

An analysis of the performance of a hierarchical wayfinding computational model using synthetic graphs

A Car^{a,*}, G. Taylor^b, C. Brunsdon^c

^a*School of Geoinformation, Carinthia Tech Institute, Richard-Wagner-Strasse 19, A-9500 Villach, Austria*

^b*Department of Computer Science, University of Glamorgan, UK*

^c*Department of Geography, University of Newcastle-upon-Tyne, UK*

Accepted 17 November 2000

Abstract

This paper describes a set of experiments, which use different levels of hierarchical shortest path computations. We investigate a graph–subgraph structural hierarchy as a mechanism imposed on an input data set, allowing a human or computer to access only a subset of the data necessary for a task like path retrieval. It challenges the *selection of relevant data* further used by people and, in turn, by computers for a particular analytical purpose. Consideration of the main principles for the design of such hierarchies raises a number of theoretical and practical research questions related to spatial information. The paper introduces the idea of adapting the principles of hierarchical wayfinding to modeling decision making, which is becoming increasingly important for advanced applications like Geographic Information Systems (GIS). The results of the experiment, which utilizes three kinds of synthetic graphs, are described. A number of important conclusions are presented, not leastwise that the benefits of hierarchical wayfinding over non-hierarchical wayfinding algorithms increases as the number of nodes in a graph increases, particularly in graphs with recognizable form. © 2001 Elsevier Science Ltd. All rights reserved.

Keywords: Hierarchical wayfinding; Hierarchical spatial reasoning; Graph–subgraph; Structural hierarchy; Abstraction; Hierarchy

* Corresponding author. Tel.: +43-4242-2004-140; fax: +43-4242-2004-179.

E-mail address: car@cti.ac.at (A. Car).

1. Introduction

The shortest path problem arises in various disciplines, and therefore, has received a lot of attention in the literature (see Ahuja, Magnanti, & Orlin, 1993) for a review and analysis). Many of the developed algorithms have been used for querying spatial graph databases in advanced applications such as geographic information systems (GIS) and time-critical Intelligent Transportation Systems (ITS; Collier & Weiland, 1994). Although proven useful, these algorithms are often inefficient due to the growing size of graphs. This is where the need for modifying the existing strategies or developing new ones to improve performance of these algorithms emerged.

In this paper we discuss the use of hierarchy to overcome this problem. In path finding and navigation, hierarchy has been used for partitioning the underlying network into smaller networks. This speeds up path computation, but the resulting paths are not always guaranteed to be optimal (Shapiro, Waxman, & Nir, 1992). Our approach considers hierarchy as a method humans use for organising spatial information and retrieving its relevant bits in spatial tasks like wayfinding (see for example Allen, 1999; Golledge, 1995; Hirtle & Hudson, 1991; Hirtle & Jonides, 1985; Tversky, 1992). To design a conceptual model for hierarchical wayfinding, we used the hierarchical spatial reasoning method to combine human knowledge and experience in wayfinding with standard shortest path algorithms. Detailed discussion of the theoretical background, the conceptual model, and the formal specification of the model available as program code written in the functional programming language Gofer can be found in Car (1997).

We have been developing a computational model for hierarchical wayfinding using the formal specification of the concept as a guideline. The model has two main components: a hierarchical graph contains a set of subgraphs, each of which is formed according to some criteria like road classes, travel speed, scenery, or a combination thereof; a hierarchical wayfinding algorithm consists of a general-purpose shortest path algorithm (in our case the Dijkstra algorithm) and a set of reasoning rules for path retrieval in the hierarchical graph. The basic idea of the hierarchical wayfinding is the *stepwise reduction* of the original graph causing the general-purpose shortest path algorithm to operate on a subgraph. This idea follows the assumption that hierarchical wayfinding occurs in a subgraph with higher-speed/time segments. Hence, hierarchy is used to reduce the graph size and so significantly decreases the path retrieval time, but at possible loss of optimality.

In this paper we argue that the principles of hierarchical wayfinding can be adopted in modelling decision making. Consider the following example: solving a set of shortest paths using a full network would represent a person with perfect or complete information and decision making capabilities. Such a person can be a taxi driver or a local citizen living in that area for a long period of time. Systematic reduction of the original graph would model less and less perfect information and, in turn, decision making. A level containing the least detail, i.e. the smallest sub-network, can represent knowledge of a person visiting a town for the first time and knowing nothing about it. Such an approach allows for reasoning to be simpler at levels with fewer details (Fotheringham, 1992). The idea discussed in this paper is to use different levels

of hierarchical shortest path computations to approximate decision making and potentially missing information. The major questions asked here are:

1. How does processing on such a graph structure affect the calculated path, in particular for different types of networks, e.g. random, radial, Manhattan?
2. Which conditions lead to suboptimal wayfinding?

This is where the currently developed computational model of hierarchical wayfinding will be used. A series of experiments is conducted on different kinds of synthetic graphs to observe the model's behaviour, and preliminary results will be presented. We expect the test results to allow for deriving hierarchization parameters for spatial networks. The hierarchization parameters appear to be highly important for deriving at least rough default behaviour of the hierarchical wayfinding algorithm. Real networks may have many special cases, and therefore make the derivation of general case behaviour difficult to achieve. This is why in the first instance we decided to test the hierarchical algorithm on synthetic graphs only. Consideration of the main principles for the design of such hierarchies raises a number of theoretical and practical research questions related to spatial information in general, and spatial hierarchical reasoning in particular. Our long-term goal is to develop a tool for simulation of hierarchical reasoning in networks.

2. Related work

Using hierarchies of abstraction for reducing the computational cost of particular operations is an old idea and common to many disciplines and application areas: graph theory and spatial databases (Agrawal & Jagadish, 1994; Erwig, 1994), spatial reasoning (Papadias & Egenhofer, 1996), spatial search (Miller, 1993) and planning (Glasgow, 1995); GIS and ITS (Jing, Huang, & Rudensteiner, 1996; Shekhar, Kohli, & Coyle, 1993). There is a number of proposals to design and develop computational theory of the cognitive map which also include hierarchy as an important organising principle (Chown, Kaplan, & Kortenkamp, 1995; Gopal, Klatzky, & Smith, 1989; O'Neill, 1991), and which have been used in robotics for map building and navigation (Kuipers & Byun, 1991; Remolina, Fernandez, Kuipers, & Gonzales, 1999).

