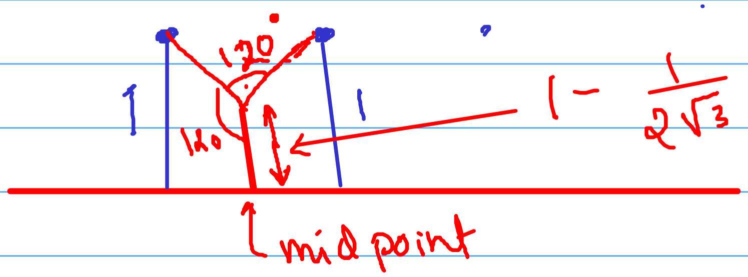
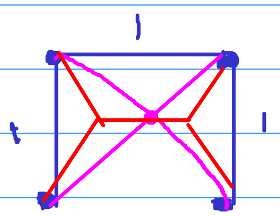
Feb 15 Deadline for assign 1 Feb 17

## Water Pipeline network.



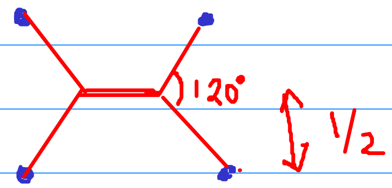

$$\frac{\sqrt{3}}{2}+1 = 1.86$$

## Road Network

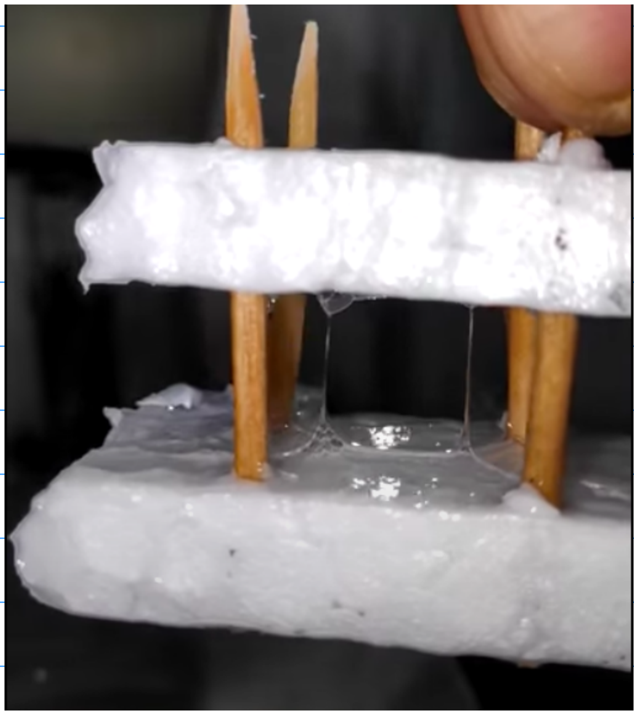



$3 > 2\sqrt{2}$

$2.82$




$1+\sqrt{3} = 2.73$

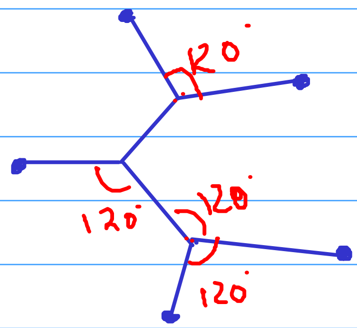


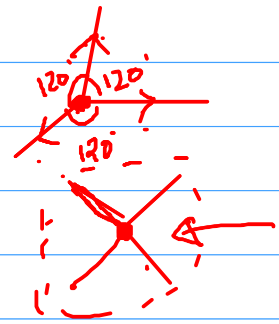
Soap film tries to minimize the total surface area while connecting all the terminals.

[Local optimal

# Eulidean Steiner Tree

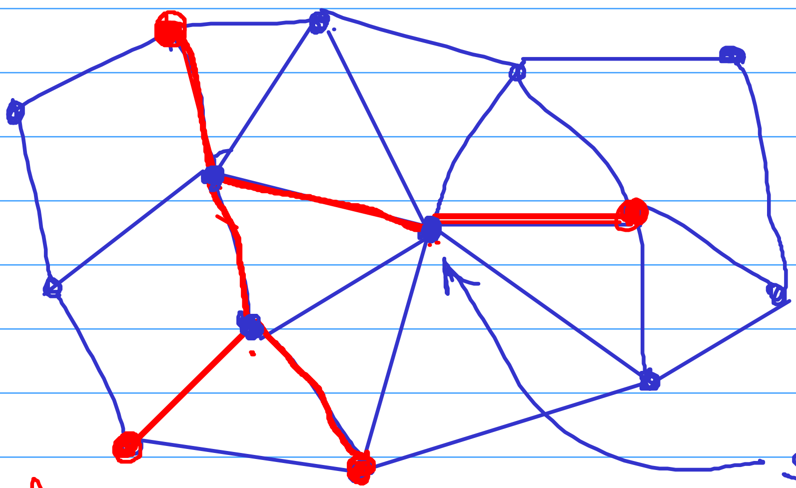
NP-hard

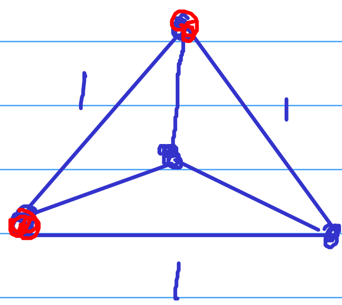


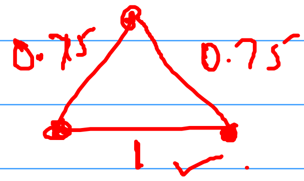




Given a set of point in 2-d space, connect all of them with minimum total length

## Steiner Tree on graphs






steiner vertices.

NP-hard.

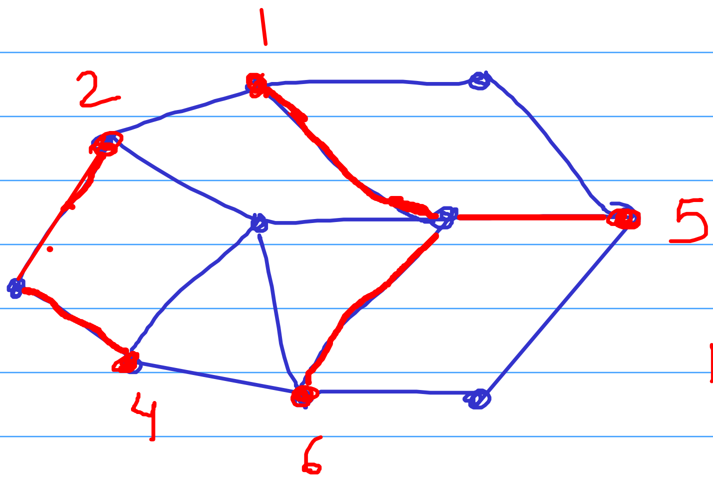
Given a graph with edge weights and a set of terminals T, find the minimum weight tree that spans T.

# Steiner Forest in graphs.

NP-hard.

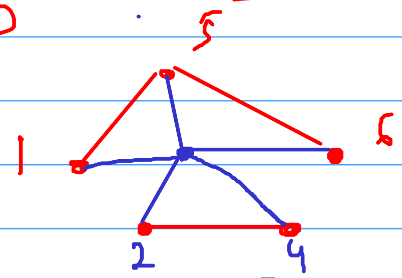
Given pairs of terminals, we want minimum weight subgraph connecting all pairs.

Example (1, 5) (1, 6) (2, 4)




{1, 5, 6} ✓

{2, 4} ✓




$T_1, T_2, \ldots, T_k$

Generalizations of

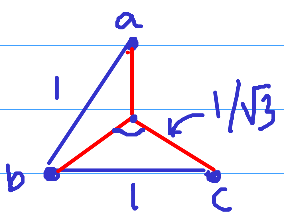
(1) Shortest Path (2) MST.

