to appear in: *Handbook 5 Transport Geography and Spatial Systems*. Hensher D., Button K., Haynes K. and Stopher P. *(*eds.); Pergamon, 2003

GIS AND NETWORK ANALYSIS

Manfred M. Fischer
Department of Economic Geography & Geoinformatics
Vienna University of Economics and Business Administration
Rossauer Lände 23/1
A-1090 Vienna, Austria

1 INTRODUCTION

Both geographic information systems (GIS) and network analysis are burgeoning fields, characterised by rapid methodological and scientific advances in recent years. A geographic information system (GIS) is a digital computer application designed for the capture, storage, manipulation, analysis and display of geographic information. Geographic location is the element that distinguishes geographic information from all other types of information. Without location, data are termed to be non-spatial and would have little value within a GIS. Location is, thus, the basis for many benefits of GIS: the ability to map, the ability to measure distances and the ability to tie different kinds of information together because they refer to the same place (Longley *et al.*, 2001).

GIS-T, the application of geographic information science and systems to transportation problems, represents one of the most important application areas of GIS-technology today. While traditional GIS formulation's strengths are in mapping display and geodata processing, GIS-T requires new data structures to represent the complexities of transportation networks and to perform different network algorithms in order to fulfil its potential in the field of logistics and distribution logistics.

This paper addresses these issues as follows. The section that follows discusses data models and design issues which are specifically oriented to GIS-T, and identifies several improvements of the traditional network data model that are needed to support advanced network analysis in a ground transportation context. These improvements include turn-tables, dynamic segmentation, linear referencing, traffic lines and non-planar networks. Most commercial GIS software vendors have extended their basic GIS data model during the past two decades to incorporate these innovations (Goodchild, 1998).

The third section shifts attention to network routing problems that have become prominent in GIS-T: the travelling salesman problem, the vehicle routing problem and the shortest path problem with time windows, a problem that occurs as a subproblem in many time constrained routing and scheduling issues of practical importance. Such problems are conceptually simple, but mathematically complex and challenging. The focus is on theory and algorithms for solving these problems. The chapter concludes with some final remarks.

2 NETWORK REPRESENTATION AND GIS-T NETWORK DATA MODELS

2.1 Terminology

A network is referred to as a *pure network* if only its topology and connectivity are considered. If a network is characterised by its topology and flow characteristics (such as capacity constraints, path choice and link cost functions) it is referred to as a *flow network*. A *transportation network* is a flow network representing the movement of people, vehicles or goods (Bell and Iida, 1997).

The approach adopted almost universally is to represent a transportation network by a set of *nodes* and a set of *links*. The nodes represent points in space and possibly also in time, and the links tend to correspond to identifiable pieces of transport infrastructure (like a section of road or railway). Links may be either *directed*, in which case they specify the direction of movement, or *undirected*.

In graph theoretical terminology, a transportation network can be referred to as a *valued graph*, or alternatively a *network*. Directed links are referred to as *arcs* while undirected links as *edges*. Other useful terms with some intuitive interpretations are a *path* which is a sequence of distinct nodes connected in one direction by links; a *cycle* which is a path connected to itself at the ends; and a *tree* which is a network where every node is visited once and only once. The relationship between the nodes and the arcs, referred to as the *network topology*, can be specified by a *node-arc incidence matrix*: a table of binary or ternary variables stating the presence or absence of a relationship between network elements. The node-arc incidence matrix specifies the network topology and is useful for network processing.

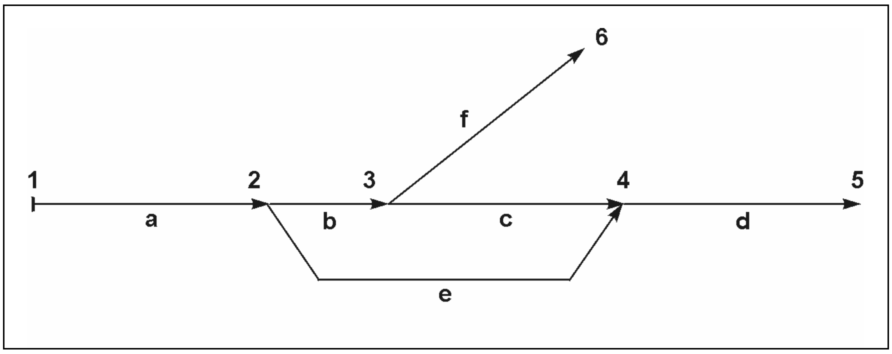
2.2 The network data model

The heart of any GIS is its data model. A data model is an abstract representation of some real-world situation used to organise data in a database. Data models typically consist of three major components. The first is a set of data objects or *entity types* that form the basic building blocks for the database. The second component is a set of general *integrity rules* which constrain the occurrences of entities to those which can legally appear in the database. The final component includes *operators* that can be applied to entities in the database (Miller and Shaw, 2001).

Data modelling involves three different levels of abstraction: conceptual, logical and physical levels. Conceptual data models describe the organisation of data at a high level of abstraction, without taking implementation aspects into account. The *entity-relationship* and the *extended entity-relationship* models are the most widely used conceptual data models. They provide a series of concepts such as entities, relationships, attributes, capable of describing the data requirements of an application in a manner that is easy to understand and independent of the criteria for managing and organising data on the system. A logical data model translates the conceptual model into a system-specific data scheme, while low-level physical data models provide the details of physical implementation (file organisation and indexes) on a given logical data model (Atzeni *et al.*, 1999).

The network data model is the most popular conceptual model to represent a network within a GIS environment. The model – a special type of the node-arc-area data model that underlies many basic vector GIS databases – is built around two core entities: the *Node* (a zero-dimensional entity) and the *Arc* (a one-dimensional entity). Current GIS data models typically represent a network as a collection of arcs with nodes created at the arc intersections. The planar embedding of the node-arc data model guarantees topological consistency of the network.

