A Search Set Model of Path Tracing in Graphs

Jessica Q. Dawson, Tamara Munzner and Joanna McGrenere

Abstract

We present a predictive model of human behaviour when tracing paths through a node-link graph, a low-level abstract task that feeds into many other visual data analysis tasks that require understanding topological structure. We introduce the idea of a *search set*, namely the paths that users are most likely to search, as a useful intermediate level for analysis that lies between the global level of the full graph and very local level of the shortest path between two nodes. We present potential practical applications of a predicted search set in the design of visual encoding and interaction techniques for graphs. Our predictive model is based on extensive qualitative analysis from an observational study, resulting in a detailed characterization of common path-tracing behaviours. These include the conditions under which people stop following paths, the likely directions for the first hop people follow, the tendency to revisit previously followed paths, and the tendency to mistakenly follow apparent paths in addition to true topological paths. We verified the prominence of a previously proposed tendency that people follow the closest-to-geodesic branch between a node and the goal, but found complex interactions between this tendency and others we

observed, including the impact of path continuity on behaviour. The algorithmic implementation of our predictive model is robust to a broad range of parameter settings. We provide a preliminary validation of the model through a hierarchical multiple regression analysis comparing graph readability factors computed on the predicted search set to factors computed globally and for the shortest-path solution. The tested factors included edge-edge crossings, node-edge crossings, path continuity, and path length. Our approach provides modest improvements for predictions of response time and error using search-set factors. We also found key differences in the relative weighting of the importance of the factors that affect response time versus error.

Keywords

Graph readability, path tracing, evaluation, regression analysis, graph drawing metrics, aesthetic criteria, models.

Introduction

We present a characterization of human behaviour during the visual data analysis of graphs that are visually encoded as nodes connected by edges. This characterization arises from an extensive qualitative analysis from an observational study that focused on the low-level task of tracing paths through the graph, a task abstraction that underlies the many higher-

level analysis tasks that entail understanding topological structure. In the study, 12 participants completed path-tracing tasks by demonstrating their search progress on a tablet. The detailed characterization of common path-tracing behaviours was the base for a predictive model of paths that users are likely to search.

Our model is built around the concept of a *search set*, which we propose as a way to capture an important facet of human behaviour: it is the set of all paths that a user follows while attempting to find the shortest path between a source and goal node. The search set provides a scope of analysis that lies in between the global level of the entire graph and the local level of the shortest-path solution to the path-tracing problem. Our model predicts this set of paths that participants are most likely to search as ordered discrete groups of paths that are equivalence classes, where within each group all of the paths are postulated to be equally likely paths.

Much of the previous work on characterizing human behaviour during the visual analysis of graphs has been devoted to understanding what factors affect the quality of the layout. Many factors have been proposed, such as the number of edge-edge crossings, the total curvature of edge bends, and the total area of the drawing. Early work[1–4] simply proposed factors and then immediately incorporated them into optimization-based layout algorithms. The factors were considered as constraints to minimize or maximize, and thus a

major emphasis was on factors that are amenable to automatic computation. Subsequent work[5–12] has since begun to determine *graph readability* – whether and how properties of graph layouts, including these longstanding factors and more recently proposed ones, directly affect human *graph reading behaviour* and their understanding of graph structure in the context of specific tasks. This initial work has yielded some intriguing preliminary results, but the characterization of human behaviour during the visual analysis of graphs is far from complete.

The search set concept can be applied to this quality assessment problem by calculating these factors on only the subsets of the graph encountered during a specific tracing task; we hypothesized that accounting for the paths most relevant to the user's search would improve upon previous work that has measured factors across the entire graph globally, or on the solution path locally. As a demonstration of this application of the search set, and as a preliminary validation of our predictive model, we conducted a careful comparison of graph readability factors through a hierarchical multiple regression analysis. Our results show a modest benefit of measuring factors on the search set over previous work.

Our work has two primary contributions: (1) a detailed characterization of path-tracing behaviours based on observational data of human subjects, and (2) a predictive model of the search set. We also provide two secondary contributions: (3) the introduction of the

concept of the search set itself, as an intermediate level for behavioural analysis that lies between the full global graph and the narrow solution path considered in some previous work; and (4) a multiple regression analysis that provides preliminary support for the predictive model. A more detailed articulation of each of these contributions is provided in the later Discussion and Future Work section.

The paper begins with motivation, background, and the research questions that guided our work. We continue with the related work on observation of human graph reading behaviour and evaluating factors for graph readability, Next, we describe our user study, which included observation of users completing a path-tracing task. For clarity, we present our analysis in three separate sections. First, we present our qualitative analysis approach, and provide descriptions of common human path-tracing behaviours that we identified. Second, we discuss our predictive behavioural model for the search set. Third, we conduct a preliminary validation of our search set model by comparing the effectiveness of measuring factors at the solution-path, search-set and global levels for predicting path-tracing difficulty. We conclude with a discussion of the implications of our findings regarding human path-tracing behaviour and the search set concept, the value of our methodology for untangling the importance of different graph readability factors, the practical applications of a predictive model of the search set in terms of implications for the

design of visual encoding and interaction techniques for graphs, the limitations of our study and analyses, and our plans for future work.

Motivation and background

We first discuss six considerations that motivated this work: why characterizing behaviour benefits the information visualization community, why path tracing is an interesting abstract task to study, why we conjecture that the search set would be a useful scope to investigate, what behaviours have already been identified, how a predictive search set model could be used in practice, and why to analyze with multiple regression.

Why characterize behaviour?

Characterizing human behaviour during visual data analysis is the underlying goal of most experimental work in visualization. This characterization is useful in its own right as a theoretical foundation to visualization knowledge.[13] It directly informs the subsequent use of the exact techniques for visual encoding and interaction that are studied.[14] More broadly, this kind of empirical work often spurs the design of new techniques[15,16] and supports the development and refinement of quantitative metrics for the quality of a visual encoding that better correspond to human judgments of its utility.[12,17]

Why path tracing?

The abstract task[18] of path tracing is a canonical low-level task that serves as a building block for the many higher-level tasks that involve understanding topological structure of graphs.[19] Path tracing has been widely studied[7–9,11,20–25] because it underlies many real-world use cases for visual data analysis with graphs.[19,26,27] A concrete example is a medical investigator generating a hypothesis about disease transmission in a graph where nodes represent people and edges represent known contact between them, who is checking whether a short path exists between one infected individual and another.

The low-level task of path tracing tasks for graphs laid out as node-link diagrams is similar in spirit to the low-level task of quantity judgement for tabular data. Many experiments to characterize the accuracy of length, angle, and area judgements have been conducted for table layout techniques such as bar charts and line graphs[13,28], scatterplots[29], and horizon charts[14]; in contrast, graph techniques are less well characterized.

Why introduce a search set?

The common evaluation of factors for laying out a graph is *global*; that is, one or more factors would be measured across the entire graph. Recent work suggests that for path-tracing tasks a *local* approach is better. Ware et al.[23] investigated the effect of factors

measured along the *solution path*; that is, the specific path that is the correct solution for a specific path-tracing problem. They showed that this approach was effective for predicting path-tracing difficulty, and found no additional benefit from including globally measured factors. The intuition behind this result is that global measurements take into account too much of the graph: a graph that scores poorly globally for a set of factors may nevertheless have paths that are easy to trace in regions of minimal clutter.

However, just as global measurement may consider too much of the graph, we suspect that measuring local factors only on the solution path does not take enough of the graph into account. We propose that an even better solution lies between these two extremes, where the full subset of the graph that is relevant for the task at hand is considered; we name this subset the *search set*. Specifically, we hypothesize that the prediction of path-tracing difficulty can be improved by accounting for the impact of important factors on the search set: in this case, all of the paths that a user follows during the tracing task before encountering the solution. Our logic is that if crossings on the solution path slow a user down, then so should other crossings on paths investigated before the solution path is found.

Search set factors may also be more broadly applicable than solution-path factors because some instances of path-tracing tasks do not have solutions. In a disconnected

graph, for example, a solution path between two points may not exist, but it is possible to calculate a set of paths that a user is likely to follow while making that determination.

What behaviours are already identified?

A previous study from Huang et al. identified the *geodesic tendency;*[7] that is, when attempting to trace a path from a source node to a goal node, people have a tendency to follow the branch that is the *closest-to-geodesic* from the current node to the goal. A *geodesic* is the geometric straight line between two points.

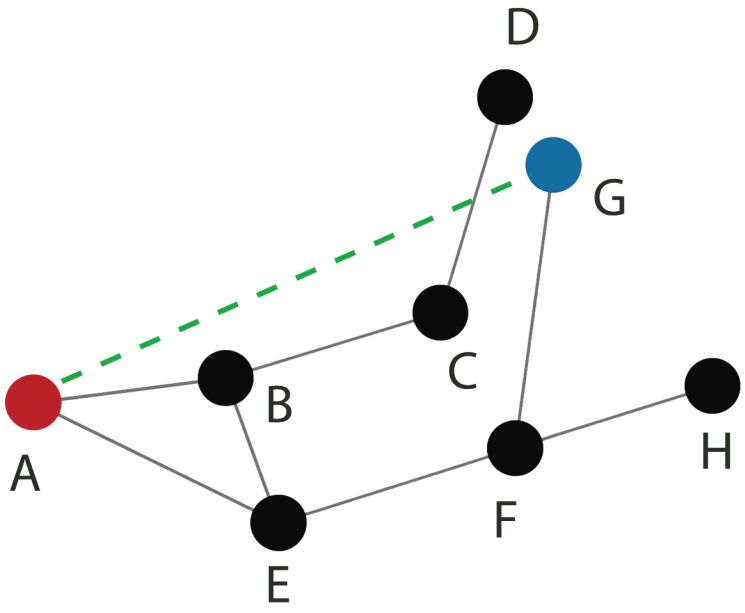

Figure 1 -When tracing a path from A to G, the geodesic tendency predicts that a user would first follow the incorrect path A–B–C–D that is closer to the straight line between A and G, before following the solution-path A–E– F–G.

Redrawn from Huang et al.[7]

Figure 1 shows an example: users looking for the path from A to G would likely start by following the series of closest-to-geodesic branches from A to B to C to D. They would next deviate from the closest-to-geodesic and follow A to E, before returning to following the closest-to-geodesic branches from E to F to G. In this case, the solution path was the second path explored.

Our initial investigations showed us that the geodesic tendency concept is a clear step in the right direction, but lacks sufficient predictive power for a complete model of human behaviour characterization. We noted that a corollary of the geodesic tendency is that certain paths are unlikely to be followed, either because they are not on a closest-to-geodesic branch, or because they would naturally fall after the solution path in an exploration sequence. For example, in Figure 1 AEFH would not be followed because the solution path AEFG is already found. Our definition of the search set is exactly the likely set of paths that a person would search along the way to finding the solution path. In this case, the set is the paths ABCD and AEFG.

What are the uses of a predictive search set model?

A predictive search set model could be used for factor measurement, for general salience measurement, for interaction techniques that dynamically adjust layouts, and for static layout algorithms.

Factor measurement: Our original motivation for developing a predictive model of path tracing was to predict the difficulty of path tracing in the context of experimental design. We noted this open problem when designing a controlled experiment to investigate different visual encoding techniques for graphs, and found no reliable way to control for or measure path tracing difficulty despite the significant previous work on graph readability factors; it was a confounding variable that distorted our experimental results. Our preliminary validation of the model addresses exactly this application.

Salience measurement: A predictive model provides a *salience* measure for an edge that is targeted with respect to a specific query of two nodes. Given a layout of the entire graph and the two specific nodes as input, a predictive model of the search set provides an ordered list of paths (or sets of equivalence classes of paths) as output. The ranking of a particular path against that set can be checked: does it appear early in the list, late in the list, or not at all? This list can then be used as a black box by any visual encoding or interaction technique that takes a path or an edge as input and provides a rank as output.

Jänicke and Chen[30] discuss many uses of visual salience within a general comparison framework. They proposed an image-space salience metric that is guided by observations of low-level human visual perception but is agnostic to the data type. A predictive search set model offers an alternative way to gauge the salience that is informed by human

behaviour and dataset semantics in terms of topological structure, in addition to the geometric layout of the visual encoding. Search-set salience can be used in all of the applications that they propose, and also as a core primitive for graph layout in any context that requires measuring layout quality or changing a layout with respect to a subset rather than all of the graph.

Interaction techniques: Interaction approaches that rearrange a subset of the nodes, such as the Bring and Go[31] technique, typically minimize the cognitive impact of disruption to an original layout by maintaining spatial consistency; search-set salience could guide the movement to be aligned with behavioural tendencies. Search-set salience might also support new techniques that affect a larger portion of the graph by suggesting relatively subtle local changes rather than extreme rearrangements.

It could also be useful for permanent rearrangement. For example, if the user interactively indicates that a small set of nodes are important to emphasize, the graph could be rearranged so that paths between them are easier to follow according to the model's prediction. A search-set model could also guide prioritizing specific paths deemed to be important according characterized behaviours; for example, the predicted direction of the first hop in the path.

Static layout: Layout techniques that measure the quality of multiple layout alternatives and choose the best result, as with Design Galleries[32], could use search-set salience as a black box. Search-set salience provides interesting possibilities in guiding multilevel layout techniques that achieve speed and quality improvements with multi-pass approaches that act on subsets of the nodes separately, such as incremental refinement from coarse to fine levels of a compound graph hierarchy.[33] Search-set salience would support subsets that are not spatially localized to a contiguous geometric region because it is based on topology. It could also be used to develop new global post-processing layout improvement techniques in a similar spirit to node overlap removal.[34] A novel family of two-pass layout approaches could optimize for search-set salience as a second pass with respect to a small set of important nodes or edges identified in a first pass using an appropriate importance measure[35] such as a centrality metric for social network analysis.[36]

Why use multiple regression?

Evaluative studies of factors affecting graph readability have predominantly relied on significance testing to conclude that a factor is important or not. A handful of studies have further attempted to create priority lists of factors based on their relative importance, but these have largely been based on significance testing[21,37] or human judgements.[38,39] Such approaches are limited in their ability to untangle how different factors interact[40] and the

magnitude of the effects are rarely reported. In their study of factors on the solution path, Ware et al. introduced the use of multiple regression for evaluating the impact of factors on task difficulty[23], and argued for its further use in evaluation of graph readability factors. Multiple regression inherently provides a measure of effect size, and has the additional benefit of assigning quantitative weights to factors, which can be useful when considering the importance of one factor over another.

To our knowledge, only one other study, by Huang and Huang[8], has used multiple regression for untangling the relative contributions of factors: they examine the relative impact of global edge-edge crossings and global crossing angles on four different measures of task difficulty. We differ from previous work in our focus on the incremental validity of these factors. *Incremental validity* is a concept from clinical psychology that focuses on "the degree to which a measure explains or predicts a phenomenon of interest, relative to other measures,"[41] and on the utility of variables in terms of cost and efficiency.[42] Although the term *incremental* typically has negative connotations in discussions of research contributions, we note that here it is being used in a specific technical sense of determining whether useful information has been added beyond what is already available. In particular, we show that factors measured on the search set show modest improvements

over what can be explained with previously studied factors measured on the solution-path and global levels.

Related work

We discuss the previous work most closely related to the behavioural analysis that we conducted to build our predictive model, and to the factor-based analysis that we use to validate it.

Descriptions of human graph-reading behaviour

Our work is situated within two veins of evaluative studies that have looked to human behaviour to assess and explain graph readability: studies using eye tracking to gather data while humans read graphs, and studies focused on human behaviour when manually arranging graphs.[5–7,11,12,24,25,43,44]

Several studies have used eye tracking towards the goal of understanding and describing how users actually read graphs. Pohl et al. [6] found that force-directed layout outperformed orthogonal and hierarchical layout on a set of 5 tasks, one of which was identifying a path between two points. For each task, the authors used eye-tracking data to briefly explain their results in terms of observed behaviours, but did not dig into untangling the relationship between behaviours and the characteristics of the different layout styles.

Burch et al.[5] similarly used eye tracking to study visual exploration behaviours of participants when solving a typical hierarchy exploration task in traditional, orthogonal, and radial hierarchical layouts. Their results are also primarily descriptive, and the authors are only able to make limited recommendations for layout creation based on their findings.

