LINEAR OPTIMIZATION

LECTURE 19

Recall: **PRIMAL - DUAL ALGORITHMS : a generic template**

Optimization problem

↓

Frame it as an ILP

↓

Generate an LP from the ILP, and write its dual

We now have a primal - dual pair. Let us say they are in the following form (this is not necessary, we choose this for illustration)

Primal

min $c^T x$

subj. to $Ax \leq b$

$x \geq 0$

Dual

max $b^T y$

subj. to $A^T y \leq c$

$y \geq 0$

↳ How do we find the optimum combinatorially?

PRIMAL - DUAL ALGORITHMS : a generic template (contd.)

Initialization: Find a feasible soln. y_0 of the dual. Typically for the problems under consideration, $c \geq 0$ and hence $y=0$ will be feasible.

Iterative step: After the i^{th} step, say we have a feasible soln. y_i.

Let $(A^T)'$ denote the rows of dual that are tight at y_i:

$$(A^T)' y_i = c'$$

c' is c restricted to rows of $(A^T)'$.

- If possible, find a $\bar{y}$ s.t.

adding any multiple of $\bar{y}$ to y keeps $(A^T)'(y + \epsilon\bar{y}) \leq 0$ →

$$(A^T)' \bar{y} \leq 0 \quad \text{and} \quad b^T\bar{y} > 0$$

cost at $\bar{y} > 0$

- and find an ϵ so that:

$$A^T(y_i + \epsilon\bar{y}) \leq c$$

$(A^{T})'(y_i + \epsilon\bar{y})$
$= (A^T)' y_i + \epsilon(A^T)'\bar{y}$
$= c' + \leq 0$

(we find the maximum such ϵ)

$$\text{Set: } y_{i+1} = y_i + \epsilon\bar{y}$$

Termination: When the iterative step cannot be performed anymore.

Optimality: From the final y that is obtained at the end of the algorithm, generate a primal solution x and use either complementary slackness or strong duality to show that x and y are optima of primal & dual resp.

PRIMAL - DUAL ALGORITHM FOR MINIMUM SPANNING TREES

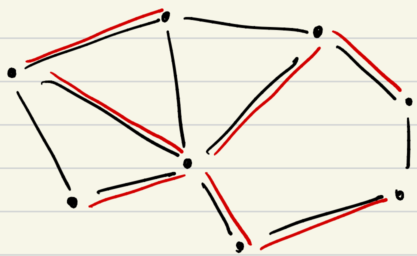
Minimum spanning tree problem (MST):

Given an undirected graph $G = (V, E)$ with positive costs on edges $c : E \longmapsto \mathbb{N}_{>0}$

Assume that:
- G is connected
- different edges have different costs

Find a spanning tree of G with minimum cost.

Spanning tree: - a subset of edges which forms a tree and which covers every vertex (ie, every vertex has one of the edges incident on it in the subset)



(weights not illustrated)

Today's goal: Design a primal-dual algorithm for MST.

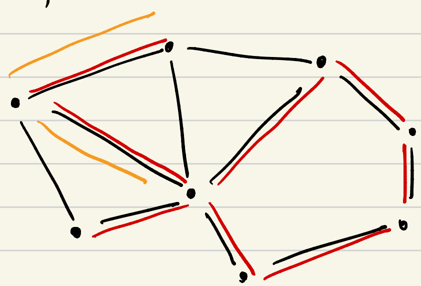
Reference: Lectures 20 and 21 from Prof. Sundar's notes.

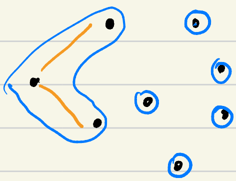
Part 1: Some graph theoretic observations

We want to pick a subset of edges S from the graph that satisfies two conditions:

- (i) S forms a tree
- (ii) S covers every vertex

Our strategy would be to pick a subset S that forms one connected component.





But then, this connected component may contain cycles. This is where we will use the objective function. Since we have assumed $c_e > 0$ for all edges e, the minimum among all such subsets would contain no cycles.

Find a subset of edges which forms one connected component and has the minimum cost.

We will now look at a criterion for ensuring connectedness.

