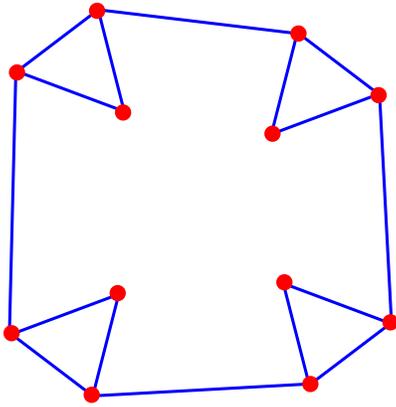


Problem Set 5

Due: Tuesday, October 23th, 2012

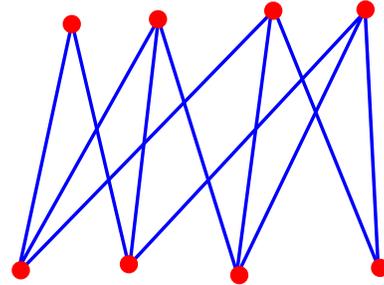
We will drop (ignore) your lowest score on any one problem.

Problem 1. Characterize which of the following graphs are generically minimally rigid, generically non-minimally rigid, or generically flexible. Justify your answer using any of the theorems from lecture.



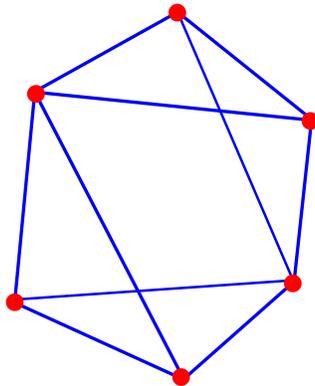
(a)

http://courses.csail.mit.edu/6.849/fall12/psets/rigid_a.pdf



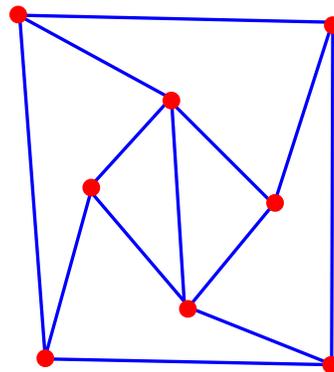
(b)

http://courses.csail.mit.edu/6.849/fall12/psets/rigid_b.pdf



(c)

http://courses.csail.mit.edu/6.849/fall12/psets/rigid_c.pdf

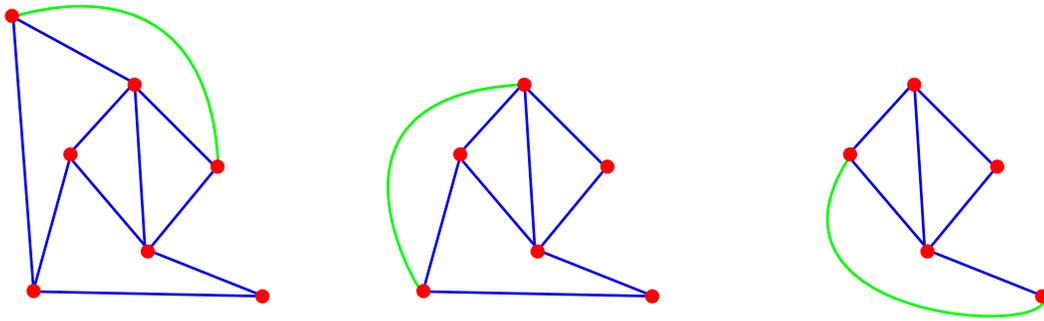


(d)

http://courses.csail.mit.edu/6.849/fall12/psets/rigid_d.pdf

Solution:

- (a) This graph is generically flexible. It has 12 vertices and 16 edges, which is too few to be rigid by Laman.
- (b) This graph is generically flexible. It has 8 vertices and 11 edges, which is too few to be rigid by Laman.
- (c) This graph is non-minimally generically rigid. By removing any of the non-crossing edges, it is easy to see the resulting graph can be constructed by type 1 Henneberg operations.
- (d) This graph is minimally generically rigid. The first three reverse-Henneberg operations are shown and the rest of the type-1 Henneberg operations needed can be seen easily.

