A Search Set Model of Path Tracing in Graphs: Supplementary Materials

Jessica Q. Dawson, Tamara Munzner and Joanna McGrenere

This supplementary material contains two sections. In the first section, we provide a detailed description of our search set model, as well as the algorithmic implementation and parameter selection. In the second and third sections, we discuss examples of the visualizations used for the preliminary and qualitative analyses presented in the paper.

The Search Set Model

Table 1 shows a summary of our predictive behavioural search set model. 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 output from the model is ordered discrete groups of paths that are equivalence classes, which we call batches. Within each batch a set of paths is unordered and considered to be similarly likely; together, the paths in these batches compromise the search set.

The model searches out from both anchors. When searching out from one anchor, the other anchor is used as the target for paths, and vice versa. This choice is based on our observations that participants regularly switched between using the red and blue nodes as anchors and targets in our study task. To begin in Step 1, we select a batch of likely candidate branches from each anchor to comprise the first hop in a path. The batches correspond to the groups of likely directions of search described in the previous section: *directly towards, towards, away,* and *directly away*. To start we create a batch comprising the most likely candidate branches, those that are in the directly towards group. All the candidates in one batch must be exhausted before generating the next batch, because we expect each candidate in the same group to have a similar likelihood.

Once the first batch is generated, Step 2 of the model iterates through each candidate, and from this candidate follows the closest-to-geodesic branch between the candidate and the target. If two potential branches have very similar angles or overlap, then both are

A 3-step predictive behavioural model of the search set

Input: a connected network with a unique solution between a source node and a goal node.

Anchors: source node, goal node
Target: target = goal (when anchor = source), and vice versa

The 3-Step Model:

Step 1: Generate Batch: from each anchor, generate a batch of likely candidates branches for the first hop in a path:
If all first hop branches have been considered: revisit each batch in sequence and generate likely branches for second hop;

Step 2: Follow paths from batch candidates: for each candidate, follow along the closest-to-geodesic branch towards the target until a stopping condition is met:
At each hop, add the current path, if not yet contained, to the search set;

Step 3: When out of candidates: check if the solution has been found from both anchors:
If yes: Stop the search;
Else: repeat from Step 1 with the next most likely batch of candidates.

Output: Ordered batches of paths, where the paths are unordered within each batch, comprising the predicted search set between the source and goal nodes.

Stopping Conditions:
S1) reach the target
S2) reach a node already in the current path (cycle)
S3) number of hops = solution-path length minus one *
S4) pass target with respect to anchor *

*Exception to allow one additional hop if:
i. the hop forms a straight line
ii. the hop goes *directly towards* target

Table 1 – Summary of our behavioural model for predicting a search set.

saved as the closest-to-geodesic branch. This choice is based on our observation that real users make imprecise rather than precise angle judgements. At each hop in the path, we add the path so far to the search set. Thus, if the model follows a path for four hops, we would add four paths to the search set: the first path would contain the first hop, the second path the first and second hops, and so on. This choice is based on the observation that participants often did not simply search all the way to the maximum number of hops allowed by a trial, but instead they revisited parts of paths again and again. The search set contains only one copy of each path, even if that path is encountered multiple times.

To determine when the model stops following a particular path, we constructed four different stopping conditions, S1 – S4, which are directly based on the common stopping patterns that we characterized in the previous section.

Finally, in Step 3 we check to see if the search set contains the solution path for the graph from both the source to the goal, and from the goal to the source. The decision to stop only after finding the answer in both directions is to account for our observation that many participants use the goal as an anchor for search, and the fact that a single individual might only find the solution in either direction. If the solution path has not been found in both of these directions, we return to Step 1 and generate a new batch of candidates using progressively larger angles from the straight line to the target than were used in the previous batch. Once all of the one-hop branches around both anchors have been considered in previous iterations, we expand the candidates to include two hops around the anchor. To do this, for each subsequent batch we revisit a previous batch starting with the first batch. For each one-hop candidate in the revisited batch, we select all of the likely second hop candidates, again relying on the concept of groups of likely directions. If the solution path cannot be found in both directions after iterating on all of these batches, we judge the task to be very difficult and stop after the last batch has been processed.

Algorithmic Implementation and Parameter Selection

We programmed an algorithmic implementation of our model so we would be able run it on the graphs in our data sample. To implement the model we had to assign specific parameters for the angle boundaries of each batch, as well as for the stopping conditions and the choice of geodesic shortest branches.

Our final parameters are shown in Table 2. We iterated on these parameters extensively before settling on the final choices. In order to measure the fit of the model using different parameters, we ran the algorithm on the 24 training set graphs and observed the overlap between the predicted search set and the data collected during the study.

