Chapter 6. Network Models

Note. In this chapter, we describe four algorithms related to graph theory. We consider (1) finding a minimal spanning tree of a graph, (2) the shortest-route algorithm, (3) the maximal flow algorithm, and (4) the Critical Path Method (CPM) algorithm. Applications of these algorithms include (respectively) (1) constructing a network connecting given locations in such a way as to minimize distance/cost (Taha gives the example of connecting natural-gas wellheads with a network of pipelines), (2)finding a shortest route between *two* cities in a given network of roads, (3) determining the maximum flow of a fluid though a network of connected pipelines, and (4) determining a time schedule for the activities of a construction project.

Section 6.1. Scope and Definition of Network Models.

Note. In this section, we give a few definitions used by Taha. Unfortunately, Taha uses the term "network" to indicate both a directed graph and an undirected graph. We do our best to follow Taha's terminology, while relating the operations research terminology to the corresponding traditional (and unambiguous) graph theoretic terminology. We give links to the relevant online graph theory notes.

Definition. A *network*, (N, A), is a set N of *nodes* and a set A of *arcs* (or "*branches*"), where the arcs are ordered pairs of nodes.

Example 6.1.A. Suppose $N = \{1, 2, 3, 4, 5\}$ and $A = \{(1, 2), (1, 3), (2, 3), (2, 5), (3, 4), (3, 5), (4, 2), (4, 5)\}$. Then the network (N, A) can be represented with the following drawing:

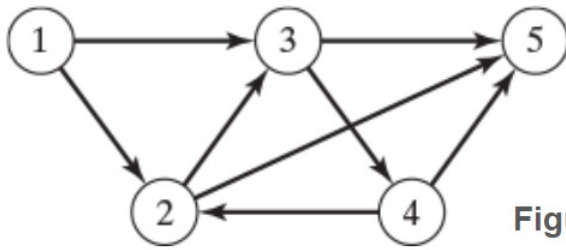

Figure 6.1

So at this point, the term "network" corresponds to the term s "directed graph" and "digraph" form graph theory; see my online notes of Introduction to Graph Theory (MATH 4347/5347) on Section 6.2. Conservative Graphs or my online notes for Graph Theory 1 (MATH 5340) on Section 1.5. Directed Graphs. Taha mentions the term "flow" at this stage, but we try to avoid this undefined term for now, and save this concept for Section 6.4. Maximal Flow Model.

Definition. An arc of a network (N, A) is *directed* (or *oriented*) if for any two nodes $n_1, n_2 \in N$ we have at most one of arcs (n_1, n_2) and (n_2, n_1) in A. A *directed network* has all arcs directed. A *path* is a set of arcs joining two distinct nodes, padding through other nodes in the network (according to the direction of the arcs). A path is a *cycle* (or *loop*) if it connects a node back to itself through other nodes.

Example 6.1.B. In the network represented by Figure 6.1, the arcs $(1, 2), (2, 3), (3, 4), (4, 5)$ form a path joining node 1 to node 5. The arcs $(2, 3), (3, 4), (4, 2)$ form a cycle in the network.

Note. Taha is vague with his following definitions. There is a blurring of the distinction between arcs and directed arcs. These ideas are clearly defined in the setting of directed graphs in my online notes for Graph Theory 1 (MATH 5340) on Section 1.5. Directed Graphs. Notice in particular the definitions of indegree, outdegree, source, sink, directed path, and directed cycle.

Definition. A network is *connected* if every two distinct nodes are joined by some path in the network.

Note. Taha states (page 218) that the network represented in Figure 6.1 is connected. This is the case if we ignore the directions on the arcs, but there is not a path from node 5 to any other node in the network. We will mostly be interested in directed graphs and often will put weights on the arcs in this chapter, so we do our best to combine Taha's terminology with more traditional graph theoretic terminology.

Note. For the remainder of this section we consider "graphs," as opposed to "networks," in which the directions on the arcs are ignored (and are called "edges" of the graph, represented by sets of nodes of size two instead of ordered pairs of nodes). Ignoring all the directions in a network, we get the "underlying graph" of the network (as defined in the online notes on directed graphs mentioned above).

Definition. A *tree* is a cycle-free connected graph. A *spanning tree* in a graph is a subgraph that is a tree and has the same nodes as the original graph.

Note. If we ignore the directions on the arcs of the network in Figure 6.1 (that is, if we consider the underlying graph of this network) then the graph in Figure 6.2 (left) is a tree in the underlying graph. The graph in Figure 6.2 (right) is a spanning tree in the underlying graph. Notice that the idea of a spanning tree only makes sense in the setting of a "larger" graph on a given set of nodes. Such a larger graph is not necessary for the idea of a "tree" (though such a larger graph may be part of the conversation, as it is here).

