Rigid components in fixed-lattice and cone frameworks*

Matthew Berardi† Brent Heeringa‡ Justin Malestein† Louis Theran†

Abstract

We study the fundamental algorithmic rigidity problems for generic frameworks periodic with respect to a fixed lattice or a finite-order rotation in the plane. For fixed-lattice frameworks we give an $O(n^2)$ algorithm for deciding generic rigidity and an $O(n^3)$ algorithm for computing rigid components. If the order of rotation is part of the input, we give an $O(n^4)$ algorithm for deciding rigidity; in the case where the rotation's order is 3, a more specialized algorithm solves all the fundamental algorithmic rigidity problems in $O(n^2)$ time.

1 Introduction

The geometric setting for this paper involves two variations on the well-studied *planar bar-joint* rigidity model: *fixed-lattice periodic frameworks* and *cone frameworks*. A *fixed-lattice periodic framework* is an infinite structure, periodic with respect to a lattice, where the allowed continuous motions preserve the lengths and connectivity of the bars as well as the periodicity with respect to a fixed lattice. See Figure 1(a). A *cone framework* is also made of fixed-length bars connected by universal joints, but it is finite and symmetric with respect to a finite order rotation; the allowed continuous motions preserve the bars' lengths and connectivity and symmetry with respect to a fixed rotation center. Cone frameworks get their name from the fact that the quotient of the plane by a finite order rotation is a flat cone with opening angle $2\pi/k$ and the quotient framework, embedded in the cone with geodesic "bars", captures all the geometric information [13]. Figure 2(a) shows an example.

A fixed-lattice framework is *rigid* if the only allowed motions are translations and *flexible* otherwise. A cone-framework is *rigid* if the only allowed motions are isometries of the cone, which is just rotation around the cone point, and *flexible* otherwise. A framework is *minimally rigid* if it is rigid, but ceases to be so if any of the bars are removed.

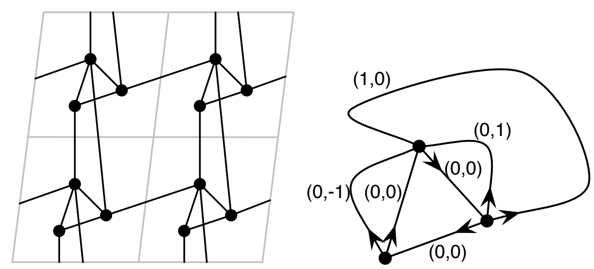
*Extended abstract. Full proofs are in [2].

†Department of Mathematics, Temple University, {mberardi,justmale,theran}@temple.edu

‡Department of Computer Science, Williams College, heeringa@cs.williams.edu

Generic rigidity The combinatorial model for the fixed-lattice and cone frameworks introduced above is given by a *colored graph* $(G, \boldsymbol{\gamma})$: $G = (V, E)$ is a finite directed graph and $\boldsymbol{\gamma} = (\gamma_{ij})_{ij \in E}$ is an assignment of a group element $\gamma_{ij} \in \Gamma$ (the "color") to each edge ij for a group Γ. For fixed-lattice frameworks, the group Γ is $\mathbb{Z}^2$, representing translations; for cone frameworks it is $\mathbb{Z}/k\mathbb{Z}$ with $k \geq 2$ a natural number. See Figure 1(b) and Figure 2(b).

The colors can be seen as efficiently encoding a map ρ from the oriented cycle space of G into Γ; ρ is defined, in detail, in Section 2. If the image of ρ restricted to a subgraph G' contains only the identity element, we define the Γ-*image* of ρ to be *trivial* otherwise it is *non-trivial*.



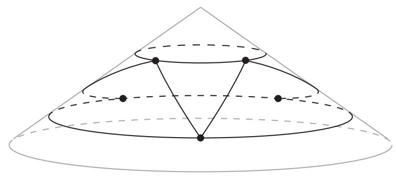
(a) (b)

Figure 1: Periodic frameworks and colored graphs: (a) part of a periodic framework, with the representation of the integer lattice $\mathbb{Z}^2$ shown in gray and the bars shown in black; (b) one possibility for the the associated colored graph with $\mathbb{Z}^2$ colors on the edges. (Graphics from [12].)

