2007

Graded Sparse Graphs and Matroids

Audrey Lee
University of Massachusetts Amherst

Ileana Streinu
Smith College, streinu@cs.smith.edu

Louis Theran
University of Massachusetts Amherst

Journal of Universal Computer Science, vol. 13, no. 11 (2007), 1671-1679
submitted: 10/9/06, accepted: 19/11/07, appeared: 28/11/07 © J.UCS

Graded Sparse Graphs and Matroids[1]

Audrey Lee
(University of Massachusetts, Amherst, MA, USA
alee@cs.umass.edu)

Ileana Streinu
(Smith College, Northampton, MA, USA
streinu@cs.smith.edu)

Louis Theran
(University of Massachusetts, Amherst, MA, USA
theran@cs.umass.edu)

Abstract: Sparse graphs and their associated matroids play an important role in rigidity theory, where they capture the combinatorics of some families of generic minimally rigid structures. We define a new family called **graded sparse graphs**, arising from generically pinned bar-and-joint frameworks, and prove that they also form matroids. We also address several algorithmic problems on graded sparse graphs: **Decision**, **Spanning**, **Extraction**, **Components**, **Optimization**, and **Extension**. We sketch variations on **pebble game algorithms** to solve them.

Key Words: computational geometry, hypergraph, rigidity theory, matroid, pebble game

Category: F.2.2, G.2.2

1 Introduction

A **bar-and-joint framework** is a planar structure made of fixed-length **bars** connected by **universal joints**. Its allowed **motions** are those that preserve the lengths and connectivity of the bars. If the allowed motions are all **trivial rigid motions**, then the framework is **rigid**; otherwise it is flexible.

Laman's foundational theorem [Laman 1970] characterizes generic minimally rigid bar-and-joint frameworks in terms of their underlying graph. A Laman graph has $2n-3$ edges and the additional property that every induced subgraph on n' vertices spans at most $2n'-3$ edges. Laman's theorem characterizes the graphs of generic minimally rigid frameworks as Laman graphs.

Laman's hereditary counts have been recently generalized [Whiteley 1996, Lee and Streinu 2007, Streinu and Theran 2007] to (k, ℓ)*-sparse graphs* and *hypergraphs*, which form the independent sets of a matroid called the (k, ℓ)**-sparsity matroid**.

[1] C. S. Calude, G. Stefanescu, and M. Zimand (eds.). *Combinatorics and Related Areas. A Collection of Papers in Honour of the 65th Birthday of Ioan Tomescu.*

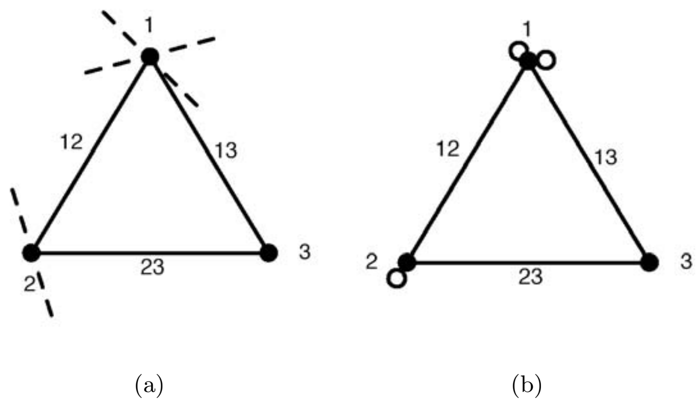

Figure 1: Example of a bar-slider framework and its associated graph: (a) a bar-slider framework; (b) the same framework given combinatorially as a graph with edges and loops

In [Streinu and Theran 2007b, Lee et al. 2007] we considered the problem of **pinning** a bar-and-joint framework by adding **sliders**. Pinning means completely immobilizing the structure by eliminating all the degrees of freedom, including the trivial rigid motions (rotations and translations). We do this by constraining vertices to move along generic lines, much like a slider joint in mechanics. A **slider** at vertex i is a line L_i associated with the vertex. A structure made from bars, joints, and sliders is called a **bar-slider** framework.

