Vertex Cover

Linear Progamming and Approximation Algorithms

Joshua Wetzel

Department of Computer Science
Rutgers University–Camden
wetzeljo@camden.rutgers.edu

March 24, 2009

What is Linear Programming?

- Optimize a linear objective fn subject to linear constraints.
- Examples:

$$\begin{array}{ll} \min & 7x_1 + x_2 + 3x_3 \\ \text{s.t.} & \\ & x_1 - x_2 + 3x_3 \geq 10 \\ & 5x_1 + 2x_2 - x_3 \geq 6 \\ & x_1, x_2, x_3 \geq 0 \end{array}$$

$$\begin{array}{ll} \max & 6x_1 + 2x_2 - 4x_3 \\ \text{s.t.} & \\ & 5x_1 + x_2 - 2x_3 \leq 14 \\ & 2x_1 - 2x_2 + 4x_3 \leq 20 \\ & x_1, x_2, x_3 \geq 0 \end{array}$$

Feasible Solutions: Upper Bounding OPT

Linear Program:

$$\begin{aligned} \min \quad & 7x_1 + x_2 + 3x_3 \\ \text{s.t.} \quad & \\ & x_1 - x_2 + 3x_3 \geq 10 \\ & 5x_1 + 2x_2 - x_3 \geq 6 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

Feasible Solutions:

$$x_1 = 2,\ x_2 = 1,\ x_3 = 4$$

$$7(2) + 1(1) + 5(4) = 35$$

$$x_1 = \frac{8}{5},\ x_2 = \frac{1}{2},\ x_3 = 3$$

$$7(\tfrac{8}{5}) + 1(\tfrac{1}{2}) + 5(3) = 26.7$$

- All constraints are satisfied.
- $OPT \leq 35, \quad OPT \leq 26.7$

Finding a Lower Bound on OPT

Linear Program:

$$\begin{aligned} \min \quad & 7x_1 + x_2 + 3x_3 \\ \text{s.t.} \quad & \\ & x_1 - x_2 + 3x_3 \geq 10 \\ & 5x_1 + 2x_2 - x_3 \geq 6 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

$$OPT \geq ?$$

Lower Bounding OPT: Adding the Constraints

$$\begin{array}{rl} \min & 7x_1 + x_2 + 3x_3 \\ \text{s.t.} & \\ & x_1 - x_2 + 3x_3 \geq 10 \\ + & 5x_1 + 2x_2 - x_3 \geq 6 \end{array}$$

$$6x_1 + x_2 + 2x_3 \geq 16$$

$$OPT \geq 16$$

Lower Bounding OPT

$$\begin{array}{rrlr} & \min & 7x_1 + x_2 + 3x_3 & \\ \text{s.t.} & & x_1 - x_2 + 3x_3 \geq 10 & (y_1) \\ & + & 5x_1 + 2x_2 - x_3 \geq 6 & (y_2) \end{array}$$

$$x_1(\boxed{y_1 + 5y_2}) + x_2(\boxed{-y_1 + 2y_2}) + x_3(\boxed{3y_1 - y_2}) \geq \boxed{10y_1 + 6y_2}$$

$10y_1 + 6y_2$ is a **lower bound** on OPT if:

$$\begin{aligned} y_1 + 5y_2 &\leq 7 \\ -y_1 + 2y_2 &\leq 1 \\ 3y_1 - y_2 &\leq 3 \end{aligned}$$

LP to Lower Bound OPT

$$
\begin{aligned}
\max \quad & 10y_1 + 6y_2 \\
\text{s.t.} \quad & \\
& y_1 + 5y_2 \leq 7 \\
& -y_1 + 2y_2 \leq 1 \\
& 3y_1 - y_2 \leq 3 \\
& y_1, y_2 \geq 0
\end{aligned}
$$

Primal LP and Dual LP

Primal LP:

$$\begin{aligned} \min \quad & 7x_1 + x_2 + 3x_3 \\ \text{s.t.} \quad & x_1 - x_2 + 3x_3 \geq 10 \\ & 5x_1 + 2x_2 - x_3 \geq 6 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

Dual LP:

$$\begin{aligned} \max \quad & 10y_1 + 6y_2 \\ \text{s.t.} \quad & y_1 + 5y_2 \leq 7 \\ & -y_1 + 2y_2 \leq 1 \\ & 3y_1 - y_2 \leq 3 \\ & y_1, y_2 \geq 0 \end{aligned}$$

- Every primal LP has a corresponding dual LP.
- If the primal is a min problem, the dual is a max problem.
- There is a dual constraint corresponding to each primal variable.