The most widely used logical data model that supports the node-arc representation of networks is the georelational model. This model separates spatial and attribute data into different data models. A logical spatial data model (the vector data model) that encodes nodes and arcs maintains the geometry and associated topological information, while the associated attribute information is held in relational database management (RDBMS) tables. Unique identifiers associated with each spatial entity (node, arc) provide links to records in the relational model and its data on the entity's attributes. This hybrid data management strategy was developed to take advantage of a relational database management system to store and manipulate attribute information (Longley *et al.*, 2001). But this solution does not allow the relationships between a spatial object and its attributes have their own attributes (Goodchild, 1998). Though the solution is neither elegant nor robust, it is effective and the georelational model is widely present in GIS software (Miller and Shaw, 2001).



(a) Example network for the relational model example

Arc ID	Street Name	Lanes	Other Attributes
a	High Street	2	
b	High Street	4	
c	High Street	4	
d	High Street	2	
e	River Way	2	
f	Hill Street	2	

(b) A simple arc table

Arc ID	Stop Light	Other Attributes
1	n	
2	y	
3	n	
4	y	
5	n	
6	n	

(c) A simple node table

Arc ID	Street Name	Lanes	From Node	To Node
a	High Street	2	1	2
b	High Street	4	2	3
c	High Street	4	3	4
d	High Street	2	4	5
e	River Way	2	2	4
f	Hill Street	2	3	6

Node ID	Stop Light?	Arc Links
1	n	a
2	y	a, b
3	n	b, c, f
4	y	c, d, e
5	n	d
6	n	f

(d) Pointers added to the arc and node tables to represent connectivity

Figure 1 Relational data model representations of the arcs and nodes of a network (adapted from Goodchild 1998)

The relational structure to support the planar network model typically consists of an arc relation and a node relation. The structure may be illustrated as a representation of the simple network shown graphically in Figure 1(a). The model implemented in GIS represents each arc of the network as a polyline entity. Associated with each entity will be a set of attributes, conceived as the entries in one row of a rectangular table (see Figure 1(b)). Properties may include information about the transverse structure such as the number of lanes or information on address locations within the network. Commonly included attributes are arc length, free flow travel time, base flow and estimated flow. The base and estimated flows usually refer to the observed flow and the flow estimated from some modelling exercise (Miller and Shaw, 2001).

The node relation typically contains a node ID field and relevant attributes of the node, such as, for example, the presence of a light. Figure 1(d) shows a scheme that includes storage of pointers from arcs to nodes, and from nodes to arcs, to store the network's topology (connectivity). Each arc has an inherent direction defined by the order of points in its polyline representation as illustrated by arrows in Figure 1(a) (Goodchild, 1998). It is noteworthy that the node-arc representation disaggregates a transportation system into separate subnetworks for each mode within a single base network. Transfer-arcs link the subnetworks. The pseudo-arcs represent modal transfers.

2.3 Non-planar networks and the turn-table

The planar network data model has received widespread acceptance and use. Despite its popularity, the model has limitations for some areas of transportation analysis, especially where complex network structures are involved. One major problem is caused by the planar embedding requirement. This requirement forces nodes to exist at all arc intersections and, thus, ensures topological consistency of the model. But intersections at grade cannot be distinguished from intersections with an overpass or underpass that do not cross at grade. This difficulty in representing underpasses or overpasses may lead to problems when running routing algorithms (Kwan *et al.*, 1996). The drawbacks in planar topology for network representations have motivated interest in non-planar network models. Planar models force nodes at all intersections (see Figure 2(a)), while non-planar network models (see Figure 2(b)) do not. Non-planar networks are broadly defined as those networks which permit arcs of the network to cross without a network node being located at the intersection. There is no implicit or explicit contact between the line segments at the point of intersections (Fohl *et al.*, 1996).

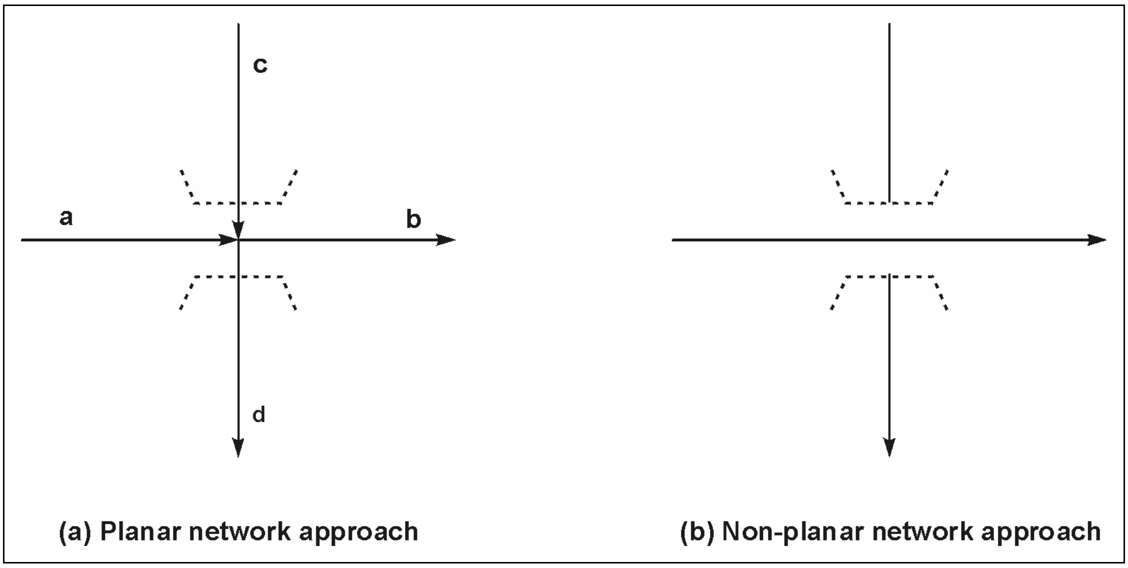


Figure 2. Illustration of (a) planar and (b) non-planar network approaches to represent an overpass

But non-planar data models provide only a partial solution to the problem of connectivity. In transportation network analysis it may be necessary to include extensive information on the ability to connect from one arc to another. Drivers, for example, may force turn restrictions or trucks may be limited by turning radius. Such situations require more than the simple ability to represent the existence of a crossing at grade or an underpass (Goodchild, 1998).