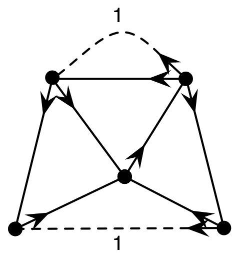
In 2009 Elissa Ross announced the following theorem:

Theorem 1 ([12],[15]) *A generic fixed-lattice periodic framework with associated colored graph $(G, \boldsymbol{\gamma})$ is minimally rigid if and only if: (1) G has n vertices and $2n - 2$ edges; (2) all non-empty subgraphs G' of G with m' edges and n' vertices and trivial $\mathbb{Z}^2$-image satisfy $m' \leq 2n' - 3$; (3) all non-empty subgraphs G' with non-trivial $\mathbb{Z}^2$-image satisfy $m' \leq 2n' - 2$.*

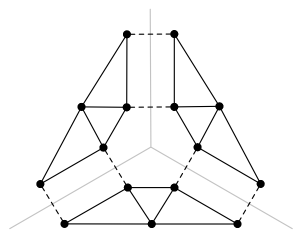
The colored graphs appearing in the statement of Theorem 1 are defined to be *Ross graphs*; if only conditions (2) and (3) are met, $(G, \boldsymbol{\gamma})$ is *Ross-sparse*. Ross graphs generalize the well-known *Laman graphs* which



(a)



(b)



(c)

Figure 2: Cone-Laman graphs: (a) a realization of the framework on a cone with opening angle $2\pi/3$ (graphic from Chris Thompson); (b) a $\mathbb{Z}/3\mathbb{Z}$-colored graph (edges without colors have color 0); (c) the developed graph with $\mathbb{Z}/3\mathbb{Z}$-symmetry (dashed edges are lifts of dashed edges in (b)).

are uncolored, have $m = 2n - 3$ edges, and satisfy (2). By Theorem 1 the maximal rigid sub-frameworks of a generic fixed-lattice framework on a Ross-sparse colored graph (G, γ) correspond to maximal subgraphs of G with $m' = 2n' - 2$; we define these to be the *rigid components* of (G, γ).

Malestein and Theran [13] proved a similar statement for cone frameworks:

Theorem 2 ([13]) *A generic cone framework with associated colored graph (G, γ) is minimally rigid if and only if: (1) G has n vertices and $2n - 1$ edges; (2) all non-empty subgraphs G' of G with m' edges and n' vertices and trivial $\mathbb{Z}/k\mathbb{Z}$-image satisfy $m' \leq 2n' - 3$; (3) all non-empty subgraphs G' with non-trivial $\mathbb{Z}/k\mathbb{Z}$-image satisfy $m' \leq 2n' - 1$.*

The graphs appearing in the statement of Theorem 2 are called *cone-Laman graphs*. We define *cone-Laman-sparse* colored graphs and their rigid components similarly to the analogous definitions for Ross-sparse graphs, with $2n' - 1$ replacing $2n' - 2$.

Ross and cone-Laman graphs are examples of the "Γ-graded-sparse" colored graphs introduced in [12, 13]. They are all matroidal families [12, 13], which guarantees that greedy algorithms work correctly on them.

Main results In this paper we investigate the algorithmic theory of crystallographic rigidity of fixed-lattice and cone frameworks. Given a colored graph (G, γ), we are interested in the rigidity properties of an associated generic framework. Lee and Streinu [9] define three fundamental algorithmic rigidity questions: **Decision** *Is the input rigid?*; **Extraction** *Find a maximum subgraph of the input corresponding to independent length constraints*; **Components** *Find the maximal rigid subframeworks of a flexible input.*

We give algorithms for these problems with running times shown in the following table

	Decision	Extraction	Components
Fixed-lattice	$O(n^2)$	$O(n^3)$	$O(n^3)$
Cone $k \neq 3$	$O(n^4)$	$O(n^5)$	$O(n^5)$
Cone $k = 3$	$O(n^2)$	$O(n^2)$	$O(n^2)$

Novelty Previously, the only known efficient combinatorial algorithms for any of these problems were pointed out in [12, 13]: the Edmonds Matroid Union algorithm yields an algorithm with running times $O(n^4)$ for **Decision** and $O(n^5)$ **Extraction**. Recently, Ross presented a **Decision** algorithm for Ross graphs very similar to ours [15]. A folklore randomized algorithm based on Gaussian elimination gives an $O(n^3 \operatorname{polylog}(n))$ algorithm for **Decision** and **Extraction** of most rigidity problems, but this doesn't easily generalize to **Components**.