We selected parameters to be consistent and generalizable, rather than being overly fit to our particular data set. For the first-hop batches we chose a set of parameters that fit with the general divisions what we observed to define the *directly toward, toward, away* and *directly away* groups. The larger angle that defines the *toward* group was set to 100° as opposed to 90°, thus including options that strictly speaking point 'away' from the target, again based on our observations regarding participants' imprecise angle judgements. We experimented extensively with these parameters to convince ourselves that the model is not brittle, and tried out more than a few dozen combinations. We found that as long as the construction of the batches followed a general separation of directions for first hop candidates into these four groups, the fit did not change dramatically in response to small changes in the exact parameters. The parameters for the second hop candidates, for which we used broader groups of toward (inclusive of directly toward) and away, as well as the parameters for the stopping conditions and the threshold for including more than one closest to geodesic branch, were similarly robust.

Search Set Algorithm Parameters

Sizes for Groups of Directions

From each anchor, select one-hop candidates:

Batch	Description	Range
1	*Directly towards* target	0° - 50°
2	*Towards* target	50° - 100°
3	*Away* from target	100° - 165°
4	*Directly away* from target	165° - 180°

From each candidate in batch, select two-hop candidates:

Batch	Description	Range
5 - 8	*Directly towards* or *towards* target: - Batch 5 from batch 1 candidates - Batch 6 from batch 2 candidates - Batch 7 from batch 3 candidates - Batch 8 from batch 4 candidates	0° - 100°
9 - 10	*Away from* target: - Batch 9 from batch 1 - Batch 10 from batch 2	100° - 165°

Threshold for choosing multiple closest-to-geodesic branches

Angular divergence from straight line	< 13°

Stopping condition angle definitions

Past the target (angle from geodesic)	90°
Exception i) Straight line (angle between hops)	165° - 180°
Exception ii) Directly towards	0° - 50°

Table 2 – Parameters used in the algorithmic implementation of the final search set model.

Visualizations used in Preliminary Analysis

This section of the supplementary material contains examples of the visualizations used for the preliminary analysis. All of the visualizations were programmed in Java using the Prefuse toolkit[1].

Exploratory visualizations for preliminary analysis (Version 1)

This visualization was the first that we developed to support exploration of the node hover data for the preliminary node-based analysis.

One static image was generated per graph trial. An example of one of these images is shown in Figure 1. The visualization displayed the node hovers from all participants for one graph trial, aggregated onto a single graph image. The size of a node encodes the total number of times it was hovered over by all participants; in other words, a node would become larger if multiple participants hovered over it, and/or if one participant revisited the node repeatedly. The graph id is shown in a label on the top left. Grey-scale encodes the number of participants that hovered over a particular node at least once; white nodes were not hovered over by any users. The source and goal nodes are indicated through the colour of their outlines, red and blue respectively.