Feb 16

## Approximation Algorithms

### Euclidean

Make a graph with Euclidean distances and find MST




$\sqrt{3}$ — OPT

2 — Algo

Conj: Always gives an approx factor $\leq 2/\sqrt{3}$ $\approx 1.15$

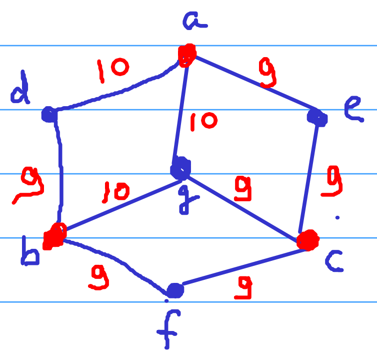
# Steiner tree on graphs.

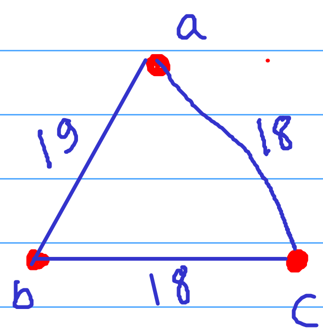
→ Construct a complete graph $G'$ on the terminal nodes with shortest path lengths as edge weights

→ Find MST $M$ on $G'$

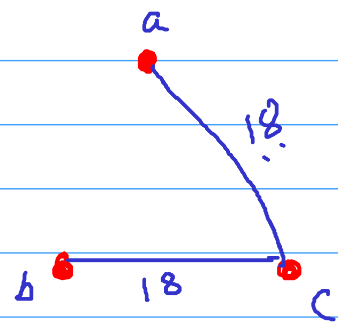
→ For each edge in $M$, replace with the corresponding shortest path in $G$.

→ Remove cycles, if any.



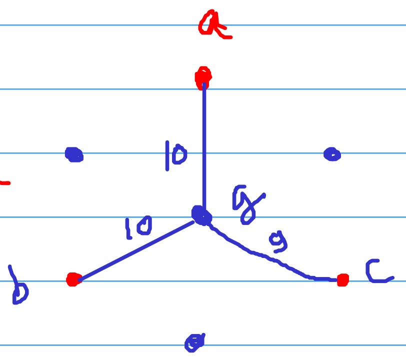




$\boxed{G'}$

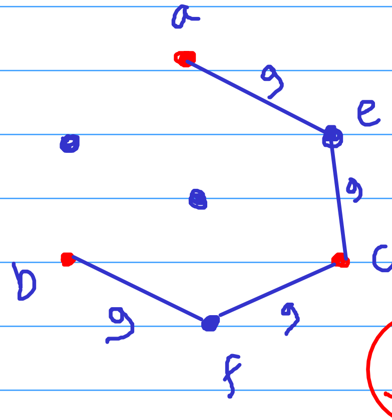



$\boxed{\text{MST in } G'}$

OPT




29




Steiner Tree in $G$

36

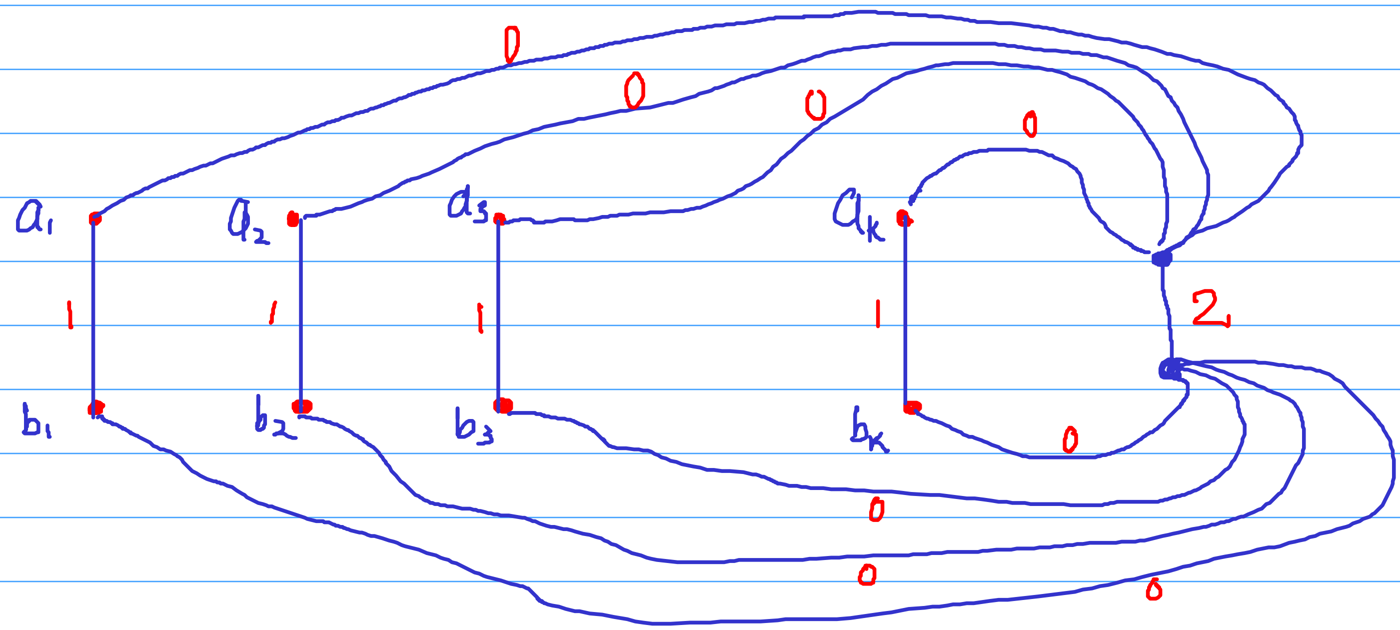
$\simeq 2(k-1)\cdot 10$ $\qquad$ $10k$ (opt)

In general approx factor can go close to 2.

Similar example with $k$ terminal vertices and one center vertex

There is a proof that the approx factor is at <u>most 2</u> on all graphs. (Extra Reading/Thinking)

Similar idea doesn't work for Steiner forest




Terminal pairs $(a_1, b_1)$ $(a_2, b_2)$ ... $(a_k, b_k)$

OPT = 1 Algorithm $\Rightarrow$ k

Towards 2-approx algorithm for steiner Tree/Forest

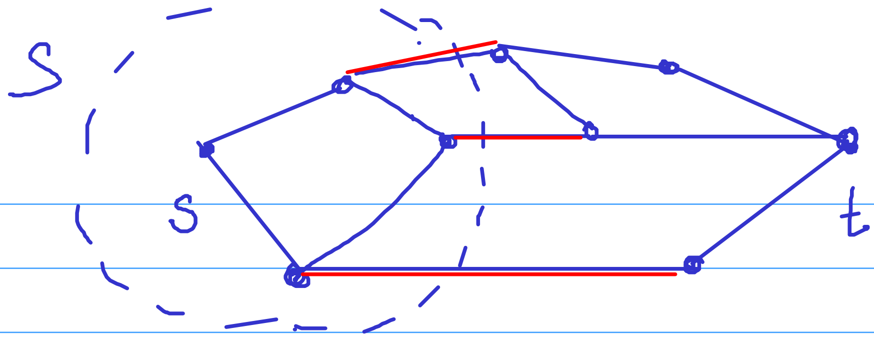
Integer Program for Steiner tree.