We model a bar-slider framework combinatorially with a graph that has vertices for the joints, with **edges** (2 endpoints) for the bars and **loops** (1 endpoint) for the sliders. Figure 1 shows an example of a bar-slider framework and its associated graph; the sliders are shown as dotted lines running through a vertex. The main rigidity result of [Streinu and Theran 2007b] is a Laman-type theorem.

Proposition 1 (Bar-slider framework rigidity). *Let G be a graph with $2n-k$ edges and k loops. G is realizable as a generic minimally pinned bar-and-slider framework if and only if: (1) Every subset of n' vertices spans at most $2n'-3$ edges (not counting loops), and (2) Every induced subgraph on n' vertices spans at most $2n'$ edges and loops.*

The generalization of the Laman counts from Proposition 1 leads to a **pinning matroid**, which has as its bases the graphs of minimally pinned generic bar-slider frameworks.

Contributions.

In this paper, we generalize the counts from Proposition 1 to what we call **graded sparsity** on hypergraphs. Graded sparsity has the same relationship to bar-slider structures as sparsity has to bar-and-joint frameworks. Complete definitions will be given in Section 3. We also briefly indicate the algorithmic solutions to the following fundamental problems (first posed in the context of pinning in [Streinu and Theran 2007b]), generalized to graded sparsity.

Decision problem: Is G a graded sparse graph?

Spanning problem: Does G contain a spanning graded sparse graph?

Extraction problem: Find a maximum sized graded sparse subgraph of G.

Optimization problem: Compute an optimal graded sparse subgraph of G with respect to an arbitrary linear weight function on the edges.

Extension problem: Find a minimum size set of edges to add to G, so that it becomes spanning.

Components problem: Find the *components* (which generalize rigid components to graded sparsity) of G.

For these problems, we give efficient, easily implementable algorithms based on pebble games for general sparse graphs (see the papers [Lee and Streinu 2007], [Streinu and Theran 2007], [Lee et al. 2005]).

2 Preliminaries

In this section, we give the necessary background (from previous work) to understand our contributions. We start with sparse graphs and hypergraphs.

2.1 Sparse graphs and hypergraphs.

A **hypergraph** $G = (V, E)$ is a finite set V of n vertices with a set E of m edges that are subsets of V. We allow multiple distinguished copies of edges; i.e., our hypergraphs are multigraphs. The **dimension** of an edge is the number of vertices in it; we call an edge of dimension d a d-edge. We call the vertices in an edge its **endpoints**. The concept of directed graphs extends to hypergraphs. In a directed hypergraph, each edge is given an **orientation** "away" from a distinguished endpoint, which we call its **tail**.

A hypergraph is (k, ℓ)**-sparse** if every edge-induced subgraph with m' edges spanning n' vertices satisfies $m' \leq kn' - \ell$; a hypergraph that is (k, ℓ)-sparse (shortly, sparse) and has $kn - \ell$ edges is called (k, ℓ)**-tight** (shortly, tight). Maximal tight subgraphs of a sparse hypergraph are called **components**.

Sparse hypergraphs have a matroidal structure, first observed by White and Whiteley in the appendix of [Whiteley 1996]. More specifically:

Proposition 2 ([Streinu and Theran 2007]). *Let G be a hypergraph on n vertices. For large enough n, the (k, ℓ)-sparse hypergraphs form the independent sets of a matroid that has tight hypergraphs as its bases.*

When $\ell \geq dk$, all the edges in a sparse hypergraph have dimension at least d, because otherwise the small edges would violate sparsity and the matroid would be trivial. The (k, ℓ)-sparsity matroid (for $0 \leq \ell < dk$) is defined on the ground set K_n^+, the complete hypergraph on n vertices, where edges of dimension d have multiplicity dk.

2.2 Pebble games

Pebble games are a family of simple construction rules for sparse hypergraphs. For history and references, see [Lee and Streinu 2007, Streinu and Theran 2007]. In a nutshell, the pebble game starts with an empty set of vertices with k pebbles on each vertex and proceeds through a sequence of moves. Each move either adds a directed edge or reorients one that is already there, using the location of the pebbles on the graph to determine the allowed moves at each step.