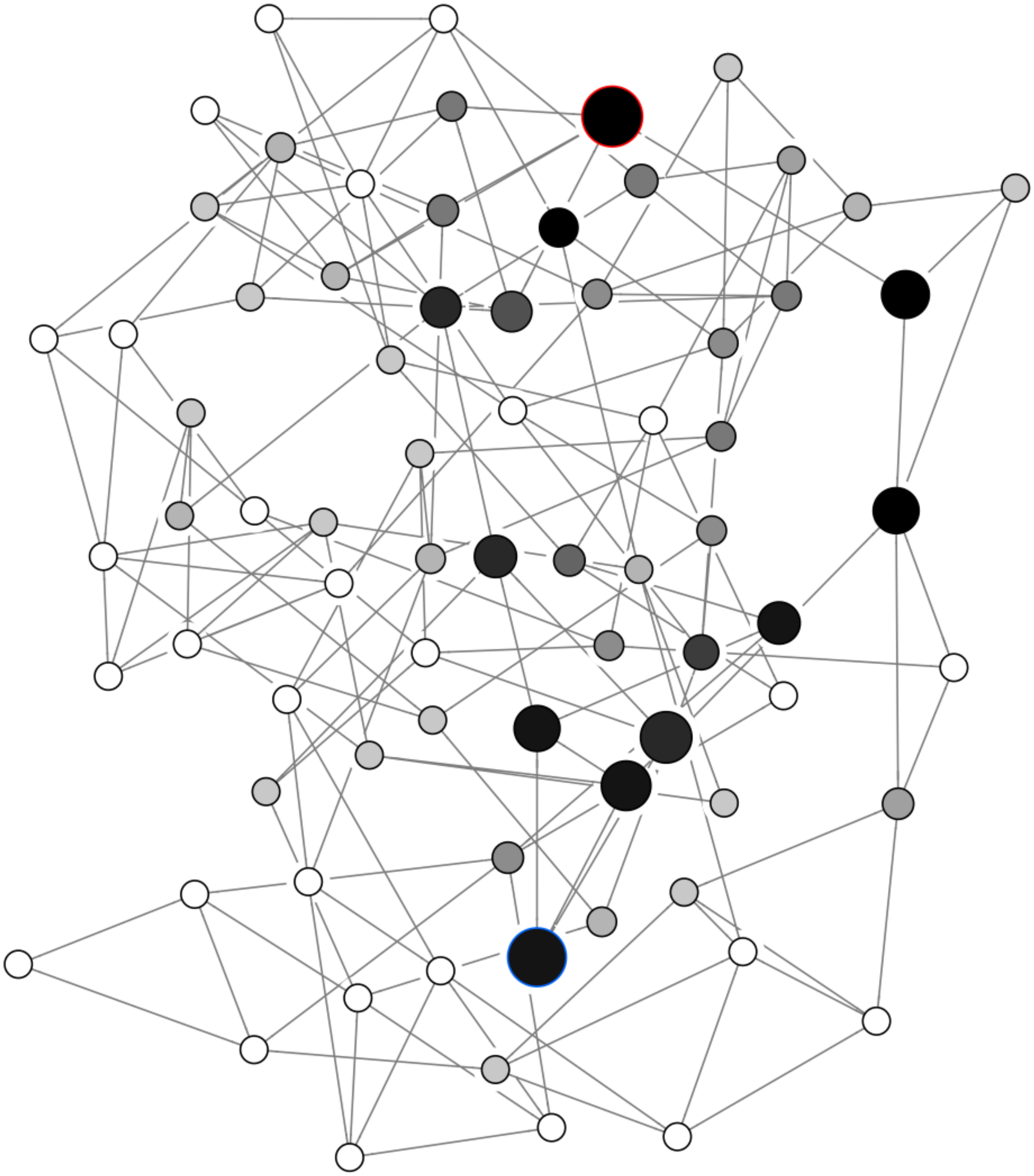

Figure 1 – Screen shot of version 1 of the exploratory visualization used in the node-based analysis.

Exploratory visualization for preliminary analysis (Version 2)

We developed a second visualization to support the preliminary node-based analysis. This version improves over version 1 by supporting interaction and providing multiple views of the data.

The visualization displayed the node hovers from all participants for one graph trial at a time. It consisted of an aggregate view and a series of small multiples showing the data from each participant. During the preliminary analysis we tried three different ways of presenting the aggregate view: versions 2.1 and 2.2 were primarily used for exploration of the data, while version 2.3 was used for confirmation of the convex hull pattern. The following subsections detail these iterations.

The visualization supported navigation between graphs using the arrow keys on the keyboard. The investigator could jump to a particular graph by pressing the space bar, and then entering the graph id when prompted. When the investigator hovered over a node, it was highlighted in every small-multiple view and its id was shown in a tooltip.

A screenshot of the visualization as it appeared on the investigator's monitor is shown in Figure 2. The aggregate view is on the left (version 2.1 shown), and the small multiples view is shown on the right.

.