An important aspect of shortest path determination deals with *selection of relevant data*. Best-first search or heuristics¹, e.g. in A* algorithm, are used to eliminate nodes which do not satisfy some heuristic measure like a distance or cost, or some rule of thumb. A* is considered to be significantly important to the research of path finding problems in ITS due to its good performance (Shekhar et al., 1993). Another approach applies *hierarchy* mainly to partition a base graph into smaller graphs, and

¹ "Heuristics are criteria, methods, or principles for deciding which among several alternative courses of action promises to be the most effective in order to achieve some goal. They represent compromises between two requirements: the need to make such criteria simple and, at the same time, the desire to see them discriminate correctly between good and bad choices" Pearl, J. (1984). *Heuristics*. Reading, MA: Addison-Wesley.

so save storage and decrease computing time. Two groups of hierarchical algorithms emerge: one guarantees optimality of path retrieval (Jing, Huang, & Rundensteiner, 1998; Shekhar, Fetterer, & Goyal, 1997) while the other does not (Huang, Jing, & Rundensteiner, 1995; Shapiro et al., 1992). In all suggested solutions precomputed paths in a two-level graph are used, except for Huang et al. (1995) who propose a multi-level graph which classifies edges according to road types with preference of high-speed roads. Sungwong and Pramanik (1996) propose another hierarchical multi-graph model for representing topographic maps, which is based on the assumption that humans usually read maps in an increasing scale order. The path retrieval is faster due to precomputed paths, but the optimality is not discussed. The authors touch an important modelling aspect relevant also for our study: consideration of *human factors* in solutions for the shortest path problem. This is in line with research by Elliot and Lesk (1982) on algorithmic solutions producing optimal results which are non-satisfactory for humans. Their problem on giving driving directions combining a street map and yellow pages, involves both human factors (choice of the best kind of route) and algorithmic problems (choice of a good shortest path algorithm). The investigation shows that humans combine different search methods, suggesting hierarchical search as the right answer, "...as it shares many elements with what people do, and is reasonably comparable with computer requirements" (p. 261).

The approach often taken in the reviewed literature (e.g. Huang et al., 1995), uses rather sophisticated data structures and precomputations of shortest paths to improve the efficiency of the algorithm. Our approach uses a simple data structure, that of graph–subgraph structural hierarchy. Furthermore, it uses no precomputation, that is, it creates subgraphs *on-the-fly* according to the reasoning rules for that particular structural hierarchy. Hierarchization is based on the assumptions that people divide graphs in smaller subgraphs according to some classification criteria like speed or travel time (Hirtle & Hudson, 1991). In such a case wayfinding occurs in smaller but relevant datasets. The basics of our model are briefly discussed in the following section.

3. Hierarchical wayfinding

The conceptual model of hierarchical wayfinding, based on the theory of hierarchical spatial reasoning (HSR), describes the hierarchical structure of a network and explains how to reason on such a structure. The modelling process follows the requirements of hierarchical reasoning with structural hierarchies (Car & Frank, 1994):

1. design a model for the non-hierarchical case consisting of a single-level model of a network and a general-purpose, non-hierarchical wayfinding algorithm;
2. introduce a hierarchical structure and a set of rules stating how to form subgraphs, when to change levels, and how to turn single-level partial paths into a final result; and

- compare the results achieved by the hierarchical algorithm to those achieved by the non-hierarchical algorithm to find out if the investment in hierarchization pays off.

The last requirement represents quality control for reasoning with structural hierarchies. The comparison of the results is necessary because the errors produced in hierarchically organised systems approximate the ones observed in humans performing similar operations (e.g. Reno-San Diego problem, Stevens & Coupe, 1978). Hence, hierarchical spatial reasoning follows patterns of human cognition, as it computes approximations either to compensate for missing information or to simplify reasoning by selecting only necessary information (Fotheringham, 1992). Hence, hierarchical algorithms producing suboptimal solutions are also of interest.

A question arising then is “What is a good enough result?” rather than “What is the optimal solution?”. This is similar to the economics’ term *satisficing*, related to discovering any qualified object with as little search effort as possible (Simon, 1956). For example, Timpf and Frank (1997) and Winter (1999) address this issue showing that hierarchical spatial reasoning can compute increasingly better results and stops the computation when the good-enough result is achieved.

The model of hierarchical wayfinding is developed in accordance to the requirements of HSR. In the next subsections we briefly discuss the non-hierarchical and hierarchical case using parts of the formal specification for illustration (written as Gofer code).

3.1. Non-hierarchical case

- In the ontology² for the non-hierarchical case a road network is represented as a graph containing nodes and edges all being on a single level:

```
2. type Name = String
   data Pos x = Pos x x      -- 2d position
   data Node = Node Name (Pos x) -- node has a name and a
   position
   data Edge n = Edge Name n n -- edge has a name and 2
   nodes
   data Graph e n = Graph [e n] -- graph is a list of
   edges
```

- Item 2 (above): a part of the formal specification: type- and data-declarations indicating structures of objects.

² An ontology is an abstracted, idealized model of reality containing only those objects, relations among them, and rules that govern them, which are of interest in a particular reasoning system being designed Davis, E. (1990). *Representations of commonsense knowledge*. San Mateo, California: Morgan Kaufmann Publishers. A definition which is the closest to our use of the term ontology is “An ontology is a specification of a conceptualization.” (Tom Gruber, <http://ksl-web.stanford.edu/kst/what-is-an-ontology.html>, the web site was accessed on 10 April, 2000).

4. The graph is (a) embedded in Euclidean space allowing for edge length computation; (b) is planar excluding the cases like underpasses and overpasses; (c) is bidirectional allowing for traversing edges in both directions; and (d) has edges with assigned weights. This graph is abstracted from any further detail that a road network may have, in order to preserve the simplicity of the model. A part of the formal specification for graphs is given in items 5 and 6.

