Design and Analysis of Algorithm

Lecture-28:
Approximation Algorithm

Contents

Introduction

In the traveling-salesman problem we are given a complete undirected graph $G\ (V, E)$ that has a nonnegative integer cost $c(u, v\ )$ associated with each edge $(u, v) \in E$, and we must find a a tour of G with minimum cost.

let c(A) denote the total cost of the edges in the subset $A \subseteq E$

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

Triangle inequality

In many practical situations,

the least costly way to go from a place u to a place w is to go directly, with no intermediate steps.

We formalize this notion by saying that the cost function c satisfies the triangle inequality if, for all vertices $u,\ v,\ w \in V$,

$$c(u, w) \leq c(u, v) + c(v, w)$$

The traveling-salesman problem with the triangle inequality

When the cost function satisfies the triangle inequality, we may design an approximate algorithm for the Travelling Salesman Problem that returns a tour whose cost is never more than twice the cost of an optimal tour.

The idea is to use Minimum Spanning Tree (MST).

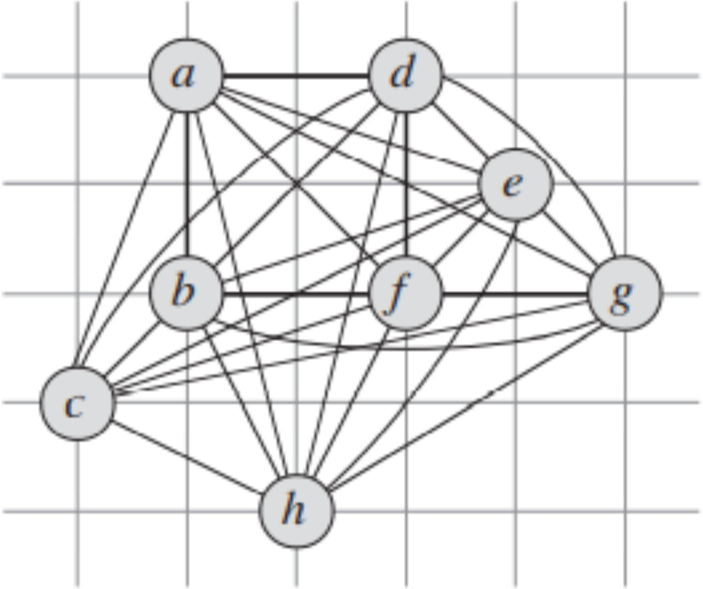
Algorithm

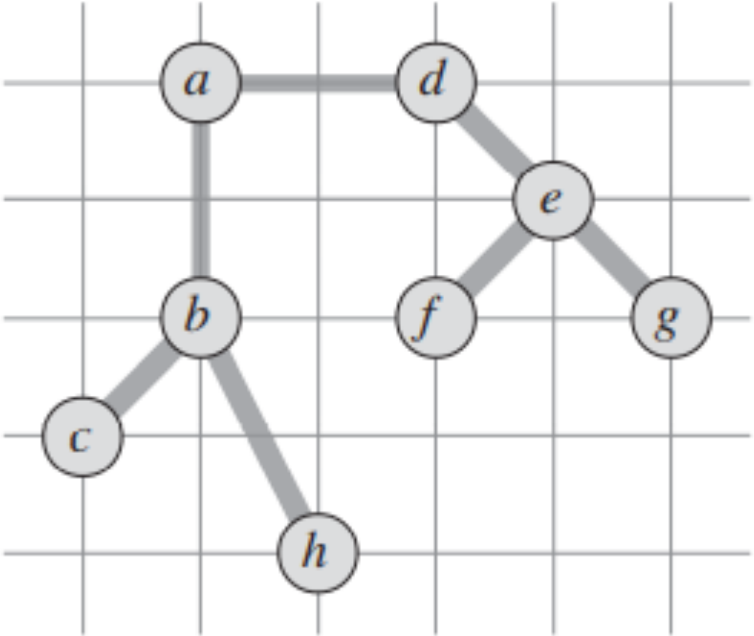
APPROX-TSP-TOUR (G, c)

1. select a vertex $r \in G.V$ to be a "root" vertex
2. compute a minimum spanning tree T for G from root r using MST-PRIM. (G c r)
3. let H be a list of vertices, ordered according to when they are first visited in a preorder tree walk of T
4. return the path.

