A 12/11-Approximation Algorithm for Minimum 3-Way Cut

David Karger (MIT), Phillip Klein (Brown), Cliff Stein (Columbia)
Mikkel Thorup (AT&T), Neal Young (UCR)

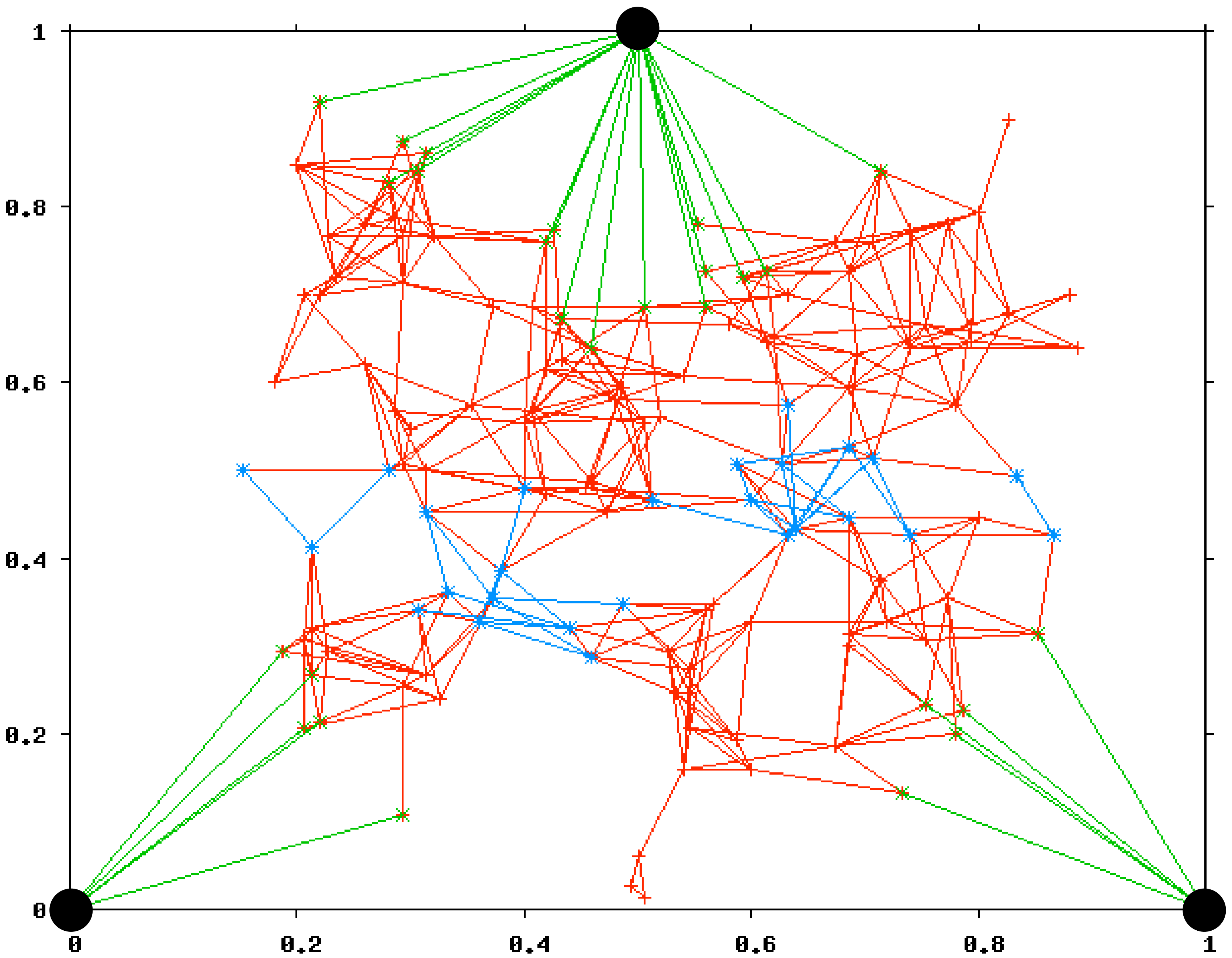
``As the field of approximation algorithms matures, methodologies are emerging that apply broadly to many NP-hard optimization problems. One such approach has been the use of metric and geometric embeddings in addressing graph optimization problems. Faced with a discrete graph optimization problem, one formulates a relaxation that maps each graph node into a metric or geometric space, which in turn induces lengths on the graph's edges. One solves this relaxation optimally and then derives from the relaxed solution a near-optimal solution to the original problem."