Pebble games are indexed by non-negative integer parameters k and ℓ. Initially, every vertex starts with k pebbles on it. An edge may be added if at least $\ell + 1$ pebbles are present on its endpoints, otherwise it is rejected. When an edge is added, one of the pebbles is picked up from an endpoint and used to "cover" the new edge, which is then directed away from that endpoint. Pebbles may be moved by reorienting edges. If an endpoint of an edge, other than its tail, has at least one pebble, this pebble may be used to cover the edge. The edge is subsequently reoriented away from that endpoint, and the pebble previously covering the edge is returned to the original tail.

The pebble game is used as a basis for algorithms that solve the fundamental sparse graph problems in [Lee and Streinu 2007, Streinu and Theran 2007]. The next proposition captures the results needed later.

Proposition 3 ([Streinu and Theran 2007]). Pebble games for sparse hypergraphs: *Using the pebble game paradigm, the* **Decision** *problem for sparse hypergraphs with edges of dimension d can be solved in $O(dn^2)$ time and $O(n)$ space. The* **Spanning**, **Extraction** *and* **Components** *problems for hypergraphs with m edges of dimension d can be solved in $O(n^d)$ time and space or $O(nmd)$ time and $O(m)$ space.* **Optimization** *can be solved in either of these running times plus an additional $O(m \log m)$.*

Not that in a hypergraph with edges of dimension d, m may be $\Theta(n^d)$.

2.3 Related work

Because of their relevance to rigidity, Laman graphs, and the related families of sparse graphs, have been extensively studied. Classic papers include [Whiteley 1988, Whiteley 1996], where extensions to sparsity matroids on graphs and hypergraphs first appeared. A more detailed history and comprehensive developments, as well as recent developments connecting sparse graphs and pebble game algorithms appear in [Lee and Streinu 2007, Streinu and Theran 2007]. The graded matroids of this paper are a further generalization.

Another direction is encountered in [Servatius and Whiteley 1999], which consider length and direction constraints (modeled as edges of different colors). The associated rigidity counts require $(2, 3)$-sparsity for monochromatic subgraphs and $(2, 2)$-sparsity for bichromatic subgraphs. This extends sparsity in a slightly different direction than we do here, and is neither a specialization nor a generalization of our graded sparsity.

Our algorithms fall into the family of generalized pebble games for sparse hypergraphs [Lee and Streinu 2007, Streinu and Theran 2007, Lee et al. 2005]. They are generalizations of an [Jacobs and Hendrickson 1997]'s algorithm for Laman graphs, formally analyzed in [Berg and Jordán 2003].

3 Graded sparsity

In this section, we define the concept of graded sparsity and prove the main result.

Definitions

Let $G = (V, E)$ be a hypergraph. A **grading** $(E_1, E_2, \ldots, E_s)$ of E is a strictly decreasing sequence of sets of edges $E = E_1 \supsetneq E_2 \supsetneq \cdots \supsetneq E_s$. An example is the **standard grading**, where we fix the E_i's to be edges of dimension at least i. This is the situation for the pinned Laman graphs of [Streinu and Theran 2007b], where the grading consists of ≥ 1-edges and 2-edges.

Fix a grading on the edges of G. Define $G_{\geq i}$ as the subgraph of G induced by $\cup_{j \geq i} E_i$. Let $\boldsymbol{\ell} = \{\ell_1 < \ell_2 < \cdots < \ell_s\}$ be a vector of s non-negative integers. We say that G is $(k, \boldsymbol{\ell})$**-graded sparse** if $G_{\geq i}$ is (k, ℓ_i)-sparse for every i; G is $(k, \boldsymbol{\ell})$**-graded tight** if, in addition, it is (k, ℓ_1)-tight. To differentiate this concept from the sparsity of Proposition 2, we refer to $(k, \boldsymbol{\ell})$-graded sparsity as **graded sparsity**. The **components** of a graded sparse graph G are the (k, ℓ_1)-components of G.

Main result

It can be easily shown that the family of $(k, \boldsymbol{\ell})$-graded sparse graphs is the intersection of s matroidal families of graphs. The main result of this paper is the following stronger property.

Theorem 4 (Graded sparsity matroids).
The $(k, \boldsymbol{\ell})$-graded sparse hypergraphs form the independent sets of a matroid. For large enough n, the $(k, \boldsymbol{\ell})$-graded tight hypergraphs are its bases.