$G(V, E)$ $T \subseteq V$ set of terminals.

for $e \in E$ $x_e \in \{0, 1\}$

$$\text{Min} \sum_{e \in E} w_e x_e$$

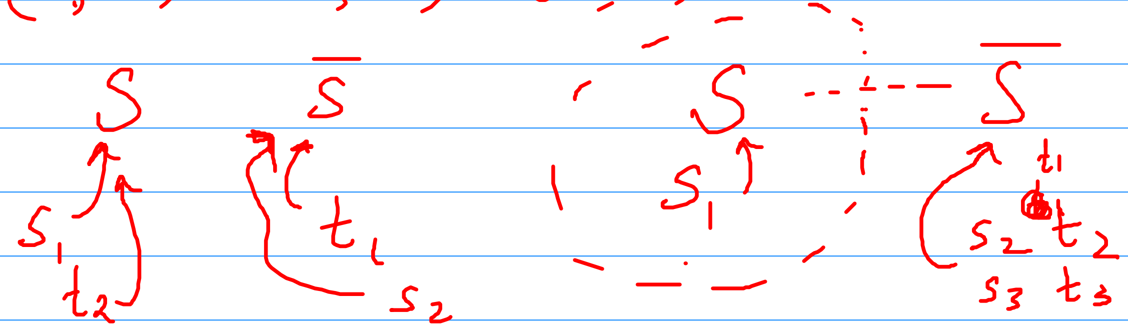
Constraints?




Take any $S \subseteq V$ s.t. $s \in S$ $t \notin S$

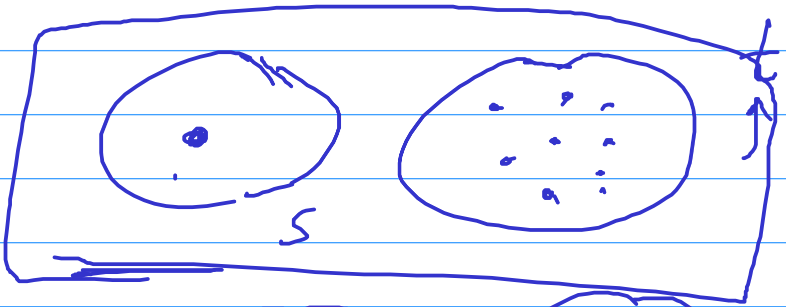
$$\left.\sum_{e \in \delta(S,\bar{S})} x_e \geq 1 \right\} \text{ for every } S \subseteq V$$

$\rightarrow$ $(s_1, t_1)$ $(s_2, t_2)$ $(s_3, t_3)$

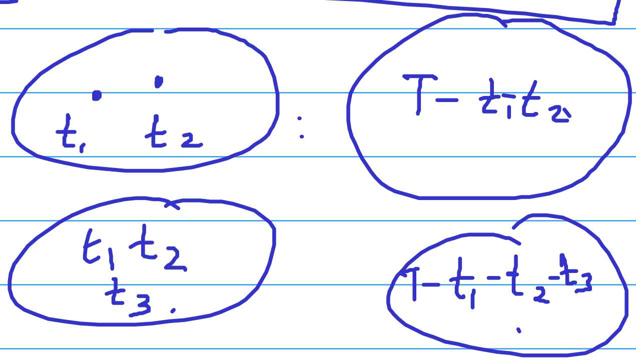


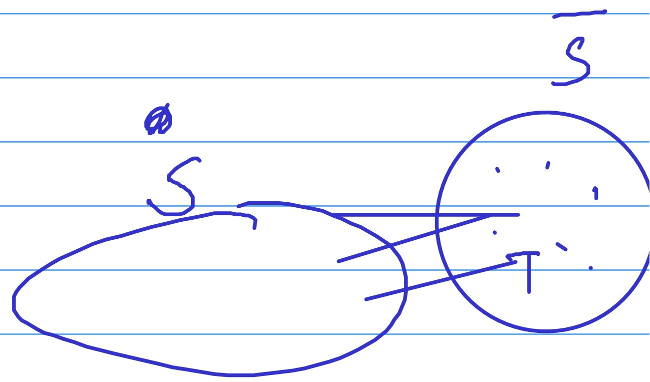

$\rightarrow$ T

$\forall$ $S \subseteq V$ s.t. $\exists$ $t \in T$




$t \in S$ $T \setminus t \subseteq \bar{S}$




$$\sum_{e \in (S,\bar{S})} x_e \geq 1$$

Feb 18

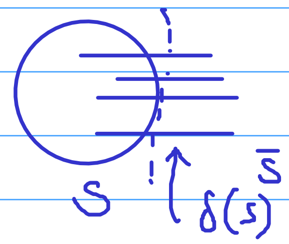
# Integer Program for steiner tree

$G(V, E)$ $T \subseteq V$ set of terminals.

$x_e \in \{0,1\}$ for $e \in E$

$$\min \sum_{e \in E} w_e x_e$$

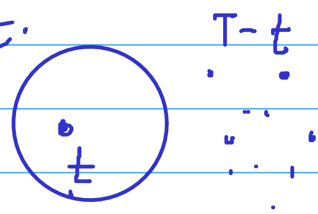
Constraints:




Option 1 for every $t \in T$

consider all sets $S \subseteq V$ s.t.

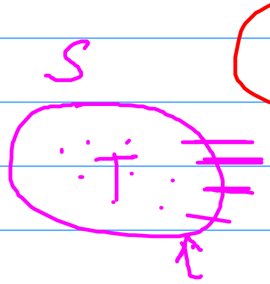
$t \in S$ $T - t \subseteq \bar{S}$

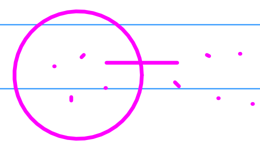



and write $\displaystyle\sum_{e \in \delta(S)} x_e \geq 1$

Problem

$T = (t_1, t_2, t_3, t_4)$

Option 2 all sets $S \subseteq V$ s.t.

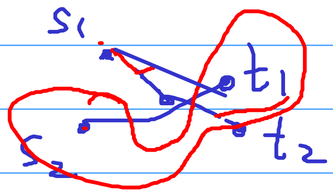
$T$ is split by $S$ i.e.,

Both $T \cap S$ and $T \cap \bar{S}$ are non-empty.

**Def:** $S \subseteq V$ is an interesting cut w.r.t. $T$ if both $T \cap S$ and $T \cap \bar{S}$ are non-empty.

**Def:** $S \subseteq V$ is an interesting cut w.r.t.

$(s_1, t_1)$ $(s_2, t_2)$ $\ldots$ $(s_k, t_k)$




if for some $1 \le i \le k$

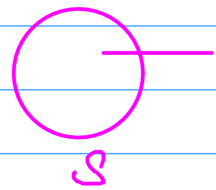
$S = \{s_i\}$ interesting

$s_i \in S$ and $t_i \in \bar{S}$

**Claim:** Let $F \subseteq E$ be a set of edges such that for every interesting cut $S \subseteq V$,

HW

we have $|F \cap \delta(S)| \geq 1$




Then in the graph $(V, F)$,

all vertices of $T$ are connected with each other.

