



# GEO

## **A Geographic Information System for Transportation Applications**



**Guy Lapalme  
Jean-Marc Rousseau  
Suzanne Chapleau  
Michel Cormier  
Pierre Cossette  
Serge Roy**



# ROUTE

**R**outing, scheduling and dispatching problems are among the major components of the activities related to the transportation of goods (or persons) from the point of production (pick-up) to the point of delivery (destination). The goal is to provide the best service to the customer at minimum cost to the producer. As these two objectives are often contradictory, namely that better service is more costly, the transportation enterprise must optimize its resources to find an economical way to distribute its goods or services while maintaining the goals and constraints of its marketing strategy.

Physical distribution is intimately related to the other activities of the enterprise and is a major factor of the price of goods and services. Kearney [7] has shown that distribution costs represented 16% of the net value of goods and a quarter of this was incurred by the transportation of finished products from warehouses to clients. Optimization of transportation costs is thus a very important economic problem that must deal with all kinds of characteristics and constraints on route lengths, vehicle capabilities, working conditions of drivers, schedules, and on types of merchandise. It is

generally recognized that optimization procedures in transportation can reduce delivery cost from 10% to 20% and even more when enterprises solely provide transportation (for example, bus service authorities, postal services, or couriers).

A great number of algorithms and methods have been developed for solving routing and scheduling problems but many aspects and constraints have yet to be taken into account [1]. These solutions are almost always heuristic ones because no efficient algorithms have yet been developed for the sizes required for practical problems. Solving a routing and scheduling problem involves many compromises within a set of conflicting goals and constraints: using the main roads rather than the shortest route, giving a better average service rather than a better overall service. Consequently, the results must be evaluated by hand. This process usually prompts more modifications which in turn must be evaluated.

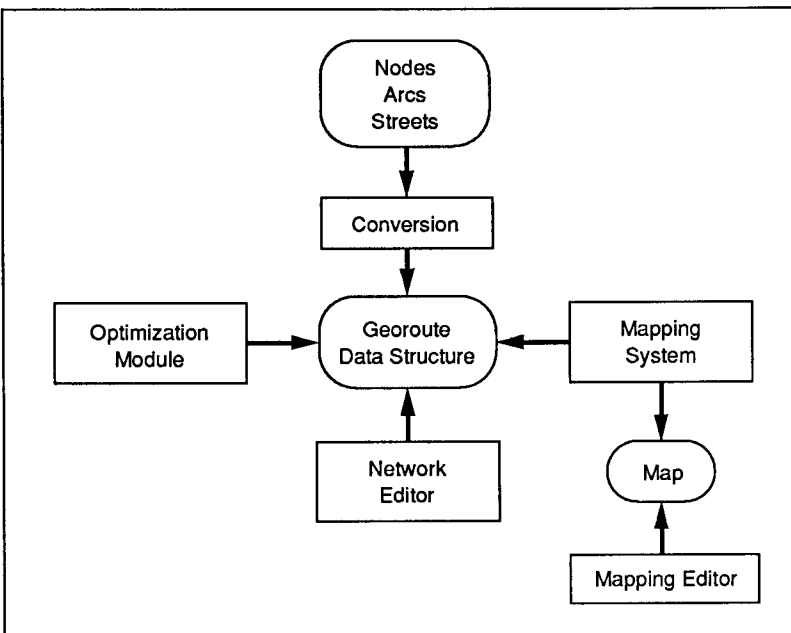
The sensitive spot in route planning is the design of algorithms, but other activities are also implied: finding the location of the clients, computing the distance between them, evaluating their service time, managing the fleet of vehicles.

When all data is collected and when the objectives are well defined, an appropriate algorithm is run. Finally, the solutions are evaluated and quite often modified to take into account constraints that could not be modeled in the algorithms. Interactive graphics can help in all the steps involved in the planning and implementation of routes [8]. The clients and the resources can be more easily registered using graphic input and checked immediately with graphic output because this information usually has geographic components. The evaluation of alternative solutions is also eased because the human eye is very quick at detecting irregularities and discrepancies on a route.

For many years, the Centre de recherche sur les transports de l'Université de Montréal has been involved in the development of graphic tools combined with operation research methods for dealing with transportation problems. This article describes *GeoRoute*<sup>™</sup>, a graphical software package for the management of information associated with street networks. This system is an integration of many years of research and development in this area in cooperation with clients.