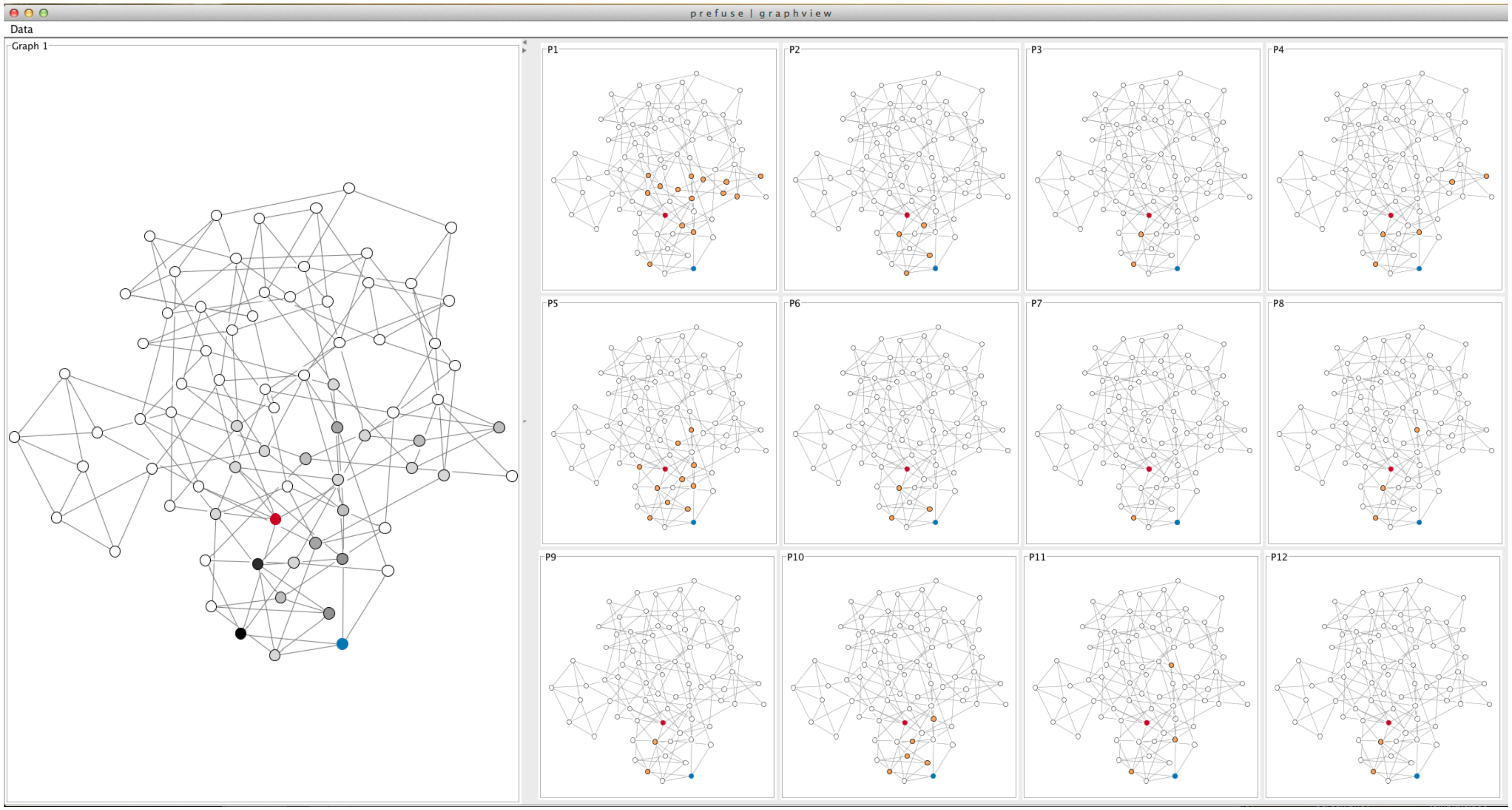

Figure 2– Screen shot of version 2 of the exploratory visualization used in the preliminary node-based analysis.

Aggregate view (Version 2.1): The first version of the aggregate view used grey-scale to encode the number of participants that hovered over a particular node at least once, as in version 1 of the exploratory visualization. The label on the top left shows the graph id. Figure 3 shows a close-up screenshot of this view.

