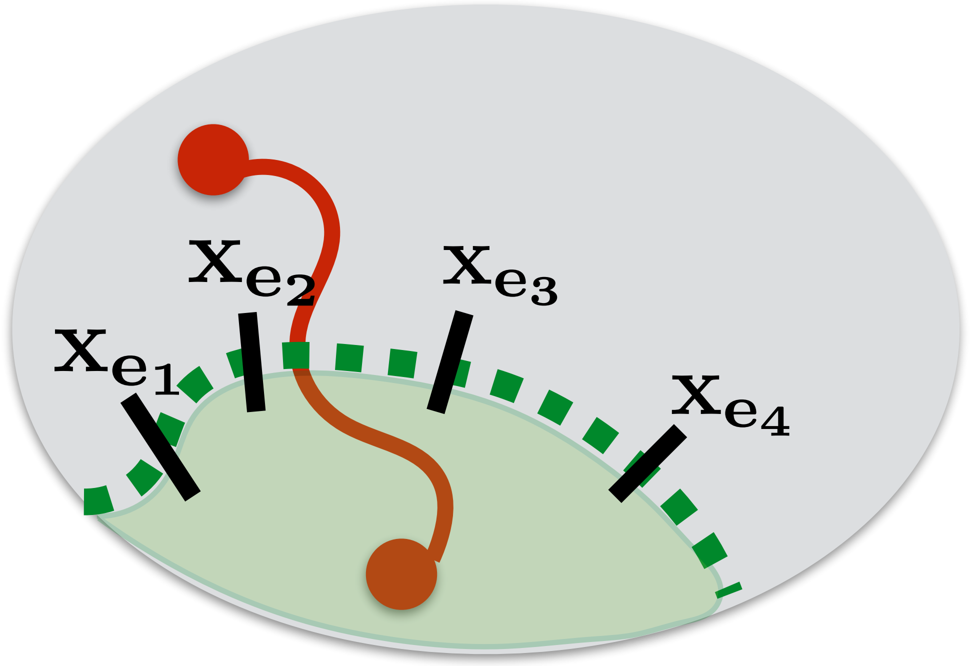
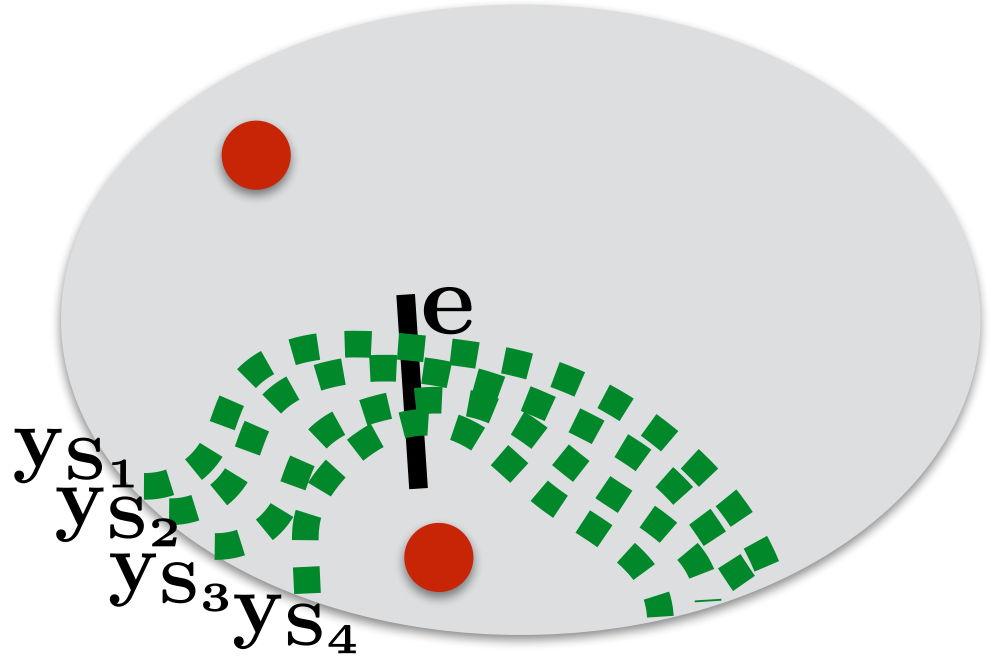
Steiner forest



Initialization:

$$\mathrm{x} \leftarrow 0, \mathrm{y} \leftarrow 0$$



Iteration: while x not satisfiable

in parallel, raise every unfrozen $\mathrm{y_S}$ with minimal S

stopped by tight constraint (e)

