2D Minimal Graph Rigidity is in NC for One-Crossing-Minor-Free Graphs

Rohit Gurjar, Kilian Rothmund, Thomas Thierauf

STACS 2026

Parallel Complexity

NC^k: Problems that can be solved in $\mathcal{O}(\log^k n)$ time with polynomially many processors

quasi-NC^k: Like NC^k but with quasi-polynomial ($2^{\log^{\mathcal{O}(1)} n}$) many processors

RNC^k: Like NC^k but with randomness

Parallel Complexity

NC^k: Problems that can be solved in $\mathcal{O}(\log^k n)$ time with polynomially many processors

quasi-NC^k: Like NC^k but with quasi-polynomial ($2^{\log^{\mathcal{O}(1)} n}$) many processors

RNC^k: Like NC^k but with randomness

- Solving linear systems is in NC^2 [Mulmuley 1987]
- Bipartite perfect matching is in RNC^2 [Mulmuley et al. 1987] and in quasi-NC^2 [FGT 2016]

Parallel Complexity

NC^k: Problems that can be solved in $\mathcal{O}(\log^k n)$ time with polynomially many processors

quasi-NC^k: Like NC^k but with quasi-polynomial ($2^{\log^{\mathcal{O}(1)} n}$) many processors

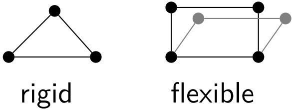
RNC^k: Like NC^k but with randomness

- Solving linear systems is in NC^2 [Mulmuley 1987]
- Bipartite perfect matching is in RNC^2 [Mulmuley et al. 1987] and in quasi-NC^2 [FGT 2016]

Open problem: Is bipartite p.m. in $\mathsf{NC} = \bigcup_{\forall k \in \mathbb{N}_0} \mathsf{NC}^k$?

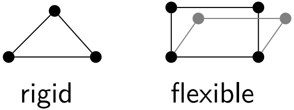
Rigidity of Frameworks

- A *bar-and-joint framework* (G, p) consists of
  - a graph $G = (V, E)$
  - and a realization p in $\mathbb{R}^d$, i.e. $p = (p_i)_{i \in V}, p_i \in \mathbb{R}^d$
- (G, p) is *rigid* if every continuous motion $p(t)$ of its vertices that preserves the edge-lengths such that $(G, p(0)) = (G, p)$ is *trivial*
  Trivial motions for d=2: Translations in x- and y-direction, rotation
  In general there are $\binom{d+1}{2}$ many trivial motions in $\mathbb{R}^d$

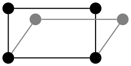


Rigidity of Frameworks

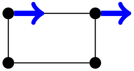
- A *bar-and-joint framework* (G, p) consists of
  - a graph $G = (V, E)$
  - and a realization p in $\mathbb{R}^d$, i.e. $p = (p_i)_{i \in V}, p_i \in \mathbb{R}^d$
- (G, p) is *rigid* if every continuous motion $p(t)$ of its vertices that preserves the edge-lengths such that $(G, p(0)) = (G, p)$ is *trivial*
  Trivial motions for d=2: Translations in x- and y-direction, rotation
  In general there are $\binom{d+1}{2}$ many trivial motions in $\mathbb{R}^d$



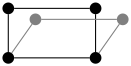
- Testing rigidity of a framework is coNP-hard [Abbot 2008]



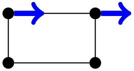
instead for motions



test for velocities at $t = 0$

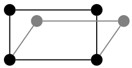


instead for motions

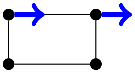


test for velocities at $t = 0$

1. Motion has edge-length constraints: $\|p_i(t) - p_j(t)\|^2 = \|p_i - p_j\|^2$ for all $ij \in E$

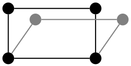


instead for motions

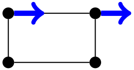


test for velocities at $t = 0$

1. Motion has edge-length constraints: $\|p_i(t) - p_j(t)\|^2 = \|p_i - p_j\|^2$ for all $ij \in E$
2. Differentiate with respect to t: $\langle p_i(t) - p_j(t), p_i'(t) - p_j'(t)\rangle = 0$ for all $ij \in E$