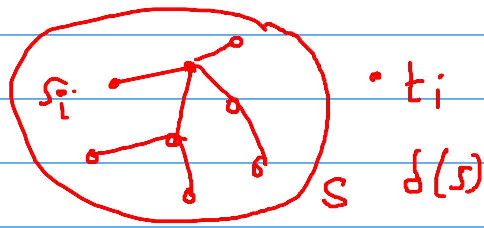
**Claim** Given pairs $(s_1, t_1)$ $(s_2, t_2)$ $\ldots$ $(s_k, t_k)$

Let $F \subseteq E$ be s.t. for every

interesting cut $S$ with respect to terminal pairs

HW

$|F \cap \delta(S)| \geq 1$




Then in graph $(V, F)$

$s_i$ is connected to $t_i$ for each $i$.

$$\boxed{w_e \geq 0} \text{ assumption}$$

$$\boxed{\text{LP-OPT} \geq \alpha \cdot \text{OPT}}$$

**LP**

$$\min \sum_{e \in E} w_e x_e$$

S

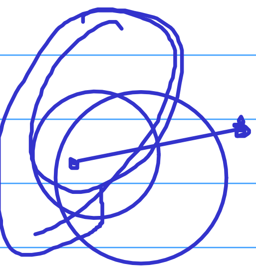
$$x_e \geq 0 \quad \text{for } e \in E$$

$$\sum_{e \in \delta(S)} 1 \cdot x_e \geq 1 \quad \text{for interesting cuts } S \subseteq V$$

**Dual LP**

$y_S$ for each interesting cut $S \subseteq V$

$$y_S \geq 0$$



$$\text{Max} \sum_S y_S$$

$$\sum_{\substack{S \\ \text{s.t. } e \in \delta(S)}} 1 \cdot y_S \leq w_e \quad \text{for } e \in E$$

## Complementary Slackness

(1) for each $e \in E$

Dual CS: either $x_e = 0$ or $\displaystyle\sum_{\substack{S \\ \text{s.t. } e \in \delta(S)}} y_S = w_e$ — easy to maintain

(2) for each $S$ interesting.

Primal CS: either $y_S = 0$ or $\displaystyle\sum_{e \in \delta(S)} x_e = 1$ — [Not easy to maintain]

$$y_S \neq 0 \Rightarrow \sum_{e \in \delta(S)} x_e = 1$$

Does there exist a steiner tree/forest and a dual feasible solution satisfying complementary slackness?

No. From the assigment, we know that

LP optimal solution may not come from Steiner Tree/Forest

In the primal dual schema, we will only maintain the Dual CS condition. We will ignore the primal CS, but at the end of the algorithm we will hope that primal CS is satisfied approximately.

Mar 4

## Primal - Dual Schema (High level picture)

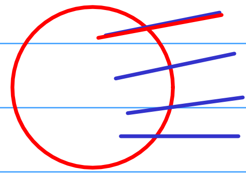
1. Start with a feasible dual solution.
   e.g., $y = 0$.
   And primal $x = 0$.

2. At any stage let $F \subseteq E$ be the set of tight edges, i.e., edges for which the dual constraints are tight.

We can set $x_e = 1$ only among the tight edges (Dual CS)

3. Does the set $\{e : x_e = 1\}$ give you a desired Steiner tree/forest?
   - If yes, stop
   - If No → Pick a dual variable to increase so that a new edge becomes tight.

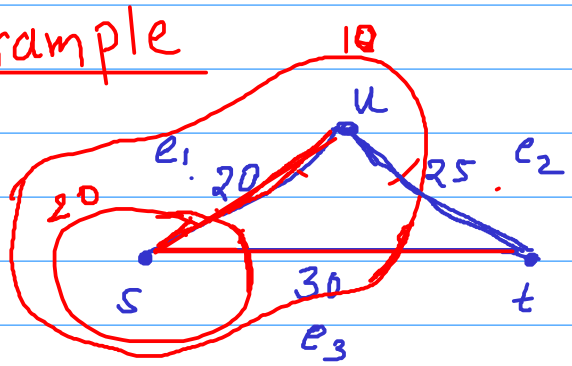
Which dual variables can be increased? $\delta(S)$




If $S \subseteq V$ is interesting cut s.t. a tight edge belongs to $\delta(S)$ then $y_S$ cannot be increased.

equivalently, $y_S$ can be increased only if all the edges in $\delta(S)$ are non-tight

Example




Terminal pair $(s,t)$

Interesting cuts $\{s\}$ $\{s,u\}$

LP

Min $20x_1 + 25x_2 + 30x_3$

s.t. $x_1, x_2, x_3 \geq 0$

$\{s\}$ $x_1 + x_3 \geq 1$

$\{s,u\}$ $x_2 + x_3 \geq 1$

Dual LP

Max $y_s + y_{su}$

$y_s, y_{su} \geq 0$

$\to y_s \leq 20$ $e_1$

$\to y_{su} \leq 25$ $e_2$

$\to y_s + y_{su} \leq 30$ $e_3$

Initialize $\quad y_s = 0 \quad y_{su} = 0$ No tight edges.

$x_1, x_2, x_3 = 0$.

Increase $y_s$ up to 20. $\quad y_s = 20$

Tight edges : $\{e_1\}$. Set : $x_1 = 1$.

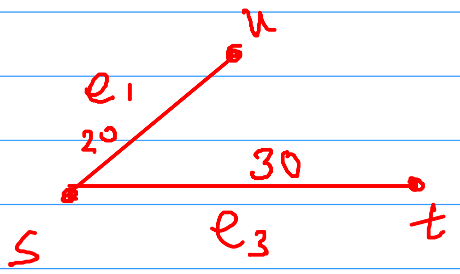
The set $\{e_1\}$ does not give a Steiner tree.

Increase $y_{su}$ up to 10

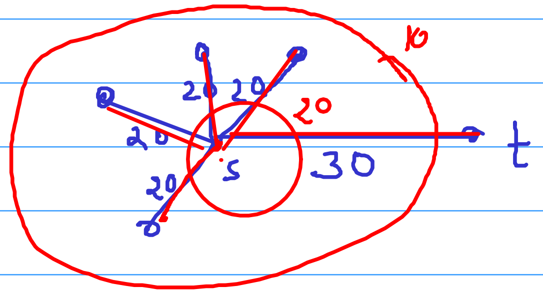
Tight edges : $\{e_1, e_3\}$. Set $x_3 = 1$

The set $\{e_1, e_3\}$ gives a Steiner tree (s-t path).

Stop.




Primal feasible Solution.




Idea 1 Pruning:

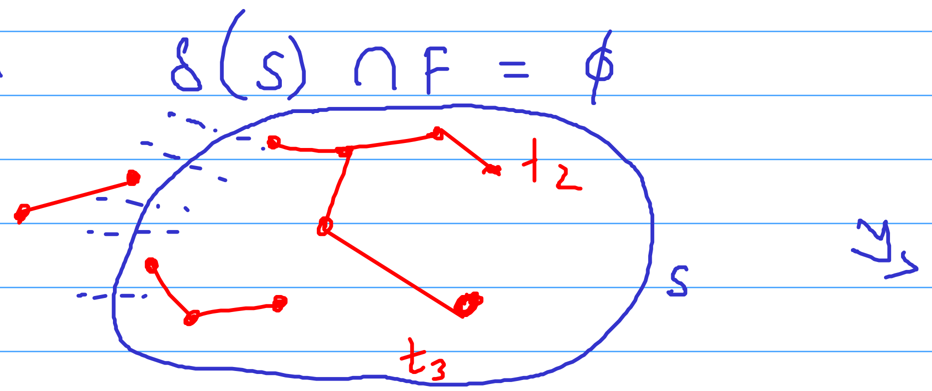
In the final feasible solution throw out edges which are not used in connecting terminal pairs.

