Tuesday, October 25

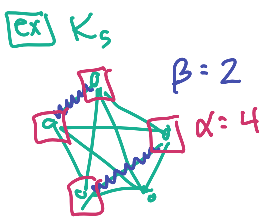
1. Welcome!
2. Looking Ahead: Project Workday Thursday
   ↳ bring a device to work on
3. HW 6 posted Thursday
   ↳ Possibly penultimate?
4. Questions?
5. Matchings!
6. Outro
   ↳ small work: read over feedback, resource check.

Math 494: Discrete Optimization 10/25

Happy last day of matchings! Today we'll start out by continuing to discuss matchings in bipartite graphs and use this to develop an approximation algorithm for matchings in arbitrary graphs. After that, we'll change gears to talk about minimum cost matchings in bipartite graphs.

Recall: we mentioned that for a bipartite graph, the maximum matching and the minimum vertex cover are equal in size. This was because the matching polytope and the covering polytope are both integral, which means relaxing to the LP doesn't change the problem. However, we also saw that for nonbipartite graphs, like the Petersen graph, we may not have that equality.

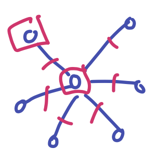
Example: Recall that $\alpha(G)$ is the size of the minimum vertex cover while $\beta(G)$ was the size of the maximum matching. Calculate $\alpha(K_{2k+1})$ and $\beta(K_{2k+1})$. As a follow up, why might we consider these particular graphs?



For K_{2K+1} : $\alpha = 2K$

$\beta = K$

We'll still want to find vertex covers of nonbipartite graphs, though, so we'll instead employ something called an appoximation algorithm: an algorithm that outputs a bound on the question we want to solve. This first one will be, honestly, pretty naive but surprisingly good.



Algorithm 1 ApproxCover(G)

$C \leftarrow \emptyset$
while $E \neq 0$ **do**
  Pick any edge $\{u, v\} \in E$
  $C \leftarrow C \cup \{u, v\}$
  Delete all edges incident to either u or v
end while
return C

Example: In the space below, describe what this algorithm is doing.

* this makes a vertex cover by finding any edge, including its ends in C and deleting all incident edges (we know they're covered)

* it also makes a maximal matching!

Example: So how does this approximation do? Does it put us in the ballpark of $\alpha(G)$? Well, recall that $\beta(G) \leq \alpha(G)$. Can we get an upper bound?

**

vertices in cover returned by algorithm

Claim. *$|C| \leq 2\alpha$, and ~~in particular, $\beta \leq 2\alpha$~~.*

Pf: let M be the maximal matching induced by C.

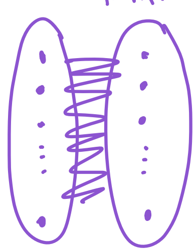
$$|C| = 2|M| \leq 2|M^*| = 2\beta \underset{**}{\leq} 2\alpha$$

M^*: maximum matching

Such an approximation algorithm is called a *2-approximation* or occasionally, a *2-factor approximation.*

Follow up: We might ask ourselves, "can we do any better than two?" Prove that we can't by considering $K_{n,n}$.

$K_{n,n}$



note, the alg. will always return

$|C| = \underline{2n}$

optimal: $\alpha = n$

Emonds Blossom Alg.

Tutte's thm

So let's shift gears to talking about a minimum weight perfect matching in a bipartite graph.

Example: Let G be a bipartite graph G, where $V(G) = U \cup W$ and $|U| = |W|$. Further, suppose $c : E(G) \to \mathbb{R}^+$ be a cost function of the edges. Write an integer program for the minimum cost perfect matching problem.

size of soln space: $n!$

$$P \begin{cases} \min & \sum c_e x_e \\ \text{st} & \sum_{e \sim v} x_e = 1 \quad \forall v \in V \\ & x_e \geq 0 \quad \in \mathbb{Z}. \end{cases}$$

$$D \begin{cases} \max & \sum y_v \\ \text{st} & y_u + y_v \leq c_{uv} \quad \forall uv \in E \\ & y_v \text{ free.} \end{cases}$$

matrix sketch

$$\begin{bmatrix} \min & [\; c \;][x] \\ \text{st} & \text{vert}\overset{\text{edges}}{\begin{bmatrix} A & \vdots \end{bmatrix}} \begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} \vdots \\ 1 \end{bmatrix} \end{bmatrix}$$

2 ones in col. of primal

2 ones in row of dual

Follow up: Dualize your program in the space above.

$$A_{v,e} = \begin{cases} 1 & v \in e \\ 0 & v \notin e \end{cases}$$

Fortunately, since we're still bipartite, this IP also enjoys fact that relaxing to a linear program does not change the solution space. We can work with the primal and dual interchangeably. We'll construct a primal/dual algorithm.

Primal/Dual Algorithm:

(1) Start with a feasible solution to the dual with an associated infeasible primal solution.

(2) Pivot/Augment/Change the dual feasible solution in a way that preserves feasibility.

(3) Terminate when the associated primal solution is feasible. By weak duality, we have optimal.

Wait... have we seen something like this?

Kinda, with Ford's alg!

Min Cost Perfect Matching Algorithm: we build a potential y_v on the vertices that satisfy the bounds $y_u + y_v \leq c_{uv}$ that aim for *equality* across low cost edges.

