SUPPORTING VISUAL INTERACTIVE LOCATIONAL ANALYSIS USING MULTIPLE ABSTRACTED TOPOLOGICAL STRUCTURES

Paul J. Densham
National Center for Geographic Information and Analysis
Department of Geography
State University of New York at Buffalo
Buffalo, New York 14261-0023
densham@geog.buffalo.edu

and

Marc P. Armstrong
Departments of Geography and Computer Science
The University of Iowa
316 Jessup Hall
Iowa City, Iowa 52242
armstrng@umaxc.weeg.uiowa.edu

ABSTRACT

Data structures for automated cartography traditionally have been based on either vector or tessellation models. We describe a set of topological abstractions, derived from a vector approach to the representation of geographic space, that were developed to support the interactive solution of location-selection problems. When augmented with an appropriate representation of geometry, these abstractions are used to generate cartographic displays that support interactive decision-making. The advantages of this approach include: the use of the same data abstractions for analysis and display purposes; support for multiple representations of networks and, therefore, a degree of scale independence; and, finally, support for highly interactive problem-solving and decision-making because map generation can be decomposed into parallel processes.

1.0 INTRODUCTION

Decision-makers faced with ill-structured social and environmental spatial problems increasingly are adopting spatial decision support systems (SDSS) to help them find solutions. When decision-makers try to resolve such problems, they can work within four spaces (Densham and Goodchild, 1990).

- o *Objective space* contains all feasible options available for a given problem.
- o *Decision space* is bounded by the decision-maker's knowledge, objectives and values. It contains those options considered feasible by the decision-maker.
- o *Model space* is determined, for a given model, by its variables and their values, its parameter values, and the relationships represented in the model.
- o *Geographic space* provides a context for a decision and is depicted using cartographic and other graphical representations. Because of its richness and complexity, geographical space is abstracted and represented in objective, decision and model spaces.

For any given problem, SDSS users normally wish to use the system to help them explore their decision space. Because an individual's decision space typically is some subset of objective space, the challenge facing SDSS designers is to empower their users and to expand their decision space to include much of the objective space. One approach to this challenge is to provide users with modelling capabilities and complementary representations of geographic space; in concert, these representations enable users to evaluate solutions from models using their expertise, objectives and values by synthesizing mathematical and cartographic representations of a problem.

Despite the emphasis on exploring decision spaces in the SDSS literature, many current systems support a process of decision-making in which graphical representations play a very restricted role (Densham, 1993a). This process often follows more traditional modes of GIS use, in which a system's capabilities are sequenced in a linear, "work-flow" approach that results in a final graphical product. To explore a decision space, however, decision-makers require a series of capabilities, consisting of both analytical and graphical representations, that can be combined in flexible sequences. Moreover, decision-makers increasingly wish to interact with a system through any of the representations of a problem that it provides. For example, in our applied work with decision-makers we often display a map of a locational configuration and listen as they articulate questions. In most cases, such questions would be answered if the user could manipulate the graphic directly - by dragging a facility to a new location, by adding or removing a facility, or making some other change - and the system would respond by invoking the appropriate analytical capability. While such linkages are very similar to those found in the direct manipulation interfaces of word processors and object-based drawing packages, they typically are missing from geoprocessing software. Consequently, we listen as decision-makers articulate their questions and move their fingers across the screen to illustrate the kinds of actions they would like to make; we then try to translate these questions into an appropriate sequence of analytical operations.

In this paper, we show how multiple topological structures can be used to represent the analytical and cartographic components of a range of location-selection problems. These structures support bi-directional linkages between the analytical and cartographic representations of problems. As such, they provide a foundation for building a SDSS that supports a visual and interactive approach to decision-making; one that synthesizes the goal-seeking nature of traditional analytical modelling with the intuitive "what-if" approach supported by GIS.