To resolve this problem, the standard fully intersected planar network data model has been extended by adding a new structure, called the *turn-table*. Table 1 shows a turn-table for the layout used in Figure 2. For each ordered pair of arcs incident at a node, a row of attributes in the table gives appropriate characteristics of the turn (yes/no), together with links to the tables that contain the attributes of the arcs. In this way, a data model with a planar embedding requirement can represent overpasses and underpasses by preventing turns (Goodchild, 1998).

From Arc	To Arc	Turn ?
a	c	n
a	b	y
a	d	n
b	a	y
b	c	n
b	d	n
c	a	n
c	b	n
c	d	y
d	a	n
d	b	n
d	c	y

Table 1 Layout of a turn-table for the layout used in Figure 2(a)

2.4 Linear referencing systems and dynamic segmentation

While geographic features are typically located using planar referencing systems, many characteristics associated with a transportation network are located by means of a linear rather than coordinate-based system. These characteristics include data on transportation-related events and facilities (often termed feature data). In order to use linear-referenced attributes in conjunction with a spatially referenced transportation network, there must be some means of linking the two referencing systems together (Spear and Lakshmanan, 1998).

Linear referencing systems typically consist of three components (Sutton, 1997; Vanderohe and Hepworth, 1996): a *transportation network*, a *linear referencing method* and a *datum*. The transportation network is represented by the conventional node-arc network. The linear referencing method determines an unknown location within the network using a defined path and an offset distance along that path from some known location (Miller and Shaw, 2001). The datum is the set of objects (so-called reference or anchor points) with known georeferenced locations that can be used to anchor the distance calculations for the linear referenced objects.

There are different linear referencing methods. Nyerges (1990) identifies three major strategies, namely, *road name and kilometer(mile)point referencing*, *control section locational referencing*, and *link and node locational referencing*. Road name and kilometerpoint is a system familiar to anyone who has driven on highways in Europe or the USA. This system consists of a road naming convention (that is, a standard procedure for assigning names to highways and streets) and a series of kilometerpoint references (that is, distance calculations along the network, typically measured in fractions of a kilometer or mile). Kilometer point referencing requires a designated point of reference (for example a kilometer 0) as a datum (see Figure 3). This is often an end point of the route or where the route crosses a provincial or a national boundary (Miller and Shaw, 2001).

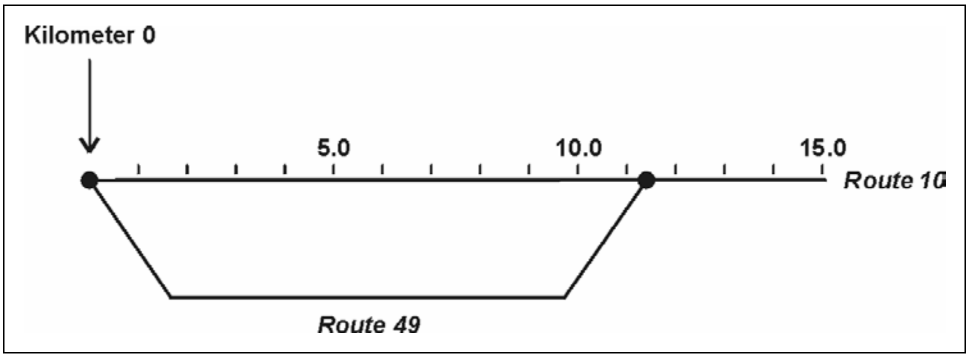
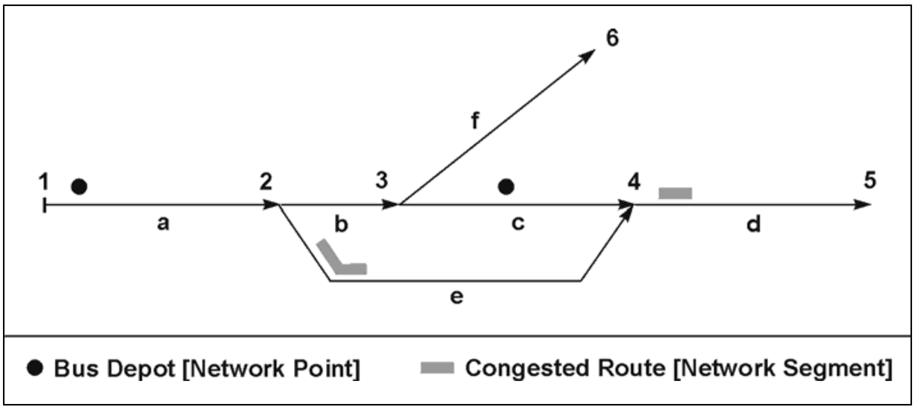


Figure 3 Kilometerpoint referencing

Due to road modifications and other changes in road geometry, kilometerpoint referencing can become increasingly inaccurate over time. In other words, the reference kilometerpoint may not reflect the actual distance from the point of origin. This may cause problems when maintaining historical records of transportation events. This requires some type of translation factor to adjust distances (Nyerges, 1990; Miller and Shaw, 2001).