problem: 3-way cut

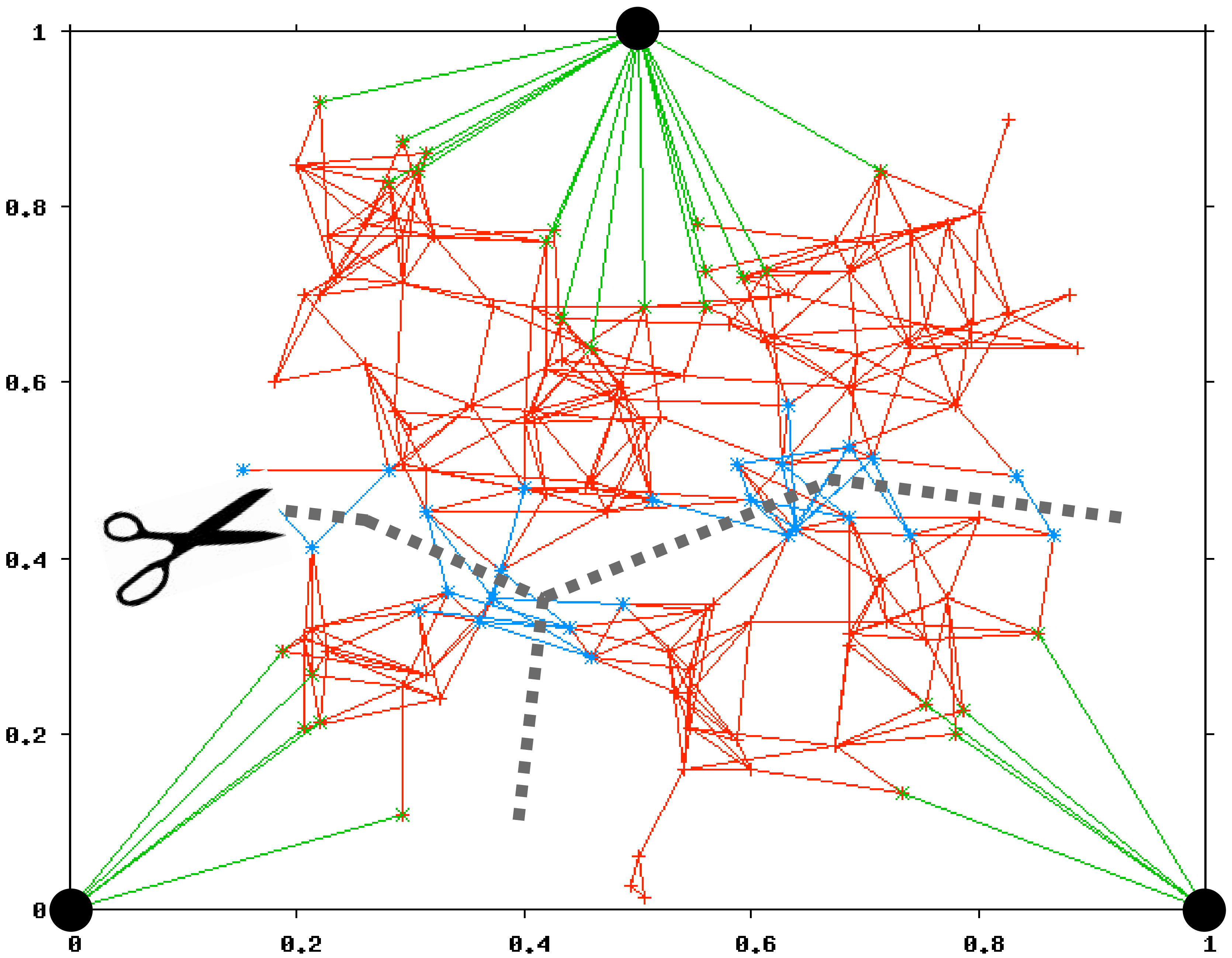
- input: undirected graph, three terminal nodes
- output: three-way cut (subset of edges whose removal separates the terminals)
- objective: minimize number of edges cut

