Geometric Constraint Solving

Part 3
Triangle Solvers for 2D
Solver Architecture

Phase 1: Decompose
- Geo elements become graph vertices V
  - Vertex attribute: number of coordinates
- Geo constraints become graph edges E
  - Edge attribute: number of equations
- Generically well-constrained
Problem Translation

- Geo elements $\mapsto$ graph vertices $V$
  - Vertex attribute: number of coordinates
- Geo constraints $\mapsto$ graph edges $E$
  - Edge attribute: number of equations
- Generically well-constrained
  - $0 \leq \sum w(v) - \sum w(e) \leq K$
  - where $K$ depends on the geometry
- Subgraph condition needed

Running Example
Graph Analysis Objectives

- Identify subproblems and generate a plan for solving

Vertex / Edge Attributes
Solvable Subsystems

and larger subgraphs ...

Cluster Building

- Start with core, a minimal subgraph

- Make sequential extensions repeatedly
Start of Cluster 1

Cluster 1 completing
Cluster 2

Result of Cluster Solving
Cluster Merge

- Match shared geo elements – an assembly problem
- Can be reduced to base constructions

Graph Structure for Merge

Core

Sequential extensions
Summary So Far

- We decompose the graph into clusters that will be solved separately
- Clusters are merged into larger ones
- The decomposition is recursive
- Clusters are substructured into core and sequential extensions. This applies recursively

About the Scope

- The patterns are triangle decomposition
- The solver is successful on a wide variety of problems but not all
- The scope can be extended with more complex core patterns and merge rules
- There is a general decomposition algorithm that does not require a catalogue of patterns
Theory of Triangle Decomposition

1. Only quadratic equations need to be solved
2. Decomposition is Church-Rosser: any order is fine
3. Church-Rosser is maintained by the general decomposition algorithm
4. Different decompositions result in congruent geometry

Phase 2: Solve
Solver Architecture

Algebraic solver to determine coordinates

- Triangle decomposition $\Rightarrow$ only quadratic equations need to be solved

Cluster Core

**Simple Core:**
Any edge incident to weight 2 vertices

**Solution:**
Geo elements placed in standard position

**Inadmissible Edges:**
Two lines that should be parallel
Sequential Element Placing

Six rules arise:

\[(p, p) \rightarrow p\]
\[(p, l) \rightarrow p\]
\[(l, l) \rightarrow p\]
\[(p, p) \rightarrow l\]
\[(p, l) \rightarrow l\]
\[(l, l) \rightarrow l\]

Rule (*) is underdetermined

Sequential Rules (1)

\[(p, p) \rightarrow p\]
\[(p, l) \rightarrow p\]
Sequential Rules (2)

\[(p, p) \rightarrow l \quad (p, l) \rightarrow l\]

Result of Cluster Solving

How to combine these clusters?
Cluster Merging

- Looks like an assembly problem: Match the shared geometric elements.
- Executed in these steps:
  1. Place third element wrt other two
  2. Position clusters by two elements
- More details later

Derived Constraints

**Key Concept:**
- Measure the position of shared elements in each cluster
- Derive constraints from positions
Execution

Triangle decomposition works well in 2D and has good theoretical properties

A general decomposition exists and is more expensive

Church-Rosser generalizes on the graph side

Congruence result may not generalize

Summary, Part 3

- Triangle decomposition works well in 2D and has good theoretical properties
- A general decomposition exists and is more expensive
- Church-Rosser generalizes on the graph side
- Congruence result may not generalize