Journal of Visual Languages and Computing (1998) 9, 647–657
Article No. vl980093

Performance of Layout Algorithms: Comprehension, not Computation

HELEN C. PURCHASE

The Department of Computer Science and Electrical Engineering, The University of Queensland, St. Lucia 4072, Brisbane, Queensland, Australia

Received 6 January 1998; accepted 9 July 1998

Many algorithms address the problem of rendering an abstract graph structure as a diagram in as efficient and as elegant a manner as possible. The criteria for judging the worth of these algorithms are typically the extent to which they conform to common aesthetic criteria (e.g. minimising the number of crossings, maximising symmetry), or their computational efficiency. The algorithms are not usually judged on their ability to produce diagrams that maximise humans' performance on tasks which require their use.

This paper presents an example experimental methodology for considering the relative worth of eight layout algorithms with respect to human performance, together with details of an experiment using a single graph. The results indicate that, with the exception of one algorithm, there is no statistical difference between the performance data of the algorithms when applied to this graph, indicating that they produce drawings of comparable difficulty. This result is despite the different aesthetic bases for the algorithms.

1. Introduction

MANY GRAPH LAYOUT ALGORITHMS produce diagrams which represent an underlying graph structure, attempting to depict the relational information in a form that makes it easier to read, understand and use [1]. These algorithms tend to embody aesthetic criteria, the assumption being that the resultant graph drawing helps the human reader to understand and remember the information embodied in the graph [2].

Comparisons between the quality of the various algorithms have traditionally been from a computational point of view: quantitative criteria such as run time, the length of the longest and shortest edges, and distribution of the nodes are used to determine the superiority of one algorithm over another [3].

Few studies have been performed to compare the quality of layout algorithms from the perspective of human comprehension and use. Himsolt includes an informal 'rating' system (between 1 and 5 stars), but this is simply based on viewing the drawings [4]. Blythe, in looking at the most appropriate layouts for the depiction of social networks [5], considers the two criteria of clustering and central proximity, but common layout algorithms were not investigated in that study.

It is important that human experiments be performed on these algorithms, so that, rather than judging an algorithm by its computational efficiency, the products of the algorithms themselves can be judged with respect to how much they assist human

1045-926X/98/060647+11 $30.00/0

comprehension. In particular, with the increasing use of automatic graph layout algorithms in a large number of software environments for application domains (e.g. social networks [5], entity relationship diagrams [6], object-oriented design diagrams [7]), the relative worth of these algorithms for human comprehension needs to be determined.

Previous studies [8, 9] reported experiments on the human comprehension and use of graph drawings with respect to the individual aesthetics on which layout algorithms are traditionally based. The studies considered the effect of aesthetics separately, and found that reducing the number of edge crossings is the most important aesthetic, while there is some evidence that reducing edge bends and increasing perceptual symmetry also contribute to better human comprehension. Orthogonality, and maximising the minimum angle between adjacent edges were found to have no effect.

In the experiment reported here, eight different graph layout algorithms were investigated, with respect to a single graph. Briefly, the experiment entailed subjects answering questions about eight drawings of the same graph, each drawing produced by a different algorithm. Measurements were taken of both the number of errors made and the time taken to answer the questions. Using statistical tests, the effect of the algorithms on the response time and the number of errors was determined. Tukey's WSD pairwise comparison procedure [10] was then used to determine if there were significant performance priorities between the algorithms.

Experiments were run on-line to study these eight algorithms, and the results indicate that, in general, the subjects performed equally well on the products of all the algorithms for this graph; the only exception was the algorithm by Seisenberger, which was significantly more difficult than the two force-directed algorithms and Tukelang's incremental algorithm.

This paper describes the nature of the on-line system used for the experiments and the experimental methodology (the algorithms, graph drawings, experiment and data), and presents and discusses the results.

2. The Experiment

2.1. Definition

There are two ways in which human performance using graph drawings may be measured. A purely *relational* method measures the efficiency and accuracy with which people can read a graph structure and answer questions about it. Such graph-theoretic questions need to be generic and application-independent, and may include questions of the form "What is the shortest path from node A to node B?" A more application-specific method would rather consider a graph *interpretation* task: in this case it is more appropriate that the effectiveness of the graph drawing is measured within the context in which the application-specific graph is usually used. Thus, instead of eliciting answers to specific questions asked about the graph itself, it is more suitable to look at whether the graph has assisted the user in accomplishing a particular application task. Suitable questions for this approach would include (in the area of software engineering) 'What object classes would be affected by changing the external interface to class X?'