- NP-HARD

3-way cut



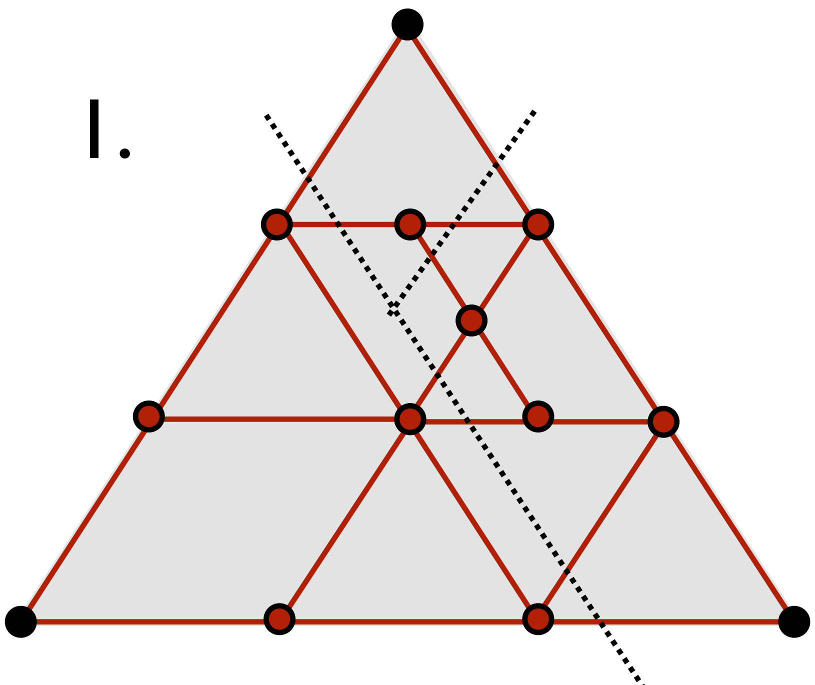
3-way cut



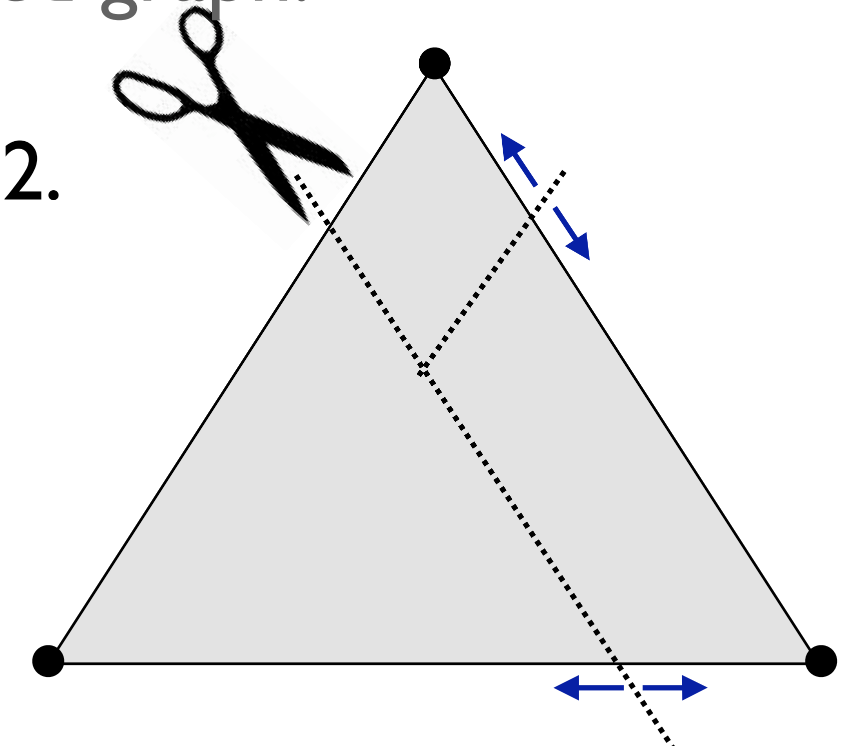
Approach [Calinescu et al, 1998]

1. Embed graph into triangle.
2. Cut *triangle* using randomized cutting scheme.
   ... induces cut of embedded graph.

1.

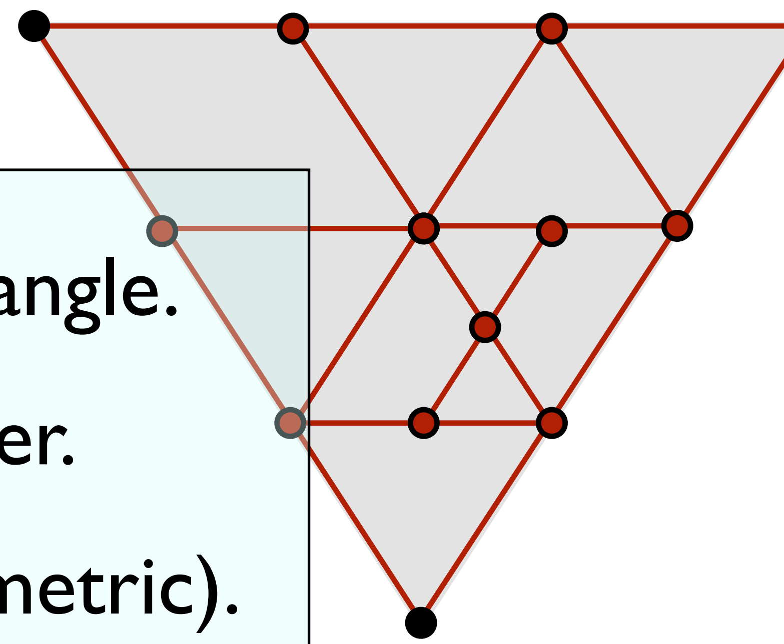


2.



goal: Bound expected number of edges cut.

Step 1: embedding



a. Assign vertices to points in the triangle.

b. Constrain each terminal to a corner.

c. Minimize sum of edge lengths (L_1 metric).

- Optimal embedding via linear program.
- Value of LP is at most |optimal 3-cut| .

