only way that the lattice reduction step can fail.

4.1. Labeled setting

In labeled problems, the algorithm has access to n , and thus also c . Since the computation of S^{σ_p} succeeds if and only if Λ has exactly c medium sized vectors in it, the algorithm can stop and report failure if there are more than c medium sized vectors in Λ .

4.2. Unlabeled setting

In the unlabeled setting, the algorithm does not have access to c , and so there is a possibility that it reports an incorrect result. Here we describe some, to our knowledge non-exhaustive, tests that can detect a failure to correctly compute a basis for S^{σ_p} .

Seymour's algorithm may report that the matroid represented by the linear space S obtained in the lattice reduction step is not graphic. Since S^{σ_p} does represent a graphic matroid, whenever Seymour's algorithm fails, $S \neq S^{\sigma_p}$.

Even if an incorrectly computed S represents a graphic matroid, of a graph H necessarily not isomorphic to G , the orientation-finding step may fail. If H is not 2-connected, then the failure will occur when that step tries to find an ear-decomposition. If H is 2-connected, the algorithm may find an orientation H^τ of H that is not a DAG, and, hence, τ is not vertex consistent.

Finally, if the previous tests all pass, we can check that the output configuration \mathbf{q} has length measurements consistent with ℓ . If there is no noise, the check is exact: we can verify that each edge of the output graph G has, in $\ell(\mathbf{q})$, the same length as the corresponding entry of ℓ . If not, the algorithm has failed. When there is noise, the test is not exact, but a residual error in the measured lengths larger than that allowed by Theorem 1.3 indicates the algorithm has failed.

While we expect these tests are useful in practice, the possibility that all of them pass for an incorrectly computed S^{σ_p} remains a (low probability) possibility.

5. Denser graphs

The limiting factor of our algorithm is its need for highly accurate measurements. Here we describe an option for certain denser graphs, which can require many fewer bits of accuracy.

Definition 5.1. We say that G has a *k-basis*, for some k , and orientation σ , the \mathbb{R} -cycle space of G^σ has a cycle induced basis $\{\mathbf{w}_{\rho, \sigma}\}$, where each of the cycles ρ has at most k edges. ■

If we assume that G has a k -basis, then, given a noisy measurement $\ell = \ell(\mathbf{p}) + \varepsilon$, with $\varepsilon \in \{-1, 0, 1\}^m$ we can compute a basis for $\text{Lat}(\ell)$ by enumerating every vector

$$\hat{\mathbf{w}} = \begin{pmatrix} \mathbf{w} \\ e \end{pmatrix}$$

where \mathbf{w} has at most k non-zero entries, each of which is ± 1 and $-k \leq e \leq k$ and checking whether it is in $\text{Lat}(\ell)$ using the lattice generating matrix $L(\ell)$. Assuming that G has a k -basis, a maximal linearly-independent subset of the vectors of this form will span a subspace of $\text{Lat}(\ell)$ that contains \hat{S}^{σ_p} . A similar analysis to that in Section 3.2, shows that the span of the vectors returned by the brute-force procedure is, with high probability over a uniformly selected \mathbf{p} , a basis for \hat{S}^{σ_p} . Once we have recovered \hat{S}^{σ_p} , we can proceed as if we had used LLL.

The complexity of the brute-force approach to find a k -basis for S^σ is exponential in k , but polynomial in m . If k is fixed and independent of m , it is polynomial, albeit much slower than LLL for even modest k . The advantage of the brute-force procedure is that, because it solves the shortest-vector problem for $\text{Lat}(\ell)$ exactly, we can take b to be any quantity that is $\omega(m \log m)$, since we no longer have to account for the approximation factor in the LLL algorithm.

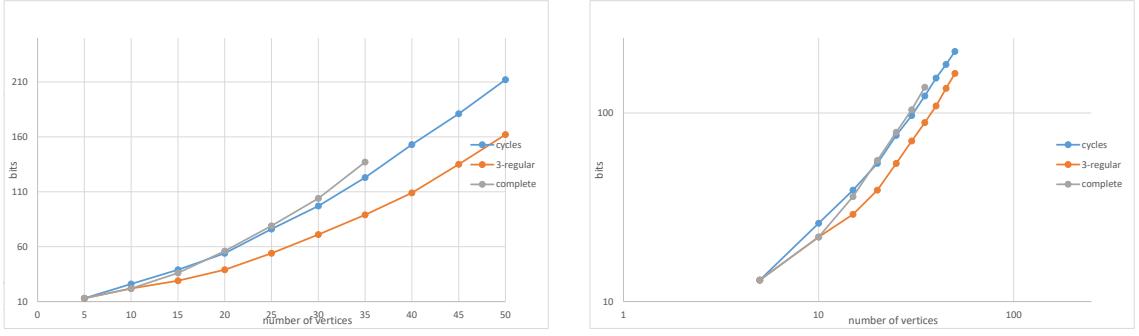


Figure 2: Number of bits required for the LLL step to recover the cycle space with probability 0.9 vs. number of vertices. Log-Log on right.