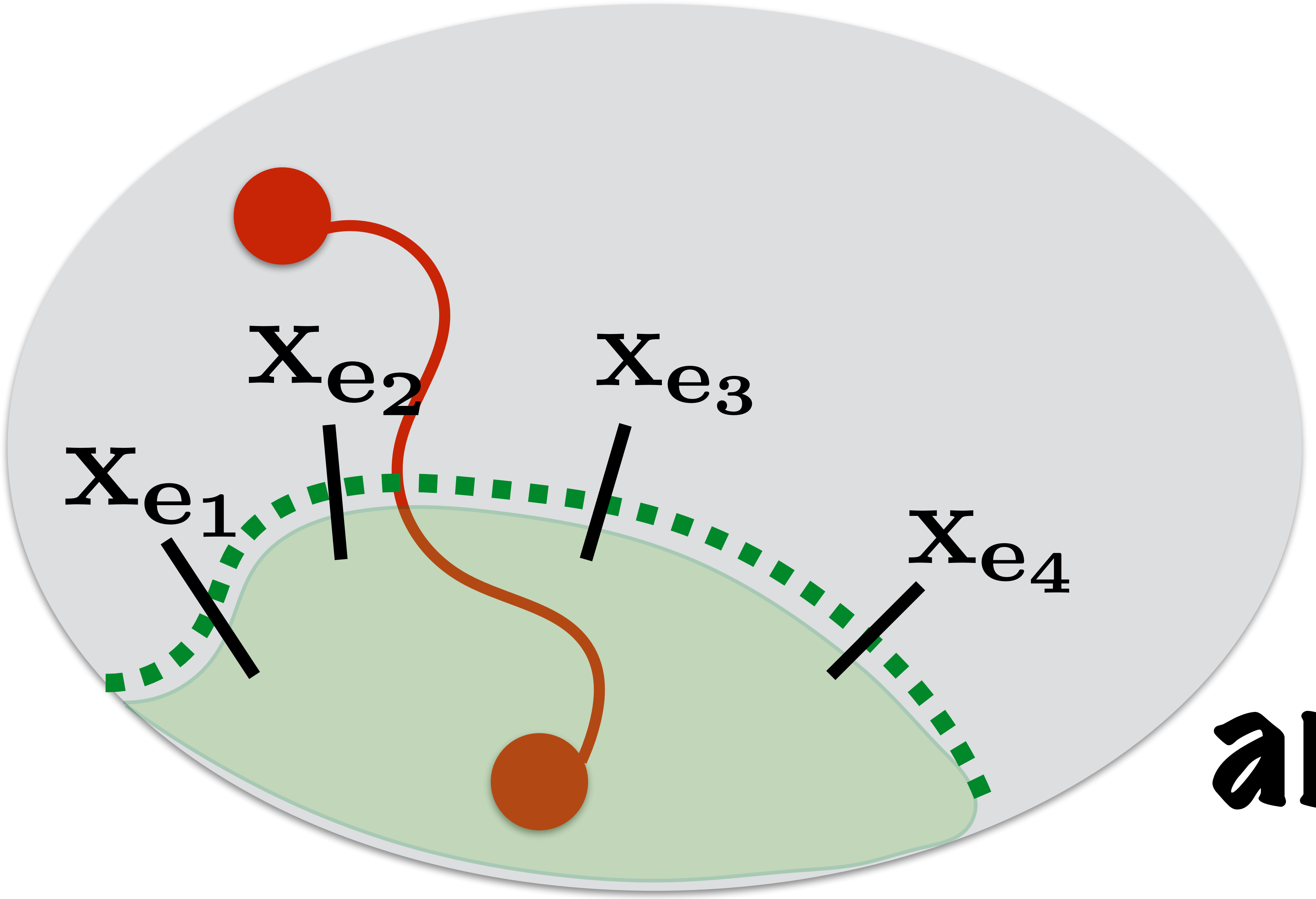
$$\mathrm{x_e} \leftarrow 1$$

freeze $\mathrm{y_S}$ in tight constraints

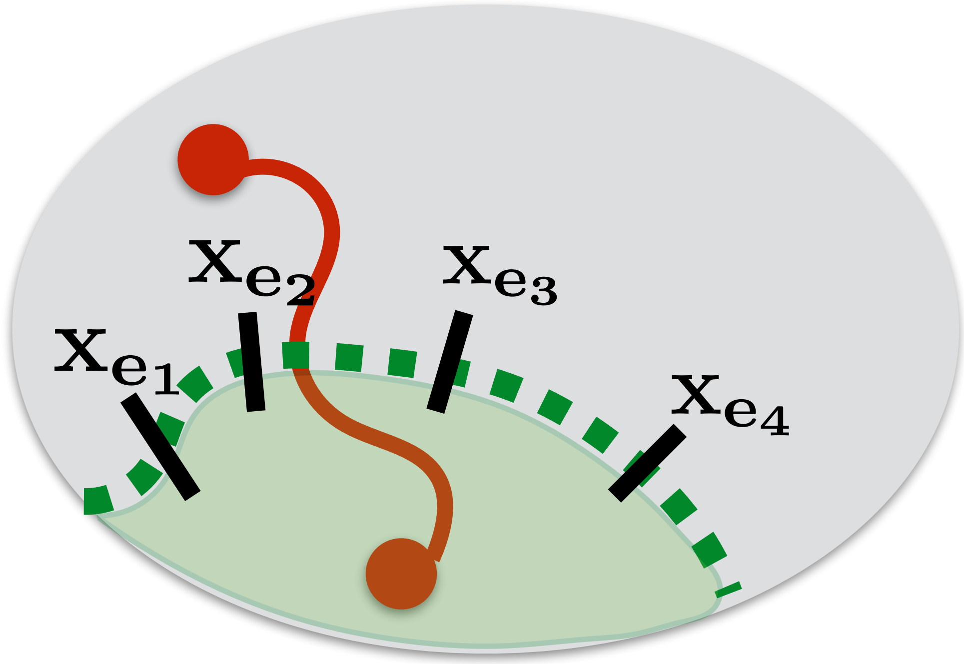
Do we ever get stuck?