LP for finding optimal embedding

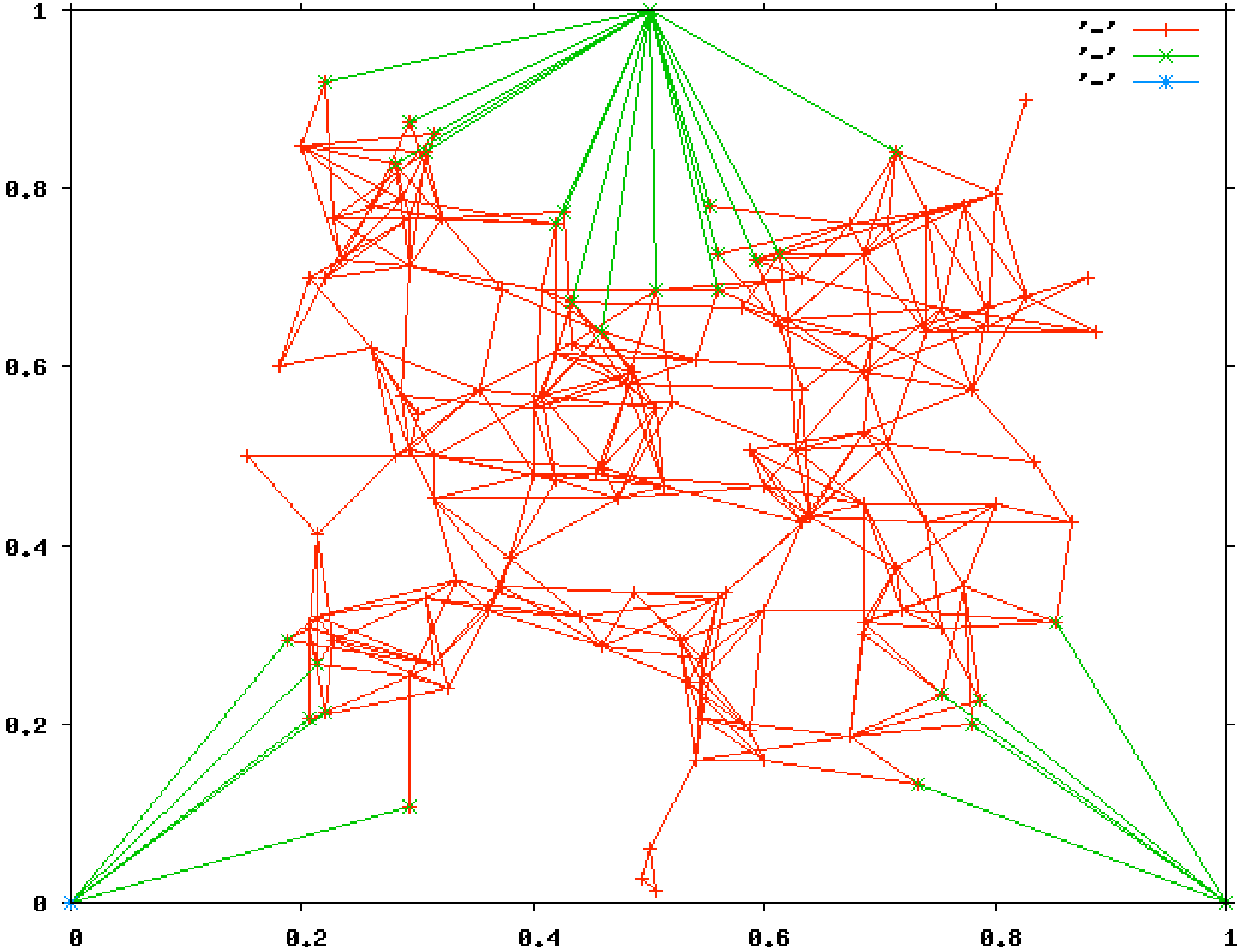
$$\text{minimize} \quad \frac{1}{2} \sum_{(u,v)\in E} d_{uv}$$

$$(x_{t1}, y_{t1}, z_{t1}) = (1, 0, 0)$$
$$(x_{t2}, y_{t2}, z_{t2}) = (0, 1, 0)$$
$$(x_{t3}, y_{t3}, z_{t3}) = (0, 0, 1)$$

$$(\forall u) \quad x_u + y_u + z_u = 1$$
$$(\forall u, v) \quad d_{uv} \geq |x_u - x_v| + |y_u - y_v| + |z_u - z_v|$$

Each vertex u is mapped to a point (x_u, y_u, z_u),
determined by the LP,
to minimize sum of embedded edge lengths.

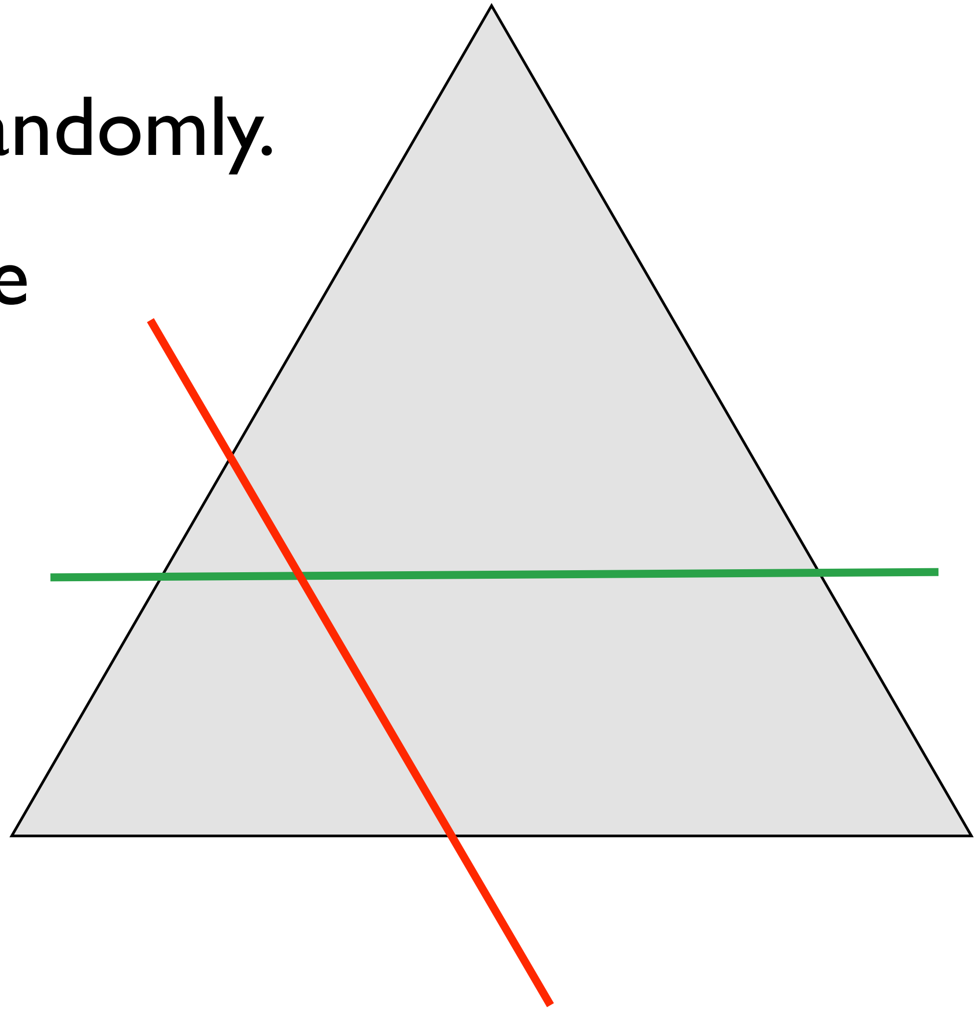
Embedding (animated)



