Part I

Weighted vertex cover

Weighted vertex cover problem

\[ G = (V, E) \]
Each vertex \( v \in V \): cost \( c_v \).
Compute a vertex cover of minimum cost.

1. vertex cover: subset of vertices \( V \) so each edge is covered.
2. \textbf{NP-Hard}
3. ...unweighted \textbf{Vertex Cover} problem.
4. ... write as an integer program (IP):
5. \( \forall v \in V : x_v = 1 \iff v \text{ in the vertex cover}. \)
6. \( \forall vu \in E : \text{covered. } \implies x_v \lor x_u \text{ true. } \implies x_v + x_u \geq 1. \)
7. minimize total cost: \( \min \sum_{v \in V} x_v c_v \).

Weighted vertex cover – rounding the LP

1. Optimal solution to this LP: \( \hat{x}_v \) value of var \( X_v, \forall v \in V \).
2. optimal value of LP solution is \( \hat{\alpha} = \sum_{v \in V} c_v \hat{x}_v. \)
3. optimal integer solution: \( x'_v, \forall v \in V \) and \( \alpha'_I \).
4. \textbf{Any valid solution to IP is valid solution for LP!}
5. \( \hat{\alpha} \leq \alpha'_I. \)
   Integral solution not better than \textbf{LP}.
6. Got fractional solution (i.e., values of \( \hat{x}_v \)).
7. Fractional solution is better than the optimal cost.
8. Q: How to turn fractional solution into a (valid!) integer solution?
9. Called \textbf{rounding}.

Weighted vertex cover

\[
\begin{align*}
\text{min} & \quad \sum_{v \in V} c_v x_v, \\
\text{such that} & \quad x_v \in \{0, 1\} \quad \forall v \in V \quad (1) \\
& \quad x_v + x_u \geq 1 \quad \forall vu \in E.
\end{align*}
\]

\[
\begin{align*}
\text{min} & \quad \sum_{v \in V} c_v x_v, \\
\text{s.t.} & \quad 0 \leq x_v \quad \forall v \in V, \\
& \quad x_v \leq 1 \quad \forall v \in V, \\
& \quad x_v + x_u \geq 1 \quad \forall vu \in E.
\end{align*}
\]
How to round?

1. consider vertex \( v \) and fractional value \( \hat{x}_v \).
2. If \( \hat{x}_v = 1 \) then include in solution!
3. If \( \hat{x}_v = 0 \) then do not include in solution.
4. if \( \hat{x}_v = 0.9 \) \( \implies \) LP considers \( v \) as being 0.9 useful.
5. The LP puts its money where its belief is...
6. ...\( \alpha \) value is a function of this “belief” generated by the LP.
7. Big idea: Trust LP values as guidance to usefulness of vertices.
8. Pick all vertices \( \geq \) threshold of usefulness according to LP.
9. \( S = \{ v \mid \hat{x}_v \geq 1/2 \} \).
10. Claim: \( S \) a valid vertex cover, and cost is low.
11. Indeed, edge cover as: \( \forall vu \in E \) have \( \hat{x}_v + \hat{x}_u \geq 1 \).
12. \( \hat{x}_v, \hat{x}_u \in (0, 1) \)
   \( \implies \hat{x}_v \geq 1/2 \) or \( \hat{x}_u \geq 1/2 \).
   \( \implies v \in S \) or \( u \in S \) (or both).
   \( \implies S \) covers all the edges of \( G \).

Cost of solution

- Cost of \( S \):
  \[
  c_S = \sum_{v \in S} c_v = \sum_{v \in S} 1 \cdot c_v \leq 2 \sum_{v \in S} \hat{x}_v \cdot c_v \leq 2 \sum_{v \in V} \hat{x}_v c_v = 2\alpha \leq 2\alpha',
  \]
  since \( \hat{x}_v \geq 1/2 \) as \( v \in S \).
  \( \alpha' \) is cost of the optimal solution \( \implies \)

Theorem

The Weighted Vertex Cover problem can be 2-approximated by solving a single LP. Assuming computing the LP takes polynomial time, the resulting approximation algorithm takes polynomial time.

The lessons we can take away

Or not - boring, boring, boring.

1. Weighted vertex cover is simple, but resulting approximation algorithm is non-trivial.
2. Not aware of any other 2-approximation algorithm does not use LP. (For the weighted case!)
3. Solving a relaxation of an optimization problem into a LP provides us with insight.
4. But... have to be creative in the rounding.