Do we ever get stuck?
Suppose we do

cut: x not feasible?

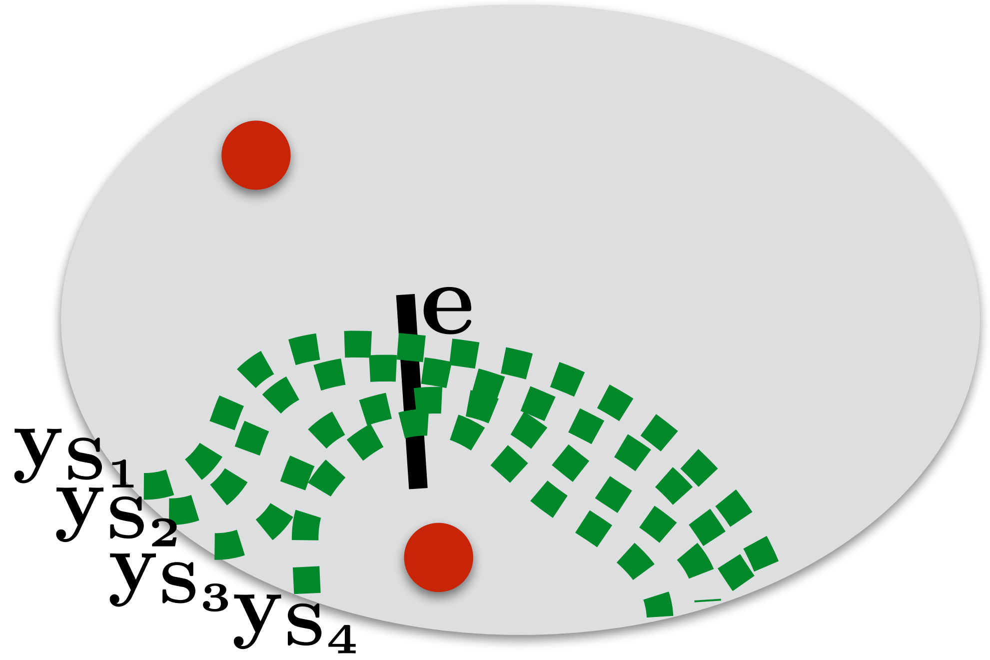


If we cannot raise y(S)
it's because
some e in the cut is tight
and then we would have put x(e)
in solution
QED



Initialization:

$$x \leftarrow 0, y \leftarrow 0$$



Iteration: while x not satisfiable

in parallel, raise every unfrozen y_S with minimal S

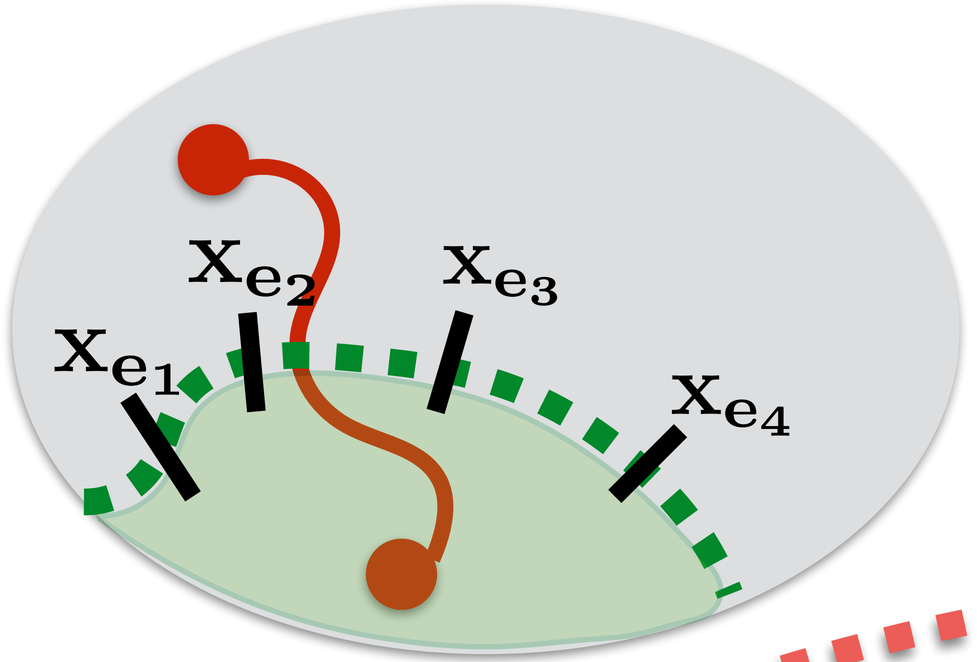
stopped by tight constraint (e)

