Dynamic Delivery of Transportation Services with GIS

Jay Lee

Jay Lee is an assistant professor in the Department of Geography at the University of Georgia. His research interests include various aspects of geographic information systems in algorithm development and geographic information analyses methods.

Abstract: *Geographic information systems (GIS) provide an enhanced environment for analysis, evaluation and decision-making in many areas of urban and regional planning. The emergence of GIS has enabled significant advances to be made in the storage, retrieval, processing and presentation of large amounts of geographically referenced data. However, in applying GIS to the delivery of transportation services, a great deal of additional research and development has to be mobilized to exploit the potential of GIS technology for routing transportation vehicles, especially in dealing with real-time inputs in dynamic settings. As the first step, this paper demonstrates the feasibility of adding functional modules to GIS in order to process on-line spatial information to update road networks, to locate them, to incorporate real-time inputs of the services requested, and to perform on-screen routing operations. Although still being limited to a single-vehicle/multiple-location case, a prototype computerized routing system is used to describe the structure of a hierarchical, integrated spatial data model for transportation networks; simple heuristic algorithms are used for handling simple, dynamic routings.*

Service vehicles running on a transportation network can be routed by different methods. *Static* routings refer to the routes and schedules that service vehicles follow when the numbers and the locations of the requested services are all known *prior to* operations. Examples for static routings include routings for public or school buses, garbage collection trucks, scheduled ships and aircraft. Alternatively, *dynamic* routings refer to last-minute design and execution of routes for service vehicles that are en route. Examples in this case include taxicabs, dial-a-ride services, delivery trucks, unscheduled ships and aircraft.

There is no single algorithm to solve all vehicle routing problems. Appropriate algorithms depend on the types of service vehicles and their settings. For example, algorithms for routing school buses will not perform as well for routing garbage trucks or taxicabs. Because of this complexity, there exists a large body of literature on problems of vehicle routing. Good reviews are available in Lin and Kernighan (1973), Mole (1979), Bodin et al. (1983), Christofides (1985), Bott and Ballou (1986), Golden and Assad (1987).

Dynamic routing problems require special attention for processing real-time inputs and are more critical in time constraints than are static problems; thus they compose their own research area. Literature on dynamic vehicle routing problems can be found in reviews scattered in several related fields: Psaraftis (1980), Larson and Odoni (1980), Brown and Graves (1981), Bell et al. (1983), Psaraftis (1983), Powell (1985), Sexton and Bodin (1985), Desrosiers, Dumas and Soumis (1986), Bertsekas and Gallager (1987), Golden and Assad (1988).

This paper first discusses the basic differences between static and dynamic vehicle routing problems. Special characteristics and requirements for simple dynamic settings are reviewed as the basis for constructing a computerized system for dynamic routings. A hierarchical data model which integrates network and related information into a spatial database is described also for this purpose. As an example, a prototype system is presented to demonstrate the concepts developed in this paper. The dynamic settings in this paper emphasize the handling of the real-time inputs.

It is, of course, impossible to include all types of service vehicles and various settings (such as time window, queuing considerations, driver work rules, vehicle capacity, etc.) in this discussion. However, this paper, hopefully, will initiate ideas and stimulate the construction of systems that deal with other real-world constraints in models.

Dynamic Vehicle Routing

Dynamic vehicle routings can be defined as the dispatching of vehicles to satisfy multiple demands for services that evolve in a real-time (dynamic) fashion. Vehicles in this category include taxicabs, delivery trucks, ships, aircraft, etc. Services by these vehicles usually include picking up, delivering and dropping off passengers (or goods) at some stops in a transportation network. Routes are represented by a series of connected network links, e.g., street blocks in a transportation network. Inputs to a dynamic vehicle routing problem include the nature of service, the locations and the time of services requested. Example networks may be street networks or voyage networks for ships or aircraft. Stops in a network to be included in the vehicle routes may be pre-fixed, or non-fixed. Stops may be known *prior to* routing operations or may not be known in advance. If not known in advance, they will become real-time inputs during route execution. The format of inputs to the vehicle routing system determines the nature of a routing problem.

If the assumed inputs, the numbers and the locations of the requested services, to a vehicle routing problem do not change, either during the execution of the algorithm that solves it or during the eventual execution of the route, it is said to be a *static* vehicle routing problem. Alternatively, inputs may (and generally will) change (or be updated) during the execution of the algorithm and/or the eventual execution of the route in a *dynamic* setting. It should be noted that algorithm execution and route execution are processes that evolve concurrently in a dynamic situation.

