The Perception of Graph Properties

In Graph Layouts

by

Utkarsh Soni

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved June 2018 by the
Graduate Supervisory Committee:

Ross Maciejewski, Chair
Stephen Kobourov
Jorge Sefair

ARIZONA STATE UNIVERSITY

August 2018

ABSTRACT

When looking at drawings of graphs, questions about graph density, community structures, local clustering and other graph properties may be of critical importance for analysis. While graph layout algorithms have focused on minimizing edge crossing, symmetry, and other such layout properties, there is not much known about how these algorithms relate to a user's ability to perceive graph properties for a given graph layout. This study applies previously established methodologies for perceptual analysis to identify which graph drawing layout will help the user best perceive a particular graph property. A large scale ($n = 588$) crowdsourced experiment is conducted to investigate whether the perception of two graph properties (graph density and average local clustering coefficient) can be modeled using Weber's law. Three graph layout algorithms from three representative classes (Force Directed - FD, Circular, and Multi-Dimensional Scaling - MDS) are studied, and the results of this experiment establish the precision of judgment for these graph layouts and properties. The findings demonstrate that the perception of graph density can be modeled with Weber's law. Furthermore, the perception of the average clustering coefficient can be modeled as an inverse of Weber's law, and the MDS layout showed a significantly different precision of judgment than the FD layout.

DEDICATION

To my mom, dad, Ishaan, Yatarth, Ishita, Siddhi and other friends for their support

ACKNOWLEDGMENTS

Foremost, I would like to thank my advisor and mentor Dr. Ross Maciejewski for providing me the opportunity to work in his lab. This thesis would not have been possible without his guidance and continuous support. I would also like to give thanks to Dr. Jorge Sefair and Dr. Stephen Kobourov for being a part of my graduate committee. This work was supported in part by the U.S. Department of Home-land Security's CAOE, Award 2017-ST-061-QA0001-01 and the National Science Foundation, Grant Nos. 1639227, 1712119, and 1740858.

TABLE OF CONTENTS

Page

CHAPTER

CHAPTER Page

APPENDIX

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

Given a particular graph, there are multiple graph drawing algorithms for producing a graph visualization. These algorithms remove edge crossings, depict symmetric substructures, and organize vertices and edges according to various heuristics and optimization techniques. Each graph drawing algorithm attempts to improve a user's ability to interpret a graph [58] through its own optimization criteria and computational method. Experiments that compare the performance of different graph layout algorithms typically consider the visual properties of the graph drawings (e.g. vertex size [51]), the extent to which common aesthetics are emphasized [59], or their computational complexity. However, graphs can be characterized in many different quantitative ways, not only the number of vertices and edges, but also the structure of the graph (e.g., density, diameter, clustering coefficient, degree distribution etc.). To date, most comparisons between graph layout algorithms have focused on theoretical properties such as the computational complexity of an algorithm. For example, Battista et al. [22] compared the running time of 4 orthogonal layout algorithms, Himsolt [38] compared the runtimes of 10 layout algorithms present in the GraphEd system [39](a widely used graph editor), and Hachul et al. [34] compared running time of 6 computationally efficient graph layout algorithms on a wide variety of graphs(consisting of both artificial and real-world graphs). But little work has explored the perception of graph properties with respect to a graph layout.

This thesis presents the results of two experiments that focus on comparing graph layout algorithms with respect to the extent to which they support the perception of underlying properties of the graph. The hypothesis is that some graph layout algorithms may make it easier for a viewer to discern graph properties than others. Following the example of perception studies in psychology and the work of Rensink et al. [62], we focus on measuring the just noticeable difference (JND) – that is, the smallest difference between two property values that can be perceived by humans – for two graph properties across three different graph layout algorithms.

Rensink et al. [62] used a JND approach for the perception of correlation in scatterplots, showing that such perception can be modeled using Weber's law – that is, the JND between the perception of a given correlation (the target stimulus) and a different one is a constant ratio of the original target stimulus. Harrison et al. [36] used this JND methodology as a means of comparing nine different ways in which correlation can be visualized (e.g. scatterplot, parallel coordinates, donut charts). Subsequent work by Kay and Heer [46] revisited the data collected by Harrison et al. [36] and enhanced the analysis with log transformation and censored regression to inclusively embrace all individual data points. This same methodology offers a means for comparing graph layout algorithms with respect to perception of graph properties, providing a quantifiable means of determining which algorithms outperform others with respect to perception of important properties. For example, if a communication network is to be drawn so that people can easily discern the density of the network – which is the best layout algorithm to use?

This thesis applies the experimental methodology of Rensink et al. [62] to determine if the perception of graph properties can also be modeled following Weber's law. As in Harrison et al. [36], this thesis adopts this methodology for a crowdsourcing environment on Amazon's Mechanical Turk, and applies it to two graph properties (graph density and average local clustering coefficient) across three different graph layout algorithms (Force Directed - FD [41], Circular [24], and Multi-Dimensinal Scaling - MDS [7]). Thus, this thesis is able to:

- Demonstrate that the perception of graph properties can (sometimes) be modeled using Weber's law;
- Provide a means for comparing the effectiveness of different graph layouts for perceiving graph properties.

This work is important because graph drawings are increasingly being used in a variety of non-research areas (e.g., fraud detection, criminal networks, marketing) and depicting them in a way that makes the salient properties easy to perceive will make them more useful for the appropriate domain task. All stimuli and responses used in this experiment have been included in the appendix.

Chapter 2

RELATED WORK

This section first reviews various classes of graph layout algorithms like Force-Directed layouts, distance based layouts, orthogonal layouts and tree drawing layouts. Then the related work in comparisons of different graph layouts has been presented. Subsequently, this section reviews various graph properties, graph mining techniques, and several graph visualization systems. Finally, the studies that use Just Noticeable Difference for analyzing perception of properties are shown.

2.1 Graph Layout Algorithms

Research on the best way to automatically visualize relational data with node-link diagrams has been active for several decades. The early primer by Di Battista et al. [23] provides a good overview of the field of graph layout algorithms, and researchers in this area continue to develop new approaches and improve on existing ones. More recent are comprehensive surveys of the area by Gibson et al. [31] and von Landesberger et al. [71]. Gibson et al. [31], review many graph drawing techniques developed over the past 35 years. These include force-directed, dimension-reduction and computationally improved multi-level graph drawing methods. Gibson et al. offers comparisons between and within layouts of each classes. They also speculate that the force-directed layout algorithms could be redesigned to place more emphasis in showing a particular structural feature or property of the network. For instance, LinLog, ForceAtlas, and OpenOrd layouts are already designed to aid the display of clustering in the network.

Von Landesberger et al. [71], in their review of graph layouts, focus on techniques used to specifically visualize large graphs. Starting with node-link and space filling techniques(treemap, sunburst[65], icicle plots[68] etc) to draw trees, they move on to the node-link based and the matrix-based representations for both static and dynamic graphs. They also review various user interaction techniques used for network visual analysis like graphical fisheye views [63] and guided panning(navigation along edges of selected node) [54].

Graph layout algorithms fall into different categories. Force-Directed algorithms [48] are based on a physical springs model and attempt to achieve minimum energy. Although, direct application of Force-Directed layout is computationally expensive for large graphs($O(|V|^2)$), Yifan Hu [41] proposed a multi-level force-directed algorithm for drawing large graphs efficiently. This layout algorithm consists of a multi-level subroutine that coarsens a large graph into a much smaller graph($G^0, G^1..., G^k$ where G^{i+1} is coarser than G^i) for faster run-time. Then a Force-Directed algorithm is applied using a Barnes and Hut [3] octree approximation that further reduces the computational cost. In this technique, when calculating the repulsive force on a vertex u exerted from vertices that are far away, the group of these far-away vertices can be treated as one supernode(placed at their center of mass), and the overall force can be approximated as the force between the vertex u and the supernode. Then the output of the applying FD-algorithm on the coarsest graph is prolongated back to its previous state(G^{k-1}). Prolongation typically involves putting nodes that merged into vertex u in G^i in random positions near u in G^{i-1}. Refinement follows prolongation where the FD-algorithm is again applied to G^{k-1}(however less number of iterations are needed as the network already has near optimal energy). This prolongation and refinement continues in similar until G^0 is reached.

FD-algorithms have been widely used for graph visualization. However, Brandes et al. [7] suggest the use of distance-based graph drawing algorithms when the requirement of visualization is to depict the graph-theoretic distances properly. Multidimensional scaling(MDS) algorithms are a popular class of distance-based graph layout algorithm. They project high-dimensional data onto low dimensional space, typically preserving the graph-theoretic distance between pairs of vertices [13]. These techniques usually use majorization to minimize the involved objective function. The principle of majorization given by Leeuw et al. [19] is to find a surrogate function that majorizes a particular objective function that is hard to minimize directly. Let the function to be minimized be $f(x)$. The majorization principle involves finding a simpler function $g(x, y)$ such that for all x, $g(x, y) \geq f(x)$ where y is a constant called supporting point. However, it is required that the surrogate function touches the curve of $f(x)$ at y. Then clearly, any value of x say $x*$ that minimizes $g(x, y)$ will also minimize $f(x)$. This process is repeated with the next surrogate function $g(x, y)$ but now $y = x*$. The process converges when $f(x) - f(x*)$ is less than a threshold. With respect to network, majorization technique is used to optimize the stress in the network, $\sigma(X)$ defined as $\sum_{i<j} w_{ij}(\delta_{ij} - d_{ij}(X))^2$ where δ_{ij} is the geodesic distance between vertices i and j, w_{ij} is the weight of the edge, and d_{ij} is the distance between them in the graph drawing.

Besides FD and distance-based graph layouts, other popular classes of layout algorithms have been developed like the orthogonal layout methods that insists on keeping all edges as horizontal or vertical [25] and circular layout methods that place all vertices on the circumference of a circle in a careful ordering to satisfy an aesthetic like minimizing crossings. Dougrusoz et. al. [24] proposes an implementation of circular layout that emphasizes clusters within graph. The algorithm given by Dougrusoz et.