Huang et al.[7,25] used eye tracking to study users completing path-tracing tasks, with the goal of actually observing the effect of edge-edge crossings on the user's gaze. This work is the most similar to our own in terms of goals: they identify and provide evidence for the existence of a specific behavioural tendency, the geodesic tendency, which strongly affects path tracing. We build on the observational approaches just described to complete a deeper study and characterization of human graph reading behaviours through a full model of path tracing, where the geodesic tendency is clarified and extended in the context of additional tendencies, and show that it is possible to predict a set of paths that a group of users is likely to follow

From a detailed analysis of eye-tracking data, Körner[9,24] developed a sequential model of graph comprehension and examined the impact of factors at its different stages. This cognitive model is intended to disambiguate between potential underlying mechanisms of visual cognition within a high-level framework. In contrast, our work provides a behaviour-

based model specific enough to be used for measuring factors, and we do not attempt to provide any explanations for the cognitive mechanisms.

In contrast to this work using eye tracking, we asked users to illustrate their search progress and demonstrate their thinking by tracing their paths on a tablet. This approach follows a second vein of observational studies where users were asked to manually generate or arrange graphs, and then their behaviour and the resulting graphs were analysed to reveal what factors and criteria they used.[11,12,43,44.] When tasked with creating understandable graphs, the participants in one study by Purchase et al.[44] favoured minimizing edge crossings and maximizing orthogonality. Van Ham and Rogowitz[12] asked users to create layouts that best represented the structure of a data set with distinct clusters found that users also sought to minimize edge-crossings, but also tended to create distinct convex hulls to delineate clusters as distinct perceptual groups. Our work is similar in its emphasis on behavioural analysis through observation of user process and interaction with graphs. While this previous work led to a refined understanding of the impact of existing factors, the described behaviours are primarily about graph creation tasks and do not attempt to model the behaviours exhibited in completing tasks that require reading the graphs.

Evaluation of factors for graph layouts

Many studies have sought to evaluate the impact of factors on human understanding of graphs. Factors studied include edge bends,[20,21] edge length,[12,23] orthogonality,[11,21,44] angular node resolution,[21,37] edge crossing angles,[8,45] clustering,[12] node spacing,[11,37] edge stress,[11] and edge-edge crossings.[8,9,11,12,21,23,24,37,44] More recent studies explored the effects of visual features on memorability[46] and on mental map preservation using dynamic layouts.[47] However, many factors that are commonly incorporated into layout algorithms remain unexamined by controlled experiments. One such factor is node-edge crossings, which we evaluate for the first time in our analysis.

Impacts of layout style and factors on task performance: Some previous work has shown that some factors may have a varying impact depending on the task that a user must perform with a graph. Purchase[21] used a shortest path identification task, as well as two tasks related to graph connectivity, in a study that concluded that edge-edge crossings are the most important factor. However, in a study of factors impacting sociogram use, Huang et al. [37] concluded that edge-edge crossings are only important for path-tracing tasks. Similarly, in his study of eye movements, Körner[24] found evidence that edge-edge crossings have no impact during 'search' tasks, but do have significant impact during the 'comprehension' tasks that involve considering the edges between nodes. Conversely, Dwyer et al. [11] found

no effect of edge-edge crossings for either path-tracing or connectivity tasks. These analyses focus on global layout and do not discuss trade-offs or implications of factors. In our study, we focus specifically on untangling the factors that effect path-tracing task difficulty. We find that the concept of search set may shed some light on the underlying reasons for the mixed results in previous work, as covered in the later Discussion and Future Work section.

Measuring factors at local, search set and global levels: The prior studies that we have discussed thus far have all focused on globally measured factors, with only one exception; Ware et al.[23] studied factors measured on the solution path. In exploring the effectiveness of factors for predicting response time in a shortest path identification task, they identified a new factor of path continuity, and found significant effects of solution-path length, path continuity, total and average line length of the path, total branches on the path, average crossing angle on the path and total edge-edge crossings on the path. Further, they showed that with only four of these factors – solution-path length, continuity, edge-edge crossings and branches - they could account for 78.4% of the variance of response time in their study, and they identified path length and continuity as having the largest contributions. A crucial finding was that the globally measured edge-edge crossings did not account for any additional variance on top of the solution-path factors. To our knowledge, only one other

study[8] compares the effect of graph readability factors measured locally in addition to globally on response time. We are the first to do so for error. Our study is also the first investigation of factors for either time or error at three different levels: we introduce the search-set level in addition to the global and solution-path levels.

Untangling factor importance across measures: In examining different tasks, the most common measure of graph readability used in previous work has been response time. Some studies have also examined the relationships of factors to measures such as error, user preference, and (less commonly) cognitive load.[48] While previous work has recognized that, for example, the factors that make a task take longer do not necessarily correspond to those that increase the likelihood of error, little work has examined the relative difference of factor importance for different measures of difficulty. In our later discussion, we provide a nuanced discussion of the ways that several factors affect response time and error in varying ways.

User study design

We collected data through a lab-based observational user study with 12 participants, who were asked to complete 144 unique path-tracing trials over two sessions, while using a Wacom Cintiq tablet to demonstrate the paths that they followed.

Research questions

Our study design was guided by a set of initial questions about the plausibility of the search set concept, and its use as a basis for measuring factors that impact path-tracing difficulty: (Q1) can we identify distinct path-tracing behaviours and evidence of the search set? (Q2) how common are these path-tracing behaviours? (Q3) can we predict the search set based on observed path-tracing behaviours? and (Q4) how much improvement over previous work is gained by calculating factors for graph layout on a predicted search set? To answer these questions we performed an extensive analysis, which we present in multiple parts in subsequent sections. We answered Q1 and Q2 through observation and characterization of path-tracing behaviours. We then explored Q3 by incorporating these observed behaviours into the development of a simple predictive behavioural model for the search set. Finally, we focused on answering Q4 through a hierarchical multiple regression analysis to compare factors measured on the solution-path, search-set, and global levels.

Our intention to observe and characterize path-tracing behaviours to explore the search set concept guided our choice of a tablet interface for recording the participants' search process. The study was predominantly designed to support the planned hierarchical multiple regression presented later, which influenced the design of the task, the graphs used, the procedure, and the number of trials.

Piloting and rationale

In designing the study, we were particularly concerned with how easily users would be able to physically trace their search process, in tuning the difficulty of the experimental task, and in ensuring that the interface was useable without interaction.

We began by piloting the study with 6 participants recruited from the authors' department, who were available for extensive piloting but who had little previous knowledge of the research project. The sessions lasted about 30 minutes. Participants were tasked with finding the shortest path (of length 2 to 5 hops) between two nodes in graphs printed out on paper. The graphs had either n = 50, 75 or 100 nodes (where the number of edges = $2n$, as dictated by the Watts Strogatz model described later in this section). During the task, participants were asked to trace their search progress by pointing at the nodes that they considered with a capped pen. One of the authors observed participants during the session, and also videotaped the pen movements from above for later review. For the final design of the observational sessions we chose to display the graphs on an interactive tablet screen in order to support data logging for later analysis.

Physically tracing vs. eyetracking: One goal of piloting was to investigate whether having users physically trace their search process would allow us to adequately capture the paths

that users directly reasoned about during that process; we concluded that this design would indeed suffice for the exploratory nature of our investigation.

Although eye tracking has been widely used in previous work, it entails high-overhead analysis for targeted questions; our goal was to analyze low-level data quickly to focus on broader considerations in later analysis phases. Moreover, it was important for our study that we be able to identify exact nodes that users were considering in very dense regions of the graph. We were concerned that that eye tracking would not guarantee sufficient resolution because of limitations of precision of both the data that can be collected and of the common methods for visualizing the data: heat maps, for example, cannot show progression of paths over time, and gaze plots often suffer from overplotting.[49] Further, we wanted to generate a dataset that could eventually be analyzed statistically as part of our model development and validation, which meant that we needed records of paths that could be compared to the predicted paths our model would produce. While there are methods and algorithms for extracting and comparing scanpaths, it would have required considerable extra development and analysis effort in order to match the eye-gaze data to predicted paths. In contrast, having participants point directly at nodes on the tablet allowed us to capture exactly which nodes they were considering and store the paths directly – we were then able to both visualize this data in time series plots, as well as use this data for

comparison against the output of our predictive model, with minimal massaging or transformation required.

We observed from the pilot sessions that, with a small amount of practice, participants became accustomed to moving the pen at the same time that they were searching and this tracing functioned much as a think-aloud observation equivalent. We did not observe any problems arising from occlusion of the screen by the participants' hand, but we did note that participants did not always hover over nodes that they could reason about with their peripheral vision. The inability to pick up peripheral vision is also a known constraint of eye tracking.[50]

Tuning task difficulty: Another goal of piloting was to tune the difficulty of our experimental task. Because we were interested in overall path tracing behaviour, we wanted to have a combination of trials that would span the range of difficulty from easy to hard – but not impossible – to achieve a good mix of both success and failure cases to study. In real life, if a task is impossible users will simply give up, a tendency that can confound controlled experiments. Ware and Bobrow[51] report an example where the difficult tasks were too difficult and had shorter times than the easier tasks because the users gave up.

In our piloting, we found that 5 hop paths were often too difficult for the graph sizes and density we used – our participants frequently gave up without completing the task, and

often became discouraged. Thus, for the final design of the observational session, we chose to use 2, 3 and 4 hop paths only, with 4 hops being the maximum that we felt users could reasonably complete without any additional technique support.

We found that a graph size of 75 nodes and 150 edges graph reliably produced trials in the target range of difficulty. While this size may appear small compared to many real-world datasets, our experiments with larger graphs of up to 100 and 200 edges resulted in trials that were too difficult. We carefully tuned the complexity of the visual appearance to approximate the information density of complex situations while still allowing for controlled experimentation, and we succeeded in surpassing the 42-node size used by the previous study of local path tracing.[23] A major constraint on graph size was the size of tablet screen – nodes had to be large enough to be easily acquired with the tablet pen.

Finally, we also noted that a subset of our pilot participants would search for a long time (up to 5 minutes) before giving up, and their search behaviour became less consistent and more chaotic over time. In the final experiment, we capped search time at 90 seconds. This cap ensured that the experiment could be finished within reasonable length of time, and that we would be capturing common behaviours in a realistic situation.

Avoiding interaction: Our final goal was to ensure that the interface was usable for the tracing task without any scaffolding in the form of interactive techniques. Many interactive

techniques are used in practice for highlighting, navigation, and rearrangement. Examples include simple colour highlighting of the segment underneath the cursor (the combination of an edge and the two nodes that it attaches to), highlighting using alternate channels such as oscillating motion,[51] highlighting larger topologically connected sets such as all 1-hop neighbours of a node,[52] more elaborate interaction techniques such as Bring and Go[31] that rearrange the layout temporarily, navigation support for zooming and panning, and allowing users to manually rearrange nodes and edges to disambiguate occlusion. This experiment is designed to understand what humans do in the static case, which we consider to be the natural baseline. As we argue above, one possible use of a predictive model is exactly to determine when these scaffolding techniques are necessary and when they could be dispensed with, either globally or locally. Moreover, many of these techniques would introduce a confounding effect of interaction time.

We observed that participants struggled with node-edge crossing ambiguities to the point where the task was too difficult. To provide purely static support for resolving node-edge crossing ambiguities, in the final version of the interface each node was drawn with a small white halo around it, as shown in Figure 2: edges terminating at that node would pass on top of the halo and connect directly to the node, but unconnected crossing edges were drawn underneath the halo, resulting in a small gap between the edge and the node.

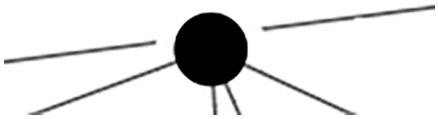

Figure 2 - Example of a halo drawn around a node to support identification of node–edge crossings. The near- horizontal edge crosses the node. The four other edges, which pass through the halo, connect to the node.

Participants

We recruited 12 participants using fliers posted on campus (4 female, aged 20 – 33, M = 23.4). All were students with normal or corrected-to-normal vision and regular colour vision. They each received $10 per hour of participation, and a bonus $5 for returning to complete both sessions.

Task

We used a shortest-path identification task. In each experimental trial, participants were shown a graph with a source and a goal node coloured red and blue respectively, and were asked to find the shortest path through the graph from the red node to the blue node. The remaining nodes were coloured black. Participants were told explicitly that the path would always be 2, 3, or 4 hops in length. Participants were also asked to complete the task as fast as they could while also trying to avoid making unnecessary errors.

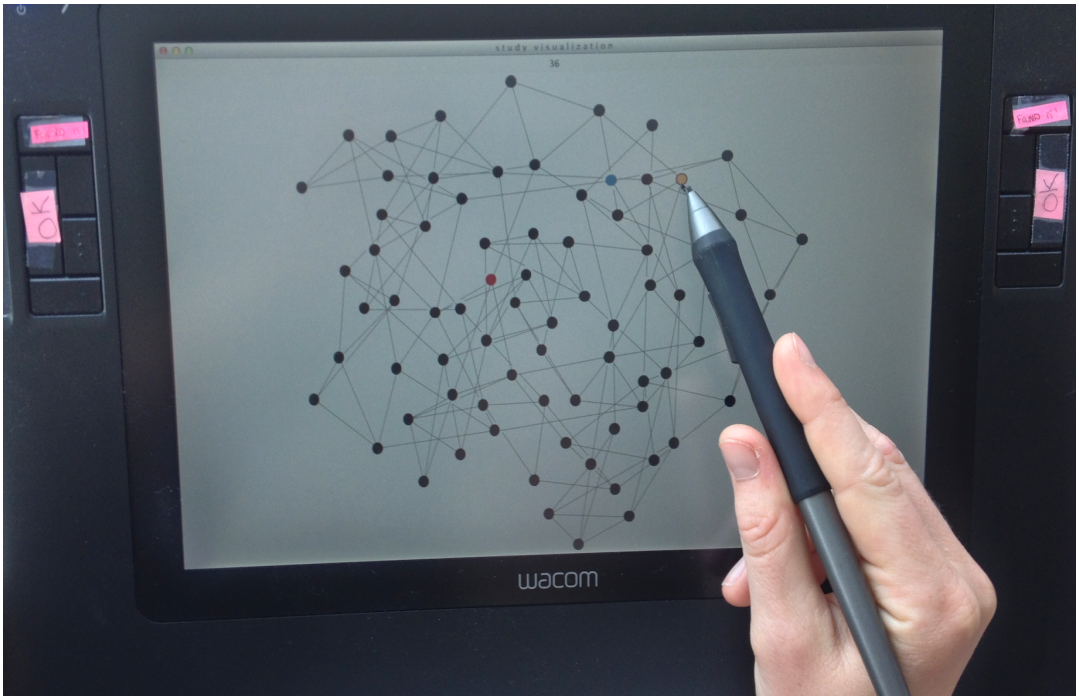

Figure 3 -Example of the user study interface. Graphs were displayed on a Wacom Cintiq tablet screen, and participants hovered over nodes with the pen to demonstrate their search process as they completed the study task. The FOUND IT! and OK buttons are indicated with the pink labels and are present on both sides of the screen.

While searching for the path, participants were asked to use the tablet pen to hover over the nodes in the paths that they considered. Nodes became highlighted when hovered over by the tablet pen. Figure 3 shows an example of a graph displayed on the tablet.

Each trial consisted of two phases. In the search phase, participants were given a maximum of 90 seconds to find the shortest path and then press a button labelled FOUND IT! located on the side of the screen. In the answer phase, participants were given 20 seconds to demonstrate the path that they had found by selecting each node in the path with the tablet pen, and then press OK to submit their answer. To select or deselect a node, participants were required to hover over it and then press a button on the side of the pen. Time remaining in each phase was displayed at the top of the screen, and the colour of the node highlighting changed depending on the phase: orange highlighting for the search phase, green highlighting for the answer phase. If participants ran out of time in the answer phase, the nodes selected at the time-out point were automatically taken as the participant's answer.

Participants were asked to limit their search for the answer to the search phase. During piloting we noted that participants sometimes realized during the answer phase that they had not actually found the correct answer, and sometimes felt pressured to keep trying to find it even though the search phase was over. To address this issue in the actual experiment, we told study participants during training that we wanted to know about such mistakes, and instructed to them to select the nodes they had originally thought made up the

answer if this occurred. Finally, each trial concluded by showing participants the correct answer to the trial, before prompting them to begin the next trial.

Dataset and graphs

We generated 144 graphs for use in the user study and subsequent analysis. We also generated an additional 9 practice graphs, which were only used for practice by participants and were not included in later analysis.

Sample size: The sample size of 144 graphs was deliberately chosen to provide enough graphs to create two discrete subsets, a training set (24 graphs) and a validation set (120 graphs). Set sizes were determined by a power analysis (described later). The size of each set was determined by the needs of our planned analysis evaluation, as well as a maximum number of trials that we could expect participants to complete in a single session. These sets are used in two separate stages of analysis:

- a qualitative analysis of human path-tracing behaviours and development of a predictive model.
- a regression analysis that acts as an example application of the search set, and as validation of the predictive model.