LP Duality Theorems

$$\boxed{Dual_{Feasible} \leq Dual_{OPT} = Primal_{OPT} \leq Primal_{Feasible}}$$

The Primal LP:

$$\begin{aligned} \min \quad & 7x_1 + x_2 + 3x_3 \\ \text{s.t.} \quad & x_1 - x_2 + 3x_3 \geq 10 \\ & 5x_1 + 2x_2 - x_3 \geq 6 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

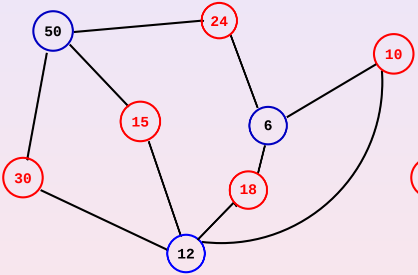
The Dual LP:

$$\begin{aligned} \max \quad & 10y_1 + 6y_2 \\ \text{s.t.} \quad & y_1 + 5y_2 \leq 7 \\ & -y_1 + 2y_2 \leq 1 \\ & 3y_1 - y_2 \leq 3 \\ & y_1, y_2 \geq 0 \end{aligned}$$

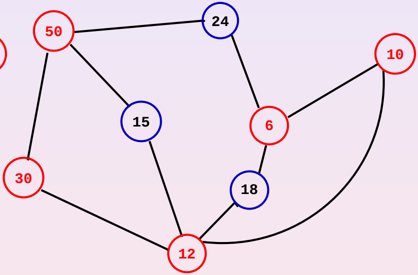
Vertex Cover

- **Input:**
  - Given $G = (V, E)$
  - Non-negative weights on vertices
- **Objective:**
  - Find a least-weight collection of vertices such that each edge in G in incident on at least one vertex in the collection.

Vertex Cover: Example



$COST = 97$



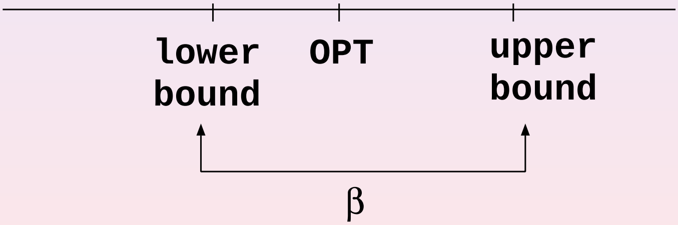
$COST = 108$

Approximation Algorithms

- NP-hard problems.
  - No optimal poly-time algorithms are known
- β-approximation alg., *A*, for a minimization problem *P*
  - poly-time algorithm.
  - for every instance *I* of *P*, *A* produces solution of cost at most $\beta \cdot OPT(I)$
  - $OPT(I)$?

Approximation Algorithms

- compute a lower bound on *OPT*.
- compare cost of our solution with the lower bound.



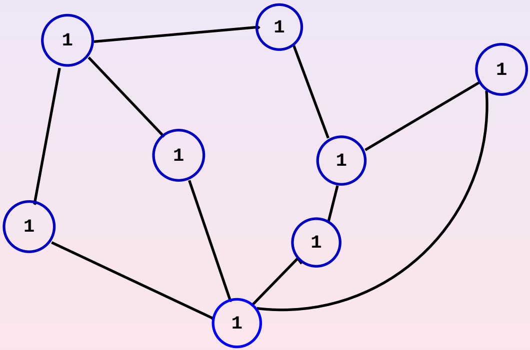
Vertex Cover

- **Input:**
  - Given $G = (V, E)$
  - Non-negative weights on vertices
- **Objective:**
  - Find a least-weight collection of vertices such that each edge in G in incident on at least one vertex in the collection