Why?

well, if $y_u + y_v < c_{uv}$, then $x_{uv} = 0$ → having slack on the edge in the dual means we cannot pick the edge for the matching in primal

↑ complement. slackness.

Algorithm 2 MinPerfect($G = (U \cup W, E), c$)

$y_v \leftarrow \frac{1}{2}\min\{c_e : e \in E\}$ for all $v \in V$

loop

  Construct a graph H where $V(H) = V(G)$ and $E(H) = \{uv : y_u + y_v = c_{uv}\}$

  if H has a perfect matching M **then**

    stop M is a minimum cost perfect matching of G

  end if

  Let S be a deficient set of H

  if all edges of G with an vertex in S have an vertex in $N_H(S)$ **then**

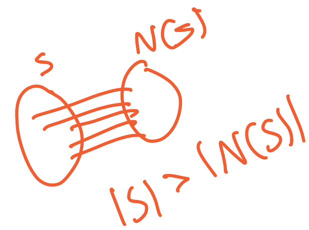
    stop S is a deficient set in G

  end if

  $\varepsilon \leftarrow \min\{c_{uv} - y_u - y_v : uv \in E, u \in S, v \notin N_H(S)\}$

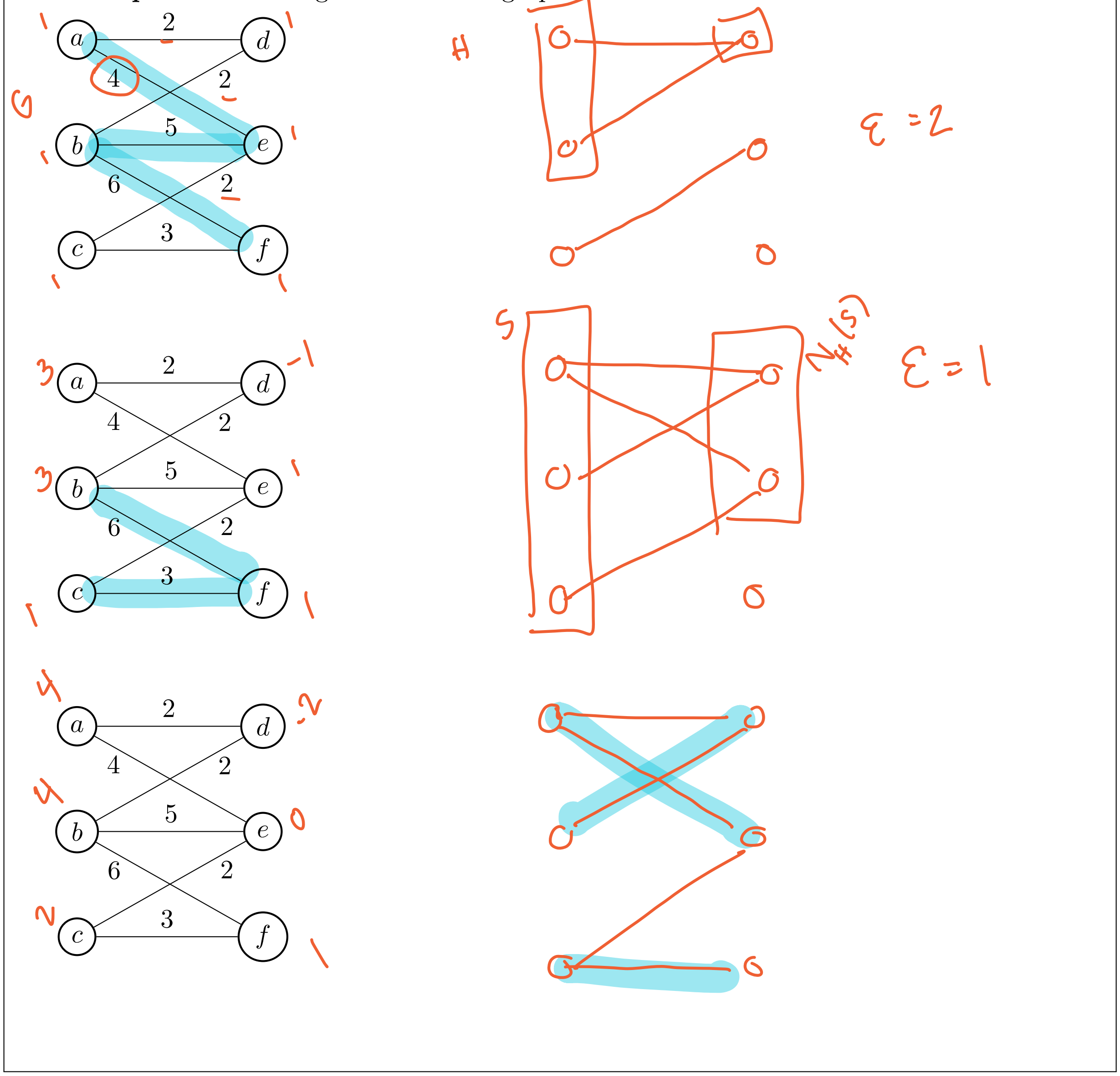
  $$y_v \leftarrow \begin{cases} y_v + \varepsilon & \text{for } v \in S \\ y_v - \varepsilon & \text{for } v \in N_H(S) \\ y_v \text{ otherwise} & \end{cases}$$

end loop



Room for notes:

Example: Run the algorithm for the graph below.



Claim. *If G has a perfect matching and c is rational, then this algorithm produces a minimum matching.*