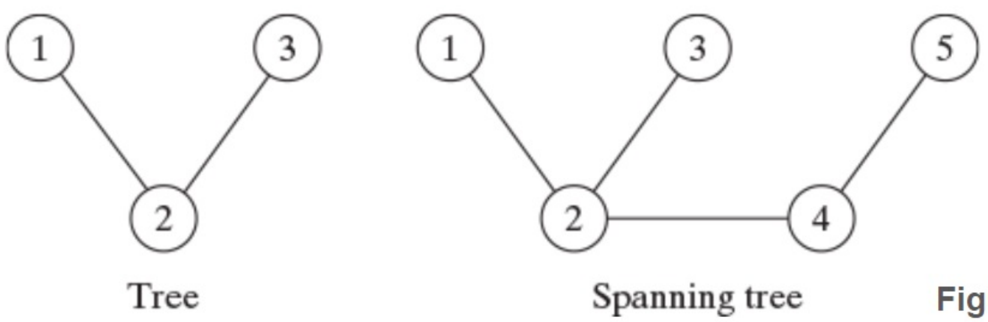

Figure 6.2

Example 6.1.1. We now consider an example that has a prominent role in the historical development of graph theory. The history is discussed in my online notes for Introduction to Graph Theory (MATH 4347/5347) on Section 3.1. Eulerian Circuits and Graph Theory (MATH 5430) on Section 3.3. Euler Tours. The "Bridges of Königsberg" problem is based on seven bridges that join two islands and the mainland in the Pergel River in the town of Königsberg Prussia (today, this is the city of Kaliningrad, Russia). See Figure 6.3. The question arose as to whether it was possible to make a round-trip in such a way that each bridge is crossed exactly once. The most prolific mathematician of all times, Leonhard Euler (April 15, 1707–September 18, 1783), converted this puzzle into a graph theory problem. He introduced a node for each landmass and used edges to create the graph given in Figure 6.4. Euler then translated the Königsberg Bridge Problem into the existence of a "walk" in the graph of Figure 6.4 that includes each edge of the graph exactly once and begins and ends at the same node; such a walk is now called an "Eulerian circuit." Once stated in this way, it is rather clear that no such Eulerian circuit exists (consider the number of edges at each node; there is an odd number in each case, eliminating the possibility of visiting a landmass on one bridge, then leaving on another bridge to return to your starting point). Stated in graph theory terms, since each node is of off "degree," an Eulerian circuit does not exist.

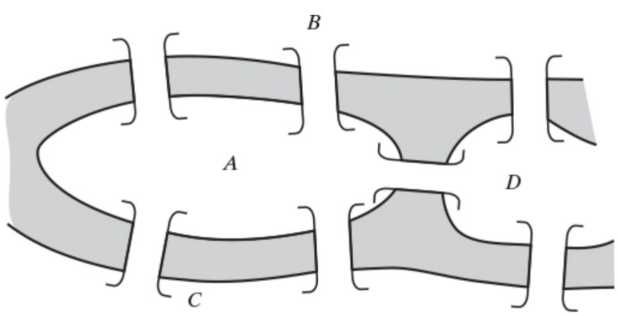

Figure 6.3 Bridges of Königsberg

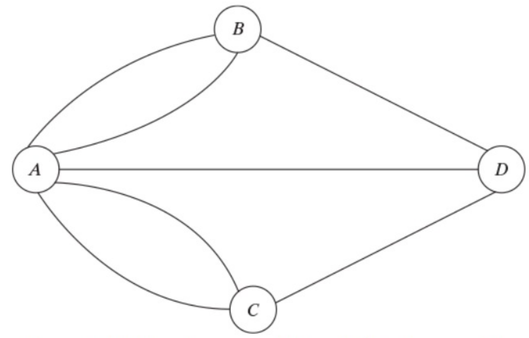

Figure 6.4 Network representation of Königsberg problem

Note. For accuracy and rigor, we now give the formal definition of a network, as stated in Mathematical Modeling Using Graph Theory (MATH 5870). "Network" is defined in Section 7.1. Transportation Networks (notes are currently [spring 2022] still in preparation for this course) as follows. Notice that the term "vertex" is used in place of "node."

Definition. A *network* $N = (x, y)$ is a digraph D (the *underlying digraph* of N) with two distinguished vertices, a *source* x and a *sink* y, together with a nonnegative real valued function c, called the *capacity function*, defined on its arc set A. The value of c on arc a is the *capacity* of a. The vertices other than the source and sink are *intermediate vertices*.

Note. Notice that the object given in Figure 6.1 satisfies part of this definition of a network. The source is 1 and the sink is 5. The intermediate vertices are those in set $I = \{2, 3, 4\}$. However, a capacity function is not given, so it does not completely satisfy this rigorous definition of a network.

Revised: 5/3/2022