This type of approach, using a training set and a disjoint validation set, is commonly used in the machine learning communities for model selection and validation[53], and was

intended to support testing whether or not the model derived from the training set generalized to the validation set.

Graph generation: To support both reproducibility and analysis, we generated the graphs and layouts in advance of the experiment. We used the Watts-Strogatz model[54] to create graphs with small-world properties, following the argument of Auber et al. and others that these represent realistic models of networks from many application domains.[55] The Watts-Strogatz algorithm parameters were tuned during pre-piloting experimentation; we used degree-4 edges in the initial circle lattice, and a 15% probability of random reattachment. We selected a graph size of 75 nodes and 150 edges as the best balance between density and difficulty from those tested in piloting, as discussed above; this edge density ratio of 2 falls well within the limits discussed by Melançon for synthesis of realistic graphs.[56]

We then laid out the graphs by running the force-directed placement included in the Prefuse toolkit for 5 seconds to lay out each graph, and saved only graphs with an aspect ratio of 0.8–1.12 to ensure that nodes would appear at a similar size on the screen. We use a layout with straight edges because this representation is by far the most common in real world applications.

A unique shortest path was selected for each graph. To generate the paths, each of the graphs was randomly assigned a source node, and then breadth-first search was performed

to assign a goal node to create a single shortest path of 2, 3 or 4 hops. An equal number of graphs for each of these solution-path lengths were generated. The coordinates of the pre-generated laid-out graphs, along with the assigned solutions, were stored as XML files for later use.

Interface

The Cintiq tablet was inclined to a slight angle, about 25 degrees from the horizontal. The OK and FOUND IT! input buttons used in the task appeared on both sides of the screen, to support both left handed and right handed users. These buttons were configured to accept input only during the relevant stages of the trial to reduce the incidence of mistakes.

Apparatus

The experiment was conducted on a Wacom Cintiq 12WX direct input pen tablet, which featured a 12” screen, and was connected to a 13" 2.7 GHz Intel Core i7 Mac Book Pro with 8 GB of RAM and Mac OS X Lion 10.7.2. The experiment software was coded in Java using the Prefuse toolkit.[57]

For each trial the system recorded a log of the participants’ pen movements, the graph nodes that had been hovered over (computed as any intersection between the cursor position and the node geometry), the task completion time, and the final answer.

Procedure

The total experiment length was over 2 hours and thus was split across two sessions to avoid participant fatigue. The first session took between 1-1.5 hours, and the second session took ~1 hour. Participants were able to complete the experiment on the same day, but were required to wait a minimum of one hour between sessions.

In the first session, participants were asked to confirm at this time they had normal or corrected-to-normal vision and regular colour vision, and then completed a brief questionnaire on their background. The experimenter demonstrated the tablet and the task, and then walked the participant through a series of steps to configure the tablet. The tablet was configured to use the participant's dominant hand. Participants then completed the built-in calibration utility until both the experimenter and participant were satisfied with the pen tip cursor alignment. When returning for the second session, participants repeated the tablet configuration and were reminded of all instructions. Before starting experimental trials, participants completed an equal number of practice trials of each possible solution-path length – 6 practice trials in the first session (two of each length), 3 in the second session (one of each length) – and the experimenter provided feedback to ensure they understood the task.

For each trial, participants completed the task with one of the 144 pre-generated graphs. The presentation order of the graphs was randomized across both sessions, while the practice graphs were shown in the same order. Participants completed 6 blocks of 12 trials at a time, for a total of 72 trials per session. Between each block, participants were required to take a one-minute break. Each session contained an equal number of graphs with each possible solution-path length – 24 each of the 2, 3 and 4-hop graphs – but these were not controlled for within blocks. The use of blocks in the experiment was only to ensure that participants took consistent breaks.

After each session the participants rated the task on a Likert scale from 1-low to 7-high according to the overall difficulty, and the mental and physical effort required. A post-experiment interview followed the second session. The Likert scale data did not reveal any interesting trends and thus did not factor into our analyses; we do not report on it any further in this article.

Qualitative analysis of path tracing behaviours

The focus of the first part of our analysis for the user study pertains to our first two questions concerning the search set concept: (Q1) can we identify distinct path-tracing behaviours? and (Q2) how common are these path-tracing behaviours?

We began with a preliminary analysis of the nodes hovered over by participants during the study trials, and visualized that data to explore what each participant's search set looked like. This early exploration motivated the central qualitative analysis described below. First, we manually identified and described paths from the hovered-over nodes in a subset of the study trials. From that analysis, we characterized a number of common path-tracing behaviours. Once we were able to describe how participants traced paths, we could then develop a predictive behaviour model of the search set.

Preliminary node-based analysis of search set

We began with a preliminary analysis of the data collected, focusing on the overlap between the total set of nodes that each participant hovered over at least once during the trial for each of the 144 graphs.

The success rate for the user study trials was low. On average, participants successfully completed 58.7% of the trials (*SD* = 11.7%, *min* = 34.7%, *max* = 79.2%). Not surprisingly, the success rate decreased substantially as the length of the path became longer. The 2-hop paths had a 76.2% success rate (*SD* = 8.4%, *min* = 62.5%, *max* = 91.7%), the 3-hop paths had a 60.8% success rate (*SD* = 13.1%, *min* = 35.4%, *max* = 79.2%), and the 4-hop paths had a 32.8% success rate (*SD* = 17.3%, *min* = 6.3%, *max* = 68.8%). We felt that this level

was appropriate for our study given our desire to analyze both successful and unsuccessful attempts, and represented a diverse range of cases from easy to hard.

On average, only 6.1% of the nodes for a given graph that were hovered over by at least one participant were also hovered over by the 11 other participants (*min* = 0%, *max* = 25%), and these often included the nodes on the correct path for the trial. While we had expected to see some individual variability, we were nevertheless surprised by the extent of this apparent lack of overall commonality given the previous work on geodesic tendency, and so we chose to dig deeper to into the question of what behaviours dictated participants' search patterns.

Visualization of node hover overlap. We generated a number of visualizations of the node hover data to further explore how the overlap varied. In this section we present views from one of the visualizations that we created. Additional details about these views, and all the visualizations developed for this analysis, can be found in the Supplementary Material.

The visualization we discuss here shows a graph and all of the nodes that were hovered over by participants for the corresponding experimental trials. The visualization included twelve small multiple views, each of which displayed the nodes hovered over by a single participant in the trial. The graph for a trial was laid out as in the experiment, with the

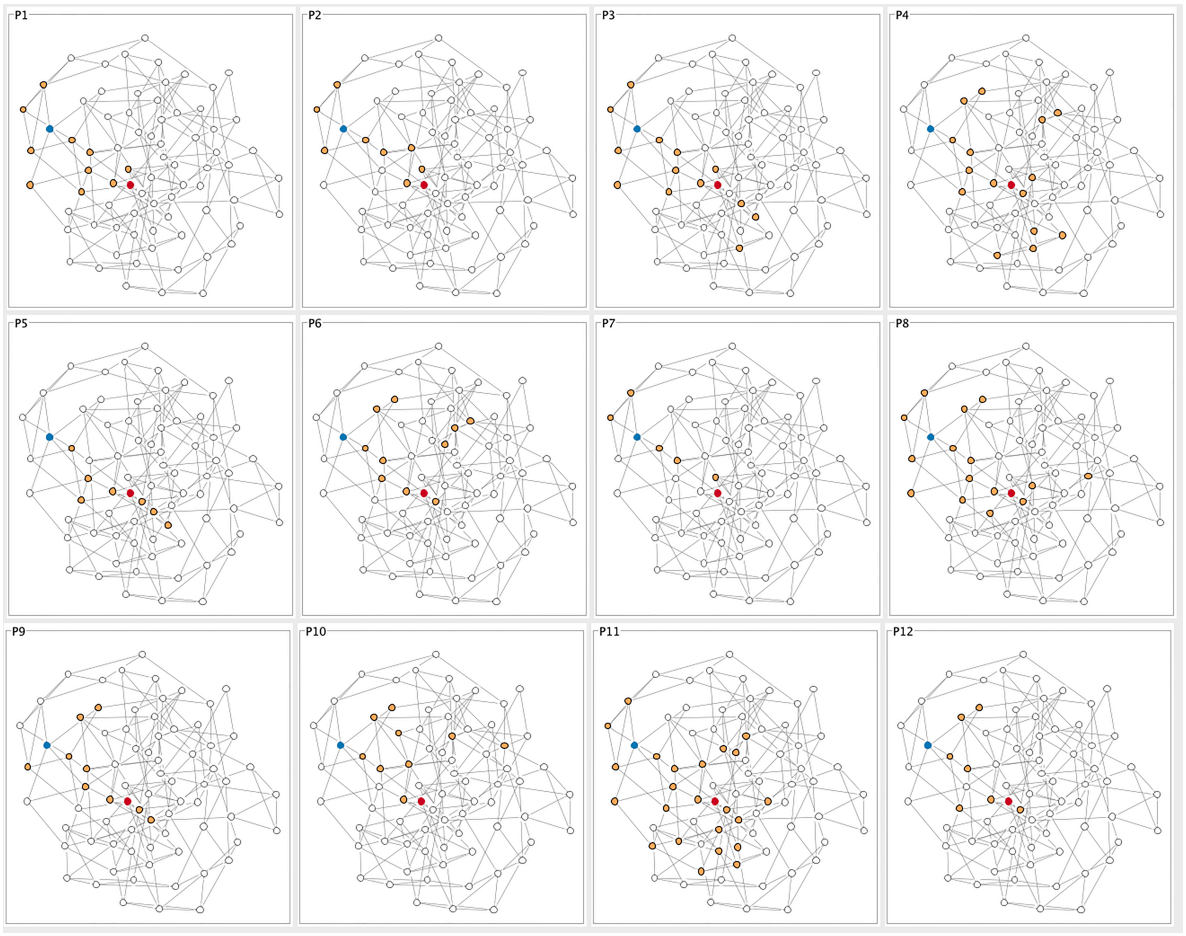

Figure 4 - Example of small-multiple visualizations of all the hovered-over nodes for one graph trial. There is one small multiple per participant, labelled by the participant number in the top left. Hovered nodes are coloured in orange, with the remaining nodes shown in white. The source and goal nodes are coloured red and blue, respectively; hovers on these nodes are not shown.

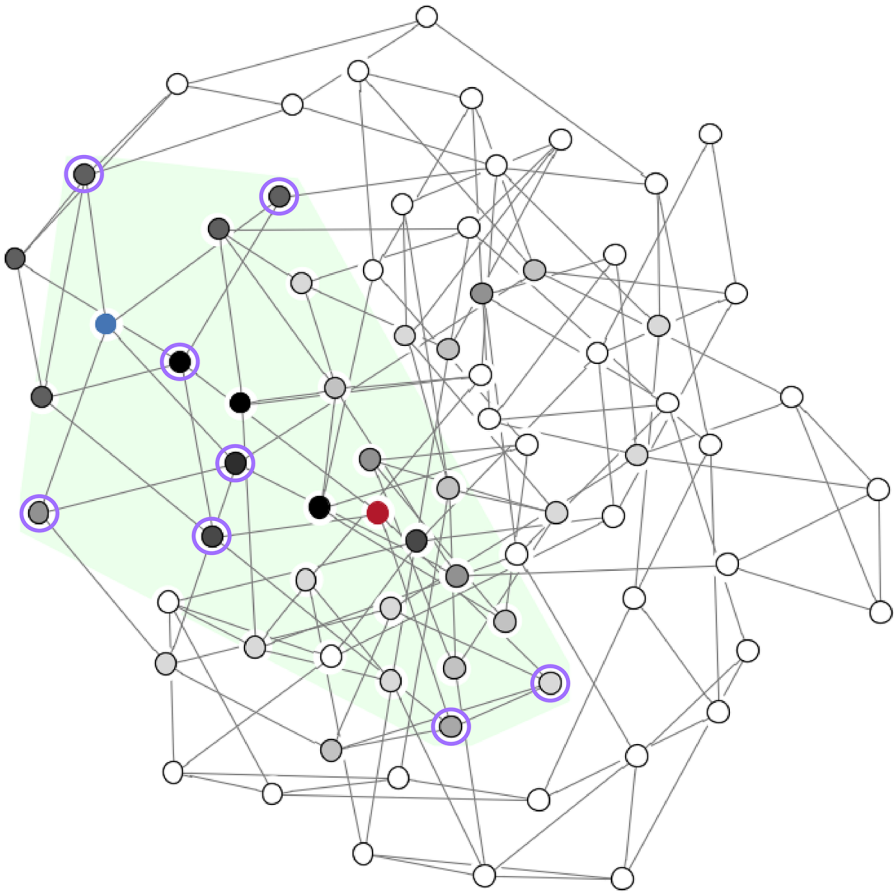

Figure 5 - Example of an aggregate visualization showing all nodes hovered over by all participants on one graph trial. Node colour changes from light grey to dark as the frequency with which the node was hovered increases. The source and goal nodes are coloured red and blue, respectively. The convex hull of the source node and goal node and their 1-hop neighbours (annotated with purple circles) are shaded in light green.

nodes in each small multiple coloured according to whether or not they had been hovered over by that particular participant, as shown in Figure 4. A different view, shown in Figure 5, aggregated the hovers of all participants onto a single view of the graph, with the frequency of hovers encoded in grey-scale

By examining these visualizations we noticed that subsets of the participants' hovered-over nodes would often overlap, even though the total overlap across all participants was small. When we incorporated the frequency with which each node was visited, we saw that the most frequently hovered-over nodes tended to fall in a convex hull around the red and blue nodes and their respective 1-hop neighbours, as shown in Figure 5. Three participants alluded to this convex hull behaviour in the post-experiment interview, stating for example that they "often tried to look in the area between the red and blue nodes" (P11). On average, 93% of the total node hovers for a given graph fell inside the convex hull (*min* = 73.1% *max* = 100%). This consistency suggested to us that although the participants' approaches were not identical, there were in fact some similarities in how participants were tracing paths.

Qualitative analysis method

Motivated by the findings in the preliminary node-based analysis, we moved from considering the data simply in terms of node hovers to reconstructing the paths searched by

the participants. By looking at the progression of paths over time, we hoped to characterize common human path-tracing behaviours.

Given the variability we observed in the node-based analysis, we did not feel that we had a deep enough understanding to extract the paths through computation alone. Instead, we chose to manually extract paths from the node hover data using qualitative coding after applying some algorithmic filtering. The data was then manually transformed from *hovers* to *steps*, which were then coded as *paths*. One investigator performed all of this qualitative analysis.

Data sample. The training set was made up of 8 graphs for each of the 3 possible hop lengths, for a total of 24 graphs. These were selected randomly from the total set of 144 graphs. We analyzed all 12 participant trials from the user study for each of these graphs, for a total of 288 trials. We reserved the larger validation set (of the remaining 120 graphs) for a hierarchical regression analysis that served as a validation of our predictive behavioural model, and which we discuss in a later section.

Data preparation and visualization. The raw data in the log files were *hovers* over a node, as described earlier. Some of these hovers were deemed to be spurious and were eliminated from further consideration. Some were automatically filtered out based on a quantitative threshold, while others were discarded as a result of the qualitative analysis process

described below. In the automatic filtering, hovers lasting less than 5ms were discarded. This threshold was derived from a combination of quantitative analysis and observation while building the visualization; we found that less than 5ms was an unrealistically short length of time to hover over a node while actually tracing a path. Most discarded hovers seemed to be caused inadvertently when participants transitioned their search from one area of a graph to another.

After the initial automatic filtering, we manually transformed the hover data into a sequence of *steps*. Initially a step was created for each individual node hover, in temporal order. From these steps, the investigator could then compose a *path*: a complete sequence of nodes that constituted an intended single path-tracing attempt on the part of the participant. In order to assist the investigator in identifying topological paths, the automatic filtering consolidated two or more successive node hovers into the same step when they were connected by edges.

Despite this automatic process, a path could still be split across multiple steps for several different reasons. First, some paths consisted of a combination of topological and apparent connections between nodes. We saw many examples where participants followed *apparent* paths that were not true topological connections (there was no edge in the graph between consecutive hovers), but these were mistaken for a true topological connection

because of node-edge crossings. Second, some paths were split across multiple steps because of spurious hovers in the log that the investigator judged to be incidental to what the participant was actually considering at the time. These were typically nodes crossing or near to a path that the participant followed repeatedly, or nodes hovered over during transitions between different parts of the graph.

