

Approximation Algorithms



Outline and Reading



- ◆ Approximation Algorithms for NP-Complete Problems
 - Approximation ratios
 - 2-Approximation for Vertex Cover
 - 2-Approximation for TSP special case

Approximation Ratios and optimizations problems

We are trying to minimize (or maximize) some cost function $c(S)$ for an optimization problem. E.g.

- ◆ Finding a minimum spanning tree of a graph.
 - Cost function – sum of weights of edges in the graph
- ◆ Finding a cheapest traveling salesperson tour (TSP) in a graph.
- ◆ Finding a smallest vertex cover of a graph
 - Given $G(V,E)$, find a **smallest** set of vertices so that each edge touches at least one vertex of the set.

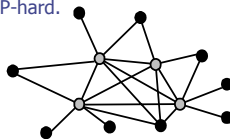
Approximation Ratios



- ◆ An approximation produces a solution T
 - T is a **δ -approximation** to a minimization problem if $c(T) \leq \delta \cdot \text{OPT}$
 - We assume $\delta > 1$
 - **Examples:**
 - Will show how to find a p path in a graph, that visits all vertices, and $w(p) \leq 2w(p^*)$. Here p^* is the cheapest TSP path.

Vertex Cover

- ◆ A **vertex cover** of graph $G=(V,E)$ is a subset W of V , such that, for every $(u,v) \in E$,
 - $u \in W$ and/or $v \in W$.
- ◆ OPT-VERTEX-COVER: Given an graph G , find a vertex cover of G with smallest size.
- ◆ OPT-VERTEX-COVER is NP-hard.



A 2-Approximation for Vertex Cover

- ◆ Let OPT be the opt solution.
- ◆ Every chosen edge e has both ends in C .
- ◆ But e must be covered by at least one vertex of OPT. So, one end of e must be in OPT.
- ◆ $|C| \leq 2|\text{OPT}|$.
- ◆ (there are ≤ 2 vertices of C for each vertex of OPT.)
- ◆ That is, C is a 2-approx. of OPT
- ◆ Running time: $O(|E|)$

```

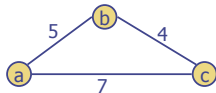
Algorithm VertexCoverApprox(G)
Input graph  $G$ 
Output a vertex cover  $C$  for  $G$ 
 $C \leftarrow$  empty set ;  $H \leftarrow E$ 
/*  $H$  – what is left to be covered */
while  $H$  has edges (not empty){
     $(u,v) \leftarrow$  An edge of  $H$ .
    Add both  $u$  and  $v$  to  $C$ 
    for each edge  $f$  of  $H$  incident to  $v$  or  $w$ 
        Remove  $f$  from  $H$ 
    }
return  $C$ 
    
```

Special Case of the Traveling Salesperson Problem



◆ **OPT-TSP**: Given a **complete**, weighted graph, find a cycle of minimum cost that visits each vertex.

- OPT-TSP is NP-hard
- Special case: edge weights satisfy the triangle inequality (which is common in many applications):
 - ◆ $w(a,b) + w(b,c) \geq w(a,c)$



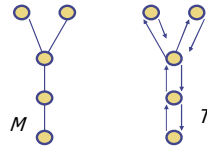
Complete – there is an edge between every pair of vertices

7

From MST to cycles



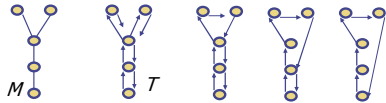
If we are given a MST M of G , a traversal T of M is constructed by picking a source vertex s , and visit the nodes of the graph in a DFS order.



8

Algorithm for constructing approx-TSP C

- ◆ Construct a minimum spanning tree M for G
 - Construct an tour traversal T of M (might intersect itself).
 - **Problem** – a vertex might appear many times on T .
 - **Solution** – shortcutting:
 - ◆ marching along T but:
 - ◆ Each time we have (u,v) followed by (v,w) in T , and v has already been visited, we replace these two edges by the edge (u,w)
 - note that (u,w) is in the graph, and the shortcutting does not increase the total price.



9

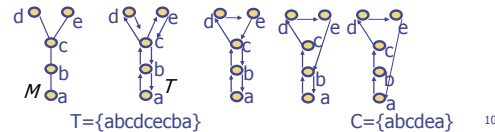
A 2-Approximation for TSP Special Case

Algorithm $TSP_{Approx}(G)$

Input weighted complete graph G , satisfying the triangle inequality
Output a TSP tour T for G

$M \leftarrow$ a minimum spanning tree for G
 $T \leftarrow$ an tour traversal of M , starting at some vertex s
 $C \leftarrow T$

For each vertex v in C (in traversal order)
if this not is v 's first appearance in C **then** delete v from C
Return C



10

A 2-Approximation for TSP

Claim: $w(M) \leq w(OPT)$

Proof:

Recall – M is the minimum spanning tree.
 OPT is a cycle that visit every vertex of G .
 (so it is a tree+one edge)

M is the cheapest such tree.

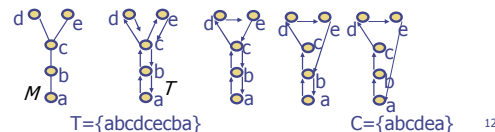
Conclusion: $2w(M) \leq 2w(OPT)$

11

A 2-Approximation for TSP

Claim: $w(C) \leq w(T) = 2w(M) \leq 2w(OPT)$

- ◆ **Proof:**
- ◆ The optimal tour is a spanning tour; hence $2w(M) \leq 2w(OPT)$.
- ◆ The tour T uses each edge of M twice; hence $w(T) = 2w(M)$.
- ◆ Each time we shortcut a vertex in the tour we will **not** increase the total length, so
 - ◆ $w(C) \leq w(T)$.



12