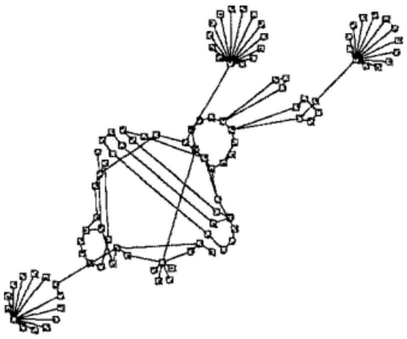


Figure 1: Sample drawing produced by the circular layout [24]

al. [24] first partitions the nodes into clusters. Each cluster is then placed on a circle of some computed radius. A virtual cluster is also formed that contains one node from each cluster. The nodes within the virtual cluster are connected if their corresponding clusters also have interconnections. Finally, all the clusters undergo edge crossings minimization starting from first, the virtual cluster, then the *main site* cluster(these are clusters whose node remain in the virtual cluster after each node of degree one is removed iteratively), and then finally all the remaining clusters. Figure 1 shows one of the graphs produced by the layout algorithm.

Other graph layout techniques have also been proposed to facilitate fast calculation of large networks and human-centered designs. Galán and Mengshoel [30] proposed a Neighborhood Beautification(NB) layout technique which has each node passing messages to its neighbors to adjust their position in each NB iteration. Each NB iteration consists of three message passing phases which conform the following aesthetics: minimize edge crossings, maximize edge length uniformity, and maximize angular resolution(the angles between any pair of edges from a node u to its consecutive neighbors is set to $360°/degree(u)$). An advantage over Force-Directed algorithms is that unlike them, the NB technique does not have the issue of getting stuck in local minima for large graphs. Also, for most of the tested graphs by Galán and Mengshoel [30], the NB technique needed a shorter runtime.

FD and distance-based layouts, focus on minimizing the energy or stress of the graph system rather than working on pleasing important aesthetics for graph readability like minimizing edge-crossings [58]. However, recently, Kieffer et al. [47] proposed an algorithm that was designed specifically to meet 9 aesthetic criteria. Keiffer et al. suggest that no algorithm produces layout of quality comparable to those made by humans. Thus, they first performed a user study where participants were asked to manually improve layouts of 8 graphs. The initial layout that was given contained a lot of edge crossings and bends. Then, a tournament style voting was used to rank the created layouts for each graph. Analyzing the correlation between various aesthetic value and rank of layouts, a set of 9 final aesthetics was given that included previously known aesthetics like minimizing stress, bends, crossings and two novel aesthetics: placing tress outside the layout and creating aesthetic bend points that empasize symmetry and places edges on opposite ends of nodes with degree 2. Following these 9 aesthetics, Keiffer et al. [47] propose the HOLA algorithm that has the following 4 stages:

1. Toplogical decomposition: Leaves are removed from the graph until none remain. The graph that is left is called the *core* graph.
2. Layout of the core: First, the core is laid out with a stress-minimizing algorithm and the overlap between nodes is removed. This is followed by orthogonalization where nodes are visited by decreasing degree and their neighbors are placed such that their is atmost one neighbor in each cardinal compass direction(NORTH, SOUTH etc). Then the chains in the graphs are visited and bends are introduced to minimize stress and maximize symmetry.

3. Tree layout and placement: Each tree is added back to the graph and laid out with symmetric layout. The direction of each tree is kept at one of the 4 cardinal or 4 ordinal directions(NORTHWEST, SOUTHEAST etc) such that their is minimum increase in stress.
4. Opportunistic improvement: This final steps tweaks the layout such that no obvious small change would improve it further. It includes, for example, a step where the layout aligns nearby nodes that were previously only almost aligned.

A final study showed that HOLA achieves better task performance than the best available orthogonal layout algorithm.

There are also specific algorithms for trees [61] and planar graphs [56]. Reingold et. al. [61] proposed an algorithm for drawing trees in which the drawings were aesthetically pleasing(nodes at same level were along a straight line, the lines through each levels were parallel and the parent node was centered over its left and right sons that were positioned necessarily to its left and right respectively). The idea was to visit each node in a postorder-traversal. At each node, the left and right subtrees were first superimposed at the root and then moved apart until their was a minimum separation between the nodes present at the closest contours of the two subtress.

This thesis uses three layout algorithms commonly implemented across a variety of different graph drawing systems, and each algorithm is an example of a different category. The first one is a multi-level force-directed algorithm by Yifan Hu [41], available as sfdp (scalable force-directed placement) in GraphViz [26] and as YifanHu in Gephi [4]. The implementation provided by Gephi was used. The second algorithm is one of the most effective dimensionality reduction algorithms, as shown in an experimental study by Brandes and Pich [7]. This algorithm applies classical multi-dimensional scaling, followed by stress majorization. This thesis uses the implementation provided in the MDSJ library in Gephi(https://gephi.org/plugins/#/plugin/mdslayout).

The third algorithm is a circular layout. It places all vertices evenly spaced along a circle and attempts to reduce the number of crossings. Again, Gephi's implementation of circular layout was used. The first two algorithms chosen are representative of the major methods (force-directed and MDS respectively), while the third algorithm (circular) is a good "generic" option provided by most graph drawing systems. Other algorithms and layout categories should be explored in the future; however, this thesis focuses on these algorithms due to their ubiquity in freely available graph drawing systems.

2.2 Graph Layout Comparisons

Most of the work that compares different graph layout algorithms focus on comparing their computation complexity [23] or how much they conform to commonly known graph aesthetics [59]. Helen et al. [58] first studied the importance of different aesthetic qualities in graph drawings through human experiments. They looked at five aesthetics: minimizing edge bends, minimizing edge crossings, maximizing the minimum angles between outgoing edges, maximizing orthogonality and maximizing the network symmetry. Human experiments were conducted to see which of the aesthetics influenced human understanding of graphs the most. In the experiment, each participant was shown 2 graph drawings for each aesthetic(the drawing represented a strong or weak presence of the aesthetic). For each drawing, the participant was asked a few questions(like length of shortest path between two given nodes) and the number of errors made and the time taken to answer was noted. Statistical tests found that minimizing edge crossings was most significant to reduce both errors and time, whereas maximizing symmetry was only significant for time and minimizing bends was only significant for errors.

This work was followed by Helen et al. [59] where they compared 8 automatic graph layout algorithms with respect to human performance. Although, the layouts were not directly compared by the extent to which they conform to aesthetics, the effects of aesthetics were assumed to be present since different layouts satisfy different aesthetics. Similar to the work by Helen et al. [58], the participant was shown 8 graph drawings corresponding to 8 different graph layouts and asked graph-based questions while their error and time to respond was noted. Tukey's pairwise comparison found that Seisenberger [64] drawing produced significantly more errors than Fructherman and Reingold [29], force-directed incremental algorithm [69], and Kamadi and Kawai [44] algorithm. Another study for edge crossings aesthetic was conducted by Stephen et. al. [49], more recently, in which the significance of edge crossings on task performance was studied for both small and large graphs with varying edge densities. Although two layout algorithms were used(FDP and MDS), their effect on task performance was not analyzed separately. A correlation between graph aesthetics and the stress(objective function minimized by MDS algorithm) in the two layout algorithms was also computed for 9 graphs to find that only crossings and energy showed some positive correlation.

The impact of the aesthetic of crossing angles on the task of finding the length of shortest path between two nodes was studied by Huang et al. [42]. To nullify the impact of confounding aesthetic factors a study was conducted with purposely-generated graphs that only differed in the angles edge crossings made with the path whose length was to determined(16 sets of drawings were made, each set had 7 drawings of same graph with different crossing angles: 10°, 15°, 20°, 30°, 50°, 70°, or 90°). The participant was shown 128 drawings and asked to compute path length. Statistically, it was found that crossing angles did affect response time, and that as crossing angle increase, the response time decreased. Further, Huang et al. [42] conducted a similar study with real-world graphs to show that the overall minimum and average crossing angle in the graph has significant impact on response time. Finally, a study for direct comparison of MDS variants(classical scaling, distance scaling, pivot MDS, and landmark MDS) and Force-directed layout was performed by Brandes et al. [7]. The study showed that for graph drawings that represented graph-theoretic distance, MDS layouts were better than FD layout for graphs whose graph-theoretic distance is well representable in low dimension. However they fail to create good quality drawings for small world and scale free graphs. Moreover, it was found that majorization process led to lowest stress values when initialized with output of classical scaling. The final stress values gets progressively worse when starting from the output of fm3 [33], grip [66], hde [35] to random initialization. Finally, PivotMDS [8] and LandmarkMDS [21] were tested with varying strategies in picking up the pivot points to find that in most cases, PivotMDS performed better with *maxmin* strategy in which the next pivot point is chosen such that it is furthest from current pivot points.

This thesis directly compares different graph layouts with respect to perception of properties. No control over aesthetics was made when generating the graph layouts.

2.3 Graph Properties, Graph Mining and Graph Visualization Systems

The structure of any graph can be characterized by a set of graph-theoretic properties, the most trivial being order (number of vertices) and size (number of edges). These properties describe the nature of a graph, independent of the way it is drawn – other examples include density (the proportion of the number of edges to the maximum possible), diameter (the longest shortest path between pairs of vertices), number of connected components (subgraphs with a path between any pair of vertices), vertex degree distribution (represented as a histogram), and transitivity (the extent to which vertices are clustered together by edges). The focus of my thesis was on two graph properties: the graph density(GD) and the average local clustering coefficient(ALCC). The local clustering coefficient of a vertex is given as the ratio of the number of edges between its neighbors to the total number of possible edges between them. Its average value over all the vertices in the network is defined as the network's average local clustering coefficient.