Step 2: cutting the triangle (Calinescu et al's scheme)

a. Choose 2 of 3 sides randomly.

b. Choose a random slice parallel to each sides.

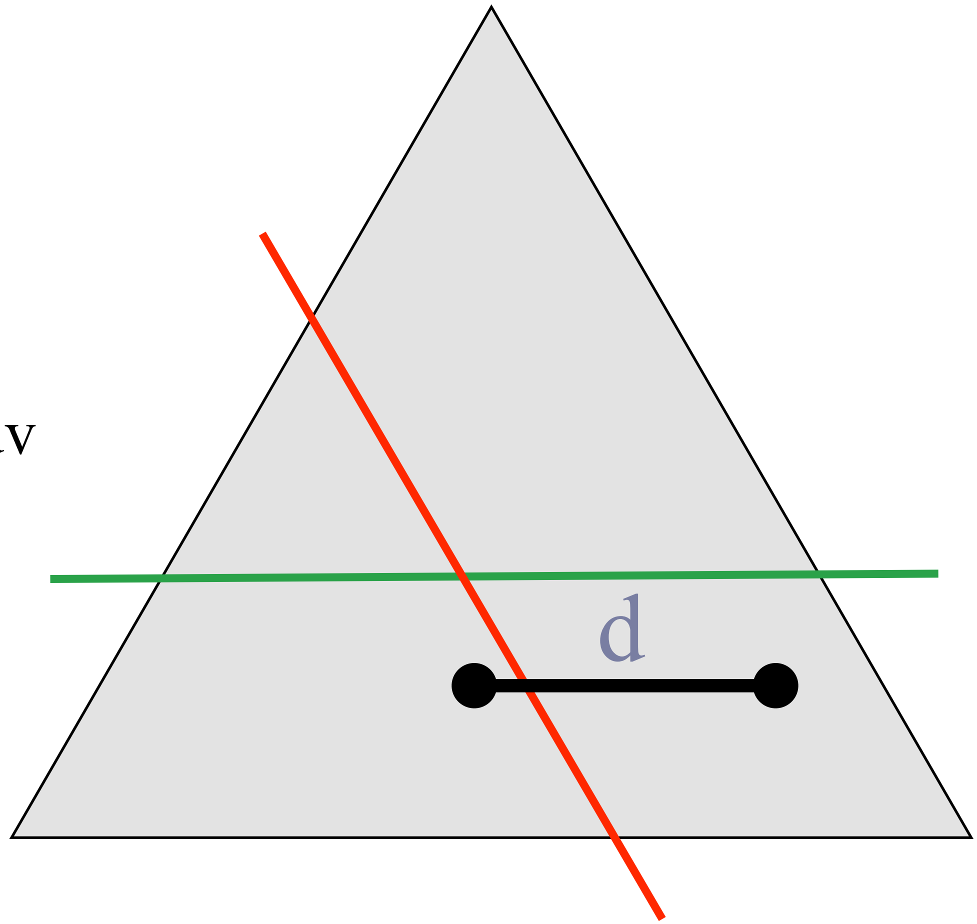


Pr[edge (u,v) cut] ≤ (4/3) d_{uv}

a. Pr[cut by red] = (2/3) d_{uv}

b. Pr[cut by green] = (2/3) d_{uv}

c. Pr[cut] ≤ 2×(2/3) d_{uv}



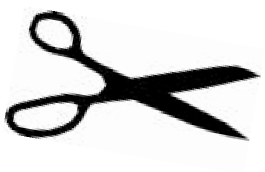
Expected #edges cut ≤ 4/3 OPT

lemma: $\Pr[\text{edge } (u,v) \text{ cut}] \le \frac{4}{3} d_{uv}$

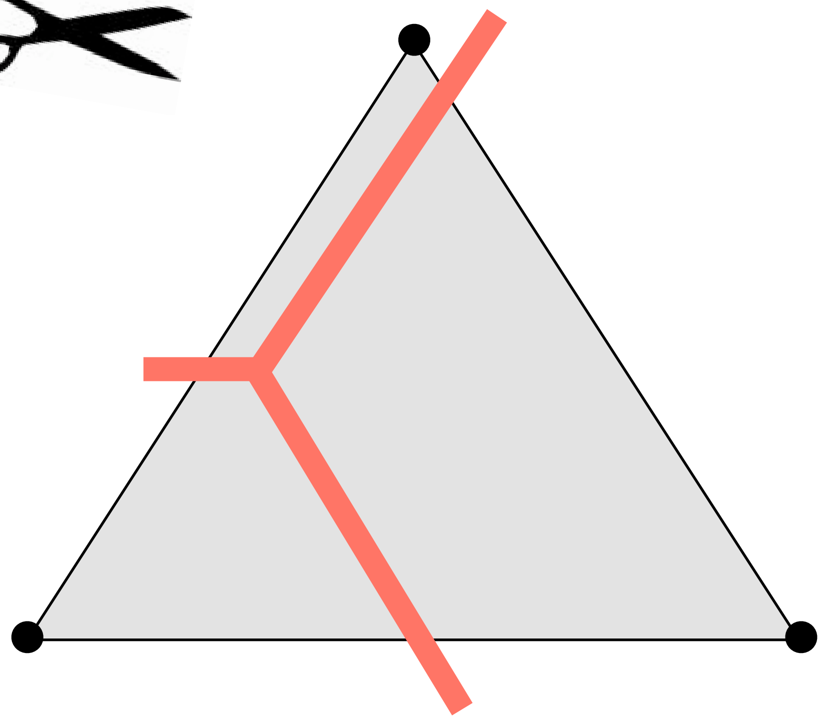
corollary:

$$\text{expected number of edges cut} \le \frac{4}{3} \sum_{(u,v)\in E} d_{uv}$$
$$= \frac{4}{3} |\text{value of LP}|$$
$$\le \frac{4}{3} |\text{optimal 3-cut}|$$

Better cutting scheme



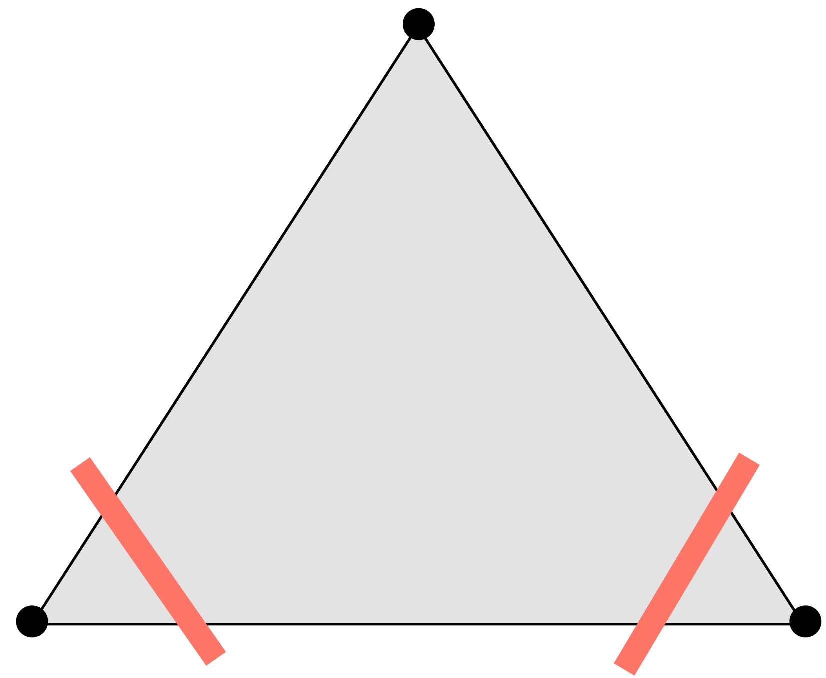
ball cut



(probability 8/11)

