Structures Over Graph for Handling Population Data

Malathy Naguleswaran
University of Canterbury
Christchurch, New Zealand
Phone: +64 3 364-2987 Fax: +64 3 364-2569
Email: malathy@cosc.canterbury.ac.nz

Presented at the 10th Colloquium of the Spatial Information Research Centre, University of Otago, New Zealand, 16-19 November, 1998

Abstract

We suggest that population data could be modelled using graphs whose edges have a line geometry. We introduce the concept of distributed graph. Population data can be modelled using this where population is considered to be evenly distributed over its edges. Three distinct structures over graphs useful for modelling population data are presented. Firstly a modifiable graph aggregation scheme for viewing a graph with details hidden is introduced. Secondly a novel way to represent a new graph, resulting from the addition of new line geometries, as an extension of the old graph is suggested. Thirdly the concept of graph partition based on distributed graph is developed to represent service zones.

Keywords and phrases: urban population data; graph aggregation; subgraph.

1 Introduction

Networks are a dominant theme in the urban geographic space. A Network is the basic structure for all facility management such as electricity, telephone, gas and water. Population is dispersed over the street network and movement of population occurs over the transportation networks. These networks are represented as space graphs within data models.

The assumed starting point for representing space graphs is a set of nodes N and a set of edges E between pairs of nodes. Each node in N has a unique Euclidean coordinate pair and each edge in E has a line geometry. The two end nodes of an edge must be distinct. The line geometry of the edges must not intersect. In other words it is a planar undirected graph.

We observe that population data can also be gainfully represented using space graphs, the approach that was started by the US census bureau Tiger files (Peuquet, 1990). Classical models for managing space graphs treats nodes and edges as indivisible units. The classical model while adequate for facility management applications, such as electricity networks and water supply networks, places restrictions on modelling population data. The difference derives mainly from considerations of privacy. Facility management graphs deliver services to demand centres whose domain related characteristics are not subjected to any privacy law. For example facility management databases may maintain the location and characteristics of each of their demand locations. In contrast it is not permissible under privacy laws to maintain population characteristics to such levels of details. The main source of population characteristics, census data, is available only in terms of census blocks. Our analysis for modelling population data starts with the assumption that population data can be made available as distributed data along the graph edges. Thus the need to treat edges of a population graph as continuous rather than discrete components of the graph. In this paper we propose the following three structures over space graphs to facilitate modelling of population data.

Aggregation of a graph: A population graph can be aggregated to form a graph of smaller size for efficiency in urban modelling. The aggregated parts of this graph will be subgraphs of the base graph and will also aggregate the associated population data.

Extension of a space graph: We also suggest a scheme for representing a new space graph resulting from the addition of new line geometries that intersect the line geometries of the edges of an existing space graph. This feature can be used to study new residential developments during planning.

Partition of space graph: When the population distribution is represented as graph, service zones over the graph can be maintained as a collection of subgraphs of the base graph. The zoning for different services are in fact collections of subgraphs where each such collection covers the entire base graph. Queries that relate to different zoning can therefore be answered by subgraph matching, which can be more efficient than polygon overlays.

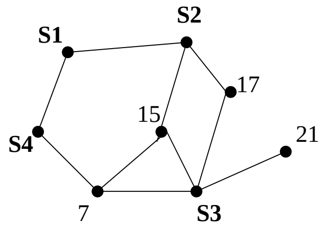
We use a sample planar graph, shown in Figure 1 (a), for illustrating the representation schemes. This same graph is represented as an edge collection in the first two columns of Table 1 in Figure 1. For example the two entries ‘1’ and ‘2’ in the first row represents the edge ‘1-2’ in the graph and the two entries ‘2’ and ‘5’ in the second row represents the edge ‘2-5’ in the graph and so on. Figures 2 and 3 also contains this same graph representation.