There are many differences between static and dynamic vehicle routing problems. In a dynamic situation, the time dimension usually is essential and the routing problems may be open-ended. Future information may be imprecise or unknown *prior to* routings or the execution of routes. Therefore, an efficient information update mechanism is essential. In addition to the time constraints, resequencing and re-assignment decisions may be warranted because of real-time inputs. Usually, fast computation speed and fast on-line access to the system and to the database are necessary.

A static routing system has the luxury of knowing all inputs, i.e., customer requests, cancellations, etc., in advance. Solution speed is less critical, however; in a dynamic setting, the routing system is subject to imprecise or unknown inputs as well as greater time constraints. In many situations, a dynamic routing system would be forced to adapt heuristic procedures (mostly local re-routings) to accommodate real-time inputs for quicker solutions which are sometimes not as optimal.

Human Interaction and Re-start Capability

To incorporate the real-time input setting, a dynamic routing system needs to be interactive with its users (for example, vehicle dispatchers). A static routing system may be run in *batch* mode but a dynamic routing system needs to have human interaction, or even intervention, not only because of real-time inputs but also because of greater system control by the human. Some obvious, "quick-and-dirty" routing executions may be performed by experienced dispatchers to avoid otherwise lengthy computation time. A system may become "hung up" with an extreme situation, e.g., a request that is located geographically far away from all other requests. If it would be excessively nonprofitable for the system to include this request at a particular time, it would be deferred (or denied) by the system operator indefinitely so that other requests may be served more efficiently. "Human in loop" would allow system users to overwrite a computer at will and to re-start the system whenever it is necessary. It is also necessary that dynamic routing systems have the capabilities of efficiently updating any routes and schedules at any time within the execution of a plan, without compromising key decisions already made.

Hierarchical Data Structure and User-Friendly Interface

Besides the need for human interaction and the re-start capability, a hierarchical data structure and a user-friendly interface of a routing system are also extremely important for a dynamic routing system. In a dynamic setting, the first step in routings is usually to identify the locations of the service requests within the transportation network. A hierarchical data structure allows the routing system to search for the location of any service request only through those relevant data (of the specific level of aggregation). It would drastically reduce the computation time required. Data links, i.e., street blocks, and point data are usually preprocessed and organized into different levels within the hierarchy. When performing routing operations, relevant data links or locations can then be identified with quick and direct searches.

In practice, the hierarchical data structure can be implemented in many formats. A simple example for implementing this concept is to utilize GIS editing routines to attach each network link (e.g., street block) with a hierarchical code: the freeways and the major arterial routes are associated with higher hierarchical codes and local street blocks are associated with lower hierarchical codes. Additional criteria such as number of lanes can be used to further classify link codes. For routes of shorter distances, local street blocks are used. Routes of longer distances (e.g., covering a large metropolitan area) can be computed on a system which includes only the freeways and the major arterial routes and then connect the requests for services by locally loading the location to the nearest arterial routes.

A user-friendly vehicle routing system implies a system that is easy to operate; for example, a system with graphic display to assist users in constructing spatial interrelations among information used in the system will be preferred to a system with only text ouput. Static routing procedures typically are not organized in an interactive environment as they usually run in batch mode with fixed input/output formats. Routing procedures in a dynamic setting, however, are very different and require a higher degree of interaction between the system and the operators.

Several unique features of an interactive system describe a user-friendly dynamic routing system. First of all, a graphic display is a must in a dynamic routing system so that users can quickly and visually construct spatial relationships among all information used. Decisions such as arranging priority of requested services by their relative locations to an existing route can be simplified through visual inspection. Otherwise, lengthy computation may be necessary for this purpose, and therefore prolong the response time by the system. Secondly, flexible input formats are also much preferred in a dynamic routing system as they reduce the input preparation time.

Different ways to input service requests can significantly affect the efficiency of the routing system. For example, by pointing to points on screen with a mouse rather than by typing in coordinates of the points will obviously result in tremendous savings in both preparation time and computation time. Finally, a menu-driven system interface is much preferred if compared with a command-driven system. The format of a menu-driven user interface does not require users to memorize many command formats and parameters. Options of operations are usually listed in the menu for users to make decisions. Training time for using the system is therefore drastically reduced.