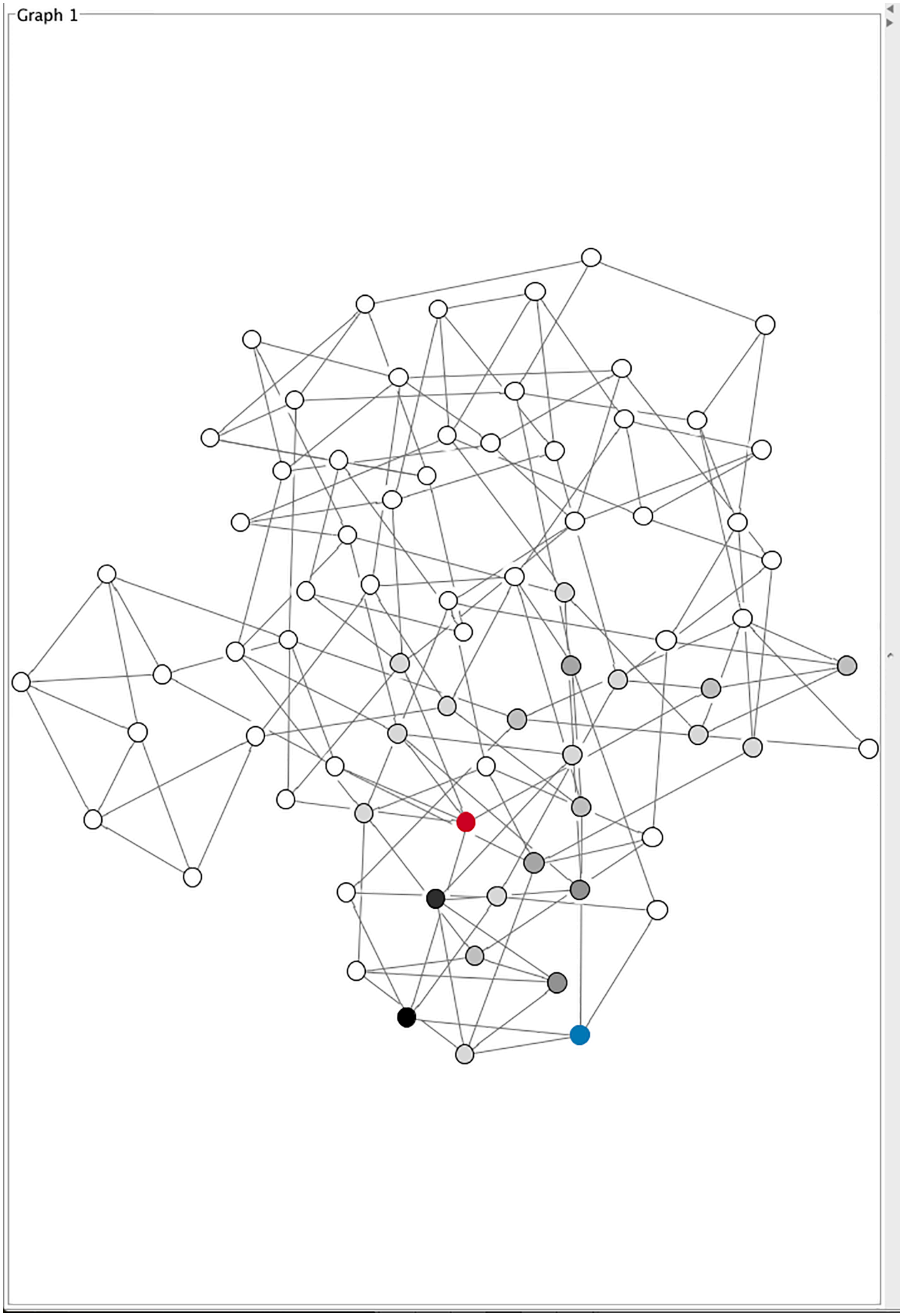

Figure 3– Screen shot of Version 2.1 of the aggregate view

Aggregate view (Version 2.2): The second version of the aggregate view used grey-scale to encode the frequency of node hovers across all participants. We made this change from the previous version of this view (version 2.1) because we thought that the frequency of hovers was more interesting than whether or not participants just visited a node, as the hover frequency says more about a node's importance to the task. The label on the top left shows the graph id as in the previous version. An example of this version of the view is shown in Figure 4.

In the very first visualization we built we used a size encoding for frequency, as in Figure 1. However, the distortion caused by the change in size made the graph more difficult to analyze, and so we changed to using grey-scale for this value.

In the end, the difference between this version and version 2.1 was subtle for many of the graphs, but allowed us to identify additional areas of the graph that were heavily used. For example, in Figure 4 we note that a few of the nodes on the right hand side of the graph, including a 1-hop neighbour of the red node, are dark and were therefore hovered over frequently. More frequently hovered over nodes suggests that participants who searched in this area spent quite a bit of time there. By comparison, version 2.1 shown in Figure 3 does not capture the importance of these particular nodes because only a couple of participants hovered over them.

