a different path, its accumulative cost must be recomputed. The lowest accumulative cost is then assigned to the reactivated cell. This process continues until all cells in the output raster are assigned with their least accumulative costs to the source cell.

Figure 17.4 illustrates the cost distance measure operation. Figure 17.4*a* shows a raster with the source cells at the opposite corners. Figure 17.4*b* represents a cost raster. To simplify the computation, both rasters are set to have a cell size of 1. Figure 17.4*c* shows the cost of each lateral link and the cost of each diagonal link. Figure 17.4*d* shows for each cell the least accumulative cost. Box 17.2 explains how Figure 17.4*d* is derived.

A cost distance measure operation can result in different types of outputs. The first is a least accumulative cost raster as shown in Figure 17.4*d*. The second is a direction raster, showing the direction of the least-cost path for each cell. The third is an allocation raster, showing the assignment of each cell to a source cell on the basis of cost distance measures. The fourth type is a shortest path raster, which shows the least cost path from each cell to a source cell. Using the same data as in Figure 17.4, Figure 17.5*a* shows two examples of the least-cost path and Figure 17.5*b* shows the assignment of each cell to a source cell. The darkest cell in Figure 17.5*b* can be assigned to either one of the two sources.

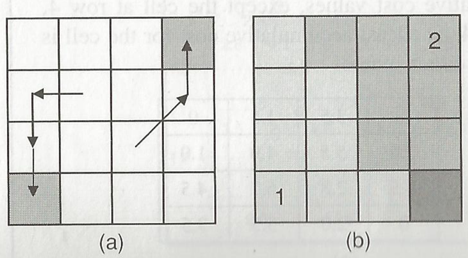


Figure 17.5
The least-cost path (*a*) and the allocation raster (*b*) are derived using the same input data as in Figure 17.4.

17.1.5 Options for Least-Cost Path Analysis

The outcome of a least-cost path analysis is directly influenced by the selection of cost factors and, perhaps more importantly, the weighting of each factor. This is why, in some studies (e.g., Atkinson et al. 2005; Choi et al. 2009), least-cost path analysis has often been integrated with multicriteria evaluation (Malczewski 2006), a topic to be discussed in detail in Chapter 18.

When the terrain is used for deriving the least-cost path, the surface is typically assumed to be uniform for all directions. But in reality, the terrain changes in elevation, slope, and aspect in different directions. To provide a more realistic analysis of how we traverse the terrain, the "surface distance" can be used (Collischonn and Pilar 2000; Yu, Lee, and Munro-Stasiuk 2003). Calculated from an elevation raster or a digital elevation model (DEM), the surface distance measures the ground or actual distance that must be covered from one cell to another. The surface distance increases when the elevation difference (gradient) between two cells increases. Is it accurate to calculate distances from a DEM? According to Rasdorf et al. (2004), the surface distances calculated from a DEM along highway centerlines compare favorably with distances obtained by driving cars equipped with a distance measurement instrument. In addition to the surface distance, the vertical factor (i.e., the difficulty of overcoming the vertical elements such as uphill or downhill) and the horizontal factor (i.e., the difficulty of overcoming the horizontal elements such as crosswinds) can also be considered. ArcGIS uses the term **path distance** to describe the cost distance based on the surface distance, vertical factor, and horizontal factor.

17.2 APPLICATIONS OF LEAST-COST PATH ANALYSIS

Least-cost path analysis is useful for planning roads, pipelines, canals, transmission lines, and trails. As examples, least-cost path analysis has

been used by Rees (2004) to locate footpaths in mountainous areas, by Atkinson et al. (2005) to derive an arctic all-weather road, and by Snyder et al. (2008) to find trail location for all-terrain vehicles.

Least-cost path analysis can also be applied to wildlife movements. A common application in wildlife management is corridor or connectivity study (Kindall and van Manen 2007; Falcucci et al. 2008). In such studies, the source cells represent habitat concentration areas and the cost factors typically include vegetation, topography, and human activities such as roads. Analysis results can show the least costly routes for wildlife movement.

Least-cost path analysis is important for accessibility studies such as accessibility to medical services (e.g., Coffee et al. 2012). Accessibility is also a concept underlying studies of food deserts (Box 17.3).

17.3 NETWORK

A **network** is a system of linear features that has the appropriate attributes for the flow of objects. A road system is a familiar network. Other networks include railways, public transit lines, bicycle paths, and streams. A network is typically topology-based: lines meet at intersections, lines cannot have gaps, and lines have directions.

Because many network applications involve road systems, a discussion of a road network is presented here. It starts by describing the geometry and attribute data of a road network including link impedance, turns, one-way streets, and overpasses/underpasses. Then it shows how these data can be put together to form a street network for a real-world example.

17.3.1 Link and Link Impedance

A **link** refers to a road segment defined by two end points. Also called edges or arcs, links are the basic geometric features of a network. **Link impedance** is the cost of traversing a link. A simple measure of the cost is the physical length of the link. But the length may not be a reliable measure of cost, especially in cities where speed limits and traffic conditions vary significantly along different streets. A better measure of link impedance is the travel time estimated from the length and the speed limit of a link. For example, if the speed limit is 30 miles per hour and the length is 2 miles, then the link travel time is 4 minutes ($2/30 \times 60$ minutes).

There are variations in measuring the link travel time. The travel time may be directional: the travel time in one direction may be different from that in the other direction. In that case, the directional travel time can be entered separately

Box 17.3 Least-Cost Path Analysis for Studies of Food Deserts

Food deserts refer to socially deprived areas within cities that have poor access to supermarkets for reasonably priced foods, especially healthy foods (i.e., fruits, vegetables, and cereals). Accessibility measures, based on either raster or vector data, are therefore an important part of food desert studies. In their study of food deserts in Lawrence, Kansas, Hallett and McDermott (2011) first create a raster representation of the road network and give each cell a cost that represents the driving or walking cost of traversing the cell. Least-cost path analysis is then applied to calculate the total cost of travel from any cell to the nearest full-service grocery. The study defines a food desert as an area in which the total cost of travel is more than 10 percent of one's food budget.

in two fields (e.g., from-to and to-from). The travel time may also vary by the time of the day and by the day of the week, thus requiring the setup of different network attributes for different applications.

17.3.2 Junction and Turn Impedance

A **junction** refers to a street intersection. A junction is also called a node. A turn is a transition from one street segment to another at a junction. **Turn impedance** is the time it takes to complete a turn, which is significant in a congested street network (Ziliaskopoulos and Mahmassani 1996). Turn impedance is directional. For example, it may take 5 seconds to go straight, 10 seconds to make a right turn, and 30 seconds to make a left turn at a stoplight. A negative turn impedance value means a prohibited turn, such as turning the wrong way onto a one-way street.

Because a network typically has many turns with different conditions, a table can be used to assign the turn impedance values in the network. Each record in a turn table shows the street segments involved in a turn and the turn impedance in minutes or seconds. A driver approaching a street intersection can go straight, turn right, turn left, and, in some cases, make a U-turn. Assuming the intersection involves four street segments, as most intersections do, this means at least 12 possible turns at the intersection, excluding U-turns. Depending on the level of details for a study, we may not have to include every intersection and every possible turn in a turn table. A partial turn table listing only turns at intersections with stoplights may be sufficient for a network application.

