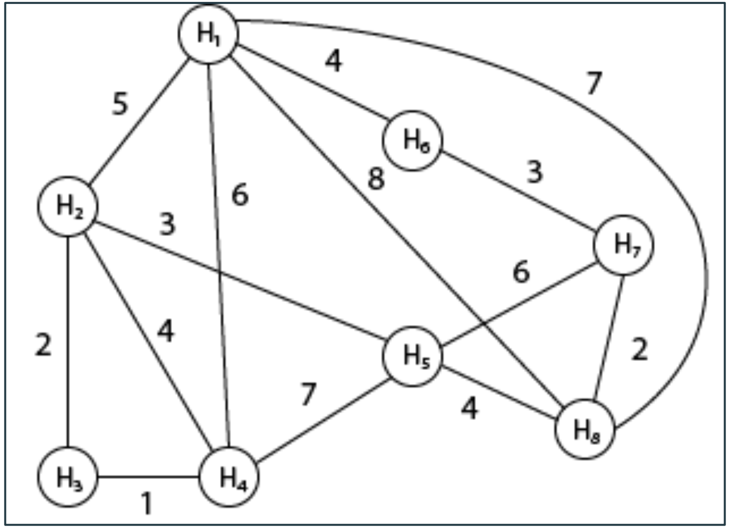
Approximation Algorithms for TSP, Vertex Cover and Set Cover

Approximation Algorithm

Definition: Let P be a minimization problem, and I be an instance of P. Let A be an algorithm that finds feasible solution to instances of P. Let A(I) is the cost of the solution returned by A for instance I, and OPT(I) is the cost of the optimal solution (mimimum) for I. Then, A is said to be an α-approximation algorithm for P if

$$\forall I, \quad \frac{A(I)}{OPT(I)} \leq \alpha \text{ where } \alpha \geq 1.$$

Traveling Salesman Problem



Source: https://www.javatpoint.com/travelling-sales-person-problem

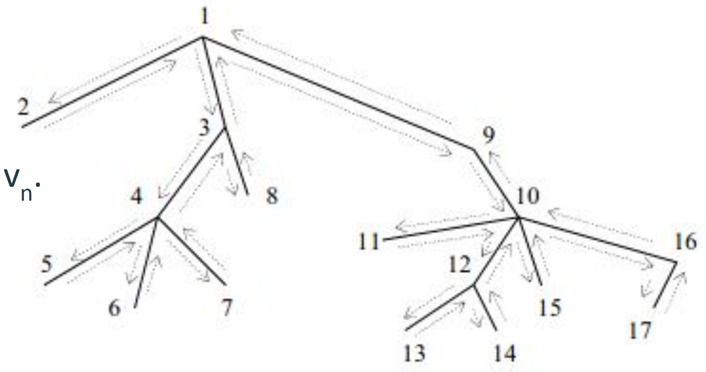
Theorem: For any constant k, it is NP-hard to approximate TSP to a factor of k.

Proof:

- Hamiltonian Cycle is NP-Complete
- If we could approximate TSP we could find a hamiltonian cycle in polynomial time
- P≠NP

Approximation Algorithm for Metric TSP

1. Compute a weighted MST of G.
2. Root MST arbitrarily and traverse in pre-order $v_1, v_2, \ldots, v_n$.
3. Output tour $v_1 \rightarrow v_2 \rightarrow \ldots \rightarrow v_n \rightarrow v_1$.



Claim: A is a 2-approximation algorithm for (metric) TSP

Proof:

If σ is a full walk along the MST in pre-order, σ* an optimum tour and T a spanning tree

$$A(I) \leq cost(\sigma) = 2 * MST(I) \qquad (1)$$

$$OPT(I) = cost(\sigma^*) \geq cost(T) \geq MST(I) \qquad (2)$$

$$(1), (2) \Rightarrow A(I) \leq 2 \times MST(I) \leq 2 \times OPT(I)$$

Christofides' Algorithm

1. Find a minimum spanning tree T
2. Find a minimum-weight perfect matching M for the odd-degree vertices in T
3. Add M to T
4. Find an Euler tour
5. Cut short

Claim: A is a 1.5-approximation algorithm for (metric) TSP

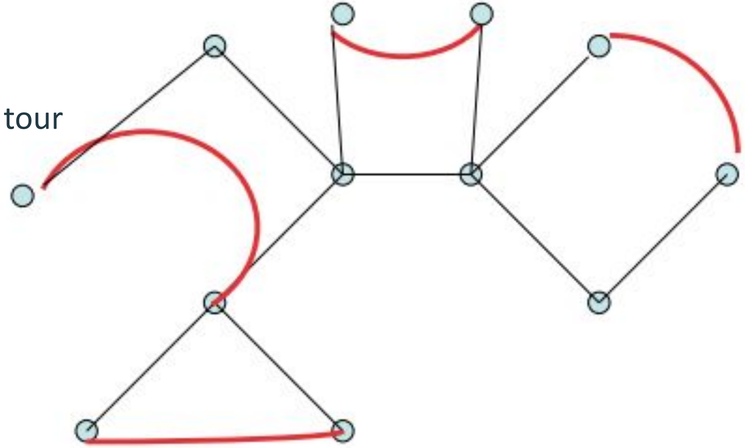
Proof:

