Vertex Cover LP

Write an LP for finding the minimum weight vertex cover

A set S of vertices is a vertex cover if for every edge (u, v), u is in S, v is in S or both are in S.

What are your variables, then how do you constrain them?

Let $w(u)$ be the weight for a vertex u. You can treat $w(u)$ as a constant.

So, what if the graph isn't bipartite?

Big idea:

Just round!

Pollev.com/robbie

If $x_u \geq \frac{1}{2}$, round up to 1.

If $x_u < \frac{1}{2}$, round down to 0

Minimize $\sum w(u) \cdot x_u$
Subject to:
$x_u + x_v \geq 1$ for all $(u, v) \in E$
$0 \leq x_u \leq 1$ for all u.

Two questions – is it a vertex cover? How far are we from the true minimum?

Recall: Finding an approximation for VC

For every edge, at least one of u,v is in the minimum vertex cover.

But instead of checking which of u, v a good idea to add, just add them both!

```
While(G still has edges)
     Choose any edge (u,v)
     Add u to VC, and v to VC
     Delete u v and any edges touching them
EndWhile
```

What have we seen this quarter?

Stable Matchings

Graph Search
- BFS/DFS
- Graph modeling

Greedy Algorithms

Divide and Conquer

Dynamic Programming
- DP on arrays
- Adding parameters
- DP on trees

Network Flow
- Using Max-Flow
- Using Min Cuts
- Assignment problems

NP-completeness
- Reductions
- Showing a problem is NP-hard
- P vs. NP

Linear Programming

Approximation Algorithms