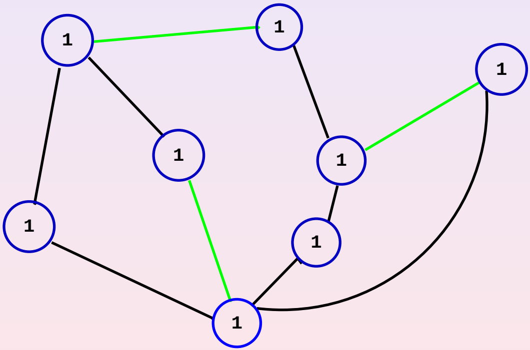
Unweighted Vertex Cover: Algorithm

- Find a maximal matching in G
- Include in our cover both vertices incident on each edge of the matching

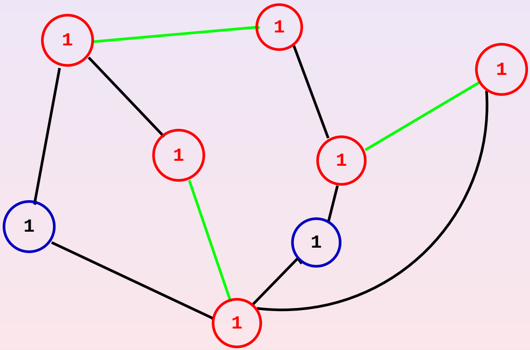
Unweighted Vertex Cover: Example



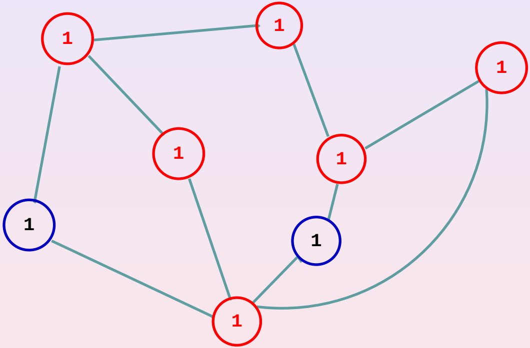
Unweighted Vertex Cover: Example



Unweighted Vertex Cover: Example

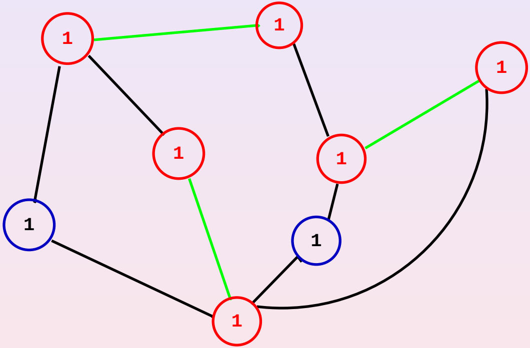


Unweighted Vertex Cover: Example



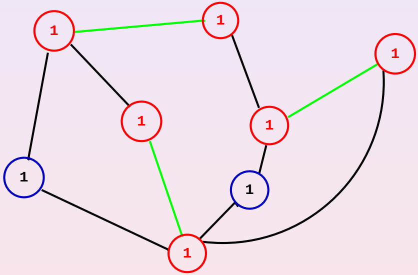
$Cost = 6$

Analysis: Feasibility



- Every black edge shares a vertex with a green edge

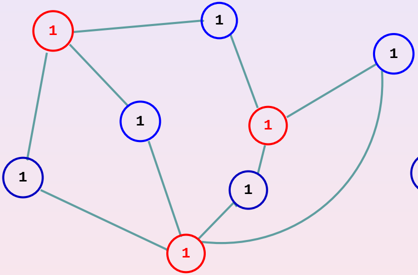
Analysis: Approximation Guarantee



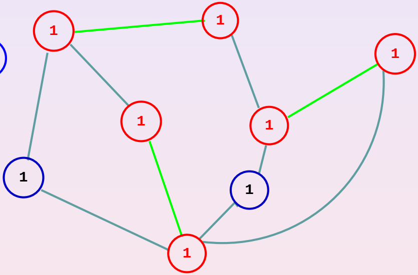
- OPT has to choose at least one endpoint from each green edge.
- We choose both endpoints for each green edge.
- Hence:

$$\text{Our Cost} \leq 2OPT$$

Unweighted Vertex Cover: Tight Example



$OPT = 3$



$COST_{Alg} = 6$

Bad Example

$OPT = 1$

$Cost_{Alg} = 201$

Vertex Cover: IP Formulation

$x_v \leftarrow 1$ if v is in our cover, 0 otherwise

$$\begin{aligned} \text{min} \quad & \sum_{v \in V} w_v x_v \\ \text{s.t.} \quad & \\ & x_a + x_b \geq 1, \ \ \forall \ \ e = (a, b) \\ & x_v \in \{0, 1\}, \ \ \forall \ \ v \in V \end{aligned}$$