7. Items 5 and 6 (above): a part of the formal specification of the conceptual model of wayfinding shown for the object Graph.
8. We use the Dijkstra algorithm as the non-hierarchical wayfinding algorithm (NHWA). The Dijkstra algorithm is the best-known general-purpose shortest path algorithm (Dijkstra, 1959). The algorithm determines the lengths or weights of the shortest paths between the given node and any other node in a graph. Dijkstra produces an optimal solution result because it can search in the entire graph. The algorithm is implemented with adjacency matrix as the graph representation and the pseudo-code can be found in (Sedgewick, 1992, p. 466).

3.2. Hierarchical case

The ontology for the hierarchical case includes the non-hierarchical graph enriched by hierarchical levels. The non-hierarchical graph must be included so that the NHTA can be applied to each level independently. The *bottom-up* hierarchization method is used to create a graph–subgraph structural hierarchy with L levels: the lowest level L in the hierarchy contains the entire dataset; the highest level, also known as root or 0-level, contains the smallest dataset. A subgraph at level $i < L$ contains all the edges at that level and higher (4 i).

A criterion is used to assign edges to different levels in a hierarchy. In our example we use the road classification as the criterion: 0-level represents motorways, 1-level main roads and 2-level local roads (Fig. 1). Edges have nodes in common, at which it is possible to switch between different levels.

An important property of a hierarchical graph is the *level ratio* k , which is specified by a classification criterion. In the case of wayfinding the ratio can be translated to the ratio of e.g. average travel speeds v in different levels: $k = v_i/v_{i-1}$. With the help of this ratio, it is possible to make statements about paths found by hierarchical and non-hierarchical algorithm, and this will be used in the analysis of the test results.

The hierarchical algorithm (HTA) is a non-hierarchical algorithm enriched by a set of rules stating how to process a hierarchical graph (Car, 1998). It involves the creation of a subgraph from the original graph and path retrieval. HTA performs bidirectional search. The structure of HTA is presented in Fig. 2. NHTA can be any algorithm used to retrieve the shortest path in a subgraph. It terminates the search when the goal node or the first node with access to a higher level is reached. Graph reduction, i.e. creation of a subgraph from an initial graph, is a stepwise process: as soon as NHTA finds a node with access to a higher level (i), a new subgraph is created containing all edges of this (i) and other higher levels (4 i). The operation subGraph creates a subgraph (Graph [e | e es, p e]) by selecting the edges from the input graph, which satisfy the condition p (Section 3.1, items 5 and 6), e.g. to select the smallest subgraph, the condition p is that $level=0$.

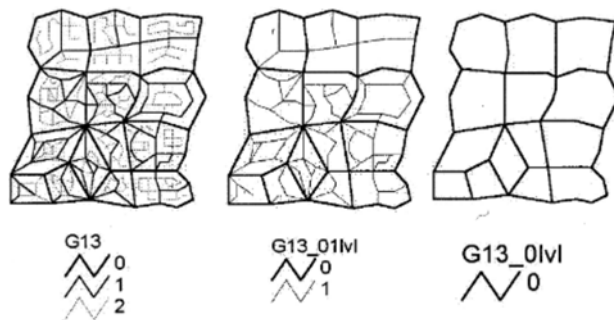


Fig. 1. A graph with three hierarchical levels: (left) full graph containing all nodes and edges (lowest level); (middle) a subgraph containing only two levels of detail; (right) the smallest subgraph containing only the highest level (0-level).

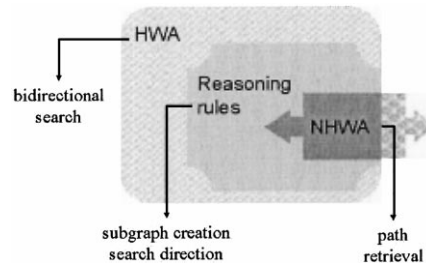


Fig. 2. The structure of the hierarchical wayfinding algorithm.

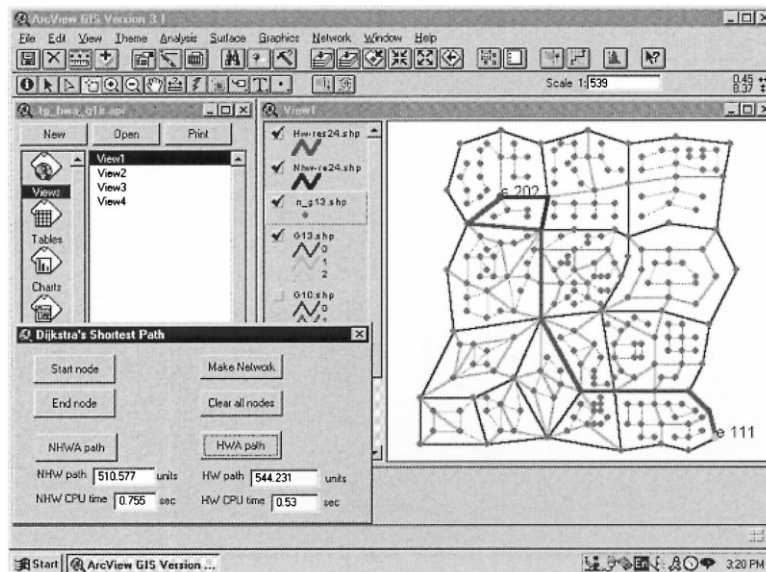


Fig. 3. ArcView application for hierarchical wayfinding: The graph G13 is presented in the View area together with paths calculated with NHWA (Nhw_res) and HWA (Hw_res). The sequence of nodes in both paths differs slightly just in the neighborhood of the node 202. The level combination is 0–1 meaning that the start node has access only to edges of level 1 and the goal node only to the edges of level 0. The window called “Dijkstra’s shortest path” provides values for path weights and CPU times, respectively.

3.3. Computational model

We implemented the hierarchical graph structure and the wayfinding algorithm in C++ (Borland) and integrated it in the software package ArcView (Fig. 3). ArcView has been selected as the application development environment primarily because of its customisation abilities: it supports GUI modification and integration of the DLLs using Avenue (Mehner, Car, & Taylor, 2000). The computational model is initially tested using the same graphs that were used to test the behaviour captured in the specification. It traces the same paths as the prototype (executable Gofor code) proving the success of the implementation so far. With the achieved

GIS application, a basis for further experiments with hierarchical wayfinding is provided. The achieved results are reported in the following sections.