Table 1

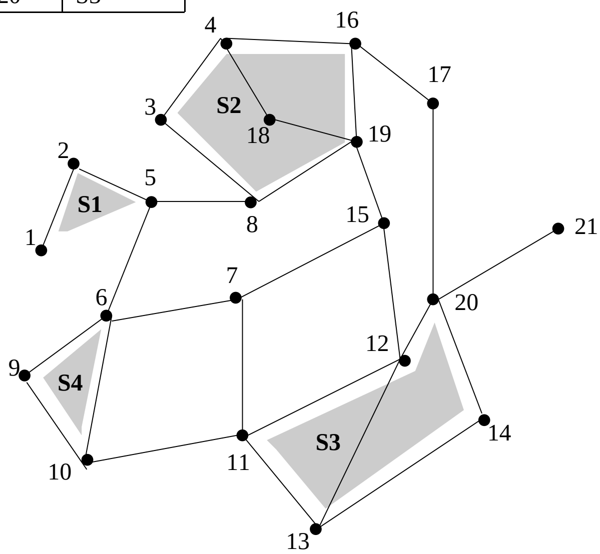
Base Graph		Aggregation
Node	Node	
1	2	aggregated
2	5	aggregated
3	4	aggregated
3	8	aggregated
4	16	aggregated
4	18	aggregated
5	6	–
5	8	–
6	7	–
6	9	aggregated
6	10	aggregated
7	11	–
7	15	–
8	19	aggregated
9	10	aggregated
10	11	–
11	12	aggregated
11	13	aggregated
12	13	aggregated
12	15	–
12	20	aggregated
13	14	aggregated
14	20	aggregated
15	19	–
16	17	–
16	19	aggregated
17	20	–
18	19	aggregated
20	21	–

Table 2

Subsiduary Graph	
Node	Node
5	S1
6	S4
8	S2
10	S4
11	S3
12	S3
16	S2
19	S2
20	S3



(b) Aggregated Graph



(a) Graph

Figure 1. Graph Aggregation

2 Representing Subgraphs of an Explicit Graph

Defining subsets of edges of a graph is central to most useful structures over graphs. The three structures described in this paper also need to represent subsets of graph edges. The most appropriate data structure for explicit graphs, over which subsets are to be defined, is an edge collection. The structures over graph described here are all based on the assumption that the base graph is represented as an edge collection. We will refer to a subset of the edge collection as a *subgraph* of the base graph.

One approach to representing subgraphs of a graph maintained as a collection of edges is to create a new collection of edge identifiers that belong to the subgraph. This will add an extra level of indirection to access the base graph from the subgraph. Subgraphs maintained as a new collection of edges rather than just edge identifiers will avoid the extra indirection by duplicating the edge data. Another approach to maintaining subgraphs is suggested, based on three considerations in relation to population graphs.

Firstly space graphs used in representing population data are not subject to frequent modifications. This suggests it is possible to assume an ordered and static edge collection. The position index of an edge in this ordered edge collection can be used as a local identifier for the edge. Fixed length edge attributes can then be recorded in a parallel collection that mirrors this static ordered edge collection. Accessing the attribute of an edge recorded in such a parallel edge collection will be efficient since its location can be calculated from the local identifier (position index). We define this parallel collection as an *Edge map* of a graph that is maintained as a static ordered edge collection. Edge maps permit the recording and accessing of fixed length attributes of the edges in the graph, using the position index of the edge in the edge collection.

A subgraph could be maintained using an edge map by recording a flag for each edge in the subgraph. Using an edge map is not space efficient for recording a single subgraph, since most of the slots in the edge map will not be utilised. The second consideration is that the three structures we propose need to maintain sets of non overlapping subgraphs. This means that a single edge map could be used to hold a set of subgraphs making it space efficient.

Thirdly the subgraphs we need to maintain are *connected subgraphs*. A connected subgraph is a subgraph where all nodes of the subgraph must be reachable from any other subgraph node. (eg: Subgraph S1, S2, S3 and S4 in Figure 1). Connected subgraphs maintained using an edge map can be retrieved by graph traversal over the base graph, starting from a single node that belongs to this connected subgraph and testing against the edge map for inclusion of each edge that is traversed. This avoids the need to scan the whole edge map to retrieve single connected subgraphs. This method of accessing connected subgraphs makes this scheme time efficient.

The three proposed structures use edge maps to represent subgraphs.

3 Graph Aggregation

Graph aggregation is useful for viewing and manipulating population graphs at a larger scale, at which details are hidden. Such an aggregated view should also aggregate the data associated with the graph edges. The aggregation must be modifiable at run time. Algorithms over the aggregated graph could then be run in less time than the same algorithm over the base graph.