The $O(n^2)$ running time of **Decision** for fixed-lattice frameworks equals that from the *pebble game* [3, 8, 9] for the corresponding problem in finite frameworks. Although there are faster **Decision** algorithms [5] for finite frameworks, the pebble game is the standard tool in the field due to its elegance and ease of implementation. Our algorithms for cone frameworks with order 3 rotation are a reduction to the pebble games of [3, 8, 9].

The $O(n^3)$ running time for **Extraction** and **Components** in fixed-lattice frameworks is worse by a factor of $O(n)$ than the pebble games for finite frameworks. However, it is equal to the $O(n^3)$ running time from [9] for the "redundant rigidity" problem. Computing *fundamental Laman circuits* (definition in Section 2) plays an important role (though for different reasons) in both of these algorithms.

Roadmap and key ideas Our main contribution is a pebble game algorithm for Ross graphs, from which we can deduce the corresponding results for general cone-Laman graphs. Intuitively, the algorithmic rigidity problems should be harder for Ross graphs than for Laman graphs, since the number of edges allowed in a subgraph depends on whether the $\mathbb{Z}^2$-image of the subgraph is trivial or not. To derive an efficient algorithm we use three key ideas (detailed definitions are given in Section 2):

- The Lee-Streinu-Theran [11] approach of playing several copies of the pebble game for (k, ℓ)-graphs

[9] with different parameters to handle different sparsity counts for different types of subgraphs.

- A *new* structural characterization of the edge-wise minimal colored graphs which violate the Ross counts (Section 3).
- A *linear time* algorithm for computing the Γ image of a given subgraph (Section 4).

Our algorithms for general cone-Laman graphs then use the Ross graph **Decision** algorithm as a subroutine. When the order of the rotation is 3, we can reduce the cone-Laman rigidity questions to Laman graph rigidity questions directly, resulting in better running times.

Motivation Periodic frameworks, in which the lattice *can* flex, arise in the study of *zeolites*, a class of microporous crystals with a wide variety of industrial applications, notably in petroleum refining. Because zeolites exhibit *flexibility* [16], computing the degrees of freedom in *potential* [14, 18] zeolite structures is a well-motivated algorithmic problem.

Other related work The general subject of periodic and crystallographic rigidity has seen a lot of progress recently[4, 12, 13], see [7] for a list of announcements. Bernd Schulze [17] has studied Laman graphs with a free $\mathbb{Z}/3\mathbb{Z}$ action in a different context and Elissa Ross's recent thesis studies rigidity of infinite periodic frameworks.[15].

2 Preliminaries

In this section, we introduce the required background in colored graphs, hereditary sparsity, and introduce a data structure for least common ancestor queries in trees that is an essential tool for us.

Colored graphs and the map ρ A pair $(G, \boldsymbol{\gamma})$ is defined to be a *colored graph* with Γ a group, $G = (V, E)$ a finite, directed graph on n vertices and m edges, and $\boldsymbol{\gamma} = (\gamma_{ij})_{ij \in E}$ is an assignment of a group element $\gamma \in \Gamma$ to each edge.

Let $(G, \boldsymbol{\gamma})$ be a colored graph, and let C be a cycle in G with a fixed traversal order. We define $\rho(C)$ to be

$$\rho(C) = \sum_{\substack{ij \in C \\ ij \text{ traversed} \\ \text{forwards}}} \gamma_{ij} - \sum_{\substack{ij \in C \\ ij \text{ traversed} \\ \text{backwards}}} \gamma_{ij}$$

Since Γ is always abelian in this paper, we need not be concerned with the particular order of summation, and since we are interested in whether $\rho(C)$ is trivial or not, we are not concerned with sign. For a subgraph G' of G, we define $\rho(G')$ to be *trivial* if its image on cycles spanned by G' contains only the identity and *non-trivial* otherwise. We need the following fact about ρ.