The proof of Theorem 4 is based on the circuit axioms for matroids. See [Oxley 1992] for an introduction to matroid theory. We start by formulating $(k, \boldsymbol{\ell})$-graded sparsity in terms of circuits.

For a (k, ℓ)-sparsity matroid, the (k, ℓ)-circuits are exactly the graphs on n' vertices with $kn' - \ell + 1$ edges such that every proper subgraph is (k, ℓ)-sparse.

We now recursively define a family $\mathcal{C}$ as follows: $\mathcal{C}_{\geq s}$ is the set of $(k, \boldsymbol{\ell}_s)$-circuits in G_s; for $i < s$, $\mathcal{C}_{\geq i}$ is the union of $\mathcal{C}_{\geq i+1}$ and the $(k, \boldsymbol{\ell}_i)$-circuits of $G_{\geq i}$ that do not contain any of the elements of $\mathcal{C}_{\geq i+1}$. Finally, set $\mathcal{C} = \mathcal{C}_{\geq 1}$.

Example: As an example of the construction of $\mathcal{C}$, we consider the case of $k = 1$, $\boldsymbol{\ell} = (0, 1)$ with the standard grading. $\mathcal{C}_{\geq 2}$ consists of the $(1, 1)$-circuits of edges; a fundamental result of graph theory [Nash-Williams 1961, Tutte 1961] says that these are the simple cycles of edges. Using the identification of $(1, 0)$-tight graphs with graphs having exactly one cycle per connected component (see [Haas et al. 2007] for details and references), we infer that the $(1, 0)$-circuits are pairs of cycles sharing edges or connected by a path. Since cycles involving edges are already in $\mathcal{C}_{\geq 2}$, $\mathcal{C}_{\geq 1}$ adds only pairs of loops connected by a simple path.

We now prove a structural property of $\mathcal{C}$ that relates $\mathcal{C}$ to $(k, \boldsymbol{\ell})$-graded sparsity and will be used in the proof of Theorem 4.

Lemma 5. *Let d, k, and ℓ_i be such that $(d - 1)k \leq \ell_i < dk$. Then, every set in $\mathcal{C}_{\geq i}$ is either a single edge or has only edges of dimension at least d.*

Proof. A structure theorem from [Streinu and Theran 2007] says that for k and ℓ_i satisfying the condition in the lemma, all sparse graphs have only edges of dimension at least d or are empty. Since any proper subgraph of a (k, a)-circuit for $a \geq \ell_i$ is (k, ℓ_i)-sparse, either the circuit has only edges of dimension at least d or only empty proper subgraphs, i.e. it has exactly one edge.

Lemma 6. *A hypergraph G is $(k, \boldsymbol{\ell})$-graded sparse if and only if it does not contain a subgraph in $\mathcal{C}$.*

Proof. It is clear that all the $(k, \boldsymbol{\ell})$-graded sparse hypergraphs avoid the subgraphs in $\mathcal{C}$, since they cannot contain *any* (k, ℓ_i)-circuit of $G_{\geq i}$.

For the reverse inclusion, suppose that G is not sparse. Then for some i, $G_{\geq i}$ is not (k, ℓ_i)-sparse. This is equivalent to saying that $G_{\geq i}$ contains some (k, ℓ_i)-circuit C. There are now two cases: if $C \in \mathcal{C}$ we are done; if not, then some $C' \subsetneq C$ is in $\mathcal{C}$, and G contains C', which completes the proof.

We are now ready to prove Theorem 4.

Proof of Theorem 4. Lemma 6 says that it is sufficient to verify that $\mathcal{C}$ obeys the circuit axioms. By construction, $\mathcal{C}$ does not contain the empty set and no sets of $\mathcal{C}$ contain each other.

What is left to prove is that, for $C_i, C_j \in \mathcal{C}$ with $y \in C_i \cap C_j$, $(C_i \cup C_j) - y$ contains an element of $\mathcal{C}$. Suppose that C_i is a (k, ℓ_i)-circuit and C_j is a (k, ℓ_j)-circuit with $j \geq i$. Let $m_i, m_j, m_\cup$ and $m_\cap$ be the number of edges in C_i, C_j, $C_i \cup C_j$, and $C_i \cap C_j$, respectively. Similarly define $n_i, n_j, n_\cup$, and $n_\cap$ for the size of the vertex sets.