In dynamic vehicle routing procedures, the information management and user interface issues are much more important than in static routing procedures. An efficient dynamic routing procedure implies more than just an efficient core routing operation. Actually, the core operation itself must be designed in such a way that the above features concerning human interaction, re-start capability, and user-friendliness can be easily implemented.

If more advanced features are to be included, an interactive user interface can be used to improve the user's decision-making process during routing operations. For example, a system may suggest alternative routes to users after analyzing routing information. The users could choose among them according to their purposes and experience. Moreover, route evaluation is possible when a system allows users to dramatically change a pre-set route and then evaluate the changed route against the optimal route. Users of such systems would be able to learn from the experience.

In summary, static and dynamic routings differ in how inputs are processed, in system response time, in system design and requirements. With these differences, one can expect the algorithms for dynamic routings to be unique from the static routing algorithms. However, there have not been enough dynamic algorithms developed for general dynamic routing purpose. In practice, dynamic routings usually adopt some of the static routing algorithms and run them in iterations to accommodate real-time inputs. Specifically, a static routing algorithm is used to generate an initial solution, and then the system relies on repeated operations for all subsequent input updates. In the next section we will discuss two ways to adopt static algorithms into a dynamic setting.

Algorithms for Dynamic Routings

Most dynamic vehicle algorithms were developed for vehicle allocation rather than specifically for routing purposes. In practice, a dynamic vehicle routing system usually adopts static routing algorithms and run them again every time a real-time input is received. Usually, the adopted algorithm must undergo a significant degree of re-design, most of it heuristic, to be tailored to the nature of the dynamic scenario.

Adoption of Static Routing Algorithms

The first and the most common approach to adopt static routing algorithms into a dynamic setting is to re-run the static routing procedures virtually from scratch each time a (significant) revision of the input occurs. For example, a new input or request appears, or another one cancels the request, or a vehicle breaks down, etc.

This approach may approximate more optimal solutions than the heuristic approach (to be described later) but will certainly take up more computation time. This will lead to delay in response time from system to users. Depending on the types of routed vehicles and the size of the transportation network, delay may range from insignificant to very significant. But in general, for a dynamic system which routes real-time inputs, this delay of responding time may be very undesirable.

Dynamic Routing with Heuristic Algorithms

An alternative and more realistic approach to the adoption of the static routing algorithms would be to handle dynamic input updates *via* a series of "local" operations, applied *via* the execution of an insertion/deletion heuristic (possibly followed by an interchange heuristic), after the static core algorithm is run. This would involve running the static algorithm just to initialize the process (for example, once a day), and rely on "local" operations for all subsequent input updates.

Local operations provide a reasonable way to handle dynamic input data. The principal advantage of local operations is the execution speed. Because it only revises established routes locally, it avoids wasting computational time to check other parts of the transportation network that are not relevant to the current input. It should be noted, however, that local re-routings work well when the number of new service requests is relatively small with respect to the number of service requests included in an existing route. But as the number of new requests increases relative to the number of known points for the existing route, the performance of local re-routings will be deteriorated. Furthermore, the spatial density of the service requests also affects the performance of local re-routings. Lower density transportation networks usually are more suitable for local re-routings. Also, the local re-routing algorithms work well for the situation of single-vehicle/multiple-location route, but may not work as well for multiple-vehicle/multiple-location situations.

Another issue concerning the heuristic approach of using the local re-routing algorithm is the tradeoffs between computing time saved and the possibility of obtaining optimal routes. For example, the local re-routing algorithm can be modified to allow users to specify different numbers of points on the existing route to be included for rerouting. One extreme is as the aforementioned situation wherein a single new request is inserted into an existing pair of points. Another extreme is to include all points in the re-routing. This implies a total re-routing and naturally will require more intensive computation. Users of such systems will have to make decisions in terms of the trade-offs between saving computation time and obtaining optimal solutions, according to their own objectives.

Dynamic Routing With GIS

A vehicle routing system is currently being tested by the author based on concepts described so far. The earlier version of the system was designed for county-level governmental entities for various static-routing purposes: school buses, garbage collection trucks, etc. Later it was modified and enhanced to process dynamic routings for service vehicles such as delivery trucks. This system is described here as a demonstration. In its current version, the system is capable of routing one vehicle at a time with real-time inputs. The local re-routing algorithm uses only one point at each side of the new request for modifying an existing route with respect to a real-time input. Although it is capable of performing both static and dynamic routing operations, it is primarily developed for simple dynamic routings for delivery vehicles. Dynamic routings with other constraints, such as time windows, vehicle capacities, and others, will require further efforts to customize the system in the future.