LP Relaxation

- Integer programs have been shown to be NP-hard
- Relax the integrality constraints $x_v \in \{0,1\}, \ \forall \ v \in V$

$$\begin{aligned} \min \quad & \sum_{v \in V} w_v x_v \\ \text{s.t.} \quad & \\ & x_a + x_b \geq 1, \ \forall \ e = (a,b) \\ & x_v \geq 0, \ \forall \ v \in V \end{aligned}$$

Why Do This?

- LP can be solved in polynomial time
- Every solution to the IP is also a solution to the LP
- Hence:

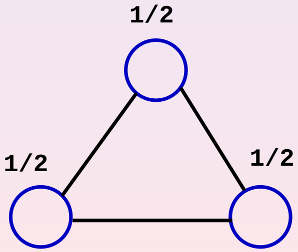
$$\boxed{OPT_{LP} \leq OPT_{IP}}$$

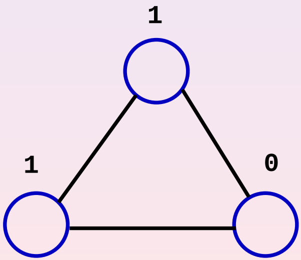
- We can use OPT_{LP} as a lower bound on OPT_{IP}

LP Solution: Example

$$x_a + x_b \geq 1, \quad \forall \quad e = (a, b)$$
$$x_v \geq 0, \quad \forall \quad v \in V$$

$$x_a + x_b \geq 1, \quad \forall \quad e = (a, b)$$
$$x_v \in \{0, 1\}, \quad \forall \quad v \in V$$





$$OPT_{LP} = 1.5$$

$$OPT_{IP} = 2$$

LP-Rounding Algorithm

- $x^* \leftarrow$ optimal LP soln.
- $\hat{x}_v \leftarrow 1$ if $x^*_v \geq \frac{1}{2}$, otherwise $\hat{x}_v \leftarrow 0$
- Include v in our cover iff $\hat{x}_v = 1$

Analysis: Feasibility

$$\min \quad \sum_{v \in V} w_v x_v$$

s.t.

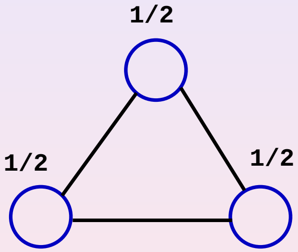
$$x_a + x_b \geq 1, \ \ \forall \ \ e = (a, b)$$
$$x_v \geq 0, \ \ \forall \ \ v \in V$$

$$x_a^* \geq \frac{1}{2} \text{ or } x_b^* \geq \frac{1}{2}$$
$$\hat{x}_a = 1 \text{ or } \hat{x}_b = 1$$

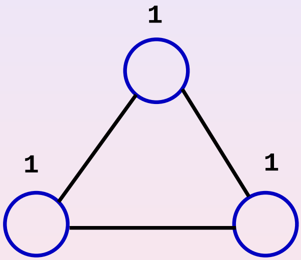
Analysis: Approximation Guarantee

$$\begin{aligned} \text{Our Cost} = \sum_{v \in C} w_v &= \sum_{v \in V} w_v \hat{x}_v \\ &\leq \sum_{v \in V} w_v (2x_v^*) \\ &= 2 \sum_{v \in V} w_v x_v^* \\ &= 2OPT_{LP} \\ &\leq 2OPT_{IP} \end{aligned}$$

LP-Rounding: Tight Example



$OPT_{LP} = 1.5$



Our Cost $= 3$

Primal-Dual Method

$$\boxed{Primal_{OPT} \leq OPT_{IP}}$$

- Solving the LP is expensive.

$$\boxed{Dual_{Feasible} \leq Dual_{OPT} = Primal_{OPT} \leq Primal_{Feasible}}$$

- **Better Alternative:**
  - Construct the dual LP
  - Construct an algorithm that manually tightens dual constraints to obtain a 'maximal' dual solution