Lemma 5 implies that y has dimension d, where $\ell_j < dk$; otherwise C_j would have to be the single edge y, and this would block C_i's inclusion in $\mathcal{C}$ (since $j \geq i$). Because $C_i \cap C_j \subsetneq C_j$, we have $n_\cap \geq d$ and $m_\cap \leq kn_\cap - \boldsymbol{\ell}_j$. By counting edges, we have

$$\begin{aligned} m_\cup &= m_i + m_j - m_\cap \geq m_i + m_j - (kn_\cap - \ell_j) \\ &= kn_i - \ell_i + 1 + kn_j - \ell_j + 1 - (kn_\cap - \ell_j) \\ &= kn_\cup - \ell_i + 2. \end{aligned}$$

It follows that $C_i \cup C_j$ cannot be (k, ℓ_i)-sparse, and by Lemma 6, this is equivalent to having an element of $\mathcal{C}$ as a subgraph.

4 Algorithms

In this section, we start with the algorithm for **Extraction** and then derive algorithms for the other problems from it.

Algorithm 1. Extraction of a graded sparse graph
Input: *A hypergraph G, with a grading on the edges.*
Output: *A maximum size $(k, \boldsymbol{\ell})$-graded sparse subgraph of G.*
Method: *Initialize the pebble game with k pebbles on every vertex.*
Iterate over the edges of G in an arbitrary order. For each edge e:

1. *Try to collect at least $\ell_1 + 1$ pebbles on the endpoints of e. If this is not possible, reject it.*

2. *If $e \in E_1$, accept it, using the rules of the (k, ℓ_1)-pebble game.*

3. *Otherwise, copy the configuration of the pebble game into a "shadow graph". Set* $d \leftarrow 1$*.*

4. *In the shadow, remove every edge of* E_d*, and put a pebble on the tail of the removed edges.*

5. *Try to collect* $\ell_{d+1} - \ell_d$ *more pebbles on the endpoints of* e*. There are three possibilities: (1) if the pebbles cannot be collected, reject* e *and discard the shadow; (2) otherwise there are* $\geq \ell_{d+1} + 1$ *pebbles on the endpoints of* e*, if* $e \in E_{d+1}$ *discard the shadow and accept* e *using the rules of the* (k, ℓ_{d+1}) *pebble game; (3) otherwise, if* $e \notin E_{d+1}$*, set* $d \leftarrow d+1$ *and go to step 4.*

Finally, output all the accepted edges.

Correctness: By Theorem 4, adding edges in any order leads to a sparse graph of maximum size. What is left to check is that the edges accepted are exactly the independent ones.

This follows from the fact that moving pebbles in the pebble game is a reversible process, except when a pebble is moved and then used to cover a new edge. Since the pebbles covering hyper-edges in E_j, for $j < i$, would be on the vertices where they are located in the $(k, \boldsymbol{\ell}_i)$-pebble game, for $j < i$, then Algorithm 1 accepts edges in E_i exactly when the $(k, \boldsymbol{\ell}_i)$-pebble game would. By results of [Lee and Streinu 2007, Streinu and Theran 2007], we conclude that Algorithm 1 computes a maximum size $(k, \boldsymbol{\ell})$-graded sparse subgraph of G.

Running time: If we maintain components, the running time is $O(n^{d^\star})$, where $d^\star$ is the dimension of the largest hyperedge in the input. Without maintaining components, the running time is $O(mnd)$, with the caveat that m can be $\Theta(n^d)$.

Application to the fundamental problems.

We can use Algorithm 1 to solve the remaining fundamental problems. For **Decison** a simplification yields a running time of $O(dn^2)$: checking for the correct number of edges is an $O(n)$ step, and after that $O(mnd)$ becomes $O(n^2 d)$.

For **Optimization**, we consider the edges in an order sorted by weight. The correctness of this approach follows from the characterization of matroids by greedy algorithms. Because of the sorting phase, the running time is $O(n^d + m \log m)$.

The components are the $(k, \boldsymbol{\ell}_1)$-components of the output. Since we maintain these anyway, the running time for **Components** is $O(n^d)$.