The scheme we suggest is based on *connected subgraphs* defined above and the *periphery nodes of connected subgraphs* . Periphery nodes are those nodes of a connected subgraph whose incident edges may not be part of that subgraph. (eg: Nodes 11, 12 and 20 are periphery nodes of subgraph S3 in Figure 1 (a))

The aggregation scheme treats a selected set of non overlapping connected subgraphs as nodes. The shaded subgraphs of the graph in Figure 1 (a)

Table 1		
Base Graph		Extension
Node	Node	
1	2	Remove
2	5	–
3	4	–
3	8	–
4	16	–
4	18	–
5	6	Remove
5	8	–
6	7	–
6	9	–
6	10	–
7	11	–
7	15	–
8	19	–
9	10	–
10	11	–
11	12	–
11	13	–
12	13	–
12	15	–
12	20	–
13	14	–
14	20	–
15	19	–
16	17	–
16	19	–
17	20	–
18	19	–
20	21	–

Table 2	
Extension Graph	
Node	Node
1	101-N
2	101-N
5	102-N
6	102-N
100-N	101-N
101-N	102-N
102-N	103-N

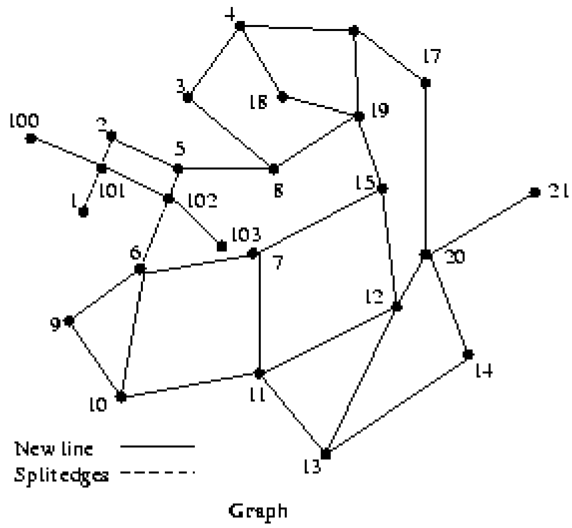


Figure 2. Graph Extension

represents such aggregated nodes. Figure 1 (b) shows a schematic view of the resulting aggregated graph.

An edge map and a subsidiary graph is used to represent an **aggregate graph**. The “Aggregate” column of Table 1 in Figure 1 is the edge map representing the aggregated graph. Table 2 of Figure 1 is the subsidiary graph needed to complete this aggregate graph representation. The subsidiary graph consists of a set of pseudo edges between the periphery nodes of each connected subgraph forming an aggregated node and the aggregated node itself. For example the first two rows of the subsidiary graph, entries 5-S1 and 6-S4 are the pseudo edges from the aggregated node S1 to its periphery nodes 5 and 6.

One way of viewing this subsidiary graph is to consider it as the encapsulation of the effort needed to traverse connected subgraphs. That is, some space has been traded off for time efficiency and complexity reduction. Graph traversal on the aggregated graph is achieved by using the base graph in conjunction with the aggregate edge map and the subsidiary graph. Graph traversal could start from a non aggregated node of the base graph or from an aggregated node. For example a traversal with start node of 7 will access edges 7-6, 7-15, 7-11. At the next step from node 6, edges 6-5, 6-10 will be accessed. At each access to an edge the aggregate edge map is used to check if this edge is consumed by an aggregated node. Edge 6-10 is consumed by the aggregated node S4. Therefore this edge will not be processed. Instead the subsidiary graph will be used to access the periphery nodes of S4 (6 and 10). The outgoing edges from these periphery nodes that are not consumed by the aggregated node will now be returned for further traversal.

Graph aggregation can be extended to many levels resulting in a hierarchy of aggregated graphs. This can be achieved by adding entries in the edge map corresponding to each level. An additional subsidiary graph needs to created for each aggregation level.

4 Graph Extension

Given a graph whose edges have a line geometry it will be useful to create a new graph resulting from the addition of new line geometries to the existing graph. To represent such a graph we first define a subgraph over the base graph. This subgraph excludes all edges that intersect with the new lines. Consider the addition of the new line 100-103 (Figure 2) to the graph, resulting in a graph extension. The “Extension” column of Table 1 in Figure 2 is an edge map that represents the subgraph of the base graph that excludes the intersecting edges. An extension to the graph is created that consists of the new edges resulting from the intersection of the new lines and existing edge geometries. The geometry of the new edges in the extension are defined as segments of either the edges of the base graph or the new line geometries that were added. The complete extended graph consists of the base graph without those edges that are marked in the edge map as “removed” and the extension graph.

Most of the slots in an extension edge map will be unused. However when an extended graph is aggregated or partitioned (next section) the same edge map can be used to record the extension edge map and the aggregate edge map or the partition edge map.