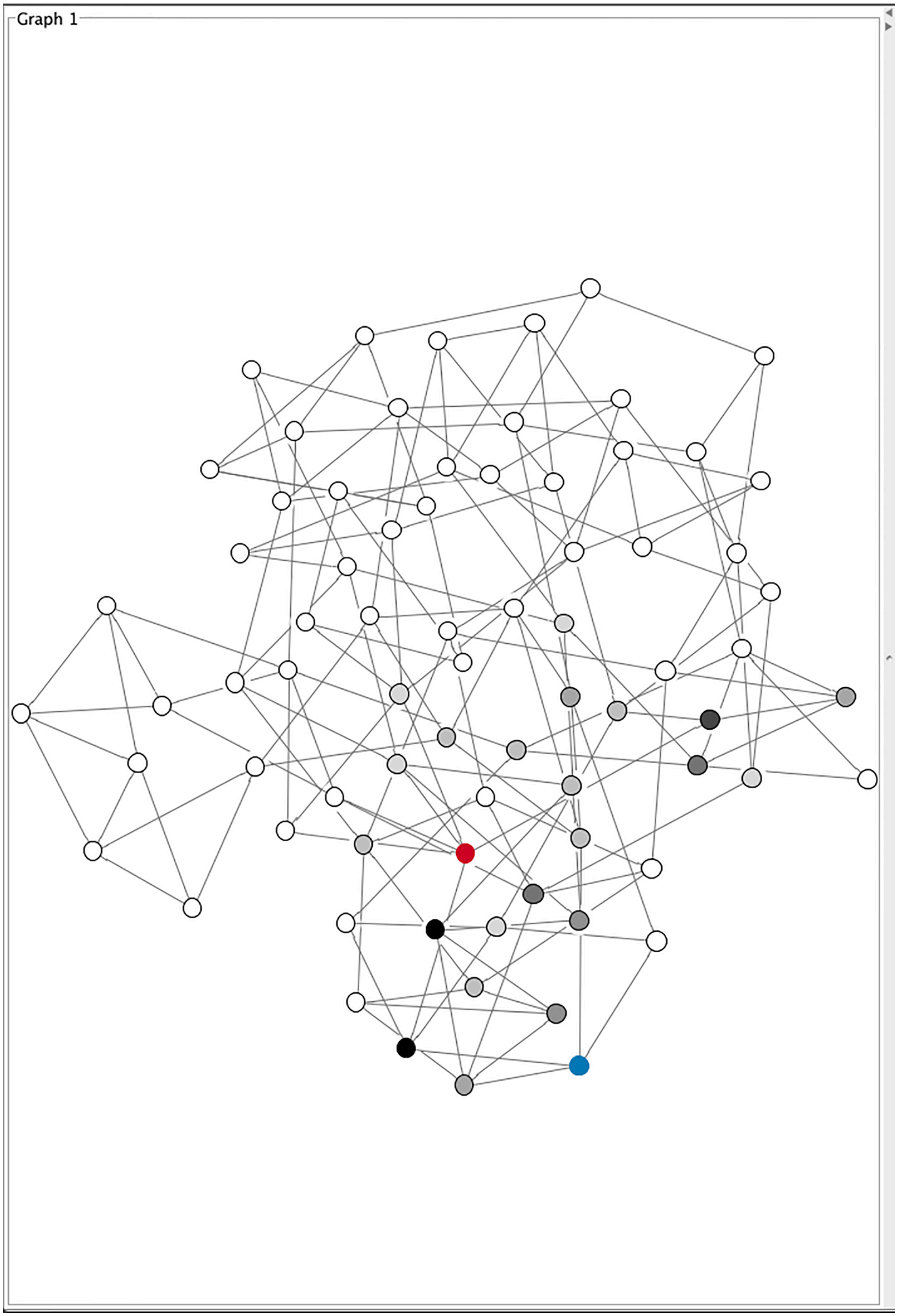

Figure 4 – Screen shot of Version 2.2 of the aggregate view

Aggregate view with convex hull (Version 2.3): We created the third and final version of the aggregate view much later, after we had completed the bulk of the preliminary analysis. The view shows the hover frequency data, encoded the same way as in version 2.2. The only change in this version is the addition of a visual representation of the convex hull around the one-hop neighbours of the source (red) and goal (blue) nodes, which is shaded in green. A close-up screenshot of this view is show in Figure 5.

This visualization was primarily used for visual confirmation of the convex hull behaviour, which we had detected using the previous versions of the visualizations.

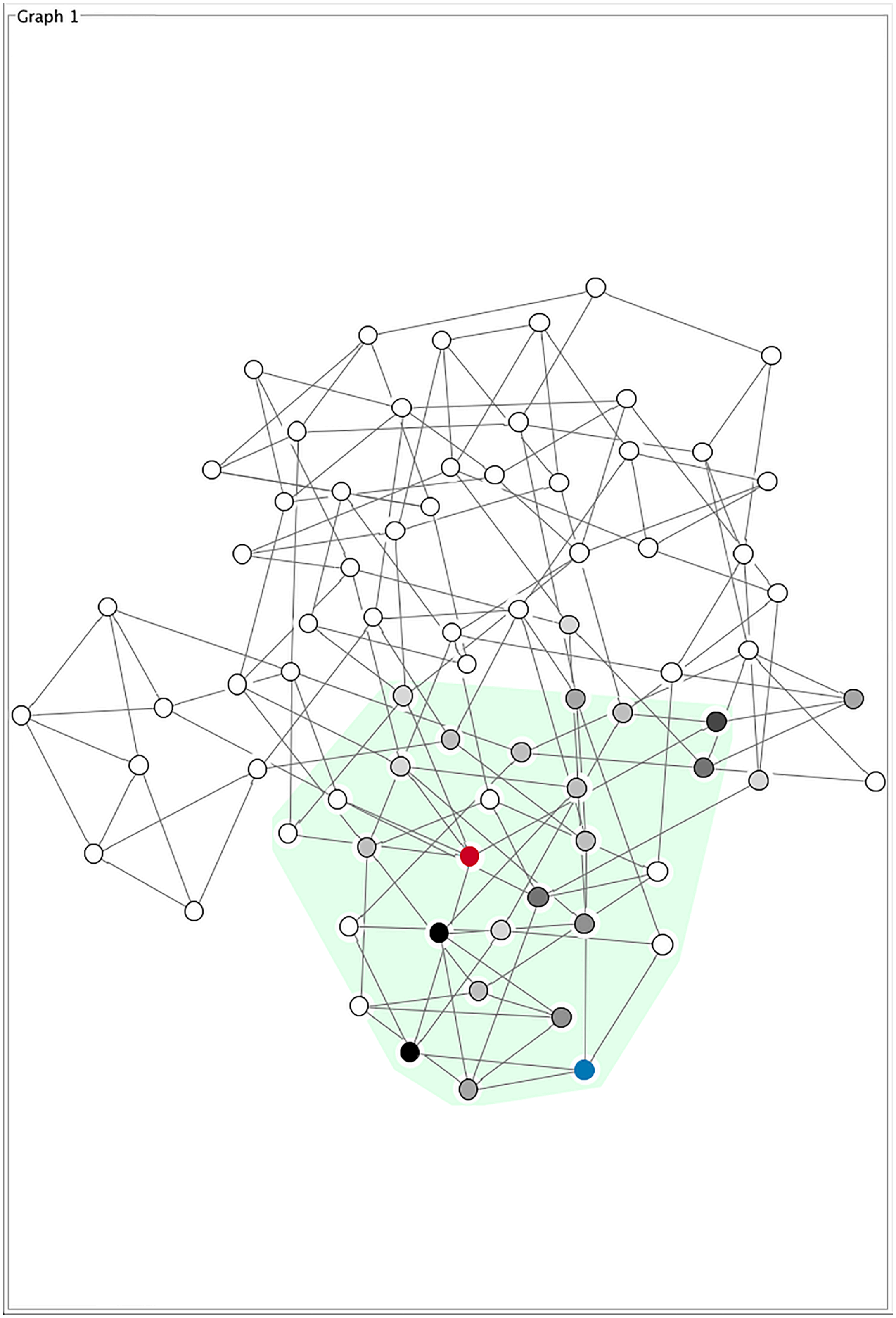

Figure 5 – Screen shot of Version 2.3 of the aggregate view

Small multiples: Figure 6 shows a close up of the small multiples used in the second version of the exploratory visualization. Each individual small multiple displays the hovers from one participant onto a graph image of the trial being visualized. The participant id is displayed in a label at the top left of each small multiple. Nodes that the participant hovered over at least once are coloured orange, while white nodes were not hovered over.