In this experiment, like the prior aesthetics experiments [8, 9], the *relational* reading of a graph drawing is considered, leaving the *interpretive* consideration of algorithms for

a later study. The questions that are used in this experiment to measure relational understandability are:

- How long is the shortest path between two given nodes?
- What is the minimum number of nodes that must be removed in order to disconnect two given nodes such that there is no path between them?
- What is the minimum number of edges that must be removed in order to disconnect two given nodes such that there is no path between them?

2.2. The Graph

As an initial attempt at addressing the problem of determining the most appropriate layout algorithm with respect to human performance, an experimental methodology was proposed, and the experiment performed using a single graph.

The graph for this experiment needed to be carefully designed so that node-pairs could be identified which gave a suitable range of values for the three questions. It was important that the answers to the three questions that the subjects would be asked were not the same for each version of the graph. Thus, a set of node-pairs was defined that would give correct answers to the first question (the shortest path) of either 2, 3 or 4; a set of node-pairs was defined that would give correct answers to the second question (the number of nodes to remove) of either 2 or 3; and a set of node-pairs was defined that would give correct answers to the third question (the number of edges to remove) of either 2 or 3.

In addition, the graph structure was limited by the algorithms that were to be applied to it: it needed to be undirected, with maximum degree 4. The graph has 17 nodes and 29 edges (Figure 1).

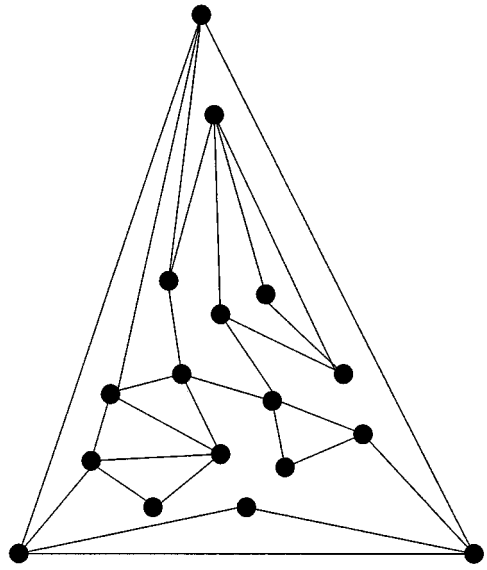

Figure 1. A drawing of the graph structure used for the experiments

2.3. The Algorithms

The algorithms are all implemented in the GRAPHED system (version 4.1.7-beta [11]), which was used for the creation of all the graph drawings. Eight algorithms were used, resulting in a set of eight experimental graph drawings. Much of the following information about the algorithms implemented in GRAPHED has been provided by Himsolt [12], and has been documented elsewhere [3, 4]. The names given to the algorithms are, as much as possible, the same as those used in the papers by Himsolt [4] and Brandenburg *et al.* [3].

FD-FR: This force directed algorithm by Fruchterman and Reingold [13] is based on the original idea by Eades [14]. It produces a drawing that attempts to display symmetric structures with few edge crossings.

FD-K: This is another force directed algorithm by Kamada and Kawai [15] based on Eades [14]. The drawing produced by this algorithm is similar to that produced by FD-FR, with a more even distribution of nodes.

POGB: This is an implementation of an algorithm for planar orthogonal grid drawing with bends minimisation, designed by Tamassia [16].

GRAPHED provides two implementations of this algorithm with differences in the manner in which the coordinates of the nodes are assigned. Himsolt [12] describes the differences in the two algorithms as being related to the starting conditions: the first algorithm (POGBa) generates the planar embedding from the drawing, while the second one (POGBb) uses a planarity test. He notes that the result is not deterministic.

PG: This planar grid drawing algorithm is based on the one designed by Woods [17]. Unlike the product of the POGB algorithm, the PG drawing has many sloped edges.

PGS: The planar grid drawing with straight line edges is based on the algorithm by Fraysseix, Pach and Pollack [18], with improvements as suggested by Chrobak and Payne [19].