6
f
1
2
3
4
5
a
b
c
d
e
● Bus Depot [Network Point]
Congested Route [Network Segment]

(a)

Arc	Distance from Arc Start	Feature
a	0.7	Bus Depot
c	2.2	Bus Depot

(b)

Arc	Distance from Arc Start to Start of Feature	Distance from Arc Start to End of Feature	Level of Congestion
d	0.5	1.2	High
e	1.0	2.5	High

Figure 4 The concept of dynamic segmentation: A simple example for (a) network points and (b) network segments (adapted from Goodchild, 1998)

The key to tie (zero-dimensional and one-dimensional) objects located at arbitrary locations on the network to the node-arc structure of the network data model is *dynamic segmentation*. The term derives from the fact that feature data values are held separately from the actual network route in database tables and then dynamically added to segments of the route each time the user queries the database (Longley *et al.* 2001). Several commercial GIS software packages provide dynamic segmentation capabilities, typically maintained at the logical level using the relational data model (Miller and Shaw, 2001). Figure 4 shows a simple illustration of the concept, for two types of objects located at arbitrary locations on the network. These entities – termed network points (point events) and network segments (line events) – are given their own attribute tables. Dynamic segmentation reduces the number of transportation features or network links that have to be maintained to represent the system and is particularly useful in situations in which the event data change frequently and need to be stored in a database due to access from other applications (Longley *et al.*, 2001).

2.5 Lanes and navigable data models

A straightforward way to enhance the basic mode-arc model for ITS (intelligent transportation systems) applications is to add information on the transverse structure of the network (Miller and Shaw, 2001). Even though certain information about the transverse structure (such as the existence of a median or the number of lanes) might be stored as attributes of arcs or network lines, it is not possible to store detailed information about individual lanes or connectivity at the lane level. There is, for example, no way to disaggregate a turn-table to store turn restrictions which are specific to lanes (Goodchild, 1998).

ITS database requirements go well beyond the traditional requirements of maintaining arc-node topology, two-dimensional georeferencing and linear referencing of events within transportation networks. A fully fletched ITS requires a high-integrity, real-time information system which will receive inputs from sensors embedded within transportation facilities and from vehicles equipped with GPS (Global Positioning System) devices and navigable data models. Navigable data models are digital geographic databases of a transportation system that can support vehicle guidance operations of different kinds. For intelligent transportation systems, this includes four functions (Dane and Rizos, 1998; Miller and Shaw, 2001). *First*, the data model has to *unambiguously translate coordinate-based locations into street addresses* and vice versa. Travellers utilise address systems for location referencing while ITS tracks a vehicle utilising a GPS receiver that can provide locations at accuracies of 5-10 m.

Second, the data model has to support *map matching*. This refers to the ability to snap a vehicle's position to the nearest location on a network segment when its estimated or measured location is outside the network. This may occur due to differences in accuracy between the digital network database and the global positioning system. *Third*, the data model has to have the capability to represent the transportation network in detail sufficient to perform different network algorithms, modeling and simulations. In the real world, a transportation network has different types of intersections that are of interest to ITS builders. For some applications, information on intersections, lanes and lane changes, highway entrances and exits etc. is important. Other applications may require geometric representation of road curvature and incline. *Fourth*, the data model must not only assist the traveler in selecting an optimal route based on stated criteria such as travel time, cost and navigational simplicity, but also support *route guidance*. This refers to navigational instructions and is a challenging task in real time.

Although dynamic segmentation can be used to enhance the traditional node-arc structure for ITS applications much of the high-resolution positional information provided by in-vehicle GPS receivers is lost when referenced within the traditional network structure. While 50 m accuracy may be sufficient to locate a vehicle on a road, better than 5 m will be required to locate to the lane level. Such accuracies are well beyond the capability of many of the currently available network databases. Achievement of better than 5 m accuracy with GPS requires the use of differential techniques and a high quality of geodetic control (Goodchild, 1998).

Fohl *et al.* (1996) describe a prototype lane-based navigable data model where each lane is represented as a distinct entity, with its own connectivity with other lanes, but its geometry is obtained from the standard linear geometry of the road. No attempt is made to store the relative positions of lanes, but the structure does identify such topological properties such as adjacency, and the order of lanes across the road (Goodchild, 1998). A more radical approach to navigable data models for ITS is to abandon the node-arc model entirely. Bespalko *et al.* (1996) suggest a 3-D object-oriented GIS-T data model that can distinguish between overpasses, underpasses and intersections and thereby providing guidance through complex intersections.

3 VEHICLE ROUTING WITHIN A NETWORK: PROBLEMS AND ALGORITHMS

At the core of many procedures in GIS-T software are algorithms for solving network routing problems. The problems are conceptually simple, but mathematically complex and challenging. How can we best route vehicles (trucks school buses and general passenger buses) from one location to another? The problems encountered in answering such questions have an underlying combinatorial structure. For example, either we dispatch a vehicle or we do not, or we use one particular route or another.

This section deals with node routing problems. Node routing – in contrast to arc routing – refers to routing problems where the key service activity occurs at the nodes (customers) and arcs are of interest only as elements of paths that connect the nodes (Assad and Golden, 1995). We discuss two specific problems that have become prominent in network analysis: the travelling salesman problem and the vehicle routing problem. The survey of basic network algorithms is completed by discussing a dynamic programming approach for solving the shortest path problem with time windows, a problem that occurs as a subproblem in many time constrained routing and scheduling issues.

At the outset of this section we should note that vehicle routing algorithms can be applied in one of two modes: first, variable routing, and second, fixed routing. In a variable routing context, an algorithm is utilised with actual customer delivery requirements to develop routes for the next planning horizon. Fixed routing is applied when customer demands are sufficiently stable to allow use of the same routes repeatedly (Fisher, 1995).

