USE OF GRAPH THEORY TO SUPPORT GENERALISATION

William Mackaness and Kate Beard
National Center for Geographic Information and Analysis,
348 Boardman Hall,
University of Maine,
ME 04469, USA

Abstract
This paper discusses the utilization of graph theory to aid in the subtle application of specific generalization techniques during map design. The visualization of space encapsulates the notion of context, the representation of the interdependence of salient variables and even that of aesthetics; many of the subtleties of the cartographic hand rely on a rich understanding of those relationships. It is argued that any equivalent automated system needs to have the same rich knowledge explicitly or implicitly stored with each feature. The consequences for database design from this perspective are very different from ones that require efficiency or cater to spatial query.

Generalization techniques are not applied wholesale to map objects. Their application varies both within each class of objects (such as all roads) and between different classes of objects (linear roads are not treated in the same way as linear rivers). The authors consider that the utilization of graph theory affords a means of capturing the context under which generalization techniques are selected and applied: the graph can be weighted to reflect the varying emphasis in the relationships between objects and sets of objects, as well as help select objects according to their relevance, context and proximity to salient features. Conventionally graph theory relies on being able to identify links between discrete objects. The paper looks at these problems and identifies some of the functional requirements of the database.

1.0 Introduction
Graph theory has been applied to many many problems in addition to the obvious ones of traveling salespersons, circuit board design and minimum spanning networks; they include minimum vertex coloring, modelling markov chains, enumeration of chemical molecules, network flows and scheduling or critical path analysis. Their application to Geography is not new either, and many are familiar with the problem of the bridges of Konigsberg.

This paper looks at the application of graph theory from the perspective of map generalization and generalization of connectivity. The objectives governing the 'successful' application of generalisation techniques include ordering, distinction, comparison, combination, and recognition of relations. To know if these objectives are being met demands that we first know what these relationships are between features. There are many design decisions in map design and was argued by Mackaness (1991), a very rich knowledge of the data is required for successful design, least of which is knowledge of the role played by each feature:- does the feature provide a contextual setting, and does it connect, divide, or include other objects? Is it unique or isolated, or is it representative of other features? All these factors (and many more) govern a features likely inclusion or exclusion from a map. Furthermore the role of that feature may change according to different map themes and scales. All this preamble is to emphasize that the application of graph theory in map generalisation is
part of a wider utilization of other theories that will help to formalize all the knowledge required to make intelligent and aesthetic decisions in map design.

Webster (1990) demonstrates how spatial relationships between entities, which have been defined in co-ordinate geometry, may be formulated as a rule base.

It is easy to see generalisation as a set of operators (simplify, omit, etc.) whose application has been governed by controls (such as scale, aesthetics, content, theme), but generalisation is not the mere analysis of space but the communication of spatial concepts. This paper discusses the application of graph theory in the context of network generalization; sometimes the term linear generalization is used interchangeably. In the context of this paper, linear generalization should not be confused with line simplification (reducing the number of points used to represent a line); line simplification has little to do with map design, only with the efficient storage of linear features.

2.0 A Brief Review of Graph Theory
For an excellent introduction into graph theory and all that it affords, the reader is referred to Hartsfield and Ringel (1990) and Wilson (1979). This paper touches on a small part of graph theory looking particularly at finite weighted and directional graphs. It begins by describing some of the terms used in graph theory and some of their more basic properties. The section following looks at some rules for generalization derived from the application of graph theory to map generalization.

A graph is made of vertices (A, B, C, D, E in Figure 1), and edges (AB, AD, CD, BC, DE, CE, BD, AC). The degree of a vertex is the number of edges that have that vertex as an endpoint (for example vertex D is of degree 4, having edges BD, AD, CD, and DE). The degrees for each vertex are shown in Figure 1a.