*GeoRoute* is unique in several ways:

Metropolitan area	Population	Number of streets	Number of links	Size of file in MB
Toronto	3,000,000	20,183	95,925	28
Montreal	2,500,000	11,176	65,712	20
Hamilton	400,000	4,594	22,913	12
Edmonton	350,000	1,579	21,034	8
Hull, Gatineau	150,000	1,787	10,164	4
Boston (CBD)	750,000	5,245	20,618	8



**FIGURE 1.**

**Overview of the *GeoRoute* system modules**

- It is based on a street network structure specifically designed to store and manipulate large networks and runs on personal and minicomputers (MS-DOS microcomputers and VAX/VMS minicomputer). The network is not in vector or bit-mapped form, but rather is based on an original street network data structure.
- It can handle both node-routing and arc-routing problems.
- It has an automated cartography facility that generates maps from the database of streets and intersections. As the maps are computed from the database information, they can be easily

customized and they can be edited using the interactive map editor, also part of the system.

- It has a very flexible route optimization system, generating routes that can also be graphically modified by the interactive route editor.

These features are the modules of *GeoRoute* that all evolve around the *GeoRoute* data structure as illustrated in Figure 1. They are described in the following sections.

**Interactive Network Editor**

The Network Editor is a graphics-based tool for the display and manipulation of street networks and

associated information. The network is stored as a series of streets, street segments and intersections. Traffic constraints are stored for each segment and intersection as are civic addresses and zone information. A relational database can also be linked with the network to store information that describes the items located on the network. The *GeoRoute* network structure divides the area into small submaps. These are transparently loaded as needed by the application, using an algorithm similar to the ones used in virtual memory systems. Since the operations performed on a network are of a special nature, we were able to optimize the accesses to the database of regions. This means that the maximum size of the network manipulated is limited only by the amount of disk space available. Table 1 presents the sizes of some of the networks already transferred into the *GeoRoute* network structure. The user is never aware of this division in submaps—all work is done on the whole structure—but as the modifications are usually confined to a small region, display is almost instantaneous.

**More Details on the Network Structure**

The street network is made up of the following four elements:

- 1) *nodes* locations where a link begins or ends
- 2) *links* connections between two nodes
- 3) *streets* sets of ordered links sharing a common identification
- 4) *link plottings* sequences of intermediate points describing the shape of the link.

As stated earlier, the area covered by a network is divided into rectangular regions, each identified by a unique number. The nodes and links whose locations are in a given region are labeled with the region number. Using a region structure is an efficient way to speed up most of the graphic and search operations, since only the information of the appropriate re-

gion needs to be processed.

The street structure itself is created by two lists:

- 1) the list of links forming a street
- 2) the list of links connected at a given intersection

Therefore, it is possible to follow a street, link by link, from its beginning to its end, and know which connections are possible in the network.

For location-finding and routing purposes the following information is attached to links:

- the allowed directions of vehicles (one-way or two-way streets)
- the type of vehicles allowed on the link
- the prohibited and penalized turns at both extremities of the link
- a mean speed for vehicles

Depending on the application, other information can be assigned to links and easily maintained by the editor.

The data to construct a GeoRoute database can originate from a variety of sources. Programs have been written to extract and process information from Statistics Canada Area Master Files and U.S. Census Tiger Files. A method has also been developed to generate a street structure in situations where only the nodes and links are available.

### Editing the Basic Information

Once the network is created, the network editor is used to modify existing streets, to create new ones and to add information regarding the civic numbers, the postal codes and traffic constraints which include maximum vehicle speed restrictions, one-way streets and prohibited turns. As the network editor displays the street network in graphical form, the user can explore or modify the network using either a keyboard for textual information or the mouse for graphical positioning. Quite often both modes are possible: for example, the location of a client can either be given by its street address, an inter-

section or by pointing with the mouse. Even if the user works with a map image, it should be stressed that the network is always stored in the database format and modifications apply to the database. The map image is recomputed from the database, so there is no risk of inconsistency between the map and the database.

The network modification commands can change the structure of the network as shown in Figure 2 or update the descriptive information associated with the various elements as in Figure 3.

