Approximation Algorithms

For some NP-hard problems, one can devise a polynomial time algorithm that approximate the optimal solution within some constant factor.

An algorithm has an approximation ratio, $\rho(n)$, if for any input of size n, the cost c of the solution produced by the algorithm is within $\rho(n)$ factor of the cost c^* of the optimal solution.

$$\rho(n) \geq \max\left(\frac{c}{c^*}, \frac{c^*}{c}\right)$$

For minimizer: $\frac{c}{c^*} \leq \rho(n)$

For maximizer: $\frac{c^*}{c} \leq \rho(n)$.

Approximation scheme.

These algorithms takes $\epsilon > 0$ ~~as~~ as an input along with the instance of the problem, such that, for any fixed ϵ, the algorithm is a $(1+\epsilon)$-approximation algorithm.

Approximation schemes allows better approximations with more resources.

An approximation scheme is a polynomial time approximation scheme if for any fixed $\epsilon > 0$, the scheme runs in time polynomial in n, the size of the input.

A fully polynomial-time approximation scheme runs in time polynomial both in n & $\frac{1}{\epsilon}$.

$O\left(\left(\frac{1}{\epsilon}\right)^2 n\right)$ ... fully polynomial scheme

$O(n^{2/\epsilon})$ .... <u>not</u> FPS.