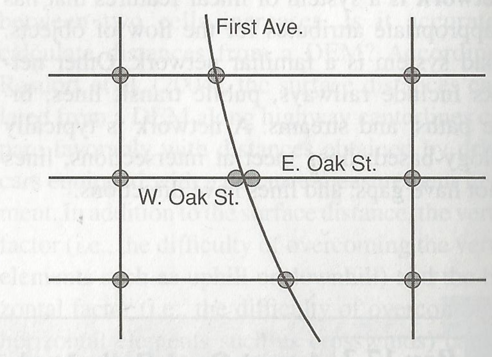
17.3.3 One-Way or Closed Streets

The value of a designated field in a network's attribute table can show one-way or closed streets. For example, the value T can mean that a street segment is one-way; F, it is not one-way; and N, it does not allow traffic in either direction. The direction of a one-way street is determined by the beginning point and the end point of the line segment.

17.3.4 Overpasses and Underpasses

There are at least two methods for including overpasses and underpasses in a network. The first method uses nonplanar features: an overpass and the street underneath it are both represented as continuous lines without a node at their intersection (Figure 17.6). The second method treats overpasses and underpasses as planar features and uses an elevation field to separate the two intersecting lines. In Figure 17.7, a higher "elevation" value (1) is assigned to the junction used by the overpass and a lower value (0) is assigned to the street underneath the overpass.

Figure 17.6
First Ave. crosses Oak St. with an overpass.
A nonplanar representation with no nodes is used at the intersection of Oak St. and First Ave.



Street name	F-elev	T-elev
First Ave.	0	1
First Ave.	1	0
W. Oak St.	0	0
E. Oak St.	0	0

Figure 17.7
First Ave. crosses Oak St. with an overpass.
A planar representation with two nodes is used at the intersection: one for First Ave., and the other for Oak St. First Ave has 1 for the T-elev and F-elev values, indicating that the overpass is on First Ave.

17.4 PUTTING TOGETHER A NETWORK

Putting together a road network involves three tasks: gather the linear features of the network, build the necessary topology for the network, and attribute the network features. Here we use an example from Moscow, Idaho. A university town of 25,000 people, Moscow, Idaho, has more or less the same network features as other larger cities except for overpasses or underpasses.

17.4.1 Gathering Linear Features

The TIGER/Line files from the U.S. Census Bureau are a common data source for making preliminary street networks in the United States. They are free for download at the Census Bureau website (**http://www.census.gov/geo/www/tiger/**). The TIGER/Line files are in shapefile format and measured in longitude and latitude values based on NAD83 (North American Datum of 1983) (Chapter 2). To use it for real-world applications, a preliminary network compiled from the TIGER/Line files must be converted to projected coordinates. Road network data can also be digitized or purchased from commercial companies.

17.4.2 Editing and Building Network

A network converted from the TIGER/Line files has the built-in topology: The streets are connected at nodes and the nodes are designated as either from-nodes or to-nodes. If topology is not available in a data set (e.g., a shapefile or a CAD file), one can use a GIS package to build the topology. ArcGIS, for example, can build network topology using a shapefile or a geodatabase (Box 17.4).

Editing and updating the road network is the next step. When superimposing the road network over orthophotos or high-resolution satellite images, one may find that the street centerlines from the TIGER/Line files deviate from the streets. Such mistakes must be corrected. Also, new streets must be added. In some cases, pseudo nodes (i.e., nodes that are not required topologically) must be removed so that a street segment between two intersections is not unnecessarily broken up. But a pseudo node is needed at the location where one street changes into another along a continuous link (rare but does

Box 17.4 Network Dataset

ArcGIS stores networks as network datasets. A network dataset combines network elements and data sources for the elements. Network elements refer to edges, junctions, and turns. Network sources can be shapefiles or geodatabase feature classes. A shapefile-based network consists of a network dataset and a junction shapefile, both created from a polyline shapefile (e.g., a street network shapefile). Task 3 in the applications section covers a shapefile-based network. A geodatabase network includes both network elements and sources as feature classes in a feature dataset (see Task 4). Unlike a shapefile-based network, a geodatabase network can handle multiple edge sources (e.g., roads, rails, bus routes, and subways) and can connect different groups of edges at specified junctions (e.g., a subway station junction connects a subway route and a bus route). Regardless of its data sources, a network dataset is topological, meaning that the geometric features of edges and junctions must be properly connected. Network Analyst has the Build tool to build the topology (i.e., connectivity) between the network elements.

happen). Without the pseudo node, the continuous link will be treated as the same street. After the TIGER/Line shapefile has been edited and updated, it can be converted to a street network in a GIS.

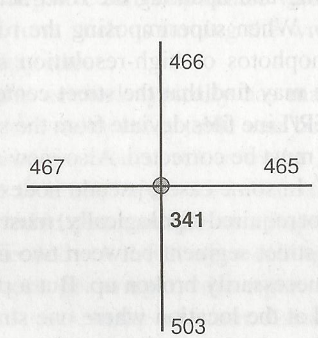
17.4.3 Attributing the Network Features

Network attributes in this example include the link impedance, one-way streets, speed limits, and turns. The link impedance value can be the physical distance or the travel time. The physical distance is the length of a street segment. The travel time can be derived from the length and the speed limit of a street segment. Roads converted from the TIGER/Line files have the field MTFCC (MAF/TIGER feature class code), classifying roads as primary road, secondary road, and so on, which can be used to assign speed limits. Moscow, Idaho, has three speed limits: 35 miles/hour for principal arterials, 30 miles/hour for minor arterials, and 25 miles/hour for all other city streets. With speed limits in place, the travel time for each street segment can be computed from the segment's length and the speed limit.

Moscow, Idaho, has two one-way streets serving as the northbound and the southbound lanes of a state highway. These one-way streets are denoted by the value T in a direction field. The direction of all street segments that make up a one-way street must be consistent and pointing in the right direction. Street segments that are incorrectly oriented must be flipped.

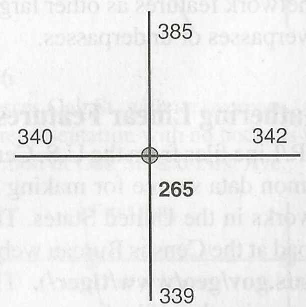
Making a turn table for street intersections with traffic signals is next. Each turn is defined by the edge that the turn starts, the edge that the turn ends, and the turn impedance value in minutes or seconds.

Figure 17.8 shows a street intersection at node 341 with no restrictions for all possible turns except U-turns. This example uses two turn impedance values: 30 seconds or 0.5 minute for a left turn, and 15 seconds or 0.25 minute for either a right turn or going straight. Some street intersections do not allow certain types of turns. For example, Figure 17.9 shows a street intersection at node 265 with stop signs posted only for the east–west traffic. Therefore, the turn impedance values are assigned only to turns to be made from edge 342 and edge 340.