4. Experiment set-up

4.1. Data sets

Table 1 contains a list of graphs used in the experiment. Random graphs have no particular form: a countryside road network or a town developed with hardly any planning are such examples. Radial graphs have a centre node whose outgoing edges are usually the more important roads: cities such as Vienna provide good examples. In a Manhattan graph, roads form a perpendicular and often regular grid, which is the case with road networks in a number of the US cities. The graphs G10, G11, G12 are smaller versions of G13. This holds for radial (G3x) and Manhattan graphs (G4x).

The edges in each graph were classified according to the average travel speeds. In a 3-level hierarchy the following level ratios are used:

1. 50:80:130 km/h

These speeds represent local roads, state roads and motorways and corresponds to travelling by a car, i.e. car transportation.

2. 1:10:100 km/h

These speeds correspond closely to public transportation, e.g. pedestrian: bus/tram:regional transport/high speed trains

4.2. Data sampling

Both NHWA and HWA retrieve a path which has the minimum path weight. The path weights $W_{NH_{ij}}$ and $W_{H_{ij}}$ between the nodes i and j are calculated as the sum of weights of all edges in the path:

Table 1
Synthetic graphs used in the experiment

Graph Id	Form	No. of nodes	No. of edges
G10	Random	41	81
G11	Random	101	177
G12	Random	97	204
G13	Random	264	370
G3	Radial	207	392
G31	Radial	325	565
G32	Radial	511	809
G41	Manhattan	280	445
G42	Manhattan	351	574
G43	Manhattan	518	876
G44	Manhattan	693	1184

$$W_{NH_{ij}} = \sum \text{edge length} \times \text{weight}$$

$$W_{H_{ij}} = \sum \text{edge length} \times \text{weight}$$

Note that the total path weight for $W_{H_{ij}}$ is the sum of the partial minimum weight paths retrieved at each hierarchical level.

Sample data consisted of paths calculated by the HWA and NHWA for 30 to 120 pairs of nodes per graph, total of 730 observations for all graphs. The choice of the node pairs depended on the level combination and graph coverage. For every calculation the following data was collected: start node, end node, level combination, path weight and CPU time for NHWA, path weight and CPU time for HWA, sequences of nodes in the corresponding paths. All calculations were conducted on a Pentium 233MHz PC, physically on the same machine under the same circumstances. Values for path weight and CPU time were calculated three times, and their means used as sample data. This procedure was adopted in order to eliminate as much as possible the instabilities of the operating system. No other applications were running during the testing time.

5. Analysis of the results

The analysis-step is the fulfilment of the third HSR requirement (see Section 3). It is expected to show whether the introduction of hierarchization pays off, and if so, to what extent. The analysis is based on comparison of the differences in path weights (DW) and in CPU-times (DCPU).

1. Difference in path weights (DW)

Let $W_{NH_{ij}}$ be the weight of the optimal path between the nodes i and j retrieved by NHWA, and $W_{H_{ij}}$ be the weight of a path retrieved by HWA, then the difference between these paths is

$$DW_{ij} = W_{H_{ij}} - W_{NH_{ij}} \text{ units}$$

and its sub-optimality rate (SOR) can be measured:

$$SOR_{ij} = \frac{W_{H_{ij}} - W_{NH_{ij}}}{W_{NH_{ij}}} \times 100\%$$

2. Difference in CPU times (DCPU)

Let $CPU_{H_{ij}}$ be the CPU time for a path computed by HWA between the nodes i and j , and $CPU_{NH_{ij}}$ be the CPU time for a path computed by NHWA for the same pair of nodes, then the difference between these times is:

$$DCPU_{ij} = CPU_{H_{ij}} - CPU_{NH_{ij}}$$

DCPU will be positive when HWA is faster than NHWA.

Furthermore, we intended to derive parameters for spatial hierarchization in networks. The assumption is that the size and form of a graph, number of levels, and level ratio combination are the good candidates for parameters for hierarchization in graphs. Therefore, the results are presented in four major groups: general findings, dependence on graph size, dependence on levels and dependence on graph form.

Statistical analysis³ was based on mean values for differences in path weight (DW) and the CPU times (DCPU) which were calculated per graph and for level combinations within each graph.

5.1. General findings

Generally, HWA finds longer paths but needs less CPU time than NHWA. Path weight differences (Fig. 4) vary from approximately 10 to 26% for the 50:80:130 ratio, and only up to 2% for the 1:10:100 ratio. This seems to be better performance than, for example, that of up to 50% for graphs with only two levels reported in (Shapiro et al., 1992).

Differences in CPU times (Fig. 5) for both level ratios are quite similar. They range from 0 seconds indicating that both algorithms need the same time, to approximately 0.25 seconds. This is equivalent to 0–28% difference indicating that generally HWA is faster than NHWA. Actual mean CPU times per graph are given in Fig. 6.

A deviation in behaviour is noticed for the largest random graph G13 and the smallest Manhattan graph G41 for the 1:10:100 ratio: negative DCPU (~ 0.02 s) indicates that NHWA performed faster than HWA. This deviation in behaviour is explained by the overhead of introducing the hierarchy (creation of subgraphs) into the processing. Very small graphs do not benefit from that as our results show. Further explanation is given in the following subsection.

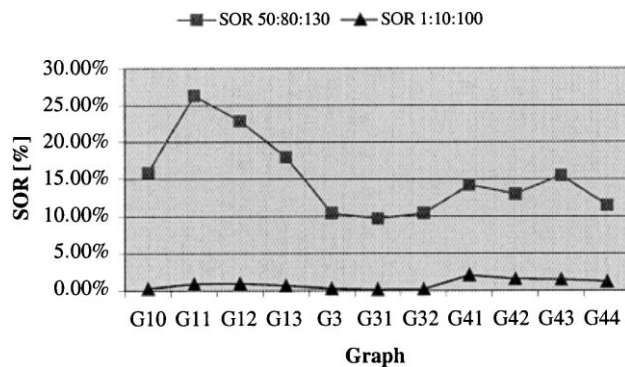


Fig. 4. Suboptimality rate (SOR) for path length differences.

³ We used the statistical package R (the R Development Core Team, <http://www.stats.bris.ac.uk/R>).