or

corner cut

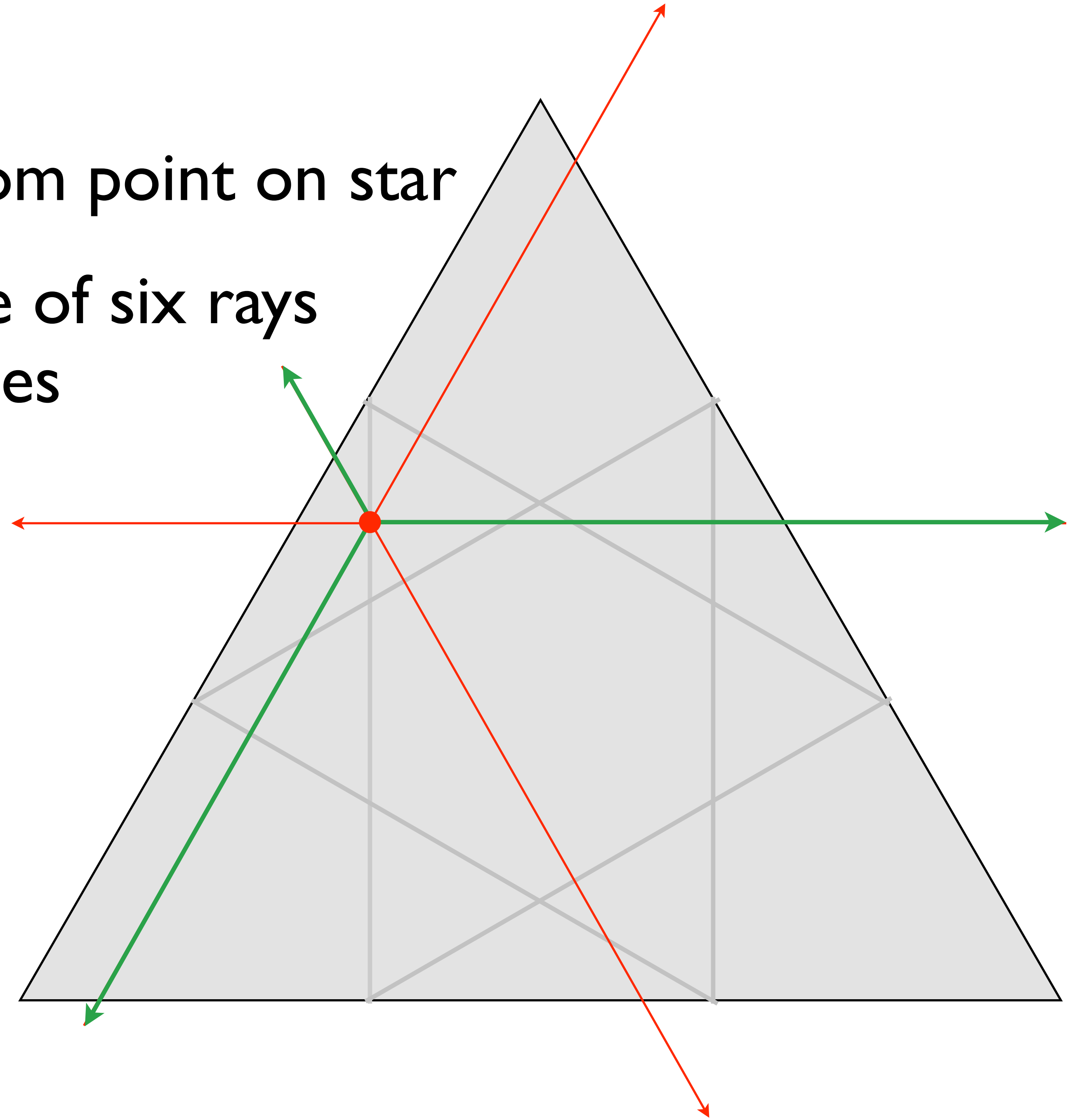


(probability 3/11)