Lemma 3 ([12, Lemma 2.2]) *Let $(G, \boldsymbol{\gamma})$ be a colored graph. Then $\rho(G)$ is trivial if and only if, for* any *spanning forest T of G, ρ is trivial on every fundamental cycle induced by T.*

(k, ℓ)-sparsity and pebble games The hereditary sparsity counts defining Ross and cone-Laman graphs generalize to *(k, ℓ)-sparse graphs* which satisfy $m' \le kn' - \ell$ for all subgraphs; if in addition the total number of edges is $m = kn - \ell$, the graph is a *(k, ℓ)-graph*. We also need the notion of a *(k, ℓ)-circuit*, which is an edge-minimal graph that is not (k, ℓ)-sparse; these are always $(k, \ell - 1)$-graphs [9]. If G is any graph, a *(k, ℓ)-basis* of G is a maximal subgraph that is (k, ℓ)-sparse; if G' is a (k, ℓ)-basis of G and $ij \in E(G) - E(G')$, the *fundamental (k, ℓ)-circuit of ij with respect to G'* is the unique (k, ℓ)-circuit in $G' + ij$. See [9] for a detailed development of this theory. As is standard in the field, we use "$(2, 3)$-" and "Laman" interchangeably.

Although (k, ℓ)-sparsity is defined by exponentially many inequalities, it can be checked in quadratic time using the *pebble game* [9], an incremental approach that builds a (k, ℓ)-sparse graph G one edge at a time. Here, we will use the pebble game as a "black box" to: (1) Check if an edge ij is in the span of any (k, ℓ)-component of G in $O(1)$ time [9, 10]; (2) Assuming that G plus a new edge ij is (k, ℓ)-sparse, add the edge ij to G and update the components in amortized $O(n^2)$ time [9]; (3) Compute the fundamental circuit with respect to a given (k, ℓ)-sparse graph G in $O(n)$ time [9].

Least common ancestors in trees Let T be a rooted tree with root r and let i and j be any vertices in T. The *least common ancestor* (shortly, LCA) of i and j is defined to be the vertex where the (unique, since T is a tree) paths from i to r and j to r first converge. If either i or j is r, then this is just r. A fundamental result of Harel and Tarjan [6] is that LCA queries can be answered in $O(1)$ time after $O(n)$ preprocessing.

3 Combinatorial lemmas

In this section we prove structural properties of Ross and cone-Laman graphs that are required by our algorithms.

Ross graphs Let $(G, \boldsymbol{\gamma})$ be a colored graph and suppose that G is a $(2, 2)$-graph. We can verify that $(G, \boldsymbol{\gamma})$ is Ross by checking the $\mathbb{Z}^2$-images of a relatively small set of subgraphs.

Lemma 4 ([2]) *Let $(G, \boldsymbol{\gamma})$ be a colored graph and suppose that G is a $(2, 2)$-graph. Then $(G, \boldsymbol{\gamma})$ is a Ross* *graph if and only if for* any *Laman basis L of G, the fundamental Laman circuit with respect to L of every edge $ij \in E - E(L)$ has non-trivial $\mathbb{Z}^2$-image.*

Figure 3 shows two examples. The point is that we can pick *any* Laman basis L of G. The main idea is that G being a $(2,2)$-graph forces all Laman circuits to be edge-disjoint, from which we can deduce all of them are fundamental Laman circuits of every Laman basis.

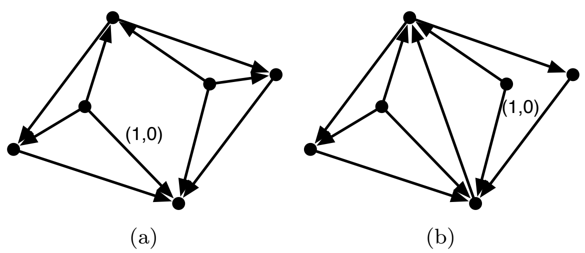


Figure 3: Examples of Ross and non-Ross graphs (edges without colors have color $(0,0)$): (a) a Ross graph; the underlying graph is itself a Laman circuit; (b) the underlying graph is a $(2,2)$-graph, but the uncolored K_4 subgraph has trivial image, so this is not a Ross graph. Note that K_4 is a Laman circuit, illustrating Lemma 4

Cone-Laman graphs Because cone-Laman graphs have an underlying $(2,1)$-graph, the statement of Lemma 4, with $(2,1)$- replacing $(2,2)$- does *not* hold for cone-Laman graphs. The analogous statement, proven in the full version is:

Lemma 5 ([2]) *Let (G,γ) be a colored graph. Then (G,γ) is a cone-Laman graph if and only if: (1) G is a $(2,1)$-graph; (2) for* any *$(2,2)$-basis R of G, the fundamental $(2,2)$-circuit G' with respect to R of $ij \in E(G) - E(R)$ becomes a Ross graph after removing* any *edge from G'; (3) for* any *Laman-basis L of G, the fundamental Laman-circuits with respect to L have non-trivial Γ-image.*