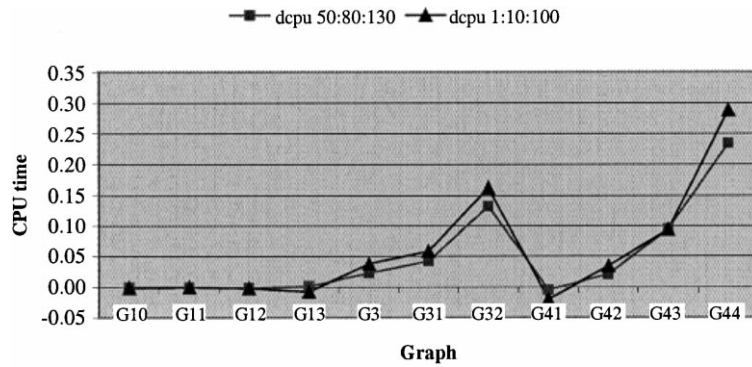


Fig. 5. Differences in CPU times.

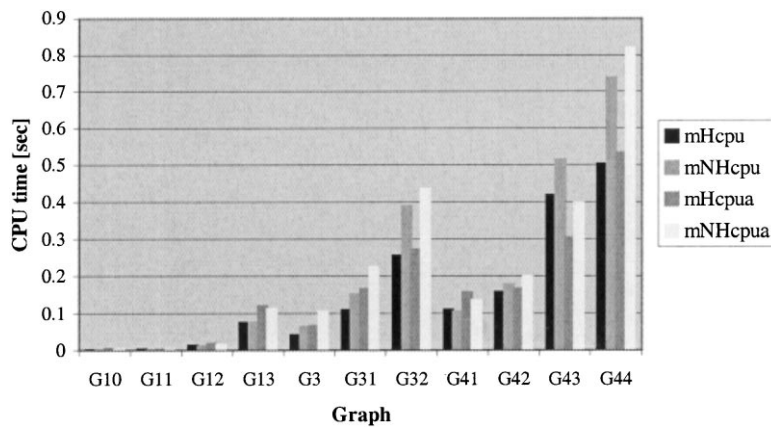


Fig. 6. Mean CPU times per graph for both level ratios: 50:80:130 — the two bars from the left, 1:10:100 — the two bars from the right.

5.2. HWA performance wrt graph size

The performance of a shortest path algorithm depends on the size of a graph, i.e. it increases with the increasing number of nodes in that graph. We investigated a general trend of HWA performance with respect to the graph size. Figs. 7 and 8 both show that the graph size does influence the performance of HWA, which confirms our expectations.

For the 50:80:130 level ratio path weight difference (DW) tends to vary significantly for smaller graphs with not more than 300 nodes. The variability then calms down but shows a tendency to increase with the increasing graph size. In the case of the level ratio 1:10:100 no dependence of DW on the graph size can be observed. This is explained by the fact that for large level ratios like 1:10, NHWA

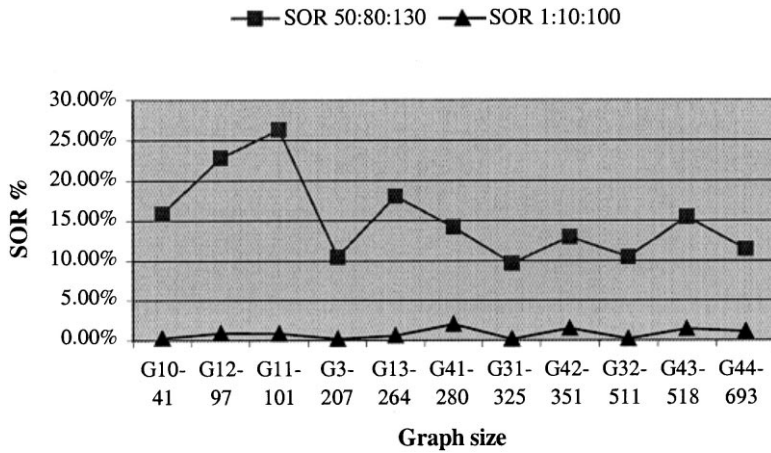


Fig. 7. Suboptimality rate (SOR) wrt the graph size.

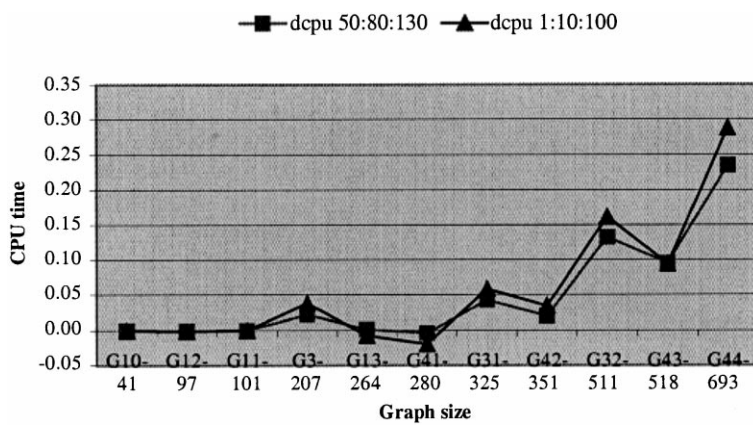


Fig. 8. Difference in CPU times wrt graph size.

will behave in the same way as HWA because the edges it chooses will correspond to those HWA chooses to switch levels.

Difference in CPU times grows with the increasing size of a graph and this holds for both level ratios. Again the size of 300 nodes seems to be a break-point indicating that the introduction of hierarchization for smaller graphs is not beneficial. On the contrary, HWA can perform worse than NHWA as in the cases of G13 and G41 for the 1:10:100 level ratio.

5.3. HWA performance wrt levels

This step of the analysis investigates the influence of the level ratio (50:80:130 and 1:10:100) and the level combination (0-0, 0-1, 1-1, 0-2, 1-2, 2-2) on the performance

of HWA. The level combination indicates the lowest level the start node has access to and lowest level the goal node has access to (see example in Fig. 3).