Node#	Arc1#	Arc2#	Angle	Minutes
341	503	467	90	0.500
341	503	466	0	0.250
341	503	465	–90	0.250
341	467	503	–90	0.250
341	467	466	90	0.500
341	467	465	0	0.250
341	466	503	0	0.250
341	466	467	–90	0.250
341	466	465	90	0.500
341	465	503	90	0.500
341	465	467	0	0.250
341	465	466	–90	0.250

Figure 17.8
Possible turns at node 341.



Node#	Arc1#	Arc2#	Angle	Minutes
265	339	342	–87.412	0.000
265	339	340	92.065	0.000
265	339	385	7.899	0.000
265	342	339	87.412	0.500
265	342	340	–0.523	0.250
265	342	385	–84.689	0.250
265	340	339	–92.065	0.250
265	340	342	0.523	0.250
265	340	385	95.834	0.500
265	385	339	–7.899	0.000
265	385	342	84.689	0.000
265	385	340	–95.834	0.000

Figure 17.9
Node 265 has stop signs for the east–west traffic. Turn impedance applies only to turns in the shaded rows.

17.5 Network Analysis

A network with the appropriate attributes can be used for a variety of applications. Some applications are directly accessible through commands in a GIS package. Others require the integration of GIS and specialized software.

17.5.1 Shortest Path Analysis

Shortest path analysis finds the path with the minimum cumulative impedance between nodes on a network. Because the link impedance can be measured in distance or time, a shortest path may represent the shortest route or fastest route.

Shortest path analysis typically begins with an impedance matrix in which a value represents the impedance of a direct link between two nodes on a network and an ∞ (infinity) means no direct connection. The problem is to find the shortest distances (least cost) from a node to all other nodes. A variety of algorithms can be used to solve the problem (Zhan and Noon 1998; Zeng and Church 2009); among them, the most commonly used algorithm is the Dijkstra algorithm (1959).

To illustrate how the Dijkstra algorithm works, Figure 17.10 shows a road network with six nodes and eight links and Table 17.1 shows the travel time measured in minutes between nodes. The value of ∞ above and below the principal diagonal in the impedance matrix in Table 17.1 means no direct path between two nodes. To find the shortest path from node 1 to all other nodes in Figure 17.10, we can solve the problem by using an iterative procedure (Lowe and Moryadas 1975). At each step, we choose the shortest path from a list of candidate paths and place the node of the shortest path in the solution list.

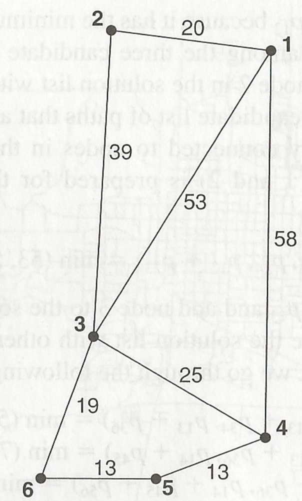


Figure 17.10
Link impedance values between cities on a road network.

The first step chooses the minimum among the three paths from node 1 to nodes 2, 3, and 4, respectively:

$$\min(p_{12}, p_{13}, p_{14}) = \min(20, 53, 58)$$

TABLE 17.1 The Impedance Matrix Among Six Nodes in Figure 17.10

	(1)	(2)	(3)	(4)	(5)	(6)
(1)	∞	20	53	58	∞	∞
(2)	20	∞	39	∞	∞	∞
(3)	53	39	∞	25	∞	19
(4)	58	∞	25	∞	13	∞
(5)	∞	∞	∞	13	∞	13
(6)	∞	∞	19	∞	13	∞

We choose p_{12} because it has the minimum impedance value among the three candidate paths. We then place node 2 in the solution list with node 1.

A new candidate list of paths that are directly or indirectly connected to nodes in the solution list (nodes 1 and 2) is prepared for the second step:

$$\min (p_{13}, p_{14}, p_{12} + p_{23}) = \min (53, 58, 59)$$

We choose p_{13} and add node 3 to the solution list. To complete the solution list with other nodes on the network we go through the following steps:

$$\min (p_{14}, p_{13} + p_{34}, p_{13} + p_{36}) = \min (58, 78, 72)$$
$$\min (p_{13} + p_{36}, p_{14} + p_{45}) = \min (72, 71)$$
$$\min (p_{13} + p_{36}, p_{14} + p_{45} + p_{56}) = \min (72, 84)$$

Table 17.2 summarizes the solution to the shortest path problem from node 1 to all other nodes.

A shortest-path problem with six nodes and eight links is simple to solve. A real road network, however, has many more nodes and links. Zhan and Noon (1998), for example, lists 2878 nodes and 8428 links in the State of Georgia network with three levels of roads (interstate, principal arterial, and major arterial). This is why, over the years, researchers have continued to propose new shortest-path algorithms to reduce the computation time (e.g., Zhen and Church 2009).

Shortest-path analysis has many applications. Perhaps the common application is to help a driver find the shortest route between an origin and a destination. This can be done via a navigation system, either in a vehicle or on a cell phone. Shortest routes are also useful as measures of accessibility. Thus they can be used as input data to a wide range of accessibility studies including park-and-ride facilities (Farhan and Murray 2005), urban trail systems (Krizek, El-Geneidy, and Thompson 2007), food deserts (Apparicio, Cloutier, and Shearmur 2007), and community resources (Witten, Exeter, and Field 2003).

TABLE 17.2 Shortest Paths from Node 1 to All Other Nodes in Figure 17.11

From-Node	To-Node	Shortest Path	Minimum Cumulative Impedance
1	2	p_{12}	20
1	3	p_{13}	53
1	4	p_{14}	58
1	5	$p_{14} + p_{45}$	71
1	6	$p_{13} + p_{36}$	72

17.5.2 Traveling Salesman Problem

The **traveling salesman problem** is a routing problem, which stipulates that the salesman must visit each of the select stops only once, and the salesman may start from any stop but must return to the original stop. The objective is to determine which route, or tour, the salesman can take to minimize the total impedance value. A common solution to the traveling salesman problem uses a heuristic method (Lin 1965). Beginning with an initial random tour, the method runs a series of locally optimal solutions by swapping stops that yield reductions in the cumulative impedance. The iterative process ends when no improvement can be found by swapping stops. This heuristic approach can usually create a tour with a minimum, or near minimum, cumulative impedance. Similar to shortest-path analysis, a number of algorithms are available for the local search procedure, with Tabu Search being the best-known algorithm. For some applications, a time window constraint can also be added to the traveling salesman problem so that the tour must be completed with a minimum amount of time delay.

17.5.3 Vehicle Routing Problem

The vehicle routing problem is an extension of the traveling salesman problem. Given a fleet of vehicles and customers, the main objective of the vehicle routing problem is to schedule vehicle routes and visits to customers in such a way that the total travel time is minimized. Additional constraints such as time windows, vehicle capacity, and dynamic conditions (e.g., traffic congestion) may also exist.

In an early application of GIS to the vehicle routing problem, Weigel and Cao (1999) used a GIS (ARC/INFO) to prepare an origin-destination cost matrix connecting customers and Sears stores as an input to external routing software. After routes were assigned, route maps were generated for dispatch drivers. ArcGIS adopts the same workflow for solving vehicle routing problems. Because routing requires complex modeling applications, the integration of GIS and special routing software in operations research and management science is likely to continue (Keenan 2008; Bräysy, Dullaert, and Nakari 2009).