2.0 REPRESENTATIONS

Historically, there have been conceptual and practical issues that have conditioned approaches to the representation of geographical relationships. These issues have been addressed in different ways by cartographers and spatial analysts. One dimension on which to measure such differences is the level of abstraction used in representing space, spatial relationships and thematic information. Cartographers have attempted to provide map users with appropriate, and often detailed, geometrical representations of space. Spatial analysts, on the other hand, have tended to employ highly abstracted representations of spatial relationships in their models. In extreme cases, only an abstracted form of topology is maintained and geometry is discarded completely - the origin-destination matrix is a prime example. While these abstracted relationships often are derived from detailed cartographic information, the extreme parsimony of the derived information has tended to preclude effective display. As a consequence of these differences, separate data structures have been developed for graphical and analytical purposes.

Cartographic data structures have benefitted from many years of development in the fields of automated cartography and GIS (e.g. Peucker and Chrisman, 1975). In contrast, analytical data structures reflect the "black-box" nature of modelling tools: analytical structures are often *ad hoc* and code-specific with little or no attention paid to placing them in a broader representational context. This results, in part, from the "stand-alone" nature of many custom-written modelling tools and the often scant attention paid to visualization by spatial analysts. Furthermore, the use of commercial modelling and analysis packages (including SAS, SPSS and GAUSS) often forces spatial analysts to remove the geometry from their problems to make them fit the available representations and structures. Paradoxically, it is the very domain-independence of their representations and structures that makes these packages feasible commercially. In contrast, cartographic data structures have evolved to reflect and accommodate the standard set of spatial primitives around which most GIS are built. Thus, points, lines, areas, and raster cells are well-defined, and their representations in GIS that support both operations and display have been refined over twenty years. Despite these refinements, recent changes in software technology have caused researchers to view these structures as collections of objects. Unfortunately, spatial analysts have paid little attention to the identification and definition of the spatial primitives and objects which underlie their algorithms and models. Consequently, analytical objects normally are not the same as display objects in either their structure, content or function. This variation has, in part, led to an asymmetry: the display of cartographic objects must reflect the current status of analytical objects (e.g. which ones are facilities) but analytical objects typically carry little or no geometrical information and often only abstracted topological information that can be used to link them to cartographic objects.

3.0 VISUAL INTERACTIVE LOCATIONAL ANALYSIS

Decision-makers increasingly work in microcomputer environments for which a plethora of word-processors, spreadsheets, drawing and charting packages, and database management systems have been developed to support multiple forms of user interaction. Such software increasingly links graphics to other forms of representation and provides mechanisms for their direct manipulation, including drag-and-drop editing. For example, if a user changes the cost of a raw material in a spreadsheet cell, a graphic that depicts forecasted profits is updated automatically. The power of complementary analytical and graphical representations that have been designed to enhance a decision-maker's understanding of a problem has been recognized in many disciplines. Operations researchers, for example, have developed graphical methods for both formulating and interacting with analytical models. These visual interactive modelling (VIM, Hurrion, 1986) tools typically support bi-directional linkages among representations. Thus, using a spreadsheet analogy, users could either change the value of a cell and see the graphic updated, or they could manipulate the graphic and see the cells in the spreadsheet updated. In a GIS, maps are used to support spatial query. This form of linkage, however, normally is found only between the database and graphics capabilities of a GIS and bi-directional linkages typically are not a part of systems that couple GIS and spatial models. This is particularly true of systems coupling locational analysis capabilities with those of GIS (Densham, 1993a).

A key component of a VIM environment for locational analysis is the user interface. Ideally, this interface supports representations of all four spaces in which decision-makers work: objective, decision, model and geographic. Although a GIS might represent geographic space using a variety of abstractions, the general absence of modelling capabilities from these systems means that they lack representations designed explicitly for objective space, decision space and model space. Furthermore, an interface must support both goal-seeking and "what-if" modes of system use, blending analytical and intuitive approaches to problem-solving. To understand better the issues and problems that must be addressed in designing an interface to represent these spaces, we have designed a model interface (Armstrong, Densham and Lolonis, 1991) for use in redistricting applications. This interface supports bi-directional linkages among graphical displays and analytical capabilities, and direct manipulation of its linked representations.