6. Experiments

We implemented the LLL step in our algorithm to explore how it performs on different graph families. We looked at three family of graphs: cycles; nearly 3-regular graphs (with all vertices degree 3 except for one which has degree 4 or 5); and complete graphs.

For a fixed number of vertices (and hence edges), we used our implementation to search for the number of bits necessary for the probability (over random point sets) of the LLL step correctly identifying the cycle space of the graph to be at least 0.9. (Here the configurations were integer, and the lengths were measured with $\{-1, 0, 1\}$ error.) For the nearly 3-regular graphs, we tried 5 different random ensembles and used the maximum number of bits required for each size. These experiments are “optimistic”, since they use knowledge of the number of vertices to avoid using a threshold value to identify small vectors in the reduced lattice basis.

In these results, we see that for all 3 families of graphs, the bit requirement grows at about the rate $n^{1.5}$ (see the log-log plot of Figure 2). More bits are needed for the cycle graphs than the nearly 3-regular graphs. This is perhaps due to the fact that the nearly 3-regular graphs have basis cycles, with each cycle much smaller than n . For the complete graphs, where $m = O(n^2)$, this gives us growth that is somewhat sub-linear in m . All of this behaviour is significantly better than the pessimistic m^2 bounds of our theory. On the other hand, even for moderate sized graphs, the required bit accuracy quickly becomes higher than we would be able to get from a physical measurement system.

The data sets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

References

- [1] A. Andoni, D. Hsu, K. Shi, and X. Sun. Correspondence retrieval. In *Conference on Learning Theory*, pages 105–126. PMLR, 2017.
- [2] S. J. Billinge, P. M. Duxbury, D. S. Gonçalves, C. Lavor, and A. Mucherino. Recent results on assigned and unassigned distance geometry with applications to protein molecules and nanostructures. *Annals of Operations Research*, 271:161–203, 2018.

- [3] R. E. Bixby and W. H. Cunningham. Converting linear programs to network problems. *Math. Oper. Res.*, 5(3):321–357, 1980. doi: 10.1287/moor.5.3.321.
- [4] M. Boutin and G. Kemper. On reconstructing n -point configurations from the distribution of distances or areas. *Adv. in Appl. Math.*, 32(4):709–735, 2004. doi: 10.1016/S0196-8858(03)00101-5.
- [5] M. Cieliebak, S. Eidenbenz, and P. Penna. Noisy data make the partial digest problem np-hard. In *International Workshop on Algorithms in Bioinformatics*, pages 111–123. Springer, 2003.
- [6] R. Connelly. On generic global rigidity. In *Applied geometry and discrete mathematics*, volume 4 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 147–155. Amer. Math. Soc., Providence, RI, 1991. doi: 10.1090/dimacs/004/11.
- [7] R. Connelly. Generic global rigidity. *Discrete Comput. Geom.*, 33(4):549–563, 2005. doi: 10.1007/s00454-004-1124-4.
- [8] G. M. Crippen and T. F. Havel. *Distance geometry and molecular conformation*, volume 15 of *Chemometrics Series*. Research Studies Press, Ltd., Chichester; John Wiley & Sons, Inc., New York, 1988.
- [9] J. Dattorro. Convex optimization and Euclidean distance geometry. Online book, 2015. URL https://ccrma.stanford.edu/~dattorro/0976401304_v2015.04.11.pdf.
- [10] I. Diakonikolas and D. Kane. Non-gaussian component analysis via lattice basis reduction. In *Conference on Learning Theory*, pages 4535–4547. PMLR, 2022.
- [11] R. Diestel. *Graph theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, Berlin, fifth edition, 2017. ISBN 978-3-662-53621-6. doi: 10.1007/978-3-662-53622-3.
- [12] P. M. Duxbury, L. Granlund, S. Gujarathi, P. Juhas, and S. J. Billinge. The unassigned distance geometry problem. *Discrete Applied Mathematics*, 204:117–132, 2016.
- [13] A. M. Frieze. On the Lagarias-Odlyzko algorithm for the subset sum problem. *SIAM J. Comput.*, 15(2):536–539, 1986. doi: 10.1137/0215038.
- [14] D. Gamarnik, E. C. Kızıldağ, and I. Zadik. Inference in high-dimensional linear regression via lattice basis reduction and integer relation detection. *IEEE Transactions on Information Theory*, 67(12):8109–8139, 2021.
- [15] I. Gkioulekas, S. J. Gortler, L. Theran, and T. Zickler. Trilateration using unlabeled path or loop lengths. *Discrete Comput. Geom.*, 71(2):399–441, 2024. doi: 10.1007/s00454-023-00605-x.
- [16] S. J. Gortler and Theran. Lattices without a big constant and with noise. *arXiv preprint arXiv:2204.12340*, 2022.
- [17] S. J. Gortler, A. D. Healy, and D. P. Thurston. Characterizing generic global rigidity. *Amer. J. Math.*, 132(4):897–939, 2010. doi: 10.1353/ajm.0.0132.