Graph mining (as a type of structured data mining) is the activity of identifying patterns in graphs. Rehman et al. provide a comprehensive review of graph mining approaches [60]. Various approaches for graph classification exists like Callut et. al. [10] proposed a technique based on D-Walks that can be used to find the unknown classes of nodes given some labeled nodes in the graph. The technique defines D-Walks as a random walk that starts and ends with node having some label C such that no other node in the walk is labeled C. Given all such walks for class C, the betweenness $B(q, C)$ is the expected number of times the node q is reached during the walks. Then the label of q is simply the class for which betweenness is maximum given same prior probability. Graph clustering is another important graph mining task where vertices of graph are grouped in such a way that there are many edges withing clusters as compared to number of edges between clusters. Blondel et al. [5] presented a heuristic method based on modularity optimization that can extract the community structure of large graphs in short run-time. The quality of network partitioning is quantified by a scalar value called modularity(higher value represent better partitioning). Initially,

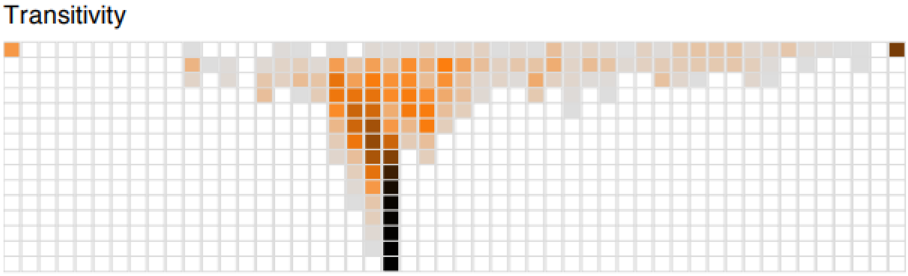


Figure 2: Transitivity facet for the largest component of a co-authorship graph [43]

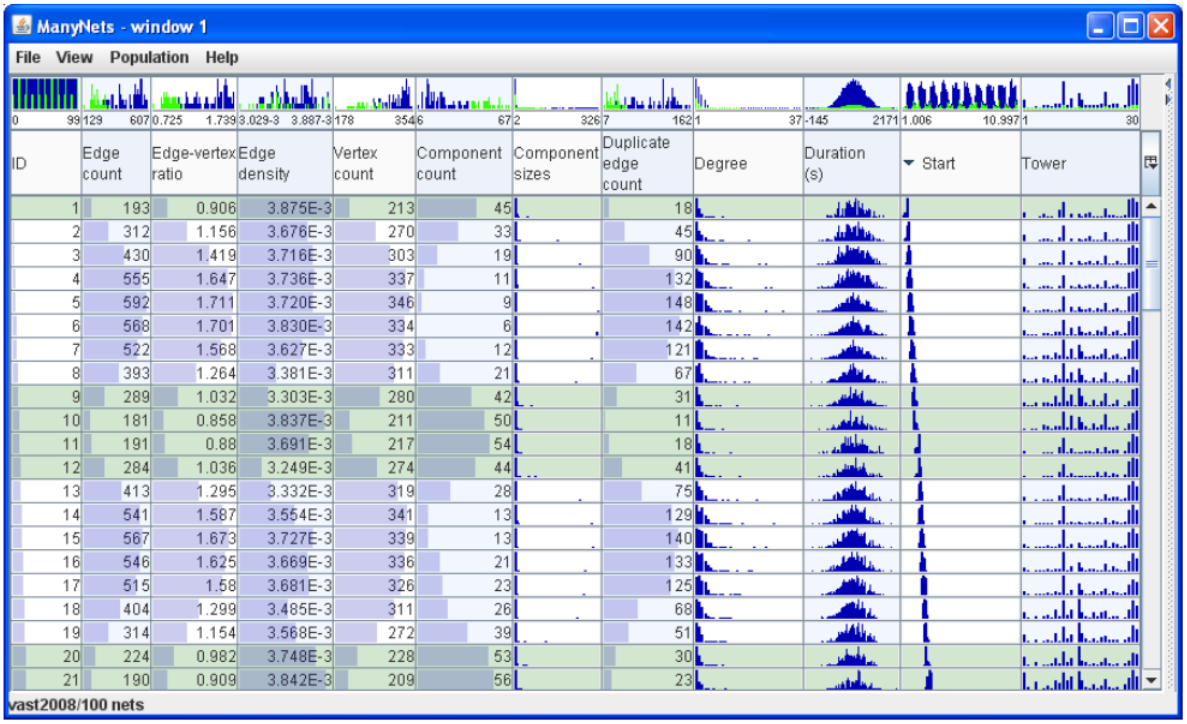
every node is assigned a different community. Subsequently, the method undergoes iterations consisting of two phases: the first phase consists of traversing every node and assigning it the community of its neighbor that increases the modularity the most(the community to which the node belongs remains unchanged in the case of no increase in modularity), while the second phase merges nodes belonging to the same community into a single node. Iterations stop when no change in modularity is observed.

Chakrabarti and Faloutsos address the issue of generating synthetic graphs that match the patterns within real-world (especially large) graphs [12]. In particular, they emphasize the importance of being able to say that two different graphs are similar to each other with respect to given properties. The PEGASUS system handles very large graphs (with "billions" of vertices), to find connected components, diameter, and vertex proximities [45].

Visualization is commonly used for depicting the values of graph properties, although this is not always using the common node-link diagram. However, the methodology used in my thesis is appropriate for comparing visualizations of any type(not just node-link diagram). Kairam et al. describe Graph-Prism [43], which visualizes several graph properties using stacked histograms and color encoding, including connectivity, transitivity, and density. Figure 2 shows one of the $B-Matrix$ facets that visualizes transitivity(fraction of closed triads to all triads in a graph) patterns for the largest component of a co-authorship graph. The color intensity of an element $B_{l,k}$ of the matrix represents the percentage of nodes with transitivity k when considering the l-level subgraphs formed around each node. The l-level subgraph of a node t includes nodes that are l hopes away from t. The facets were augmented by a node-link diagram having linked selection with the facets. Their user study asked participants to choose a synthetic graph (from a set of ten, all represented in the GraphPrism diagram format) that best matched the summary statistics of a real graph.

Visual analysis of large graphs represented by adjacency matrices is considered by van Ham et al. [70]. They propose a technique for aggregating adjacency matrices of large graphs while still maintaining important connectivity information within the network. The method uses some predefined hierarchy based on node attributes(like position within organization or geographic information) for aggregation. At each step of aggregation, the vertices at the lowest level of the hierarchy are eliminated such that if two connected vertices have different ascendants(one level up) in the hierarchy, then the edge count between the ascendants is increased by one in the aggregated matrix. In addition to the connectivity measure, van Ham et al. define two more measures that help users identify potentially interesting patterns in the network: *asymmetry* defined as $CC(X,Y)/CC(Y,X)$ where $CC(X,Y)$ denotes the edge count from node X to Y in the aggregated matrix and *deviation from expected* which measures the difference between the observed edge count between two nodes and the expected edge count between them if the edges had been distributed randomly in the network.

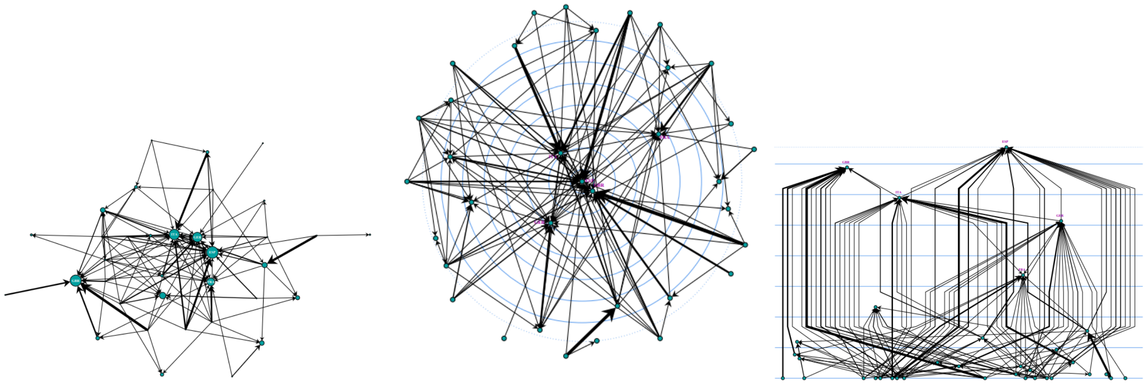
The need for examining multiple networks at once, for different graph properties,



ID	Edge count	Edge-vertex ratio	Edge density	Vertex count	Component count	Component sizes	Duplicate edge count	Degree	Duration (s)	▾ Start	Tower
1	193	0.906	3.875E-3	213	45		18				
2	312	1.156	3.676E-3	270	33		45				
3	430	1.419	3.716E-3	303	19		90				
4	555	1.647	3.736E-3	337	11		132				
5	592	1.711	3.720E-3	346	9		148				
6	568	1.701	3.830E-3	334	6		142				
7	522	1.568	3.627E-3	333	12		121				
8	393	1.264	3.381E-3	311	21		67				
9	289	1.032	3.303E-3	280	42		31				
10	181	0.858	3.837E-3	211	50		11				
11	191	0.88	3.691E-3	217	54		18				
12	284	1.036	3.249E-3	274	44		41				
13	413	1.295	3.332E-3	319	28		75				
14	541	1.587	3.554E-3	341	13		129				
15	567	1.673	3.727E-3	339	13		140				
16	546	1.625	3.669E-3	336	21		133				
17	515	1.58	3.681E-3	326	23		125				
18	404	1.299	3.485E-3	311	26		68				
19	314	1.154	3.568E-3	272	39		51				
20	224	0.982	3.748E-3	228	53		30				
21	190	0.909	3.842E-3	209	56		23				

Figure 3: Manynets [28] visualizing a time-sliced cell-phone network. Each row represents networks of calls in a 5-hour period

comes in many instances, like viewing temporal slices of evolving social networks, distribution of "motifs" in biological network, viewing clusters within networks etc [28]. The tabular interactive tool ManyNets [28] can visualize a large number ("up to several thousand") of graphs, allowing visual comparison of graphs based on properties. An example visualization is shown in Figure 3, where each row represents a single network. The columns contain either user-defined(domain dependent properties like Node ID) or default network properties(edge count, graph density etc). The table is augmented with node-link diagram generated via SocialAction [57]. Each cell with scalar values contain horizontal bars that aid in comparison between graphs. Properties with distribution like node-degree are summarized with histograms within cells. It is possible to start from a single network and subsequently break it down into