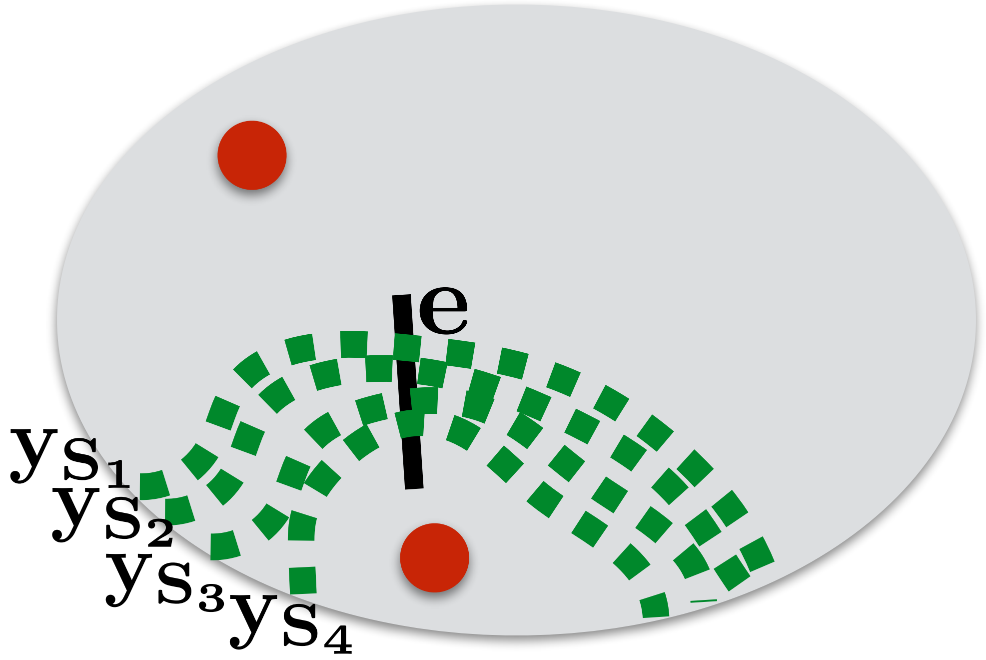
$$x_e \leftarrow 1$$

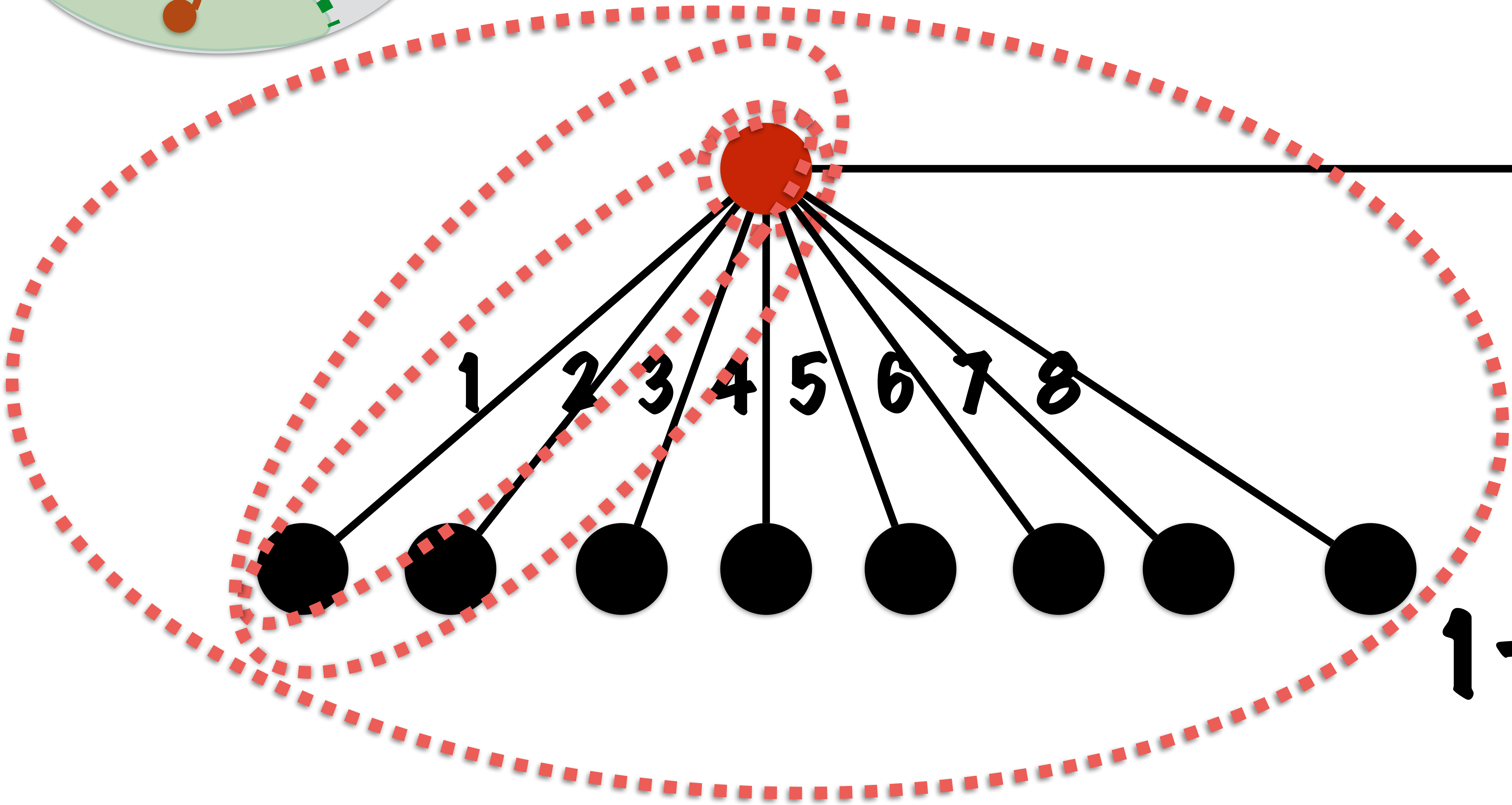
freeze y_S in tight constraints