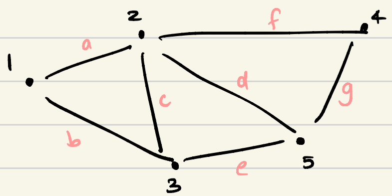
Theorem: A graph is connected

iff

for every partition π of its vertices, there are $\geq |\pi| - 1$ edges that cross the partition

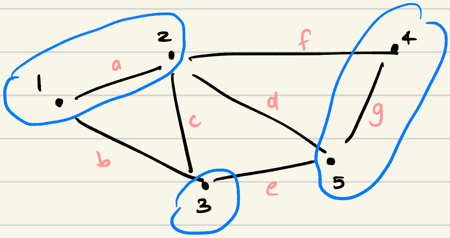
---

An edge crosses a partition if its endpoints are in different parts.

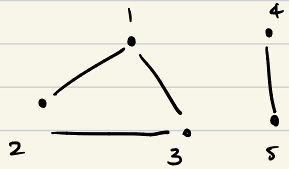


Partition π : $\{1, 2\}$ $\{3\}$ $\{4, 5\}$

Edges that cross π: $\{b, c, e, d, f\}$



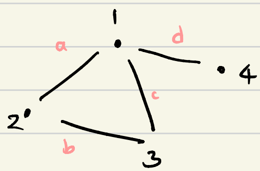
The proof of the above theorem is left as an exercise.
We will illustrate the theorem on some examples.



disconnected

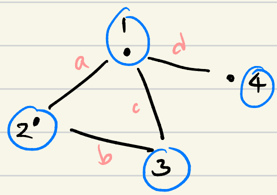
$\pi = \{1, 2, 3\} \quad \{4, 5\}$

No. of cross edges $= 0$

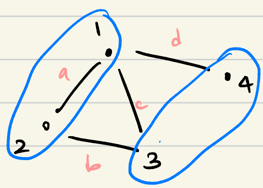


- connected.

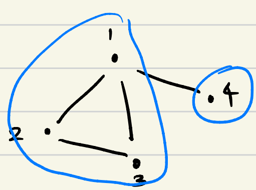
Here are some partitions:



cross edges : $4 \geq \overset{|\pi|-1}{3}$



cross edges : $3 \geq \overset{|\pi|-1}{1}$



cross edges : $1 \geq 1$

Part 2: An ILP for MST

We use the criterion that we described in Part 1.

Variables: x_e for each $e \in E$

Constraints: ensure that the subset picked satisfies the partition condition.

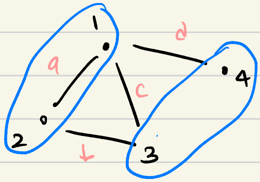
ILP: $\min \sum_{e \in E} c_e x_e$

subj. to:

$$\sum_{e \text{ crosses } \pi} x_e \geq |\pi| - 1 \quad \text{for every partition } \pi$$

$$\left.\begin{array}{l} x_e \geq 0 \\ x_e \leq 1 \\ x_e \text{ integral} \end{array}\right\} \forall e \in E$$

Notice that each partition gives rise to a constraint. For eg:



gives: $x_b + x_c + x_d \geq 1$

There are exponentially many constraints. We will not solve this LP directly.

Getting an LP from the ILP:

ILP:

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

subj. to:

$$\sum_{e \text{ crosses } \pi} x_e \geq |\pi| - 1 \quad \text{for every partition } \pi$$

$$\left.\begin{array}{l} x_e \geq 0 \\ x_e \leq 1 \\ x_e \text{ integral} \end{array}\right\} \forall e \in E$$

- Remove integrality constraint

LP

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

subj. to:

$$\sum_{e \text{ crosses } \pi} x_e \geq |\pi| - 1 \quad \text{for every partition } \pi$$

$$x_e \geq 0$$

$$x_e \leq 1$$

Further, we can remove the $x_e \leq 1$ constraint.

Exercise: Show that for every solution $\bar{x}$ of above LP, changing any x_e larger than 1 to 1 will give a feasible soln. with smaller cost.

Part 2: Primal and Dual

PRIMAL:

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

$$\sum_{e \text{ crosses } \pi} x_e \geq |\pi| - 1 \quad \forall \pi$$

$$x_e \geq 0$$

DUAL:

$$\max \sum_{\pi} (|\pi| - 1)\, y_\pi$$

$$\sum_{e \text{ crosses } \pi} y_\pi \leq c_e \quad \forall e$$

$$y_\pi \geq 0 \quad \forall \pi$$

a variable y_π for every partition π.