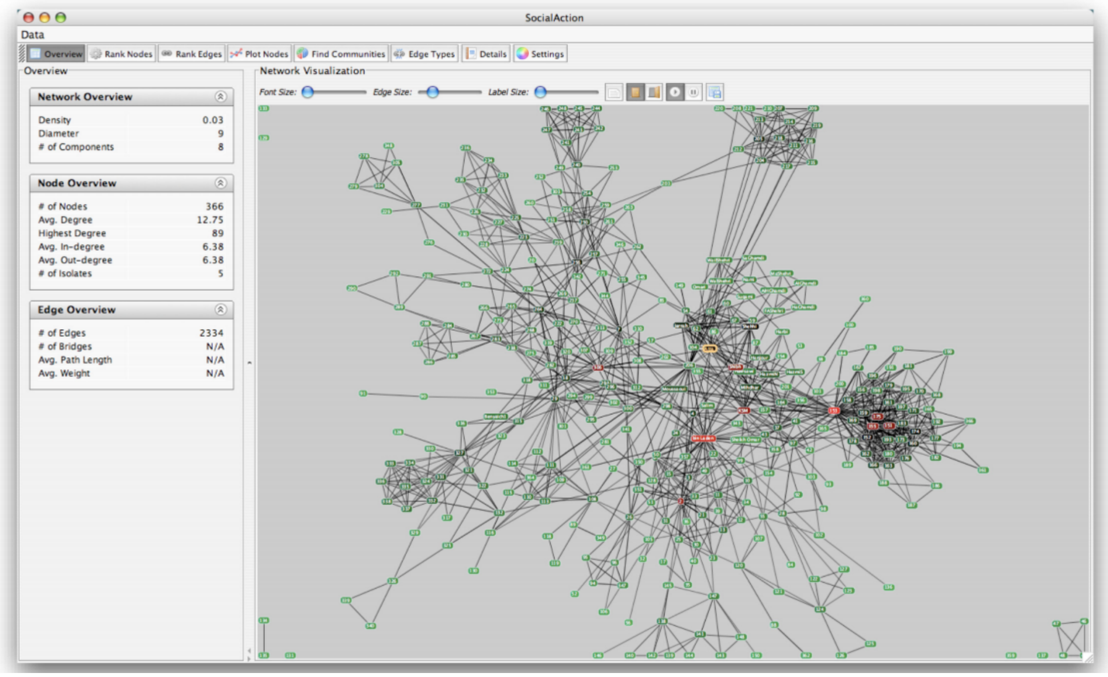


(a) Uniform layout (b) Spectral layout (c) Layered layout

Figure 4: Graph layouts provided in Visone [9]

more and more smaller networks to compare network properties at different scales. Although, only a single node-link diagram is shown at a given time in ManyNets, it is evident that the system would benefit hugely if the user was provided with multiple node-link diagram that made their graph properties evident by the layout used. In the case study presented by Freire et al. [28], the network analyst studied a trust network where each edge was labeled by the trust between source and target node. Deep insights were found after splitting the network like the pairs and triads within the largest component had lower trust values among on average than the ones found in other isolated components.

Unlike GraphPrism and ManyNets, Visone [9], on the other hand, solely depends on node-link diagram. Visone tries to represent graph-theoretic indices used in *Social Network Analysis* such as degree, betweenness, closeness, eccentricity, pagerank etc. The value of these indices for each node is conveyed via visual cues in node-link diagram like via position or size of the node while various graph aesthetics are also followed to improve readability of graphs. Visone provides four different layout options: *uniform layout*(Figure 4a) places every pair of vertices such that their distance is in proportion to their geodesic graph distance giving us information about properties like average path length and diameter, *spectral layout*(Figure 4b) that enables display of symmetry in graphs, *layered layout*(Figure 4c) where the node's Y-coordinate represents its ranked index value, while the X-coordinate is varied to minimize edge



Network Overview

Density	0.03
Diameter	9
# of Components	8

Node Overview

# of Nodes	366
Avg. Degree	12.75
Highest Degree	89
Avg. In-degree	6.38
Avg. Out-degree	6.38
# of Isolates	5

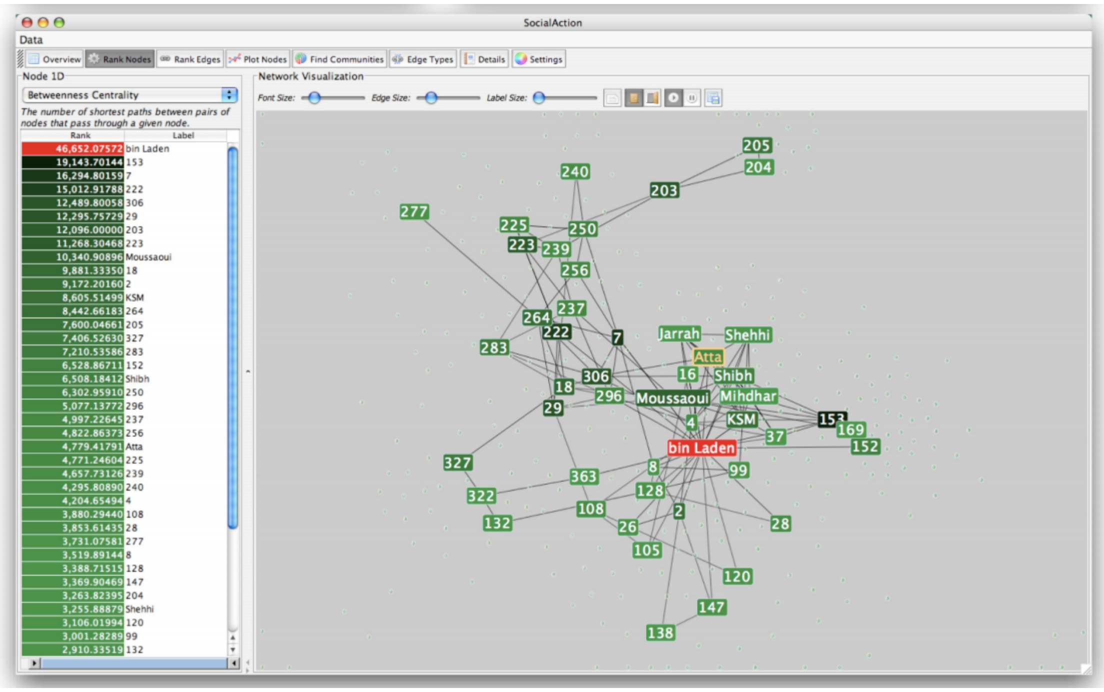
Edge Overview

# of Edges	2334
# of Bridges	N/A
Avg. Path Length	N/A
Avg. Weight	N/A

Figure 5: Initial view of the network where each node is shown and colored based on its betweenness centrality(color ranges from green to red representing low to high betweenness value) [57]

crossings and finally, *radial layout* that try to achieve a uniform distribution of nodes with minimum edge crossings.

Another tool that uses node-link diagram is SocialAction [57] in which visualization is integrated cohesively with graph statistics. While the visualization simplifies statistical results and provides patterns' discovery, statistics helps in the overall comprehension of a large network. Initially(Figure 5), the user gets the overview of the whole network statistically(being shown measures like density, diameter, average degree etc) and visually(by a force-directed layout of the graph). Moreover, each node can be colored with varying intensity from green to black representing a selected statistic's value for the node. Filtering the network is possible by eliminating nodes whose statistic's value does not fall in a given range(Figure 6) which further reveals



Betweenness Centrality

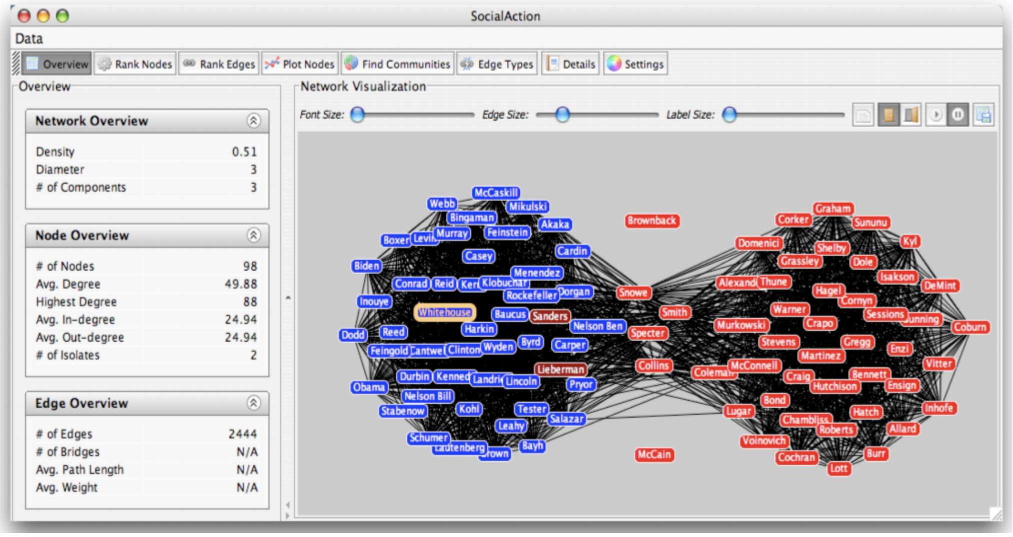
The number of shortest paths between pairs of nodes that pass through a given node.

Rank	Label
46,652.07572	bin Laden
19,143.70144	153
16,294.80159	7
15,012.91788	222
12,489.80058	306
12,295.75729	29
12,096.00000	203
11,268.30468	223
10,340.90896	Moussaoui
9,881.33350	18
9,172.20160	2
8,605.51499	KSM
8,442.66183	264
7,600.04661	205
7,406.52630	327
7,210.53586	283
6,528.86711	152
6,508.18412	Shibh
6,302.95910	250
5,077.13772	296
4,997.22645	237
4,822.86373	256
4,779.41791	Atta
4,771.24604	225
4,657.73126	239
4,295.80890	240
4,204.65494	4
3,880.29440	108
3,853.61435	28
3,731.07581	277
3,519.89144	8
3,388.71515	128
3,369.90469	147
3,263.82395	204
3,255.88879	Shehhi
3,106.01994	120
3,001.28289	99
2,910.33519	132

Figure 6: The network is filtered showing only those nodes whose betweenness centrality falls within user-specified range [57]

deeper insights on the network. For example, one of the case studies performed using SocialAction involving senatorial voting patterns revealed that when edges whose weights was less than 180 were removed, where weight represented how often two senators voted together, a clear partisanship could be observed(see figure 7 where blue and red nodes represent Democratic and Republican senators respectively).