### Advantages of Using a Structure Editor

We can see that the main innova-

tion of GeoRoute is the use and manipulation of the structure of the network. This idea is quite new in the geographical information system (GIS) and operations research community. It can be related to the controversy encountered in computer science for the programming language editors [14] where there are advantages in using either the text mode (characters, lines and pages) or the programming mode (expressions, statements and procedures). Even in a text editor such as Emacs [15], which was primarily directed toward dealing with textual entities, there are modes and commands that manipulate higher-level entities. This choice between the visual and structural can also be

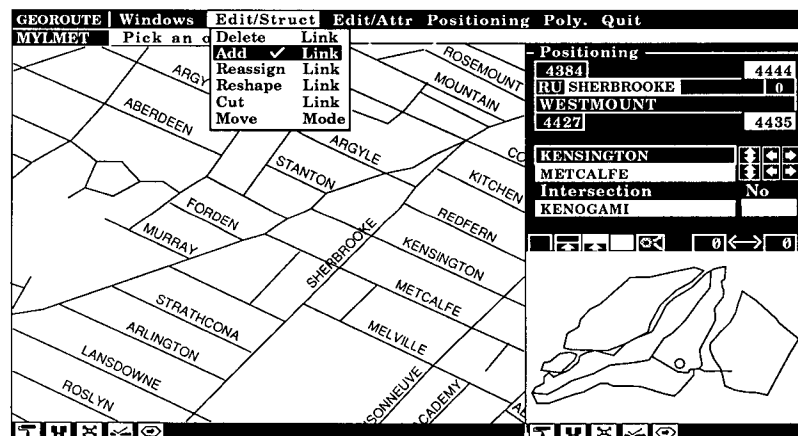


FIGURE 2. Menu for editing the network structure

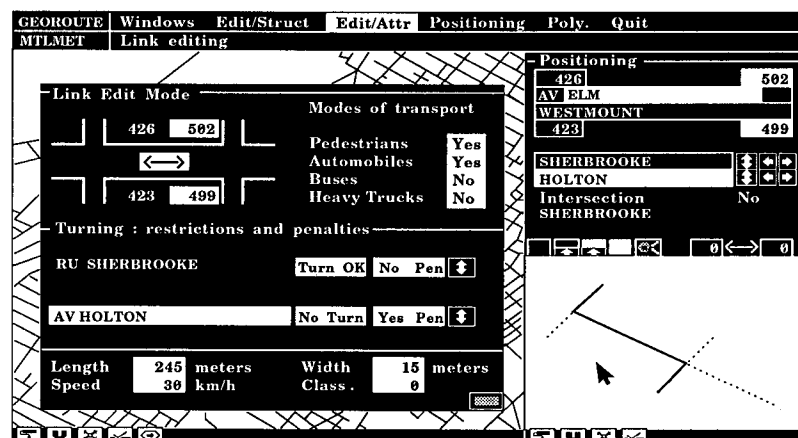


FIGURE 3. Window for editing the link attributes

transposed in our context, but currently the majority of other map editors and GISs use a CAD approach based on drawing maps that are quite difficult to adapt to routing algorithms. For instance, the Intergraph Network Analysis module, SPAN from TYDAC, and few others have optional modules to handle networks. TransCAD from Caliper Corporation has been designed to handle transportation applications. ARCInfo from ESRI also handles networks and produces maps, however the size of the network it can manipulate in the PC environment is quite limited.

In GeoRoute, we were confronted with this trade-off and we designed the system accordingly. Since our group was originally an operations research group, we knew the appropriate structures needed for the algorithms and we developed the street network to fill these requirements. In retrospect, we realize this choice was the right one because it now enables us to build a whole range of applications on this network, such as shortest path calculations, arc- and node-routing optimizations, and address locations.

We want to give a graphic editing capability for this structure, however, this implies that the graphic representation be recomputed at each display. This module has been finely tuned because the program has to run in the MS-DOS environment where memory space is a scarce resource. The time required to display a map is proportional to the number of links to appear in the window. A one kilometer window on downtown Montreal takes about 10 seconds<sup>1</sup> while the complete network takes up to 12 minutes. An update operation takes a few seconds no matter how large the network. We have shown that it is possible to have a graphic editing of a structural network representation, and the next sections describe some of the currently available applications.