# Mar 8

## How to find a dual variable to increase?

Let $F \subseteq E$ be the tight edges.
If $G(V,F)$ does not have a Steiner tree|forest
then there must be an interesting cut $S$

with $\delta(S) \cap F = \phi$

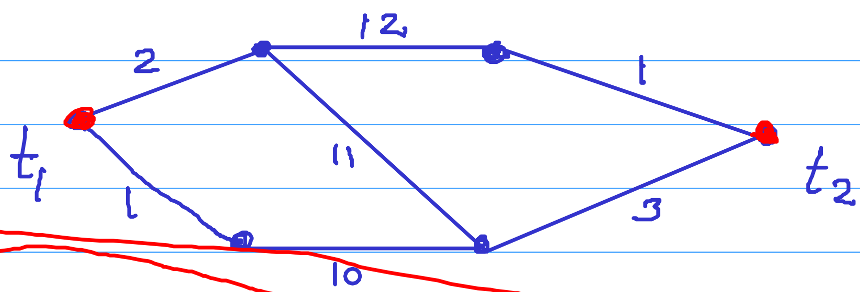


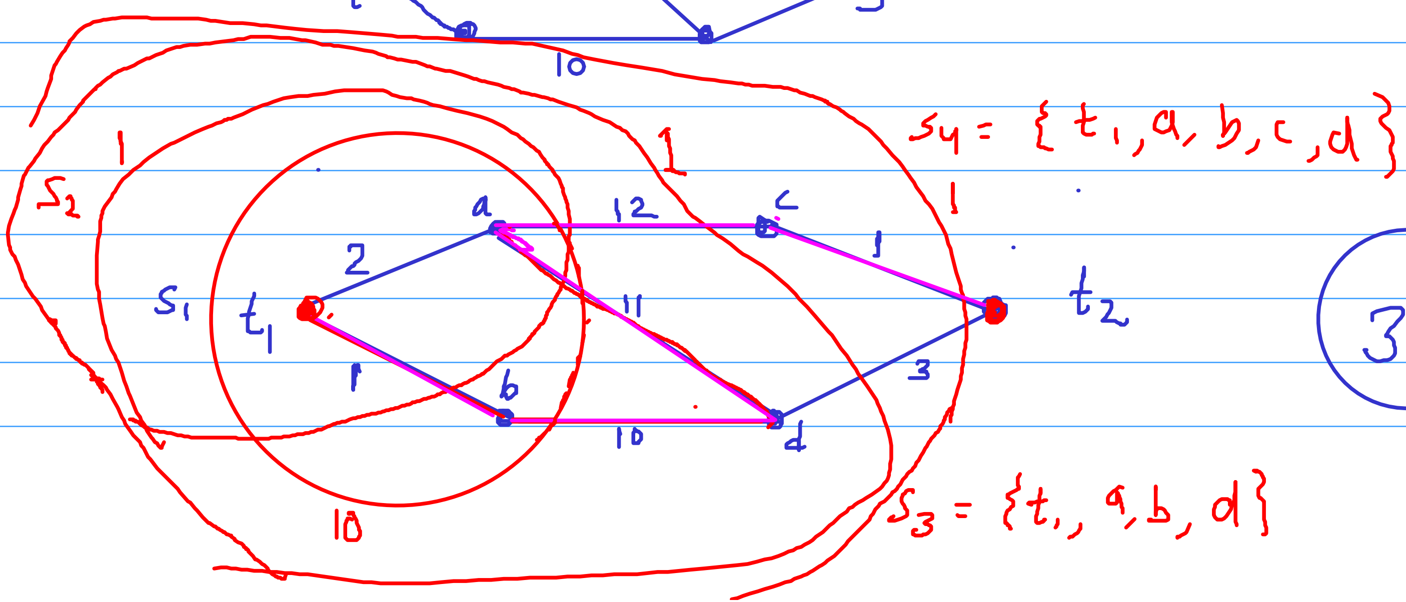

$y_s$

That is, all edges in $\delta(S)$ are non-tight.

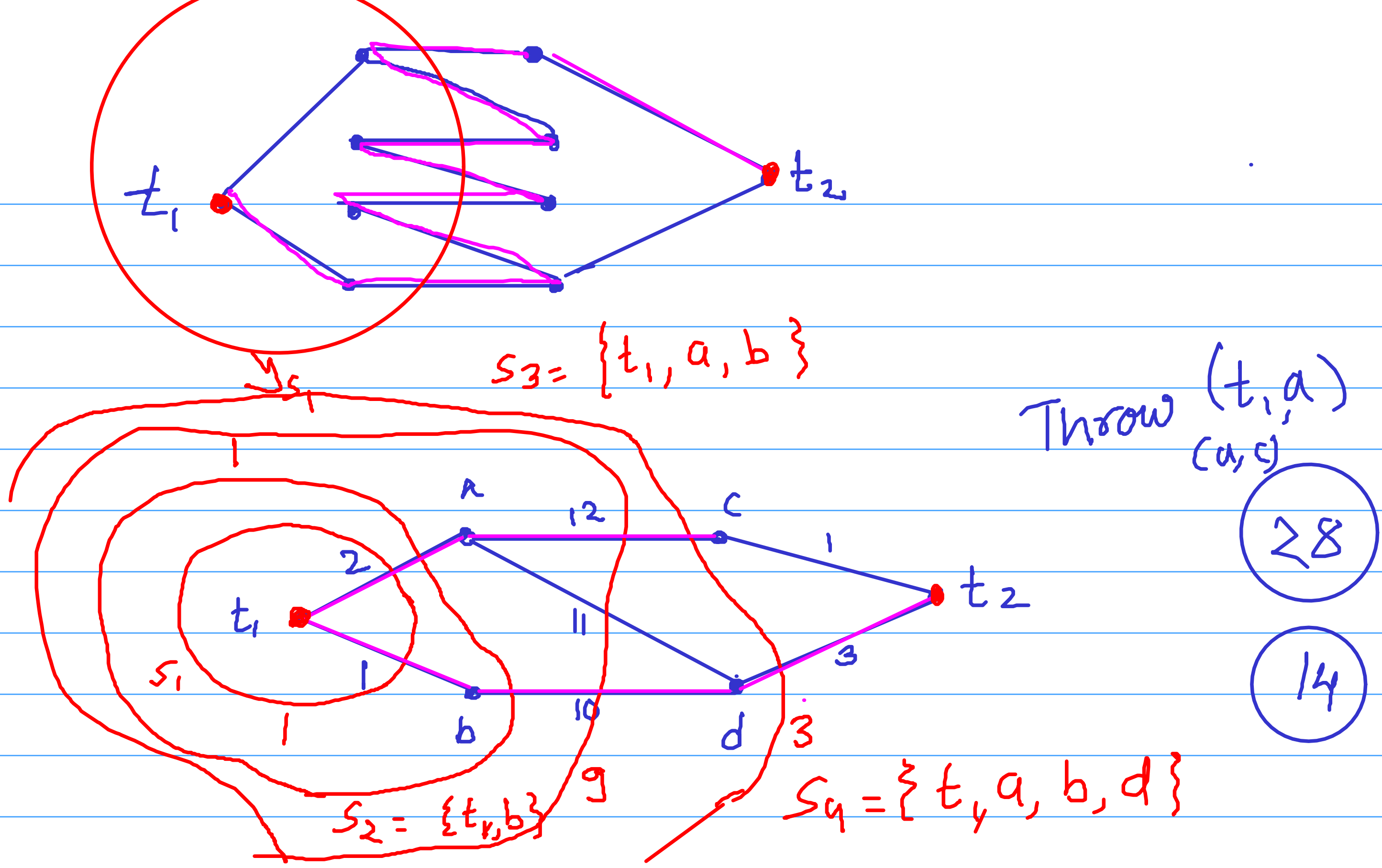
Hence, $y_S$ can be increased.

Can we choose an arbitrary dual variable to increase?