Various tools like Gephi [4], GraphViz [26], NodeXL etc are also worth mentioning as they provide several graph layout options to choose from for visualizing graphs. In addition, these tools have the feature to encode attribute information about nodes and edges via their color or size. Gephi by default provides 6 graph layouts(and more than 10 layouts as plugins) including Frutherman-Reingold, Yifan Hu, OpenOrd etc. It also provides options like zooming and panning over displayed graph, filtering out nodes



Network Overview	
Density	0.51
Diameter	3
# of Components	3

Node Overview	
# of Nodes	98
Avg. Degree	49.88
Highest Degree	88
Avg. In-degree	24.94
Avg. Out-degree	24.94
# of Isolates	2

Edge Overview	
# of Edges	2444
# of Bridges	N/A
Avg. Path Length	N/A
Avg. Weight	N/A

Figure 7: The social network of the U.S. Senator voting patterns analyzed during a case study evaluating SocialAction [57]

and edges based on attributes, and resizing or coloring nodes based on attribute values. GraphViz similarly gives many layout options like dot(layered hierarchical graph drawing), neato(equivalent to MDS layout), fdp(equivalent to Frutherman-Reingold layout), sfdp(multi-scale fdp for drawing large graphs fast) etc.

Finally, Vizter [37] also produces nodelink drawings of graphs, in this case identifying and highlighting communities in social networks on demand. Vizter was actually designed for the end-users of social networking systems to enable discovery and awareness of their on-line community. It uses a FD spring embedding layout for the node-link diagram, where nodes represent members of the system and the links represent their friendship. An egocentric view of the network is shown with the individual's node in the center and his immediate friends around the node. Users can expand any node to display its immediate friends as well. Although, the expanded node's position remains fixed, rest of the nodes move via FD-layout. Moreover, the layout is computed in real-time(through Barnes-Hut algorithm and numerical integration), so the user has the option of moving a node, while the rest of nodes adjust their positions accordingly. Several other interaction and visualization features are provided to aid discovery including search by attribute value(e.g. age), X-Ray mode to show a selected attribute's value for each node via discriminating node color, curved borders(blobs) around communities detected via Newman's community detection algorithm [55] etc.

This prior work reinforces the need for properties of graphs to be made evident to users – either through a supplementary visualization (as in ManyNets [28]), or as part of the node-line depiction (as in Visone [9]). Being able to compare two graphs according to their properties is particularly important when synthetic graphs are to be used in place of larger real-world ones [12]. Commonly, no studies have considered this question from an empirical human perception perspective. Specifically, can humans detect differences in the properties of graphs when depicted as graph drawings and are there particular layout algorithms that best support the visual perception of graph properties?

2.4 Perception of Properties: Just Noticeable Difference

Experiments in the psychology of perception use the "Just Noticeable Difference" (JND) as a means of determining the minimum distinguishable property difference between two stimuli [67]. Such experiments ask participants to indicate which of the two stimuli has a greater value of a given measurable property (e.g., which square is greener). The JND is the value difference between two stimuli that is noticed at least 50% of the time by participants [50].

The method of constant stimuli is commonly used to get the Just Noticeable Difference. This method involves showing the participants two stimuli with different values several times and asking them which stimulus has the larger value. One of the stimuli, called the *constant stimuli*, has a fixed value throughout the conducted trials while the other stimuli, called the *comparison stimuli*, varies in its value. Now, let X_t be the value of *comparison stimuli* that is correctly selected to be larger than the constant stimuli in $t\%$ of the trials(that involved the comparison stimuli). The JND is then computed as the value $(X_{75} - X_{25})/2$ [11]. The JND experimental method can be applied to any type of perception. For example, Goodfellow [32] studied the perception of sound, vision and touch. For each of them, the participants were presented with the stimuli three times in each trial. The first two stimulus were presented one second apart. The time interval between the second and last stimuli was initially kept large and steadily decreased until participant reported that the two time intervals were equal and then further decreased until participant again noticed a difference. Wilson et al. [73] studied thermal perception. The participants were initially kept at a neutral temperature of 30°C. They were then subjected to a warmer or colder stimulus for 10 seconds and asked to indicate as soon as they felt a change in thermal stimulation. The temperature difference between the skin and stimuli gave the thermal JND. In their study, JND was calculated for varying rate of change of temperature, stimulus temperature, direction of change(warm or cool), and body location where stimulus was applied. Weber's law states that if P is the property value of a stimulus, the ratio of the JND to P will be constant.

Weber' law is a historically important psychological law that relates the perception of change in a given stimulus with the magnitude of that stimulus. Weber's law states that if P is the property value, being perceived, of a stimulus, the ratio of the JND to P will be constant(k called Weber fraction). The Weber fraction represents the amount of change needed for perception at a given stimuli. So for two different

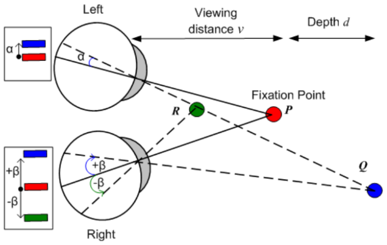


Figure 8: Geometry of binocular stereopsis [20]

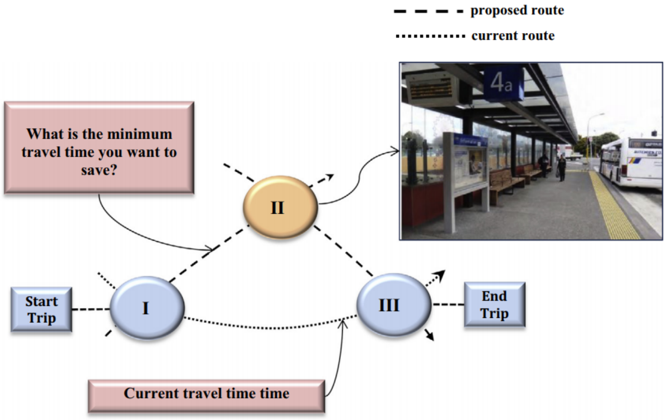


Figure 9: An illustration of the hypothetical scenario used in the user study by Camacho et al. [11]

perception model, at same stimulus, the one with higher Weber fraction will require more change in the stimulus for it to be noticeable. If perception of a property follows Weber's law, then it is possible to compare its perception under different conditions. For example, Harrison et al. [36] compares 9 visualization techniques(scatterplot, parallel coordinates, stacked area etc) for the perception of correlation between two variables.

While Weber's law experiments typically focus on low-level perceptual properties, they can be applied to any stimuli for measurable properties. De Silva et al. [20], for example, looked at the 3D stereoscopic vision. The study derives a JND in depth as seen by a viewer watching a 3D video. A 3D simulation is achieved by a process called binocular stereopsis(figure 8) in which the eye of the viewer is fixated on a point P at viewing distance μ. Then the images from point R and Q are casted on eyes' fovea at different angles to the image casted by point P that indicates depth information to the brain. The study calculates a model for JND that depends on μ and the simulated depth. The common JND calculation methodology is used where two objects are shown initially at the same depth and then one object's depth is gradually increased or decreased until the observer signals that they sensed a change. Cornelissen et al. [18] explores visual biases in the perception of body weight. The study shows that weber's law is followed in the perception of body weight by humans as it progressively gets more difficult to discriminate weight difference(i.e. JND in weight increases) as the weight of the pair of bodies increases. The experiment involved 8 reference(the fixed stimuli in pair of bodies presented) weight levels. Two different types of stimuli were used. One type was CGI images(such that factors like height, complexion etc could be maintained) and the other was real images of people. The study found that JND was higher for real images. Camacho et al. [11] considered the perception of viscosity in beverages. The experiment consisted of varying the viscosity of the liquid while other sensory properties like taste and flavor remain unchanged. The method of constant stimuli was used with 6 duplicated directional 2-alternative forced choice(AFC) tests. In each AFC test, the participant had asked to choose the thicker liquid among the pair of reference and comparison stimuli. Among the 6 AFC tests, 3 had liquid with higher viscosity than the reference liquid while the rest had lower viscosity. Chowdhury et al. [14] looked at travel time and route taking decisions. In the Weber's law experiments that were performed, the participants were given two hypothetical scenarios(one of them is shown in Figure 9) and asked what was the minimum travel time(or cost) they wished to save for them to take the route involving transfer(route through node two in Figure 9). Both scenarios had different comfort amenities at the interchange. Although experiments showed that the Weber's law was not followed, the Weber fraction on average was found to be lower for higher comfort interchange. This meant that the participants were willing to sacrifice on travel time and cost savings for more comfort.

In the field of data visualization, the psychophysical methods of calculating JNDs can similarly be applied to the perception of various properties of data including correlation. Several visualization techniques are available such as scatterplots, parallel coordinates plots, donut charts etc that can be used to communicate properties like correlation. The perception of correlation in scatterplots by the JND methodology was studied by Rensink et. al. [62]. All the scatterplots used in the experiment were $300X300$ pixels containing 100 normally distributed points along the 45° line. The participants were tested on both precision(reflecting the discrimination of correlation between plots) and accuracy(direct estimation of the correlation value). For testing precision, for each *base* correlation, the participant was shown two scatterplots side-by-side and asked to pick the one with higher correlation(one of the scatterplots showed the *base* correlation, while the other showed the *comparison* correlation). The initial difference between the plots was kept at 0.1. Each such trial followed the staircase procedure of calculating JND. In the staircase procedure, when the participant chose the scatterplot correctly, the difference in the correlation of the two scatterplots was decreased by 0.01 in the next trial to make the task more difficult. However, when the participant chose incorrectly, then the difference was increased by 0.03, thus making the task of picking the scatterplot with higher correlation more easier. For each judgment, the scatterplots were randomly generated at that time. The experiment first involved conducting at least 24 trials. Then the 24 trials were divided into 3 consecutive non-overlapping sub-windows of 8 trials each. Subsequently, the *comparison* correlation values for the trials were taken and the ratio of average variance within the sub-windows to the variance of averages of the sub-windows was calculated. The experiment stopped if this ratio was found to be lower than 0.25. Otherwise, more trials were conducted until this condition was achieved for the last 24 trials. However, the procedure stopped regardless of the condition, after 50 trials.