5 Graph Partition

By graph partition we mean partitioning the set of edges of a graph into non overlapping sets. This graph partition concept is different from a simple partition of a set of objects. In a graph partition the set of edges in each partition forms a connected subgraph. Therefore as described earlier, an edge map can used to record and access the connected subgraphs forming the graph partition(Figure 3). Graph partition can be used to record districts of the urban space such as fire service districts and school districts .

Table 1		
Node	Node	Partition
1	2	S1
2	5	S1
3	4	S2
3	8	S2
4	16	S2
4	18	S2
5	6	S1
5	8	S1
6	7	S3
6	9	S3
6	10	S3
7	11	S3
7	15	S5
8	19	S2
9	10	S3
10	11	S3
11	12	S4
11	13	S4
12	13	S4
12	15	S5
12	20	S5
13	14	S4
14	20	S4
15	19	S5
16	17	S5
16	19	S2
17	20	S5
18	19	S2
20	21	S5

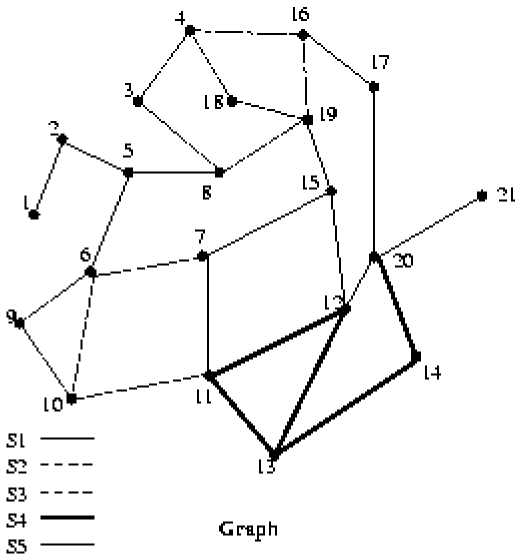


Figure 3: Graph Partition

6 Discussion

In terms of the entity relationship model a database instance itself may be viewed as a graph where all entities are considered as nodes and all relationships are considered as edges. The nodes and edges of a database instance graph are not homogeneous. A cyclic relationship defined over a single entity is a graph with homogeneous nodes and edges. These explicit graphs may be singled out for special implementation within a database management system for efficient handling of graph operations. GraphDB (Guting, 1994) is such a data model. The space graph for representing population data described in this paper is also an explicit graph with a set of structures defined over it. It too can be integrated into a database with some limitations on updates to the base graph.

The suggested structures can be compared with structures over space graphs available in existing spatial data models. The Arc-Info network module (ESRI, 1992), Smallworld geometry model (Smallworld, 1996) and TransCAD networks(TransCAD, 1999) are three systems that provide some functionality for handling space graphs. Some basic structures over space graph provided in these existing spatial data models can be identified. They are,

Routes: Defining routes that are an ordered and connected set of edges and accessing points along these routes as distances along the routes.

Subgraph: Defining subgraphs of the basic graph which in turn could be treated as a graph.

Edge Split: Splitting an edge into two edges and a new node without affecting the underlying graph.

We have described graph aggregation, graph extension and graph partition as structures over an explicit graph and suggested that population data could be modelled using these structures. The existing data models of space graph do not provide the functionality for aggregation or extension. Some do provide the graph partition functionality. This paper has addressed the data structure aspects of the proposals. The proposals have the following advantages,

- The edge map technique used allows easy overlap querying of all the structures suggested.
- Modifiable graph aggregation schemes are useful in urban modelling, as well as for viewing a graph with less detail.
- Graph extension is a novel way to represent new graphs resulting from the addition of new lines.

References

D J Peuquet. (1990) Introductory Readings in Geographic Information Systems. Taylor & Francis, .1990.

B Carre. (1979) Graphs and Networks. Clarendon Press, Oxford, 1979.

R H Guting. (1994) GraphDB: Modelling and Querying Graphs in Databases. 20th International Conference on Very Large Databases, Santiago, 1994.

ESRI. (1992) Arc/Info Network Module User Manual, ESRI International.

Smallworld. (1996) Smallworld GIS 2 User Manual, Smallworld Systems Inc.

TransCAD. (1996) TransCAD Transportation GIS Software Users Guide, Caliper Corporation.

M Erwig. (1992) Graph Algorithm = Iteration + Data Structures? Graph Theoretic Concepts in Computer Science, LNCS #657, pp 277-292. Springer-Verlag, 1992.

M Mainguenaud. (1995) Modelling the Network Component of Geographical Information Systems.International Journal of Geographical Information Systems, 9(6):575-593, 1995.