NETWORK DATA STRUCTURES

Network data structures for geographic information science (GISci) are methods for storing network data sets in a computer in order to support a range of network analysis procedures. Network data sets are among the most common in GISci and include transportation networks (e.g., road or railroads), utility networks (e.g., electricity, water, and cable networks), and commodity networks (e.g., oil and gas pipelines), among many others. Network data structures must store the edge and vertex features that populate these network data sets, the attributes of those features, and, most important, the topological relationships among the features. The choice of a network data structure can significantly influence one's ability to analyze the processes that take place across networks. This entry describes the mathematical basis for network data structures and reviews several major types of network data structures as they have been implemented in geographic information systems (GIS).

Graph Theoretic Basis of Network Data Structures

The mathematical subdiscipline that underlies network data structures is termed *graph theory.* Any *graph* or *network* (the terms are used interchangeably in this context) consists of connected sets of edges and vertices. *Edges* may also be referred to as *lines* or *arcs,* and *vertices* may be termed *junctions, points,* or *nodes.* Within graph theory, there are methods for measuring and comparing graphs and principles for proving the properties of individual graphs or classes of graphs.

Graph theory is not concerned with the shape of the features that constitute a network, but rather with the topological properties of those networks. The topological invariants of a graph are those properties that are not altered by elastic deformations (such as a stretching or twisting). Therefore, properties such as connectivity, adjacency, and incidence are topological invariants of networks, since they will not change even if the network is deformed by a cartographic process. The permanence of these properties allows them to serve as a basis for describing, measuring, and analyzing networks.

Graph theoretic descriptions of networks can include statements of the number of features in the network, the degree of the vertices of the graph (where the degree of a vertex is the number of edges incident to it), or the number of cycles in a graph. Descriptions of networks can also be based on structural characteristics of graphs, which allow them to be grouped into idealized types. Perhaps the most familiar type is *tree networks,* which have edge "branches" incident to nodes, but no cycles are created by the connections among those nodes. River networks are nearly always modeled as tree networks. Another common idealized graph type is the *Manhattan network,* which is made up of edges intersecting at right angles. This creates a series of rectangular "blocks" that approximate the street networks common in many U.S. cities. Other idealized types include bipartite graphs and hub-and-spoke networks.

If one wishes to quantitatively measure properties of graphs rather than simply describe them, there is a set of network indices for that purpose. The simplest of these is the *Beta index,* which measures the connectivity of a graph by comparing the number of edges to the number of vertices. A more connected graph will have a larger Beta index ratio, since relatively more edges are connecting the vertices. The Alpha and Gamma indices of connectivity compare proven properties of graphs with observed properties. The *Alpha index* compares the maximum possible number of fundamental cycles in the graph to the actual number of fundamental cycles in the graph. Similarly, the *Gamma index* compares the maximum possible number of edges in a graph to the actual number of edges in a graph. In each case, as the latter measure approaches the former, the graph is more completely connected. Other measures exist for applied instances of networks and consequently depend on nontopological properties of the network. The reader is directed to textbooks on the topic of graph theory for a more comprehensive review of these and other more advanced techniques.

Implementations of Network Data Structures in GIS

Nontopological Data Structures

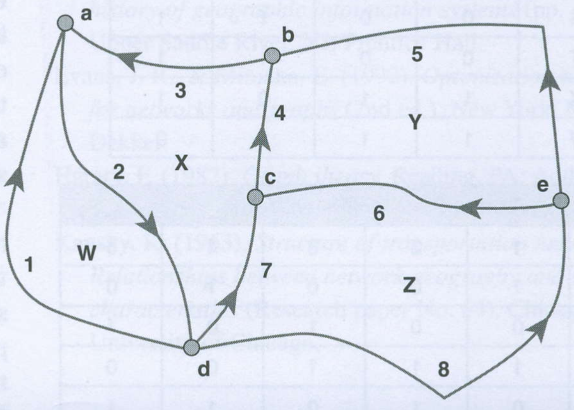
While the graph theoretic definition of a network remains constant, the ways in which networks are structured in computer systems have changed dramatically over the history of GISci. The earliest computer-based systems for automated cartography stored network edges as independent records in a database. Each record contained a starting and ending point for the edge, and the edge was defined as the connection between those points. Attribute fields could be associated with each record, and some implementations included a link from each record to a list of "shape points" that defined curves in the edges. These records did not contain any information regarding the topological properties of the edges and was therefore termed the *nontopological structure* (colloquially known as the "spaghetti" data model).