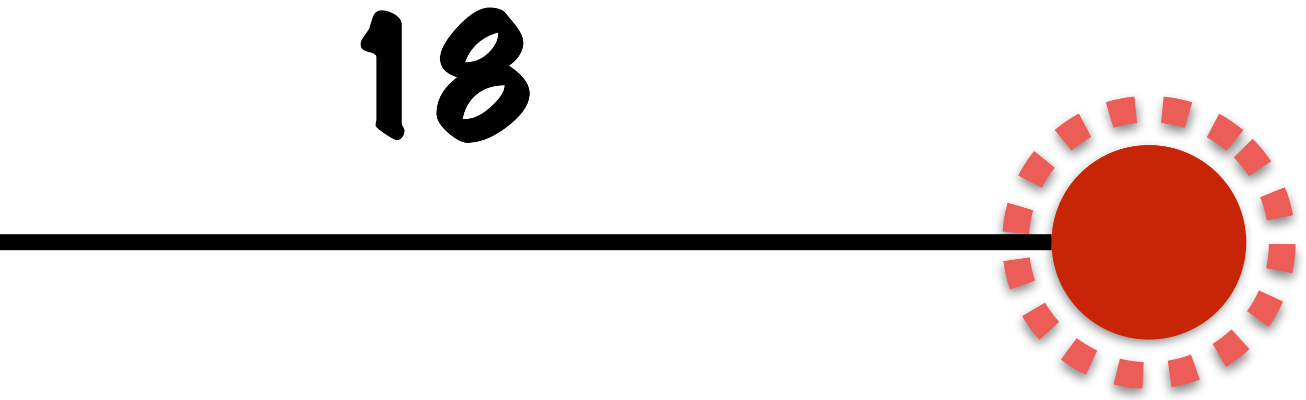
Fact: final x,y are feasible.

What about output cost?