The interface consists of three windows that display complementary maps, graphs and tables. Used together, these representations provide substantial amounts of information about the four spaces in which decision-makers work. For example, the map window may display current facility locations, and the allocations of demand to them, while the graph window contains a histogram of facility workloads and the table window lists facility attributes. If a decision-maker wishes to investigate the effects of adding a new facility, two approaches can be used. The first (goal-seeking) approach, is to select a command from a menu that identifies the optimal location for the facility and updates all three windows appropriately. The second (intuitive) approach enables the user to specify the location for the facility by pointing to a candidate location. In this latter case, the map window first must be updated to show existing facilities and all the candidate facility locations that have been identified; the user selects one of these locations and all three windows are then updated to show facility locations and allocations of demand, facility workloads, and facility attributes for the new configuration. In both approaches, the user can confirm the addition of the new facility or reject it, returning the windows to their original state.

Direct manipulation of the interface's representations is used in other contexts. For example, if decision-makers want to investigate the effects of relocating a facility, they may simply click on a facility symbol and drag it to a new location. The allocations of demand are recalculated, redisplayed, and the other windows also are updated to reflect the requested change. In another context, the capacity of a facility may be changed by dragging its bar in the workload histogram to a new level; again, the system automatically recalculates the changes and displays them.

Despite its utility, this interface has shortcomings. An extra window is required to support direct interaction with the underlying analytical operations: the user could set model parameters using either a dialogue box or by manipulating the contents of a graphical display. This window also could be used to stop a model during its solution to change its parameters or to force a particular candidate to be a facility site. This process would be facilitated by animating the solution process, providing a pedagogic tool that would enable a user to see the current best configuration and its attributes.

4.0 INTEGRATING CARTOGRAPHY AND LOCATIONAL ANALYSIS

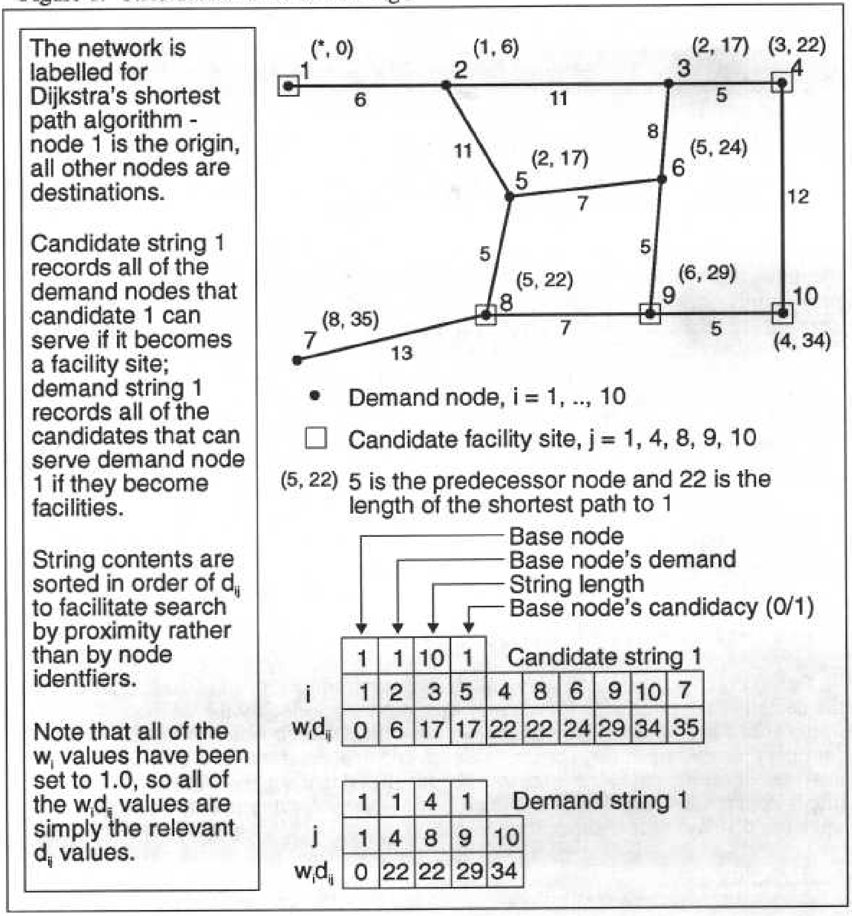
Designers of a visual interactive modelling environment must address issues beyond the design of the interface. First, every operation supported by the interface has implications for the database management system (DBMS) which must provide data for query, analysis and display and must be able to accommodate the results of analyses. Second, data structures must be identified that support both analytical and display operations. Finally, to facilitate human-computer interaction, mechanisms for updating displays in real-time are required.