Part 3: Primal-dual algorithm: the iterative step.

As an initialization, we will set $y^0_\pi = 0$ for all partitions π.

For the iterative step, we work with a given feasible soln. y^i. Let E_i be the set of edges for which the dual constraints are tight for y^i.

We first need to find a $\bar{y}$ s.t.

$$\sum_{e \text{ crosses } \pi} \bar{y}_\pi \leq 0 \quad \forall e \in E_i \quad \text{and} \quad \sum_{\pi} (|\pi| - 1)\, \bar{y}_\pi > 0$$

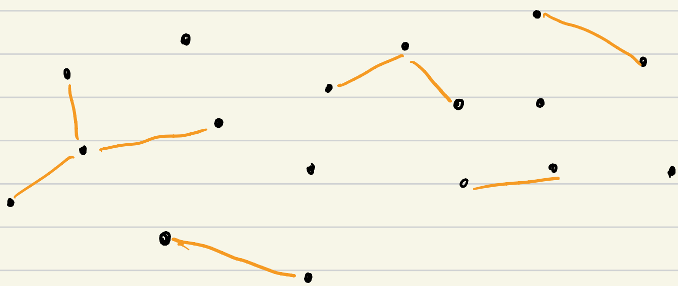
– For the iterative step, we work with a given feasible soln. y^i.
Let E_i be the set of edges for which the dual constraints are tight for y^i.

We first need to find a $\bar{y}$ s.t.

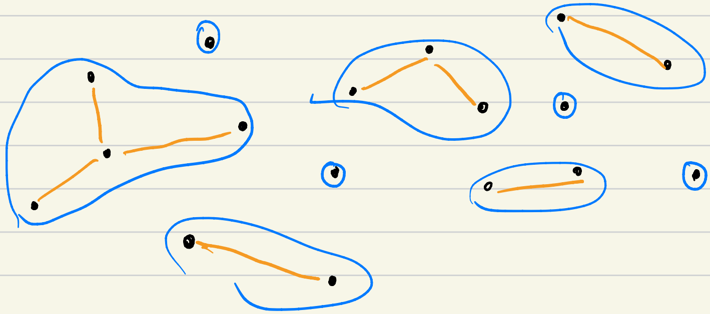
$$\sum_{e \text{ crosses } \pi} \bar{y}_\pi \leq 0 \quad \forall e \in E_i \quad \text{and} \quad \sum_{\pi} (|\pi|-1)\, \bar{y}_\pi > 0$$

– For simplicity, we will choose a $\bar{y}_\pi$ in which exactly one partition π will have 1 and others will have 0.

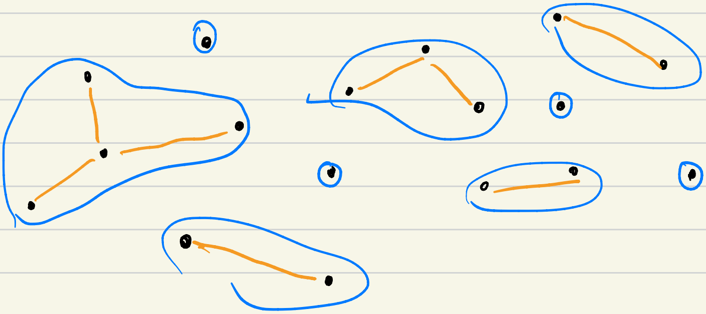
Consider the graph $G_i = (V, E_i)$



The edges E_i divide the vertices into connected components.



The edges E_i divide the vertices into connected components.



→ Let π_i be the partition given by the connected components in G_i.

Define $\bar{y}$ to be the vector with $\bar{y}_{\pi_i} = 1$ and

$\bar{y}_\pi = 0 \quad \forall \pi \neq \pi_i$

– When there are at least two components in G_i, we have

$$(|\pi_i| - 1)\,\bar{y}_{\pi_i} > 0$$ – satisfying the cost criterion.

– No edge $e \in E_i$ crosses π_i. Therefore:

$$\sum_{e \text{ crosses } \pi} \bar{y}_\pi = 0 \quad \forall e \in E_i$$ – satisfying the feasibility criterion.

