COL 758: Advanced Algorithms

Lecture 02

What we will study?

1. Randomized Algorithms

Simple and Efficient algorithms
using
Randomness

3. Linear Programming (LP) based Algorithms

Finding approximate solutions to
difficult problems using
LP (linear program)

2. Geometric Algorithms

Algorithms for problems involving
Geometric setting

Eg. Minimum pairwise distance

4. Online Algorithms

Algorithms that work when data is generated
in an on-line manner

Eg. Google Ad-words

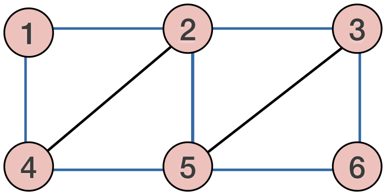
Other concepts (tentative): Gradient descent, streaming algorithms, etc.

Approximate vertex cover (LP Based Algorithm)

Given: A graph $G = (V, E)$ with n vertices.

Def: A subset $W \subseteq V$ such that for each $(a, b) \in E$,
one end-point of (a, b) lies in W.

Example:



Goal: Find a 2-approximation to smallest vertex-cover.

- Fact: Computing smallest vertex-cover is NP-hard.

Linear Program (LP) Formulation

Let $(y_1, \ldots, y_n)$ be a vector.

Add real constraint $y_i \in [0,1]$.

For each edge $(i, j) \in E$:

Add constraint: $y_i + y_j \geq 1$

Optimize: $$\min_{(y_1,\ldots,y_n)\in\{0,1\}^n} \left(\sum_{i=1}^{n} y_i\right)$$

Step 1. Solve linear program (LP).

Step 2. Transform real $y_i's$ to binary $y_i's$.

$\boxed{\text{Region}}$

L.P.

$$x + y \leq 10$$

$$x \geq 0$$

$$y \geq 0$$

$$\underline{\min\ (x - 2y)}$$

I.P.

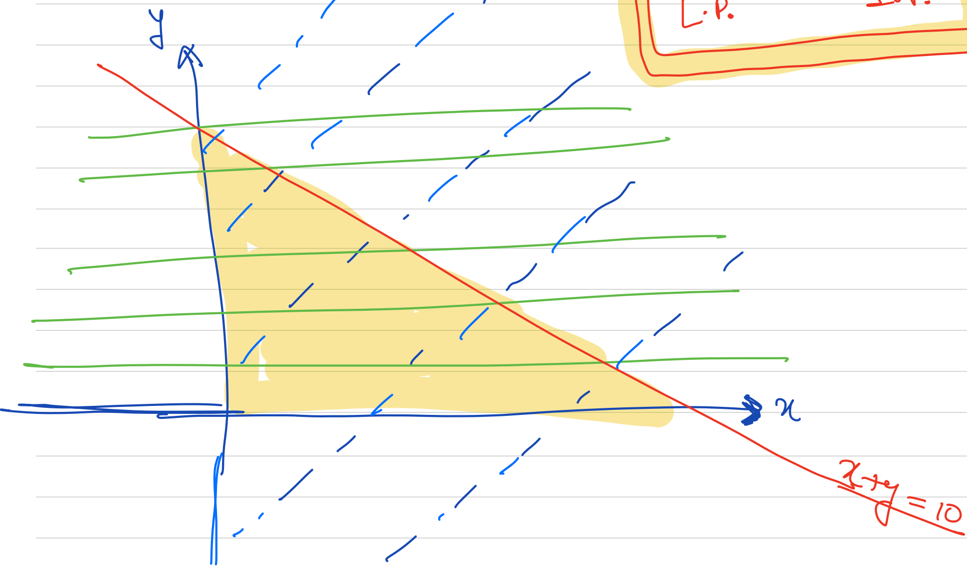
$$x + y \leq 10$$

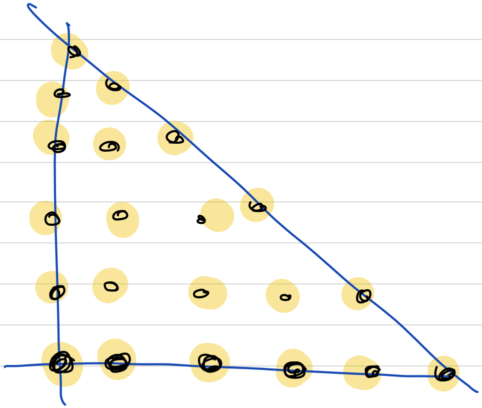
$$x \in \{0, 1, 2, \ldots\}$$

$$y \in \{0, 1, 2, \ldots\}$$

$$\underline{\min\ (x - 2y)}$$

$$\boxed{\text{min of L.P.} \leq \text{min of I.P.}}$$





I.P.