3.1 The traveling salesman problem

The simplest node routing problem is the travelling salesman problem (TSP). The TSP is a classical combinatorial optimisation problem that is simple to state but very difficult to solve. The problem is to find the least cost tour through a set of nodes so that each node is visited exactly once. The tour starts and ends from a specific location, called depot. The problem is similar to many other problems in that it is easy to describe but difficult to actually find a solution in real world contexts. It is in a precise mathematical sense difficult, namely NP-complete (non-deterministic polynomial time complete), and cannot be solved exactly in polynomial time. Although there are many ways to formally state the TSP a convenient way in doing so is an integer linear programming formulation. Assume a directly connected network G=(**N**, **A**) where **N** is the node set with |**N**|=N and **A** the arc set defined as the Cartesian product of **N** with itself (that is, **A**=**N** x **N**) then the TSP can be formulated as

$$\min_{\{x_{ij}\}} \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij}\, x_{ij} \tag{1}$$

subject to

$$\sum_{j=1}^{N} x_{ij} = 1 \qquad \text{for i=1, …, N} \tag{2}$$

$$\sum_{i=1}^{N} x_{ij} = 1 \qquad \text{for j=1, …, N} \tag{3}$$

$$\left(x_{ij} \right) \in \mathbf{X} \tag{4}$$

$$x_{ij} \in \{0,1\} \qquad \text{for i, j=1, …, N} \tag{5}$$

where c_{ij} is the arc's length from node i to node j, and the x_{ij}'s are the decision variables: x_{ij} is set to one when arc (i, j) is included in the tour, and zero otherwise. $(x_{ij}) \in \mathbf{X}$ denotes the set of subtour-breaking constraints that restrict the feasible solutions to those consisting of a single tour. The subtour-breaking constraints can be formulated in different ways. But one very intuitive formulation is:

$$\sum_{i,j \in \mathbf{S_A}} x_{ij} \le \left|\mathbf{S_A}\right| - 1,\ \mathbf{S_A} \subseteq \mathbf{A};\ 2 \le \left|\mathbf{S_A}\right| \le N - 2 \tag{6}$$

where $\mathbf{S_A}$ is some subset of **A** and | $\mathbf{S_A}$ | the cardinality of $\mathbf{S_A}$. These constraints prohibit subtours, i.e. tours on subsets with less than N nodes. If there were such a subtour on some subset of $\mathbf{S_A}$ of nodes, this subtour would conatin | $\mathbf{S_A}$ | arcs. Consequently, the left hand side of the inequality would be equal to | $\mathbf{S_A}$ | which is greater than | $\mathbf{S_A}$ |-1, and the constraint would be violated for this particular subset. Without (6), the TSP reduces to an assignment problem (Potvin, 1993).

The TSP is a classically hard problem to come up with the optimal solution. Enumerating the possibilities works well for small N. But there are (N-1)! candidate itineries from which the single optimal one must be found, where the symbol ! indicates the product of the integers from one up to and including the number, known as the number's factorial. If N=100, then the number of possible tours is 10^{200}. Many heuristic algorithms have been devised to solve the problem (for an overview see Laporte 1992 and Lawler *et al.*, 1985). These heuristics were designed to work quickly and to come close to the optimal solution. But they do not guarantee that the optimum will be found. Two broad classes of TSP heuristics can be distinguished: classical heuristic algorithms and optimisation based algorithms.

Classical TSP heuristics include tour construction procedures, tour improvement procedures and composite procedures that are based on both types of techniques. The best known *tour construction heuristics* gradually build a tour by selecting each node in turn and inserting them one by one into the current tour. Various metrics may be utilised for the choice of the next node, like the proximity to the current tour. Among the *tour improvement procedures*, the r-opt exchange heuristics are the most widely used especially the 2-opt, 3-opt (see Lin, 1965) and the interchange heuristic of Lin and Kernighan (1973). These TSP heuristics locally modify the current solution by replacing r arcs in the tour by r new arcs so as to generate a new improved tour. Characteristically, the exchange heuristics are applied iteratively until a local optimum is found, i.e. a tour that cannot be improved further via the exchange heuristic under consideration. To overcome the limitations associated with local optimality, new heuristics like tabu search, simulated annealing and computational intelligence-based techniques may be utilised to escape from local minima (see Glover, 1989; 1990, Kirkpatrick *et al.*, 1983; Potvin, 1993). *Composite procedures* make use of both tour construction and improvement techniques. They belong to the most powerful heuristics to solve the TSP. The iterated Lin-Kernighan heuristic, for example, can routinely find solutions within one percent of the optimum for travelling salesman problems with up to 10,000 nodes (Johnson, 1990).

Optimisation-based heuristics are very different in character from the classical TSP heuristics. They apply some optimisation algorithm and simply terminate prior to optimality. The most popular technique is branch-and-bound, originally applied to the TSP by Dantzig at al. (1954) and continually refined over the years. Branch-and-bound is a directed enumeration procedure that partitions the solution space into increasingly smaller subsets in an attempt to identify the subset that contains a (near) optimal solution. For each subset a bound is calculated that estimates the best possible solution in the subset. The assignment problem relaxation, for example, may be used to generate lower bounds on the optimum. If the bound for a subset indicates that it cannot contain a (near) optimal solution, the partitioning process is continued with another subset. The algorithm terminates when there are no subsets remaining. It is worth noting that travelling salesman problems with a few hundred nodes can be routinely solved to optimality.

3.2 The vehicle routing problem

There are many ways of generalising the TSP to match real world situations. Often there is more than one vehicle, and in these situations the division of stops between variables is an important decision variable (Longley *et al.*, 2001). In this subsection we consider the vehicle routing problem (VRP). The problem is to route a fixed number of vehicles through a number of demand locations such that the total cost of travel is minimised and vehicle capacity constraints are not violated. Characteristically, there is a designated location known as the depot where all vehicles have to start and end their tours. There are numerous variations of this basic VRP, including time windows for delivery, stochastic demand, multiple depots with each vehicle in the fleet assigned to a particular depot etc. The problem and its extensions are of substantial practical importance and have resulted in the development of a great many heuristic algorithms including computational intelligence procedures over the past 35 years (see Bodin *et al.*, 1983; Golden and Assad, 1986; and Fischer, 1995 for a review).