17.5.4 Closest Facility

Closest facility is a network analysis that finds the closest facility among candidate facilities to any location on a network. The analysis first computes the shortest paths from the select location to all candidate facilities, and then chooses the closest facility among the candidates. Figure 17.11 shows, for example, the closest fire station to a street address. A couple of options may be applied to the closest facility problem. First, rather than getting a single facility, the user may ask for a number of closest facilities. Second, the user may specify a search radius in distance or travel time, thus limiting the candidate facilities.

Closest facility analysis is regularly performed in location-based services (LBS), an application heavily promoted by GIS companies (Chapter 16). An LBS user can use a Web-enabled cell phone and a browser such as Google Maps to find the closest hospital, restaurant, or ATM. Closest facility is also important as a measure of quality of service such as health care (Schuurman, Leight, and Berube 2008).

17.5.5 Allocation

Allocation is the study of the spatial distribution of resources through a network. Resources in allocation studies often refer to public facilities, such as fire stations, schools, hospitals, or even open spaces (in case of earthquakes) (Tarabanis and Tsionas 1999). Because the distribution of the resources defines the extent of the service area, the main objective of spatial allocation analysis is to measure the efficiency of these resources.

A common measure of efficiency in the case of emergency services is the response time—the time it takes for a fire truck or an ambulance to reach an incident. Figure 17.12, for example, shows areas of a small city covered by two existing fire stations within a 2-minute response time. The map shows a large portion of the city is outside the 2-minute response zone. The response time to the city's outer zone is about 5 minutes (Figure 17.13). If residents of the city demand that the response time to any part of the city be 2 minutes or less, then the options are either to relocate the fire stations or, more likely, to build new fire stations. A new fire station should be located to cover the largest portion of the city unreachable in 2 minutes by the existing fire stations. The problem then

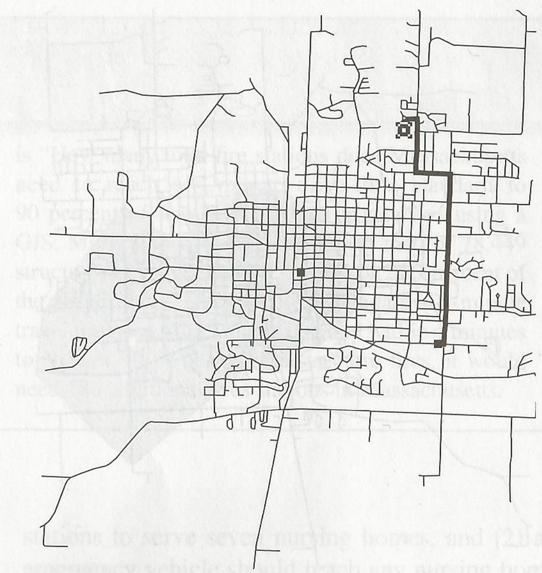


Figure 17.11
Shortest path from a street address to its closest fire station, shown by the square symbol.

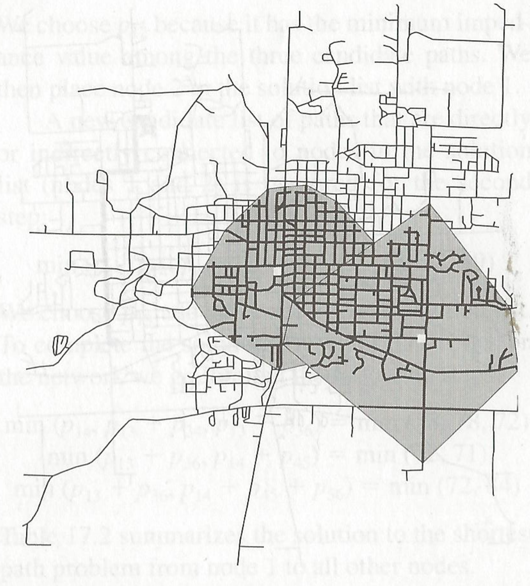


Figure 17.12
The service areas of two fire stations within a 2-minute response time.

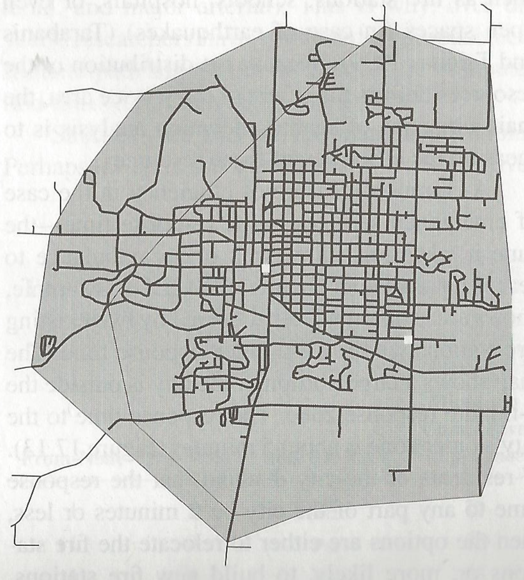


Figure 17.13
The service areas of two fire stations within a 5-minute response time.

becomes a location and allocation problem, which is covered in Section 17.5.6. Figures 17.12 and 17.13 illustrate hypothetical cases. Box 17.5, on the other hand, describes a study of response time to fires in Massachusetts.

For health care, the efficiency of allocation can be measured by the number or percentage of population served by hospitals. In their study of health care in non-urban areas of British Columbia, Canada, Schuurman et al. (2006) first defined a hospital's catchment (service area) as within one-hour travel time of the hospital on a road network. They then used spatial query (Chapter 10) to select census blocks within the catchment for analysis of variations in service coverage and population served.

17.5.6 Location–Allocation

Location–allocation solves problems of matching the supply and demand by using sets of objectives and constraints. The private sector offers many location–allocation examples. Suppose a company operates soft-drink distribution facilities to serve supermarkets. The objective in this example is to minimize the total distance traveled, and a constraint, such as a 2-hour drive distance, may be imposed on the problem. A location–allocation analysis matches the distribution facilities and the supermarkets while meeting both the objective and the constraint.

Location–allocation is also important in the public sector. For instance, a local school board may decide that all school-age children should be within 1 mile of their schools and the total distance traveled by all children should be minimized. In this case, schools represent the supply, and school-age children represent the demand. The objective of this location–allocation analysis is to provide equitable service to a population while maximizing efficiency in the total distance traveled.