Note too that, 1b is isomorphic to 1a; though the positions and drawings of the vertices and edges are different, the vertices have the same degree and the same vertices incident to one another (having only the same degree sequence does not guarantee isomorphism). Thus a graph can appear visually different from the network from which it was derived; for example in a graph derived from a road network, the crossings of edges do not necessarily represent overpasses!

Figure 2 looks at other attributes of graphs (in which not all the vertices degrees and edges are marked). Figure 2c shows an example of a loop (note the degree number for that vertex is 5 not 4), Figure 2a shows an isolated vertex (which has degree 0), and Figure 2b shows a bipartite graph. In a bipartite graph the vertex set can be divided into two disjoint sets, V1 and V2, in such a way that every edge joins a vertex of V1 with V2. It is not necessary that every vertex of V1 be joined to V2, though in this case it does. Thus 2b is an example of a complete bipartite graph.
A graph is said to be complete if, for any simple graph (this excludes directional graphs – such as Figure 7, and graphs containing loops – such as in Figure 2c), every pair of distinct vertices are adjacent. We can calculate the number of edges of a complete graph if we know the number of vertices making up the simple graph. If \( n \) is the total number of vertices, then the number of edges is exactly \( \frac{1}{2} n (n-1) \). Such an index can help describe the level of connectivity in a graph. Consider Figure 2b (which is not a complete graph although it is a complete bipartite graph). Figure 2b has 12 edges, and if it were complete, it would have 21 (according to the formula above). Thus it is 12/21 or 57% complete. The percentage of completeness is also shown for Figure 2a and 2c (the loop in Figure 2c is ignored). Though the interpretation of these values is a little subjective, it is an indicator as to the degree of 'redundant' connectivity in a network. This issue is revisited later in the paper when we use Kruskal's algorithm to derive minimum spanning trees in finite weighted graphs (see section 4.1).

A ‘disconnected set’ of a connected graph \( G \), is a set of edges whose removal disconnects \( G \). In Figure 2 above, the disconnected set of 2c is e6 and e7. A ‘separating set’ is similar to a disconnected set, but applies to vertices; if the vertices of a separating set are removed, the graph \( G \) becomes disconnected. If a separating set contains only one vertex, it is called the cut-vertex or articulation vertex. In figure 2a, \( v^1 \) is an example of an articulation vertex.

Finally, we can state that if a connected graph contains a cycle – that it is possible to visit the same vertex from different adjacent vertices) then removing an edge from the cycle will not disconnect the graph. For each of the attributes of the graph described in this section, it is possible to envision equivalent structures in natural and ethnographic networks. In this next section we will see how we can apply some of these attributes to generalization in the form of rules.

### 3.0 Some Preliminary Rules

Even at this stage we can begin to describe the attributes of features in such a way that we can formulate rules that can help us in the generalisation process. These attributes of descriptors together with examples of rules are set out in Table 1. Some of the example rules are compound and would require additional analysis through the application of theories other than just graph theory.

- ‘A feature with degree 0 is an isolated feature.’
Often features have prominence by virtue of their isolation, or because they are important to navigation (for example where both ethnographic networks and features are sparse). Thus remote wooden mountain huts are often included in maps. Thus we might have the following rule:

**IF a feature is isolated and ethnographic and one-of-a-kind and of-a-size comparable-to-its-symbol-size and map-type-is-topographic THEN**

that feature should be included in the map.

- ‘**Where a cycle exists, edges can be removed on a hierarchical basis without isolating a feature**’

An example where such a rule is applied is given in Figure 8.

- ‘**A feature with degree 1 is an "end" feature and has the qualities of a terminus**’

**IF a feature demarcates the end of a network and map-type-is-topographic or map-type-is-network THEN**

that feature should be included in the map.

- ‘**A feature with proportionately high degree compared with the mean degree of the graph, has hub like qualities**’

**IF a feature plays a key role in providing connectivity between other features and the map-type-is-network THEN**

that feature should be included in the map.

- ‘**A feature that is the sole link between other features is more important than features not serving this task but of a similar type**’

**IF a feature is the only member of a ‘disconnecting set’ THEN it should not be omitted if features of the same type and classification are removed.**

......this can be generalized further....