Order three rotations In the special case where the group $\Gamma = \mathbb{Z}/3\mathbb{Z}$, which corresponds to a cone with opening angle $2\pi/3$, we can give a simpler characterization of cone-Laman graphs in terms of their *development*. The development $\tilde{G}$ is defined by the following construction: $\tilde{G}$ has three copies of each vertex i: i_0, i_1 and i_2; a directed edge ij with color γ then generates three undirected edges $i_k j_{k+\gamma}$ (addition is modulo 3). See Figure 2(c)) for an example. The development has a free $\mathbb{Z}/3\mathbb{Z}$-action; a subgraph of $\tilde{G}$ is defined to be *symmetric* if it is fixed by this action.

Lemma 6 ([2]) *Let (G,γ) be a colored graph with $\Gamma = \mathbb{Z}/3\mathbb{Z}$. Then (G,γ) is a cone-Laman graph if and only if its development $\tilde{G}$ is a Laman graph. Moreover, the rigid components of (G,γ) correspond to the symmetric rigid components of $\tilde{G}$.*

4 Computing the Γ-image of ρ

We now focus on the problem of deciding whether the Γ-image of the map ρ, defined in Section 2, is trivial on a colored graph (G,γ). The case in which G is not connected follows easily by considering connected components one at a time, so we assume from now on that G is connected. Let (G,γ) be a colored graph and T be a spanning tree of G with root r. For a vertex i, there is a unique path P_i in T from r to i. We define σ_{ri} to be

$$\sigma_{ri} = \sum_{\substack{jk \in P_i \\ jk \text{ traversed forwards}}} \gamma_{jk} - \sum_{\substack{jk \in P_i \\ jk \text{ traversed backwards}}} \gamma_{jk}$$

The notation σ_{ri} extends in a natural way: for a a vertex j on P_i, we define σ_{ij} to be $\sigma_{ri} - \sigma_{rj}$; if σ_{ji} is defined, we define $\sigma_{ij} = -\sigma_{ji}$. The key observation is the following lemma:

Lemma 7 *Let (G,γ) be a connected colored graph, let T be a rooted spanning tree of G, let ij be an edge of G not in T, and let a be the least common ancestor of i and j. Then, if C is the fundamental cycle of ij with respect to T, $\rho(C) = \sigma_{ai} + \gamma_{ij} - \sigma_{ja}$.*

Proof. Traversing the fundamental cycle of ij so that ij is crossed from i to j means: going from i to j, from j to the LCA a of i and j towards the root, and then from a to i away from the root. □

We now show how to compute whether the Γ-image of a colored graph is trivial in linear time. The idea used here is closely related to a folklore $O(n^2)$ algorithm for all-pairs-shortest paths in trees.[1]

Lemma 8 *Let (G,γ) be a connected colored graph with n vertices and m edges. There is an $O(n+m)$ time algorithm to decide whether the Γ-image of $\rho(G)$ is trivial.*

The rest of this section gives the proof of Lemma 8. We first present the algorithm.
Input: A colored graph (G,γ)
Question: Is $\rho(G)$ trivial?
Method:

- Pick a spanning tree T of G and root it.
- Compute σ_{ri} for each vertex i of G.
- For each edge ij not in T, compute the image of its fundamental cycle in T.
- Say 'yes' if any of these images are not the identity and 'no' otherwise.

[1] We thank David Eppstein for clarifying the tree APSP trick's origins on MathOverflow.

Correctness This is an immediate consequence of Lemma 3, since the algorithm checks all the fundamental cycles with respect to a spanning tree.

Running time Finding the spanning tree with BFS is $O(m)$ time, and once the tree is computed, the σ_{ri} can be computed with a single pass over it in $O(n)$ time. Lemma 7 says that the image of any fundamental cycle with respect to T can be computed in $O(1)$ time once the LCA of the endpoints of the non-tree edge is known. Using the Harel-Tarjan data structure, the total cost of LCA queries is $O(n+m)$, and the running time follows.

The pebble game for Ross graphs We have all the pieces in place to describe our algorithm for the rigidity problems in Ross graphs.

Algorithm: Rigid components in Ross graphs
Input: A colored graph $(G, \boldsymbol{\gamma})$ with n vertices and m edges.
Output: The rigid components of $(G, \boldsymbol{\gamma})$.
Method: We will play the pebble game for $(2,3)$-sparse graphs and the pebble game for $(2,2)$-sparse graphs in parallel. To start, we initialize each of these separately, including data structures for maintaining the $(2,2)$- and $(2,3)$-components.

