MACM 201 - Discrete Mathematics

Graph Theory 4 - spanning trees

Department of Mathematics

Simon Fraser University

Spanning trees

Definition

Let G be a multigraph. A subgraph T of G is a **spanning tree** if it is spanning (so T contains all vertices in G) and is a tree.

Example:

Existence of spanning trees

Theorem

Every connected multigraph $G = (V, E)$ has a spanning tree. Two ways to select one:

(1) Start from G. If there is a cycle C in the graph, delete an edge from C. Repeat until there is no cycle left.

(2) Start from the empty graph on V. Repeatedly add edges of G to your graph while staying acyclic. Repeat until no edge can be added.

Proof (sketch)

Note

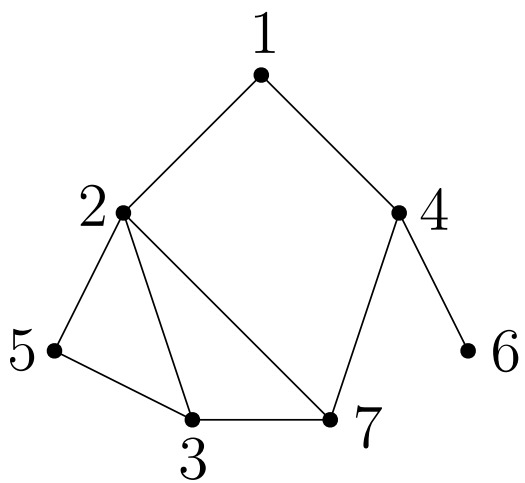
In many practical applications we are interested in finding not just some spanning tree of a graph, but one with some features that better reflect the structure of the original graph.

Depth-First Search

The Depth-First Search (DFS) algorithm (to construct a spanning tree)
`Input.` A graph $G = (V, E)$, $V = \{1, 2, \ldots, n\}$
`Output.` A set E_T of edge such that (V, E_T) is a spanning tree of G.

1. `Let` $v = 1$ and $E_T = \emptyset$. Mark 1 as visited.
2. `Let` i be the smallest neighbor of v that has not been visited.
   a) Mark i as visited.
   b) Add the edge $\{v, i\}$ to E_T and call v the parent of i in T, denoted $v = p_T(i)$.
   c) `Let` $v = i$
   d) `Return` to step 2.
3. `If` all neighbors of v have been visited `Then`
   a) `If` $v = 1$ `Then` stop (E_T contains $n - 1$ edges)
   b) `Else` return to step 2 with v replaced by its parent $p_T(v)$ (this step is the *backtrack step*).

Small Example.



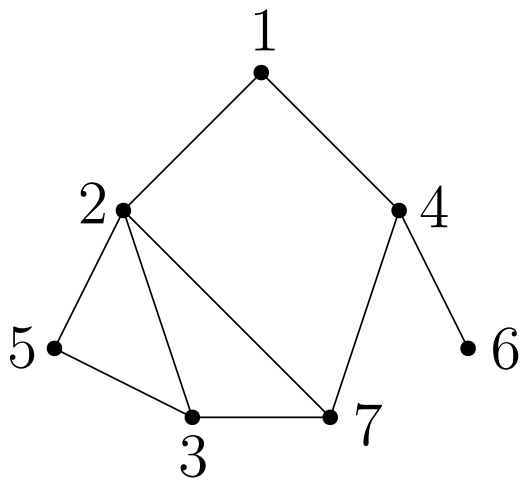
Breadth-First Search

This algorithm requires an additional data structure, a *queue* Q, such that vertices enter Q by the end and exit it by the beginning (this is the definition of a queue, think about when you take the bus: first in, first out).

The Breadth-First Search (BFS) algorithm (to construct a spanning tree)
`Input.` A graph $G = (V, E)$, $V = \{1, 2, \ldots, n\}$
`Output.` A set E_T of edge such that (V, E_T) is a spanning tree of G.

1. `Let` $v = 1$, $E_T = \emptyset$ and $Q = \emptyset$. Insert 1 in Q and mark 1 as visited.
2. `While` Q is not empty `Do`
   a) `Let` v be the vertex at the beginning of Q (the first vertex in Q).
   b) Remove v from Q
   c) `For` each unvisited neighbor i of v `Do`
      i. Insert i at the end of Q and mark i as visited.
      ii. Add the edge $\{v, i\}$ to E_T.

Small Example.



Weighted graphs

We let $\mathbb{R}^+$ denote the set of all nonnegative real numbers.

Definition

Let $G = (V, E)$ be a multigraph. A function $w : E \to \mathbb{R}^+$ is called a **edge-weighting**. For any subgraph $H = (V', E')$ of G we define the **weight** of H to be

$$w(H) = \sum_{e \in E'} w(e)$$

so the weight of a subgraph is the sum of the weights of its edges.

Definition

A **minimum spanning tree** is a spanning tree of G whose weight is minimum among all spanning trees of G.

Example: Consider a graph G with a vertex for each major city and add an edge between two cities if they are directly joined by a major road. Define an edge-weighting by the rule that the weight of an edge indicates the travel time between the cities.

