Analyzing Rigidity with Pebble Games

Audrey Lee [1] Ileana Streinu [2] Louis Theran [1]

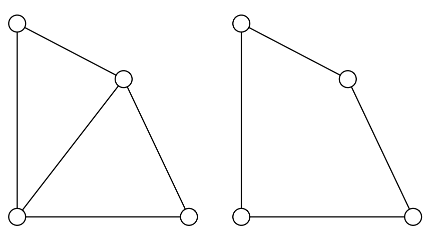
Abstract

How many pair-wise distances must be prescribed between an unknown set of points, and how should they be distributed, to determine only a discrete set of possible solutions? These questions, and related generalizations, are central in a variety of applications. **Combinatorial rigidity** shows that in two-dimensions one can get the answer, generically, via an efficiently testable *sparse graph property*.

We present a video and a web site illustrating algorithmic results for a variety of rigidity-related problems, as well as abstract generalizations. Our accompanying interactive software is based on a comprehensive implementation of the **pebble game** paradigm.

1 Rigidity Models

We start with a brief overview of the models of rigidity relevant for the presented work. For each model, we describe the **geometric constraints** that define it and the **rigidity property** we want to determine.



(a) (b)

Figure 1: Bar-and-joint structures in two-dimensions: (a) minimally rigid and (b) flexible.

Planar bar-and-joint rigidity. A **bar-and-joint framework** consists of fixed-length **bars** connected by **universal joints** allowing full rotation of the incident bars. If the only motions maintaining the lengths of all the bars are **trivial rigid motions** (translations and rotations), the framework is **rigid** (see Figure 1(a)); otherwise, it is **flexible** (see Figure 1(b) for an example in two-dimensions).

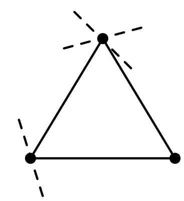


Figure 2: Minimally pinned bar-slider framework.

Planar slider pinning. A bar-and-joint framework in the plane may be additionally constrained by **sliders**, which force joints to move on fixed lines; we call such structures **bar-slider** frameworks. Allowed motions preserve the lengths of all the bars and move vertices constrained by sliders along the specified lines. In this model, we are concerned with **pinning rigidity**: a bar-slider framework is **pinned** if it is completely immobilized, i.e. rigidly attached to the plane. See Figure 2.

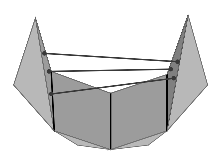


Figure 3: Minimally rigid body-hinge-and-bar structure.

Body-bar and body-hinge rigidity. A **body-bar framework** consists of **rigid bodies** connected by fixed length **bars**; the bars are rigidly attached to the bodies with universal joints. In addition to bars, **hinges** may also constrain the structure (see Figure 3 for a 3D body-hinge-and-bar structure). They constrain the incident bodies to a relative rotation about the hinge axis. A body-and-bar framework is rigid if the only motions of the framework are trivial ones, as in the bar-and-joint case; otherwise, the framework is flexible.

Parallel redrawings. Given a graph embedded in the plane, a **parallel redrawing** maintains the **direction** of each edge. If every parallel redrawing of the graph is

[1] Computer Science Department, University of Massachusetts Amherst, USA, {alee,theran}@cs.umass.edu

[2] Department of Computer Science, Smith College, Northampton, MA 01063, USA, istreinu@smith.edu.

similar to the original, the graph is said to be **direction rigid**; otherwise, it is **flexible**.

2 Rigidity analysis problems

For each model of rigidity, we are interested in the following problems; with respect to terminology, *framework, rigid* and *flexible* are to be taken in the context of a given model.

Decision: Is a framework rigid?
Extraction: Find a maximum-cardinality set of independent constraints in a framework. For a rigid input, this is a minimally rigid substructure (one which becomes flexible after removing any constraint).
Components: Detect the maximal rigid substructures (called *components*) of a flexible framework.
Optimization: Extract a set of independent constraints optimizing a given linear weight function.
Extension: Given a flexible framework, describe a set of constraints whose addition would rigidify it. (For example, give a set of sliders that pin a bar-and-joint framework.)

Generic rigidity theorems. The *generic* rigidity of these models is captured by **combinatorial** properties. Laman's theorem [2] characterizes minimally rigid 2D bar-and-joint frameworks, and Tay's theorem [8] characterizes minimally rigid body-and-bar frameworks in arbitrary dimension d ([9] and [10] observe that hinges may be represented by 5 bars). For results relating to parallel redrawings, see [10]. In [7], we present results for bar-slider pinning rigidity.