<sup>1</sup>Execution times in this article were obtained on a 16MHz PC

### Route Optimization Module

There are two classes of routing problems addressed by GeoRoute: node routing and arc routing. For each, GeoRoute offers a set of general algorithms that can be adapted to specific applications depending on the characteristics of the problem to be solved.

A distinct feature of GeoRoute is the capability to deal with a very detailed street network description transparently divided into smaller regions, as previously described. This implies we had to adapt classical OR algorithms such as shortest paths, or traveling salesman heuristics. As we also deal with permitted turns, one-way streets, U-turns, arriving on the right side of the street, speed limits and user-defined speed/distance functions, the implementation of the algorithms has become quite complex but all of this information is essential in practical environments for route planning. The system thus produces a detailed manifest for the vehicle driver and enables a precise evaluation of the routes.

### Node Routing

In a node-routing problem, the demand is located at the nodes of the network, and solving such a problem is essentially finding a good way of visiting these nodes using arcs of the network. Examples of applications are postal box servicing, school bus stop pick-ups, courier services, or delivery of goods to multiple locations. For each stop, the user can define the demand size, the service time, service time windows (earliest and latest time that the stop may be served) and whether a U-turn is permitted at that stop.

GeoRoute has defined a very general framework in which almost any conceivable algorithm can be cast; this approach was inspired by the Alto [11] system which is still more general because the end user can even design his or her own algorithms.

In *GeoRoute*, route generation is divided into four major steps:

- 1) form initial groups or clusters of stops
- 2) determine an initial sequence of stops within each cluster
- 3) exchange stops within each cluster
- 4) exchange stops between clusters

Steps 3 and 4 can be repeated to improve the final solution. For each step, GeoRoute allows the selection of one or more of several optimization methods. Clusters can either be formed sequentially or by using the Fisher-Jaikumar approach [4]. The initial sequence of stops can be determined using either *nearest insertion* (start with the furthest stop from the depot and then insert the stop that is the closest from any existing stop in the route) or a Clarke and Wright algorithm [2]. Stops can be exchanged using either 1-OPT, 2-OPT, 3-OPT [9] or OR-OPT [10] algorithms. A complete solution strategy can thus be tailored to the particular application at hand.

Solving a node-routing problem implies computing the shortest path between each stop which can be precomputed and stored in a distance matrix. At first sight, this problem is a simple application of a classical OR technique [3, 5], but in some cases (relay box delivery for example) there can be a large number of stops within a small territory, so the characteristics of the network become important [12]. In addition, security constraints can require that the stop be serviced only on the right-hand side of the vehicle and U-turns are generally prohibited. Moreover, the movement of vehicles is limited after servicing a stop located at a street corner (see Figure 4), and this implies that distance from stop  $x$  to stop  $y$  may depend on which stop  $z$  or  $z'$  has been serviced before stop  $x$  as shown in Figure 5.

When solving a problem, most of the time is spent calculating the distance matrix and improving the original solution. The time required for calculating the distance matrix depends on the distance between the clients and the density of

the networks. For a table having 150 entries located in downtown Boston, two hours were required. For a complete distance matrix of Montréal, (i.e., from one link to all other links), 20 minutes by line of the matrix are required. When appropriate, two techniques can be used to reduce calculation time. If the distances are symmetric only the triangular matrix must be calculated. Moreover, in most circumstances, it is sufficient to consider only the nearest neighbors. Improving the original solution is accomplished using stop exchanges. For routes totaling 100 stops in downtown Boston, the OR-opt algorithm takes about two hours while a 3-OPT procedure could run for about 12 hours. Heuristics have been used to discard subsets of exchanges that do not improve the solution. In addition, a time limit can be specified to restrict the duration of the exchange phase. Furthermore, a recently developed exchange algorithm based on Tabu search method should further improve the performance of the method [6]. These times might appear prohibitive, but these calculations can be and are often done during the night and do not slow the solution process. As we will see later, the major part of the time for solving a routing problem is spent in data validation and route evaluation.

### Arc Routing

Arc-routing problems consider the *cost* associated with servicing a street segment. Examples of arc-routing problems include garbage collection, home mail delivery, electric meter reading, street sweeping, street inspection, and snow removal. In mail delivery for example, the cost function is in relation to the number and type of addresses to be serviced on the segment. This information can be estimated or calculated from data stored in external databases.