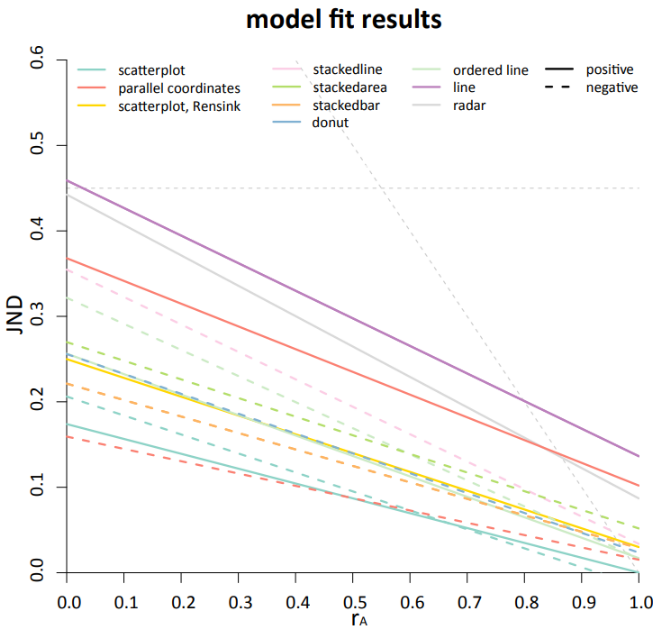


Figure 10: Regression results for multiple visualization techniques for the perception of correlation [36]. The figure also includes the regression model fit given by Rensink et al. [62]

After the experiment ends, the average difference between the *base* and *comparison* correlation for the last 24 trials is taken as the JND.

The same experimental methodology was adopted by Harrison et al. [36] where a large-scale crowd-sourced experiment with 1687 users was conducted to investigate the perception of correlation in nine common visualizations(scatterplots, parallel coordinates etc). Harrison et. al. describe that it is possible to compare the perception of correlation between different visualizations if they all follow Weber's law. The Weber model also provides a baseline using which the effect of other design elements like point color, brightness, size etc can be studied. The experiment, similar to the one followed by Rensink et al. [62], follows the staircase procedure. However, the perception of negative correlation values was also studied for the 9 visualizations.

Hence, each participant was assigned a visualization, direction(positive or negative correlation), and a pair of r values for the staircase procedure. In addition, Harrison et al. also considered the two limitations associated with the staircase procedure: the ceiling effect and the chance boundary. The ceiling effect occurs when the staircase procedure reaches an upper limit of correlation($r = 1$), before the real JND is reached. In this case, the computed JND is not accurate. The chance boundary for the staircase procedure was calculated by running its simulation 10000 times such that in each trial, a scatterplot was randomly selected to have a larger correlation with 0.5 probability. The resulting JND of 0.45 was then used as a performance threshold for the participant. A JND above this threshold indicated that the participant had not reliably perceived the correlation. Using the data collected from the crowd-sourced experiment, Harrison et al. found that all the visualizations followed Weber's law with minimum $r^2 = 0.74$. Also the Weber's model for positive and negative correlation was found to be different and a comparison among all the regression plots(Figure 10) revealed that scatterplot(positive and negative) and parallel coordinates(negative) had lower average JND values and were better for perceiving correlation.

We apply the same JND methodologies for the perception of graph properties. We aim to determine whether the perception of two graph properties (graph density and average local clustering coefficient) follow Weber's law, and which of the three graph layout algorithms (FD, MDS, and circular) best supports the perception of property differences.

2.5 Graph Generators

Generating synthetic graphs with required property values is important for perception experiments as real graphs might not give the entire range of stimulus needed for calculation of JND in Weber's law experiments. Various graph generating mechanisms exists including the Erdos-Renyi graph model [27], the Watts-Strogatz graph model [72], the Barabasi-Albert(BA) graph model[2] etc.

Each graph generator attempts to generate certain class of graphs: BA model generates power-law graphs, Watts-Strogatz generates small-world graphs while Erdos-Renyi generates graphs with binomial degree distribution. The BA graph model [2], starts with a network with small number of nodes and no edges. At each time step t, a vertex v is added to the graph with m edges. Each edge of v is connected to an existing vertex of graph via Preferential Attachment(PA) in which the probability of connecting with a vertex u is proportional to degree of u. Hence, there is a higher chance for a new vertex to attach to a node with a higher degree than a lower one, which leads to a power-law degree distribution. Holme et. al. [40] extends the basic BA model by adding an additional Triad Formation(TF) step. In the TF step, if an edge has been added between new vertex v and existing vertex u in a previously preformed PA step, then one more edge is added between the vertex v and a random neighbor of w. The PF steps causes a power-law degree distribution whereas the TF step increases the number of triads at u and hence the average local clustering coefficient of the graph is increased. The TF step can be made to occur with a given probability which makes it possible to tune the average local clustering coefficient of the generated graph. Watts-Strogatz graph model [72] generates graphs with small world properties i.e. their characteristic path length is small like the Erdos-Renyi graph and their clustering coefficient is high like a regular lattice. The procedure starts with a ring of n vertices, each connected with k neighbors and then iterates through all vertices until all edges have been tried once for rewiring with probability p. In the first iteration, the edge connecting the visited vertex to its first nearest neighbor is selected for rewiring. In the subsequent iterations, the edge connecting the vertex to its next nearest neighbor is selected. During rewiring, the edge is connected randomly to any vertex in the graph. Watts et. al. [72] showed that small-world graphs were generated for some values of p.

Another set of techniques for generating graphs with required properties involve starting with a graph with some fixed property values and then subsequently rewiring it such that the desired graph property is controlled while other properties remain constant. Bansal et. al. [1] proposed a Markov chain simulation algorithm that generates simple connected random graphs with desired clustering coefficient while keeping the graph's degree sequence fixed. In each rewiring stage, five nodes x, y_1, y_2, z_1, and z_2, connected by edges (x, y_1), (x, y_2), (y_1, z_1), (y_2, z_2), are selected in input network. Then the outer edges are swapped to create edges (y_1, y_2) and (z_1, z_2) which changes the number of triangles in the network(and hence clustering coefficient). However, the degree of involved nodes remain unchanged. Menglin et. al. [52], provide a method to generate a graph with N vertices, E edges and a desired clustering coefficient. The algorithm begins with a Erdos-Renyi connected graph with desired number of edges. Within each iteration, an edge is first deleted and then added back between any two unconnected nodes such that both deletion and addition move the clustering coefficient of the graph towards the desired value.

This thesis uses the mentioned Holme et al. [40] approach for generating networks with a given number of nodes, number of edges and a particular average local clustering coefficient value. Moreover, a novel algorithm to generate a network with a given graph density was devised. Briefly, the algorithm starts with a path graph(a connected graph with minimum number of edges) and then keeps on connecting two randomly chosen vertices until the graph density reaches within a threshold distance of the required value.

Chapter 3

METHODOLOGY

The goal of this thesis is to quantitatively evaluate human perception of selected graph properties and compare the way in which different graph layout algorithms support the perception of these properties. Five user studies were conducted to quantify the just noticeable differences (JNDs) of two graph properties and measured how these JNDs fit as a function of the property value in three different layout contexts. This thesis hypothesizes that some graph layout algorithms will be better than others at revealing graph properties.

In order to test this hypothesis, we chose two graph properties, graph density (GD) and average local clustering coefficient (ALCC), and three graph layout algorithms, Force Directed Layout [41], Circular [24], and Multi-Dimensional Scaling [7] were chosen. Graph density was chosen due to its simplicity to explain to participants and importance in expressing the connectedness of a graph. Previous studies have also explored the ability of humans to perceive density within plots [16] indicating that (minimally) perception of graph density should be measurable. Average local clustering coefficient is a measure of the degree to which vertices in a graph tend to cluster together and is commonly analyzed with respect to small world networks. Given the recent importance of visualizing small world networks in the context of social network graphs, average local clustering coefficient was chosen as our second property to measure.

Two experiments were performed. Experiment 1 consists of three user studies analyzing the perception of graph density over three layout algorithms. Experiment 2

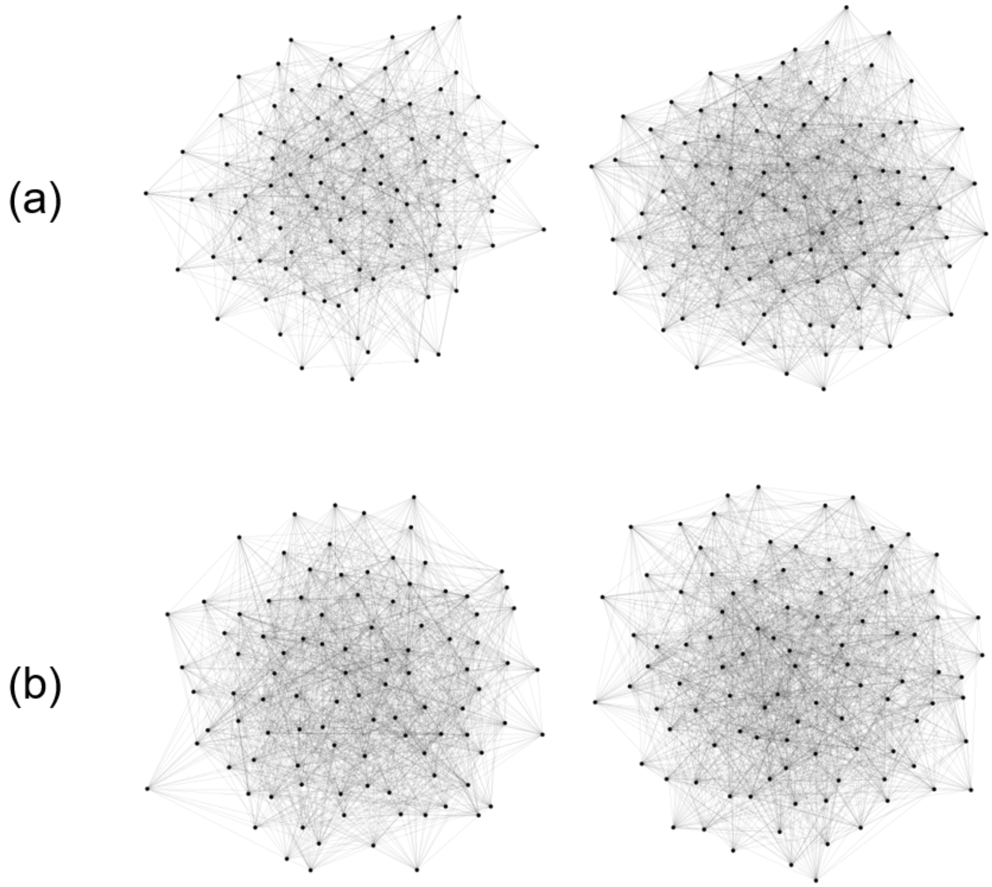


Figure 11: (a) A sample starting comparison with target value $d = 0.2$ on the left and $d = 0.3$ on the right. Participants were asked to choose which one has a higher graph density. (b) The staircase procedure converges to the JND by gradually making comparisons more difficult: $d = 0.3$ on the left and $d = 0.28$ on the right.

consists of two user studies anayzing the perception of the average local clustering coefficient over two layout algorithms.