We view the vehicle routing problem as consisting of two interlinked problems: first, finding an optimal assignment of customer orders to vehicles, and second, what route each vehicle will follow in servicing its assigned demand in order to minimise total delivery cost. To provide a precise statement of the problem we introduce notation first and then draw on Fisher and Jaikumar (1981) to specify the VRP as a non-linear generalised assignment problem.

Let $\mathbf{N}=\{1, \ldots, N\}$ be the set of demand locations (customers) and $\mathbf{K}=\{1, \ldots, K\}$ be the set of available vehicles to be routed and scheduled. Consider the network $G=(\mathbf{V}, \mathbf{A})$ where $\mathbf{V}=\mathbf{N}\cup\{0\}$ is the set of nodes with 0 representing the depot of the vehicles, and $\mathbf{A}=\mathbf{V}\mathrm{x}\mathbf{V}$ is the arc set that contains all arcs (i, j) with i, $j\in\mathbf{V}$. c_{ij} denotes the cost of direct travel from point i to point j, b_k the capacity (for example, weight or volume) of vehicle k, and a_i the size of the order of customer $i\in\mathbf{N}$, measured in the same units as the vehicle capacity. Define the flow variable y_{ik} as 0-1 variable equal to one if the order from customer i is delivered by vehicle k, and zero otherwise, and $y_k:=(y_{0k}, \ldots, y_{nk})$, then the vehicle routing problem can be formulated as the following non-linear generalised assignment problem (Fisher, 1995):

$$\min_{\{yk\}} \sum_{k=1}^{K} f\left(y_k\right) \tag{7}$$

subject to

$$\sum_{i=1}^{N} a_i\, y_{ik} \le b_k \qquad \text{for k=1, …, K} \tag{8}$$

$$\sum_{k=1}^{K} y_{ik} = \begin{cases} K & i=0 \\ 1 & i=1, ..., N \end{cases} \tag{9}$$

$$y_{ik} \in \{0,1\} \qquad \text{for i=0, 1, …, N, k=1, …, K} \tag{10}$$

Equations (8)-(19) are the constraints of the assignment problem and guarantee that each route begins and ends at the depot (i=0), that each customer (i=1, …,n) is serviced by some vehicle, and that the load assigned to the vehicle is within its capacity. $f(y_k)$ represents the cost of an optimal TSP tour of all the points in $\mathbf{V}(y_k)=\{i|y_{ik}=1\}$ that must be made for each vehicle k to service its assigned customers. Defining $x_{ijk}=1$ if vehicle k travels directly from i to j, and $x_{ijk}=0$ otherwise, the function $f(y_k)$ can be defined mathematically as

$$f\left(y_k\right) = \min \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij}\, x_{ijk} \tag{11}$$

such that

$$\sum_{i=1}^{N} x_{ijk} = y_{jk} \qquad \text{for j=0, …, K} \tag{12}$$

$$\sum_{j=1}^{N} x_{ijk} = y_{ik} \qquad \text{for i=0, …, K} \tag{13}$$

$$\sum_{(i,j)\in SxS} x_{ijk} \le |\mathbf{S}|-1,\ \mathbf{S} \subseteq \mathbf{V}\left(y_k\right),\ 2 \le |\mathbf{S}| \le N \tag{14}$$

$$x_{ijk} \in \{0,1\} \qquad \text{for i=0, …, N; j=1, …, N} \tag{15}$$

The Fisher and Jaikumar (1981) method is the best known heuristic to solve some mathematical programming approximation of the VRP to optimality. The heuristic replaces $f(y_k)$ with a linear approximation $\Sigma_i\, d_{ik}\, y_{ik}$ and solves the resulting linear generalised assignment problem to get an assignment of customers to vehicles (Fischer, 1995). Once this assignment has been made, a complete solution is obtained by applying any TSP heuristic to get the delivery sequence for the customers assigned to each vehicle.

To obtain the linear approximation, Fisher and Jaikumar (1981) first specify K 'seed' customers $i_1, \ldots, i_K$ which are assigned one to each vehicle. Without loss of generality customers i_k can be assigned to vehicle k for k=1, …, K. Then the coefficient d_{ik} is set to the cost of inserting customer i into the route on which vehicle k travels from the depot directly to customer i_k and back. Specifically, $d_{ik} = c_{0i} + c_{ii_k} - c_{0i_k}$. Clearly, the seed customers define the direction in which each vehicle will travel and the assignment problem completes the assignment of customers to routes given this general framework (Fisher, 1995). Seeds are generally chosen with the following rule. Choose the first seed s_1 to be a customer farthest away from the depot. If k seeds have been chosen, choose s_{k+1} to solve

$$\max_i \min\left\{c_{i0}, \min_{j=1,\ldots,k} c_{is_j}\right\} \tag{16}$$

The algorithm can be extended to accommodate a number of variations and generalisations of the above VRP such as the vehicle routing problem with time windows. This problem consists of designing a set of minimum cost routes, starting and returning at a central depot for a fleet of vehicles that services a set of customers with known demands. The service at a customer has to begin within the time window defined by the earliest time and the latest time when the customer allows the start of the service. Time windows can be hard or soft. In the *soft time windows case* the time window case can be violated at a cost. In contrast, in the *hard time window case* a vehicle is not permitted to arrive at a node after the latest time to begin service. But, if a vehicle arrives too early at a node, it is allowed to wait until the node is ready for beginning service. The costs involved in time-constrained routing and scheduling consist of fixed vehicle utilisation costs and variable routing and scheduling costs. These latter include distance and travel time costs, waiting time costs, and loading/unloading time costs (Desrosiers *et al.*, 1995).