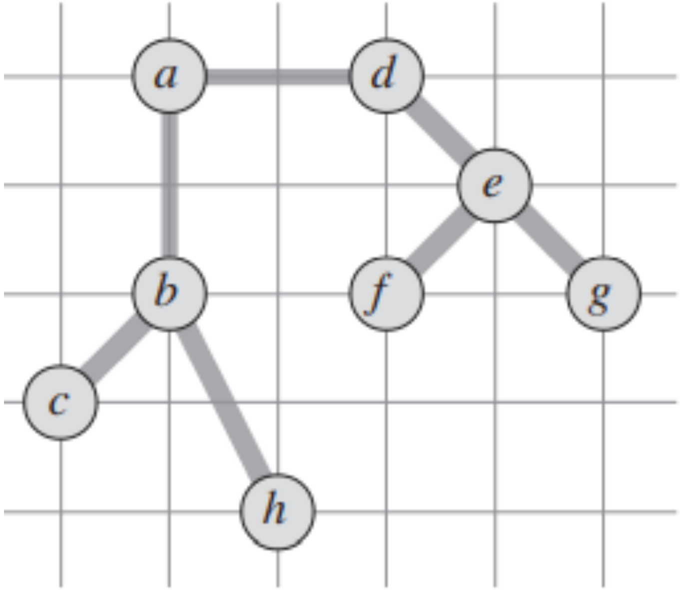
Constructing The Minimum Spanning Tree

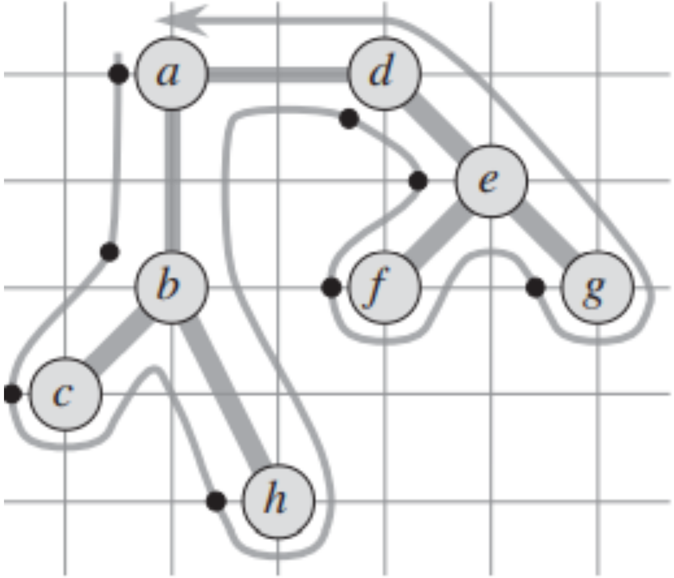
We will be using Prim's Algorithm to construct a minimum spanning tree from the given graph as an adjacency matrix.