Output cost

1+2+...+8+18

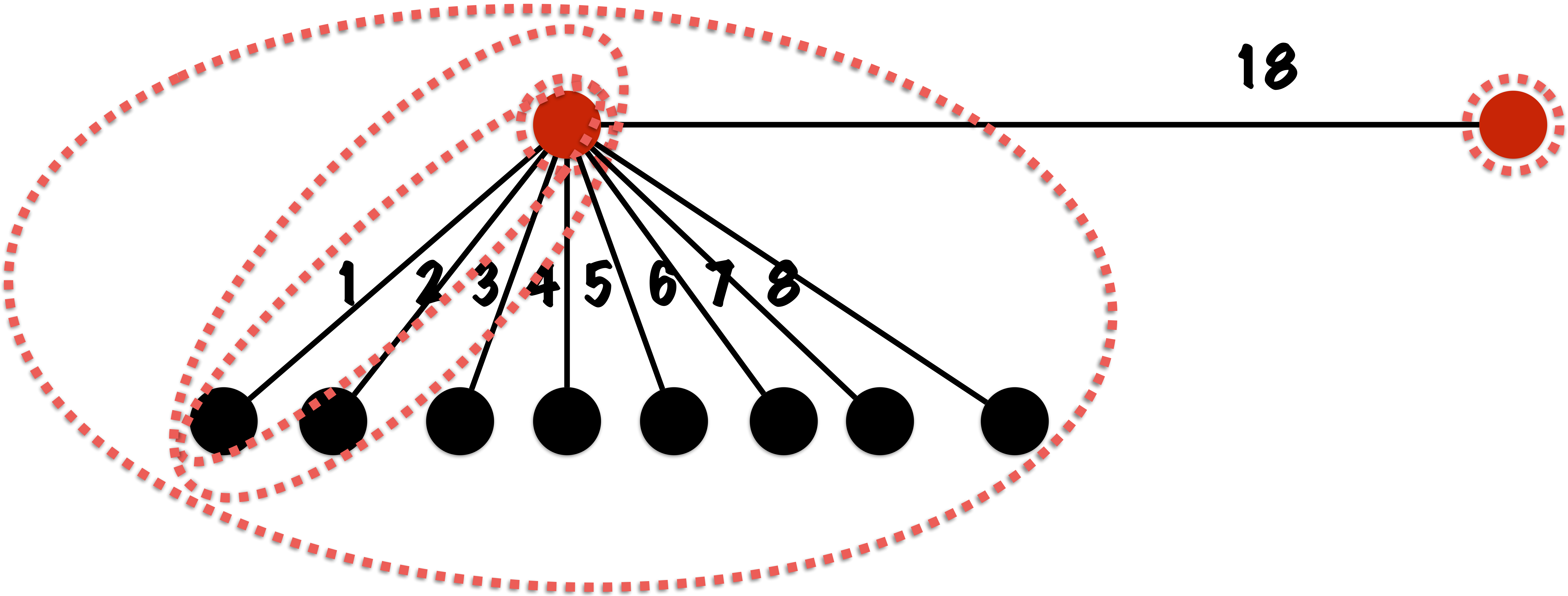
dual value

1+1+...+1+1+9=18

OPT

18

Bad!