The prototype system consists of a database and several modules: a NETWORK database which integrates both spatial and non-spatial data for the routing system; other modules for routing operations include ROUTE BUILDER, ROUTE MODIFIER, and ROUTE REPORTER. All routing modules share the spatial database, NETWORK.

An Integrated Database

The data required by the prototype dynamic routing system can be classified into two broad categories: graphic (spatial) and tabular (non-spatial) data. Graphic data refer to coordinates and their spatial relationships in a transportation network. Each data link represents a street block in the network. A series of coordinate pairs describes the positions and directions of the link. The spatial relationships among data links refer to how these links are interconnected in the network.

Tabular data refer to the information associated with data links in the transportation network. Each data link can be associated with a number of attributes such as number of lanes, one/two-way street, traffic flows, the distance of the link, number of houses located along the link, the range of their addresses, and others.

Graphic and tabular data, which may be stored in separate files, can be integrated into a GIS. GIS provides an enhanced environment for analysis, evaluation and decision-making in many application areas by integrating both tabular and graphic data. Once the data are organized into a spatial database, functional modules can be added to the GIS to carry out tasks. In our case, the functional modules added are the ROUTE BUILDER, the ROUTE MODIFIER, and the ROUTE REPORTER.

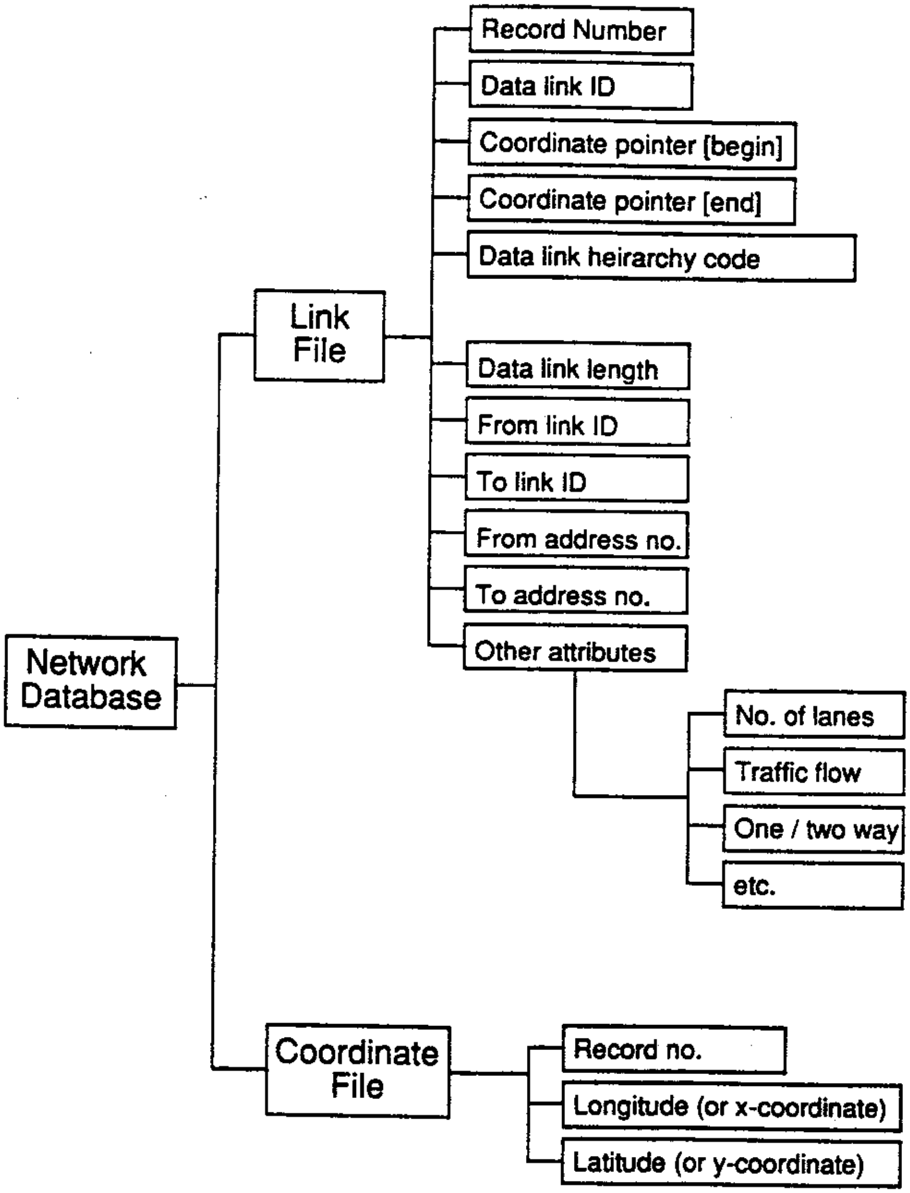
Using GIS, other tabular data for fleet management also can be incorporated into the database if desired. Maintenance records, fuel-consumption records, personnel records and other scheduling information can also be integrated.