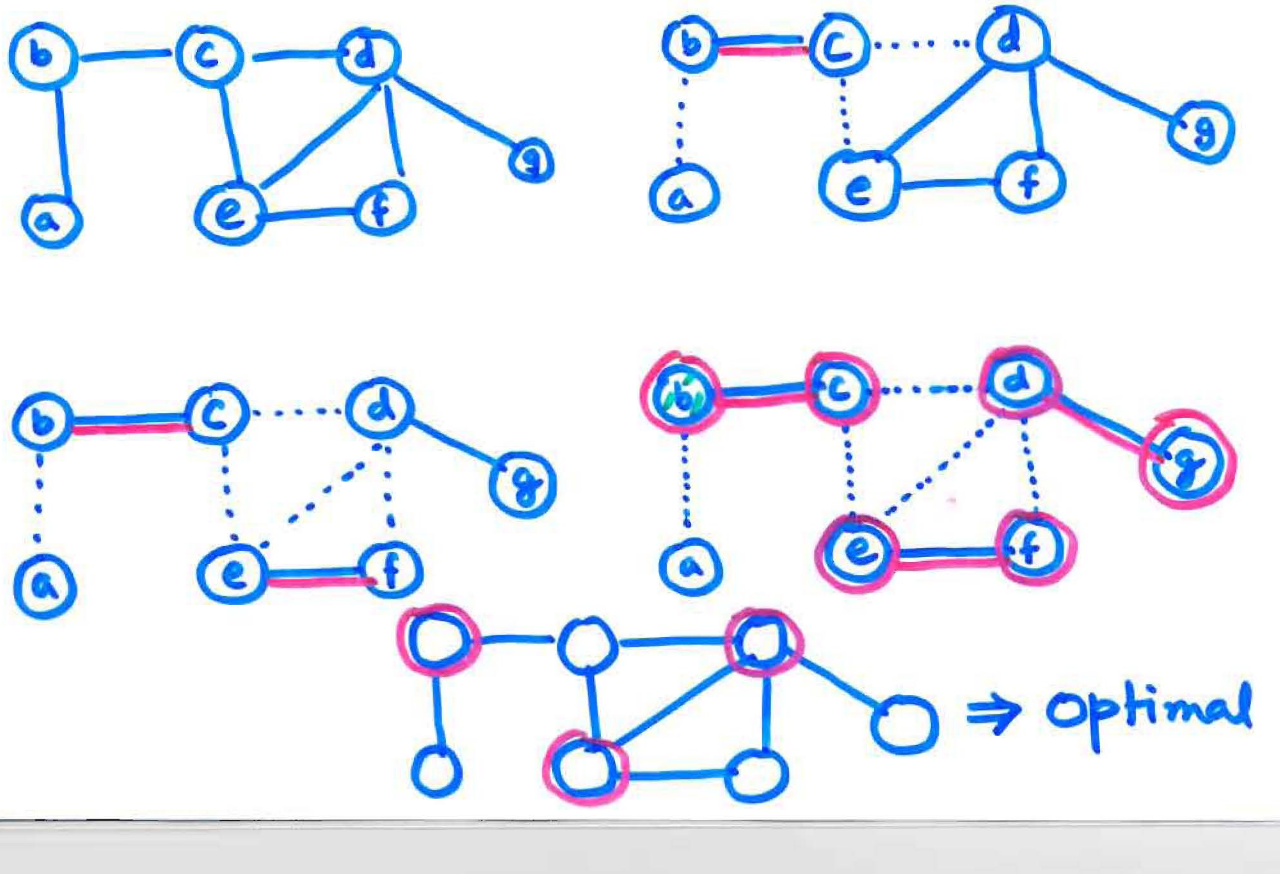
Vertex-Cover Approximation

A vertex cover of $G=(V,E)$ is a subset $V' \subseteq V$ s.t. all edges of G have at least a vertex in V'.

Vertex cover problem is to compute a vertex cover of minimum size.

Vertex cover is known to be NP-Complete

There is a very easy 2-approximation algorithm for vertex cover. Take an edge that is not yet covered and continue.



```
Approx-VC (G)
  C := ∅
  E' := E(G)
  while E' ≠ ∅ do
      let (u,v) be any edge in E'
      C := C ∪ {u,v}
      remove from E' all edges incident
                 on either u or v
  endwhile
  Return C.
```

Theorem. Approx-VC is a polynomial time 2-approximation algorithm.

Proof. Approx-VC runs in at most $O(V^2)$ time .... so polynomial.

- Let A be the set of edges picked.
- Since edges in A are vertex-disjoint any cover, and hence C^*, must contain at least one vertex of each edge in A.
- $|A| \leq |C^*|$.

Let C denote the vertices of A.

- $|C| = 2|A|$
- $|C| = 2|A| \leq 2|C^*|$.

Observation: The algorithm finds a maximal matching (not a proper subset of any other matching). Any optimal cover has to be at least as large as a matching hence maximal matching.

Traveling Salesman Problem (TSP)

Given an undirected $G=(V,E)$ with nonnegative integer cost $c(u,v)$ with each $(u,v)\in E$, find a hamiltonian cycle (no vertex repeated but going through all vertices) of minimum cost.

Assume G to be complete.

Let $\quad c(A) = \sum_{(u,v)\in A} c(u,v)$.

Triangle inequality: For any triple $u, v, w \in V$, $\quad c(u,v) \le c(u,w) + c(w,v)$.

TSP is NP-complete even if triangle inequality is imposed. We show a 2-approximation algorithm that runs in polynomial time for TSP with triangle inequality.

Without triangle inequality, there is no polynomial time constant approximation-algorithm unless $P=NP$.

Algorithm with triangle inequality:

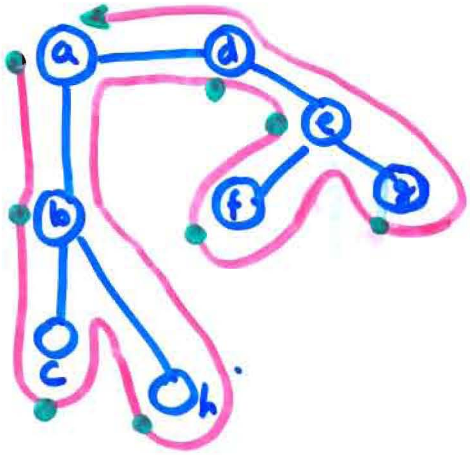
Approx-TSP(G)

Choose any $r \in V[G]$ as a "root".

Compute MST(G, r) with root r.