smaller domain

$y_i \in \{0, 1\}$

$\forall (i,j) \in E \qquad y_i + y_j \geq 1$

Min $\sum_{i=1}^{n} y_i$

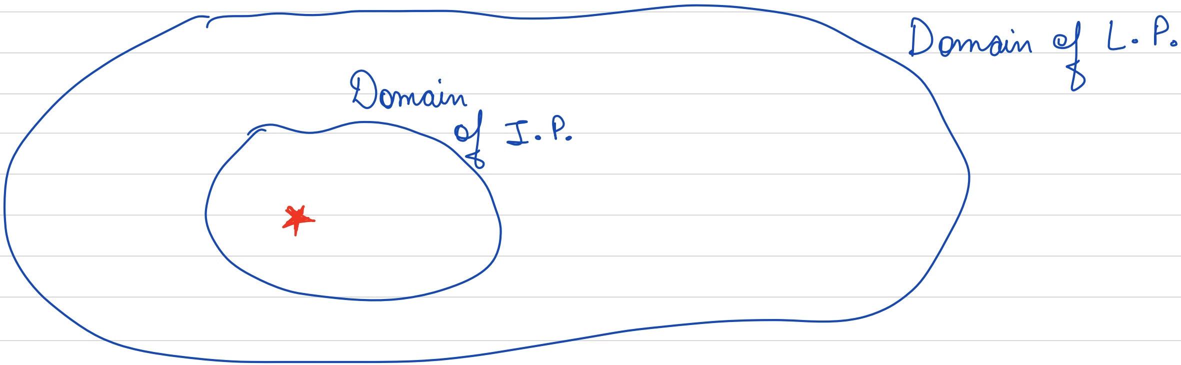
L.P.

larger domain

$0 \leq y_i \leq 1$

$\forall (i,j) \in E \qquad y_i + y_j \geq 1$

Min $\sum_{i=1}^{n} y_i$



optimal sol^n of L.P. $<$ opt sol of I.P.

Goal - Find 2 approx of vertex cover.

I.P. $y_i \in \{0,1\} \ \forall i$, $\forall (i,j) \in E \quad y_i + y_j \geq 1$, min $(y_1 + y_2 + \dots + y_n)$

L.P. $0 \leq y_i \leq 1$, $\forall (i,j) \in E \quad y_i + y_j \geq 1$ min $(y_1 + y_2 + \dots + y_n)$

(1) optimal of L.P. $- (y_1^* \ y_2^* \ \dots \ y_n^*)$

(2) Create a new vector $\boxed{(\bar{y}_1 \ \bar{y}_2 \ \dots \ \bar{y}_n)}$ s.t. $\bar{y}_i = \begin{cases} 1 & y_i^* \geq 1/2 \\ 0 & y_i^* < 1/2 \end{cases}$

Binary vector

(3) $W = \{ i \mid \bar{y}_i = 1 \}$

Rounding

Claim : size of W $\leq$ 2 · size of minimum vertex cover.

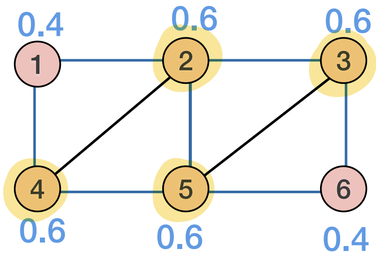
Proof

$$|W| = (\bar{y}_1 + \bar{y}_2 + \dots + \bar{y}_n) \overset{?}{\leq} (2y_1^* + 2y_2^* + \dots + 2y_n^*) = 2 \cdot \text{opt of L.P.}$$

$$\leq 2 \cdot \text{opt of I.P.}$$

$$= 2 \text{ size of smallest vertex cover.}$$

Approximate vertex cover (LP Based Algorithm)



$W = \{2, 3, 4, 5\}$

Claim 1: For any edge (i, j) if $y_i^* + y_j^* \geq 1$, then $\bar{y}_i + \bar{y}_j \geq 1$.

$y_i^* + y_j^* \geq 1$

$\Rightarrow$ either $y_i^* \geq 0.5$ or $y_j^* \geq 0.5$

$\Rightarrow$ either $\bar{y}_i = 1$ or $\bar{y}_j = 1$ $\Rightarrow \bar{y}_i + \bar{y}_j \geq 1$

Claim 2: $\sum_{i=1}^{n} \bar{y}_i \leq \sum_{i=1}^{n} 2y_i^* \leq 2 \cdot$size of smallest vertex cover.

Linear Program (LP) Formulation

Let $(y_1, \ldots, y_n)$ be a vector.

Add real constraint $y_i \in [0,1]$.

For each edge $(i, j) \in E$:

Add constraint: $y_i + y_j \geq 1$

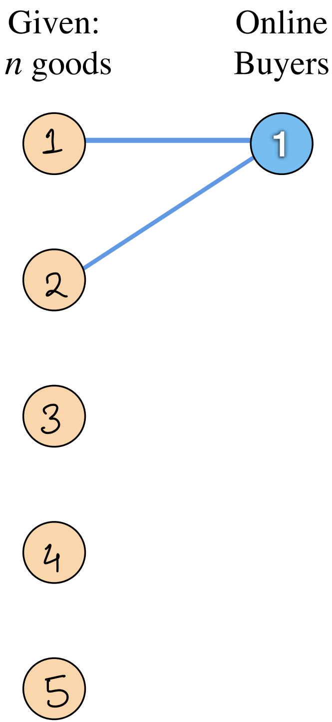
Optimize: $\min_{(y_1,\ldots,y_n)\in\{0,1\}^n} \left(\sum_{i=1}^{n} y_i\right)$

Step 1. Solve linear program in poly-time, and let $(y_1^*, \ldots, y_n^*)$ be optimal solution.