NETWORK Database

The database, NETWORK, integrates both graphic and tabular data of the transportation network. GIS editors are used to input, edit, append, sort, or display data. An interface establishes the linkages between two groups of data. The structure of the database of the prototype system is given in Figure 1.

In the prototype database, a transportation network is stored by links. Each link is stored as a record in the link file. Each record contains link identification, coordinate pointers, hierarchical codes, and other attributes. While each link is defined by an "ID," its positions and directions are defined by a series of coordinate pairs. Coordinate pointers, recorded as two attributes for each link, point to the beginning and the ending positions of coordinate pairs in the coordinate file. A series of coordinate pairs can be

FIGURE 1.
NETWORK Database.



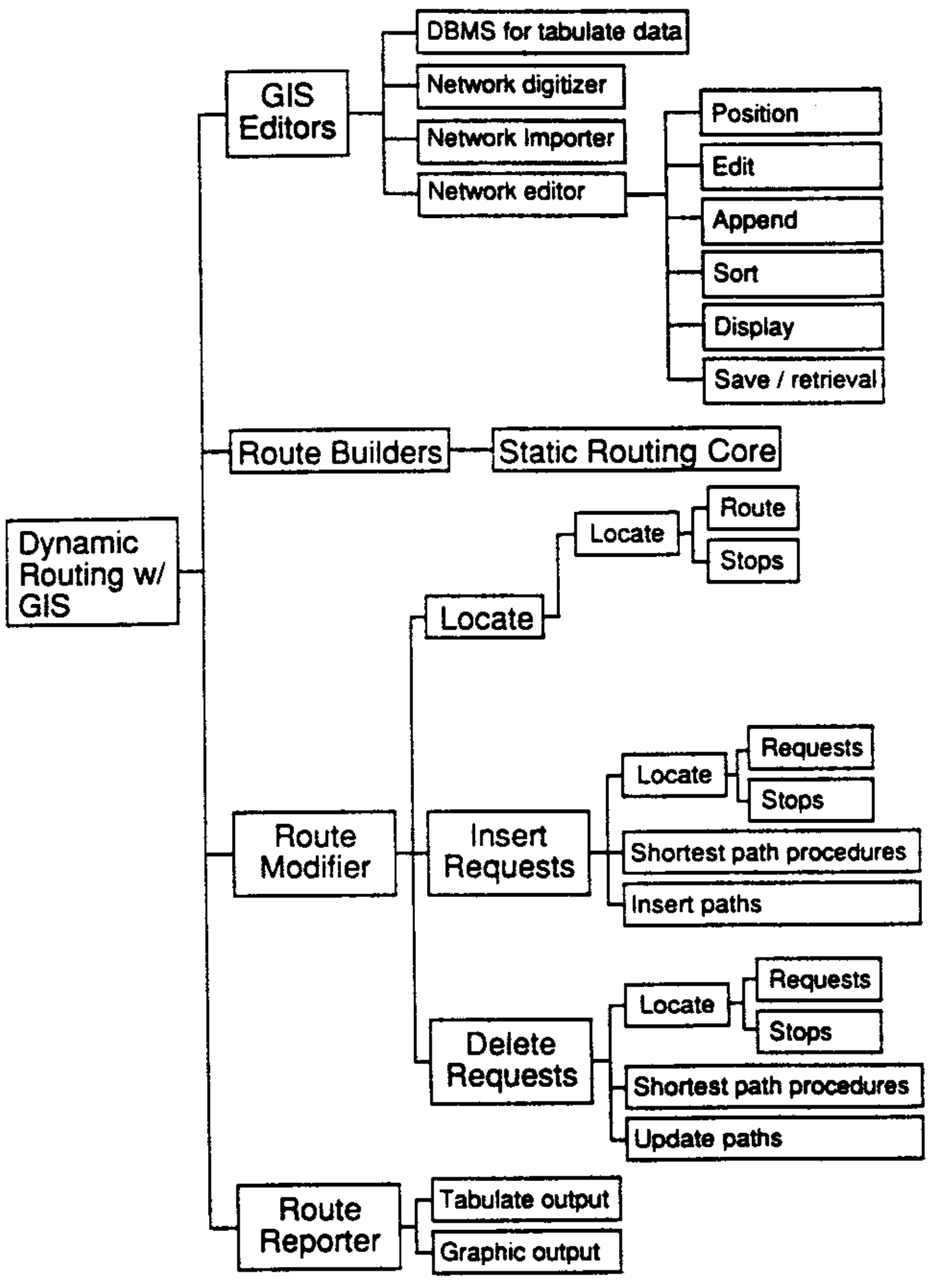
extracted from the coordinate file to describe the link's position and direction. The hierarchical codes of the network links provide means for routing at various "resolutions." Experience suggests that higher, hierarchical network links can be used to construct route segments between distant service requests. Many features can be recorded as additional attributes for each link. In the prototype system, each link record contains link (street) name, link's length, address ranges, etc.