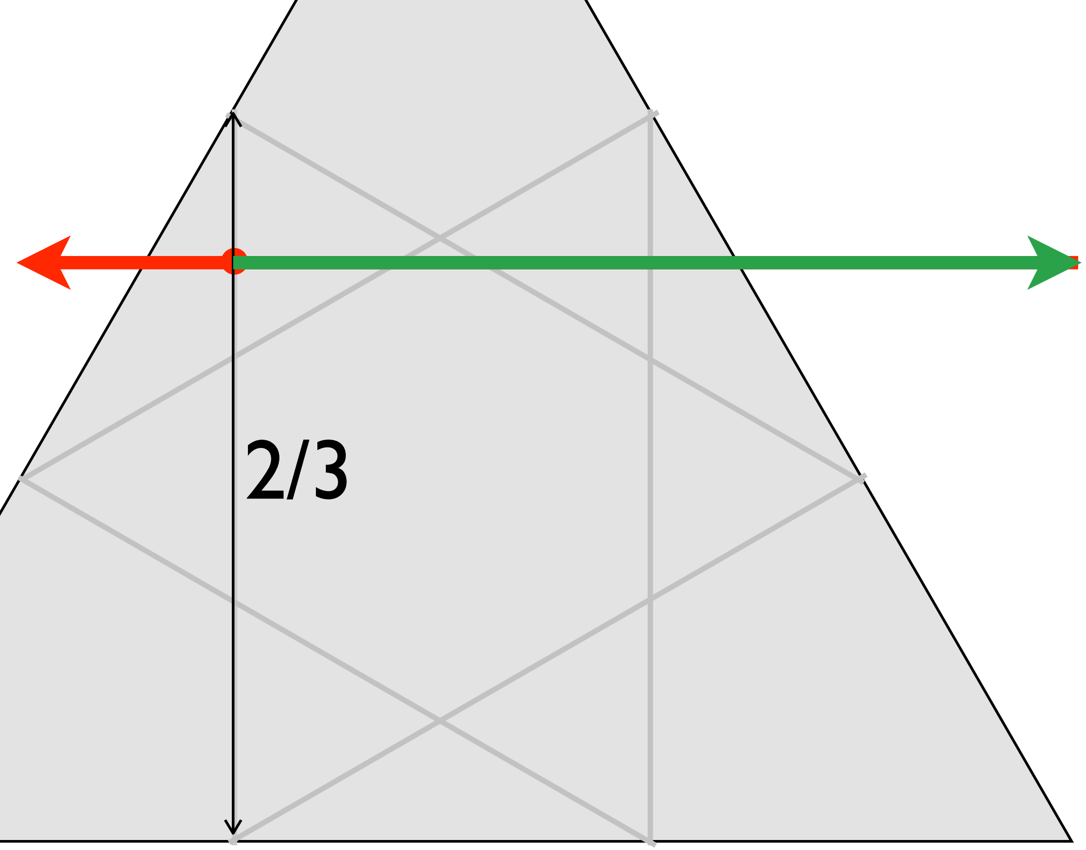
Ball cut

i. Choose random point on star

ii. Choose three of six rays parallel to sides

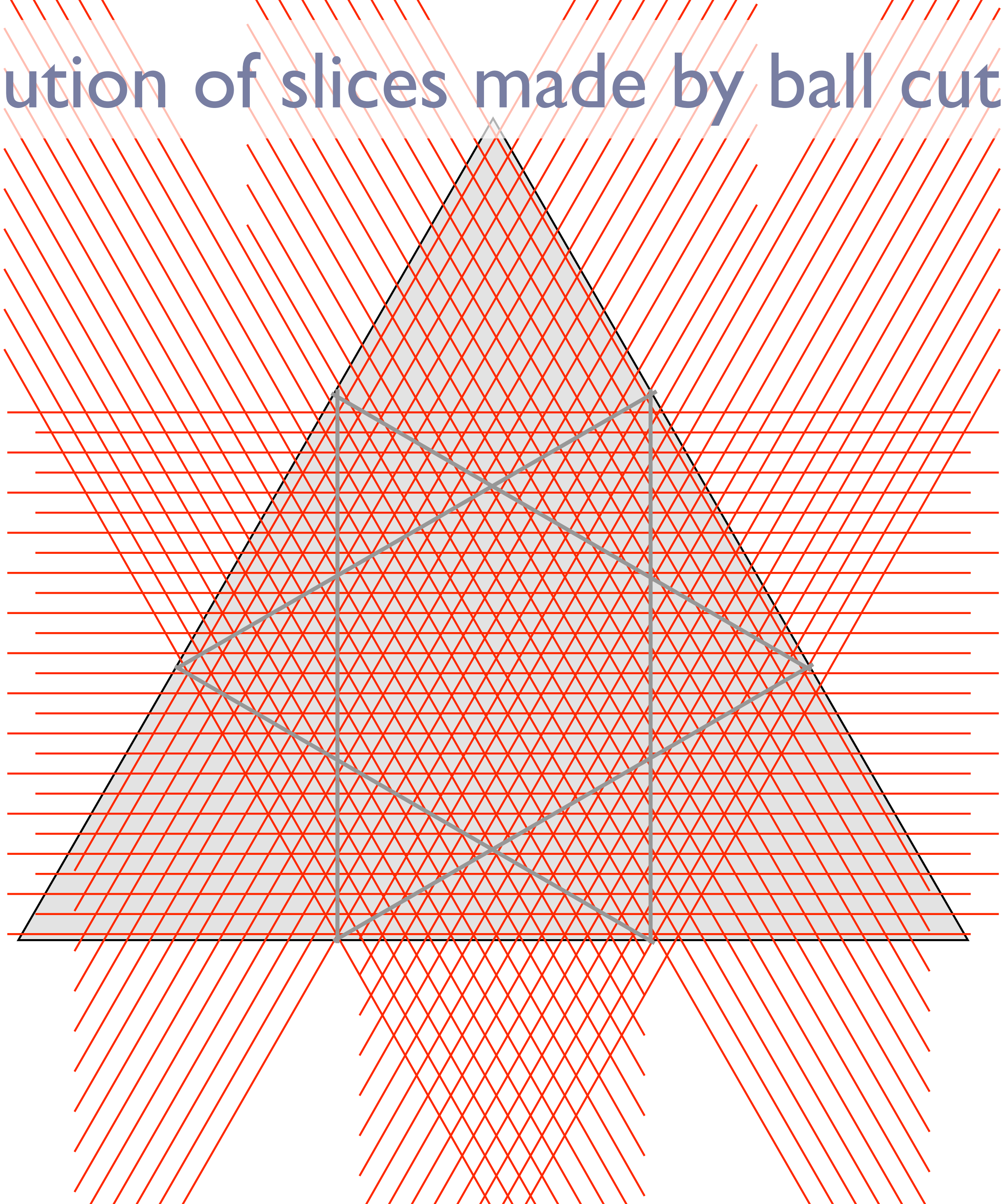


density of ball cut slices



3/2	-- density of horizontal slice
x 1/2	-- only one of two rays (red or green)
x 2	-- segment can be cut from two orientations
x 8/11	-- probability of ball cut
= 12/11	

distribution of slices made by ball cuts



Expected #edges cut ≤ 12/11 OPT

lemma: $\Pr[\text{edge } (u,v) \text{ cut}] \le \frac{12}{11} d_{uv}$

corollary:

$$\text{expected number of edges cut} \le \frac{12}{11} \sum_{(u,v)\in E} d_{uv}$$
$$= \frac{12}{11} |\text{value of LP}|$$
$$\le \frac{12}{11} |\text{optimal 3-cut}|$$

More

- Generalizes to K-way cut (ratio < 1.34...)
- K=3 case done also by Cunningham and Tang
- Meta-problem of finding an optimal cutting scheme can be formulated as an infinite LP!
- For K=3, no better cutting scheme for this LP relaxation is possible. Would need better relaxation to improve result.
- K > 3 much harder. Improve constant?

probability that edge is cut