To support our analysis, we designed a visualization in which the first 20 steps were directly visible as small-multiple views of the trial graph with some nodes and edges coloured to show hover activity. Figures 6 and 7 show an example of the small multiples used in the visualization. The first node in an automatically determined topological sequence was coloured light orange, with subsequent nodes coloured dark orange, and edges along the topological path between them also coloured orange. As additional visual support for the analysis, we included an aggregate view similar to the one used in the preliminary analysis (Figure 5), and when the investigator hovered over a node, it was highlighted in every small-multiple view and its node ID was shown in a tooltip. An example of the entire visualization can be found in the Supplementary Material. We chose to stop analysis after a maximum of 20 steps because our initial exploration showed that, just as we observed in piloting, later steps tended to be more chaotic and less representative of common behaviours.

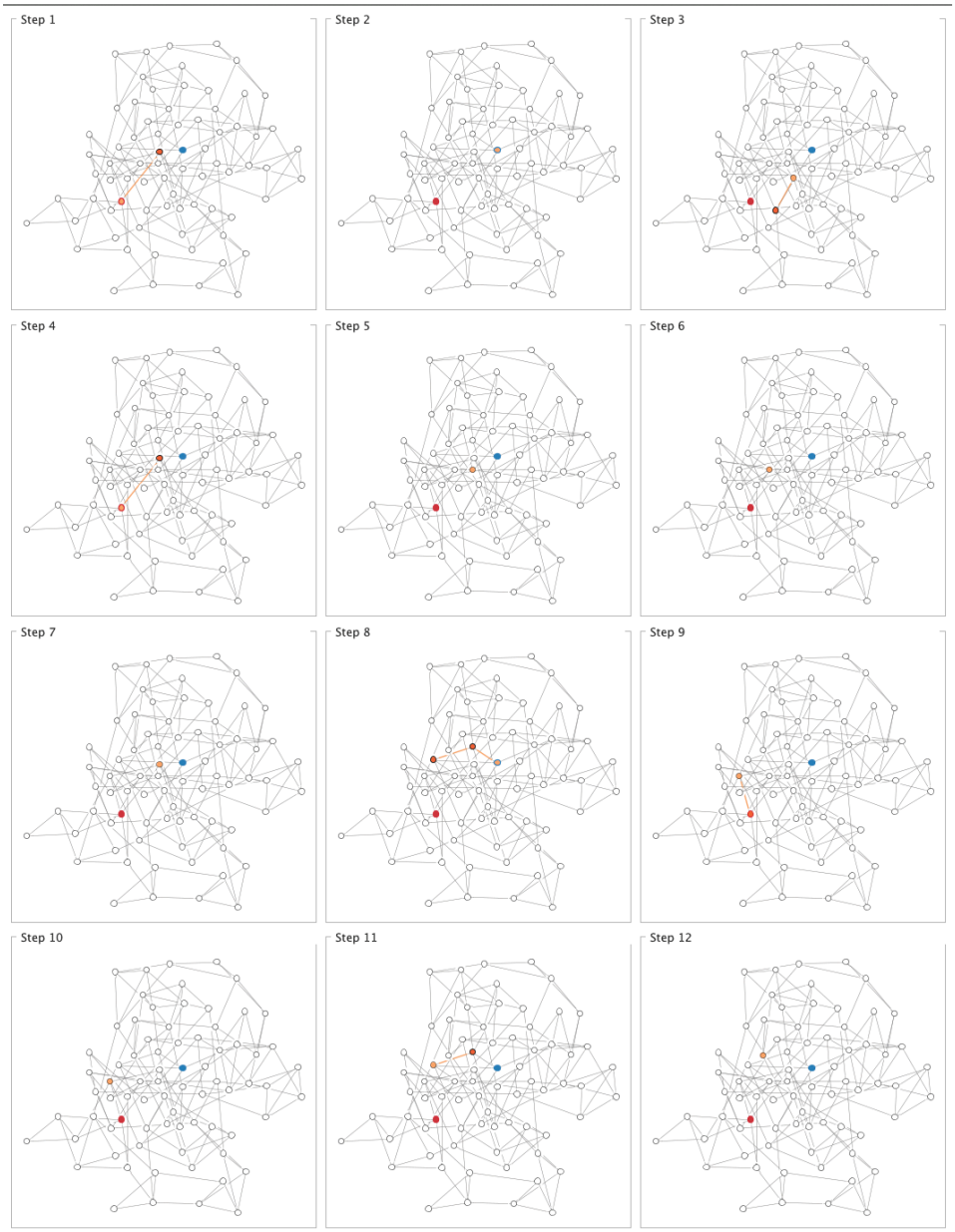

Figure 6 - Example of the small-multiple visualization of discrete steps used to support the qualitative coding process; each step is labelled in the top left. The first node in an automatically determined sequence is coloured light orange and subsequent nodes coloured dark orange; edges along the topological path between them are also coloured orange.

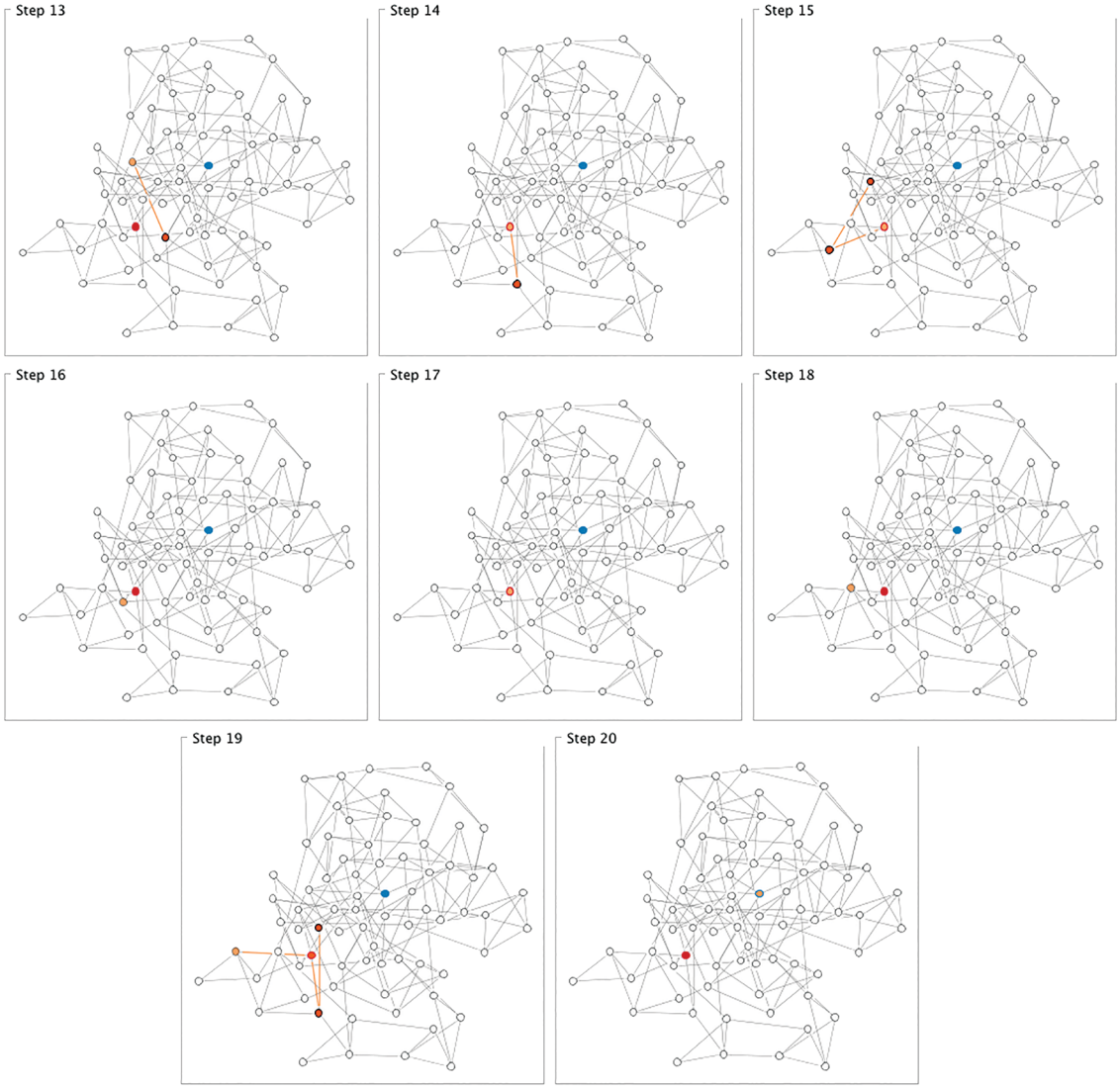

Figure 7 - Example of the small-multiple visualization of discrete steps used to support the qualitative coding process; each step is labelled in the top left. The first node in an automatically determined sequence is coloured light orange and subsequent nodes coloured dark orange; edges along the topological path between them are also coloured orange.

Coding process. All of the steps in a trial, up to the maximum of 20 visualized, were described with at least one code. The paths identified by the investigator were coded as a sequence of node IDs, in addition to a number of other attributes, which we describe next.

First, a path could be either a true *topological* path or an *apparent* path. Second, the investigator coded the *target* node that the path was going towards, which could be the *source* node (red), the *goal* node (blue), or some *other* node in the graph. Third, the *anchor* node that the path started from was identified, which again could be the source node (red), goal node (blue), or some other node in the graph.

The final two attributes for a path were used to describe the branches that the participant followed for each hop of that path. One was the *direction* (forward, right angle, or backward) with respect to the target, of each branch in a path. The investigator used her approximate judgement rather than exact angles in determining whether a branch went towards the target, at a right angle from it, or away from it (i.e., whether the branch went closer to the target, kept roughly same distance from it, or went away from it). The last attribute was whether the branch at a particular hop was the *closest-to-geodesic* branch from the associated node to the current target. We did not expect participants to be skilled at judging very small differences in angles, and observed this to be the case in early exploration of the data. Thus, when the difference between two branches on either side of

the geodesic straight line to the target was very small, or if those branches overlapped, the investigator recorded both as having the closest-to-geodesic property.

In addition to describing paths, the investigator generated codes for other types of movements by participants that emerged during the coding process as being potentially important. These were *jumps* between nodes, *switches* of a *target* and/or *anchor*, *checks* of nodes or node-edge crossings, and *doublebacks* over paths just traced. The investigator used the same attributes described above in these other codes as appropriate.

Finally, the investigator also coded *incidental* node hovers. These were hovers over nodes that occurred between two nodes in a coded path or during some other movement, but were judged to not signify the node a participant was actually considering at the time.

Example of a coded trial. To illustrate the coding process, we next walk through one example of the codes and the attributes used to code one participant trial. Figure 8 and Figure 9 demonstrate the paths identified from all of the steps in the example. Where a path spanned multiple steps, we show these steps collapsed into a single image; the red and blue nodes are labelled R and B respectively. The 20 steps coded for this trial are also shown without annotation in Figure 6 and Figure 7, as the investigator saw them during the coding process.

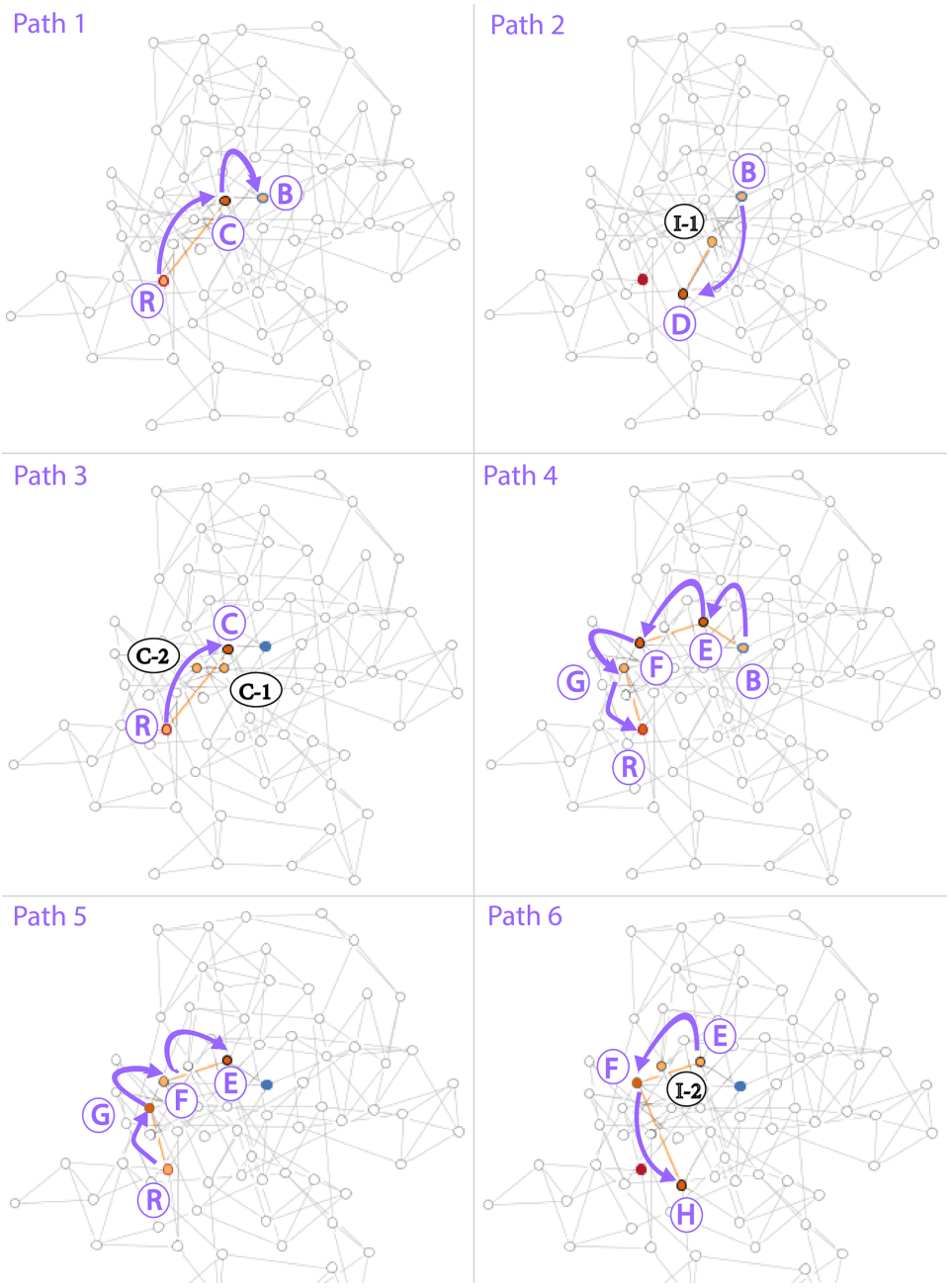

Figure 8 - Paths 1–6 extracted from steps 8–13 and collapsed into single images. Steps 8–12 are shown in Figure 6, while step 13 can be found in Figure 7.

Figure 8 shows the first six paths. In PATH 1 (R-C-B, steps 1 – 2, anchor = red, target = blue), the participant follows two hops in the closest-to-geodesic direction; the hop from R-C is a true topological path, but the second hop is apparent, because no edge exists between C-B. In PATH 2 (B-D, steps 2 – 3, anchor = blue, target = red) the participant follows one hop in the closest-to-geodesic direction. The node I1 is coded as incidental even though it connects to D with an edge, because B is also connected to D. In PATH 3 (R-C, steps 4 – 7, anchor = red, target = blue) the participant again follows one hop along the closest-to-geodesic branch, repeating part of PATH 1. C1 and C2 from steps 5 and 6, respectively, are examples of checks around node C, to which the participant returns in step 7.

PATH 4 (B-E-F-G-R, steps 8 – 9, anchor = blue, target = red) is another example of an apparent path, because there is no edge between F-G. The first hop B-E goes in a right angle direction. The remaining hops all take the closest-to-geodesic branch. PATH 5 (R-G-F-E, steps 9 – 11, anchor = red, target = blue) is a doubleback of the previous path, PATH 4. In PATH 6 (E-F-H, steps 11 – 13, anchor = E, target = red) the participant again retraces part of PATH 4, but deviates to follow a true topological connection between F-H, which is in the closest-to-geodesic direction. Step 12 shows another incidental hover, I2; it seems clear that the participant is following the E-F edge, and thus would probably not think I2 is connected to either E or F.