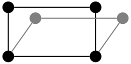


instead for motions

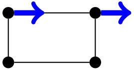


test for velocities at $t = 0$

1. Motion has edge-length constraints: $\|p_i(t) - p_j(t)\|^2 = \|p_i - p_j\|^2$ for all $ij \in E$
2. Differentiate with respect to t: $\langle p_i(t) - p_j(t), p_i'(t) - p_j'(t) \rangle = 0$ for all $ij \in E$
3. Put $t = 0, q = p'(0)$: $\langle p_i - p_j, q_i - q_j \rangle = 0$ for all $ij \in E$



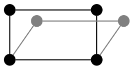
instead for motions



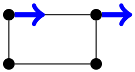
test for velocities at $t = 0$

1. Motion has edge-length constraints: $\|p_i(t) - p_j(t)\|^2 = \|p_i - p_j\|^2$ for all $ij \in E$
2. Differentiate with respect to t: $\langle p_i(t) - p_j(t), p_i'(t) - p_j'(t)\rangle = 0$ for all $ij \in E$
3. Put $t = 0, q = p'(0)$: $\langle p_i - p_j, q_i - q_j\rangle = 0$ for all $ij \in E$

- The linear system above has $\binom{d+1}{2}$ trivial solutions q coming from trivial motions
- (G, p) is *infinitesimally rigid* if there is no non-trivial solution to the linear system



instead for motions

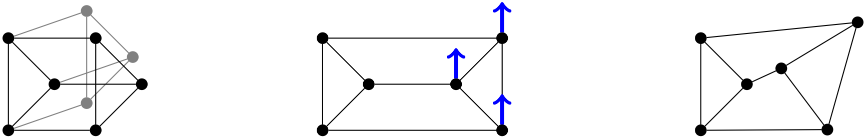


test for velocities at $t = 0$

1. Motion has edge-length constraints: $\|p_i(t) - p_j(t)\|^2 = \|p_i - p_j\|^2$ for all $ij \in E$
2. Differentiate with respect to t: $\langle p_i(t) - p_j(t), p_i'(t) - p_j'(t)\rangle = 0$ for all $ij \in E$
3. Put $t = 0, q = p'(0)$: $\langle p_i - p_j, q_i - q_j\rangle = 0$ for all $ij \in E$

- The linear system above has $\binom{d+1}{2}$ trivial solutions q coming from trivial motions
- (G, p) is *infinitesimally rigid* if there is no non-trivial solution to the linear system
- Testing inf. rigidity is in NC

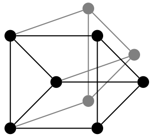
- Inf. rigidity $\Longrightarrow$ rigidity BUT rigidity $\not\Longrightarrow$ inf. rigidity



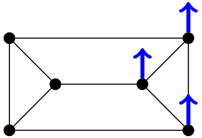
not rigid   rigid, not inf rigid   inf rigid

- How to make rigidity a property of a graph (not a framework)?

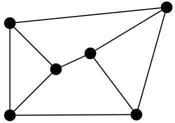
- Inf. rigidity $\implies$ rigidity BUT rigidity $\not\implies$ inf. rigidity



not rigid



rigid, not inf rigid



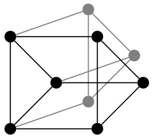
inf rigid

- How to make rigidity a property of a graph (not a framework)?

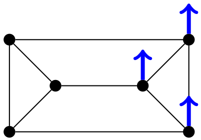
Definition [Asimov-Roth 79]

A graph is d-rigid if a random realization in $\mathbb{R}^d$ is inf. rigid with high probability
Equivalent: $\exists$ inf. rigid realization in $\mathbb{R}^d$

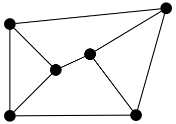
- Inf. rigidity $\Longrightarrow$ rigidity BUT rigidity $\not\Longrightarrow$ inf. rigidity



not rigid



rigid, not inf rigid



inf rigid

- How to make rigidity a property of a graph (not a framework)?