5.3.1. Level ratios

Generally, DCPU varies very little with the change of the level ratios as we observed in the previous subsections. This is not true for DW: it decreases with the increasing level ratio. Furthermore, the observations indicate that the sequences of nodes in the paths retrieved by NHWA and HWA depend on the level ratio. The analysis shows the following (note that the level ratio 50:80:130 translates to 1:1.6 ratio and 1:10:100 to the 1:10 ratio):

1. 1:1.6 level ratio: HWA and NHWA hardly ever find the same path, except when path search occurs in the smallest subgraph (e.g. containing only motorways).
2. 1:10 level ratio: in most cases HWA and NHWA find the same paths, i.e. consisting of the same sequence of nodes.

5.3.2. Level combination

The *t*-test shows that the HWA produces paths which are statistically significantly different from the paths produced by NHWA with respect to both DW and DCPU. Furthermore, the *F*-test shows that these significant differences depend on the level combinations involved.

5.3.2.1. Analysis for 50:80:130 ratio. For the *t*-test the null hypothesis states that the mean value of the path weight difference is zero ($H_0: \text{mean}(\text{DW}) = 0$). This hypothesis was rejected at a high significance level (0.01) for the level combinations excluding 0–0 and 0–1. In the 0–0 combination HWA and NHWA behave the same: in both cases the shortest path algorithm is applied to the smallest subgraph retrieving the same path. In less than half of the graphs the 0–1 combination influences the performance of HWA but at a low significance level (4 0.05). The same test was performed on CPU-time difference. The null hypothesis ($H_0: \text{mean}(\text{DCPU}) = 0$) is rejected at even higher significance level (0.001) across the level combinations.

For the *F*-test the null hypotheses state that in each graph the mean values of DW and DCPU for each level combination are all the same, i.e.

$$H_0 : \text{mean}(\text{DW})_0 = \text{mean}(\text{DW})_1 = \dots = \text{mean}(\text{DW})_2$$

$$H_0 : \text{mean}(\text{DCPU})_0 = \text{mean}(\text{DCPU})_1 = \dots = \text{mean}(\text{DCPU})_2$$

The null hypothesis for DW was rejected at the highest significance level (0.001) for all graphs. The null hypothesis for DCPU was rejected at the highest significance level (0.001) for all graphs except the smallest random graphs G10 (no significance at all) and G11 (0.01). This means that the values of DW and DCPU depend on the level combinations. Figs. 9 and 10 present the mean values calculated for individual level combinations for 50:80:130 ratio. The mean value of DCPU for the

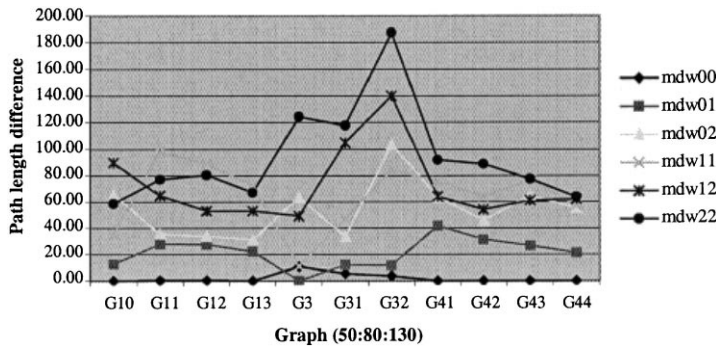


Fig. 9. Path length difference wrt level combination for the 50:80:130 ratio (note that *mdw* stands for the mean value of DW for a particular level combination).

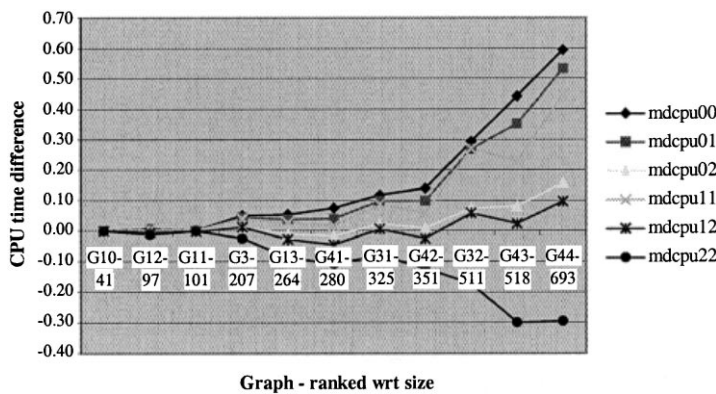


Fig. 10. CPU time difference wrt level combination for the 50:80:130 ratio (note that *mdcpu* stands for the mean value of DCPU for a particular level combination).

level combination 2–2 shows deviation in general behavior of the HWA: the algorithm’s performance deteriorates for all graphs over 100 nodes if the lowest level is involved being the worst behaviour observed.

5.3.2.2. *Analysis for 1:10:100 ratio.* The same test scenario was used here. The diagrams for 1:10:100 are very similar and are therefore not included here. In the *t*-test, the null hypothesis was not rejected for level combinations involving the top two levels (0–0, 0–1, 1–1). For the level combinations involving the 3rd level the null hypothesis was rejected at a low significance level 4 0.05 (0–2, 1–2) for less than half of the graphs, and at a slightly higher significance level of 0.01 for six graphs (2–2). In the case of the DCPU the null hypothesis was rejected at the highest significance level (0.001) for the majority of the level combinations and for all graphs again excluding the two smallest random graphs.

The *F*-test results for DW are slightly different from those reported for the 1:1.6 ratio: the null hypothesis is rejected at a low significance level (4 0.05). The null hypothesis for DCPU was rejected at the highest significance level (0.001) for all graphs except the smallest random graphs G10 and G11 (0.01) which is similar to the behaviour for the 1:1.6 ratio.

5.3.2.3. *Summary.* These results prove that the performance of the HWA depends on the level combination:

1. When the path transcends more of the levels, HWA is more likely to find a path different to that found by NHWA wrt the actual path weight (different route).
2. HWA is faster than NHWA in the most cases.
3. The sensitivity of HWA to the level ratio is more significant to the path length differences than it is to the CPU time.
4. The statistical significance of DW and DCPU is highest for the smaller level ratio (1:1.6).