The setup of a location–allocation problem requires inputs in supply, demand, and impedance measures. The supply consists of facilities at point locations. The demand may consist of individual points, or aggregate points representing line or polygon data. For example, the locations of school-age children may be represented

Box 17.5 Response Time to Fires

The response time standards of the National Fire Protection Association (NFPA) call for the arrival of an engine company at a fire scene within 4 minutes, 90 percent of the time, and for the deployment of an initial full alarm assignment within 8 minutes, 90 percent of the time. How to meet the NFPA standards in Massachusetts is addressed in Murray and Tong (2009), in which the authors answer questions from a *Boston Globe* reporter. The first and most relevant question is "How many total fire stations does Massachusetts need to reach a 4-minute drive-time standard to 90 percent of fires?" Following an analysis using a GIS, Murray and Tong (2009) report that, of 78,449 structure fires in their study, 19,385 or 24.7 percent of the total fires exceed the response standard of 4-minute travel time and, to be able to respond within 4 minutes to at least 90 percent of the structure fires, it would need 180 additional fires stations in Massachusetts.

as individual points or aggregate points (e.g., centroids) in unit areas such as census block groups. Impedance measures between the supply and demand can be expressed as travel distance or travel time. They can be measured along the shortest path between two points on a road network or along the straight line connecting two points. Shortest path distances are likely to yield more accurate results than straight-line distances.

Two most common models for solving location–allocation problems are minimum impedance (time or distance) and maximum coverage. The minimum impedance model, also called the *p*-median location model, minimizes the total distance or time traveled from all demand points to their nearest supply centers (Hakimi 1964). In contrast, the maximum coverage model maximizes the demand covered within a specified time or distance (Church and ReVelle 1974; Indriasari et al. 2010). Both models may take on added constraints or options. A maximum distance constraint may be imposed on the minimum impedance model so that the solution, while minimizing the total distance traveled, ensures that no demand is beyond the specified maximum distance. A desired distance option may be used with the maximum covering model to cover all demand points within the desired distance.

Here we will examine location–allocation problems in matching ambulance service and nursing homes. Suppose (1) there are two existing fire stations to serve seven nursing homes, and (2) an emergency vehicle should reach any nursing home in 4 minutes or less. Figure 17.14 shows that this objective cannot be achieved with the two existing

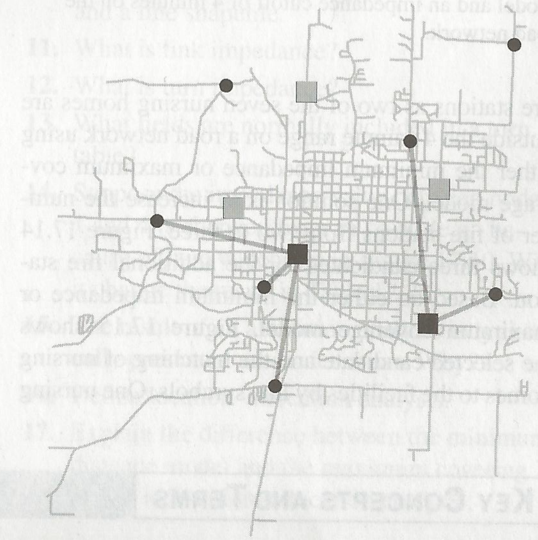


Figure 17.14
The two solid squares represent existing fire stations, the three gray squares candidate facilities, and the seven circles nursing homes. The map shows the result of matching two existing fire stations with nursing homes based on the minimum impedance model and an impedance cutoff of 4 minutes on the road network.

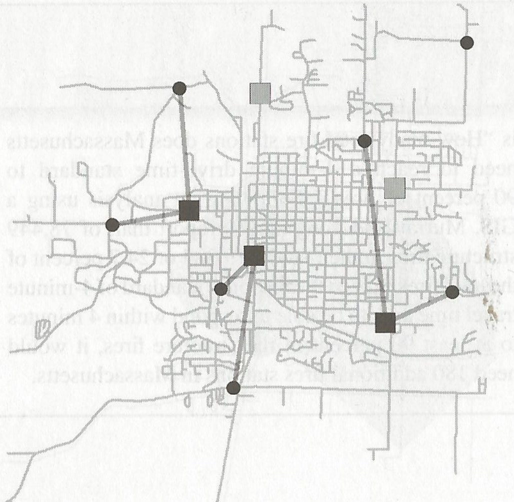


Figure 17.15
The map shows the result of matching three fire stations, two existing ones and one candidate, with six nursing homes based on the minimum impedance model and an impedance cutoff of 4 minutes on the road network.

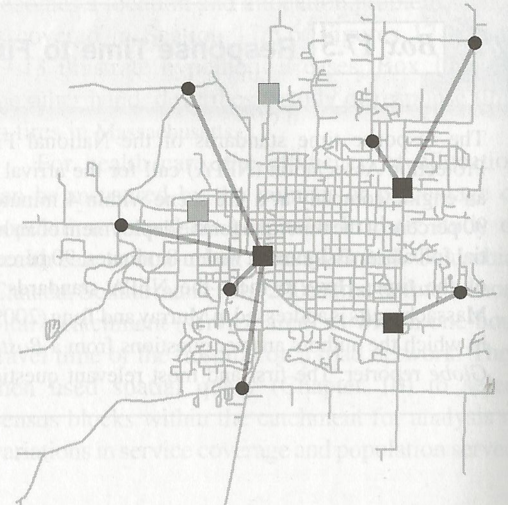


Figure 17.16
The map shows the result of matching three fire stations, two existing ones and one candidate, with seven nursing homes based on the minimum impedance model and an impedance cutoff of 5 minutes on the road network.

fire stations as two of the seven nursing homes are outside the 4-minute range on a road network using either the minimum impedance or maximum coverage model. One solution is to increase the number of fire stations from two to three. Figure 17.14 shows three candidates for the additional fire station. Based on either the minimum impedance or maximum coverage model, Figure 17.15 shows the selected candidate and the matching of nursing homes to the facilities by line symbols. One nursing home, however, is still outside the 4-minute range in Figure 17.15. There are two options to have a complete coverage of all nursing homes: one is to increase the facilities from three to four, and the other is to relax the constraint of the response time from 4 to 5 minutes. Figure 17.16 shows the result of the change of the response time from 4 to 5 minutes. All seven nursing homes can now be served by the three facilities. Notice that the selected candidate in Figure 17.16 differs from that in Figure 17.15.

KEY CONCEPTS AND TERMS

Allocation: A study of the spatial distribution of resources on a network.

Closest facility: A network analysis that finds the closest facility among candidate facilities.

Cost raster: A raster that defines the cost or impedance to move through each cell.

Junction: A street intersection.

Link: A segment separated by two nodes in a road network.

Link impedance: The cost of traversing a link, which may be measured by the physical length or the travel time.

Location–allocation: A spatial analysis approach that matches the supply and demand by using sets of objectives and constraints.

Network: A system of linear features that has the appropriate attributes for the flow of objects such as traffic flow.

Path distance: A term used in ArcGIS to describe the cost distance that is calculated from the surface distance, the vertical factor, and the horizontal factor.

Shortest path analysis: A network analysis approach that finds the path with the minimum cumulative impedance between nodes on a network.

Source raster: A raster that defines the source to which the least-cost path from each cell is calculated.

Traveling salesman problem: A network analysis scenario that finds the best route with the conditions of visiting each stop only once, and returning to the node where the journey starts.

Turn impedance: The cost of completing a turn on a road network, which is usually measured by the time delay.