Then, for each colored edge $ij \in E$:

(A) If ij is in the span of a $(2,2)$-component in the $(2,2)$-sparse graph we are maintaining, we discard ij and proceed to the next edge.

(B) If ij is not in the span of any $(2,3)$-component, we add ij to both the $(2,2)$-sparse and $(2,3)$-sparse graphs we are building, and update the components of each.

(C) Otherwise, we use the $(2,3)$-pebble game to identify the smallest $(2,3)$-block G' spanning ij. We add ij to this subgraph G' and compute its $\mathbb{Z}^2$-image. If this is trivial, we discard ij and proceed to the next edge.

(D) If the image of G' was non-trivial, add ij to the $(2,2)$-sparse graph we are maintaining and update its rigid components.

The output is the (2,2)-components in the $(2,2)$-sparse graph we built.

Correctness By definition, the rigid components of a Ross graph are its $(2,2)$-components. Step **(A)** ensures that we maintain a $(2,2)$-sparse graph; steps **(B)** and **(C)**, by Lemma 4 imply that when new $(2,2)$-blocks are formed *all* of them have non-trivial $\mathbb{Z}^2$-image, which is what is required for Ross-sparsity. Step **(D)** ensures that the rigid components are updated at every step. The matroidal property implies that a greedy algorithm is correct.

Running time By [9, 10], steps **(A)**, **(B)**, and **(D)** require $O(n^2)$ time over the entire run of the algorithm (the analysis of the time taken to update components is amortized). Step **(C)**, by [9] and Lemma 7 requires $O(n)$ time. Since $\Omega(m)$ iterations may enter step **(C)**, this becomes the bottleneck, resulting in an $O(nm)$ running time, which is $O(n^3)$.

Modifications for other rigidity problems We have presented and analyzed an algorithm for computing the rigid components in Ross graphs. Minor modifications give solutions to the **Decision** and **Extraction** problems. For **Extraction**, we just return the $(2,2)$-sparse graph we built; the running time remains $O(n^3)$. For **Decision**, we simply stop and say 'no' if any edge is ever discarded. Since we process at most $O(n)$ edges, the running time becomes $O(n^2)$.

5 Pebble games for cone-Laman graphs

We now describe our algorithms for cone-Laman graphs.

Order-three rotations We start with the special case when the group $\Gamma = \mathbb{Z}/3\mathbb{Z}$. In this case, the following algorithm's correctness is immediate from Lemma 6. The running time follows from [3, 9, 10] and the fact that the development can be computed in linear time.

Input: A colored graph $(G, \boldsymbol{\gamma})$ with n vertices and m edges.
Output: The rigid components of $(G, \boldsymbol{\gamma})$.
Method:

(A) Compute the development $\tilde{G}$ of $(G, \boldsymbol{\gamma})$.

(B) Use the $(2,3)$-pebble game to compute the rigid components of $\tilde{G}$.

(C) Return the subgraphs of G corresponding to the symmetric rigid components in $\tilde{G}$.

General cone-Laman graphs For colored graphs with $\Gamma = \mathbb{Z}/k\mathbb{Z}$, we don't have an analogue of Lemma 6, and the development may not be polynomial size. However, we can modify our pebble game for Ross graphs to compute the rigid components. Here is the algorithm:
Input: A colored graph $(G, \boldsymbol{\gamma})$ with n vertices and m edges, and an integer k.
Output: The rigid components of $(G, \boldsymbol{\gamma})$.
Method: We initialize a $(2,1)$-pebble game, a $(2,2)$-pebble game, and a $(2,3)$-pebble game. Then, for each edge $ij \in E(G)$:

(A) If ij is in the span of a $(2,1)$-component in the $(2,1)$-sparse graph we are maintaining, we discard ij and proceed to the next edge.

(B) If ij is not in the span of any $(2,3)$-component, we add ij to all three sparse graphs we are building, update the components of each, and proceed to the next edge.

(C) If ij is not in the span of any $(2,2)$-component, we check that its fundamental Laman circuit in the $(2,3)$-sparse graph has non-trivial $Z/k\mathbb{Z}$-image. If not, discard ij. Otherwise, add ij to the $(2,1)$- and $(2,2)$-sparse graphs and update components.