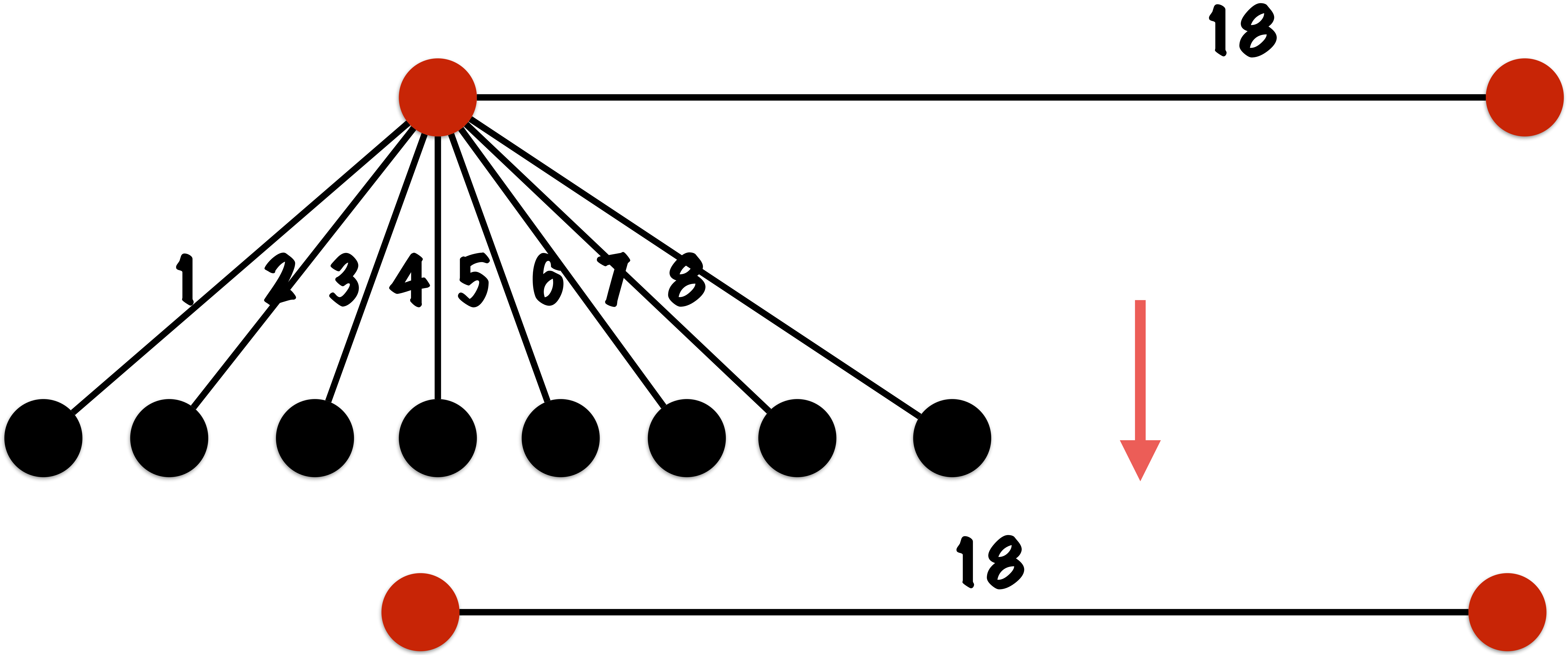


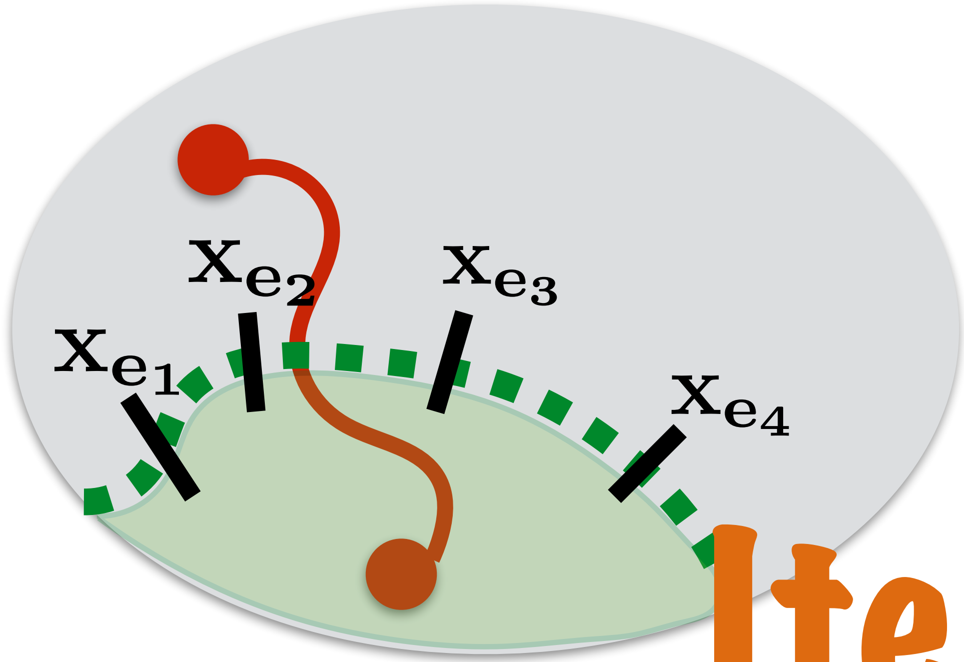
Observe: Many of the edges in the output are useless

Idea: prune useless edges

Modified algorithm

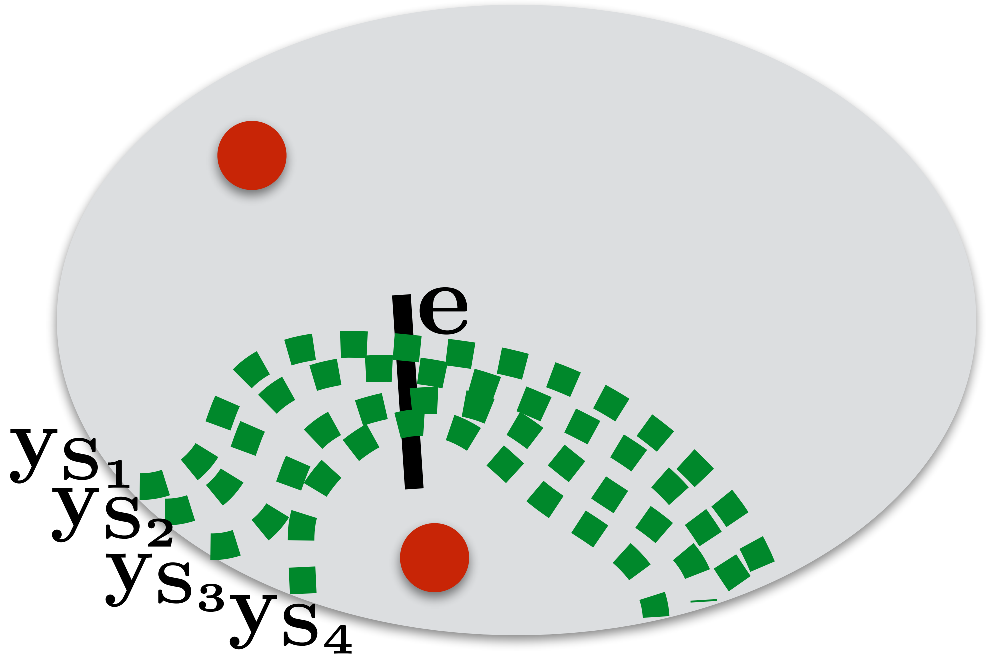
Consider set of edges defined by x
remove unnecessary edges
Output resulting set