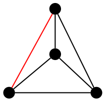
Definition [Asimov-Roth 79]

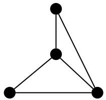
A graph is d-rigid if a random realization in $\mathbb{R}^d$ is inf. rigid with high probability
Equivalent: $\exists$ inf. rigid realization in $\mathbb{R}^d$

- Deciding d-rigidity & finding rigid realization is in RNC by taking a random realization

Minimal Graph Rigidity

- A graph is called *minimally d-rigid* if it is d-rigid and removing any edge destroys this property





- Every d-rigid graph contains a spanning minimally d-rigid subgraph

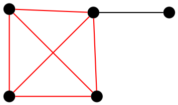
Combinatorial characterization of 2D graph rigidity

- A graph $G = (V, E)$ is called *Laman* if it has $2|V| - 3$ edges and if we have

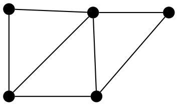
$$|F| \leq 2|V(F)| - 3 \text{ for all } F \subseteq E, F \neq \emptyset$$

Theorem [Laman 1970, Geiringer 1927]

A graph is 2D minimally rigid iff it is Laman



not Laman



Laman

Combinatorial characterization of 2D graph rigidity

- A graph $G = (V, E)$ is called *Laman* if it has $2|V| - 3$ edges and if we have

$$|F| \leq 2|V(F)| - 3 \text{ for all } F \subseteq E, F \neq \emptyset$$

Theorem [Laman 1970, Geiringer 1927]

A graph is 2D minimally rigid iff it is Laman

- Combinatorial characterization is not known for d-rigidity when $d \geq 3$

2D minimal graph rigidity is in quasi-NC^2

- Laman condition:

$$|F| \leq 2|V(F)| - 3 \text{ for all } F \subseteq E, F \neq \emptyset$$

- Note similarity to Hall's condition: For a bipartite graph with partition $L \cup R$ there is an L-saturating matching iff

$$|X| \leq |N(X)| \text{ for all } X \subseteq L$$

- Testing Laman condition reduces to bip. perfect matching [Hendrickson 92]

2D minimal graph rigidity is in quasi-NC^2

- Laman condition:

$$|F| \leq 2|V(F)| - 3 \text{ for all } F \subseteq E, F \neq \emptyset$$

- Note similarity to Hall's condition: For a bipartite graph with partition $L \cup R$ there is an L-saturating matching iff

$$|X| \leq |N(X)| \text{ for all } X \subseteq L$$

- Testing Laman condition reduces to bip. perfect matching [Hendrickson 92]

$\Rightarrow$ 2D Minimal graph rigidity is in quasi-NC^2 [FGT 2016]

2D minimal graph rigidity is in quasi-NC^2

- Laman condition:

$$|F| \le 2|V(F)| - 3 \text{ for all } F \subseteq E, F \neq \emptyset$$

- Note similarity to Hall's condition: For a bipartite graph with partition $L \cup R$ there is an L-saturating matching iff

$$|X| \le |N(X)| \text{ for all } X \subseteq L$$

- Testing Laman condition reduces to bip. perfect matching [Hendrickson 92]

$\Rightarrow$ 2D Minimal graph rigidity is in quasi-NC^2 [FGT 2016]

- Reduction in the other direction is not known!

2D minimal graph rigidity is in quasi-NC^2

- Laman condition:

$$|F| \leq 2|V(F)| - 3 \text{ for all } F \subseteq E, F \neq \emptyset$$

- Note similarity to Hall's condition: For a bipartite graph with partition $L \cup R$ there is an L-saturating matching iff

$$|X| \leq |N(X)| \text{ for all } X \subseteq L$$

- Testing Laman condition reduces to bip. perfect matching [Hendrickson 92]

$\Rightarrow$ 2D Minimal graph rigidity is in quasi-NC^2 [FGT 2016]

- Reduction in the other direction is not known!
- 2D graph rigidity (non-minimal) is not known to be in quasi-NC

Special Graph classes: Bounded Treewidth

- We have the following:

Theorem [Lovasz & Yemini 1982]

A graph G is 2D minimally rigid iff for every edge $e \in E$ the multigraph $G + e$ is the union of two edge-disjont spanning trees

Special Graph classes: Bounded Treewidth

- We have the following:

Theorem [Lovasz & Yemini 1982]

A graph G is 2D minimally rigid iff for every edge $e \in E$ the multigraph $G + e$ is the union of two edge-disjont spanning trees