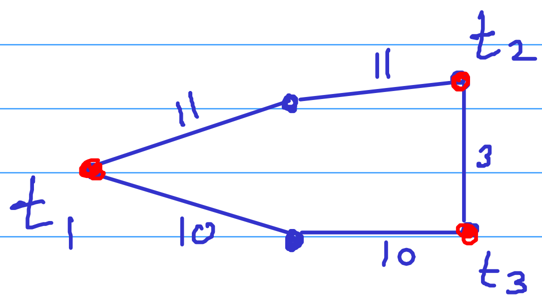

35



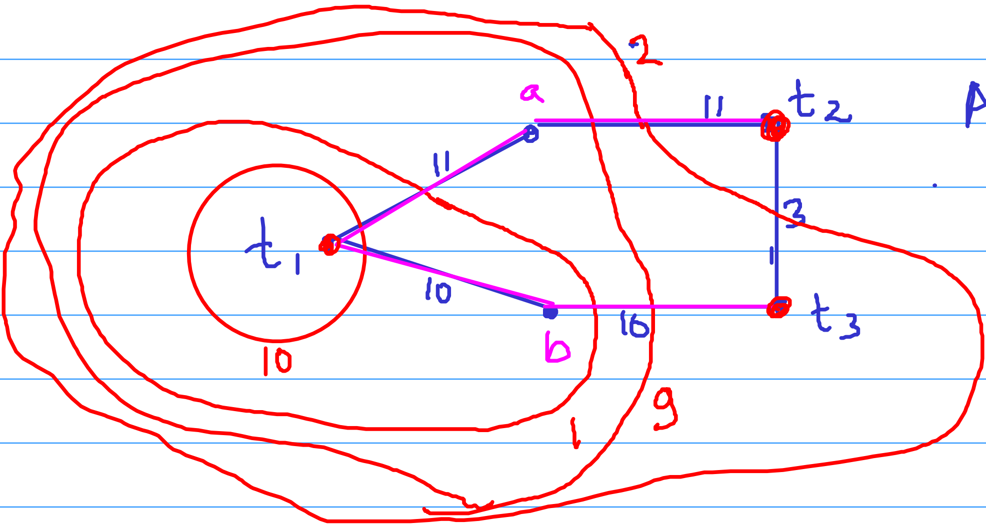

Idea 2: At any point consider the minimal set $S$ s.t. $\delta(S)$ edges are non-tight. And increase the dual variable $y_S$.

How to generalize this idea to more than two terminals?

Let $t_1, t_2, t_3$ be three terminals and we want a steiner tree connecting all three.



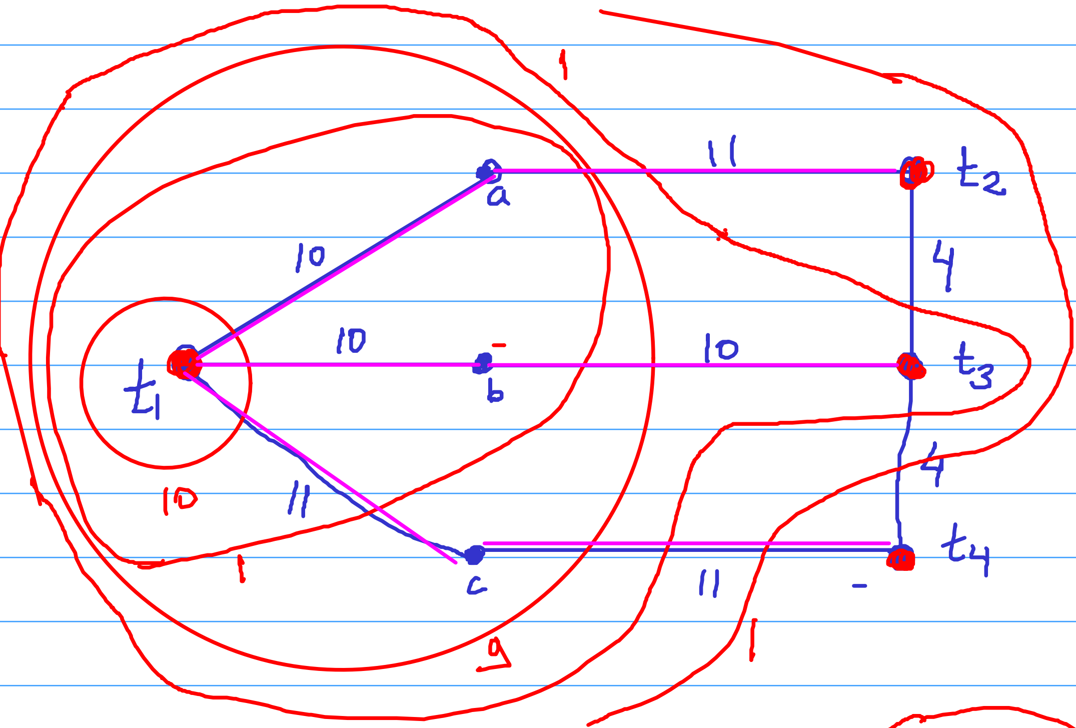
# Mar 9

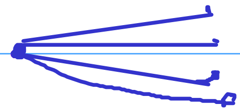



ALG → 42

OPT → 23

Not a bad Example.

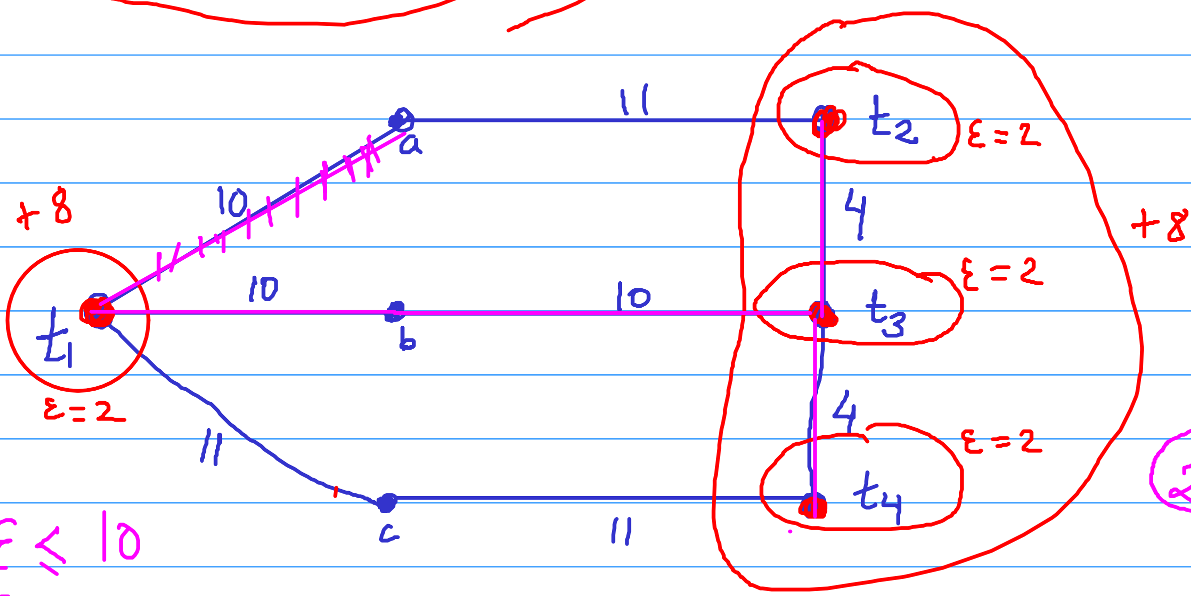


Bad Example

ALG = 63

OPT = 28




cost = 28

$2\varepsilon \leq 4$

$\varepsilon \leq 10$

$\varepsilon \leq 11$

$\varepsilon \leq 10$

$\varepsilon \leq 11$

**Idea 3** consider all inclusionwise minimal cuts that are interesting and their cut edges are non-tight

Increase all these dual variables simultaneously by the same amount.

## Final Algorithm:

Initialize $F = \phi$. Dual variables $= 0$.
F will be maintained as a forest.