If σ is a full walk along the MST in pre-order and σ* an optimum tour

$OPT(I)=cost(\sigma^*)\geq cost(T)\geq MST(I)$ (1)

$cost(M)\leq 0.5\ OPT(I)$ (2)

(1), (2) $\Rightarrow cost(C)\leq cost(T)+ cost(M)\leq 1.5*OPT(I)$



Set Cover

Problem:

Given a universe U of elements {1, . . . , n}, and

a collection of subsets $S = \{S_1, . . . , S_m\}$ of subsets of U,

together with cost a cost function $c(S_i) > 0$,

find a minimum cost subset of S, that covers all elements in U.

$C \leftarrow \emptyset$

$\text{Result} \leftarrow \emptyset$

$\texttt{while} \quad C \neq U$

$\quad S \leftarrow \arg\min_{S \in \mathcal{S}} \frac{c(S)}{|S \setminus C|}$

$\quad \forall e \in (S \setminus C) : \text{price-per-item(e)} \leftarrow \frac{c(S)}{|S \setminus C|}$

$\quad C \leftarrow C \cup S$

$\quad \text{Result} \leftarrow \text{Result} \cup \{S\}$

$\texttt{end}$

$\texttt{return} \quad \text{Result}$

Theorem: The algorithm gives us a *[ln n + O(1)]*-approximation for set cover.

Proof: Let $e_1, \ldots, e_n$ be the elements in the order they are covered by the algorithm. At iteration k, there must be a set from S that is contained in the optimal solution and covers some of the remaining elements U \ C including e_k at cost at most OPT. Thus we have

$$\text{price-per-item}(e_k) \leq \frac{OPT}{|U \setminus C|} = \frac{OPT}{n - k + 1}$$

and using this, we can upper bound the total cost of the greedy solution by

$$\text{Total Cost} = \sum_{k\in[1,n]} \text{price-per-item}(e_k) \leq \sum_{k\in[1,n]} \frac{OPT}{n-k+1} = \sum_{k\in[1,n]} \frac{OPT}{k} = OPT \cdot H_n \approx OPT\cdot(\ln n + 0.6)$$

Reminder (Harmonic Number):

$H_n = \sum_{k\in[1,n]} \frac{1}{k}$

$\lim_{n\to\infty}(H_n - \ln n) = \gamma$

$\gamma \approx 0.5772156649$

Euler–Mascheroni constant.

Vertex Cover

Algorithm 1: APPROX-VERTEX-COVER(G)

$C \leftarrow \emptyset$

while $E \neq \emptyset$

pick any $\{u, v\} \in E$

$C \leftarrow C \cup \{u, v\}$

delete all eges incident to either u or v

return C

Note: Algorithm A finds a maximal matching M

Claim 1: This algorithm gives a vertex cover

Proof: Every edge $\in$ M is clearly covered. If an edge, e $\notin$M is not covered, then M $\cup$\{e\} is a

matching, which contradict to maximality of M.

Claim 2: This vertex cover has size ≤ 2×minimum size (optimal solution)

Proof: The optimum vertex cover must cover every edge in M. So, it must include at least one of the endpoints of each edge $\in$ M, where no 2 edges in M share an endpoint. Hence, optimum vertex cover must have size

$$OPT(I) \geq |M|$$

But the algorithm A return a vertex cover of size 2|M|, so $\forall$ I we have

$$A(I) = 2|M| \leq 2 \times OPT(I)$$

implying that A is a 2-approximation algorithm.

Linear Programming

1. Linear Function:

   $f(x_1,x_2)=c_1x_1+c_2x_2$

2. Constraints:

   $a_{11}x_1+a_{12}x_2\leq b_1$

   $a_{21}x_1+a_{22}x_2\leq b_2$

   $a_{31}x_1+a_{32}x_2\leq b_3$

3. Non-negative variables:

   $x_1\geq 0$

   $x_2\geq 0$

Integer Programming:

$x_i \in \mathbf{Z}$

Example: Linear Programming in 2D

1. max $f(x_1,x_2)=4x+5y$
2. $x+y \le 20$

   $3x+4y \le 72$
3. $x \ge 0$

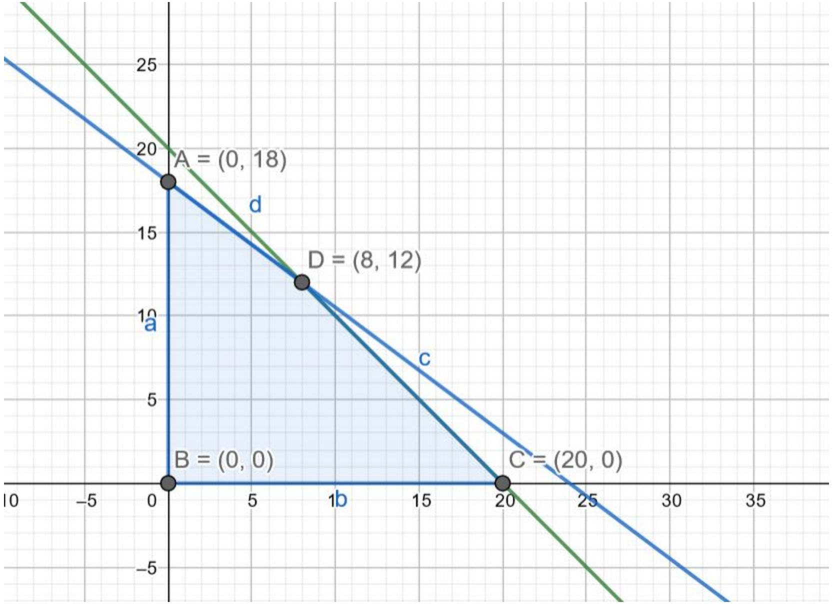
   $y \ge 0$

$f(0,18)=90$

$f(0,0)=0$

$f(20,0)=100$

$f(8,12)=92$



Weighted Vertex Cover

Input: An undirected graph $G = (V, E)$ with vertex weights $w_i \geq 0$.

Problem: Find a minimum-weight subset of nodes S such that every $e \in E$ is incident to at least one vertex in S.

IP Formulation

minimize	$\sum_{i \in V} w_i x_i$	
subject to	$x_i + x_j \geq 1$	$\forall\ (i,j) \in E$
	$x_i \in \{0,1\}$	$\forall\ i \in V$

LP Rexalation

minimize $\sum_{i \in V} w_i x_i$

subject to $x_i + x_j \geq 1 \quad \forall (i,j) \in E$

$x_i \geq 1 \quad \forall i \in V$

LP can be solved in polynomial time, while IP is NP-hard.

LP- Rounding Algorithm:

1. Let x* be an optimal solution.
   - Define $x_v=1$, if $x_v^* \geq 1/2$
   - Define $x_u=0$, if $x_u^* < 1/2$
2. Let $S = \{i \in V : x_i = 1\}$

The set S is a valid vertex cover, because for each edge (u, v) it is true hat $x_u^*+x_v^* \geq 1$, and so at least one of x_u^* or x_v^* must be at least 1/2, and so at least one of u or v belongs to S

Claim: A is a 2-approximation algorithm

Proof: $\sum_{v \in S} w(v)$

$= \sum_{v \in V} w(v) x_v$

$\leq \sum_{v \in V} w(v) \cdot 2 \cdot x_v^*$

$= 2 \cdot OPT(LP)$

$\leq 2 \cdot OPT(VC)$, every solution to the IP is also a solution to the LP, hence

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

Dual LP

Primal LP			**Dual LP**		
minimize	$\sum_{i \in V} w_i x_i$		maximize	$\sum_{e \in E} y_e$	
subject to	$x_i + x_j \geq 1$	$\forall$ edge $(i,j) \in E$	subject to	$\sum_{e:\text{ e hits v}} \leq w_v$	$\forall\ v \in V$
	$x_i \geq 1$	$\forall$ vertex $i \in V$		$y_e \geq 0$	$\forall\ e \in E$

$$\text{Dual}_{\text{Feasible}} \leq \text{Dual}_{\text{OPT}} = \text{Primal}_{\text{OPT}} \leq \text{Primal}_{\text{Feasible}}$$

Bar-Yehuda and Even Algorithm

- Inititally all edges are uncovered.
- While ∃ 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 $\sum_{e:e \text{ hits } v} y_e = w_v$)
  - $S \leftarrow S \cup \{v\}$
  - Any edge that touches v is considered to be covered
- Return S as our vertex cover

Claim 1: The set S returned is a vertex cover.

Reason: At termination, for each edge e= (u,v), at least one of u and v is tight ⇒ at least one of u and v is in S.

Claim 2: The Pricing Method is a 2-approximation algorithm for MWVC.

Proof: Let S*be an optimum vertex cover. Then w(S)≤2w(S*).

$$
\begin{aligned}
w(S) &= \textstyle\sum_{i\in S} w_i \\
&= \textstyle\sum_{i\in S}\sum_{e=(i,j)} y_e \\
&\leq \textstyle\sum_{i\in V}\sum_{e=(i,j)} y_e \\
&= 2\textstyle\sum_{e\in E} y_e \\
&\leq 2\, w(S^*)
\end{aligned}
$$