4.1 Data Structures

Because analytical and display objects often are different in structure, content and function, a central issue in coupling GIS with analytical software is the resolution of object differences between them (Nyerges, 1992). One approach to solving this problem is to employ formal methods for modelling entities and their relationships. Armstrong and Densham (1990) developed Entity-Category-Relationship (ECR) diagrams of two different "user" views of a SDSS database: a cartographic perspective and a locational analysis view. The integration of these two views at a conceptual level required that object differences be resolved. The resulting conceptual design was used with database management software to develop a logical schema and, ultimately, to implement a database.

Several specially-designed relationships and record types are used in the implemented database to minimize search and retrieval to support both analysis and display. To indicate which points in the database are also nodes on the network used for locational analysis, a record called "L_A_Nodes" is used. This record obviates searching all of the points in the database, many of which are not part of the network, to retrieve the network's geometry and topology. To increase the utility of these data after retrieval, relationships are used to indicate the linkages among nodes on the network, sorted in order of network node (L_A_Nodes) identifiers. Sorting these items in the DBMS means that they do not have to be sorted after every retrieval and before they can be processed using a shortest path algorithm. Furthermore, these relationships

Figure 1. Candidate and demand strings



also facilitate the identification and retrieval of chains depicting network links for display purposes.

The data structures used to support analytical and display operations are critical in a VIM environment. These data structures must facilitate both goal-seeking and "what-if" forms of analysis, including the generation and evaluation of configurations of facilities and their associated demand sets. Unless these structures also support display, there is a strong possibility that versioning problems will arise - at any point in time, the mathematical representation of the problem is different from the cartographic one. To support the interface described above, and VIM for locational analysis more generally, we are employing data structures developed to support interactive locational analysis.

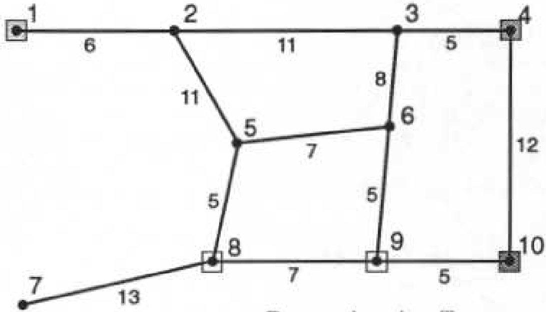
Two types of data structures have been used to develop a series of analytical operations that can be sequenced to implement heuristic location-allocation algorithms. The first structure, the allocation table (Densham and Rushton, 1992a), is a spatial accounting mechanism; it records the closest and second-closest facilities to every demand location. The second structure, the distance string, comes in two forms: candidate strings (Hillsman, 1980) and demand strings (Goodchild and Noronha, 1983). Both types of distance string store information about feasible interactions

Figure 2. The allocation table

Locate facilities at candidates 1, 4 and 10.

The allocation table records the closest and second-closest facility (j) to each demand node (i). The table is built from candidate strings, demand strings, or both.

The objective function value is the sum of the $w_i d_{ij}$ values in the second row of the table.



Demand nodes (i)

		1	2	3	4	5	6	7	8	9	10
Closest facility	j	1	1	4	4	1	10	10	10	10	10
	$w_i d_{ij}$	0	6	5	0	17	10	25	12	5	0
Second-closest facility	j	4	4	10	10	10	4	1	1	4	4
	$w_i d_{ij}$	22	16	17	12	17	13	35	22	17	12

among demand locations and candidate facility locations. Although every demand node has a demand string, only candidate nodes (potential facility locations) have candidate strings.