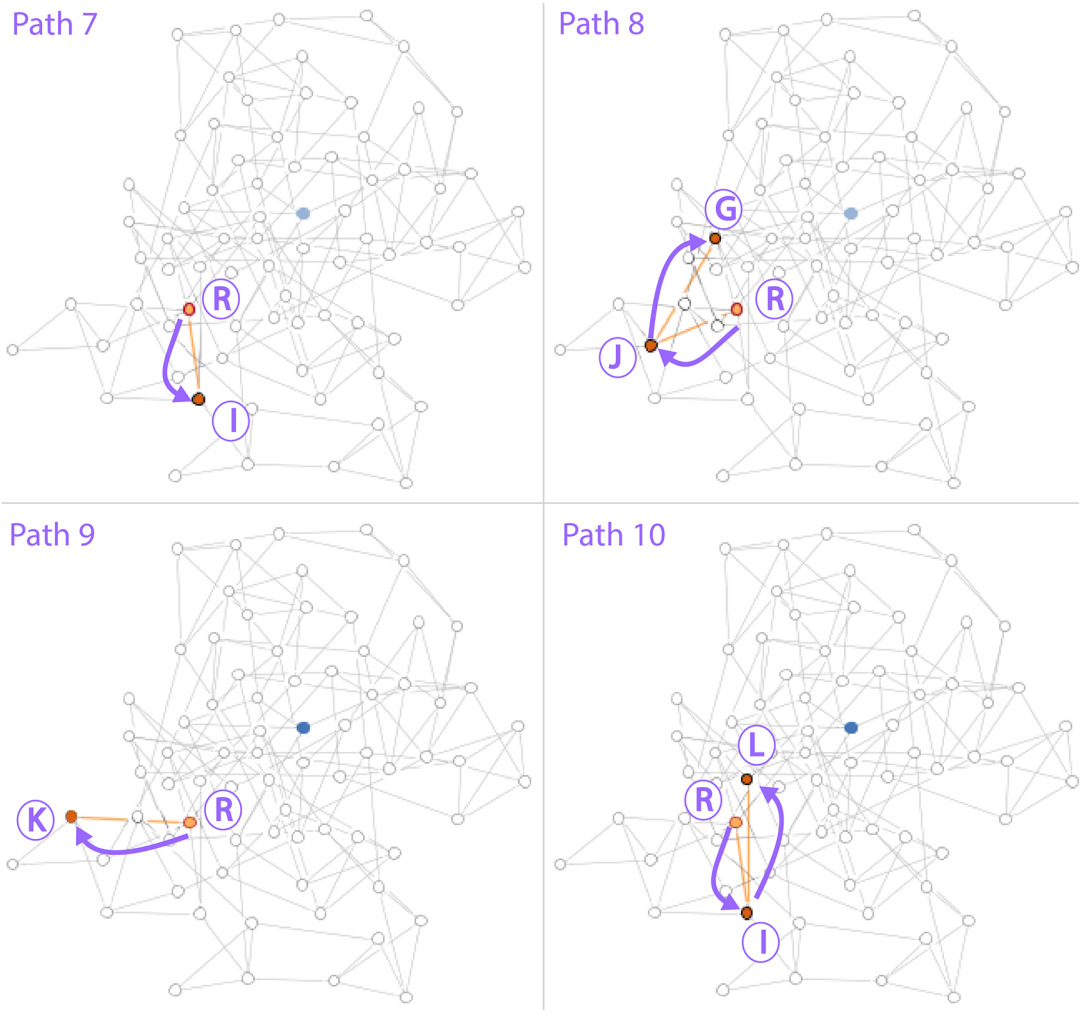

Figure 9 - Paths 7–10 extracted from steps 14–19, shown in Figure 7. Where a path spanned multiple steps, it has been collapsed into a single image.

PATHS 7 – 10 are shown in Figure 9. In PATH 7 (R-I, step 14, anchor = red, target = blue), the participant follows one hop in the backward direction. In PATH 8 (R-J-G, step 15, anchor = red, anchor = blue), the participant follows a different branch in the backward direction for the first hop, and then follows the closest-to-geodesic branch for the second hop. Between PATHS 8 and 9, another incidental hover occurs in step 16, which is not shown. In PATH 9 (R-K, steps 17 – 19, anchor = red, target = blue), the participant follows another branch in the backward direction. Finally, in PATH 10 (R-I-L, step 19, anchor = red, target = blue), the participant follows the same hop as in PATH 7 before following the closest-to-geodesic branch for the second hop. Between each of the paths from PATH 1 to PATH 6, we observe *switching,* whereas for PATHS 7 – 10, the participant continues to search around the red source node.

Final coded data set. We eliminated 11 of the 288 trials during the coding process because participants entered the answer phase of the trial without hovering over any nodes in the search phase. The investigator ultimately classified 95.8% of the steps in the remaining 277 trials with at least one code.

The remaining 4.2% of steps could not be made sense of in the context of our coding scheme, and were coded as *unclassified*. We suspect that some of these unclassified trials were caused by incomplete data, for example, if a participant missed a node with the pen tip

despite looking at it. We had anticipated this limitation of the tablet, but decided that this number was small enough to be acceptable. In addition, some of the unclassified steps may have been deliberate but uncommon types of movements that we simply did not see often enough to classify with a unique code.

Results

We now describe the behaviours that emerged during the coding process, and from subsequent analysis of the final code set. Some of these findings stem from differences between specific graphs and the common cases we observed, whereas others hold across all graphs.

Choice and use of anchors for searching. Although participants were instructed to search from the red node to the blue node, we found that they often searched from blue to red, especially when the task was more difficult. On average, the majority of paths coded across the 24 training graphs used either red or blue as the starting or anchor node (M = 86.9%, SD = 10.4%, min = 64.8%, max = 100.0%). We had expected that participants would also frequently use intermediate nodes that were part of the way along promising paths as anchors, for example, following one or two promising hops, and then choosing a node to search out from. However, we were surprised to find that this behaviour was not very common. Instead, we observed that participants were much more likely to give up

following a path and restart from a red or blue node, even if this meant immediately retracing the path they had just followed. The extracted paths in Figure 8 demonstrate this behaviour; the participant switches anchor and target before beginning each path from PATHS 1 – 6, searching back and forth in alternating directions between red and blue. The participant only uses an intermediate node once as an anchor, in PATH 6 (node E).

Prevalence of the closest-to-geodesic tendency. Participants preferentially followed paths along nodes forming closest-to-the-geodesic branches, suggesting strong evidence of a geodesic tendency. On average, the majority of the identified paths for a given graph (M = 65.5%, SD = 9.7%, min = 49.0%, max = 81.3%) fell along the closest-to-geodesic branches either for all hops (M = 39.4% SD = 10.7%, min = 15.8%, max = 56.3%), or for all but the first or last hop in the path (M = 26.2%, SD = 8.2%, min = 15.8%, max = 47.4%). In the post-experiment interviews, eight participants explicitly described strategies involving the closest-to-geodesic path.

We also examined how common it was for the very first branch followed in a trial to be the closest-to-geodesic branch for the starting anchor. The majority of participants began trials with the closest-to-geodesic branch (M = 60.3%, SD = 25.2%, min = 16.7%, max = 91.6%), which again points to the strength of the geodesic tendency. However, for six of the graphs in the training set, this number was well below 50%, and as low as 16.7%, which

suggests that other factors occasionally override this tendency or impact its strength. In particular, we noted that as the angle between the closest-to-geodesic branch and the straight line to the target increased to 90° or larger, it became more likely that the participant would pick a different branch. This observation suggests that the tendency decreases in strength the further the closest-to-geodesic branch is from the actual straight line to the target; that branch may diverge significantly. Figure 10 shows an example.

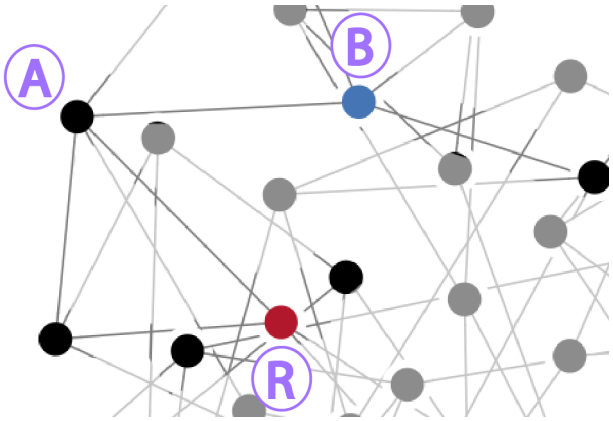

Figure 10 - Example of a portion of a graph from the study, with a 2-hop solution, where most participants did not follow the closest-to-geodesic branch in either direction (B–A or R–A) for their first hop at the beginning of the trial (the red and blue nodes are labelled R and B, respectively). We attribute this divergence from the geodesic tendency to the large angle size approaching 90°.

For this graph, only three participants started by following the closest-to-geodesic branch from red or blue, which in both cases went to A. Another interfering factor we observed was the length of the closest-to-geodesic branch with respect to the target

distance; if the target was far away and the closest-to-geodesic branch was much shorter than surrounding branches, or if the target was very close and the closest-to-geodesic branch went past the target, then the closest-to-geodesic branch seemed less likely to be searched at all.

Likely directions of search. Despite the prevalence of the geodesic tendency, participants did spend considerable time searching along other branches. Typically, the likelihood of expanding to nodes that were not along the closest-to-geodesic branch increased with the amount of time a participant spent on a trial. We saw the largest divergence from the geodesic tendency for the first hop of paths emanating from red or blue. However, participants were likely to return to the closest-to-geodesic branch for subsequent hops.

Our analysis did not suggest that there was a fully continuous ordering of the likelihood of searching in a particular direction for the first hop. For example, branches did not simply become decreasingly likely as the size of the angle with the geodesic straight line to the target increased in a directly continuous way. However, the order was also far from random: we observed similar likelihoods within discrete groups of branch directions.

As shown in Figure 11, we loosely grouped directions of the first hop into four ordered groups more specific than those we used in coding - we did not strictly define these groups in terms of exact angles. The first is *directly towards*, meaning a small angle with respect to

a line straight towards the target; next is *towards*, up to slightly beyond a right angle. We noted in our analysis of the coded data that when the angle of the first hop was just larger than a right angle, the likelihood was still similar other hops that were more clearly going toward the target, thus this definition. The third group is *away*, for even larger angles; the last and least likely category is *directly away*, for angles that were essentially in the opposite direction from the target. Participants had a roughly similar likelihood of selecting a branch within each group, starting with *directly toward*, and earlier groups were more likely to be searched to exhaustion before later ones were begun.

This grouping of branches into likely groups also extends to intermediate nodes along

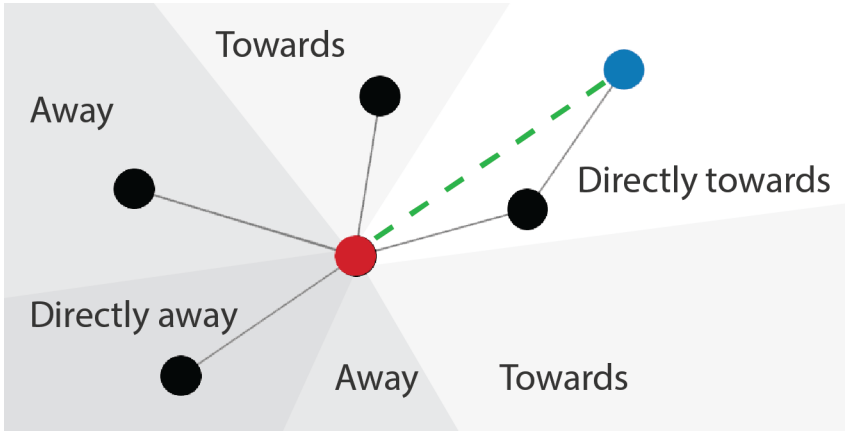

Figure 11 - Illustration of the ordered groups of similarly likely candidates for the first hop, coloured in greyscale in decreasing order of likelihood and named by their directionality with respect to the target: directly towards, towards, away and directly away.

promising paths, but the range of angles describing similarly likely branches at such nodes was much larger. The second hop in a path was more likely to go towards or directly towards the target than away, and paths where users went two subsequent hops away from the target were uncommon.

It appeared that participants tended to exhaust the options around red and blue before exhausting the options around nodes two or three hops along a path. This phenomenon partially explains our observation that participants tended to return to closest-to-geodesic candidates for subsequent hops in paths. It also provides some explanation for the relationship between the angle of the closest-to-geodesic branch and the likelihood that it would be followed first in a trial. When the closest-to-geodesic branch goes directly towards the target, it becomes very likely it will be followed first. But as the angle increases to 90° or larger, it becomes more likely that the participant would follow any other branch in the same group. We suspect that in these instances, other factors, such as path straightness, begin to take precedence.

Use of apparent and topological paths. Participants primarily followed topologically connected paths, but apparent paths created by node-edge crossings did sometimes cause significant distraction. Despite the fact that all users were trained to use the halos to identify node-edge crossings, some reported that it required extra effort to realize that they were

looking at a crossing. Such paths were a common source of error, especially when they lay on top of a branch directly connected to the red or blue nodes. We see examples of this in PATHS 1 and 3, and PATHS 4 and 5 in Figure 8. The participant examines node C in both PATHS 1 and 3, even though it forms an apparent path, presumably because it seems so promising. Similarly in PATHS 4 and 5, the participant repeatedly follows the apparent path between nodes G and F, taking considerable time to determine that there is a node-edge crossing before trying a different route from F in PATH 6.

Revisitations. We observed that participants often revisited the same path again and again. This repetitive behaviour took two forms. We saw many instances of *doublebacks,* where participants would retrace a path one or more times immediately after tracing it the first time. We also saw that participants would return to a path after tracing others, even if they had followed it multiple times before. This finding is not surprising in light of the known limits of working memory for remembering the results of previous searches [27]. P2 admitted that he would often "look at a path more times than was helpful." Some participants also related this behaviour to the tendency to search within the convex hull and along closest-to-geodesic path. P6 explained that "I would try to counteract and look for different paths, but the [closest-to-geodesic path] was more natural, and it was harder to force myself to look away."

Path stopping conditions. Contrary to what might be expected, participants often did not follow every path that they started until they reached the maximum length possible in the study (4-hops). However, we did observe some commonalities in when participants were likely to stop following a path. Some stopping conditions were largely dictated by the experimental tasks and common sense: participants typically stopped a path when the number of hops equaled the maximum of four, they had cycled to reach a node already in the current path, or they had reached the target. Other stopping conditions were less obvious. We found that participants actually tended to stop when the number of hops was one less than the maximum path length in the task. We also saw that they frequently stopped when their current path took them past, or nearly past, the target with respect to the starting anchor. We defined *past the target* generally as the line through the current target that was at a right angle to the geodesic straight line between the path anchor and the current target as illustrated in Figure 12 – Left, where the geodesic is the dashed green line and the perpendicular through the blue target is the solid green line; a user tracing from red (R) to blue (B) would likely trace R-C-D, stopping at D, which is just past the blue node, and not get to E. Consistent with other observations we have described, we note that this definition is not exact, but is dependent on participants ability to judge angles. In this case,

we noted that when the current node was very far away from the target, the *past the target* condition seemed to be met at an angle narrower than 90°.

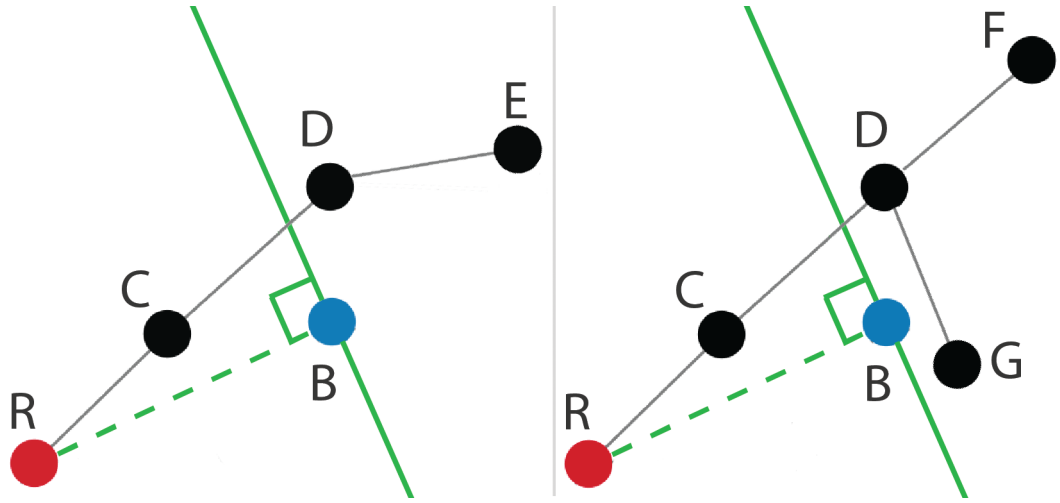

Figure 12 - Example of the past the target stopping condition. (a) A user would typically stop a path at the first node past the target with respect to the starting anchor, where past the target is the line through the current target that was perpendicular (solid green line) to the geodesic straight line between anchor and target (dashed green line). A user tracing from red (R) to blue (B) would likely trace R–C–D and stop without going to E. (b) Exceptions to this condition (and of the stopping condition of maximum hops minus 1) were when the next hop went directly towards the target (D–G) or in a straight line from the previous hop (D–F).