Constructing the Dual LP

Primal LP:

$$\begin{aligned} \min \quad & \sum_{v \in V} w_v x_v \\ \text{s.t.} \quad & \\ & x_a + x_b \geq 1, \ \forall \ e = (a, b) \qquad (y_e) \\ & x_v \geq 0, \ \forall \ v \in V \end{aligned}$$

Primal LP and Dual LP

Primal LP:

$$\min \sum_{v \in V} w_v x_v$$

s.t.

$$x_a + x_b \geq 1, \ \forall \ e = (a, b)$$
$$x_v \geq 0, \ \forall \ v \in V$$

Dual LP:

$$\max \sum_{e \in E} y_e$$

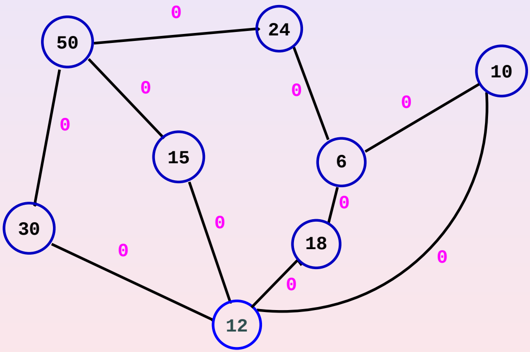
s.t.

$$\sum_{e \,:\, e \text{ hits } v} y_e \leq w_v, \ \forall \ v \in V$$
$$y_e \geq 0, \ \forall \ e \in E$$

Bar-Yehuda and Even Algorithm

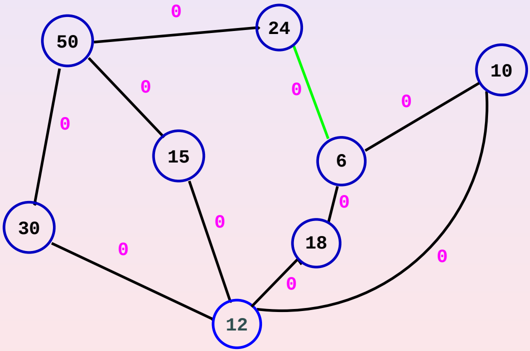
- Inititally all edges are uncovered.
- While $\exists$ an uncovered edge in G:
  - Choose an arbitrary edge, e
  - Raise the value of y_e for that edge until one of its incident vertices , v, becomes full (i.e $\sum_{e:e \text{ hits } v} y_e = w_v$)
  - $C \leftarrow C \cup \{v\}$
  - Any edge that touches v is considered to be covered
- Return C as our vertex cover

Bar-Yehuda and Even Algorithm: Example



Bar-Yehuda and Even Algorithm: Example

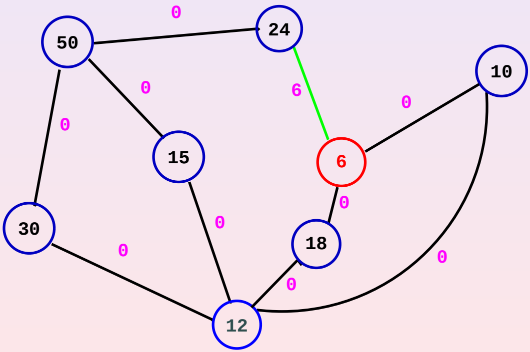
$$\sum_{e:e \text{ hits } v} y_e \leq w_v$$