3.1 Experimental Method

Based upon the perceptual analysis of correlation done by Rensink and Baldridge [62] and Harrison et al. [36], this thesis applies the same adaptive psychophysical method, a staircase procedure, to derive JNDs for the perception of the graph properties. For each user study, a list of evenly separated property values in a possible value range is designed as *base values*. Each base value graph drawing is compared to a graph drawing with another property value using two approaches (*above* or *below*). This means, for each property base value, the JND will be approached from above and below in two sets of comparisons using a staircase procedure.

To evaluate JND, participants are shown two stimuli side-by-side (in this case node-link graph drawings generated by the same algorithm), and participants will be asked to indicate which graph has a higher property value (see Figure 11 as an example of graph density). One of the two graph drawings has the base value, and the other graph drawing represents a smaller property value in the *below* approach or a larger property value in the *above* approach. Initially, the difference in the value of the properties between the two stimuli is set by a specified *initial-difference*. There is also a *step-size* parameter which is tuned to adjust the two stimuli in the staircase procedure. For example, if a participant is assigned the base value of 0.5, with a *below* approach, and the initial-difference is 0.1 and the step-size is 0.01, the first pair will have property values of 0.5 and 0.4. The participant is asked to select the graph drawing with the higher value of the property. Using a similar staircase methodology to Harrison et al. [36], if the correct choice is made (0.5), the next pair presented will be 0.5 and 0.41 (a decrease of 0.01 of the difference, making the task more difficult). If the participant answers incorrectly (0.4), the next pair will be 0.5 and 0.37 (an increase of 0.03 of the difference, making the task easier). A similar procedure is followed for the above approach. The distance changes allow the process to converge to a state such that the difference in properties between two side-by-side stimuli can be discriminated 75% of the time.

The staircase procedure ends when it reaches one of the following two conditions:

(1) the participant has done a maximum number of judgments (e.g, 50), or (2) the participant reaches the JND indicated by a convergence criterion. This convergence criterion is the same as that of Harrison et al. [36] and Rensink and Baldridge [62]. Specifically, the convergence criteria uses the last 24 judgments to determine if the participant's ability to discriminate between the two property values from the given graph drawings has stabilized. To test the stability, these 24 judgments are segmented into 3 groups of 8 sequential judgments in each, and an F-test($F(2, 21), \alpha = 0.1$) is applied on these 3 groups. When the F-test shows no significant difference between these three groups, convergence is assumed and the staircase procedure ends. No matter which ending condition a participant reaches, the final JND of the base value and the approach (*above* or *below*) is calculated by taking the average of the difference between the stimuli over the final 24 judgments.

For each trial in the experiment, the location of the base value graph is randomized (i.e., the base value graph will randomly be the left image or the right image). For each property value (both the base values and the interim values), we created a large number of possible graph drawings was generated, thus mitigating against any possible learning effect or unanticipated confounding factors. All graph drawings are pre-computed and images used are chosen through random selection from our pre-generated graph drawing pool. The same methodology is applied to both the graph density and the average local clustering coefficient experiment. Experiments differ only in the choice of base values, initial-difference, and step-size which were designed after preliminary experiments were conducted to determine feasibility.

3.2 Data Analysis Method

Once data is collected, the goal is to determine if the data can be modeled using Weber's law. Prior to model fitting, the data was analyzed to remove any base value and approach conditions that suffer from the ceiling effect, which means the obtained JND is constrained by the range of the property value available in our experiment and therefore we could not observe the true JND. The ceiling effect is quantified in our experiment over the last 24 judgments. If over 50% of the judgments are performed on data within .05 of the upper or lower bounds of the data range, then participants are bounded by a ceiling effect. The *hit rate* is the percentage of participants that are bounded by a ceiling effect within a group (value, approach). Outliers outside of 3 median absolute deviations from the median in each base value and approach condition are also removed prior to model fitting.

After data cleaning, results from the experiment were analyzed in a three-step process. First, the JNDs in each user study (where an user study consists of a single graph layout algorithm being tested for a single graph property) are tested by fitting a model of Weber's law. The fitting methodology is modeled on that of Rensink and Baldridge [62] and Harrison et al. [36]. Average JNDs are calculated for each base value and approach condition. The property values used in the model are adjusted from the base value by adding (or removing) $0.5 \times JND$ within each base value and approach condition. To test for Weber's law, the JNDs and the adjusted values are fit using a linear regression model. Next, individual data points are fit without averaging and property value adjustment using linear regression with both continuous and categorical variables, following the methodology by Kay and Heer [46]. This is followed by an analysis to test if some data transformation is required for an adequate model. Finally, results are compared between layouts using Mann-Whitney U test to see if the perception ability in different layout algorithms has comparable distributions.

Chapter 4

EXPERIMENT 1: GRAPH DENSITY

The first experiment explores the effects of graph layouts on perceiving graph density. This experiment only considers undirected graphs where graph density is defined as:

$$D = \frac{2|E|}{|V|(|V|-1)} \tag{4.1}$$

where E is the number of edges and V is the number of vertices in the graph. Graph density describes how dense a graph is, and, for a fixed number of vertices, the more edges a graph has, the higher the graph density value. In a simple and connected graph, the maximum D is 1 (for complete graphs) and the minimum D is $\frac{2}{|V|}$ (for having |V| - 1 edges connecting all vertices).

In this experiment, the goal is to quantitatively analyze people's perception of graph density given different layout algorithms. To achieve this goal, mixed design experiments were conducted such that just noticeable differences (JNDs) were collected by asking participants to compare the graph drawings of two different graph density values. Three user studies with three different graph layout algorithms (FD, Circular, and MDS) were conducted. Note that the formula of graph density used here is nonlinear with respect to the graph order. Changing the number of nodes could result in a different perceptual model [53]. This experiment, only studies graphs with 100 nodes, and future work should explore how perceptual discrimination responds as the graph size changes.

4.1 Graph Generation

To study the perception of graph density, graphs with varying density need to be generated and visualized. Considering the size of the display and potential cluttering issues of node-link diagrams, the graph order was chosen to be $|V| = 100$ in all experiments. Therefore, a simple connected graph has its graph density range from 0.02 to 1. Graphs are generated for every value in this range at intervals of 0.01, resulting in 99 different graph density values.

A simple, connected graph G with graph density D was generated stochastically by an iterative procedure. The initial graph has 100 vertices and 99 edges connecting the vertices in a path. At each step, a vertex is randomly selected and connected to another vertex that is not already its neighbor. This increases the edge count and thus the graph density. This process is repeated until the graph density comes within the range $D \pm T$. Here T is a tolerance parameter to define the accuracy of the output graph density and the value we use is 0.001. Algorithm 1 thoroughly describes this procedure.

Due to its stochastic nature, Algorithm 1 generates different graph structures for same graph density D. Different graph structures with the same graph density may vary on other graph properties, which cannot always be directly controlled for in our study. To mitigate the impact of perception on other graph properties associated with one particular graph structure, 50 graph structures for each D are generated and will be used randomly in the experiment.

Once graph structures are generated, the layout algorithm is applied to create graph drawings. The following three layout algorithms in different categories are used in our study based on their popularity and ease of use: FD [41], Circular Layout [24],

Algorithm 1: Generate graph with given graph density

```
Input   : Number of Vertices($N$), Required graph density($D$), Tolerance($T$)
Output  : connected simple graph $G$ with graph density $D$
Create a path graph $G$ with $N$ vertices $v_1, v_2, \dots, v_n$ having edges $\{v_i, v_{i+1}\}$
  where $i = 1, 2, \dots, n-1$. Let set of vertices be $V = \{v_1, v_2, \dots, v_n\}$;
Calculate the graph density of $G$ as $D_{new}$ ;
while $|D - D_{new}| > T$ do
    Randomly select a vertex $v_i$ with uniform probability ;
    Find its current neighbors $N_i$ ;
    Randomly select a vertex $v_k$ with uniform probability from the set
      $V \setminus \{N_i \cup v_i\}$ ;
    Add an edge $\{v_i, v_k\}$ to graph $G$ ;
    Recalculate graph density $d_{new}$ ;
end
```

and MDS [7]. These layout algorithms also have random factors when they position the vertices. To mitigate this impact, we randomly create 20 graph drawings using each layout algorithm for every graph structure. Therefore, for graph density, we have 99,000 (99($values$) × 50($structures$) × 20($layouts$)) graph drawings by each layout algorithm. This is the pre-generation of our graph drawing pool used in our experiment, and Figure 12 lists some example graph drawings by these three layout algorithms for different graph density values.

It is recognized that the order of the graph (the number of vertices) may also relate to a participant's ability to perceive JND. However, as the order of a graph increases, many algorithms converge towards a hairball layout. The goal of this experiment was to use a constant (modest) graph order to evaluate the perception of graph properties with respect to layout. Future studies will explore the range of graph orders in which graph properties are perceptible.