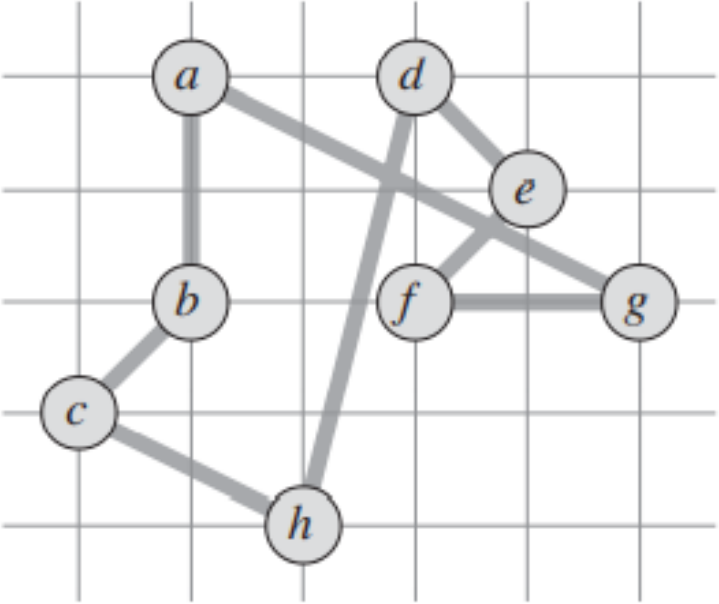


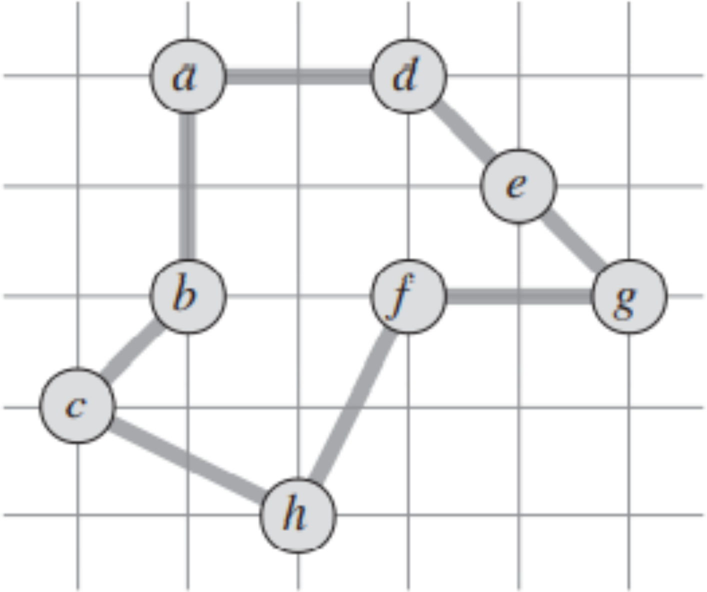
Getting the preorder walk/ Depth first search walk



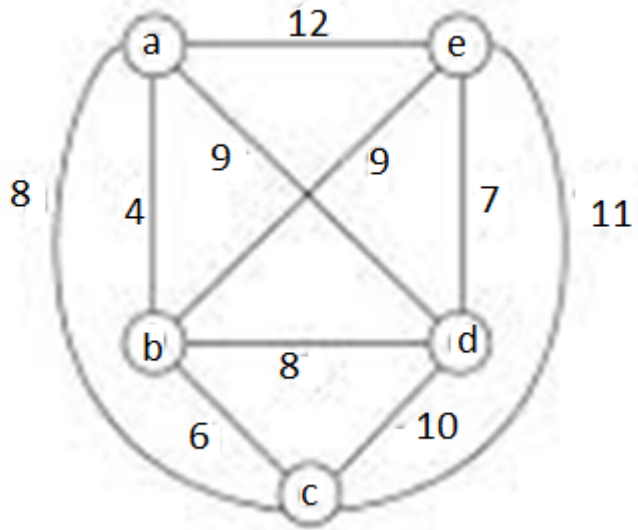


The final Path





Example



The set-covering problem

In the set covering problem, two sets are given:
A set U of elements and
A set S of subsets of the set U .

$$U = \bigcup_{x \in S} x$$

Each subset in S is associated with a predetermined cost, and the union of all the subsets covers the set U