There were two exceptions that we sometimes observed to the final two conditions of stopping at the maximum less one or going past the target. Figure 12 – Right illustrates an example of these exceptions for the past the target condition. A participant tracing from R to B along the path R-C-D would be less likely to stop at D as in the previous example if:

(i) the next hop formed a nearly straight line with the previous hop (as in D-F), (ii) or the next hop went directly towards the target (as in D-G). We suspect that these exceptions occurred for different reasons. In the case of the first exception (i), we suspect that the close to straight line created a continuous path that encouraged users to go straight from C to F, with less consideration of D. In the case of the second exception (ii), we suspect that participants were relying on peripheral vision to determine that a suitable candidate was not present, and only considered the path promising enough to keep following if the next hop went in the direction of the target.

Continuity and geodesic tendency. In previous work, Ware et al. [23] found continuity, namely the straightness of the path, to be a very important factor. Huang et al.[7] avoided variation in continuity in order to avoid confounding their results on the geodesic tendency, and conjectured that geodesic tendency takes precedence over path continuity. We often observed that continuity can take precedence over the geodesic tendency, refuting their conjecture; however, we also saw examples of precedence in the other direction. In fact, interaction between path continuity and geodesic tendency is quite complex, and cannot simply be reduced to one of these factors taking precedence over the other

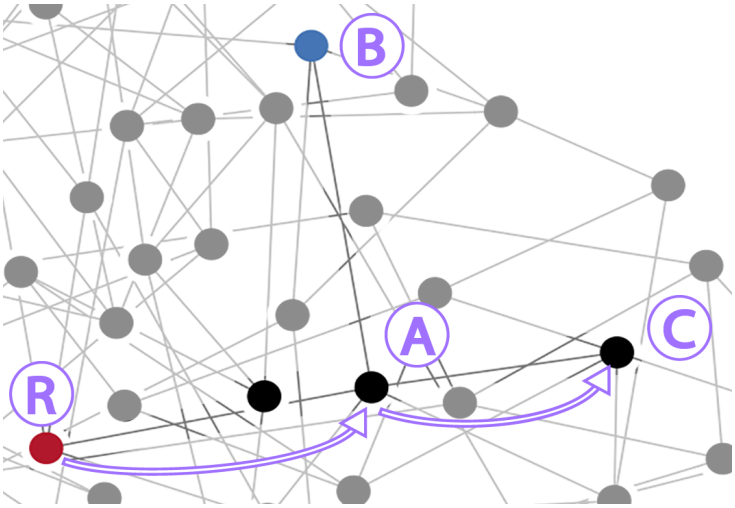

Figure 13 - Example in which a straight line appeared to interfere with geodesic tendency (the red and blue nodes labelled R and B, respectively). Some participants followed R–A–C and missed the solution path of R–A–B.

We found that in many instances participants would follow straight paths for more hops than they would "bendy" paths, and that straight paths could distract participants by causing them to miss a branch connecting to the solution. Figure 13 shows an example, where R-A-B was the solution. In this graph, three participants who followed the branch from R to A, next followed the branch from A to C, and missed the branch from A to the blue node (B). Only one of the three participants detected the solution in the steps that immediately followed. In such instances, we suspect that the Gestalt principle of continuity[27] sometimes contributes to participants perceiving the straight line formed by multiple nodes as a single hop, causing them to skip over interconnected nodes without

considering their branches. We suspect that this principle also contributed to the straight-line exception to some stopping conditions that we previously described.

Summary

Through the coding process described in this section, we determined that it is possible to identify distinct path-tracing behaviours, addressing Q1. Further, we were able to characterize and describe a number of common path-tracing behaviours exhibited by our participants, addressing Q2. The behaviours include the use of both topological and apparent paths, the conditions under which participants stop following paths, the likely directions for the first hop in a path, and the tendency to revisit previously followed paths. Unexpected behaviours included the strategy of frequent switches between source and goal nodes as the anchors in the search, and infrequency of using intermediate nodes as anchors. We verified the prominence of the previously proposed geodesic tendency, but found complex interactions between it and the other tendencies that we observed, including the impact of path continuity on behaviour, providing a more nuanced understanding of issues raised in previous work.[7]

All of these findings are useful in their own right as descriptions of human path-following behaviours when interacting with visual representations of graphs. They were also crucial in helping to develop a predictive behavioural model of search set, which we

present in the next section. While many of the behaviours that we observed could play out in different ways for different participants, enough commonalities exist to allow us to make informed guesses about the likely set of paths that a group of users may search.

A behavioural model to predict the search set

This section is devoted to our third research question: (Q3) can we predict the search set based on observed path-tracing behaviours? To explore this question, we developed a simple predictive model of the search set based on the strongest common behaviours that we described in the previous section. We next briefly describe the predictive model, and discuss our preliminary validation of its effectiveness in predicting the search set and as a basis for measuring factors for predicting task difficulty. We look at further validation approaches in the next section. A more detailed description of the model components, the algorithmic implementation, and parameter selection can be found in the Supplementary Material.

The search set model

Briefly, the model takes as input a network graph with a defined solution between two points, which are used as anchors to explore likely paths. The model is designed to predict the set of paths that a *group of users* would be likely to search, rather than the set of paths

that *one individual user* would use. The model output is an ordered set of discrete groups, where paths within each group are unordered and considered to be similarly likely; together, these paths compromise the search set.

The model begins by selecting a batch of likely candidate branches from each anchor to comprise the first hop in a path, and then follows the closest-to-geodesic branch between each of these candidates and the target. The search set contains one copy of each path followed. The conditions that determine when a model stops following a path are directly based on the common stopping patterns that we characterized in the previous section. Once all the candidates in a batch are eliminated, the model takes the next most likely set of candidate branches, and begins the path following process once again. The entire process stops once the solution path has been added to the search set (in either direction from either anchor), or all likely batches of candidates are exhausted and the task is judged to be too difficult to reasonably complete.

Validation of search set prediction

We ran the algorithmic implementation of our behaviour model to predict search sets for each of our 144 study graphs. The predicted search set produced by the algorithm contained, on average for each graph, 87% of all of the node hovers made by participants in the study. Conversely, on average for each graph, 86% of the predicted nodes were hovered

over at least once during the study. We consider these results to be an appropriate fit for a first attempt at developing a predictive behavioural model.

Using the search set to predict task difficulty

Finally, we conducted a preliminary exploration into whether or not factors measured on the search set would be effective predictors of path-tracing task difficulty. We selected the factor of edge-edge crossings, given its prevalence in previous work and our intention to use it in the hierarchical regression analysis described later. We measured search-set edge-edge crossings for each of the 24 training set graphs by summing the crossings on each path in the set (M = 330.4, SD = 298.7, min = 37, max = 1166) and we measured difficulty both by response time and by total errors.

We used bivariate Pearson correlations to examine the individual effect of search-set edge-edge crossings on average participant response time in seconds (M = 42.5, SD = 22.5, min = 10.6, max = 86.1), and total errors (M = 5.5, SD = 3.3, min = 0, max = 12) for the training set graphs. Response time was measured as the time the participant spent in the search phase, before pressing FOUND IT!. We found strong positive correlations of search

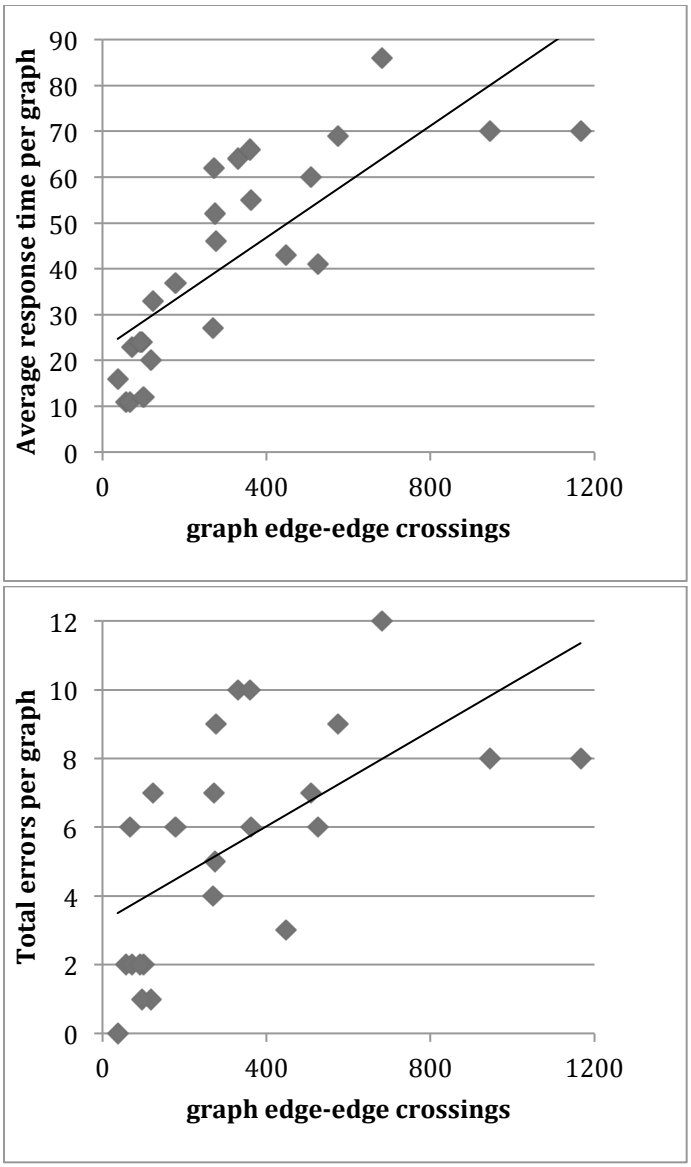

Figure 14 - Scatterplots with linear line of best fit showing relationships between search-set edge-edge crossings and our dependent variables on the training set graphs (n=24). Top: Average response time (s) per participant. Bottom: Total errors across participants.

set edge-edge crossings with response time ($r = 0.765$, $p < 0.01$), and with error ($r = 0.605$, $p < 0.01$). In general, $|r| > .10$ is considered to be a weak correlation, $|r| > .30$ a moderate correlation, and $|r| > .50$ a strong correlation.[58] Scatterplots of these relationships are shown in in Figure 14, along with the line of best fit.

Summary

With respect to our third research question, Q3, our results suggest that it is possible to accurately predict the search set for a group of users by using the human path-tracing behaviours that we characterized in the previous section. Further, our exploration into the use of the factors measured on the search set for predicting response time and total errors yielded promising results, encouraging us to perform the more in-depth validation that we present next.

Measuring graph readability factors using the search set

The focus of the last stage of our analysis is on answering our final question: (Q4) how much improvement over previous models is gained by calculating factors for graph layout on a predicted search set? This analysis was intended to serve as a validation of our predictive behavioural model as well as an example of how the search set might be used. To do this, we compared the relative importance of factors measured at three levels: the

solution path, the search set, and globally. As a part of our analysis, we also evaluated the impact of node-edge crossings, which had not been previously investigated in a user study. We present results that show a modest improvement of measuring factors on the search set over measuring on just the solution path. More crucially, we also identify important differences in the relative contributions of these factors in predicting response time and error.

Method

Our methodology follows directly from Ware et al.[23] and Huang and Huang.[8] We measured a selection of factors on different levels of the graph, which we call *predictors* in accordance with the literature on regression analysis. We use bivariate correlations to examine the individual effects of these predictors on user performance, and to determine which factors have any significant impact. We then use hierarchical multiple regression to factor out the internal relationships between the predictors, in order to examine the relative contributions of each factor in predicting performance. This approach allows us to examine the total percentage of variance in performance accounted for by the predictors, as well as any overlaps in what the predictors explain. The use of hierarchical regression follows recommendations from the literature on incremental validity[42] and on the benefit of

requiring the researcher to reason about and justify the order in which variables are entered into a regression model.

Data sample. Our sample consisted of the 120 graphs in the validation set, which were those that that remained after we removed the 24 graphs for the training set used in the earlier analysis. This number was determined through a power analysis using the following parameters: $R^2 = 0.13$, alpha = 0.05, and 9 independent variables. This analysis gave us a power level > 0.80 for 120 graphs, which is conventionally considered to be an acceptable level.[58] From this level of power we expected to be able to detect medium and large effects, where $R^2 = 0.02$ is a small effect, $R^2 = 0.13$ is a medium effect, and $R^2 = 0.26$ is a large effect.[58]

The sample consisted of an equal number of graphs with each possible solution-path length: 40 graphs each of lengths 2, 3 and 4 hops.

Dependent measures. We measured user performance on each of the 120 graphs with two dependent variables: average response time (RT) and the number of incorrect user responses (error). We chose these measures because we were interested in the impact of the predictors on both correct and incorrect answers – it is important to understand how long a user might spend only to find an incorrect answer.

Response time (RT) was recorded as the average time to complete the search phase for each trial for all 12 participants, between 0 – 90 seconds. Error was calculated as the total number of incorrect responses by participants for each graph, between 0 – 12.

Predictor variables. We selected nine different factors to measure on each of the 120 graphs in the validation set, which we used as predictor variables. A subset of our predictors were those found to be most important by Ware et al, all of which were measured on the solution path[23]: the length of the path in hops (sp-ln); the continuity of the path, calculated as the sum of the angles in degrees at each step (sp-cn); the total edge-edge crossings on the path (sp-ex); and the sum of the branches on each node on the path (sp-br).

For comparison, our analysis also looked at factors that were not measured on the solution path. We selected edge-edge crossings as a factor to measure on the search-set and global levels because edge-edge crossings are often cited as the most important metric. We measured the sum of the edge-edge crossings on the entire graph (gl-ex) and on each path on the search set (ss-ex).

We chose node-edge crossings as a second factor to compare at our three levels of interest. Node-edge crossings are widely allowed in many layout algorithms, but to our knowledge have not previously been evaluated with user studies. Our qualitative analysis results regarding apparent paths indicate that node-edge crossings might also be important

for understanding errors. We measured the sum of the node-edge crossings at each level of interest: on the solution path (sp-nx), on each path in the search set (ss-nx), and globally across the entire graph (gl-nx).

Hypotheses. Our hypotheses were as follows:

H1. *Solution path node-edge crossings (sp-nx) will account for additional variance in performance beyond other factors on the solution path.* We expected that solution-path node-edge crossings would explain variance not accounted for by the other factors of length, continuity, branches, and edge-edge crossings measured on the solution path.

H2. *Search set (*ss-*) factors will account for additional variance in performance beyond all of the solution path factors.* We expected that search-set factors would explain additional variance beyond the solution-path factors (sp-), because they account for factors on all the paths that a user might search.

H3. *Search set (ss-) factors will predict performance more efficiently than solution-path (*sp-*) factors.* The search set typically overlaps the solution path, so we suspected that search-set factors might predict more variance with fewer (or the same number) of variables.

H4. *Global factors (*gl-*) will not account for additional variance in performance beyond the solution path and search set factors*. We expected from previous work that global levels would not add additional explanation to what can be explained by the more task-relevant levels.

Our hypotheses focused on the incremental validity of factors measured on the search set, and on node-edge crossings on the solution path, neither of which had been evaluated by previous research. Although we do not make formal hypotheses about the individual effects of the factors, we expected to see some positive correlation of all factors with both dependent variables. In other words, as any of these factors increases in number for a particular graph, so should the average response time and the total number of errors made by participants. This expectation includes replicating the results of Ware et al. [23] that global edge-edge crossings, and solution-path length, continuity, branches and edge-edge crossings would be positively correlated with response time. We also expected to find significant contributions of the factors studied by Ware et al. when used in regression models.