3.3 Constrained shortest path problems

The section will be concluded by considering the shortest path (or least cost) problem (SPP) with time windows. This problem appears as a subproblem in many time constrained routing and scheduling problems and, thus,, deserves some specific attention. The problem consists of finding the least cost route between any two specified nodes in a network whose nodes can only be visited within a specified time interval. The description of the problem that follows is based on Desrosiers *et al.* (1995).

Let G=(**V**, **A**) be a network where **A** is the set of arcs and **V** the set of nodes $\mathbf{N} \cup \{o, d\}$. **N** consists of nodes that can be visited from an origin o to a destination d. With each node $i \in \mathbf{V}$ a time window $[g_i, h_i]$ is associated. A path in G is defined as a sequence of nodes $i_0, i_1, \ldots, i_k$ such that each arc (i_{k-1}, i_k) belongs to **A**. All paths start at time g_o from node i_o and finish at i_K=d no later than h_d. A path is elementary if it contains no cycles. Each arc $(i, j) \in \mathbf{A}$ has a positive or negative cost c_{ij} and a positive duration t_{ij}. Service time at node i is included in t_{ij} for all $i \in \mathbf{N}$. An arc (i, j) in the set **A** is defined to be feasible only if it respects the condition: $g_i + t_{ij} \leq h_j$.

The mathematical programming formulation of the SPP with time windows involves two types of variables: flow variables x_{ij} with $(i, j) \in \mathbf{A}$ and time variables t_i with $i \in \mathbf{V}$. Using this notation the SPP with time windows may be formulated as follows (Desrosiers *et al.* 1995):

$$\min \sum_{(i,j)\in \mathbf{A}} c_{ij}\, x_{ij} \tag{17}$$

subject to

$$\sum_{j\in \mathbf{V}} x_{ij} - \sum_{j\in \mathbf{V}} x_{ji} = \begin{cases} +1 & i=0 \\ \ \ 0 & \text{for } i\in \mathbf{N} \\ -1 & i=d \end{cases} \tag{18}$$

$$x_{ij} \geq 0 \qquad \text{for } (i,j)\in \mathbf{A} \tag{19}$$

$$x_{ij}\left(t_i + t_{ij} - t_j\right) \leq 0 \qquad \text{for } (i,j)\in \mathbf{A} \tag{20}$$

$$g_i \leq t_i \leq h_i \qquad \text{for } i\in \mathbf{V} \tag{21}$$

The objective function (17) attempts to minimise the total travel cost. Constraints (18) and (19) define the flow conditions on the network, while time windows appear in constraint (20). Compatibility requirements between flow and time variables are given in Equation (21). This non-linear problem with time windows is appealing because it can be shown that if the problem is feasible, then there is an optimal integer solution (see Desrosiers *et al.*, 1995).

The problem can be solved by dynamic programming. For the introduction of this approach, define Q(**S**, i, t) as the minimum cost of the path routing from node o to node i [$i\in \mathbf{N}\cup\{d\}$] visiting all nodes in the set $\mathbf{S}\leq\mathbf{N}\cup\{d\}$ only once, and servicing node i at time t or later. The cost Q(**S**, i, t) can be calculated by solving the following recurrence equations

$$Q(\Phi, o, g_o) = 0 \tag{22}$$

$$Q(\mathbf{S}, j, t) = \min_{(i,j)\in \mathbf{A}} \left\{ Q\left(S-\{j\}, i, t'\right) + c_{ij} \text{ with } i\in \mathbf{S}-\{j\},\ t' \leq t - t_{ij}, g_j \leq t' \leq h_i \right\}$$

$$\text{for all } \mathbf{S} \subseteq \mathbf{N} \cup \{d\} \text{ for } j\in \mathbf{S} \text{ and } g_j \leq t \leq h_j. \tag{23}$$

The optimal solution is given by

$$\min_{S\subseteq \mathbf{N}\cup\{d\}} \ \min_{g_d \leq t \leq h_d} Q(\mathbf{S}, d, t). \tag{24}$$

It is worthwhile to note that equation (23) is valid only if $g_j \leq t \leq h_j$. If $t<g_j$, then Q(**S**, j, t) = Q(**S**, j, g_j), and if $t>h_j$, then Q(**S**, j, t) = ∞. The SPP with time windows is NP-hard in the strong sense. Therefore, the above mentioned dynamic programming algorithm suggested by Desrosiers *et al.* (1995) has an exponential complexity and non pseudo-polynomial algorithm is known for this problem.

4 Concluding remarks

GIS-T, once the sole domain of public sector planning and transportation agencies, is increasingly being used in the private sector to support logistics in general and distribution and production logistics in particular. The cost of the technology is now within the reach of even smaller enterprises. The cost of acquiring the data to populate a GIS for transportation-related applications is falling rapidly. The availability of data is paralleled by GPS services to reference locations accurately. These trends suggest that GIS-T has arrived as a core technology for transportation (Sutton and Gillingham, 1997).

The performance of GIS-T software largely depends on how well nodes and links and transportation-related characteristics are arranged into a data structure. The data structure must not only represent the complexities of transport networks in sufficient detail, but also allow for rapid computation of a wide variety of sophisticated network procedures, such as TSP-, VRP- and SPP-algorithms, based on actual network drive time, and not straight-line distances.