Arc-routing problems are particularly difficult and they have received less attention in the past

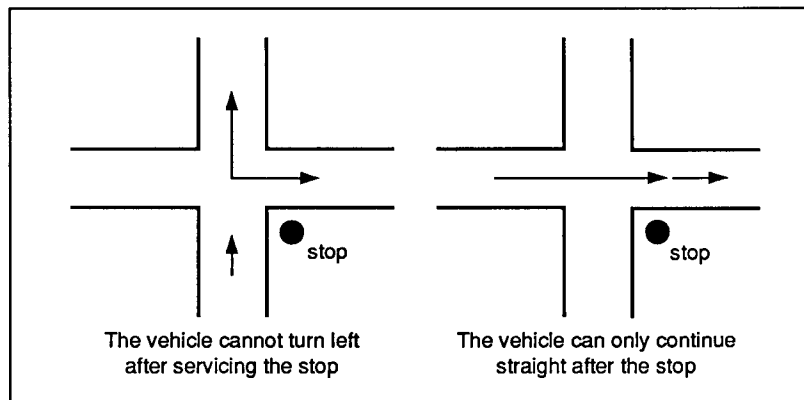
compared with node-routing problems [13]. One reason is the need for a very detailed street network, which is usually a huge data entry task. The network editor we described is a great help in that respect. The main difference between arc routing and node routing, described in the preceding section, is that the algorithms must take into account the topology of the network and cannot only use a pre-computed distance matrix.

As for node routing, GeoRoute has a general framework for arc

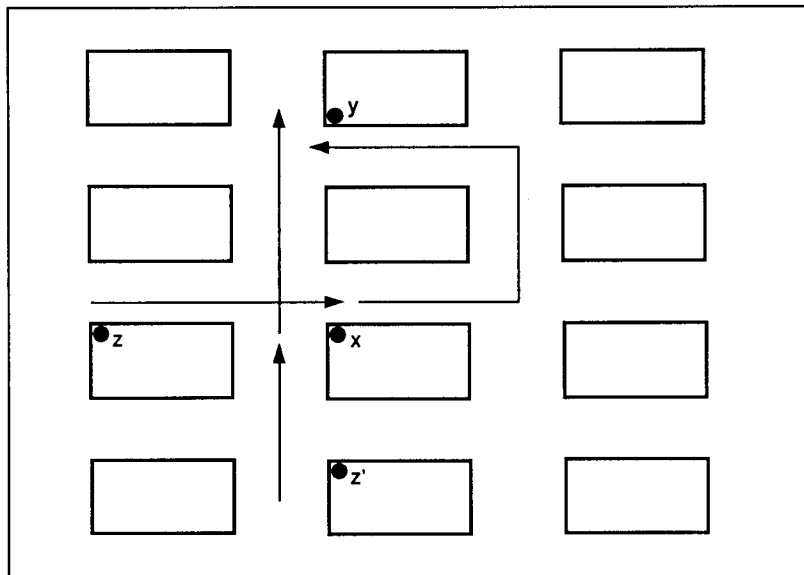
routing inspired by the general heuristic of Alto [11]. It is composed of basic operators which manipulate objects corresponding to subsets of the network. Three types of objects are identified:

- 1) *arc*: a side of a street between two intersections.
- 2) *basic unit*: a group of arcs comprising "the other side of the street" for each arc.
- 3) *cycle*: a connected group of basic units.

The objects have attributes which



**FIGURE 4.**  
Turn options when a stop is located at a street corner



**FIGURE 5.**  
Distance from  $x$  to  $y$  depends on the previous stop in the route ( $z$  or  $z'$ )





appropriately placed and postal codes are added. The lines with the arrows represent letter carrier routes computed by the route optimization module. All of this information is automatically computed from the link representation of the street network.

When the area to be shown on a map has less than 5,000 links, double-line street maps can be produced. Otherwise single-line street maps are computed. The time required to compute a map is proportional to the quantity of information to be plotted. The example shown in Figure 6 takes less than two minutes to compute, but requires five minutes to plot.

A map editor, built within the Intergraph MicroStation environment, is also available. This map editor offers commands specifically designed to improve the aspects of maps because automatically generated maps have some limitations. One is that they show only the information present in the database while, in some instances, it may be useful to add other features to facilitate the understanding of the map. There may be some overlapping between the graphic objects even though the placement algorithm tries to avoid them.