Figure 1 depicts a small, ten-node network with five candidate locations (nodes 1, 4, 8, 9 and 10) and the associated candidate and demand strings for node 1. The format of the strings is the same: a header row with four fields and two rows with fields containing node identifiers and weighted distances ($w_i d_{ij}$) respectively. The w_i term represents the weight or demand at node i; the "distance" between demand node i and candidate j is represented by d_{ij}. (Here, d_{ij} will be used to denote physical distance but it can be used to represent travel cost, travel time, and other measures of spatial separation.) A demand string records the weighted distances from the demand (base) node to all of the candidates that may serve it if they become facility sites. A candidate string stores the weighted distances to all of the demand nodes that the candidate (base node) may serve if it becomes a facility site. In both types of strings, entries are sorted in ascending order of d_{ij}, the measure of spatial separation. This ordering facilitates search and retrieval by proximity, rather than by node identifiers. Although a complete set of candidate and demand strings contains the same information, they are organized to optimize different forms of data retrieval. Demand strings are used to answer the question "What is the closest facility to this demand node?" In contrast, candidate strings answer the question "Which demand nodes can this candidate serve if it becomes a facility?"

An allocation table consists of six rows and n columns, where n is the number of demand nodes on the network. For each demand node, the first two rows of the table store, respectively, the node identifier of, and the associated weighted distance to, the closest facility; similarly, the third and fourth rows store the identifier of, and the weighted distance to, the second-closest facility. The fifth and sixth rows of the table are used to evaluate alternative locational configurations that are different from those represented in rows one to four. Figure 2 depicts the first four rows of the allocation table for the network introduced in Figure 1 with facilities sited at candidates 1, 4 and 10. An allocation table can be built in two ways: first, by searching along every demand string to find the identifiers of the first two candidates which are facility sites;

and, second, by comparing the weighted distance for each demand node in the facilities' candidate strings.

In concert, an allocation table and a set of candidate and demand strings provide a flexible representation of a locational problem. Densham and Rushton (1992a, 1992b) show how three heuristic location-allocation algorithms can be implemented using five operations defined upon distance strings and the allocation table. A microcomputer-based software package implementing these three algorithms has been developed around these data structures and operations (Densham, 1992). By defining a few additional operations, these data structures also support a further four algorithms (Densham, 1993b).