SEIS: This algorithm is documented in a thesis by Seisenberger [20]. The first step is always the PGS algorithm. This is followed by repeated compression of the drawing in both the x and y directions [12]. This algorithm is not always deterministic.

Tu: The algorithm used here is the incremental algorithm designed by Tunkelang [21]. The product is a drawing that has an even distribution of nodes similar to that expected from a force-directed algorithm.

Figure 2 shows the eight drawings of this graph produced by these eight algorithms. Note that these drawings have been variously scaled for the purposes of effective presentation: in the experiment they were displayed with all nodes being of equal size.

2.4. Experimental Methodology

The structure of the experiment was identical to that of the previous investigation of graph drawing aesthetics [8], the only difference being the experimental graph drawings used. In the previous experiment, the drawings were produced by hand (with careful varying of the individual aesthetic values as the independent variables); in this case, the drawings were produced by the algorithms in GRAPHED. For completeness, the details of the experiment are repeated here.

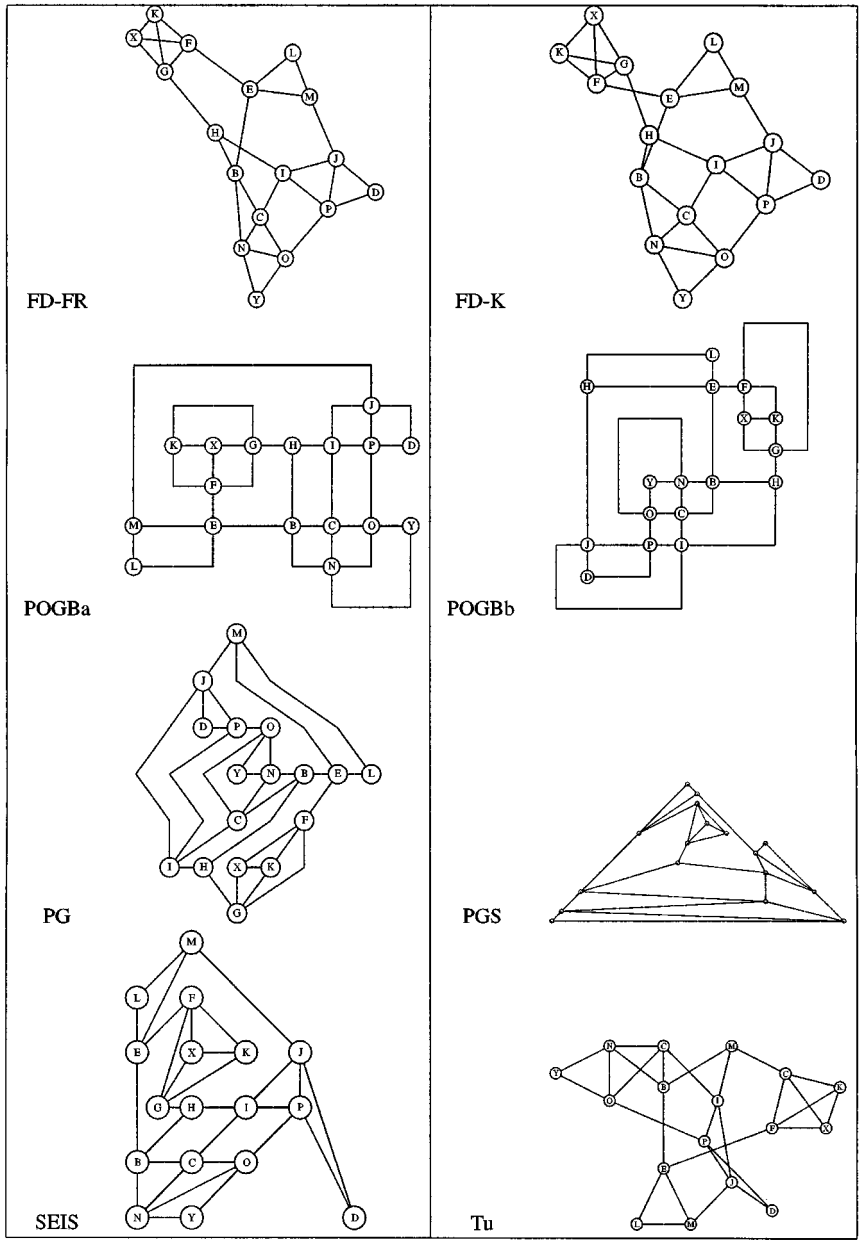

Figure 2. The eight experimental graph drawings

Experiments were run on-line. Each subject interacted with a unique experimental program, robust enough to withstand the unexpected input of a novice user. These programs were created by a system designed and implemented for the purposes of running experiments relating to graph drawings, which was used in the previous

aesthetics experiment [8]. The on-line system defined experimental programs of the following form.

