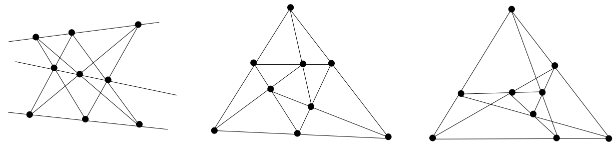

Figure 1: (*left*) configuration of Pappus's hexagon theorem; (*middle and right*) configurations consist of embedded equilateral triangles

1 Geometric Constraints Problem - algorithm problem

1.1 Input, output and problem statement

The input is universe:

- A set of primitive geometric object id's
- a set of primitive geometric constrains between objects

Desired output could be:

1. Does it have a solution in 2D?
2. Find a realization in 2D
3. Find all realizations in 2D
4. Is the set of realizations finite in 2d?
5. Does the input generically satisfy 1 and 4?

Computational Geometry - the study of efficient algorithms for solving geometric problems. Examples of problems treated by computational geometry include determination of the *convex hull* and *Voronoi diagram* for a set of points, *triangulation* of points in a plane or in space, and other related problems. But for computational geometry, the subsystem size is bounded. For general geometric constraints solving problems, the size is not bounded.

For 2D & 3D dimensional geometric constraints solving over $\mathcal{R}$.

- generic or combinatorial questions
- questions for which combinatorics alone is insufficient. (algebraic geometry is required)
- definition of genericity

Problem: If a system with only parameterless constraints (tangency, colinearity, coplanarity), how to get the corresponding 'generic' definition?

1.2 Conjecture of point/lines incidence constraint system

Here is one conjecture relates to determining dependent constraints combinatorially, when no notion of genericity is present (all constraints are parameterless). It could give hints on what the appropriate notion of genericity would be - *i.e.*, what questions can be purely combinatorially answered for such constraint systems.

Conjecture: Take any $m \times m$ {0,1} matrix M in which every row has 3 1's and every column has 3 1's. If M is realizable on the Euclidean plane as point(row) and line(column) incidences, then every realization of any $m \times m - 1$ submatrix of M automatically is a realization of M. I.e., the colinearity represented by the last column is implied by the remaining $m - 1$ colinearities. Another way to say this is that for any $m \times m - 1$ submatrix M' of M, the only realizations of M' are the realizations of M itself.

1. for $m = 7$, this is the Z_2^3 character table or Fano plane and it is not realizable (Hadamard minus 1 dim), so left hand side of conjecture doesn't even apply.

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

2. for $m = 8$, there are NO configurations that are realizable. (See http://mathworld.wolfram.com/Configuration.html)

3. for $m = 9$, are there any atleast 3 matrices besides pappus that realizable. See Fig 1

4. for $m = 10$, Hilbert's intuitive geometry book gives examples including Desargues of so-called (10-3, 10-3) configurations: do they violate the above conjecture?

And the conjecture could be generalized in 2 ways

1. in 3D, change lines to planes and put 4 instead of 3 as the number of 1's in any row or col

2. in either dimension, one can take any $m \times m - k$ submatrix M' for which M is the unique extension satisfying the 3-3 (or 4-4 in 3D) condition. In this case the only realizations of M' are the realizations of M itself, i.e., the remaining k columns in M are colinearity constraints that are dependent on the $m - k$ colinearity constraints represented by the columns of M'

1.3 Open questions

How to classify the constraint system based on how many solutions it has generically.

- none (overconstrained which includes well-overconstrained system and not well-overconstrained system)
- 1 (globally rigid wellconstrained system)
- finitely many (rigid wellconstrained system)
- infinitely many (underconstrained)

For point-distance system in 2D, all these questions are answered. But for 3D, none is answered yet.
For point-distance system in 2D/3D, these are open questions:

- in each of above 4 cases, what is the complexity of solution
- how to find minimum fan-in DR-plan
- characterize classes of graphs that have bounded size DR-plan
- characterize classes of graph for which finding minimum DR-plan has a polynomial time algorithm. Consider planar graph or fixed length genus graph or bounded graphs. (For general graphs, problem of finding minimum DR-plan is NP-hard)
- for given graph, give one good approximation algorithm to find minimum rigid subgraph in it. (problem of finding minimum rigid subgraph is NP-complete even in 2D)
- give a *good* enumeration algorithm for planar Laman graphs. Good means polynomial time and space. Laman graph has underlying matroid while planar Laman graph doesn't

1.4 Recap

1.4.1 Generic questions

- find combinatorial methods to classify solution spaces into four categories
  - none solution
  - unique solution
  - finite solutions
  - infinite solutions

  I.e., we need combinatorially detect all constraints dependencies. For example,
  - 3D rigidity bananas problem
  - point-line incidences problem. (See the above conjecture)
  - angle constraints problem in 2D. (See the previous note)
- generic complexity of solution for finite case
  - size of optimal decomposition
  - complexity of finding optimal decomposition

  why decomposition into rigid components? Distances are invariables of Euclidean groups.
- generic complexity of solution for infinite case
  - existence of a good "completion" constraints
  - complexity of "walking" or "road mapping" or "sampling" solution space in term of completion edges
  - complexity of determine feasible parameter values for completing constraints
- modifying the constraint system
  - removing overconstraints
  - forcing unique solution by adding overconstraints
  - finding alternate constraint system that has low solution complexity and preserves desired solution
  - enumerating all constraints systems of n vertices having some generic properties

1.4.2 Questions about defining genericity appropriately

- how to define genericity of parameterless constraint system (incidence, perpendicularity, parallelity, colinearity, coplanarity, ...)
- how to define symmetric constraint systems (some parameters are forced to be equal)

1.4.3 Nongeneric questions

- solving. There are 3 types of methods for solving: algebraic, numeric, semi-numeric.
- walking through solution space
- packing desired solutions
- classifying nongeneric cases. (Zeroes of polynomial system in parameters that is needed to define genericity)