Any feature which is the sole member of a disconnecting set should be retained in a network type map.

- ‘**An articulate feature displays hub like qualities and additionally provides unique connectivity between features**’

**IF a feature is articulate and the map-type-is-network THEN it should not be omitted if features of the same type and classification are removed.**

Table 1 Some attributes derived from application of graph theory

4.0 Weighted Graphs and Mimicing the Generalization of Networks
If we now look at weighted graphs, we are able to utilize a number of theorems to help generate minimum trees that pay heed to the rules identified in table 1.

The weights assigned to each edge of a graph could be indicative of distance between features, cost, travel time or traffic loading. Indeed each edge might have a number of values for each of these attributes. In Figure 3a, below of a stream network, the numbers have been assigned on the basis of Strahler’s stream ordering principle (Strahler 1960). Such an application has immediate and obvious use in network generalization as well as in symbolization (pen thickness according to stream order number illustrated in both 3a and 3b). Figure 3b shows the same network but with only stream order 3 and 4 shown. Numerous researchers have highlighted problems such as the one illustrated in 3b whereby the lakes have become ‘detached’ from the network. Most human cartographers would extend the line to catch the lakes. Problems such as these are the very ones that graph theory can provide a solution for. In this instance, the river joining the lakes, and river connecting them to the main river would comprise the disconnected set of this graph.

Figure 4 shows a weighted (non-directional) finite graph theoretic representation of an urban area at 1: 50 000 (example is taken from Keates 1989, p46). The graph has a similar orientation and shape to help the reader interpret which roads got what weight. The weights given to both the edges and vertices (indicated in the key) enable the subsequent generalization of the network at 1: 100 000 and 1: 250 000. Some vertices have been added at the periphery merely to make the graph finite. The graph was derived by viewing the map at 1: 250 000 and working up to 1:50 000. The weighting is required for both vertices and edges in order to derive both the partial separating set and disconnected set (described in Figure 2).
Figure 4 Graph theoretic representation 'reverse engineered' from an example by Keates (1989, p 46).

It is interesting to look at the degree of completeness for the three maps in Figure 4. Table 2 shows the more generalized maps to be more complete.

<table>
<thead>
<tr>
<th>Scale</th>
<th>Vertices</th>
<th>Actual Edge No.</th>
<th>1/2n(n-1) Edges</th>
<th>% Complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: 50 000</td>
<td>100</td>
<td>139</td>
<td>4950</td>
<td>3 %</td>
</tr>
<tr>
<td>1: 100 000</td>
<td>15</td>
<td>16</td>
<td>105</td>
<td>16 %</td>
</tr>
<tr>
<td>1: 250 000</td>
<td>8</td>
<td>7</td>
<td>28</td>
<td>25 %</td>
</tr>
</tbody>
</table>

Table 2 Degree of Completeness for the three maps in Figure 4

**Applying Kruskal's Algorithm to Weighted Graphs to aid the Generalization of Networks**

Given a graph with known weight information, we can use Kruskal's algorithm (also known as the greedy algorithm), to find the minimum weight spanning tree in a connected graph (for a description of this algorithm see Aho et al 1974). Figure 5b shows the result of applying Kruskal's algorithm to 5a, and figure 6 is a subjective placement of the vertices of figure 5, and assumes that the weight is indicative of time taken.

![Figure 5a](image1)

![Figure 5b](image2)

Complete 43% 25%

Figure 5: 5b is the minimum weight spanning tree of the connected graph of 5a, taken from the example given by Hartsfield and Ringel (1990, p128).

Note that we have also calculated the degree of completeness in both cases; in a situation where the percentage difference was great, we might assume from a cartographic perspective that we had lost too much of the context of connectivity and revisit the graph, adding edges of lower weights until the percentage difference fell within some pre-defined tolerance.