REVIEW QUESTIONS

1. What is a source raster for least-cost path analysis?
2. Define a cost raster.
3. Box 17.1 summarizes the various costs for a pipeline project. Use Box 17.1 as a reference and list the types of costs that can be associated with a new road project.
4. Cost distance measure operations are based on the node-link cell representation. Explain the representation by using a diagram.
5. Refer to Figure 17.4*d*. Explain how the cell value of 5.5 (row 2, column 2) is derived. Is it the least accumulative cost possible?
6. Refer to Figure 17.4*d*. Show the least accumulative cost path for the cell at row 2, column 3.
7. Refer to Figure 17.5*b*. The cell at row 4, column 4 can be assigned to either one of the two source cells. Show the least-cost path from the cell to each source cell.
8. What is an allocation raster?
9. How does the surface distance differ from the regular (planimetric) cost distance?
10. Explain the difference between a network and a line shapefile.
11. What is link impedance?
12. What is turn impedance?
13. What fields are normally included in a turn table?
14. Suppose the impedance value between nodes 1 and 4 is changed from 58 to 40 (e.g., because of lane widening) in Figure 17.10. Will it change the result in Table 17.2?
15. The result of an allocation analysis is typically presented as service areas. Why?
16. Define location–allocation analysis.
17. Explain the difference between the minimum distance model and the maximum covering model in location–allocation analysis.

APPLICATIONS: PATH ANALYSIS AND NETWORK APPLICATIONS

This applications section has six tasks. Tasks 1 and 2 cover path analysis. You will work with the least accumulative cost distance in Task 1 and the path distance in Task 2. Cost distance and path distance are alternatives to Euclidean or straight-line distance covered in Chapter 12. Tasks 3 to 6 require the use of the Network Analyst extension. Task 3 runs a shortest path analysis. Task 4 lets you build a geodatabase network dataset. Task 5 runs a closest facility analysis. And Task 6 runs an allocation analysis.

Task 1 Compute the Least Accumulative Cost Distance

What you need: *sourcegrid* and *costgrid,* the same rasters as in Figure 17.4; and *pathgrid,* a raster to be used with the shortest path function. All three rasters are sample rasters and do not have the projection file.

In Task 1, you will use the same inputs as in Figures 17.4*a* and 17.4*b* to create the same outputs as in Figures 17.4*d,* 17.5*a,* and 17.5*b.*

1. Connect to the Chapter 17 database in ArcCatalog. Launch ArcMap. Rename the data frame Task 1, and add *sourcegrid, costgrid,* and *pathgrid* to Task 1. Ignore the warning message about the spatial reference.
2. Click ArcToolbox window to open ArcToolbox. Set the current workspace as the Chapter 17 database. Double-click the Cost Distance tool in the Spatial Analyst Tools/Distance toolset. In the next dialog, select *sourcegrid* for the input raster, select *costgrid* for the input cost raster, save the output distance raster as *CostDistance*, and save the output backlink raster as *CostDirection.* Click OK to run the command.
3. *CostDistance* shows the least accumulative cost distance from each cell to a source cell. You can use the Identify tool to click a cell and find its accumulative cost.

Q1. Are the cell values in *CostDistance* the same as those in Figure 17.4*d*?

4. *CostDirection* shows the least cost path from each cell to a source cell. The cell value in the raster indicates which neighboring cell to traverse to reach a source cell. The directions are coded 1 to 8, with 0 representing the cell itself (Figure 17.17).

6	7	8
5	0	1
4	3	2

Figure 17.17
Direction measures in a direction raster are numerically coded. The focal cell has a code of 0. The numeric codes 1 to 8 represent the direction measures of 90°, 135°, 180°, 225°, 315°, 360°, and 45° in a clockwise direction.

5. Double-click the Cost Allocation tool in the Spatial Analyst Tools/Distance toolset. Select *sourcegrid* for the input raster, select *costgrid* for the input cost raster, save the output allocation raster as *Allocation*, and click OK. *Allocation* shows the allocation of cells to each source cell. The output raster is the same as Figure 17.5*b.*
6. Double-click the Cost Path tool in the Spatial Analyst Tools/Distance toolset. Select *pathgrid* for the input raster, select *costgrid* for the input cost distance raster, select *CostDirection* for the input cost backlink raster, save the output raster as *ShortestPath*, and click OK. *ShortestPath* shows the path from each cell in *pathgrid* to its closest source. *ShortestPath* is the same as Figure 17.5*a.*

Task 2 Compute the Path Distance

What you need: *emidasub,* an elevation raster; *peakgrid,* a source raster with one source cell; and *emidapathgd,* a path raster that contains two cell values. All three rasters are projected onto UTM coordinates in meters.

In Task 2, you will find the least-cost path from each of the two cells in *emidapathgd* to the source cell in *peakgrid.* The least-cost path is based on the path distance. Calculated from an elevation raster, the path distance measures the ground or actual distance that must be covered between cells. The source cell is at a higher elevation than the two cells in *emidapathgd.* Therefore, you can imagine the objective of Task 2 is to find the least-cost hiking path from each of the two cells in *emidapathgd* to the source cell in *peakgrid.*

1. Insert a new data frame in ArcMap, and rename it Task 2. Add *emidasub, peakgrid,* and *emidapathgd* to Task 2. Select Properties from the context menu of *emidasub.* On the Symbology tab, right-click the Color Ramp box to uncheck Graphic View. Then select Elevation #1. As shown in the map, the source cell in *peakgrid* is located near the summit of the elevation surface and the two cells in *emidapathgd* are located in low elevation areas.
2. Double-click the Path Distance tool in the Spatial Analyst Tools/Distance toolset. Select *peakgrid* for the input raster, specify *pathdist1* for the output distance raster, select *emidasub* for the input surface raster, specify *backlink1* for the output backlink raster, and click OK to run the command.

Q2. What is the range of cell values in *pathdist1*?

Q3. If a cell value in *pathdist1* is 900, what does the value mean?

3. Double-click Cost Path in the Spatial Analyst Tools/Distance toolset. Select *emidapathgd* for the input raster, select *pathdist1* for the input cost distance raster, select *backlink1* for the input cost backlink raster, specify *path1* for the output raster, and click OK.
4. Open the attribute table of *path1.* Click the left cell of the first record. As shown in the map, the first record is the cell in *peakgrid.* Click the second record, which is the least-cost path from the first cell in *emidapathgd* (in the upper-right corner) to the cell in *peakgrid.*

Q4. What does the third record in the *path1* attribute table represent?

Task 3 Run Shortest Path Analysis

What you need: *uscities.shp,* a point shapefile containing cities in the conterminous United States; and *interstates.shp,* a line shapefile containing interstate highways in the conterminous United States. Both shapefiles are based on the Albers Conic Equal Area projection in meters.

The objective of Task 3 is to find the shortest route between two cities in *uscities.shp* on the interstate network. The shortest route is defined by the link impedance of travel time. The speed limit for calculating the travel time is 65 miles/hour. Helena, Montana, and Raleigh, North Carolina, are two cities for this task.