To help with these situations or to simply improve the esthetics of maps, the map editor offers the following capabilities:

- move the graphic components (symbols, text, lines) of the map
- change the size of the graphic components
- create and edit a legend
- add polygons with special fill pattern to represent lakes, parks, etc.
- reshape the curves of the streets
- add new symbols (train stations, bus stops, airports)
- add routing information (e.g., toll bridges, weight limit on bridges)

### Interactive Route Editor

The evaluation of routes computed by the module described in the section on route optimization is most easily done by drawing the routes

on the map using the map generation module described in the previous section. Even with sophisticated algorithms, it is almost certain that the routes generated will not be 100% satisfactory and that a few modifications will be required. Some constraints of the problem might be ignored or neglected or special cases will occur. An experienced person (a manager or dispatcher), however, can easily improve these solutions.

GeoRoute provides an interactive route editor to allow the creation, modification and evaluation of routes. The length and other characteristics of the routes are displayed and automatically computed and updated when modifications are made. Commands are available to create or delete routes and to add, delete or transfer stops within a route. It is also possible to modify the path between two stops.

### Experience with the System and Related Work

GeoRoute is a *spinoff* of many years of research at the Centre de recherche sur les transports. These modules have been applied to solve real-life problems in the domains of postal routes, toll phone collection and school bus routes. Together they form a framework which can be customized by Logiroute to new routing applications.

Our experience so far has shown that a major part of the time for solving a routing problem is spent with the client to analyze the data and the solution. Quite often, the original network and the location of the clients must be adapted and the speed functions to transform distances into travel time must be calibrated before acceptable routes can be produced. Overall, computer solution time, even if it takes a few hours that can be done off-line, is small compared to the data validation time. In this respect, the network editor and the mapping subsystem greatly simplify this task. Using the route editor, a transportation manager can evaluate and modify about 25 routes per day;

doing this task manually would take up to a week.

To our knowledge, there are no other systems that integrate node- and arc-routing algorithms, a network editor and a mapping system that can manipulate large street networks with traffic constraints. Among the most popular routing systems, *TruckStops* from Micro-Analytic and *Roadnet* from Roadnet Technology are relatively low-cost solutions for node problems. They provide the option of using the network, but mainly as a background for the display of the routes. The *Logistic Tool Kit* from CAPS includes modules for routing and scheduling applications, but while it can manipulate networks, it is normally used only to calculate long-haul distances. *RoadShow* from Routing Technology Software has an interesting feature in its use of video image maps as background for route displays. A network can also be geocoded on the background map.

Routing application software could be built into the *Trans CAD* environment (Caliper Corporation). *RouteSmart* from Boone Distinct is such an application for arc-routing problems. This could also be done in *ARCInfo* (ESRI) environment, and with the MGE Network Analysis for Intergraph, but to our knowledge, it has not been done yet.


### Conclusion

GeoRoute is an integration in a microcomputer system of many years of research in the implementation of operations research technology. It has been used in many settings (postal delivery, school transportation and public works management) and is still being improved to expand its application span. GeoRoute can deal with applications that require a detailed representation of a street network and is a *platform* on which specific transportation-related systems can be built. For each application domain, new screens, interfaces to external databases, cost functions



and output reports may need to be defined, but the underlying street database, algorithms and map production facilities remain the same.

#### Acknowledgments

The authors want to thank the Centre de recherches sur les transports of the Université de Montréal for providing a stimulating research environment for so many years. In particular we want to thank Jacques Ferland and Jean-Yves Potvin for many fruitful discussions. We also want to thank Eric. P. Landry and his colleagues from Canada Post for their numerous comments and suggestions during this project. This project is a spin-off of the *Hotocars* project developed with a NSERC strategic grant. It has also benefited from other grants from NSERC, FCAR, and Canada Post Corporation. 