Figure 6 Fictional Networks in a Fictional Land

**Directional Graphs**
We can extend the graph to include directional graphs (an example of which is given in Figure 7).

Figure 7 Example of a directional graph

Knowledge concerning directional information can be used to help generalize. For example Figure 7 shows part of a ring road system in a town center. We know that if a connected graph contains a cycle, removing an edge from the cycle will not disconnect the graph. Thus by identifying single degree vertices as well as the direction of vertices it is possible to safely generalize to single disconnected sets without loosing the notion of overall connectivity.

Figure 7 Three states of Generalization of a road network
Note that in the third example, the 'amalgamated' line has found a new location. Though not dealt with here, the development of such an algorithm would need to preserve the topology in such a generalization!

5.0 Use of Graph Theory to Select Between Classes of Objects

A graph can also be of mixed edge types whereby each edge can convey a different type of connectivity. In Figure 8, A and B are two possible outcomes from an attempt to reduce the map content of the Original. Figure 8b is a simplified graph of the map in which nodes represent objects on the map and links (arc lines) represent the relationship between map objects. In B, all tributaries that are not first order have been systematically removed (in a similar manner to Figure 3), and we have lost the notion that Keaton is connected to the sea by a river. In this particular context, A is the desired solution but requires that the database has information on the important role played by certain sections of the road and rivers in indicating the relationships between the towns and the sea (in this case that the river provides the sole connectivity between the town and the sea).

Figure 8 Context and emphasis directly impacts the solution: A is preferred to B.

Where there is a choice, weighting the links according to theme can be used to select an appropriate outcome. Thus the I6 road is selected in preference to the tributary of the river Li, to show the context of Beatin in relation to the sea. The highlighted arcs of figure 7b, represent those links which are explicitly shown in solution A.

In example in figure 8 we can see a pedestrian trail (part of the Appalachian trail), a road vehicle network and a flight network for float planes for part of the eastern seaboard of the USA.

Figure 9 Roads, Trails and Planes for part of the Eastern Seaboard

And Figure 10 shows the graph for figure 9. By using a similar line type as in the key of figure 9, it is possible to determine the type of edge between the various vertices.
Figure 10 Graph of part of the Eastern Seaboard

From figure 10 we can envision different applications for this representation such as touring and sightseeing using different modes of transport. It is also possible to determine the likely functionality of vertices; Figure 10 shows only those vertices and edges in close proximity to the activities of walking and local light aircraft activity. Such a graph could then be used to derive 'corridor maps' (or strip maps) showing local towns and amenities.

Figure 11 A graph of vertices in proximity to 'edge activities' used as a basis for corridor maps.

Figure 11 shows the result of using this graph to derive a corridor map. Though this is a simple example, a more sophisticated version of a corridor map could be produced by weighting the graph. For example it is likely that 'second order' towns might wish to be shown, or large towns that provide a context (such as Bangor) may also wish to be shown. The first of these problems is resolved algorithmically and the second problem can be solved by weighting.
Figure 12 A map derived from a finite, unweighted graph of vertices in proximity to specified edges.

But the application of graph theory suffers from the same inadequacies of any theory that effectively treats objects as discrete. For example the Appalachian trail does not run through the middle of Monson and is about half a mile from the Seaplane base in Millinocket. Whilst at the map scales of figures 10 and 12, the graphs of Figure 9 and 11 are correct, there are scales for which the graphs would no longer apply. This leads us to ponder on one of the most debated questions in generalization: what is the most detailed level at which we should/can apply graph theory and are there graph-generalization 'cusps' similar to the ones proposed by Muller (1989)? This question is looked at more closely in a later section of this paper.