The advantages of nontopological data models include the fact that they are easy to understand and implement, they provide a straightforward platform for the capture of spatial data through digitizing, and they are efficient in terms of display for cartographic purposes. This latter advantage led to the wide acceptance of this data structure among computer-aided drafting software packages. The disadvantages of nontopological data structures include the tendency for duplicate edges to be captured, particularly coincident boundaries of polygonal features. This, in turn, leads to sliver errors, where duplicate edges are not digitized in precisely the same way. Most important for the discussion here, the lack of topological information in these data structures makes them essentially useless for network analysis. Even the most basic graph theoretic measures require knowledge of the connectivity of edges and vertices.

Due to these disadvantages, nontopological data models were essentially abandoned in mainstream GIS, but a variant data structure became extremely popular in the mid 1990s and has remained so to the present. The *shapefile* is a nontopological data structure developed by the Environmental Systems Research Institute (ESRI). The shapefile was designed primarily to allow for rapid cartographic rendering of large sets of geographic features, and the structure performs admirably in that respect. Although topological relationships are not explicitly stored in this data structure, some specialized tools have been developed that compute such relationships "on the fly" in order to support some editing and query functions. Although it is possible to complete some network analysis using this structure with customized tools, it is generally considered to be an inefficient data structure for network analysis.

Topological Data Models

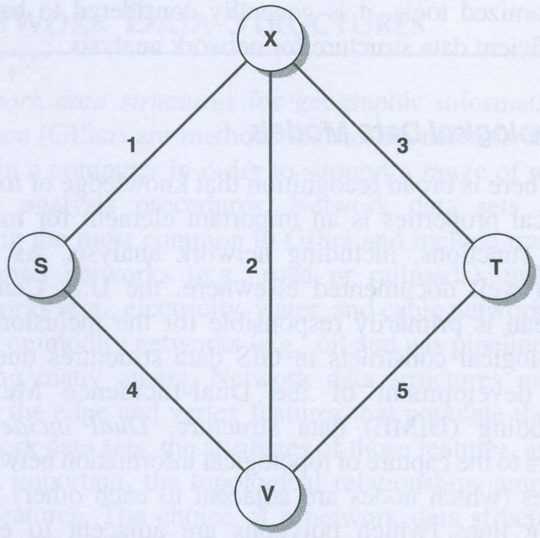
There is broad recognition that knowledge of topological properties is an important element for many GIS functions, including network analysis. As has been well documented elsewhere, the U.S. Census Bureau is primarily responsible for the inclusion of topological constructs in GIS data structures due to the development of the Dual-Incidence Matrix Encoding (DIME) data structure. *Dual incidence* refers to the capture of topological information between nodes (which nodes are adjacent to each other) and along lines (which polygons are adjacent to each other). Figure 1 provides a graphic and tabular view of



Edge ID	From Node	To Node	Left Polygon	Right Polygon
1	d	a	Null	W
2	a	d	X	W
3	b	a	X	Null
4	c	b	X	Y
5	e	b	Y	Null
6	e	c	Z	Y
7	d	c	X	Z
8	d	e	Z	Null

Polygon ID	Number of Edges	List of Edges
W	2	1, 2
X	4	2, 7, 4, 3
Y	3	4, –5, 6
Z	3	6, –7, 8

Figure 1 Dual-Incidence Data Structure



Vertex Adjacency Matrix

Vertices	S	T	X	V
S	0	0	1	1
T	0	0	1	1
X	1	1	0	1
V	1	1	1	0

Vertex-Edge Incidence Matrix

	1	2	3	4	5
S	1	0	0	1	0
T	0	0	1	0	1
X	1	1	1	0	0
V	0	1	0	1	1

Figure 2 Matrix-Based Network Data Structure

Table 1 Star Data Structure Vertex List

Vertex List

Vertices	S	T	X	V	Null
Pointers	1	3	5	8	11

Table 2 Star Data Structure Adjacency List

Adjacency List

Pointer	1	2	3	4	5	6	7	8	9	10	11
Adjacency	X	V	X	V	S	T	V	S	T	X	0

how lines and polygons are stored in this topological data structure.