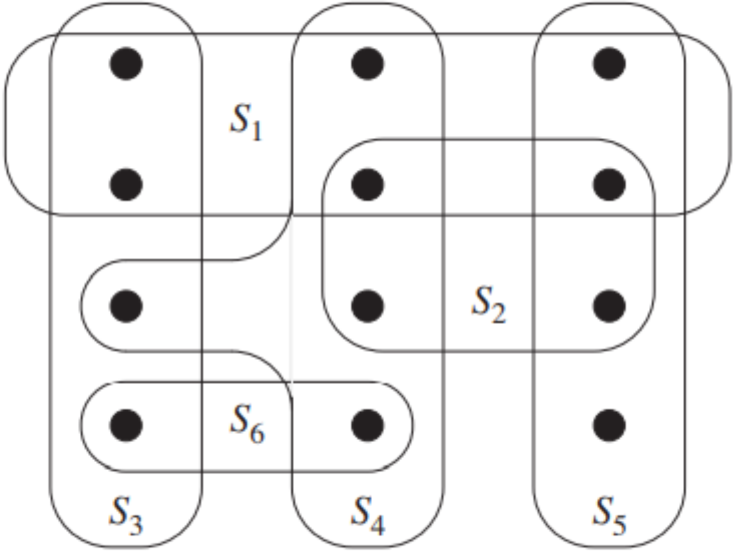
A greedy approximation algorithm

The greedy method works by picking, at each stage, the set S that covers the greatest number of remaining elements that are uncovered.

GREEDY-SET-COVER$(X, \mathcal{F})$

$U = X$
$\mathcal{C} = \emptyset$
while $U \neq \emptyset$
  select an $S \in \mathcal{F}$ that maximizes $|S \cap U|$
  $U = U - S$
  $\mathcal{C} = \mathcal{C} \cup \{S\}$
return $\mathcal{C}$

Example



X consists of the 12 black points and
$S = \{S_1, S_2, S_3, S_4, S_5, S_6\}$

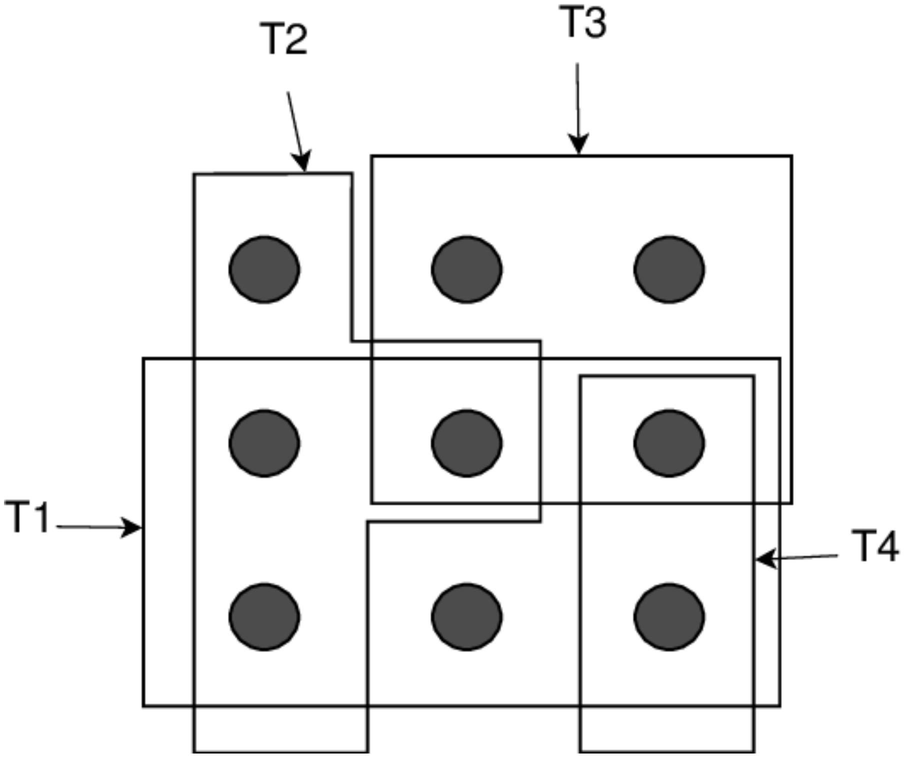
S_1

S_4

S_5

S_3

Example



T_1

T_3

T_2

X consists of the 9 black points and
S $= \{T_1, T_2, T_3, T_4\}$