Functional Modules for Routings

The actual routing procedures can be conducted by adding function modules to a GIS. In addition to GIS editors, which may be used to input and edit street networks, the design and modification of routes in the prototype system is achieved by adding three modules: a ROUTE BUILDER, a ROUTE MODIFIER, and a ROUTE REPORTER. The structure and functions of these three modules are shown in Figure 2. The routing procedures begin by a static routing with the ROUTE BUILDER. Using information (of service requests) initially available, an initial route is established. The execution can be started any time after this point, depending on the arrival of real-time inputs. Upon receiving real-time inputs, the system switches to the ROUTE MODIFIER to perform "local" re-routing and re-sequencing. Real-time inputs are incorporated accordingly. A new request can be inserted to the existing routes and any cancellation can be deleted from the existing routes.

The algorithms used for local re-routings perform quick insertions or deletions on existing routes. They are not designed to re-route the entire set of updated requests with the entire transportation network. As the local re-routing algorithm deals with only part of the entire transportation network, the response time is dramatically reduced. However, it should be noted that routes designed in this manner some-

FIGURE 2.
Functional Modules And GIS.



times are not always optimal if compared with the routes designed by applying static routings from scratch every time a new request is incorporated.

Both insertion and deletion procedures perform local re-routing by the following steps (also shown in Figure 2):

1) Locate the request in the path sequence of the identified route;
2) Identify the two neighboring stops (closest to the location of the request) in the path sequence;
3) For insertion:
   a) Compute the shortest paths from the request location to the two neighbor stops;
   b) Delete the path between two neighbor stops from route;
   c) Replace with two shortest paths (from the location of request to two neighbor stops);
4) For deletion:
   a) Delete the paths between the location of request to two neighbor stops;
   b) Compute the shortest path between the two neighbor stops and add it to the route;
5) Ready for the next real-time input.

In addition to the ROUTE BUILDER and the ROUTE MODIFIER, a ROUTE REPORTER is implemented to produce output in both graphic and tabular forms. A map on which the designed routes are highlighted can be plotted on screen. Figure 3 demonstrates an example from the prototype system. Street links included in the route are highlighted by a different color. Function keys offer various options a user can select during routing operations. Route statistics, e.g., lengths, number of service requests covered, number of U-turns, left turns included, also can be reported. If preferred, a listing of street names can be generated to accompany the graphic reports (Table 1).

Conclusions and Future Research Issues

In recent years, major developments have occurred in two areas: geographic information systems, and vehicle location and tracking. Both are expected to impact routing and dispatching significantly. Without GIS to provide an interactive, integrated environment for vehicle routings, routing procedures are limited to mostly static settings or their simple iterations. Processing tabular and graphic data with GIS, dynamic routings become feasible and have the potential for a wider range of applications.

In the preceding sections, several key differences between static and dynamic vehicle routing problems are discussed. In general, solution-speed time constraints in a static setting are less critical than that in a dynamic setting. Dynamic routing problems in some situations have different emphases than those of static routing problems. As a result, routing procedures are also conceptually different. For a dynamic routing system, achieving routings in less time while satisfying certain profit-driven criteria seems to be more important than achieving an optimal route of the least operation cost as in the case of static routings.