1. A brief description of graphs, and definitions of the terms *node*, *edge*, *path* and *path length* were presented, followed by an explanation of the three questions that the subjects would be required to answer about the experimental graph drawings. A simple example graph drawing, with the three questions and their correct answers, was shown. At this stage, the subjects were asked by the experimental administrator if they had any questions about graphs in general, or about the experiment. It was important to ensure that all the subjects knew what was expected of them.
2. The three questions were asked of six 'practise' graph drawings, to familiarise the subjects with the nature of graph drawings and the questions, and to ensure that they were comfortable with the task before tackling the experimental graph drawings. The subjects were not told that these graph drawings were not experimental.
3. A 'filler' task which engaged the subjects' mind on a small problem unrelated to graphs was presented. This ensured that their performance on the subsequent experimental graphs was not affected by any follow-on effect from the practise graphs. A simple logic puzzle, designed to take approximately 1 min, was used.
4. The eight experimental graph drawings were each displayed three times, once for each question. The system randomly ordered the presentation of the drawings, their orientation and the order of the questions. The experimental drawings were randomly interspersed with other non-experimental drawings that did not form part of the experiment.

   The questions themselves were randomised too: although the form of the three questions asked of each drawing was the same, the pair of nodes chosen for each question was randomly selected from a list of node-pairs (as defined in an external question file). This ensured that any variability in the data could not be explained away by the varying difficulty of the questions. The two relevant nodes for each question were highlighted in black on the screen, ensuring that the measured response time did not include time taken to locate the nodes. Figure 3 shows the on-line display of two questions.

   The subjects typed their answers to the questions. The use of an on-line system enabled two dependent variables to be recorded: the time taken for the subject to answer each question (the 'response time'), as well as the correctness of the answer. This allowed analysis to be performed on two measures of understanding.

The experiment was therefore controlled for the questions and the graph, the independent variable was the algorithm used to produce the drawing, and the two dependent variables were the time taken to answer the questions and the number of errors made for each drawing.

A within-subjects analysis method was used in order to reduce any variability that may have been attributable to the difference between the subjects (e.g. age, experience). Any learning effect was minimised by the large number of graph drawings used in the experiment, the inclusion of the practise graph drawings, and the randomisation of the ordering of the graph drawings. Fifty-five third-year computer science students at The University of Queensland took part in the experiment, for a reward of $10.

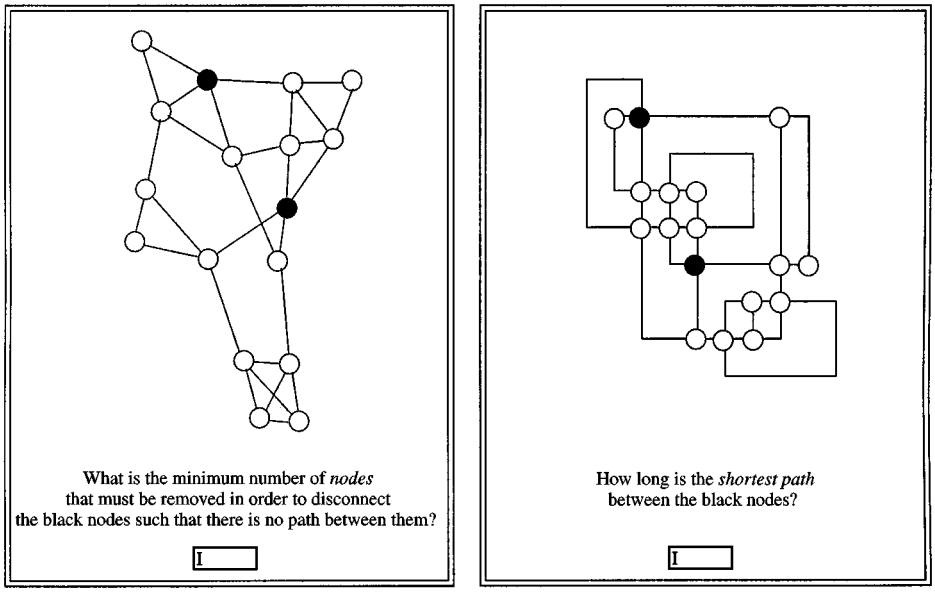

Figure 3. Example questions during the experiment: the two nodes to be highlighted for each question were chosen randomly at run time, as was the orientation of the graph drawing