(D) Otherwise ij is not in the span of any $(2,1)$-component. We find the minimal $(2,2)$-block G' spanning ij and check if $G' + ij$ becomes a Ross graph after removing *any* edge. If so, add ij to the $(2,1)$-graph we are building. Otherwise discard ij.

The output is the $(2,1)$-components in the $(2,1)$-sparse graph we built.

Analysis The proof of correctness follows from Lemma 5 and an argument similar to the one used to show that the pebble game for Ross graphs is correct. Each loop iteration takes $O(n^3)$ time, from which the claimed running times follow.

6 Conclusions and remarks

We studied the three main algorithmic rigidity questions for generic fixed-lattice periodic frameworks and cone frameworks. We gave algorithms based on the pebble game for each of them. Along the way we introduced several new ideas: a linear time algorithm for computing the Γ-image of a colored graph, a characterization of Ross graphs in terms of Laman circuits, and a characterization of cone-Laman graphs in terms of the development for $k = 3$ and Ross graphs for general k.

Implementation issues The pebble game has become *the* standard algorithm in the rigidity modeling community because of its elegance, ease of implementation, and reasonable implicit constants. The original data structure of Harel and Tarjan [6], unfortunately, is too complicated to be of much use except as a theoretical tool. More recent work of Bender and Farach-Colton [1] gives a vastly simpler data structure for $O(1)$-time LCA that is not much more complicated than the *union pair-find* data structure of [10] used in the pebble game. This means that the algorithm presented here is implementable as well.

References

[1] M. A. Bender and M. Farach-Colton. The LCA problem revisited. In *Proc. LATIN'00*, pages 88–94, 2000.

[2] M. Berardi, B. Heeringa, J. Malestein, and L. Theran. Rigid components in fixed-lattice and cone frameworks. Preprint, arXiv:1105.3234, 2011.

[3] A. R. Berg and T. Jordán. Algorithms for graph rigidity and scene analysis. In *ESA 2003*, volume 2832 of *LNCS*, pages 78–89. 2003.

[4] C. Borcea and I. Streinu. Periodic frameworks and flexibility. *Proc. of the Royal Soc. A*, 466:2633–2649, 2010.

[5] H. N. Gabow and H. H. Westermann. Forests, frames, and games: algorithms for matroid sums and applications. *Algorithmica*, 7(5-6):465–497, 1992.

[6] D. Harel and R. E. Tarjan. Fast algorithms for finding nearest common ancestors. *SIAM J. Comput.*, 13:338–355, May 1984.

[7] B. Jackson, J. Owen, and S. Power. London mathematical society workshop : Rigidity of frameworks and applications. `http://www.maths.lancs.ac.uk/~power/LancRigidFrameworks.htm`, 2010.

[8] D. J. Jacobs and B. Hendrickson. An algorithm for two-dimensional rigidity percolation: the pebble game. *J. Comput. Phys.*, 137(2):346–365, 1997.

[9] A. Lee and I. Streinu. Pebble game algorithms and sparse graphs. *Discrete Math.*, 308(8):1425–1437, 2008.

[10] A. Lee, I. Streinu, and L. Theran. Finding and maintaining rigid components. In *Proc. CCCG'05*, 2005.

[11] A. Lee, I. Streinu, and L. Theran. Graded sparse graphs and matroids. *Journal of Universal Computer Science*, 13(11):1671–1679, 2007.

[12] J. Malestein and L. Theran. Generic combinatorial rigidity of periodic frameworks. Preprint, arXiv:1008.1837, 2010.

[13] J. Malestein and L. Theran. Generic combinatorial rigidity of crystallographic frameworks. Preprint, 2011.

[14] I. Rivin. Geometric simulations - a lesson from virtual zeolites. *Nature Materials*, 5(12):931–932, Dec 2006.

[15] E. Ross. *The Rigidity of Periodic Frameworks as Graphs on a Torus*. PhD thesis, York University, May 2011.

[16] A. Sartbaeva, S. Wells, M. Treacy, and M. Thorpe. The flexibility window in zeolites. *Nature Materials*, Jan 2006.

[17] B. Schulze. Symmetric versions of Laman's theorem. *Discrete Comput. Geom.*, 44(4):946–972, 2010.

[18] M. Treacy, I. Rivin, E. Balkovsky, and K. Randall. Enumeration of periodic tetrahedral frameworks. II. Polynodal graphs. *Microporous and Mesoporous Materials*, 74:121–132, 2004.