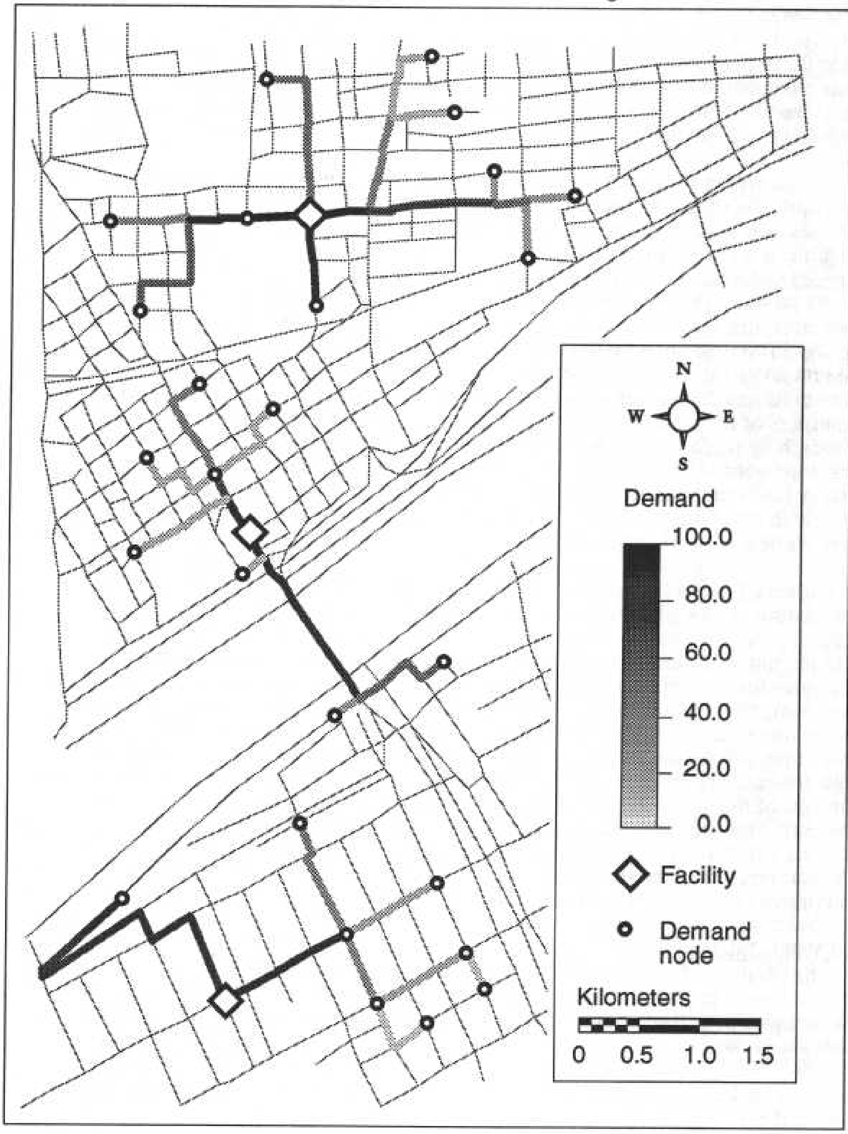
4.2 Generating Spider Maps

We have developed a taxonomy of cartographic displays that can be used to depict the results of different queries and analyses to decision-makers (Armstrong *et al.*, 1992). Each of these maps is suited to answering different questions that decision-makers articluate as they investigate location-selection problems. A "spider map," for example, is used to show the allocation of a dispersed set of demand to a smaller set of facilities; it can be used to depict an existing locational configuration or to show the results of analyses, both goal-seeking and intuitive. A canonical spider map links each demand node to its allocated facility with a straight line. The display of such a map requires geometrical, topological and thematic information: thematic information is used to differentiate among facility locations and demand node locations; geometrical information is used to locate both facilities and demand nodes in space; and, finally, topological information is used to link the demand nodes to the facilities. While distance strings and the allocation table can supply some of this information, the remainder must come from other sources.

The geometry of a network normally is discarded as a shortest path algorithm generates distance strings. This occurs because strings record only those allocations of demand nodes to candidates that are feasible using one or more criteria. Although the spatial separation of demand nodes and candidates is captured in their $w_i d_{ij}$ values, d_{ij} may represent various metrics of spatial separation and the resulting proximity values may bear little relationship to network geometry. Thus, only an abstracted form of the network's topology is retained: pairs of candidates and demand nodes that potentially can fill the roles of service provider and client. Although the allocation table is built from information stored in the strings, it contains two other pieces of information required to generate spider maps. First, the allocation table contains thematic information about which candidates are facility sites and, second, topological information that describes the allocation of demand nodes to these sites. The missing element is the geometrical data - the locations in space of each demand node and facility.

A network's geometry and full topology are required by the shortest path algorithm to build distance strings. In both the PLACE (Goodchild and Noronha, 1983) and LADSS (Densham, 1992) packages, these data are stored in two files: the links file contains the topology, while the geometry of the nodes and their thematic information are stored in the nodes file. The links file contains one record for every node on the network. Each record stores the node's identifier; its valency (the number of network links for which it is a node); the identifiers of the nodes at the other end of its links (sorted in ascending order of identifier); and the corresponding link lengths. The nodes file is a table that contains one, six-field record for every node in the links file. The fields in each record store a node's: identifier; region identifier (a polygon representing a census tract, for example); weight or demand; candidacy (1 if it is a candidate, 0 otherwise); X coordinate; and Y coordinate. The contents of these two files can be retrieved from the database described above. Moreover, because of the organization of these data in the DBMS, they do not have to be sorted before they can be used by the shortest path algorithm (Armstrong and Densham, 1991). Thus, the geometrical data required to generate a vector spider map can be retrieved directly from the database, from the nodes file on disk, or from a data structure in RAM.