- With this, 2D graph rigidity can be expressed in monadic second-order (MSO) logic

Logspace-version of Courcelle [EJT2010]

Graph properties definable in MSO-logic can be decided in Logspace $\subseteq$ NC

$\Rightarrow$ 2D graph rigidity is in NC for graphs with bounded treewidth

Planar graphs

By combining among others

- Results from Haas et al. 2003
- Results from Rollin et al. 2019
- An NC algorithm for a special flow problem on planar graphs by Miller and Naor 1995

we get

1. Deciding whether a planar graph is Laman is in NC^2
2. For a planar Laman graph we can compute an inf. rigid realization in NC^2

Main results

- A graph is *one-crossing* if it has a drawing in the plane with at most one crossing
- A graph is *one-crossing-minor-free* if it is H-minor-free for some one-crossing graph H
- Examples:
  - Planar graphs (K_5- and $K_{3,3}$-minor-free)
  - Bounded-treewidth graphs (planar-minor-free)
  - $K_{3,3}$-minor-free graphs

Theorem 1

Deciding 2D minimal graph rigidity is in NC^3 for one-crossing-minor-free graphs

Theorem 2

Computing inf. rigid realization is in NC^2 for $K_{3,3}$-minor-free 2D minimally rigid graphs

Technique

- A graph can be decomposed into 3- or 4-connected components in NC^2 [Datta et al. 2009, Elberfeld & Vazirani 2019]

Theorem [Asano 1985]

Every 3-connected $K_{3,3}$-free graph is either planar or the K_5

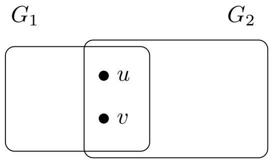
Theorem [Robertson & Seymour 1993]

For a one-crossing-minor-free graph, the 4-connected components are either planar or of bounded treewidth.

- This has been used multiple times already, e.g.
  - Computing/Counting perfect matchings
  - Graph isomorphism
  - Constructing DFS trees

Decomposing at a separating pair

- $G \setminus \{u, v\}$ splits into G_1, G_2:



Split Lemma

1. $uv \in E(G)$: G is Laman iff G_1 and G_2 are Laman
2. $uv \notin E(G)$: G is Laman iff for one graph, say G_1, $G_1 \cup \{uv\}$ and G_2 are Laman

- With this we decompose graph into 3-connected components such that G is 2D minimally rigid iff all components are 2D minimally rigid
- Split Lemma for separarting triples more elaborate

Merging inf. rigid realizations along separating pairs

Merge Lemma

Let $G = (V, E)$ be a Laman graph with a separating pair $\{u, v\}$. Let G_1, G_2 be Laman components obtained by splitting along $\{u, v\}$. Let p_1, p_2 be inf. rigid realizations of the components such that

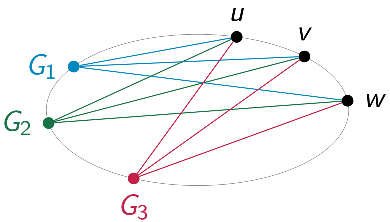
$$p_1(u) = p_2(u)$$
$$p_1(v) = p_2(v)$$
$$p_1(u) \neq p_1(v).$$

Then $p = p_1 \cup p_2$ is an inf. rigid realization of G.

- With this we compute an inf. rigid realization for $K_{3,3}$-minor-free Laman graphs

Merging realizations along separating triples

- Merging inf. rigid realization at separating triplets does not necessarily give an inf. rigid realization, e.g. $K_{3,3}$ with separating triplet $\{u, v, w\}$



- Open Problem: Is there a more advanced merging that works?

Summary

	Rigidity	Minimal Rigidity	Realization Computation
General graphs	RNC^2	RNC^2 and quasi-NC^2	RNC^2
$K_{3,3}$-minor-free graphs	RNC^2	NC^2	NC^2
One-crossing-minor-free graphs	RNC^2	NC^3	RNC^2
Bounded treewidth graphs	NC^2	NC^2	RNC^2

- Open problems: NC or quasi-NC algorithms for all entries in the table above

Thank You!

kilian.rothmund@uni-ulm.de