The DIME data structure evolved into the structure employed for the Topologically Integrated Geographic Encoding and Referencing (TIGER) files that are still used by the Census Bureau to delineate population tabulation areas. There are many advantages of the dual-incidence data structure, and the wide acceptance of the data structure combined with the comprehensive nature of the TIGER files led to its status as the de facto standard for vector representations in GIS. Two elements of this advance profoundly influenced the ability to conduct network analysis in GIS. First, the DIME structure captures incidence, which is one of the primary topological properties defining the structure of networks. As can be seen in Figure 1, all edges that are incident to a given point can be determined with a simple database query. Second, many of the features captured by the Census Bureau were streets or other transportation features. Since the Census Bureau has a mandate that covers the entire United States, this meant that a national transportation database was available for use in GIS, and this database was captured in a structure that could support high-level network analysis.

However, the dual-incidence topological data model also imposes some difficult constraints on network analysts. The Census Bureau designed the data structure in order to well define polygons with which populations could be associated. To do so, the data model had to enforce planarity. Planar graphs are those that can be drawn in such a way that no two edges cross without a vertex at that location. Thus, at every location where network features cross, a point must exist in the database. This is true regardless of whether or not a true intersection exists between the network features, and it is most problematic when modeling bridges or tunnels. While network features certainly cross each other at bridges, there is no incidence between the features, and network analysis should not permit flow between features at that point. Moreover, since planar enforcement demands that network features (such as roads) be divided at every intersection, a road that may be commonly perceived and used as a single feature must be represented as a series of records in the data structure. This repetition can increase the database size many times over and can encourage errors in the database when these multiple features are assigned attribute values.

Pure Network Data Models

The limitations on the ability to perform network analysis imposed when using common GIS data structures have necessitated the development of *pure network data models*. These include nonplanar data structures that relax planarity requirements in order to more realistically model real-world networks, data structures that support turns and directional constraints on edges in order to model the impedances encountered when moving between and along network features, and perhaps most important, data structures that allow more efficient operation of network analysis procedures.

For many network operations, it is preferable to store the topological properties of the network with matrix representations. For the network shown in Figure 2, the vertex adjacency matrix and vertex edge incidence matrix are provided.

Matrix data structures allow for intuitive and rapid query of network topological properties. However, when the network is sparse (relatively few edges connecting the vertices), the matrix may require a great deal of storage space to capture a small amount of topological information. In these cases, list-based data structures, such as the star data structure, may be preferable. The star data structure is based on two lists. The first is a list of the vertices with a pointer to a second list. The second list holds a continuous string of adjacencies for each of the vertices. The star data structure for the graph in Figure 2 consists of the *vertex list* (see Table 1) and the *adjacency list* (see Table 2).

From these two arrays, adjacency information can be found without storing extraneous information. This structure has also proven to be the most efficient structure for many network algorithms that depend on searching for arcs from a given node.

The Future of Network Data Structures

Advances in network data structures for GISci are continually occurring. The recent past has seen the development of object-oriented data structures, the introduction of dynamic networks, and the recognition that highly complex network structures are applicable to a diverse set of disciplines. One can expect to see these advances increasingly integrated with GIS and GISci.

Kevin M. Curtin

See also Census; Database Design; Database, Spatial; Data Modeling; Data Structures; Geographic Information Science (GISci); Geographic Information Systems (GIS); Network Analysis; Representation; Spatial Analysis; TIGER; Topology

Further Readings

Cooke, D. F. (1998). Topology and TIGER: The Census Bureau's contribution. In T. W. Foresman (Ed.), *The history of geographic information systems* (pp. 47–58). Upper Saddle River, NJ: Prentice Hall.

Evans, J. R., & Minieka, E. (1992). *Optimization algorithms for networks and graphs* (2nd ed.). New York: Marcel Dekker.

Harary, F. (1982). *Graph theory.* Reading, PA: Addison-Wesley.

Kansky, K. (1963). *Structure of transportation networks: Relationships between network geography and regional characteristics* (Research paper No. 84). Chicago, IL: University of Chicago.

~~NEURAL NETWORKS~~

~~*Artificial neural networks (ANN)* are pattern detection and classification tools loosely modeled on networks of neurons in human or animal~~ brains. ~~The term *neural network* is used in contexts such as~~ GIS, ~~where there is unlikely to be any confusion with actual physiological neural~~ networks. ~~This entry outlines the basic concept behind the design of neural networks and reviews aspects of their network structures before considering more practical~~ aspects, ~~such as network~~ training, ~~and issues relevant to their use in typical~~ applications.

~~Background and Definition~~

~~A neural network consists of an interconnected set of artificial *neurons*. Each neuron has a number of inputs and one output and converts combinations of~~ input