Figure 3. A spider map showing allocations and flows along network links



In many situations, a spider map that depicts the actual paths taken through the network enables a decision-maker to understand better the consequences of selecting a particular locational configuration. Figure 3 shows a network-based spider map that depicts the aggregate flows of demand along each link (McKinney, 1991). Producing such a map requires all of the information used for a vector spider map, plus a list of the links that form the paths between each demand node and its closest facility. Although a shortest path algorithm (for example, Dijkstra, 1968) identifies these paths as it builds the distance strings, all the paths underlying a string are discarded once it has been built. Consider, for example, a network of 100 nodes arranged as a ten-by-ten lattice. If all of the nodes have positive weights and also are candidates, then each candidate and demand string has 100 entries. The paths underlying a complete set of candidate and demand strings for this network traverse 66,000 links, all of which would have to be stored. A 100 node network is tiny when compared with the digital network data sets available from various sources, including TIGER files, and storing paths for such networks would require large amounts of storage. Furthermore, many of these paths will never be required. If we consider locating one facility at a corner of the 100 node lattice, only 100 of the 10,000 paths will be required to generate the spider map - traversing only 900 links.

An alternative to storing the paths is to reconstruct them when they are required. This option is attractive because the amount of computation required to build the paths for a network-based spider map normally is very much less than that required to construct a set of candidate and demand strings. Building a set of candidate and demand strings requires multiple paths to be generated per demand node, one for every candidate site on the network to which the node feasibly can be assigned. The spider map, in contrast, requires the generation of only one path per demand node, depicting the route from the node to its associated facility. The allocation table provides a list of the nodes that are allocated to each facility. To generate the paths linking each demand node to its associated facility, a shortest path algorithm can be used. Because the identifiers of the facilities and of their assigned demand nodes are known, the amount of searching performed by the shortest path algorithm is reduced. This is achieved by searching outward from a facility site only until every demand node allocated to the facility has been reached. Thus, supplementing the distance strings and the allocation table with the contents of the nodes and links files provides all the information required to generate a network-based spider map.

For very large networks, the computation required to build even one candidate or demand string can be considerable. Consequently, we have investigated the use of parallel processing to increase computational throughput and, thereby, decrease the solution times of locational analysis algorithms. We have developed a series of strategies for decomposing spatial algorithms into parallel processes (Armstrong and Densham, 1992; Ding, 1993). These strategies have been used to develop parallel processing versions of Dijkstra's algorithm (Ding *et al.*, 1992; Ding, 1993). Parallel processing can be used to reduce the time required to generate the paths depicted in a network-based spider map. Each facility, its associated list of demand nodes, and the contents of the nodes and links files can be passed to a separate processor. After generating the required paths, each processor can then aggregate the demand flows over each link, returning a list of links traversed and the associated volume of demand. The cartographic chain representing each link can then be retrieved from the database and drawn in an appropriate hue (McKinney, 1991) to depict the demand flow.

4.3 Visual Interactive Modelling

The addition of a new facility to an existing configuration changes the allocations of at least some demand nodes to facilities, requiring the update of a spider map. Using the methods described above, the following course of action occurs if a decision-maker adds a new facility:

1) The decision-maker clicks on a candidate facility site to locate a new facility there.

2) The SDSS responds by adding the new facility to the allocation table and updating its contents.

3) The first row of the allocation table and the contents of the nodes and links files are used to generate the paths through the network that will be depicted in a network-based spider map.

4) The paths, their associated lists of links and demand flows, are used to query the database and to redraw the spider map.

5) The contents of the graph and table windows also are updated.

By applying some knowledge about the spatial structure of a location-allocation model to step 3, it is possible to determine which nodes will be reallocated when the new facility is added (Densham and Rushton, 1992a). This knowledge can be used, first, to generate paths only for these reallocated nodes and, second, to recalculate the aggregate demand flows over affected links, further reducing the amount of computation required to generate the spider map.

A similar process is used when the decision-maker either relocates a facility by dragging it across the screen, or removes a facility from the configuration. The only changes occur in step 3 because a different set of operations are applied to the allocation table and distance strings. These operations determine the effects of the relocation or facility removal on the allocations of demand nodes to facilities and the value of the objective function.