3 Pebble games and sparsity

The combinatorial rigidity characterizations are based on hereditary counts on the number of edges in subgraphs. These counts have been generalized to **sparse graphs** and **hypergraphs** [3, 5, 6, 10]. A (hyper)graph G is (k, ℓ)**-sparse** if any set of n' vertices spans at most $kn' - \ell$ edges in G for non-negative integer parameters k and ℓ which allow for non-triavial graphs (e.g. for graphs, they must satisfy $\ell \in [0, 2k)$). This definition generalizes to other situations, including **graded sparsity** [5], in which the edges of G are partitioned into classes, each of which satisfies its own sparsity counts.

Our pebble games, which extend the elegant algorithm of Jacobs and Hendrickson for 2D bar-and-joint rigidity [1], are a family of graph construction rules indexed by non-negative integers k and ℓ. The (k, ℓ)-pebble game "recognize" exactly the (k, ℓ)-sparse graphs [3], and variations on them address all the algorithmic questions listed above. In particular, $(2, 3)$-pebble games solve planar rigidity and parallel redrawing questions, $k = \ell = \binom{d+1}{2}$ handle body-bar-hinge structures in dimension d, and a combination of $(2, 3)$ and $(2, 0)$-pebble games solve pinning rigidity in the plane.

The basic (k, ℓ)-pebble game is played by a single player on a directed graph. It is described in terms of an initial configuration (an empty graph on n vertices, with k pebbles on each) and two allowed moves:

Add edge move: If vertices i and j have at least $\ell + 1$ pebbles altogether on them, add the directed edge ij and pick up a pebble from one of the endpoints.
Pebble slide move: If ij is an edge and there is a pebble on j, reverse the edge ij and move the pebble from j to i.

We have developed several types (including **component**, **colored** and **graded** variations) of **pebble games** [3, 4, 5, 6] for *all* sparse graphs and hypergraphs.

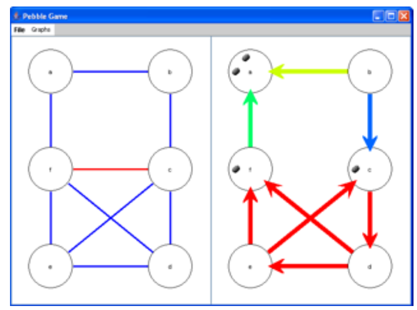


Figure 4: A snapshot of the component pebble game for 2D bar-and-joint rigidity.

Demo site. Applets, demos and an accompanying video can be found at http://linkage.cs.umass.edu/pg. The video may be independently downloaded from http://linkage.cs.umass.edu/pg/socgSubmission.html.

References

[1] D. J. Jacobs and B. Hendrickson. An algorithm for two-dimensional rigidity percolation: the pebble game. *Jour. Comput. Physics*, 137:346–365, November 1997.

[2] G. Laman. On graphs and rigidity of plane skeletal structures. *Journal of Engineering Mathematics*, 4:331–340, 1970.

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

[4] A. Lee, I. Streinu, and L. Theran. Finding and maintaining rigid components. In *Proc. Canadian Conf. Comput. Geometry*, 2005.

[5] A. Lee, I. Streinu, and L. Theran. Graded sparse graphs and matroids. *Jour. Univ. Comp. Science*, 13(10), 2007.

[6] I. Streinu and L. Theran. Sparse hypergraphs and pebble game algorithms. *European Journal of Combinatorics*, To appear.

[7] I. Streinu and L. Theran. Combinatorial genericity and minimal rigidity. In *Proc. 24th Symp. Comput. Geometry (SoCG)*, 2008.

[8] T.-S. Tay. Rigidity of multigraphs I: linking rigid bodies in n-space. *Journal of Combinatorial Theory Series*, B 26:95–112, 1984.

[9] T.-S. Tay. Linking (n - 2)-dimensional panels in n-space II: (n - 2, 2)-frameworks and body and hinge structures. *Graphs and Combinatorics*, 5:245–273, 1989.

[10] W. Whiteley. Some matroids from discrete applied geometry. In J. Bonin, J. G. Oxley, and B. Servatius, editors, *Matroid The-*

ory, volume 197 of *Contemporary Mathematics*, pages 171–311. American Mathematical Society, 1996.