1. Right-click *interstates.shp* in the Catalog tree and select Item Descriptions. On the Preview tab of the Item Descriptions window, preview the table of *interstates.shp, which* has several attributes that are important for network analysis. The field MINUTES shows the travel time in minutes for each line segment. The field NAME lists the interstate number. And the field METERS shows the physical length in meters for each line segment.
2. Select Extensions from the Customize menu. Make sure that Network Analyst is checked. Select Toolbars from the Customize menu, and make sure that Network Analyst is checked.
3. This step uses *interstates.shp* in the Catalog tree to set up a network dataset. Right-click *interstates.shp* and select New Network Dataset. The New Network Dataset dialog allows you to set up various parameters for the network dataset. Accept the default name *interstates_ND* for the name of the network dataset. Opt not to model turns. Click on the Connectivity button in the next dialog. The Connectivity dialog shows interstates as the source, end point for connectivity, and 1 connectivity group. Click OK to exit the Connectivity dialog. In the New Network Dataset dialog, opt not to model the elevation of your network features. The next window shows Meters and Minutes as attributes for the network dataset. Click Next. Select yes to establish driving directions settings, and click the Directions button. The Network Directions Properties dialog shows that the display

length units will be in miles and the length attribute is in meters. NAME in *interstates .shp* will be the street (interstate in this case) name field. You can click the cell (Type) below Suffix T . . . and choose None. Click OK to exit the Network Directions Properties dialog, and click Next in the New Network Dataset dialog. A summary of the network dataset settings is displayed in the next window. Click Finish. Click Yes to build the network. Click No to add *interstates_ND.nd* to the map. Notice that *interstates_ND.nd,* a network dataset, and *interstates_ND_ Junctions.shp,* a junction feature class, have been added to the Catalog tree.

4. Insert a data frame in ArcMap and rename it Task 3. Add *interstates_ND.nd* to Task 3. Click yes to add all feature classes that participate in *interstates_ND* to map. Add *uscities.shp* to Task 3. Choose Select By Attributes from the Selection menu. In the next dialog, make sure that *uscities* is the layer for selection and enter the following expression to select Helena, MT, and Charlotte, NC: "City_Name" = 'Helena' Or "City-Name" = 'Charlotte'.

5. The Network Analyst toolbar should show *interstates_ND* in the Network Dataset box. Select New Route from the Network Analyst's dropdown menu. The Route analysis layer is added to the table of contents with its classes of Stops, Routes, and Barriers (Point, Line, and Polygon).

6. This step is to add Helena and Charlotte as stops for the shortest path analysis. Because the stops must be located on the network, you can use some help in locating them. Right-click *Route* in the table of contents and select Properties. On the Network Locations tab of the Layer Properties dialog, change the Search Tolerance to 1000 (meters). Click OK to exit the Layer Properties dialog. Zoom in on Helena, Montana. Click the Create Network Location tool on the Network Analyst toolbar and click a point on the interstate near Helena. The clicked point displays a symbol with 1. If the clicked point is not on the network, a question mark will show up next to the symbol. In that case, you can use the Select/Move Network Locations tool to move the point to be on the network. Repeat the same procedure to locate Charlotte on the network. The clicked point displays a symbol with 2. Click the Solve button on the Network Analyst toolbar to find the shortest path between the two stops.

7. The shortest route now appears in the map. Click the Directions Window on the Network Analyst toolbar. The Directions window shows the travel distance in miles, the travel time, and detailed driving directions of the shortest route from Helena to Charlotte.

Q5. What is the total travel distance in miles?

Q6. Approximately how many hours will it take to drive from Helena to Charlotte using the interstates?

Task 4 Build a Geodatabase Network Dataset

What you need: *moscowst.shp,* a line shapefile containing a street network in Moscow, Idaho; and *select_turns.dbf,* a dBASE file that lists selected turns in *moscowst.shp*.

moscowst.shp was compiled from the 2000 TIGER/Line files. *moscowst.shp* is projected onto a transverse Mercator coordinate system in meters. For Task 4, you will first examine the input data sets. Then build a personal geodatabase and a feature dataset. And then import *moscowst .shp* and *select_turns.dbf* as feature classes into the feature dataset. You will use the network dataset built in this task to run a closest facility analysis in Task 5.

1. Right-click *moscowst.shp* in the Catalog tree and select Item Description. *moscowst.shp* has the following attributes important for this task: MINUTES shows the travel time in minutes, ONEWAY identifies one-way streets as T, NAME shows the street name, and METERS shows the physical length in meters for each street segment.

Q7. How many one-way street segments (records) are in *moscowst.shp*?

2. Preview the table of *select_turns.dbf. select_ turns.dbf* is a turn table originally created in ArcInfo Workstation. The table has the following attributes important for this task: ANGLE lists the turn angle, ARC1_ID shows the first arc for the turn, ARC2_ID shows the second arc for the turn, and MINUTES lists the turn impedance in minutes.

3. Now create a personal geodatabase. Right-click the Chapter 17 database in the Catalog tree, point to New, and select Personal Geodatabase. Rename the geodatabase *Network .mdb.*

4. Create a feature dataset. Right-click *Network .mdb,* point to New, and select Feature Dataset. In the next dialog, enter *MoscowNet* for the name. Then click Projected Coordinate Systems and Import to import the coordinate system of *moscowst.shp* to be *MoscowNet*'s coordinate system. Select None for vertical coordinates. Take the default values for the tolerances. Then click Finish.

5. This step imports *moscowst.shp* to *MoscowNet.* Right-click *MoscowNet,* point to Import, and select Feature Class (single). In the next dialog, select *moscowst.shp* for the input features, check that the output location is *MoscowNet,* enter *MoscowSt* for the output feature class name, and click OK.

6. To add *select_turns.dbf* to *MoscowNet,* you need to use ArcToolbox. Double-click the Turn Table to Turn Feature Class tool in the Network Analyst Tools/Turn Feature Class toolset to open its dialog. Specify *select_turns.dbf* for the input turn table, specify *MoscowSt* in the *MoscowNet* feature dataset for the reference line features, enter *Select_Turns* for the output turn feature class name, and click OK. *select_turn.dbf* is a simple turn table, including only two-edge turns at street intersections with traffic lights. Network Analyst allows multi-edge turns. A multi-edge turn connects one edge to another through a sequence of connected intermediate edges. Network Analyst also allows the use of fields to describe the positions along the line features involved in turns, instead of turn angles.

7. Click the plus sign next to *MoscowNet* in the Catalog tree. *MoscowSt* and *Select_Turns* should be in the dataset.

8. With the input data ready, you can now build a network dataset. In the Catalog tree of ArcMap, right-click *MoscowNet,* point to New, and select Network Dataset. Do the following in the next four windows: take the default name (*MoscowNet_ND*) for the network dataset, select *MoscowSt* to participate in the network dataset, click yes to model turns and check the box next to *Select_ Turns,* take the default connectivity settings, check None to model the elevation of your network features, make sure that Minutes and Oneway are the default attributes for the network dataset, and opt to establish driving directions. After reviewing the summary, click Finish. Click Yes to build the network. Click No to add *MoscowNet_ND* to the map. Notice that *MoscowNet_ND,* a network dataset, and *MoscowNet_ND_ Junctions,* a junction feature class, have been added to the Catalog tree.