#### References

1. Bodin, L., Golden, B., Assad, A., and Ball, M. Routing and scheduling of vehicles and crews. The state of the art. *Comput. Oper. Res.* 10, 2 (1983), 63–211.
2. Clarke, G. and Wright, J.W. Scheduling of vehicles from a central depot to a number of delivery points. *Oper. Res.* 12 (1964), 568–581.
3. Dijkstra, E.W. A note on two problems in connection with graphs. *Numerische Mathematic 1* (1959), 269–271.
4. Fisher, M.L. and Jaikumar, R. A generalized assignment heuristic for vehicle routing. *Network 11* (1981), 109–124.
5. Floyd, R.W. Algorithm 97: Shortest path. *Commun. ACM* 5 (1962), 345.
6. Gendreau, M., Hertz, A., and Laporte, G. New insertion and post-optimization procedures for the traveling salesman problem. Publication 708, Centre de recherche sur les transports, Université de Montréal, 1990.
7. Kearney, A. Improving productivity in physical distribution. Tech. Rep. CPDM, 1980.
8. Lapalme, G. and Cormier, M. Interactive graphics for routing and scheduling. In *Information Technology Applications in Transport*, P. Bonsall and M. Bell, Eds., VNU Science Press, 1986, pages 235–255.
9. Lin, S. and Kernighan, B. An effective heuristic algorithm for the traveling salesman problem. *Oper. Res.* 21 (1973), 498–516.
10. OR, I. Traveling salesman-type combinatorial problems and their relations to the logistics of blood banking. Ph.D. thesis, Dept of Industrial Engineering and Management Sciences, Northwestern University, 1976.
11. Potvin, J.-Y., Lapalme, G., and Rousseau, J.-M. ALTO: A computer system for the design of vehicle routing algorithms. *Comput. Oper. Res.* 16, 5 (1989), 451–470.
12. Rousseau, J.-M. Customization versus a general-purpose code for routing and scheduling problems: A point of view. CRT, publication #556, 1987.
13. Roy, S. and Rousseau, J.-M. The capacitated Canadian postman problem. *INFOR* 27, 1 (1989), 58–79.
14. Sandewall, E. Programming in an interactive environment: The LISP experience. *ACM Comput. Surv.* 10, 1 (1978), 35–71.
15. Stallman, R.M. EMACS: The extensible customizable, self-documenting display editor. In *Interactive Programming Environments*, D.R. Barstow, H.E. Shrobe, and E. Sandewall, Eds., Chap. 15, McGraw-Hill, 1984, pp. 300–325.

**CR Categories and Subject Descriptors:** H.4.2 [Information Systems Applications]: Types of Systems—Decision support; J.1 [Administrative Data Processing]: Business, education

**Additional Keywords and Phrases:** Geographical information system, mapping, routing, transportation

#### About the authors:

**GUY LAPALME** is professor in the Computer Science and Operations Research Department of the Université de Montréal. His research interests include the application of artificial intelligence techniques to operations research problems, natural language generation and functional programming. **Author's Present Address:** DIRO, Université de Montréal, C.P. 6128, Succ "A," Montréal, Québec, Canada, H3C 3J7, lapalme@iro.umontreal.ca.

**JEAN-MARC ROUSSEAU** is professor in the Computer Science and Operations Research Department of the

Université de Montréal. He is also vice president of GIRO Inc. and president of LOGIROUTE Computer Systems Inc. His research interests include transit operations research, manpower scheduling and routing, scheduling of vehicles, arc-routing and Geographic Information Systems for transportation applications. **Author's Present Address:** Centre de recherche sur les transports, Université de Montréal, C.P. 6128, Succ "A," Montréal, Québec, Canada, H3C 3J7.

**SUZANNE CHAPLEAU** is system analyst at LOGIROUTE Computer Systems Inc. Her interests include object-oriented programming, graphic interfaces and shortest paths algorithms.

**MICHEL CORMIER** is project manager at LOGIROUTE Computer Systems Inc. His interests include graphic interfaces for routing applications, automatic mapping and school transportation.

**PIERRE COSSETTE** is project manager at LOGIROUTE Computer Systems Inc. His interests include automatic mapping, arc routing and postal applications.

**SERGE ROY** is director of LOGIROUTE Computer Systems Inc. His interests include routing, scheduling and geographic information systems for transportation applications.

**Authors' Present Address:** Logiroute Computer Systems Inc., 75, rue de Port-Royal est, Bureau 500, Montréal, Québec, Canada, H3L 3T1.

*GeoRoute* is a registered trademark of Giro.

*MS-DOS* is a registered trademark of Microsoft Corp.

*VAX/VMS* is a registered trademark of Digital Equipment Corp.

*Intergraph MicroStation* is a registered trademark of Intergraph.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

© ACM 0002-0782/92/0100-080 \$1.50