Linear correlations for individual effect

Descriptive statistics for response time (RT) and error, as well as the predictor variables, are shown in Table 1. Upon inspection, the distributions for response time and error were

Descriptive Statistics				
	M	***SD***	***Min***	***Max***
RT	38.71	22.83	5.94	85.15
Error	4.84	3.75	0	12
gl-ex	304.59	51.66	195	434
gl-nx	71.34	13.99	35	108
ss-ex	380.28	294.99	6	1382
ss-nx	170.73	136.53	0	684
sp-ex	14.05	7.51	1	41
sp-nx	5.66	2.99	0	15
sp-ln	3.00	0.82	2	4
sp-cn	159.06	94.14	1	422
sp-br	17.00	3.90	11	26

Table 1 – Descriptive statistics for predictors and for the dependent variables of response time (RT) and error for the test set graphs (n=120). Predictors are grouped by level of measurement; those that our study is the first to evaluate are shaded.

Pearson Correlation Coefficients (*r*)									
	gl-ex	gl-nx	ss-ex	ss-nx	sp-ex	sp-nx	sp-ln	sp-cn	sp-br
RT	-.046	-.092	.772*	.721*	.528*	.495*	.816*	.753*	.807*
Error	.006	-.018	.699*	.672*	.407*	.447*	.661*	.687*	.636*
* $p < 0.01$									

Table 2 – Pearson correlation coefficients (*r*) between predictor variables and the dependent variables of response time (RT) and error. Predictors are grouped by level of measurement; those that our study is the first to evaluate are shaded.

found to have a positive skew, so we performed square root transformations[59] on both variables to improve their distribution. We report on the Pearson correlation coefficients (r) between predictor variables and the dependent variables in Table 2. We found significant positive correlations between all predictors and the dependent variables, with the exception of those at the global level. These results show that all factors measured on the solution path and search set were moderate to strong individual predictors of response time and error.

Multicollinearity between factors

We also inspected the correlations between all of the predictor variables to detect multicollinearity; that is, two or more highly correlated predictors. Collinearity between two predictors prevents us from understanding the degree to which either of the two predictors entered into the model impacts the dependent variables, thus, standard practice in regression analysis is to omit one. Choosing to omit some of these predictors allows us to better examine the extent of the contributions of the remaining predictors, but leaves questions surrounding the omitted variables to future work.

We identified two pairs of highly correlated predictors ($r > .90$) that were cause for concern: search-set edge-edge crossings (ss-ex) correlated with search-set node-edge (ss-nx) crossings, and solution-path length (sp-ln) correlated with solution-path branches (sp-

br). We omitted search set node-edge crossings, because the correlation with each dependent variable was weaker. We suspect that the relationship between solution-path length and branches stems from our graph generation model, so we would not necessarily expect to see it in other types of graphs. We chose to keep solution-path length (and omit solution-path branches) because previous work suggests that it more commonly accounts for a larger variance in performance than does the number of branches.[23]

Hierarchical multiple regression analysis

We constructed two separate hierarchical multiple regression models, one for response times and one for errors, the results of which are shown in Table 3. We included all of the predictors that significantly correlated with our dependent variables, but excluded solution-path branches (sp-br) and search-set node-edge (ss-nx) crossings because of multicollinearity. For each regression model, we also confirmed that the assumptions of homoscedasticity (similar variance in the dependent variables) and linearity were met.

The predictors were blocked as follows: block one contained solution-path length (sp-ln), continuity (sp-cn), and edge-edge crossings (sp-ex), block two contained solution-path node-edge crossings (sp-nx), and block three contained search-set edge-edge crossings (ss-ex). By placing the individual factors of interest into blocks two and three, we were able to examine the incremental validity of each factor.

We report on the standardized beta coefficients (β) at each step, which indicate the individual contribution of each predictor to the model. We also report on R^2, a measure of the amount of variation accounted by the predictor(s) included in the model at each step, and adjusted R^2, which takes into account the number of predictors in the model. All significant results were $p < 0.01$. For additional guidance in understanding the statistics, we recommend Field [59] for an entertaining introduction to interpreting the results of multiple regression analyses.

Response time model. After Step 1, the regression model accounted for 75.2% of the variance ($R^2 = 0.752$). The relative contributions of the three predictors can be further understood by examining their individual beta values, the highest of which came from solution-path length (sp-ln) ($\beta = 0.487$), followed by continuity (sp-cn) ($\beta = 0.359$) and edge-edge crossings (sp-ex) ($\beta = 0.160$). These results replicate the relative importance of these factors found by Ware et al. [23]

Adding solution-path node-edge crossings (sp-nx) in Step 2 accounted for an additional 2% of the variance ($R^2 = 0.772$, $\Delta R^2 = 0.020$). Finally, adding search-set edge-edge crossings (ss-ex) in Step 3 accounted for an additional 1.8% of the variance ($R^2 = 0.790$, $\Delta R^2 = 0.018$). The final regression model accounted for 79% of the variance in response time, and contains three statistically significant variables: Solution path length (sp-ln) had

Standardized Beta Coefficients (β values)						
	RT			Error		
	Step 1	Step 2	Step 3	Step 1	Step 2	Step 3
sp–ln	0.487*	0.458*	0.389*	0.303*	0.267*	0.168
sp-cn	0.359*	0.358*	0.298*	0.441*	0.440*	0.355*
sp-ex	0.160*	0.083	0.027	0.101	0.004	-0.075
sp-nx		0.171*	0.097		0.217*	0.113
ss-ex			0.242*			0.342*
Adj. R^2	0.745	0.764	0.781	0.533	0.563	0.597
R^2	0.752	0.772	0.790	0.545	0.578	0.614
ΔR^2		0.020*	0.018*		0.033*	0.037*
* $p < 0.01$						

Table 3 – Summary of results from the hierarchical multiple regression analysis of measured factors on response time and error. Predictors that our study is the first to evaluate are shaded.

the highest beta value (β = 0.389), followed by continuity sp-cn (β = 0.298) and search-set edge-edge crossings ss-ex (β = 0.242).

Error model. After Step 1, the model accounted for 54.5% of the variance (R^2 = 0.545). Only solution-path length (sp-ln) (β = 0.303) and continuity (sp-cn) (β = 0.441) made significant contributions. Adding solution-path node-edge crossings (sp-nx) in Step 2 accounted for an additional 3.3% of the variance (R^2 =0.578, ΔR^2 = 0.033). Finally, adding search-set edge-edge crossings (ss-ex) in Step 3 accounted for an additional 3.7% of the variance (R^2 = 0.614, ΔR^2 = 0.037).

The final model accounted for 61.4% of the variance in error. Only search-set edge-edge crossings (ss-ex) (β = 0.342) and solution-path continuity (sp-cn) (β = 0.355) were significant contributors to the final model.

Summary

Our results replicate previous findings in the literature that the factors of path length, continuity, edge-edge crossings, and branches have a significant individual effect on response time when measured on the solution path.[23] We further found significant individual effects for node-edge crossings measured on the solution path, and both node-edge and edge-edge crossings measured on the search set. We did not see any significant individual effect at the global level for edge-edge (gl-ex) and node-edge (gl-ex) crossings.

Through regression modelling we showed that we can predict 79% of the variance in response time using only three predictors: solution-path length is the most important, followed by solution-path continuity, and then search-set edge-edge crossings. Our results from the final step of the regression model for response time suggest that measuring crossings on the search set has incremental validity over measuring them on just the solution path – search-set edge-edge crossings added only an additional 1.8% to the total variance explained, a small effect, but it also removed the need for solution-path edge-edge and node-edge crossings, making for a more efficient model in terms of the number of factors needed for maximal variance prediction.

We found that the relative importance of the factors differed quite dramatically for error from what we found for response time. Our results showed that all of the factors we measured on the solution-path and search-set levels had strong individual effects on error. Similar to our results for response time, our results in the final step of the regression model for error suggest that measuring crossings on the search set has incremental validity over the solution path, explaining an additional 3.7%, which is a small effect. The final regression model accounted for 61.4% of the variance in error using only two predictors, search-set edge-edge crossings and solution-path continuity, which were very similar in importance.

We found some evidence that, at the solution-path level, node-edge crossings may be more important than edge-edge crossings. Adding solution-path node-edge crossings in step 2 of both models had a small effect, explaining an additional 2% of variance in response time, and 3.3% more in error, but in the case of response time it also reduced the contributions of solution-path edge-edge crossings to insignificant levels. These results suggest that for layouts that allow node-edge crossings, it may be the more important factor to control for relative to edge-edge crossings at the solution-path level. We were not able to examine the relative effects of node-edge crossings at the search-set level due to the multicollinearity with search-set edge-edge crossings, but our results about the individual effects of the factor suggest that it may be of similar importance. This conjecture is further evidenced by our observations of the difficulty that apparent paths caused for participants during the study.

Summary of hypotheses. All four of the hypotheses were supported, although two were only partially explored because of we were not able to include search set node-edge crossings in our multiple regression models due to limitations in our study. We summarize the outcomes for each.

H1. *Solution path node-edge crossings (*sp-nx*) will account for additional variance in performance beyond other factors on the solution path.* **Supported**. Solution path node-edge crossings explained additional variance for both dependent measures.

H2. *Search set (*ss-*) factors will account for additional variance in performance beyond all of the solution path factors.* **Supported** for search-set edge-edge crossings, but we were not able to examine search-set node-edge crossings in this analysis. Adding search-set edge-edge crossings accounted for additional variance in both dependent measures.

H3. *Search set (ss-) factors will predict performance more efficiently than solution-path (sp-) factors.* **Supported** for search-set edge-edge crossings, but we were not able to examine node-edge crossings. The overlap between the search set and solution path considerably reduced the relative contributions of node-edge and edge-edge crossings measured on the solution path, such that the search set edge-edge crossings accounted for additional variance in performance without requiring an increase in the total number of predictors required.

H4. *Global factors (*gl-*) will not account for additional variance in performance beyond the solution path and search set factors*. **Supported**. We found no significant

relationship of node-edge or edge-edge crossings measured globally with either dependent measure.

Discussion and future work

The main goal of this research was to dig deeper into what makes path tracing in graphs difficult. We did so by characterizing human path-tracing behaviour, both as a worthwhile pursuit in its own right and in service of developing a predictive model of the search set, as our primary contributions. We also present as secondary contributions the concept of the search set itself, and the preliminary validation of the predictive behavioural model through multiple regression analysis of graph readability factors. We now discuss how our research has addressed these goals, including the limitations of our approach and possible routes for future work.

The characterization of path tracing behaviours

Our characterization of path-tracing behaviours in graphs extends beyond the previously proposed geodesic tendency.[7] While we did find strong supporting evidence for this tendency, we also found many situations in which it falls short for explaining what people do. We sharpened the description and shortened the term that was used in previous work, where this phenomenon has been called the *geodesic path* tendency. Our discussion

emphasizes that it entails following the *closest-to-geodesic branch*. We find this description more evocative because it emphasizes that a decision is made many times along a path, once for each perceived hop, rather than only once for the entire path.

Our observations revealed a more complex behavioural framework, within which the geodesic tendency plays a major role but can be overridden by other tendencies: the tendency to continue following straight lines, the tendency to avoid directions that point away rather than towards the target, and the tendency to be misled into tracing apparent branches that are not in fact true topological connections. Moreover, a full model of path-tracing behaviour requires understanding when people stop tracing one path in order to try another, and where they begin their next tracing attempt. From our observations we also characterized a number of behavioural based stopping conditions, such as the tendency of users to stop searching soon after going *past the target*, making paths that do so much harder to find.

The behavioural framework we present here can act as a baseline against which to compare further work. While we believe that the framework should allow reasonable guesses for parameters that could be used for a range of similar situations, our study design and our analytic approach were necessarily limited by balancing precision and completeness against the time available to conduct this research. More observational work

can be done to untangle the relationships between the geodesic path tendency and other tendencies that we characterize, in order to model exactly how they interact and under what conditions each should take priority. One parameter space to explore in future work is the characteristics of the graph itself: size, edge density, and synthesis technique (for example, hierarchically clustering a base graph rather than permuting a mesh according to a preferential attachment model). Another large parameter space worth exploring is the visual encoding technique used to lay out the graph, including layouts via algorithms such as multi-level methods[33,60,61] or constraint optimization[62] rather than relatively naïve force-directed placement.[63] The layout technique directly affects the search set, since it determines which paths are closest to the geodesic, and thus it is likely that a search-set model should be customized for families of layout approaches; however, we conjecture that it is not necessary to create one for each individual algorithm. Another space of alternatives is how edges are drawn, for example as curved lines rather than the simple straight-line encoding that we studied.[64,65] Moreover, it would be useful to see whether and how the addition of scaffolding interaction techniques such as highlighting may change the nature of the behaviours we described here. Finally, it would be useful to investigate how behaviours differ for other abstract tasks, for example those that combine reading attribute information with topological structure traversal.[19]

While we found the recorded path-tracing data from the Cintiq tablet to be quite rich, and sufficient for our study, we know that it did not capture the complete picture of what users were doing. Some of the noise in our logged data can be attributed to instances where users visually examined nodes but forgot to point at them with the pen. We chose not to use eye tracking in our study primarily because of its high overhead with respect to the analysis required in the development of our predictive model. A follow-up study could combine the tablet approach with eye tracking using tools to automatically compare or correlate node hover and eye tracking data to examine how well pointing and eyes match up, and to potentially capture aspects that the tablet misses.

Qualitative analysis through coding always involves a degree of subjectivity: a different investigator might describe some of the path-tracing behaviours that we identified in a different way, or even identify other behaviours that we did not. A useful follow-up analysis could employ additional coders to examine the reliability of our single investigator's codes, and potentially expand upon our findings. Alternative visual analysis techniques may bear fruit; our approach to exploring trajectory data with small multiples that show the evolution over time for a single person and a single layout is only one possible tactic. Andrienko et al.[49] discuss many alternatives for the visual analysis of trajectories: flow maps, clustering by flow similarity, and frequent sequence discovery

seem like the most appropriate choices to try. Quantitative computational methods such as machine learning might reveal different patterns than human judgement yields, and are another promising avenue for future research to explore.

The predictive search set model

Our predictive behavioural model allowed us to predict a set of paths that users were likely to follow at fairly high accuracy (87%). We consider this model a good first step: it captures most of the behaviours that we observed in a robust way that avoids overfitting the training set in the first analysis phase. We encourage future research on search set models that strive for further breadth, completeness, and accuracy. For example, although we noted in our characterization of behaviours that users could be quite distracted by apparent paths caused by node-edge crossings, our final predictive model only accounts for true topological paths. A more complex model could take into account both true topological paths and apparent paths, thereby supporting layout algorithms featuring nuanced adjustments to local regions of the graph to eliminate node-edge crossings on important paths. Future work could lead to models that support relative rankings of paths within the equivalence classes that we propose, or even more specific priorities to different paths within the search set based on their relative salience, supporting a layout adjustment

algorithm that determines whether a particular path is sufficiently high priority to merit a layout change.

The search set concept

A secondary contribution of this work is the concept of a search set. It appears to be an apt model for real human behaviour: we have shown it to be predictable when applied to path-tracing tasks. We promote the idea of a search set to analyse exactly the subset of a graph that is relevant to a particular task, at an intermediate level between completely global and the strictly local single path that is the answer to a specific query.

The search set concept may serve to illuminate aspects of human behaviour that have been difficult to unravel thus far; it may serve to explain the variation in results on global edge-edge crossings found in previous research. Evaluation results for this factor have been very mixed;[8,11,20,37] our own study was one of several to find a lack of effect of global edge-edge crossings on performance. Our conjecture is that the effect depends on the size of the search set in relation to the size of the full graph. In a small graph, a user may search most of the graph to complete a task, so global measurements of factors will heavily overlap with the search set. Our study used somewhat larger and denser graphs than have typically been in used in previous work, for a smaller overlap; this difference may explain the lack of any significant relationship between the global factors and our dependent variables.

Factor measurement for model validation

Although factor measurement was the initial impetus for our investigation, in the end it was relegated to a supporting role in validating our predictive behavioural model. We applied our predictive model of a search set to the problem of factor measurement both as validation that the model itself is a reasonable approximation of the human behaviour we had observed, and as an example of how the technique can be used. We consider the results of the regression analysis to be encouraging evidence that the concept of a search set is on target; indeed, we see a modest quantitative improvement for even this first attempt at a predictive model.

Our findings pertain specifically to one type of path-tracing task in graphs. It would be useful to understand how the relative importance of the factors we examined in our study differs for different abstract tasks, such as browsing or comparison. Future research could also explore whether the incremental improvements seen by extending measurement of edge-edge crossings from the solution path to the search set also hold true of other factors. The differences we found in our analysis between how the various factors influence response time and error strengthens the case that no single factor that dominates graph readability, so we should seek to understand a factor's priority or importance in a specific context. This idea has received limited practical attention beyond Eades et al.[66], who

showed that compromises between factors based on their relative importance can lead to better layouts. Future research should continue to examine how factors might be traded off to provide the best support for particular user tasks or priorities.