- Arbitrarily choose e and raise y_e until a vertex is full

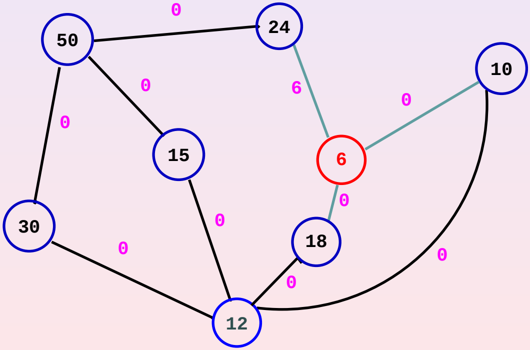
Bar-Yehuda and Even Algorithm: Example

$$\sum_{e:e \text{ hits } v} y_e \leq w_v$$



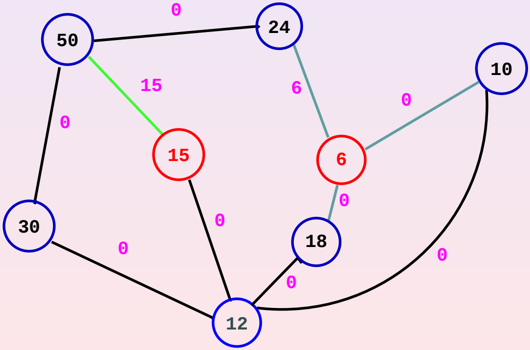
Bar-Yehuda and Even Algorithm: Example

$$\sum_{e:e \text{ hits } v} y_e \leq w_v$$



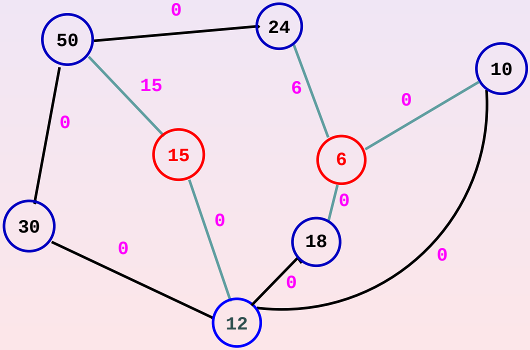
Bar-Yehuda and Even Algorithm: Example

$$\sum_{e:e \text{ hits } v} y_e \leq w_v$$



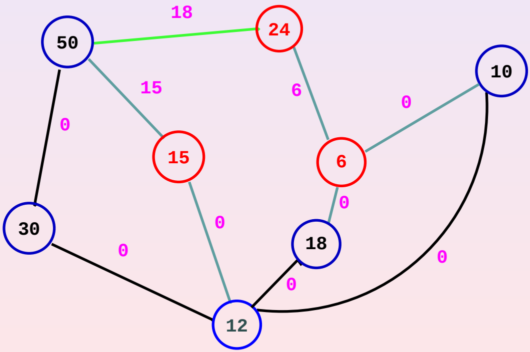
Bar-Yehuda and Even Algorithm: Example

$$\sum_{e:e \text{ hits } v} y_e \leq w_v$$



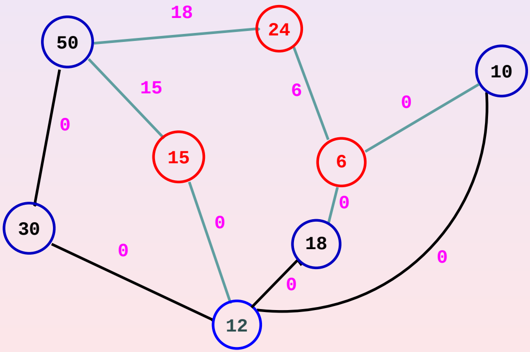
Bar-Yehuda and Even Algorithm: Example

$$\sum_{e:e \text{ hits } v} y_e \leq w_v$$



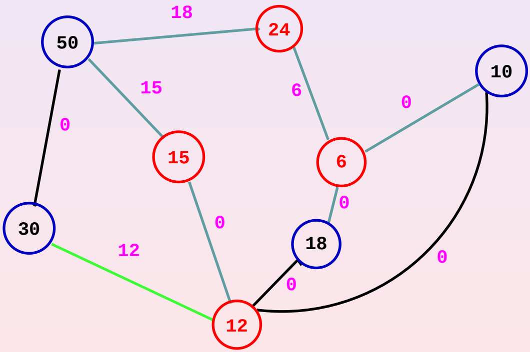
Bar-Yehuda and Even Algorithm: Example

$$\sum_{e:e \text{ hits } v} y_e \leq w_v$$



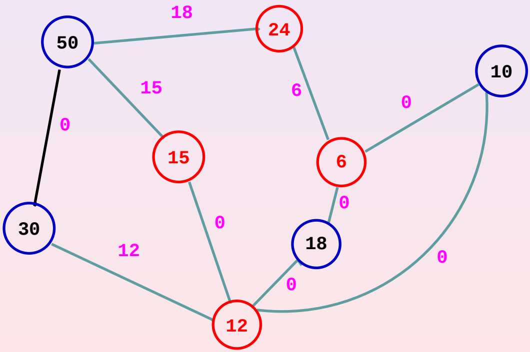
Bar-Yehuda and Even Algorithm: Example

$$\sum_{e:e \text{ hits } v} y_e \leq w_v$$



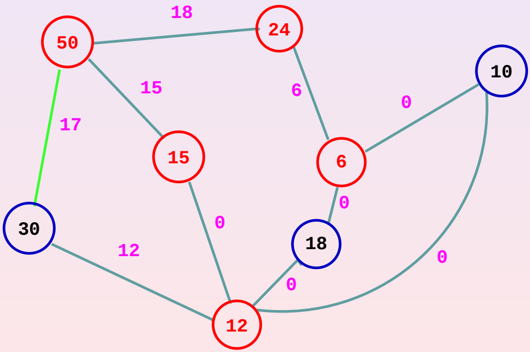
Bar-Yehuda and Even Algorithm: Example

$$\sum_{e:e \text{ hits } v} y_e \leq w_v$$

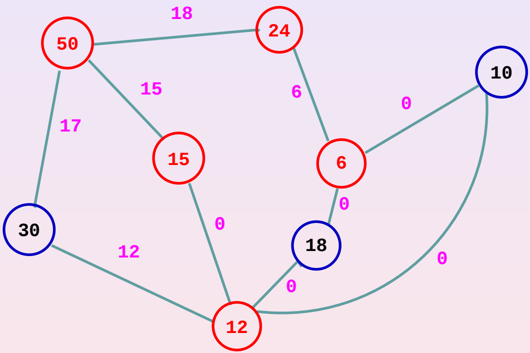