6.0 Connectivity Influencing the Likelihood of Inclusion in a Map

The context of a map is an important component of recognition process and provides a setting for the salient message that the designer wishes to convey. Contextual information may be an inset, a section of coast or a capital. Part of that context is the representation of the connectivity that exists between features and this information can be indicative of its remoteness or centrality. Figure 13 attempts to show how knowledge of a features connectivity to other features can govern the inclusion of other features.
Kuujjuaq is a remote town of Labrador, with an approximate population of 500. It has no road connection, but is connected to the sea by a bay, and can be reached overland by driving to Sept-Îles, taking the train to Schefferville and canoeing the river Caniaspiscau between the months of June to October. The trip takes a minimum of 20 days from the put–in site. The graph in figure 12 shows this. Immediately we can identify edges of the disconnecting set (such as the railway and the river). Additionally we can identify members of the separation set (such as Schefferville). Thus Schefferville is an articulation vertex. Following the rules identified in table 1, we can state that whilst normally railways and rivers of first order might not be included in maps of certain scales and themes, that figure 13a represents the minimum requisite information necessary to convey Kuujjuaq's association or links with the rest of the world. Thus graph theory can be used to secure the connectivity context of features when generalization techniques are being applied.

7.0 Graph Resolution and the Use of Fuzzy Logic
Conventionally graph theory relies on being able to identify links between discrete objects (in the example above the sea and the town). But at different levels of detail the behavior of objects changes and they no longer become discrete. A town might be defined as a set of 'pooled' buildings and we would expect some of those buildings to provide certain functions (churches, city hall etc.).
And we must also accommodate the inter-links that occur at varying levels of detail; that a house is ‘connected’ to a road, or that continents are connected by airways.

But we also need to accommodate the non discrete nature of data

Fuzzy set theory (Zadeh 1965) helps us to avoid the constraining world of discrete space, though as Wang et al. 1990 acknowledge, it does not solve the problem of highly commingled attributes. Finer classification or higher resolution in combination with fuzzy set theory appears to be the solution, though analysis costs rise alarmingly.

8.0 Conclusion

This paper has shown how graph theory can be applied to help with a number of design problems; in the first instance, weighted graphs and/or directional graphs can be used to hierarchically omit or delete classes of features. The examples given in this paper included tributary removal based on Strahler's stream ordering, and road classification enabled the simplification of urban road networks.

Whilst different cities, towns and villages will have different mean degrees and completeness, graph theory can be used to prevent the road network of a town falling below a level at which the town becomes disconnected. Whilst this would have the effect of de-emphasizing the difference in urban street complexity (and therefore by inference its size), the authors consider that it is more important to represent the connectivity between features than their difference in size (especially as there are other ways of showing size difference through use of symbols and text).

Whilst its obvious application is in identifying remote features as well as maintaining connectivity between features, it can also be used to help select features in general proximity to other features. Thus it is possible to create 'strip' maps. For example towns and historical sites of interest along the Appalachian trail, or making a map showing views from a train, on a 'train' map.

Not only can it be used to identify remote features, but also key links between features, such as a minor road that crosses over a pass, linking two remote mountain villages.

For reasons of discussion, the examples have centered around finite graphs, but the authors acknowledge that the use of fuzzy sets and fuzzy graphs are better able to represent

As Goodchild notes there are a multiplicity of possible conceptual data models for spatial data and “the choice between alternative models constrains the functions available” (Goodchild 1992 p36). For design, what is required is a database that explicitly encompasses spatial data from the design perspective rather than being seen as the concatenation of layers; layers which are made up of points, lines and areas and which are themselves abstractions of reality. The interpretation of a feature is itself dependent on a context provided by other features. Thus a features inclusion or exclusion from a map graphic is not solely based on theme and scale, but also its interaction with other features, and the competition for space. The application of graph theory enables us to formalize some of the relationships that exist between features and thus gain a better insight into how features interact and in turn help us to make better decisions in map design.

9.0 References


Gardiner V 1982 'Stream Networks and Digital Cartography' *Cartographica* 19(2):38-44.


Sedgewick R 1984 *Algorithms* Addison Wesley


Wang F et al 1990

Wilson R J 1979 *Introduction to Graph Theory*  London: Longman.


Zadeh 1965


**Bibliography**