Regression versus ANOVA for factor characterization

We echo and emphasize the call of Ware et al.[23] for the benefits of regression analysis over simply testing for statistical significance with methods such as analysis of variance. Untangling the relationships between factors will help characterize the algorithms that use these factors, and it will also help develop guidelines of how to map between algorithms and the requirements of specific visual encoding and interaction techniques.[67] A small methodological contribution in this article is that we advocate for hierarchical rather than stepwise multiple regression, based on recommendations from the clinical psychology literature on incremental validity.[42]

Conclusion

In this article, we proposed the concept of the search set: the subset of the graph that is likely to be carefully investigated by a user in carrying out a path-tracing task in a graph. The search set concept was motivated by our interest in determining path difficulty for the purposes of experimental comparisons of techniques, and we focused on this application in

our work. We also presented range of potential practical applications that a predictive search set model in the design of visual encoding and interaction techniques for graphs. A primary contribution of the work is a characterization of common human path-tracing behaviours based on detailed qualitative analysis of observations of people using visual representations of graphs for path tracing. These include verification of the closest-to-geodesic tendency, and descriptions of conditions under which people stop following paths, the likely directions for the first hop in a path, the tendency to revisit previously followed paths, and the tendency to mistakenly follow apparent paths in addition to true topological paths. Another primary contribution of this work is an initial predictive behavioural model of the new concept of a search set that is based on these observed behaviours and is robust to a range of parameters. We validated the search set model by measuring graph readability factors on this set, in comparison with measuring them globally on the entire graph or very locally on only the single path that is the correct solution. The factors tested included edge-edge crossings, node-edge crossings, path continuity, and path length. The modest improvements that we achieved in the efficiency and total variance accounted for in predicting response time and error are encouraging evidence that the concept of a search set has merit, even though our model is a first attempt at algorithmic instantiation of complex human behaviour. A secondary contribution of this article is the careful comparison of the

relative importance of factors measured at these three levels of a graph through multiple regression analysis. We also found key differences in the relative weighting of the importance of the factors that affect response time versus error.

Acknowledgements

We thank Matt Brehmer, Joel Ferstay, Stephen Ingram for feedback on paper drafts and Stephen North for feedback on the project.

Funding

This work was supported by a gift from AT&T Research, a Postgraduate Scholarship from the Natural Sciences and Engineering Research Council of Canada (NSERC), and a Discovery Grant from NSERC.

References

1. Tamassia R. On embedding a graph in the grid with the minimum number of bends. *SIAM J. Comput* 1987; 16(3):421–444.

2. Battista GD, Eades P, Tamassia R, and Tollis I. Graph drawing: algorithms for the visualization of graphs. Upper Saddle River: Prentice Hall; 1998.

3. Ferrari D and Mezzalira L. On drawing a graph with the minimum number of crossings. Technical Report, Istituto di Elettrotecnica ed Elettronica, Politecnico di Milano. Report no. 69-11, 1969.;

4. Eades P. A heuristic for graph drawing. *Congr Numer* 1984; 42:149–160.

5. Burch M, Heinrich J, Konevtsova N, Höferlin M, and Weiskopf D. Evaluation of traditional, orthogonal, and radial tree diagrams by an eye tracking study. *IEEE Trans Vis Comput Graph* 2011; 17(12):2440–2448.

6. Pohl M, Schmitt M, and Diehl S. Comparing the readability of graph layouts using eyetracking and task-oriented analysis. In: *Proceedings of the Fifth Eurographics conference on Computational Aesthetics in Graphics, Visualization and Imaging*, pp. 49–56. Aire-la-Ville, Switzerland: Eurographics Association.

7. Huang W, Eades P, and Hong S. A graph reading behavior: geodesic-path tendency. In: *Proceedings of IEEE Pacific Visualization Symposium*, Beijing, China, 20-23 April 2009, pp. 137–144. Washington, DC: IEEE Computer Society.

8. Huang W and Huang M. Exploring the relative importance of number of edge crossings and size of crossing angle: a quantitative perspective. *Int J Adv Intell* 2011; 3(1):25–42.

9. Körner C. Sequential processing in comprehension of hierarchical graphs. *Appl Cogn Psychol* 2004; 18(4):467–480.

10. Dunne C and Shneiderman B. Improving graph drawing readability by incorporating readability metrics: a software tool for network analysts. HCIL Technial Report, University of Maryland, USA. Report no. HCIL200913, 2009;

11. Dwyer T, Lee B, Fisher D, Quinn KI, Isenberg P, Robertson G, and North C. A comparison of user-generated and automatic graph layouts. *IEEE Trans Vis Comput Graph (Proc InfoVis)* 2009; 15(6):961–968.

12. Ham F van and Rogowitz B. Perceptual organization in user-generated graph layouts. *IEEE Trans Vis Comput Graph (Proc InfoVis)* 2008; 14(6):1333–1339.

13. Cleveland WS and McGill R. Graphical perception: theory, experimentation, and application to the development of graphical methods. *J Am Stat Assoc* 1984; 79(387):531–554.

14. Heer J, Kong N, and Agrawala M. Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualizations. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Boston, USA, 4-9 April 2009, pp. 1303–1312. New York, NY: ACM.

15. Kornaropoulos EM and Tollis IG. Dagview: an approach for visualizing large graphs. In: *Proceedings of the 20th international conference on Graph Drawing Pages*, Redmond, Washington, 19-21 September 2012, pp. 499–510. Berlin, Heidelberg: Springer-Verlag.

16. Yuan X, Che L, Hu Y, and Zhang X. Intelligent graph layout using many users' input. *IEEE Trans Vis Comput Graph (Proc InfoVis)* 2012; 18(12):2699–2708.

17. Sedlmair M, Tatu A, Munzner T, and Tory M. A taxonomy of visual cluster separation factors. *Comput Graph Forum (Proc EuroVis)* 2012; 31(3):1335–1344.

18. Brehmer M and Munzner T. A multi-level typology of abstract visualization tasks. *IEEE Trans Vis Comput Graph (Proc InfoVis)* 2013; 19(12):2376–2385.

19. Lee B, Plaisant C, Parr CS, Fekete J-D, and Henry N. Task taxonomy for graph visualization. In: *Proceedings of the 2006 AVI workshop on BEyond time and errors:*

novel evaluation methods for information visualization, Venezia, Italy, 23-26 May 2006, pp. 1–5. New York, NY: ACM Press.

20. Purchase HC, Cohen RF, and James MI. Validating graph drawing aesthetics. In: *Proceedings of the Symposium on Graph Drawing*, Passau, Germany, 20-22 September 1995, pp. 435–446. London, UK: Springer-Verlag.

21. Purchase H. Which aesthetic has the greatest effect on human understanding? In: *Proceedings of the 5th International Symposium on Graph Drawing*, Rome, Italy, 18-20 September 1997, pp. 248–261. London, UK: Springer-Verlag.

22. Purchase H. Performance of layout algorithms: comprehension, not computation. *J Vis Lang Comput* 1998; 9(6):647–657.

23. Ware C, Purchase H, Colpoys L, and McGill M. Cognitive measurements of graph aesthetics. *Inf Vis* 2002; 1(2):103–110.

24. Körner C. Eye movements reveal distinct search and reasoning processes in comprehension of complex graphs. *Appl Cogn Psychol* 2011; 25(6):893–905.

25. Huang W. Establishing aesthetics based on human graph reading behavior: two eye tracking studies. *Pers Ubiquitous Comput* 2013; 17(1):93–105.

26. Ghoniem M, Fekete J-D, and Castagliola P. A comparison of the readability of graphs using node-link and matrix-based representations. In: *Proceedings of the IEEE Symposium on Information Visualization*, Austin, USA, 10-12 October 2004, pp. 17–24. Washington, DC: IEEE Computer Society.

27. Ware C. Information visualization: perception for design. 3rd ed. San Francisco: Morgan Kaufman; 2000.

28. Heer J and Bostock M. Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Atlanta, USA, 10-15 April 2010, pp. 203–212. New York, NY: ACM.

29. Rensink RA and Baldridge G. The perception of correlation in scatterplots. *Comput Graph Forum* 2010; 29(3):1203–1210.

30. Jänicke H and Chen M. A salience-based quality metric for visualization. In: *Proceedings of the 12th Eurographics/IEEE - VGTC conference on Visualization*,

Bordeaux, France, 9-11 June 2010, pp. 1183–1192. Aire-la-Ville, Switzerland: Eurographics Association.

31. Moscovich T, Chevalier F, Henry N, Pietriga E, and Fekete J-D. Topology-aware navigation in large networks. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems Pages*, Boston, USA, 4-9 April 2009, pp. 2319–2328. New York, NY: ACM.

32. Marks J, Andalman B, Beardsley PA, Freeman W, Gibson S, Hodgins J, Kang T, Mirtich B, Pfister H, Ruml W, Ryall K, Seims J, and Shieber S. Design galleries: a general approach to setting parameters for computer graphics and animation. In: *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, Los Angeles, USA, 5-7 August 1997, pp. 389–400. New York, NY: ACM Press.

33. Archambault D, Munzner T, and Auber D. Topolayout: multilevel graph layout by topological features. *IEEE Trans Vis Comput Graph.* 2007; 13(2):305–317.

34. Dwyer T, Marriott K, and Stuckey PJ. Fast node overlap removal. In: *Proceedings of the 13th international conference on Graph Drawing Pages*, Limerick, Ireland, 12-15 September 2005, pp. 153–164. Berlin, Heidelberg: Springer-Verlag.

35. Borgatti SP. Centrality and network flow. *Soc Networks* 2005; 27:55–71.

36. Correa C, Crnovrsanin T, and Ma K-L. Visual reasoning about social networks using centrality sensitivity. *Trans Vis Comput Graph* 2012; 18(1):106–120.

37. Huang W, Hong S, and Eades P. Layout effects on sociogram perception. In: *Proceedings of the 13th International Symposuim on Graph Drawing*, Limerick, Ireland, 12-14 September 2005, pp. 262–273. Berlin, Heidelberg: Springer-Verlag.

38. Himsolt M. Comparing and evaluating layout algorithms within GraphEd. *J Vis Lang Comput* 1995; 6:255–273.

39. Purchase HC, Allder J, and Carrington D. Graph layout aesthetics in UML diagrams: user preferences. *J Graph Algorithms Appl* 2002; 6(3):255–279.

40. Vicente K and Torenvliet GL. The earth is spherical (p<0. 05): alternative methods of statistical inference. *Theor Issues Ergon Sci* 2000; 1(3):248–271.

41. Haynes SN and Lench HC. Incremental validity of new clinical assessment measures. *Psychol Assess* 2003; 15(4):456–466.

42. Hunsley J and Meyer GJ. The incremental validity of psychological testing and assessment: conceptual, methodological, and statistical issues. *Psychol Assess* 2003; 15(4):446–455.

43. Purchase HC, Pilcher C, and Plimmer B. Graph drawing aesthetics: created by users not algorithms. *IEEE Trans Vis Comput Graph (Proc. InfoVis)* 2012; 18(1):81–92.

44. Purchase HC, Plimmer B, Baker R, and Pilcher C. Graph drawing aesthetics in user-sketched graph layouts. In: *Proceedings of the Australasian Conference on User Interface*, Brisbane, Australia, 18-22 January 2010, pp. 80–88. Darlinghurst, Australia: Australian Computer Society, Inc.

45. Huang W, Hong S, and Eades P. Effects of crossing angles. In: *Proceedings of IEEE Pacific Visualization Symposium*, Kyoto, Japan, 5-7 March 2008, pp. 41–46. Washington, DC: IEEE Computer Society.

46. Marriott K, Purchase H, Wybrow M, and Goncu C. Memorability of visual features in network diagrams. *IEEE Trans Vis Comput Graph (Proc InfoVis)* 2012; 18(12):2477–2485.

47. Purchase H, Hoggan E, and Görg C. How important is the "mental map"? - an empirical investigation of a dynamic graph layout algorithm. In: *Proceedings of 14th International Symposium on Graph Drawing*, Karlsruhe, Germany, 18-20 September 2006, pp. 184–195. Berlin, Heidelberg: Springer-Verlag.

48. Huang W, Hong S, and Eades P. Predicting graph reading performance: a cognitive approach. In: *Proceedings of Asia-Pacific Symposium on Information Visualisation*, Tokyo, Japan, 1-3 February 2006, pp. 207–216. Darlinghurst, Australia: Australian Computer Society, Inc.

49. Gennady L. Andrienko, Natalia V. Andrienko, Michael Burch DW. Visual analytics methodology for eye movement studies. *IEEE Trans Vis Comput Graph (Proc InfoVis)* 2012; 18(12):2889–2898.

50. Kim S-H, Dong Z, Xian H, Upatising B, and Yi JS. Does an eye tracker tell the truth about visualizations?: findings while investigating visualizations for decision making. *IEEE Trans Vis Comput Graph (Proc InfoVis)* 2012; 18(12):2421–2430.

51. Ware C and Bobrow R. Supporting visual queries on medium-sized node–link diagrams. *Inf Vis.* 2005; 4(1):49–58.

52. Barsky A, Munzner T, Gardy J, and Kincaid R. Cerebral: visualizing multiple experimental conditions on a graph with biological context. *IEEE Trans Vis Comput Graph (Proc InfoVis)* 2008; 14(6):1253–1260.

53. Bishop CM. Pattern recognition and machine learning (information science and statistics). New York: Springer-Verlag; 2006.

54. Watts DJ and Strogatz SH. Collective dynamics of "small-world" networks. *Nature* 1998; 393(6684):440–442.

55. Auber, D., Chiricota, Y., Jourdan, F., Melancon G. Multiscale visualization of small world networks. In: *Proceedings of Information Visualization*, Seattle, USA, 19-21 October 2003, pp. 75 – 84. Washington, DC: IEEE Computer Society.

56. Melancon G. Just how dense are dense graphs in the real world? In: *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization*, Venezia, Italy, 23-26 May 2006, pp. 1–7. New York, NY: ACM.

57. Heer J, Card S, and Landay J. Prefuse: a toolkit for interactive information visualization. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Portland, USA, 2-7 April 2005, pp. 421–430. New York, NY: ACM.

58. Cohen J. Statistical power analysis for the behavioral sciences. L. Erlbaum Associates; 1988.

59. Field A. Discovering statistics using SPSS. 3rd ed. London: SAGE Publications; 2009.

60. Hachul S and Jünger M. Drawing large graphs with a potential-field-based multilevel algorithm. In: *Proceedings of the 12th international conference on Graph Drawing*, New York, USA, 29 September - 2 October 2004, pp. 285–295. Berlin, Heidelberg: Springer-Verlag.

61. Archambault D, Munzner T, and Auber D. Grouseflocks: steerable exploration of graph hierarchy space. *IEEE Trans Vis Comput Graph.* 2008; 14(4):900–913.

62. Dwyer T. Scalable, versatile and simple constrained graph layout. *Comput Graph Forum (Proc EuroVis 09)* 2009; 28(3):991–998.

63. Brandes U. Keynote address: why everyone seems to be using spring embedders for network visualization, and should not. In: *Proceedings of the 2011 IEEE Pacific Visualization Symposium*, Hong Kong, China, 1-4 March 2011, pp. xii. Washington, DC: IEEE Computer Society.

64. Holten D, Isenberg P, van Wijk JJ, and Fekete J-D. An extended evaluation of the readability of tapered, animated, and textured directed-edge representations in node-link graphs. In: *Proceedings of the 2011 IEEE Pacific Visualization Symposium*, Hong Kong, China, 1-4 March 2011, pp. 195–202. Washington, DC: IEEE Computer Society.

65. Riche NH, Dwyer T, Lee B, and Carpendale S. Exploring the design space of interactive link curvature in network diagrams. In: *Proceedings of the International*

Working Conference on Advanced Visual Interfaces, Capri, Italy, 22-25 May 2012, pp. 506–513. New York, NY: ACM.

66. Huang W, Eades P, Hong S-H, and Lin C-C. Improving multiple aesthetics produces better graph drawings. *J Vis Lang Comput* 2013; 24(4):1–11.

67. Meyer M, Sedlmair M, Quinan PS, and Munzner T. The nested blocks and guidelines model. *Inf Vis* 2013; Epub ahead of print 10 December 2013. DOI: 10.1177/1473871613510429