We found the hover frequency data displayed in version 2.2 of the aggregate view to be sufficient for our exploration during the preliminary node-based analysis, and thus did not make a frequency version of the small multiples.

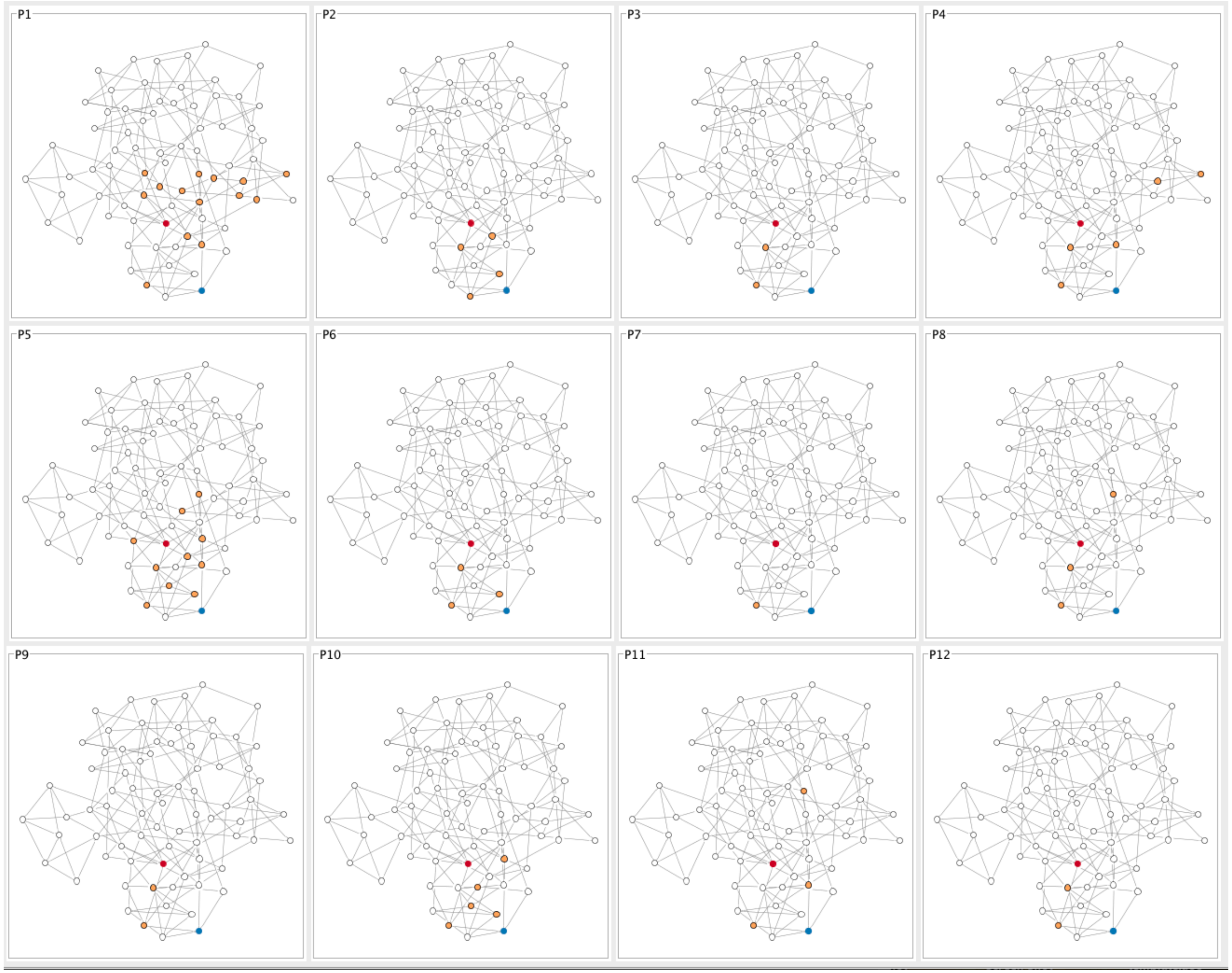

Figure 6 – The small multiples view used in the second version of the exploratory visualization for the node-based analysis

Visualization for Qualitative Analysis of Path-Tracing Behaviours

This section provides more detail on the visualization developed to support the qualitative analysis of path-tracing behaviours. The visualization was programmed in Java using the Prefuse toolkit[1].

Figure 7 shows a screenshot of the entire visualization used in the qualitative analysis as it appeared on the investigator's monitor. The visualization displays the data from one participant trial at a time.