5.4. Dependence on graph form

Further analysis has shown that there is statistical evidence that the performance of HWA depends on the form of a graph. The same testing scenario is applied and the same null hypotheses were stated but this time wrt to the graph forms.

The *t*-test shows that the HWA produces paths which are statistically significantly different from the paths produced by NHWA. Furthermore, the *F*-test shows that these differences statistically significantly depend on the form of graphs involved.

5.4.1. Analysis for 50:80:130 ratio

The *t*-test for DW rejected the null hypothesis at the highest significance level (0.001) except for the smallest random G10 and radial G3 graphs (0.01). For DCPU, however, the hypothesis is rejected at the highest significance level only for three graphs G31, G43, G44, at 0.05 level for G31 and is not rejected for all other graphs. The *F*-test rejects the null hypothesis at the highest significance level for both DW and DCPU, and proves though the dependence of the graph form and the performance of HWA.

5.4.2. Analysis for 1:10:100 ratio

The *t*-test for DW rejects the null hypothesis for majority of the graphs (eight out of 11), but at significance levels varying from 0.001 to 0.05. The *t*-test for DCPU rejected the hypothesis for only four graphs (at 0.001–0.01 levels) and does not reject it for the others. However, the *F*-test shows the same results as for the 50:80:130 ratio confirming the dependence of the graph form and the performance of HWA.

5.4.3. Summary

There is statistical evidence that the graph forms influences the performance of the HWA. In the results presented here, radial and Manhattan forms seem to be the

influential ones. We conclude that HWA outperforms NHWA in graphs of a recognisable structure. However, to make more specific statements further experiments are necessary.

6. Conclusions

We proposed a model for decision making which uses wayfinding in structural hierarchies. The theoretical background for this model is in hierarchical spatial reasoning. Its implementation offers an environment for testing the model on various graphs. A series of experiments was conducted on different kinds of synthetic graphs to observe the model's behaviour. Statistical analysis of the results proves that our hierarchical model is significantly more efficient than a non-hierarchical solution in graphs with recognisable structure.

A challenge of modelling decision-making is to develop a natural network hierarchy which can represent individuals' perceptions and abilities when wayfinding. The first step in that direction is to answer the question: *How does processing on such a graph structure affect the calculated path, in particular for different types of networks and which conditions lead to suboptimal wayfinding?*

From the test results we derived a number of parameters for hierarchization of spatial networks. These parameters influence the performance of the hierarchical algorithm and thus the behaviour of the model. They are graph size, level ratio, level combination, and graph form:

1. As the graph size (i.e. the number of nodes in a graph) increases the benefit of using HWA over NHWA increases. For graphs with less than 300 nodes hierarchization does not make any significant difference.
2. The level ratio has little effect on CPU times. Difference in path weights decrease with the increasing level ratio. For 1:1.6 (50:80:130) ratio HWA and NHWA hardly ever find the same path, whereas for 1:10 (1:10:100) ratio HWA and NHWA find the same paths in most cases.
3. The level combination significantly influences the performance of HWA:
 - the more levels transcended, the more likely that HWA and NHWA find different paths; and
 - HWA is faster than NHWA, but is inferior to NHWA only when the level combination includes the lowest hierarchical level (the worst performance is observed for 2–2, Fig. 10).
4. The graph form influences the performance of HWA, i.e. it outperforms NHWA in graphs of a recognisable structure such as radial or Manhattan.

A remaining open question, which we hardly touched in this paper, is that of sub-optimal, i.e. *satisficing* results produced by a hierarchical spatial reasoning method. Can we measure the goodness of a result? If so, what are the good candidates? Are they related to a particular application, or are they application-independent? This is worth a separate study.

The hierarchization parameters are important component in the description of the behaviour of the hierarchical wayfinding algorithm. Real networks may have many special cases, and therefore make the derivation of general case behaviour difficult to achieve. This is why we tested the hierarchical algorithm on synthetic graphs to determine default behaviour.

The long-term goal of this line of research is to develop a more generalised model for hierarchical reasoning in networks. Further experiments with HWA are needed to establish default behaviour:

1. vary the number of levels (the number of levels is fixed at three in this experiment); and
2. achieve stable behaviour wrt the set of reasoning rules by experimenting with different algorithms (e.g. A*) and using real datasets (to capture special cases and add them to the rules).

Hierarchical wayfinding is useful for GIS applications where a non-hierarchical solution can be improved through the proposed hierarchization, for example when dealing with large data sets or incomplete data and approximate reasoning. It can be used in a planning process as a reverse engineering tool to discern the most useful hierarchy for a network: e.g. to determine the location of exits from motorways, converting a road from one class to another, or investigate the effects of additional nodes/edges on the network behaviour.

The current model is inappropriate for time-critical applications like ITS but can be useful in commercial route finding and networking modules for calculating alternative routes, or in investigations of in-vehicle navigation techniques as a mechanism for selecting relevant data (Taylor & Blewitt, 1999). Application areas that could benefit from this model are transportation, tourism and any kind of spatial planning.

Acknowledgements

Help of Henny Mehner and Doerte Steup in coding and collecting testing data is greatly acknowledged. Harvey Miller helped in the process of articulating the idea presented in this paper.

References

- Agrawal, R., & Jagadish, H. (1994). Algorithms for searching massive graphs. *IEEE Transactions on Knowledge and Data Engineering*, 6(2), 225–238.
- Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1993). *Network flows: theory, algorithms, and applications*. Englewood Cliffs, NJ: Prentice Hall.
- Allen, G. L. (1999). Cognitive abilities in the service of wayfinding: a functional approach. *The Professional Geographer*, 51(4), 554–561.
- Car, A. (1997). *Hierarchical spatial reasoning: theoretical consideration and its application to modeling wayfinding*. Vienna, Austria: Dept. of Geoinformation, Technical University Vienna.