We now have $\bar{y}$. The next step is to find ϵ s.t.

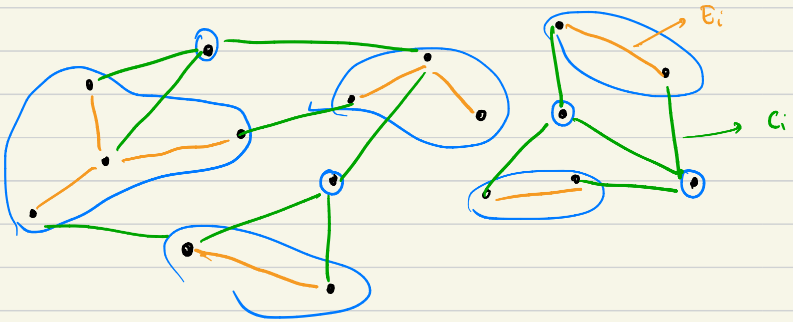
$$\sum_{e \text{ crosses } \pi} (y^i_\pi + \epsilon \bar{y}_\pi) \leq c_e \quad \forall e.$$

We now have $\bar{y}$. The next step is to find ϵ s.t.

$$\sum_{e \text{ crosses } \pi} (y^i_\pi + \epsilon \bar{y}_\pi) \leq c_e \quad \forall e.$$

$$\hookrightarrow \sum_{e \text{ crosses } \pi} y^i_\pi + \epsilon \sum_{e \text{ crosses } \pi} \bar{y}_\pi \quad \forall e$$

- Let C_i be the set of edges that cross π_i (shown in green below)



- Edges out of C_i do not cross π_i. Therefore by our construction of $\bar{y}$,

$$\sum_{e \text{ crosses } \pi} \bar{y}_\pi = 0 \quad \text{for all } e \notin C_i.$$

Hence we need to find an ϵ s.t.

$$\sum_{e \text{ crosses } \pi} (y^i_\pi + \epsilon \bar{y}_\pi) \leq c_e \quad \text{for all } e \in C_i$$

i.e., $$\sum_{e \text{ crosses } \pi} y^i_\pi + \epsilon \sum_{e \text{ crosses } \pi} \bar{y}_\pi \leq c_e \quad \text{for all } e \in C_i$$

Hence we need to find an ϵ s.t.

$$\sum_{e \text{ crosses } \pi} \left(y^i_\pi + \epsilon \bar{y}_\pi \right) \leq c_e \quad \text{for all } e \in C_i$$

ie.,
$$\sum_{e \text{ crosses } \pi} y^i_\pi + \epsilon \sum_{e \text{ crosses } \pi} \bar{y}_\pi \leq c_e \quad \text{for all } e \in C_i$$

ie.,
$$\sum_{e \text{ crosses } \pi} y^i_\pi + \epsilon \cdot 1 \leq c_e \quad \text{for all } e \in C_i$$

$$\epsilon = \min_{e \in C_i} \left\{ c_e - \sum_{e \text{ crosses } \pi} y^i_\pi \right\}$$

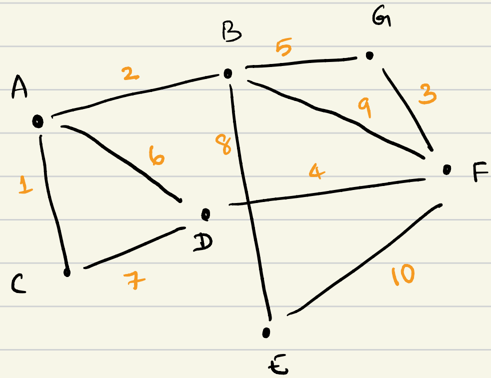
PRIMAL - DUAL ALGORITHM:

- Assume $y^0_\pi = 0 \;\; \forall \pi$.

Notice that we do not want to explicitly write this out since there are exponentially many of them.

- Iterative step: At the i^{th} iteration: let E_i denote the set of edges that are tight for y^i.
  - Find the connected components of $G_i = (V, E_i)$. Let π_i be the partition given by these components.
  - Increase $y^i_{\pi_i}$ until some edge becomes tight.
- Termination: Previous step terminates when the tight edges form one connected component.