The small multiples on the right visualized a maximum of 20 steps. These were the primary views used by the investigator in performing the qualitative coding. The hovered node(s) in each step are coloured orange. The first node in a collapsed topological sequence is coloured light orange, with subsequent nodes coloured dark orange, and edges along the topological path between them also coloured orange.

The aggregate view on the left was similar to that used in version 2 of the exploratory visualization for the preliminary node-based analysis. For the qualitative analysis, the aggregate view provided a useful overview by showing the nodes that the participant hovered over at least once, which are coloured in black. As additional support, when the investigator hovered over a node, it was highlighted in every small-multiple view and its id was shown in a tooltip. The graph id and participant id for the trial is shown in a label at the top left of the aggregate view, while the small multiples are labeled with the steps that they show.

The visualization also supported navigation between trials using the arrow keys on the keyboard: up/down arrow keys moved between participants, while left/right arrow keys moved between graphs. The investigator could jump to a particular graph by pressing the space bar, and then entering the graph id when prompted.

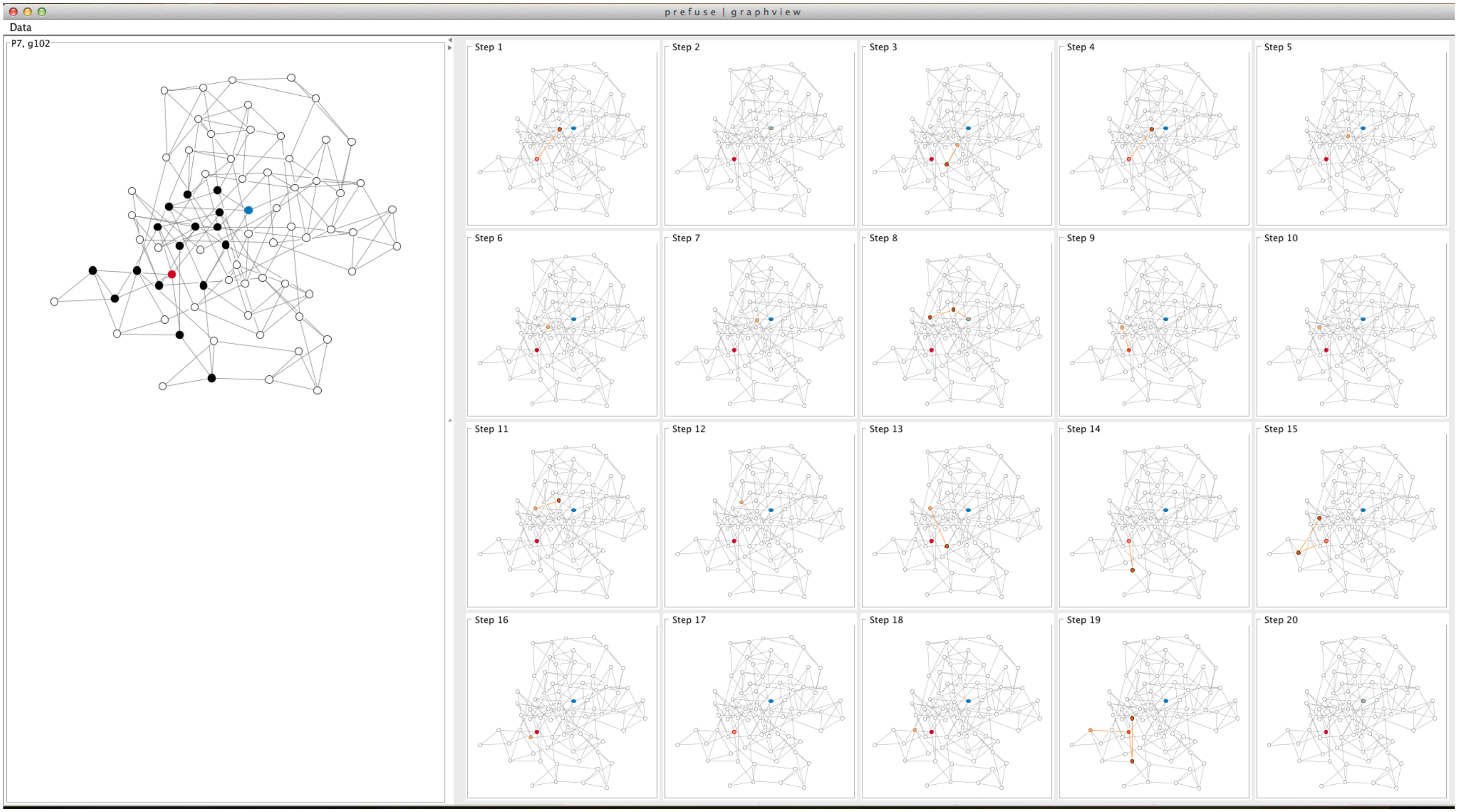

Figure 7 – Screenshot of the visualization used for the qualitative analysis of path-tracing behaviours,

References

1. 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.