It should be noted that in applying these concepts for practical applications, a computerized routing system would have to be tailored to meet the specific settings for each particular application. This involves incorporating specific constraints into the computerized routing systems. A number of possible constraints in routing systems have been treated to various degrees of success in the literature of operations research and transportation management. These constraints include time windows, driver work rules, vehicle capacity constraints, and

FIGURE 3.
Prototype Dynamic Routing System (Graphic Display).

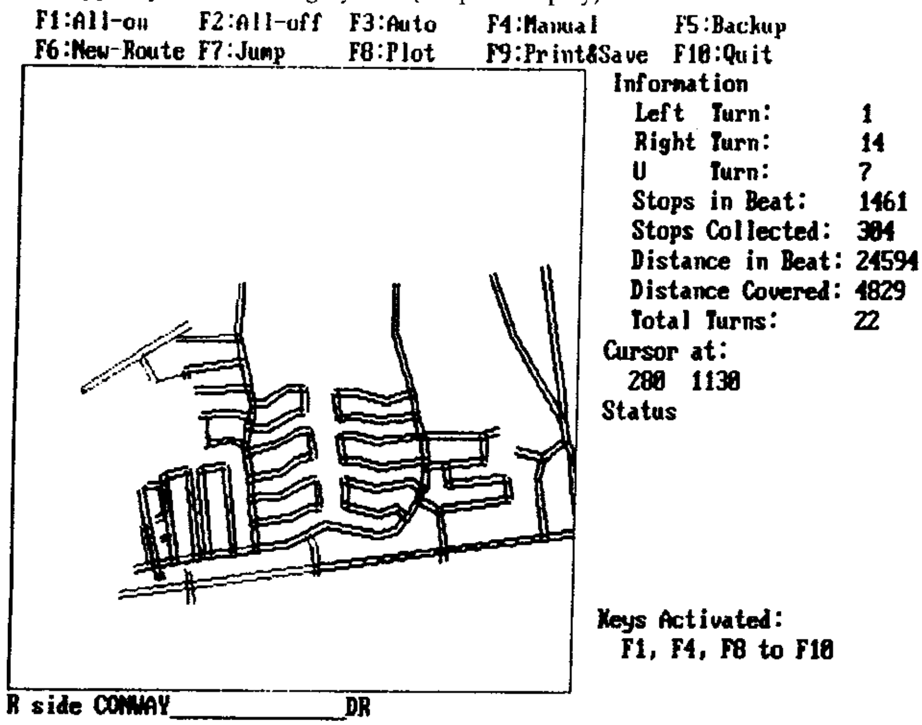


TABLE 1. Prototype Dynamic Routing System (Tabular Output).

Street	From	To	Side	Stops	Next turn
BRADLEY_______AV	JALNA_BV	LON__MUN	right	0	right
JALNA_______BV	CONWA_DR	BRADL_AV	right	0	right
CONWAY_______DR	JALNA_BV	CONWA_CT	right	29	right
CONWAY_______CT	CONWA_DR	________	right	5	U turn
CONWAY_______CT	CONWA_DR	________	right	0	right
CONWAY_______DR	CONWA_CT	JALNA_BV	right	46	U turn
CONWAY_______DR	CONWA_LN	JALNA_BV	right	0	right
CONWAY_______LN	CONWA_DR	CONWA_DR	right	25	U turn
CONWAY_______LN	CONWA_DR	CONWA_DR	right	25	right
CONWAY_______DR	CONWA_LN	CONWA_LN	right	23	ahead
CONWAY_______DR	JALNA_BV	CONWA_LN	right	5	right
JALNA_______BV	CONWA_DR	CONWA_DR	right	26	U turn
JALNA_______BV	________	CHESW_CI	right	5	ahead
JALNA_______BV	CHESW_CI	CONWA_DR	right	19	right
CONWAY_______DR	CHESW_CI	JALNA_BV	right	8	right
CHESWICK_______CI	CONWA_DR	SARAH_CR	right	1	right
SARAH_______CR	CHESW_CI	CHESW_CI	right	24	right
CHESWICK_______CI	SARAH_CR	JALNA_BV	right	4	U turn
CHESWICK_______CI	CONWA_DR	JALNA_BV	right	10	left
SARAH_______CR	SARAH_CT	CHESW_CI	right	1	right
SARAH_______CT	SARAH_CR	________	right	7	U turn
SARAH_______CT	SARAH_CR	________	right	0	right
SARAH_______CR	CHESW_CI	SARAH_CT	right	7	right
CHESWICH_______CI	SARAH_CR	SARAH_CR	right	16	U turn
CHESWICK_______CI	CONWA_DR	JALNA_BV	right	18	ahead

others. What constraints and how to incorporate them into a dynamic routing system are indeed pragmatic issues for future research.