- Car, A. (1998). Hierarchical wayfinding — a model and its formalization. In M. Craglia, & H. Onsrud, *Geographic information research: transatlantic perspectives* (pp. 419–434). London: Taylor & Francis.
- Car, A., & Frank, A. U. (1994). General principles of hierarchical spatial reasoning — the case of wayfinding. In T.C. Waugh, *Spatial data handling (SDH 94)*. Edinburgh, Scotland: The IGU Commission on GIS, and the Association for Geographic Information.
- Chown, E., Kaplan, S., & Kortenkamp, D. (1995). Prototypes, location, and associative networks (plan): towards a unified theory of cognitive mapping. *Cognitive Science* (19), 1–51.
- Collier, W. C., & Weiland, R. J. (1994). Smart cars, smart highways. *IEEE Spectrum*, April, 27–33.
- Davis, E. (1990). *Representations of commonsense knowledge*. San Mateo, California: Morgan Kaufmann Publishers.
- Dijkstra, E. W. (1959). A note on two problems in connection with graphs. *Numerische Mathematik*(1), 269–271.
- Elliott, R. J., & Lesk, M. E. (1982). *Route finding in street maps by computers and people*. National AAAI-82 Conference, Los Altos, CA, American Association for Artificial Intelligence.
- Erwig, M. (1994). Graphs in spatial databases. Doctoral thesis, Fernuniversität Hagen.
- Fotheringham, A. S. (1992). Encoding spatial information: the evidence for hierarchical processing. In A. U. Frank, I. Campari, & U. Formentini, *Theories and methods of spatio-temporal reasoning in geographic space* (Vol. 639; pp. 269–287). Berlin: Springer-Verlag.
- Glasgow, J. (1995). A formalism for model-based spatial planning. In A. U. Frank, & W. Kuhn, *Spatial information theory—a theoretical basis for GIS* (Vol. 988; pp. 501–518). Berlin: Springer.
- Golledge, R. G. (1995). Path selection and route preference in human navigation: a progress report. In A. U. Frank, & W. Kuhn, *Spatial information theory—a theoretical basis for GIS* (Vol. 988; pp. 207–222). Berlin: Springer.
- Gopal, S., Klatzky, R. L., & Smith, T. R. (1989). NAVIGATOR: a psychologically based model of environmental learning through navigation. *Journal of Environmental Psychology* (9), 309–331.
- Hirtle, S. C., & Hudson, J. (1991). Acquisition of spatial knowledge for routes. *Journal of Environmental Psychology* (11), 335–345.
- Hirtle, S. C., & Jonides, J. (1985). Evidence of hierarchies in cognitive maps. *Memory and Cognition*, 13(3), 208–217.
- Huang, Y.-W., Jing, N., & Rundensteiner, E. A. (1995). *Hierarchical path views: a model based on fragmentation and transportation road types*. The 3rd ACM Workshop on Geographic Information Systems.
- Jing, N., Huang, Y.-W., & Rundensteiner, E. (1996). *Hierarchical optimization of optimal path finding for transportation applications*. ACM Conference on Information and Knowledge Management.
- Jing, N., Huang, Y.-W., & Rundensteiner, E. A. (1998). Hierarchical encoded path views for path query processing: an optimal model and its performance evaluation. *IEEE Transactions on Knowledge and Data Engineering*, 10(3), 409–432.
- Kuipers, B. J., & Byun, Y.-T. (1991). A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems*, 8(8), 17.
- Mehner, H., Car, A., & Taylor, G. (2000). From theory and formalization to successful GIS implementation: lessons learned. GIS Research UK (GISRUK 2000) Conference, York, UK, 4–7 April 2000.
- Miller, H. J. (1993). Modeling strategies for the spatial search problem. *Papers in Regional Science: the Journal of the RSAI*, 72(1), 63–85.
- O'Neill, M. (1991). A biologically based model of spatial cognition and wayfinding. *Journal of Environmental Psychology* (11), 299–320.
- Papadias, D., & Egenhofer, M. (1996). *Algorithms for hierarchical spatial reasoning*. University of Maine: NCGIA and Department of Spatial Information Science and Engineering.
- Pearl, J. (1984). *Heuristics*. Reading, MA: Addison-Wesley.
- Remolina, E., Fernandez, J. A., Kuipers, B., & Gonzales, J. (1999). Formalizing regions in the spatial semantic hierarchy: a AH-graphs implementation approach. In C. Freksa, & D. M. Mark, *Spatial information theory. Cognitive and computational foundations of geographic information science, COSIT '99* (Vol. 1661; pp. 109–124). Berlin: Springer-Verlag.
- Sedgewick, R. (1992). *Algorithms in C++*. Reading, MA: Addison-Wesley.

- Shapiro, J., Waxman, J., & Nir, D. (1992). Level graphs and approximate shortest path algorithms. *Networks*, 22, 691–717.
- Shekhar, S., Fetterer, A., & Goyal, B. (1997). Materialization trade-offs in hierarchical shortest path algorithms. In M. Scholl, & A. Voisard, *Advances in spatial databases. Proceedings of the 5th International Symposium, SSD '97* (Vol. 1262; pp. 94–111). Berlin: Springer Verlag.
- Shekhar, S., Kohli, A., & Coyle, M. (1993). *Path computation algorithms for advanced traveller information systems*. IEEE 9th International Conference on Data Engineering.
- Simon, H. (1956). Rational choice and the structure of the environment. *Psychological Review* (63), 129.
- Stevens, A., & Coupe, P. (1978). Distortions in judged spatial relations. *Cognitive Psychology* (10), 422–437.
- Sungwong, J. & Pramanik, S. (1996). *HiTi Graph Model of topographical road maps in navigation*. 12th International Conference on Data Engineering, New Orleans, LA.
- Taylor, G. & Blewitt, G. (1999). *Virtual Differential GPS and Road Reduction Filtering by using Map Matching*. 12th International Technical Meeting, ION GPS'99, Nashville, USA.
- Timpf, S., & Frank, A. U. (1997). Using hierarchical spatial data structures for hierarchical spatial reasoning. In S. C. Hirtle, A. U. Frank, *Spatial information theory — a theoretical basis for GIS, International Conference COSIT'97* (Vol. 1329; pp. 69–83). Berlin: Springer-Verlag.
- Tversky, B. (1992). Distortions in cognitive maps. *Geoforum*, 23(2), 131–138.
- Winter, S. (1999). Topological relations in hierarchical partitions. In C. Freksa, & D. M. Mark, *Spatial information theory. cognitive and computational foundations of geographic information science, COSIT '99* (Vol. 1661; pp. 141–155). Berlin: Springer-Verlag.