3. Results

The average number of errors and the average response time for the eight experimental graph drawings are shown in both tabular and chart form in Figure 4. The within-subject analysis of variance showed that the main effect of the algorithms was *significant* for errors ($F_{7,378} = 2.856$, $\alpha = 0.05$) but *not significant* for response time ($F_{7,378} = 1.464$, NS).[a]

To determine the relative effect of the algorithms on the average number of errors, and attempt to place a priority ordering on their difficulty, the set of drawings were then subject to a Tukey's pairwise comparison to determine which algorithms differed significantly from one another.

The Tukey's WSD pairwise comparisons procedure showed that, for the error data, the SEIS drawing produced significantly more errors than the FD-FR ($F_{8,378} = 10.12$, $\alpha = 0.05$), Tu ($F_{8,378} = 11.486$, $\alpha = 0.05$) and FD-FK ($F_{8,378} = 14.63$, $\alpha = 0.05$) drawings. There were no other significant pairwise differences.

4. Analysis and Discussion

The average response times for the products of the eight algorithms were not significantly different, implying that the subjects did not perceive the drawings of this graph to be of varying difficulty. But the average number of errors for the drawings *were*

[a] The statistical analysis used here is a standard ANOVA analysis [22], based on the critical values of the F distribution: α is the level of significance, and results that are not significant are indicated by NS.

	FD-FR	FD-K	POGBa	POGBb	PG	PGS	SEIS	Tu
Errors	0.164	0.109	0.236	0.255	0.291	0.236	0.436	0.145
Time	66.79	68.54	77.55	74.76	73.23	78.41	78.67	77.74

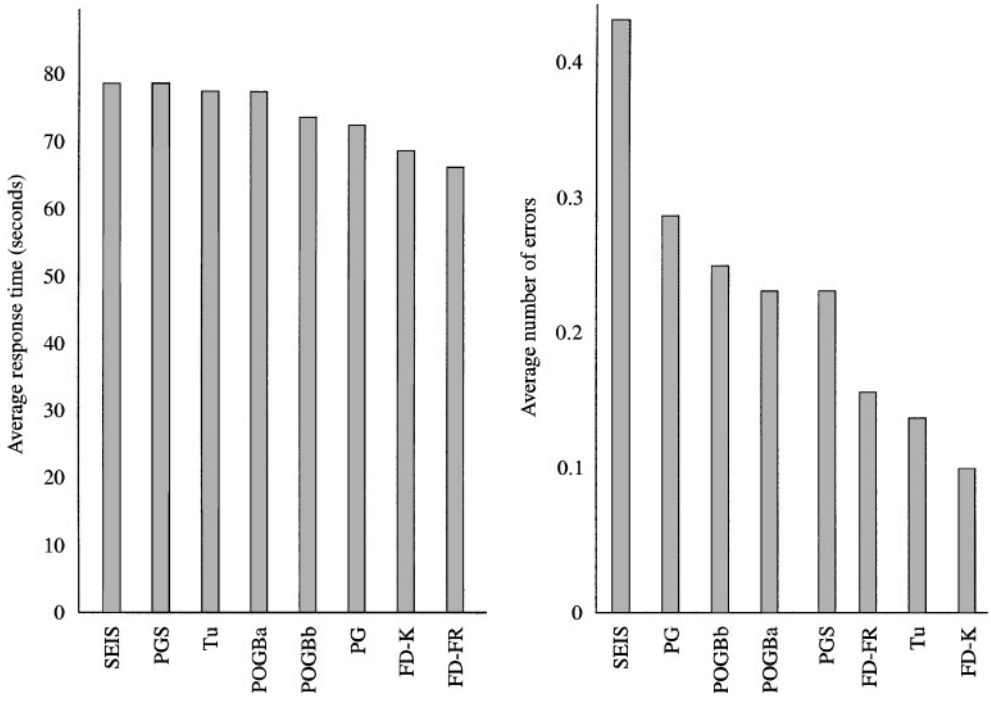

Figure 4. The average response time and average number of errors for each graph drawing

significant, indicating that, despite this perception, there was indeed a difference in the difficulty of the drawings. Further analysis revealed where this variance lay: the SEIS drawing produced significantly more errors than the two force-directed drawings and Tukelang's incremental algorithm.