5.0 CONCLUSIONS

We have shown how abstracted topological data structures, used for locational analysis, can be supplemented with geometrical and topological information to produce cartographic displays. Several advantages are realized when this approach to map generation is adopted. First, the same data abstractions are used for analysis and display purposes, obviating versioning problems. Second, the data abstractions can be implemented as objects with both analytical and display methods. Third, a degree of scale independence results because this approach supports multiple representations of networks. Fourth, this approach is suitable for highly interactive problem-solving and decision-making because many of the components of spider maps, and other maps used in locational decision-making, can be generated independently and can be decomposed into parallel processes. Finally, in a pedagogic context, the solution processes to various algorithms can be animated to depict different spatial search strategies.

ACKNOWLEDGEMENTS

Partial support for this paper from the National Science Foundation (SES-90-24278) is acknowledged. This paper is a contribution to Initiative 6 (Spatial Decision Support Systems) of the National Center for Geographic Information and Analysis which is supported by the National Science Foundation (SES-88-10917).

REFERENCES

Armstrong, M.P. and P.J. Densham 1990, Database organization strategies for spatial decision support systems: *International Journal of Geographical Information Systems*, Vol. 4, pp. 3-20.

Armstrong, M.P. and P.J. Densham 1992, Domain decomposition for parallel processing of spatial problems: *Computers, Environment, and Urban Systems*, Vol. 16(6), pp. 497-513.

Armstrong, M.P., P.J. Densham and P. Lolonis 1991, Cartographic visualization and user interfaces in spatial decision support systems: *Proceedings*, GIS/LIS '91, pp. 321-330.

Armstrong, M.P., P.J. Densham, P. Lolonis and G. Rushton 1992, Cartographic displays to support locational decision-making: *Cartography and Geographic Information Systems*, Vol. 19(3), pp. 154-164.

Densham, P.J. 1992, *The Locational Analysis Decision Support System (LADSS)*, NCGIA Software Series S-92-3, NCGIA, Santa Barbara.

Densham, P.J. 1993a, Integrating GIS and spatial modelling: visual interactive modelling and location selection: *Geographical Systems*, Vol. 1(1), in press.

Densham, P.J. 1993b, Modelbase management for heuristic location-allocation algorithms: manuscript available from the author.

Densham, P.J. and M.F. Goodchild 1990, *Spatial Decision Support Systems: Scientific Report for the Specialist Meeting*, NCGIA Technical Report 90-5, NCGIA, Santa Barbara.

Densham, P.J. and G. Rushton 1992a, Strategies for solving large location-allocation problems by heuristic methods: *Environment and Planning A*, Vol. 24, pp. 289-304.

Densham, P.J. and G. Rushton 1992b, A more efficient heuristic for solving large p-median problems: *Papers in Regional Science*, Vol. 71(3), pp. 307-329.

Dijkstra, E. 1959, A note on two problems in connection with graphs: *Numerishe Mathematik*, Vol. 1, pp. 101-118.

Ding, Y. 1993, *Strategies for Parallel Spatial Modelling Using MIMD Approaches*, Unpublished Ph.D. Thesis, Department of Geography, State University of New York at Buffalo, Buffalo.

Ding, Y., P.J. Densham and M.P. Armstrong 1992, Parallel processing for Network analysis: decomposing shortest path algorithms on MIMD computers: *Proceedings*, 5th International Spatial Data Handling Symposium, Vol. 2, pp. 682-691.

Goodchild, M.F. and V. Noronha 1983, *Location-Allocation for Small Computers*, Monograph No. 8, Department of Geography, The University of Iowa, Iowa City.

Hillsman, E.L. 1980, *Heuristic Solutions to Location-Allocation Problems: A Users' Guide to ALLOC IV, V, and VI*, Monograph No. 7, Department of Geography, The University of Iowa, Iowa City.

Hurrion, R.D. 1986, Visual interactive modelling: *European Journal of Operational Research*, Vol. 23, pp. 281-287.

McKinney, J.V. 1991, *Cartographic Representation of Solutions to Location-Allocation Models*, Unpublished Master's Project, Department of Geography, State University of New York at Buffalo, Buffalo.

Nyerges, T. 1992, Coupling GIS and spatial analytic models: *Proceedings*, 5th International Spatial Data Handling Symposium, Vol. 2, pp. 534-543.

Peucker, T.K. and N.R. Chrisman 1975, Cartographic data structures: *The American Cartographer*, Vol. 2(1), pp. 55-89.