Initialization:

$$x \leftarrow 0, y \leftarrow 0$$



Iteration: while x not satisfiable

in parallel, raise every unfrozen y_S with S minimal

stopped by tight constraint (e)

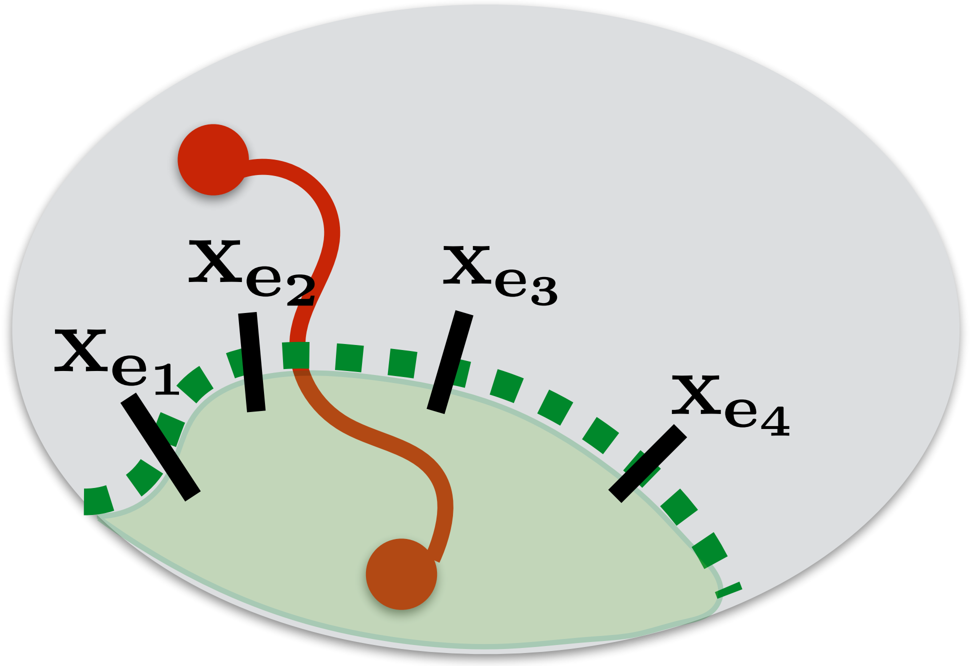
$$x_e \leftarrow 1$$

freeze y_S in tight constraints

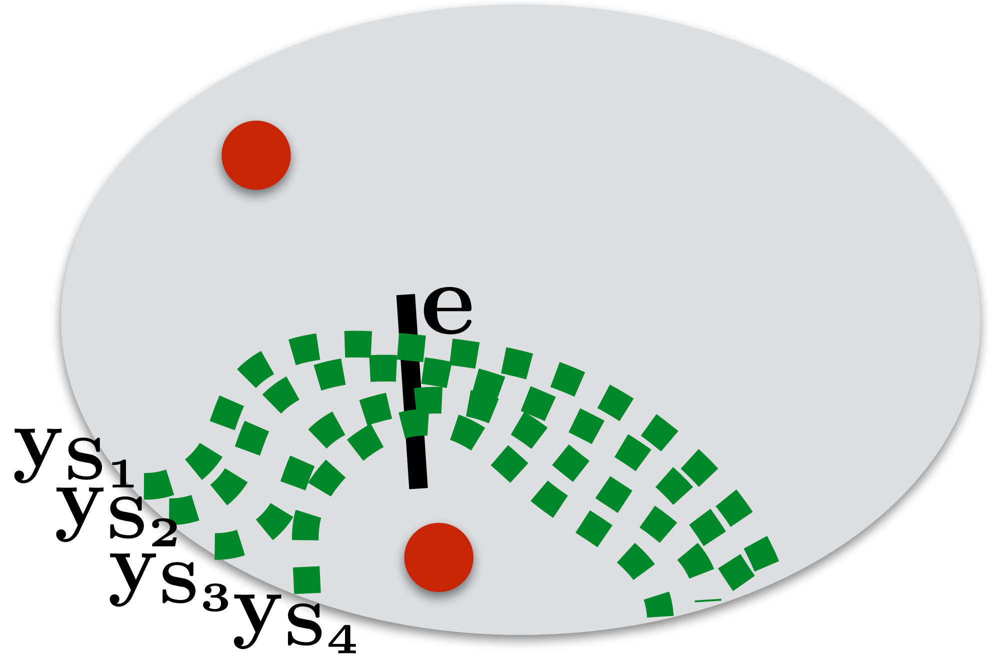
Pruning: let F={edges defined by x}

for each edge e of F in reverse order

remove e if unnecessary



Theorem



It's a 2-approximation for Steiner forest

Steiner forest