At any stage consider the connected components in $(V, F)$

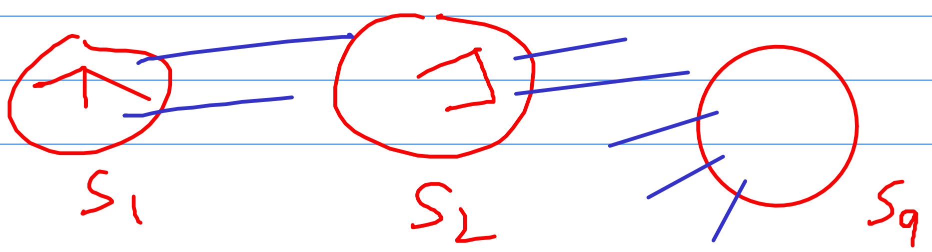
say, $S_1, S_2, \ldots, S_q$

All the tight edges are within these components.

Hence, $\delta(S_1), \delta(S_2), \ldots, \delta(S_q)$ have their cut-edges as non-tight.

Potentially, all the dual variables

$y_{S_1}, y_{S_2}, \ldots, y_{S_q}$ can be increased simultaneously.



# Define Active connected components

$C_i$ is said to be active if it separates some terminal pair, i.e., for some $j$, $s_j$ is inside but $t_j$ is not.

(otherwise $C_i$ is not an interesting cut and there is no dual variable $y_{s_i}$)

- Increase dual variables for all the active connected components simultaneously by the same amount.

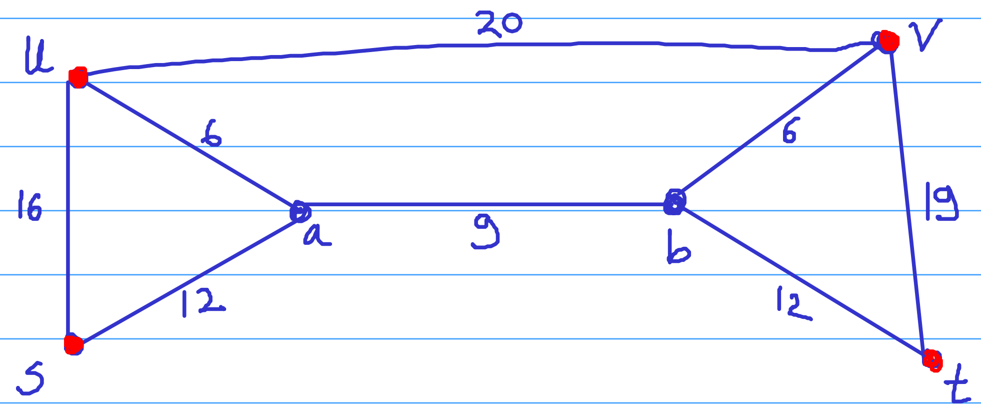
- Include the new tight edge in F.

- If there are more than one new tight edge then include one by one as long as we don't create cycles.

- Stop when all terminal pairs are connected

- Pruning: finally delete any edges from F that are not used in connecting terminal pairs.

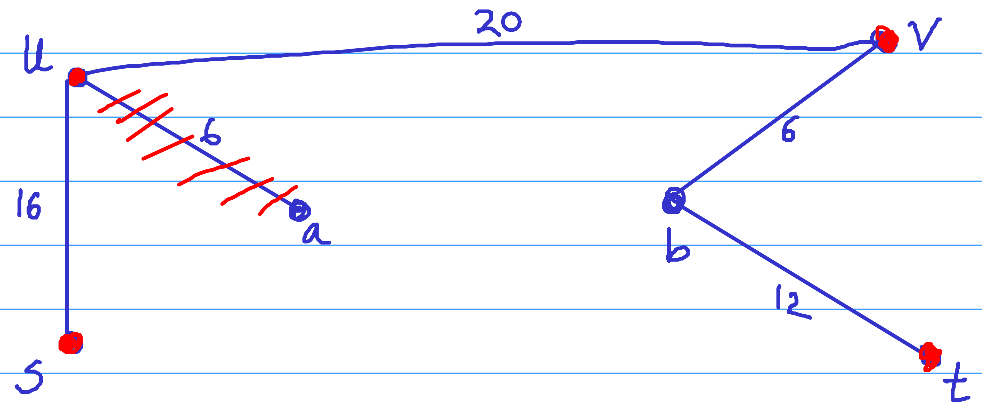
Example from Vazirani book.




Terminal Pairs $(s,t)$ $(u,v)$.

HW

ALG




## Analysis

Is the primal cs condition satisfied approximately?

$$y_s \neq 0 \implies \sum_{e \in \delta(s)} x_e = 1$$

can we say $\sum_{e \in \delta(s)} x_e \leq \gamma$ ?

for some const $\gamma$

This will imply $\gamma$ - approx solution.

Different analysis will show 2-approx.

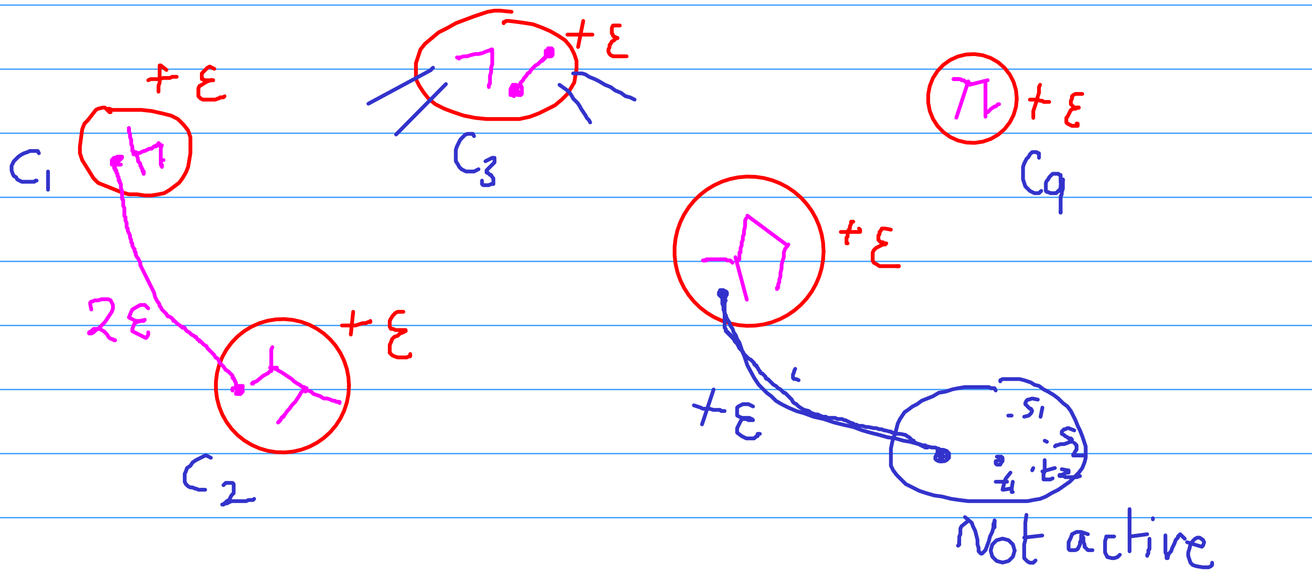
At any stage compare the changes in the primal objective value and the dual objective value.