Bar-Yehuda and Even Algorithm: Example

$$\sum_{e:e \text{ hits } v} y_e \leq w_v$$

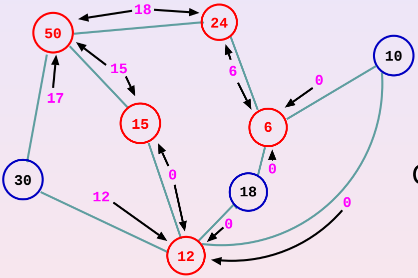


Bar-Yehuda and Even Algorithm: Example



Cost = 107

Bar-Yehuda and Even Algorithm: Analysis



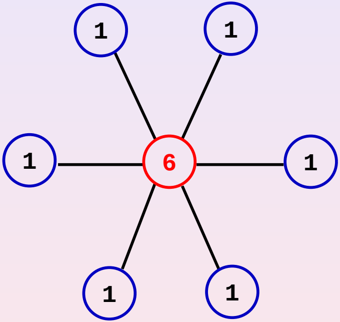
$$\sum_{e: e \text{ hits } v} y_e \leq w_v$$

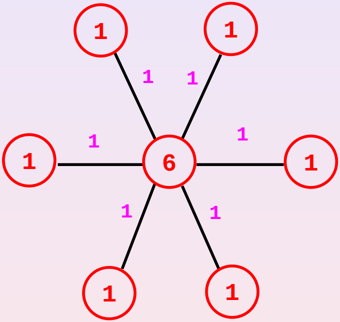
Dual Obj. Fn:

$$\max \sum_e y_e$$

$$\begin{aligned}
\text{Our Cost} &= \text{wt(red vertices)} \\
&\leq 2 \sum_{e \text{ hits red}} y_e \\
&\leq 2 \sum_e y_e \\
&= 2DFS \\
&\leq 2OPT
\end{aligned}$$

Bar-Yehuda and Even Algorithm: Tight Example

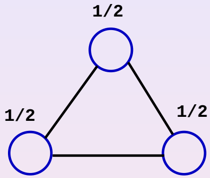


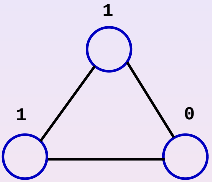


$COST_{OPT} = 6$

$COST_{\text{Bar-Yehuda}} = 12$

Integrality Gap





$OPT_{LP} = 1.5$ $OPT_{IP} = 2$

- For a complete graph of n vertices
- $OPT_{LP} = n/2$
- $OPT_{IP} = n - 1$
- $\lim_{n \to \infty} \frac{OPT_{IP}}{OPT_{LP}} = \lim_{n \to \infty} \frac{n-1}{(n/2)} = 2$

Reference

- **R. Bar-Yehuda and S. Even.** A linear time approximation algorithm for the weighted vertex cover problem. *J. of Algorithms* 2:198-203, 1981.

Thank You.