Another direction for future research is to improve the system so that routing information can be analyzed and presented to the user in a form that will facilitate improved decision-making. For example, the computerized routing system will suggest alternative routes; the users can choose one that best serves the user's purpose and experience. Another option will allow users to alter a pre-set route dramatically. The system could evaluate the change against the final chosen route (after the dynamic modification) or against an optimal route computed *post hoc* and show how the route could have been improved. A wide range of similar conditions can be easily identified for future research.

In summary, the use of GIS in a dynamic routing system allows the routing efforts to be performed with a hierarchical structure, and that dramatically improves the computation speed of routings by bypassing the irrelevant parts of the network. In addition, the ability of GIS to process geographic information, together with tabular and graphic information, allows users to enter, edit, and to display transportation networks efficiently. Interaction among system and users *via* GIS graphic displays makes routing systems more user-friendly with quick realization of spatial interrelationships for the system operators. Finally, adding functional modules to GIS not only enhances GIS functionality but also expands feasible application areas for GIS.

References

Bell, W., L.M. Dalberto, M.L. Fisher, A.J. Greenfield, R. Jaikumar, P. Kedia, T.G. Macj, and P.J. Prutzman. 1983. "Improving the Distribution of Industrial Gases with an On-line Computerized Routing and Scheduling Optimizer." *Interfaces*, 13:4–23.

Bertsekas, D.P. and R. Gallager. 1987. *Data Networks*, NJ: Prentice-Hall.

Bodin, L., B. Golden, A. Assad, and M. Ball. 1983. "Routing and Scheduling of Vehicles and Crews: the State of the Art." *Computers and Operations Research*, 10:97–108.

Bott, K. and R. Ballou. 1986. "Research Perspectives in Vehicle Routing and Scheduling." *Transportation Research*, 20A:239–243.

Brown, G. and G. Graves. 1981. "Real-time Dispatch of Petroleum Tank Trucks." *Management Science*, 27:19–31.

Christofides, N. 1985. "Vehicle Routing." in *Combinatorial Optimization Annotated Bibliographies*, edited by O'hEigeartaigh, M. et al., New York: Wiley, pp. 148–163.

Desrosiers, J., Y. Dumas, and F. Soumis. 1986. "A Dynamic Programming Solution of the Large-scale Single-vehicle Dial-a-ride Problem with Time Windows." *American Journal of Mathematical and Management Sciences*, 6, Nos. 3, 4: pp. 301–325.

Golden, B. and A. Assad. 1987. "Perspectives on Vehicle Routing: Exciting New Developments." *Operations Research*, 34:803–810.

Golden, B. and A. Assad. 1988. *Vehicle Routing: methods and studies*. New York: North-Holland. 479 pp.

Larson, R.C. and A.R. Odoni. 1980. *Urban Operations Research*. NJ: Prentice-Hall.

Lin, S. and B.W. Kernighan. 1973. "An Effective Heuristic Algorithm for the Travelling Salesman Problem." *Operations Research*, 21:498–516.

Mole, R. 1979. "A Survey of Local Delivery Vehicle Routing Methodology." *Journal of Operational Research Society*, 30:245–252.

Powell, W. 1985. "An Operational Planning Model for the Dynamic Vehicle Allocation Problem with Uncertain Demands." Department of Civil Engineering, Working paper EES-85-15, Princeton University.

Psaraftis, H.N. 1980. "A Dynamic Programming Solution to the Single-vehicle Many-to-many Immediate Request Dial-a-Ride Problem." *Transportation Science*, 14:130–154.

Psaraftis, H.N. 1983. "An Exact Algorithm for the Single Vehicle Many-to-many Dial-a-ride Problem with Time Windows." *Transportation Science*, 17: pp. 351–357.

Sexton, T.R. and L. Bodin. 1985. "Optimizing Single Vehicle Many-to-many Operations with Desired Delivery Times: I. Scheduling, II. Routing." *Transportation Science*, 19: pp. 378–410.