Example:

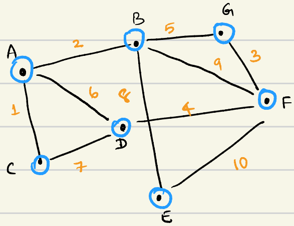


y_0

$E_0 = \phi$

$G_0 = (V, E_0)$

π_0



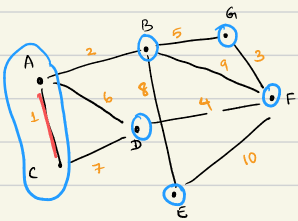
y^1

$E_1 = \{ AC \}$

2: $y_{\pi_0} + y_{\pi_1} + \dots \le 2$

5: $y_{\pi_0} + y_{\pi_1} + \dots \le 5$

π_1

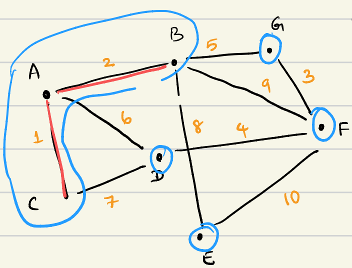


y^2

$E_2 = \{ AC, AB \}$

3: $y_{\pi_0} + y_{\pi_1} + y_{\pi_2} + \dots \le 3$

π_2

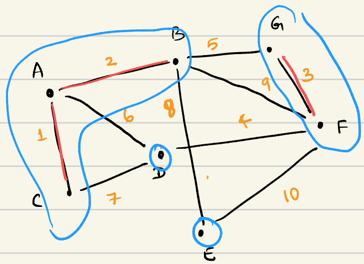


y^3

$E_3 = \{ AC, AB, GF \}$

4: $y_{\pi_0} + y_{\pi_1} + y_{\pi_2} + \dots \le 4$

π_3

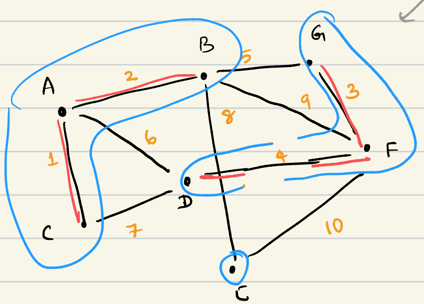


$E_3 = \{AC, AB, GF\}$

y^4

$E_4 = \{AC, AB, GF, DF\}$

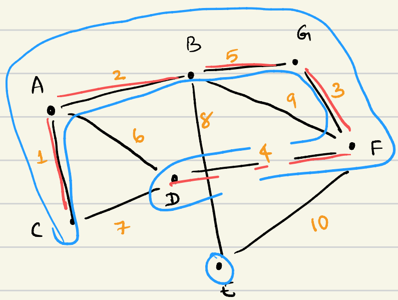
π_4



y^5

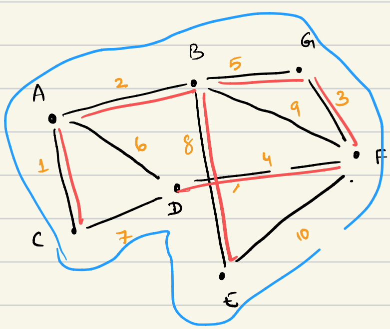
$E_5 = \{AC, AB, GF, DF, BG\}$

π_5



$E_6 = \{AC, AB, GF, DF, BE\}$

π_7



$E_7 = \{AC, AB, BG, GF, DF, \cancel{EF}\ BE\}$

↓

Minimum Spanning Tree

Summary:

PRIMAL - DUAL ALGORITHM:

- Assume $y^0_\pi = 0 \ \forall \pi$.

  Notice that we do not want to explicitly write this out since there are exponentially many of them.

- Iterative step: At the i^{th} iteration: let E_i denote the set of edges that are tight for y^i.
  - Find the connected components of $G_i = (V, E_i)$. Let π_i be the partition given by these components.
  - Increase $y^i_{\pi_i}$ until some edge becomes tight.

- Termination: Previous step terminates when the tight edges form one connected component.

Correctness: Why is this primal-dual algorithm correct?

Complexity: Is the running time a polynomial?

We will see this in the next lecture. This primal-dual algorithm in fact mimicks Kruskal's algorithm for MST.