Task 5 Find Closest Facility

What you need: *MoscowNet,* a network dataset from Task 4; and *firestat.shp,* a point shapefile with two fire stations in Moscow, Idaho.

1. Insert a new data frame and rename it Task 5. Add the *MoscowNet* feature dataset and *firestat.shp* to Task 5. Turn off the

MoscowNet_ND_Junctions layer so that the map does not look too cluttered.

2. Make sure that the Network Analyst toolbar is available and *MoscowNet_ND* is the network dataset. Select New Closest Facility from the Network Analyst dropdown menu. The Closest Facility analysis layer is added to the table of contents with its analysis classes of Facilities, Incidents, Routes, and Barriers (Point, Line, and Polygon). Make sure that Closest Facility is checked to be visible.
3. Click Show/Hide Network Analyst Window on the Network Analyst toolbar to open the window. Right-click Facilities (0) in the Network Analyst window, and select Load Locations. In the next dialog, make sure that the locations will be loaded from *firestat,* before clicking OK.
4. Click the Closest Facility Properties button in the upper right of the Network Analyst window. On the Analysis Settings tab, opt to find 1 facility and to travel from Facility to Incident. Uncheck the box for Oneway in the Restrictions window. Click OK to dismiss the dialog. Click Incidents (0) to highlight it in the Network Analyst window. Then use the Create Network Location tool on the Network Analyst toolbar to click an incident point of your choice on the network. Click the Solve button. The map should show the route connecting the closest facility to the incident. Click the Directions Window button on the Network Analyst toolbar. The window lists the route's distance and travel time and details the driving directions.

Q8. Suppose an incident occurs at the intersection of Orchard and F. How long will the ambulance from the closest fire station take to reach the incident?

Task 6 Find Service Area

What you need: *MoscowNet* and *firestat.shp,* same as Task 5.

1. Insert a new data frame and rename it Task 6. Add the *MoscowNet* feature dataset and *firestat.shp* to Task 6. Turn off the *MoscowNet_ND_Junctions* layer. Select New Service Area from the Network Analyst's dropdown menu. Click Show/Hide Network Analyst Window to open the window. The Network Analyst window opens with four empty lists of Facilities, Polygons, Lines, and Barriers (Point, Line, and Polygon).
2. Next add the fire stations as facilities. Right-click Facilities (0) in the Network Analyst window and select Load Locations. In the next dialog, make sure that the facilities are loaded from *firestat* and click OK. Location 1 and Location 2 should now be added as facilities in the Network Analyst window.
3. This step sets up the parameters for the service area analysis. Click the Service Area Properties button in the upper right of the Network Analyst window to open the dialog box. On the Analysis Settings tab, select Minutes for the impedance, enter "2 5" for default breaks of 2 and 5 minutes, check direction to be away from Facility, and uncheck Oneway restrictions. On the Polygon Generation tab, check the box to generate polygons, opt for generalized polygon type and trim polygons, select not overlapping for multiple facilities options, and choose rings for the overlay type. Click OK to dismiss the Layer Properties dialog.
4. Click the Solve button on the Network Analyst toolbar to calculate the fire station service areas. The service area polygons now appear in the map as well as in the Network Analyst window under Polygon (4). Expand Polygons (4). Each fire station is associated with two service areas, one for 0–2 minutes and the other for 2–5 minutes. To see the boundary of a service area (e.g., 2 to 5 minutes from Location 1), you can simply click the service area under Polygon (4).
5. This step shows how to save a service area as a feature class. First select the service area (polygon) in the Network Analyst window. Then right-click the Polygon (4) layer in the window, and select Export Data. Save the data as a feature class in *MoscowNet.* The feature class attribute table contains the default fields of area and length.

Q9. What is the size of the 2-minute service area of Location 1 (fire station 1)?

Q10. What is the size of the 2-minute service area of Location 2 (fire station 2)?

Challenge Task

What you need: *uscities.shp* and *interstates.shp,* same as Task 3.

This challenge task asks you to find the shortest route by travel time from Grand Forks, North Dakota, to Houston, Texas.

Q1. What is the total travel distance in miles?

Q2. Approximately how many hours will it take to drive from Grand Forks to Houston using the interstates?

REFERENCES

Apparicio, P., M. Cloutier, and R. Shearmur. 2007. The Case of Montréal's Missing Food Deserts: Evaluation of Accessibility to Food Supermarkets. *International Journal of Health Geographics* 6:4 doi: 10.1186/1476-072X-6-4.

Atkinson, D. M., P. Deadman, D. Dudycha, and S. Traynor. 2005. Multi-Criteria Evaluation and Least Cost Path Analysis an Arctic All-Weather Road. *Applied Geography* 25: 287–307.

Bräysy, O., W. Dullaert, and P. Nakari. 2009. The Potential of Optimization in Communal Routing Problems: Case Studies from Finland. *Journal of Transport Geography* 17: 484–90.

Choi, Y., H. Park, C. Sunwoo, and K. C. Clarke. 2009. Multi-Criteria Evaluation and Least Cost Path Analysis for Optimal Haulage Routing of Dump Trucks in Large Scale Open-Pit Mines. *International Journal of Geographical Information Science* 23: 1541–67.

Church, R. L., and C. S. ReVelle. 1974. The Maximal Covering Location Problem. *Papers of the Regional Science Association* 32: 101–18.

Coffee, N., D. Turner, R. A. Clark, K. Eckert, D. Coombe, G. Hugo, D. van Gaans, D. Wilkinson, S. Stewart, and A. A. Tonkin. 2012. Measuring National Accessibility to Cardiac Services using Geographic Information Systems. *Applied Geography* 34: 445–55.

Collischonn, W., and J. V. Pilar. 2000. A Direction Dependent Least Costs Path Algorithm for Roads and Canals. *International Journal of Geographical Information Science* 14: 397–406.

Dijkstra, E. W. 1959. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik* 1: 269–71.

Falcucci, A., L. Maiorano, P. Ciucci, E. O. Garton, and L. Boitani. 2008. Land-Cover Change and the Future of the Apennine Brown Bear: A Perspective from the Past. *Journal of Mammalogy* 89: 1502–1511.

Farhan, B., and A. T. Murray. 2005. A GIS-Based Approach for Delineating Market Areas for Park and Ride Facilities. *Transactions in GIS* 9: 91–108.

Hakimi, S. L. 1964. Optimum Locations of Switching Centers and the Absolute Centers and Medians of a Graph. *Operations Research* 12: 450–59.

Hallett IV, L. F., and D. McDermott. 2011. Quantifying the Extent and Cost of Food Deserts in Lawrence, Kansas, USA. *Applied Geography* 31: 1210–1215.

Hepner, G. H., and M. V. Finco. 1995. Modeling Dense Gas Contaminant Pathways over Complex Terrain Using a Geographic Information System. *Journal of Hazardous Materials* 42: 187–99.

Indriasari, V., A. R. Mahmud, N. Ahmad, and A. R. M. Shariff. 2010. Maximal Service Area Problem for Optimal Siting of Emergency Facilities. *International Journal of Geographical Information Science* 24: 213–30.

Keenan, P. 2008. Modelling Vehicle Routing in GIS. 2008. *Operational Research* 8: 201–18.