These results therefore indicate that, apart from the SEIS algorithm, all the algorithms produce drawings that are of comparable difficulty. While the chart of the average number of errors in Figure 4 seems to indicate that the force-directed algorithms have better performance than the grid-based ones, there is no statistical evidence to support this.

This is an interesting result: despite the differing aesthetic bases for the algorithms, they produce comparable human performance results. In particular, this outcome is unexpected given the results of the previous experiment which considered individual aesthetics [8]. In this prior experiment, there was overwhelming evidence to support the reduction of the number of edge crossings over all other aesthetics, partial support for minimising edge bends and maximising symmetry, and no support for orthogonality and maximising the minimum angle.

The outcome of the experiment reported here shows no difference between the force-directed algorithms FD-FR, FD-K and Tu (which tend to maximise symmetry at

the expense of a few crosses, with no bends), and the grid-based algorithms POGBa and POGBb (which maximise orthogonality, have no crosses, and minimise the number of bends). Given the strong support for reducing the number of crosses in the prior experiment (with only partial support for symmetry) the expectation may be that the grid-based algorithms would produce better performance. This was not the case.

There are two possible explanations for this outcome. It may be that it is not possible to separate the effect of individual aesthetics, as was attempted in the prior experiment, and that the interactions between the aesthetics are also important. For example, the combination of both reducing the number of bends and maximising orthogonality may produce a more significant effect than merely reducing the number of edge crossings. Further experimentation will need to be performed in order to investigate the interaction effects of the aesthetics.

Secondly, the result may be explained intuitively by considering that, in this experiment, the force-directed algorithms produced drawings which included only very few crossings. It may be that the number of edge crossings only has a significant effect on understandability when there are many of them: perhaps a 'critical mass' of crossings needs to be reached before their number creates a serious understandability problem. This issue of critical mass may, of course, also be relevant to the other aesthetics: perhaps it is the case that orthogonality only has a positive effect if the orthogonality measure of the drawing is greater than a certain amount.[b]

5. Conclusions

The aim of these empirical tests was to investigate graph drawing algorithms, to indicate to the designers of systems which algorithms are best from a human readability point of view. The experiment has shown that for a simple graph, only one of the eight algorithms gives a poor performance when compared with the others.

This result indicates that despite the range of aesthetic bases for these layout algorithms, it is difficult to say that one algorithm is 'better' than another from a relational understanding point of view.

This conclusion must, of course, be interpreted within the limitations of a formal, controlled experiment. It is common knowledge that all formal experiments are limited by their parameters [10]: this is an inevitable consequence of the controlled experimental method. In particular, this preliminary study has considered the performance effect on a single graph, with three specific, relational questions, and the generalisability of these results are therefore limited to within these clearly defined parameters.

These limitations do not detract from the significance of this experiment, however. This is the first attempt to compare graph drawing algorithms from a human, rather than a computational, point of view, and the first attempt at defining an experimental methodology by which this important question may be considered.

[b] Purchase and Leonard have devised computational metrics by which the 'amount' of an aesthetic in a graph drawing may be measured [23]. Metrics for the aesthetics of crosses, bends, maximising the minimum angle, orthogonality, upward flow and symmetry have been formally defined.

The experiment therefore opens a wide, new field of empirical investigation, and there are many avenues for further studies which may either corroborate or invalidate the results presented here. In particular:

- Would the same algorithmic comparative results be forthcoming on a set of different graphs (for example, graphs with different structures and sizes)?
- Would the same algorithmic comparative results be forthcoming with different relational performance measurements (for example, if subjects were asked to trace paths between nodes, determine the maximum degree of the graph, or identify sub-graphs)?
- Would the same algorithmic comparative results be forthcoming if an interpretive approach were taken (for example, if subjects were asked to read the graph in the context of application domains like object-oriented design diagrams or data-flow diagrams)?