References

Assad, A. A. and Golden, B. L. (1995) Arc routing methods and applications. In: *Handbooks in Operations Research and Management Science, Volume 8* (M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser, eds.), pp. 375-483. Elsevier, Amsterdam

Atzeni, P., Ceri, S., Paraboschi, S. and Terlone, R. (1999) *Database Systems* McGraw Hill, Berkshire

Bell, M. and Iida, Y. (1997) *Transportation Network Analysis* John Wiley, Chichester

Bespalko, S. J., Sutton, J. C., Wyman, M., Veer, J. A. van der and Sindt, A. D. (1998) Linear referencing systems and three dimensional GIS, Paper presented at the 1998 Annual Meeting of the Transportation Research Board, TRB Paper No. 981404

Bodin, L., Golden, B. L., Assad A. and Ball, M. (1983) Routing and scheduling of vehicles and crews: The state of the arte *Computers & Operations Research*, **10(20)**, 63-211

Dane, C. and Rizos, C. (1998) *Positioning Systems in Intelligent Transportation Systems* Artech House, Boston

Dantzig, G. B., Fulkerson, D. R. and Johnson S. M. (1954) Solution of a large-scale traveling salesman problem *Operations Research*, **7**, 58-66

Desrosiers, J., Dumas, Y., Solomon, M. M. and Soumis, F. (1995) Time constrained routing and scheduling. In: *Handbooks in Operations Research and Management Science, Volume 8* (M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser, eds.), pp. 35-139. Elsevier, Amsterdam

Dueker, K. T. and Ton, T. (2000) Geographical information systems for transport. In: *Handbook of Transportation Modelling* (D. A. Hensher and K. J. Button, eds.), pp. 253-269. Pergamon, Amsterdam

Fisher, M. (1995) Vehicle routing. In: *Handbooks in Operations Research and Management Science, Volume 8* (M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser, eds.), pp. 1-33. Elsevier, Amsterdam

Fisher, M. and Jaikumar, R (1981) A generalized assignment heuristic for vehicle routing *Networks,* **11**, 109-124

Fohl, P., Curtin, K. M., Goodchild, M. F. and Church, R. L. (1996) A non-planar, lane-based navigable data model for ITS. In: *Proceedings, Seventh International Symposium on Spattial Data Handling, Delft, August 12-16* (M. J. Kraak and M. Molenaar, eds.), pp. 7B, 17-7B, 29.

Glover, F. (1989) Tabu search, part I *ORSA Journal on Computing*, **1(3)**, 190-206

Glover, F. (1990) Tabu search, part II *ORSA Journal on Computing*, **2(1)**, 4-32

Golden, G. B. and Assad, A. A. (1986) Perspectives on vehicle routing. Exciting new developments, *Operations Research*, **14**, 803-810

Goodchild, M. F. (1992) Geographical data modeling *Computers and Geosciences*, **18(4)**, 401-408

Goodchild, M. F. (1998) Geographic information systems and disaggregate transportation modeling *Geographical Systems*, **5**, 19-44

Goodchild, M. F. (2000) GIS and transportation: Status and challenges *GeoInformatica*, **4(2)**, 127-139

Johnson, D. S. (1990) Local optimization and the traveling salesman problem. In: *Automata, Languages and Programming* (G. Goos and J. Hartmanis, eds.), pp. 446-461. Springer, Berlin, Heidelberg and New York

Kirkpatrick, S., Gelatt, C. D. and Vecchi M. P. (1983) Optimization by simulated annealing *Science,* **220**, 671-680

Kwan, M.-P., Golledge, R. G. and Speigle, J. M. (1996) A review of object-oriented approaches in geographic information systems for transportation modelling, Draft, Department of Geography, University of California at Santa Barbara

Laporte, G. (1992) The travelling salesman problem: An overview of exact and approximate algorithms *European Journal of Operational Research*, **59(2)**, 231-247

Lawler, E. L., Lenstra, J. K., Rinnoy Kan A. H. G. and Shmoys, D. B. (1985) *The traveling salesman problem: A guided tour of combinatorial optimization* Wiley, Chichester

Lin, S. (1965) Computer solutions of the travelling salesman problem *Bell System Technical Journal*, **44**, 2245-2269

Lin, S. and Kernighan, B. (1973) An effective heuristic algorithm for the traveling salesman problem *Operations Research,* **21**, 498-516

Longley, P. A., Goodchild, M. F., Maguire, D. J. and Rhind, D. W. (2001) *Geographic Information Systems and Science* John Wiley, Chichester

Miller, H. J. and Shaw, S.-L. (2001) *Geographic Information Systems for Transportation. Principles and Applications* Oxford University Press, Oxford

Nyerges, T. L. (1990) Locational referencing and highway segmentation in a geographic information system *ITC Journal,* **60(3)**, 27-31

Powell W. B., Jaillet, P. and Odoni, A. (1995) Stochastic and dynamic networks and routings. In: *Handbooks in Operations Research and Management Science, Volume 8* (M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser, eds.), pp. 141-295. Elsevier, Amsterdam

Potvin, J.-Y. (1993) The travelling salesman problem: A neural network perspective *ORSA Journal on Computing*, **5(4)**, 328-348

Spear, B. D. and Lakshmanan, T. R. (1988) The role of GIS in transportation planning and analysis *Geographical Systems*, **5**, 45-58

Sutton, J. (1997) Data attribution and network representation issues in GIS and transportation *Transportation Planning and Technology*, **21**, 25-44

Sutton, J. and Gillingwater, D. (1997) Geographic information systems and transportation – Overview *Transportation Planning and Technology*, **21**, 1-4

Thill, J.-C. (2000) Geographic information systems in for transportation in perspective *Transportation Research Part C*, **8**, 3-12

Vonderohe, A. and Hepworth, T. (1996) *A Methodology for Design of a Linear Referencing System for Surface Transportation*, Research Report, Sandia National Laboratory, Project AT-4567

Waters, N. M. (1999) Transportation GIS: GIS-T. In: *Geographic Information Systems, Vol. 2: Management Issues and Applications* (P. A. Longley, M. F. Goodchild, D. J. Maguire and D. W. Rhind, eds.), pp. 827-844. John Wiley, Chichester

Worboys, M. F. (1995) *GIS. A Computing Perspective* Taylor & Francis, London