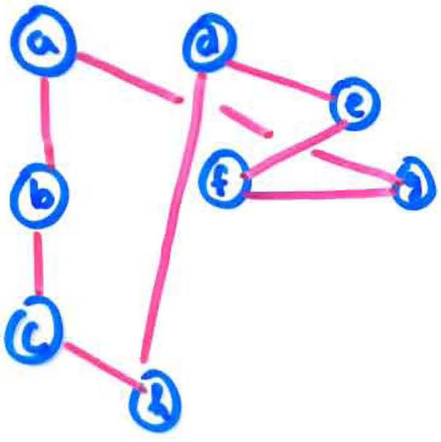
Let L be the list of vertices in a preorder walk of MST T

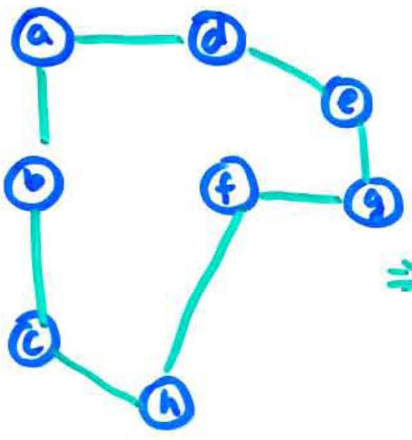
return the hamiltonian cycle H that visits the vertices in order L



L= a b c b h b a d e f e g e d a

H= a b c h d e f g a





⇒ optimal

Theorem Approx-TSP is a polynomial 2-approximation algorithm for TSP with triangle inequality

Proof. - Obviously Approx-TSP runs in polynomial time ($O(V^2)$).

- Let H^* be the optimal tour.
- Since deleting an edge from H^* provides a spanning tree

  $c(T) \leq c(H^*)$ where T is MST.

- A full preorder walk W of T visits all vertices in the walk and visits all edges exactly twice.

  $c(W) = 2c(T)$

- $c(W) \leq 2c(H^*)$, the cost of W is within a factor of 2 of the optimal

- Delete vertex u from W if it is visited twice. So, if $u' u u''$ are the sequences visiting u twice delete edges $u'u$ and uu'' and replacing them with $u'u''$

- repeat the procedure till every vertex is visited exactly once.

- each delete decrease (or cannot increase) the cost because of triangle inequality.

- $c(H) \leq c(W) \leq 2c(H^*)$.

TSP without Triangle inequality:

Theorem If $P \neq NP$, then for any constant $\rho \geq 1$ there is no polynomial-time approximation algorithm with approximation ratio ρ for the TSP without triangle inequality.

Proof. By contradiction.

- Suppose for some $\rho \geq 1$, there is an algorithm A with approximation ratio ρ running in polynomial time.
- Assume ρ is an integer (scale if necessary).
- Use A to solve the hamiltonian-cycle problem in polynomial time. Since Hamiltonian cycle problem is NP-complete this would imply NP = P.
- Let $G = (V, E)$ be an instance of the Hamiltonian-cycle problem.
- construct complete graph $G' = (V, E')$ as:

$$E' = \{(u,v) : u, v \in V \text{ and } u \neq v\}.$$

$$c(u,v) = \begin{cases} 1 & \text{if } (u,v) \in E \\ \rho|V| + 1 & \text{otherwise.} \end{cases}$$

- G' can be created from G in polynomial time

- Consider TSP on G'.
- if G has a Hamiltonian cycle H, c assigns a cost of 1 to each edge in H, hence G' contains a tour of cost $|V|$.

- if G does not have a Hamitonian cycle, then any tour in G', uses an edge that is not in E.
- These tours have cost at least
$$(\rho|V|+1) + (|V|-1) = \rho|V| + |V| > \rho|V|.$$

- A is guaranteed to return a tour of cost no more than ρ times the cost of optimal tour.
- So, A on G' must return the optimal tour if G contains the hamiltonian cycle.
- If G has no hamiltonian cycle, A returns a tour of cost more than $\rho|V|$.
- Therefore, running A on G', we can determine if G has a Hamiltonian cycle or not.