The problems associated with formal experiments requiring the participation of human subjects (which are not present in similar computational investigations) prevented these additional questions from being considered within this initial experiment. By presenting this methodology and these results, it is hoped that this new area of investigation may be adopted more widely, so that the answers to these important issues may be resolved for the benefit of both algorithm designers and users.

Acknowledgements

I am very grateful to Michael Himsolt (who assisted extensively with the definitions and descriptions of the algorithms as implemented in GRAPHED), to Murray James (who designed and developed the on-line system), to David Leonard and Daniel Naumann (who assisted with the experiments), and to Julie McCreddon (who guided the statistical analysis).

References

1. G. Di Battista, P. Eades, R. Tamassia & I. Tollis (1994) Algorithms for drawing graphs: an annotated bibliography. *Computational Geometry: Theory and Applications* 4.
2. R. Tamassia, G. Di Battista & C. Batini (1988) Automatic graph drawing and readability of diagrams. *IEEE Transactions on Systems, Man and Cybernetics* SMC-18, 61–79.
3. F. Brandenburg, M. Himsolt & C. Rohrer (1995) An experimental comparison of force-directed and randomised graph drawing algorithms. In: *Proceedings of Graph Drawing Symposium*, 1995. F. Brandenburg, ed., Lecture Notes in Computer Science, Vol. 1027, Springer, Passau.
4. M. Himsolt (1995) Comparing and evaluating layout algorithms within GRAPHED. *Journal of Visual Languages and Computing* 6.
5. J. Blyth, C. McGrath & D. Krackhardt (1995) The effect of graph layout on inference from social network data. In: *Proceedings of Graph Drawing Symposium*, 1995. F. Brandenburg, ed., Lecture Notes in Computer Science, Vol. 1027, Springer, Passau.
6. C. Batini, M. Talamo & R. Tamassia (1984) Computer aided layout of entity-relationship diagrams. *Journal of Systems and Software* 4, 163–173.
7. G. Booch (1990) *Object-Oriented Design.* Benjamin-Cummings, Menlo Park, CA.
8. H. C. Purchase (1997) Which aesthetic has the greatest effect on human understanding? In: *Proceedings of Graph Drawing Symposium*, 1997. G. Di Battista, ed.), Lecture Notes in Computer Science, Vol. 1353, Springer, Rome, Italy.

9. H. C. Purchase, R. F. Cohen & M. James. An experimental study of the basis for graph drawing algorithms. *ACM Journal of Experimental Algorithmics* 2.
10. R. Gottsdanker (1978) *Experimenting in Psychology*. Prentice-Hall, Englewood Cliffs, NJ.
11. M. Himsolt (1990) GRAPHED user manual. Universität Passau.
12. M. Himsolt (1997) Personal communication.
13. T. Fruchterman & E. Reingold (1991) Graph drawing by force-directed placement. *Software-Practice and Experience* 21, 1129–1164.
14. P. Eades (1984) A heuristic for graph drawing. *Congressus Numeratum* 42, 149–160.
15. T. Kamada & S. Kawai (1989) An algorithm for drawing general undirected graphs. *Information Processing Letters* 31, 7–15.
16. R. Tamassia (1987) On embedding a graph in the grid with the minimum number of bends. *SIAM Journal of Computing* 16, 421–444.
17. D. Woods (1982) Drawing planar graphs. Ph.D. thesis, Department of Computer Science, Stanford University, 1982. Technical Report STAN-CS-82-943.
18. H. de Fraysseix, J. Pach & R. Pollack (1990) How to draw a planar graph on a grid. *Combinatorica* 10, 41–51.
19. M. Chrobak & T. H. Payne (1990) A linear time algorithm for drawing a planar graph on a grid. Technical Report UCR-CS-90-2, Dept. of Math. and Comput. Sci., Univ. California Riverside.
20. K. Seisenberger (1991) Termgraph: Ein system zur zeichnerischen darstellung von strukturierten agenten und petrinetzen. Technical Report, University of Passau.
21. D. Tunkelang (1992) A practical approach to drawing undirected graphs. Technical Report CMU-CS-94-161, Carnegie Mellon School of Computer Science.
22. P. R. Hinton (1995) *Statistics Explained*. Routledge, London.
23. H. C. Purchase & D. Leonard (1996) Graph drawing aesthetic metrics. Technical Report 361, University of Queensland Department of Computer Science.