Finally, for **Extension**, the matroidal property of $(k, \boldsymbol{\ell})$-graded sparse graphs implies that it can be solved by using the **Extraction** algorithm on K_n^+, considering the edges of the given independent set first and the rest of $E(K_n^+)$ in any desired order. The solution to **Spanning** is similar.

Acknowledgement

This research was supported by the NSF CCF 0728783 grant of Ileana Streinu.

References

[Berg and Jordán 2003] Berg, A. R., Jordán, T.: "Algorithms for graph rigidity and scene analysis"; Proc. 11th ESA, LNCS volume 2832, Springer (2003), 78-89.

[Haas 2002] Haas, R.: "Characterizations of arboricity of graphs"; Ars Combinatorica 63 (2002), 129-137.

[Haas et al. 2007] Haas, R., Lee, A., Streinu, I., Theran, L.: "Characterizing sparse graphs by map decompositions"; Journal of Combinatorial Mathematics and Combinatorial Computing, vol. 62, 3–11 (2007); http://arxiv.org/abs/0704.3843.

[Hendrickson 1992] Hendrickson, B.: "Conditions for unique graph realizations"; SIAM Journal of Computing 21, 1 (1992), 65-84.

[Jacobs and Hendrickson 1997] Jacobs, D. J., Hendrickson, B.: "An algorithm for two-dimensional rigidity percolation: the pebble game"; Journal of Computational Physics 137 (1992), 346-365.

[Laman 1970] Laman, G.: "On graphs and rigidity of plane skeletal structures"; Journal of Engineering Mathematics 4 (1970), 331-340.

[Lee and Streinu 2007] Lee, A., Streinu, I.: "Pebble game algorithms and sparse graphs"; Discrete Mathematics, to appear; http://arxiv.org/abs/math/0702129.

[Lee et al. 2005] Lee, A., Streinu, I., Theran, L.: "Finding and maintaining rigid components"; Proceedings of the 17th Canadian Conference on Computational Geometry, Windsor, Ontario (2005); http://cccg.cs.uwindsor.ca/papers/72.pdf.

[Lee et al. 2007] Lee, A., Streinu, I., Theran, L.: "The slider-pinning problem"; Proceedings of the 19th Canadian Conference on Computational Geometry, Ottawa, Ontario (2007); http://linkage.cs.umass.edu/publications/sliders-cccg.pdf.

[Nash-Williams 1961] Nash-Williams, C. S. J. A.: "Edge-disjoint spanning trees of finite graphs"; Journal London Math. Soc. 36 (1961), 445-450.

[Oxley 1992] Oxley, J. G.: "Matroid theory"; The Clarendon Press Oxford University Press, New York (1961).

[Servatius and Whiteley 1999] Servatius, B., Whiteley, W.: "Constraining plane configurations in CAD: Combinatorics of directions and lengths"; SIAM Journal on Discrete Mathematics 12(1) (1999), 136-153.

[Streinu and Theran 2007] Streinu, I., Theran, L.: "Sparse hypergraphs and pebble game algorithms" European Journal of Combinatorics, special issue for Oriented Matroids'05, to appear (2008); http://arxiv.org/abs/math/0703921.

[Streinu and Theran 2007b] Streinu, I., Theran, L.: "Combinatorial genericity and minimal rigidity"; Manuscript (2007); http://arxiv.org/abs/0712.0031.

[Tay 1984] Tay, T.-S.: "Rigidity of multigraphs I: linking rigid bodies in n-space"; Journal of Combinatorial Theory Series B 26 (1994), 95-112.

[Tutte 1961] Tutte, W. T.: "On the problem of decomposing a graph into n connected factors"; Journal London Math. Soc. 142 (1961), 221-230.

[Whiteley 1988] Whiteley, W.: "The union of matroids and the rigidity of frameworks"; SIAM Journal Discrete Mathematics 1, 2 (1988), 237-255.

[Whiteley 1996] Whiteley, W.: "Some matroids from discrete applied geometry"; In: Matroid Theory (Bonin, J., Oxley, J. G., Servatius, B., Editors), volume 197 of *Contemporary Mathematics*, American Mathematical Society (1996), 171-311.