Let $F$ be the final set of edges after Pruning

Primal objective $P = \sum_{e \in F} w_e = \sum_{e \in F} \sum_{S:\, e \in \delta(S)} y_S$

Dual objective $D = \sum_{S} y_S$

At any stage say there are $q$ active componets



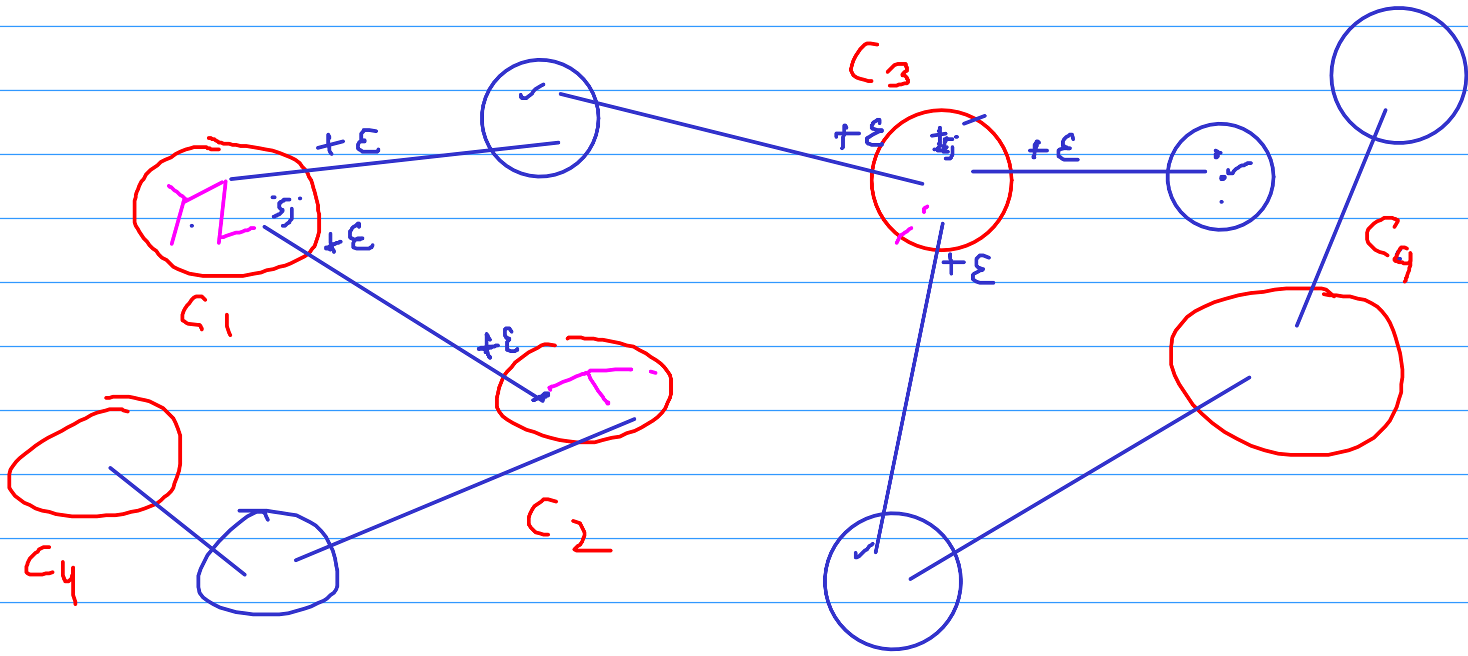

Increase in $D = \underline{q\varepsilon}$

Increase in $P = \sum_{e \in F} \sum_{S:\, e \in \delta(S)} \text{change in } y_S$

The only dual variables updated in this stage are $y_{C_1}, y_{C_2}, \ldots, y_{C_q}$

Thus the only edges which get affected are

$$\{ e \mid e \in \delta(C_j) \text{ for some } 1 \le j \le q \}$$



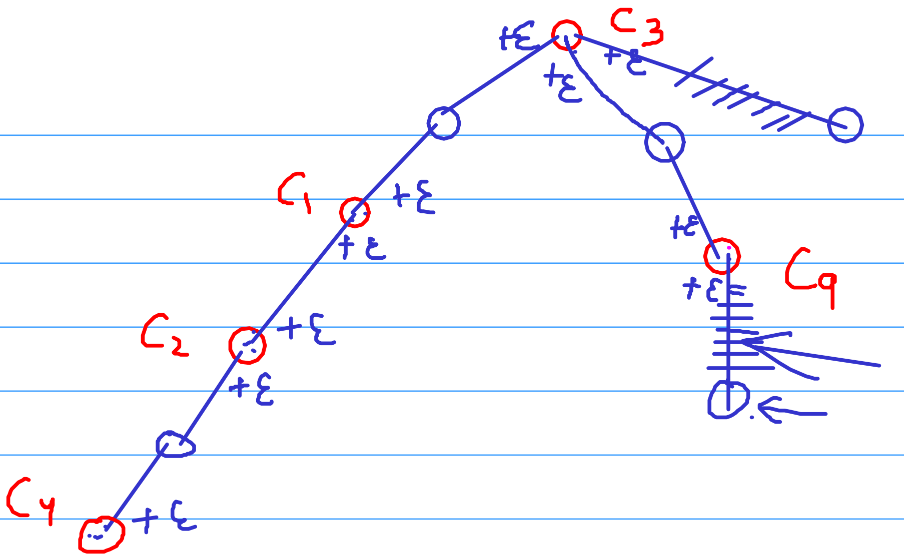

If $e$ is inside an active component or an inactive component then no change.

If $e$ has one endpoint in $C_i$ and the other end point in $C_j$ then it sees $+2\varepsilon$

If $e$ has one endpoint in $C_j$ but other end point in an inactive component the it sees $+\varepsilon$.

Note that if an edge doesn't lie on a path connecting two active components then it will be pruned out.

Rooted tree at some $C_j$

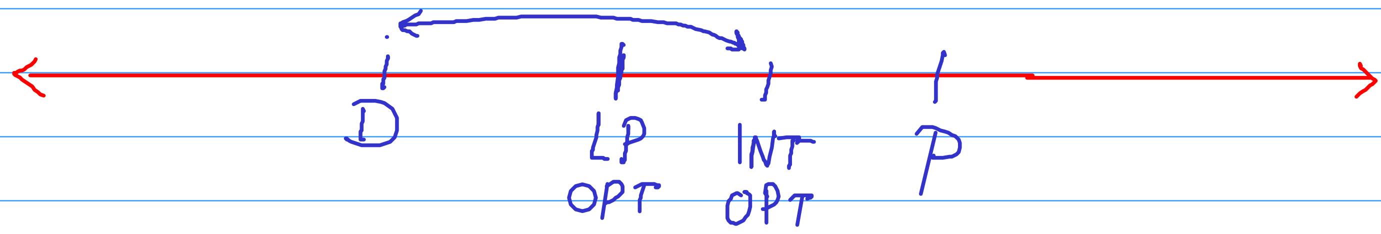



Counting: between any $C_i$ and it's nearest ancestor active node the edges will see a total increment of $+2\varepsilon$.

Total change in $P \leq (q-1) \cdot 2\varepsilon$.

$\Delta P \leq 2 \Delta D$.

At the end: $\underline{P \leq 2D} \leq 2 \cdot$ Min W ST/F.




(Primal = Dual) (Min Weight Steiner Tree/Forest)

What if we skip Pruning? Will the analysis hold?

If $T = V$ then will the algorithm give minimum spanning Tree or only a 2-approximation?

Prove that when we have only one terminal pair then the algorithm will give the exact optimal solution.

<u>Exercise</u>

Survivable Network design Problem.

Given pairs of vertices $(s_1, t_1), (s_2, t_2) \cdots (s_k, t_k)$ and numbers $n_1, n_2, \ldots, n_k$, find a minimum weight <u>subgraph $H$</u>, where for each $i$, there are at least $n_i$ edge-disjoint paths between $s_i$ and $t_i$.