Step 2. Compute a new vector $(\bar{y}_1, \ldots, \bar{y}_n)$ such that $\bar{y}_i = 1$ iff $y_i^* \geq 0.5$

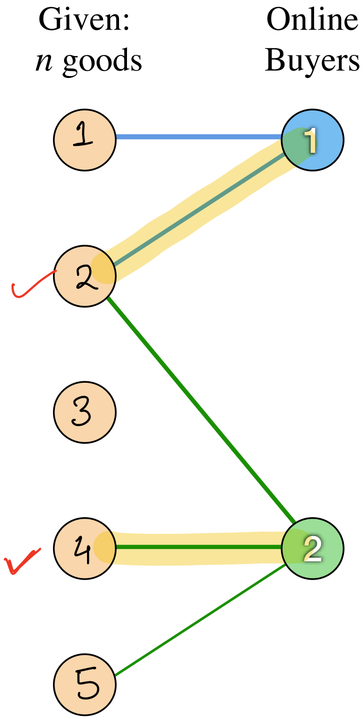
Step 3. Put v_i in W iff $\bar{y}_i = 1$. Return W.

IV. Online Algorithm: Bi-partite Matching



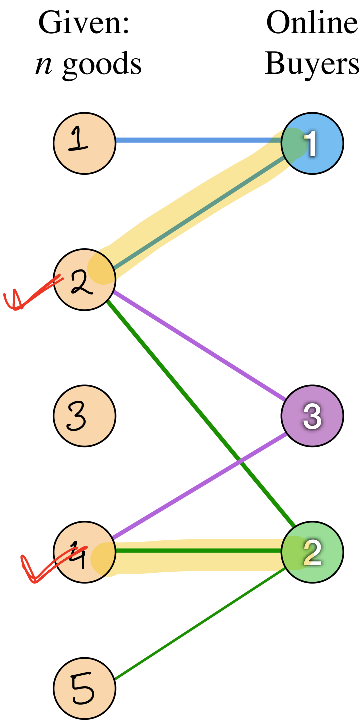
Information of buyers not known apriori

IV. Online Algorithm: Bi-partite Matching



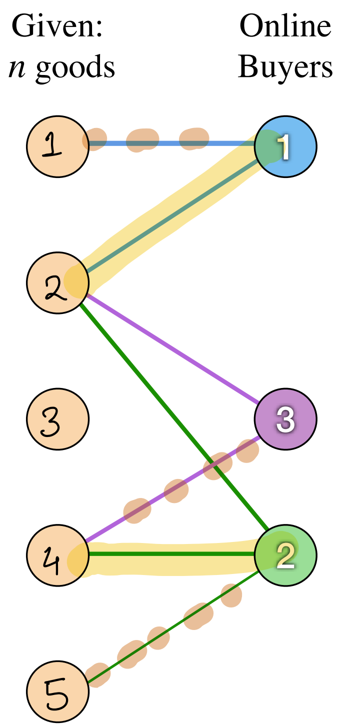
Information of buyers not known apriori

IV. Online Algorithm: Bi-partite Matching



Information of buyers not known apriori

IV. Online Algorithm: Bi-partite Matching



Information of buyers not known apriori

Goal: A strategy to obtain a matching of large size.

Assumption: G contains a perfect matching after all buyers are added.

After n buyers are added we have a matching of size "n"

Competitive ratio $= \frac{|M|}{n}$

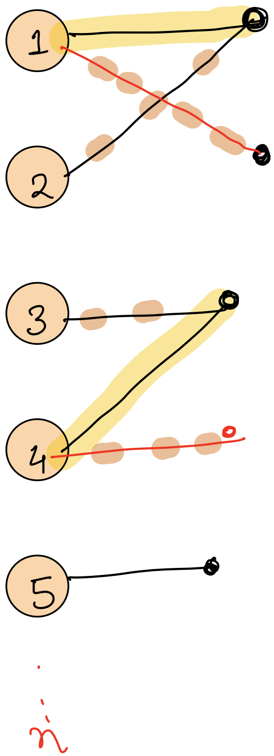
matching obtained ONLINE

IV. Online Algorithm: Bi-partite Matching

Assumption: G contains a perfect matching after all buyers are added.

Given: n goods

Online Buyers



For any deterministic strategy we can find an instance I s.t. $\frac{|M_I|}{n} \leq 1/2$

$$\min_{\text{instance } I} \left(\frac{|M|}{n} \right) \leq \frac{1}{2}$$

Randomized — expected value of $\frac{|M_I|}{n} \geq 1 - \frac{1}{e} \approx 0.63$