- [18] S. J. Gortler, L. Theran, and D. P. Thurston. Generic unlabeled global rigidity. In *Forum of Mathematics, Sigma*, volume 7. Cambridge University Press, 2019.
- [19] B. Hendrickson. Conditions for unique graph realizations. *SIAM J. Comput.*, 21(1):65–84, 1992. doi: 10.1137/0221008.
- [20] B. Jackson and T. Jordán. Connected rigidity matroids and unique realizations of graphs. *J. Combin. Theory Ser. B*, 94(1):1–29, 2005. doi: 10.1016/j.jctb.2004.11.002.
- [21] P. Juhás, D. Cherba, P. Duxbury, W. Punch, and S. Billinge. Ab initio determination of solid-state nanostructure. *Nature*, 440(7084):655–658, 2006.
- [22] J. C. Lagarias and A. M. Odlyzko. Solving low-density subset sum problems. *J. Assoc. Comput. Mach.*, 32(1):229–246, 1985. doi: 10.1145/2455.2461.
- [23] P. Lemke, S. S. Skiena, and W. D. Smith. Reconstructing sets from interpoint distances. In *Discrete and computational geometry*, pages 597–631. Springer, 2003.
- [24] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261(4):515–534, 1982. doi: 10.1007/BF01457454.
- [25] L. Lovasz. Computing ears and branchings in parallel. In *26th Annual Symposium on Foundations of Computer Science (sfcs 1985)*, pages 464–467, 1985. doi: 10.1109/SFCS.1985.16.
- [26] B. Mohar. The Laplacian spectrum of graphs. In Y. Alvi, G. Chartrand, O. R. Oellermann, and A. J. Schwenk, editors, *Graph Theory, Combinatorics, and Applications*, volume 2, pages 871–898. Wiley, 1991.
- [27] L. T. Nguyen, J. Kim, and B. Shim. Low-rank matrix completion: A contemporary survey. *IEEE Access*, 7:94215–94237, 2019. doi: 10.1109/ACCESS.2019.2928130.
- [28] P. Q. Nguyen and D. Stehlé. LLL on the average. In *International Algorithmic Number Theory Symposium*, pages 238–256. Springer, 2006.
- [29] J. Oxley. *Matroid theory*, volume 21 of *Oxford Graduate Texts in Mathematics*. Oxford University Press, Oxford, second edition, 2011. ISBN 978-0-19-960339-8. doi: 10.1093/acprof:oso/9780198566946.001.0001.
- [30] J. B. Saxe. Embeddability of weighted graphs in k -space is strongly NP-hard. In *Proceedings of the 17th Allerton Conference in Communications, Control and Computing*, pages 480–489, 1979.
- [31] P. D. Seymour. Recognizing graphic matroids. *Combinatorica*, 1(1):75–78, 1981. doi: 10.1007/BF02579179.
- [32] S. S. Skiena and G. Sundaram. A partial digest approach to restriction site mapping. *Bulletin of Mathematical Biology*, 56(2):275–294, 1994.
- [33] A. M.-C. So and Y. Ye. A semidefinite programming approach to tensegrity theory and realizability of graphs. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 766–775. Society for Industrial and Applied Mathematics, 2006.

- [34] M. J. Song, I. Zadik, and J. Bruna. On the cryptographic hardness of learning single periodic neurons. *Advances in neural information processing systems*, 34:29602–29615, 2021.
- [35] W. T. Tutte. An algorithm for determining whether a given binary matroid is graphic. *Proc. Amer. Math. Soc.*, 11:905–917, 1960. doi: 10.2307/2034435.
- [36] H. Whitney. 2-Isomorphic Graphs. *Amer. J. Math.*, 55(1-4):245–254, 1933. doi: 10.2307/2371127.
- [37] H. Whitney. On the Abstract Properties of Linear Dependence. *Amer. J. Math.*, 57(3):509–533, 1935. doi: 10.2307/2371182.
- [38] I. Zadik, M. J. Song, A. S. Wein, and J. Bruna. Lattice-based methods surpass sum-of-squares in clustering. In P-L. Loh and M. Raginsky, editors, *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pages 1247–1248. PMLR, 02–05 Jul 2022. Full version appears as arXiv:2112.03898.