Application: Analysis of a pathogen outbreak

BC Center for Disease Control, has sequenced the genomes of the bacteria responsible for the disease tuberculosis (*Mycobacterium tuberculosis*) taken from a few hundred of patients of an outbreak of tuberculosis (TB) in BC. We want to see if we can cluster the patients into groups, where in each group, all patients are likely to have been infected with the same strain of the pathogen (i.e. the same source).

This has important applications as not all strains of TB are treated the same way: some are resistant to first-line antibiotics and need to be treated with stronger antibiotics, but then you want to be careful that they do not acquire resistance to these stronger antibiotics and then infect other patients and resistance spreads in the population and this is a very serious problem.

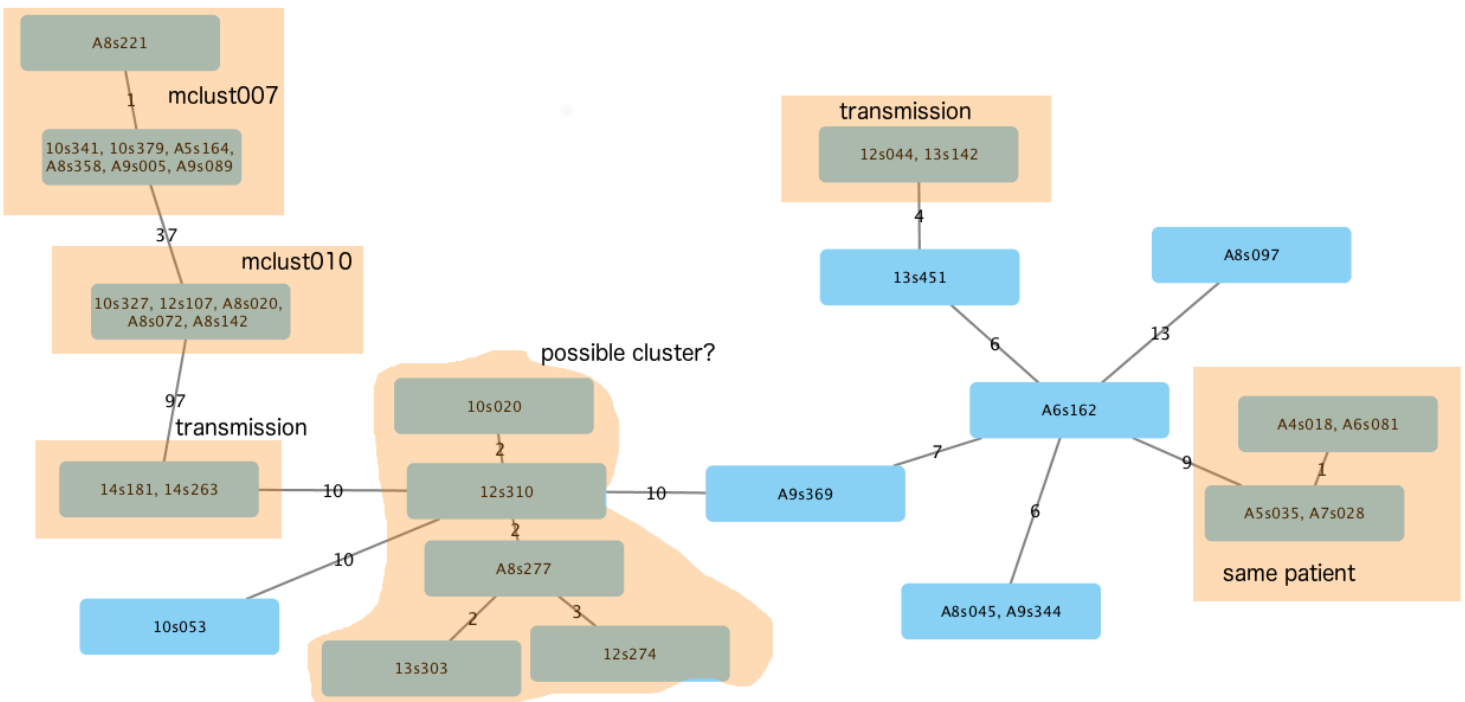
The techniques we use rely on advanced tools in computational biology (to detect mutations in some – a few hundred – selected genes of *Mycobacterium tuberculosis* in the patients). But once we have this set of mutations for all our patients, we use two simple tools we did see in MACM201: the Hamming distance and minimum spanning trees.

Application: Analysis of a pathogen outbreak

We do the following:

a) We build a complete graph $G = (V, E)$ where each vertex E represents a patient and the edge between two vertices x and y is weighted by the Hamming distance between the genomes of *Mycobacterium tuberculosis* sequenced in both patients.

b) We compute a minimum spanning tree in this graph, and from this tree, we can easily see (visualize) potential clusters of patients infected by pathogens whose genomes have very similar genomes, forming the clusters we are looking for.



Kruskal's algorithm to compute a minimum spanning tree

Input: a connected multigraph $G = (V, E)$ with an edge-weighting.
Output: a minimum spanning tree of G.

1. Set $E^* = \emptyset$
2. *While* (V, E^*) is not connected:
   Choose an edge e of minimum weight so that $(V, E^* \cup \{e\})$ does not contain a cycle and add e to E^*
3. Output the tree (V, E^*)

Example.