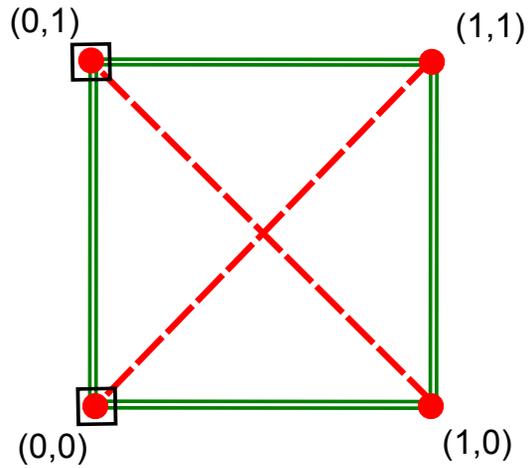


Problem 2. Give an efficient algorithm that checks whether a given graph is *redundantly rigid*, i.e., the removal of any single edge results in a graph that is generically rigid (not necessarily minimally).

Solution: One simple way of doing this is to run the pebble covering algorithm on every sub-graph created by removing a single edge. If all of these are generically rigid, then our condition is met. This algorithm runs in $O(V^2E)$ time.

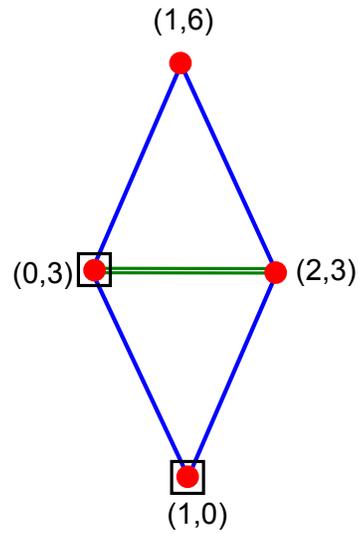
We can do this more efficiently. First, run the original pebble algorithm, making note of the edges which are redundant. We now enter a new phase, where we move pebbles from edges not marked as redundant to edges marked as redundant, in a BFS manner, until we see there is no path allowing a pebble to be moved off of an edge, or the entire graph is marked as redundant. Performing this additional search requires $O(V^2)$ time giving the same running time as the original pebble algorithm.

Problem 3. Write down and solve the linear programs for the infinitesimal motion of the tensegrities below. You should be able to solve the linear programs (or prove they have no solution) by hand, using geometric arguments. Alternatively, Matlab, Mathematica, and Maple have linear-program solvers, and there are also online applications to do so. Blue lines represent bars; green, double lines represent struts; and red, dashed lines represent cables. Vertices with a box around them are pinned to the plane.



(a)

http://courses.csail.mit.edu/6.849/fall12/psets/tenseg_a.pdf



(b)

http://courses.csail.mit.edu/6.849/fall12/psets/tenseg_b.pdf

Solution:

- (a) First, let us label the vertices as such:

$$\begin{array}{cc} a & b \\ c & d \end{array}$$

The two pinned vertices give us the following constraint:

$$d_x(a) = d_y(a) = d_x(c) = d_y(c) = 0$$

The edges give us the following inequalities:

$$\begin{aligned} ab &\rightarrow d_x(b) \geq 0 \\ ad &\rightarrow d_x(d) \leq d_y(d) \\ bc &\rightarrow d_x(b) + d_y(b) \leq 0 \\ bd &\rightarrow d_y(b) \geq d_y(d) \\ cd &\rightarrow d_x(d) \geq 0 \end{aligned}$$

Solving, either by symmetry or standard methods, we discover that there are no infinitesimal motions.

- (b) First, let us label the vertices as such:

$$\begin{array}{ccc} & a & \\ b & & c \\ & d & \end{array}$$

The two pinned vertices give us the following constraint:

$$d_x(b) = d_y(b) = d_x(d) = d_y(d) = 0$$

The edges give us the following inequalities:

$$ab \rightarrow d_x(a) - 3d_y(a) = 0$$

$$ac \rightarrow 3d_x(a) - 3d_x(c) - d_y(a) + d_y(c) = 0$$

$$bc \rightarrow d_x(c) \geq 0$$

$$cd \rightarrow d_x(c) - d_x(d) + 3d_y(c) - 3d_y(d) = 0$$

After manipulating the equations, one can see this admits an instantaneous motion in which points a and c move down and to the right with a horizontal velocity three times that of the vertical velocity.

MIT OpenCourseWare
<http://ocw.mit.edu>

6.849 Geometric Folding Algorithms: Linkages, Origami, Polyhedra
Fall 2012

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.