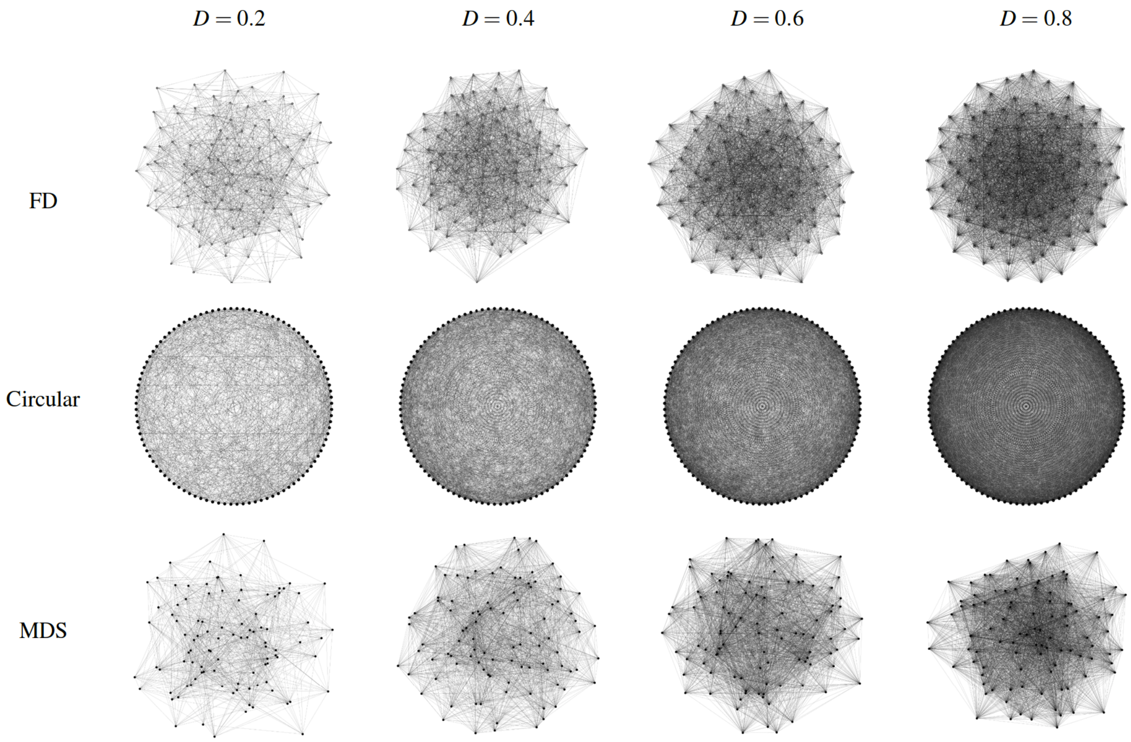


Figure 12: Examples of the graph drawings by the three layout algorithms at several graph density values

4.2 Procedure

The experimental procedure is a mixed 7×2 design in which there are 7 base values (0.2, 0.3, 0.4, 0.5, 0.6, 0.7, and 0.8) and two approach conditions (*above* and *below*). For each base value, D, the JND will be estimated from *above* and *below.*

Graph density may not be a widely known term and it is possible that our participants on Amazon Mechanical Turk (AMT) may not know about this concept. As such, the experiment presents an introduction page describing the meaning of graph density and presents example graph drawings with low and high density values. To verify that participants understand the concept of graph density, they have to pass a screening session with 20 judgments. These judgments are designed to be easy (difference value > 0.2 between the two stimuli) so as to not exclude participants based on their perceptual ability. The goal of the screening is only to ensure that they understand the concept and know what the task is. For the first 5 judgments, participants will receive feedback on their choice. If they choose incorrectly, they have to explicitly make the right choice before they can move to the next judgment. For the final 15 judgments of the screening, there is no feedback, and the participant has to make at least 10 correct choices to continue to the real session.

For each layout algorithm, the conditions in this experiment include the seven base values and two approaches (*above* and *below*). Each participant is randomly assigned two base values and both *above* and *below* trials will be conducted for each base value. This results in four trials per participant and each trial consists of at most 50 judgments. After each judgment, the screen will flash gray to notify participants that a new set of images to be judged have been rendered. In practice, the experiment takes approximately 10 minutes to complete. Following the completion of all four trials, a demographics questionnaire was given to participants. Finally, a short debriefing is provided. Payout rates were $.50 per participant.

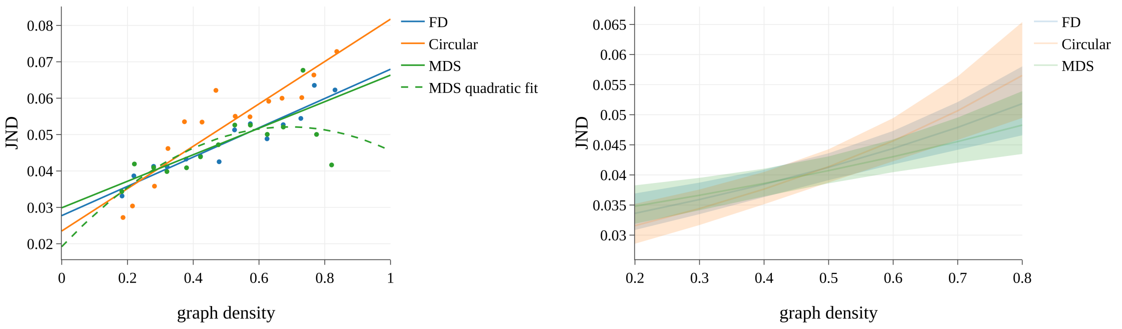
4.3 Results

105 participants were recruited for the user study of Circular layout, another 105 participants were recruited for the user study of the MDS layout, and finally, 102 participants were recruited for the user study of the FD layout. For each user study, this yields 30 data samples for every base value and approach condition and 420 (408 for FD) data samples in total. Prior to analysis, we removed outliers that are outside

Table 1: Parameters in the Model of Weber's Law

Property	Layout	β_0	β_1	R^2	r	RMS
GD	FD	.0277	.0402	.904	.95	.0026
	Circular	.0235	.0582	.832	.91	.0052
	MDS	.0337	.0261	.438	.66	.0059
ALCC	FD	.5763	-.6478	.823	-.90	.033
	Circular	–	–	–	–	–
	MDS	.3619	-.3796	.911	-.95	.013

of 3 median absolute deviations from the median in each base value and approach condition. 37, 12 and 21 samples were removed for FD, MDS, and Circular respectively ($< 10\%$). No group level ceiling effects were observed.



(a) (b)

Figure 13: Regression results for graph density user studies. (a) The model fit for the averaged individual JNDs. (b) The model fit for individual points after the Box-Cox transformation where the colored area indicates the 95% confidence interval.

Among all the participants that shared their gender and age information post study, the age and gender distribution in the Circular, MDS and FD layout experiments were 60 females and 42 males(age varying from 18-68), 64 females and 39 males (age varying from 18-76), and 62 females and 34 males (age varying from 19-71) respectively.

4.3.1 The Model of Weber's Law

Following classical work on perceptual laws [62, 67], the individual JNDs were averaged over (value, approach) groups and calculate the adjusted graph density value off the base value before fitting a regression model. The adjusted density value, D_A, of each base density value, D, is calculated by shifting towards the approach direction by half of the average JND of the group.

$$D_A = D + 0.5 \times a_i \times JND \tag{4.2}$$

$$a_i = \begin{cases} 1 & \text{if approach is from } \textit{above} \\ -1 & \text{if approach is from } \textit{below} \end{cases} \tag{4.3}$$

The average JNDs are further fit by the adjusted graph density values through a linear regression.

$$JND = \beta_0 + \beta_1 D_A + \epsilon \tag{4.4}$$

The same modeling process is applied to the data collected in the three user studies about graph density with different layout algorithms. The model coefficients (β_0 and β_1), R^2, the root-mean-square (RMS) error, and the correlation (r) between D_A and JNDs are listed in Table 1. Figure 13a shows the fit lines for the three layout algorithms along with the observed average JNDs. Among the three layout algorithms (FD, Circular, and MDS), both FD and Circular layout have a high goodness-of-fit ($R^2 \approx 90\%$ and $R^2 \approx 83\%$ respectively) and appear to follow Weber's law. For these layouts, participants were able to better discriminate between graphs when the density is lower and such ability decreases linearly when graph density increases.

While the perception of density in the FD and Circular layout follow Weber's law, it is found that the linear model for the MDS layout only explains 44% of the variance. It can be seen in Figure 13a (green) that when the graph density becomes large, the model fails to fit the underlying data. However, applying a quadratic regression to the MDS data results in a better goodness-of-fit ($R^2 = 0.53$), as shown in Figure 13a (green dashed line). Furthermore, calculating Cook's D [17] for the MDS data, one can find one leverage point of $D = 0.8$ and approach *above* (which has the largest Cook's D $= 1.14$, ~ 8 times the mean Cook's D of observations). After removing this observation, the goodness-of-fit of a linear model increased to $R^2 = 0.70$. This indicates that the MDS layout may follow Weber's law within a smaller range (specifically, [0.2, 0.7]) of graph density values.

4.3.2 Fitting Individuals

While models were found to fit the data, averaging individual JNDs could result in a loss of individual variance [46, 15]. As such, this thesis also analyzed the data following the approach of Kay and Heer [46], who re-analyzed Harrison et al.'s data to include individual variance. Taking the raw base value of graph density and the individual JNDs, a linear regression model was fit that includes the approach (*above*/*below*) as a categorical variable, a_i, which is defined in Equation 4.3. This model uses the raw base value of graph density, approach, and the interaction of these two variables and is defined as:

$$JND_i = \beta_0 + \beta_1 \times D_i + \beta_2 \times a_i + \beta_3 \times a_i \times D_i + \epsilon_i \tag{4.5}$$

To test the model's adequacy, we examined the residual distributions (Figure 14 (left)). By observation, it is found that the residuals are skewed compared to a normal distribution, and a Box-Cox transformation for each dataset was applied using

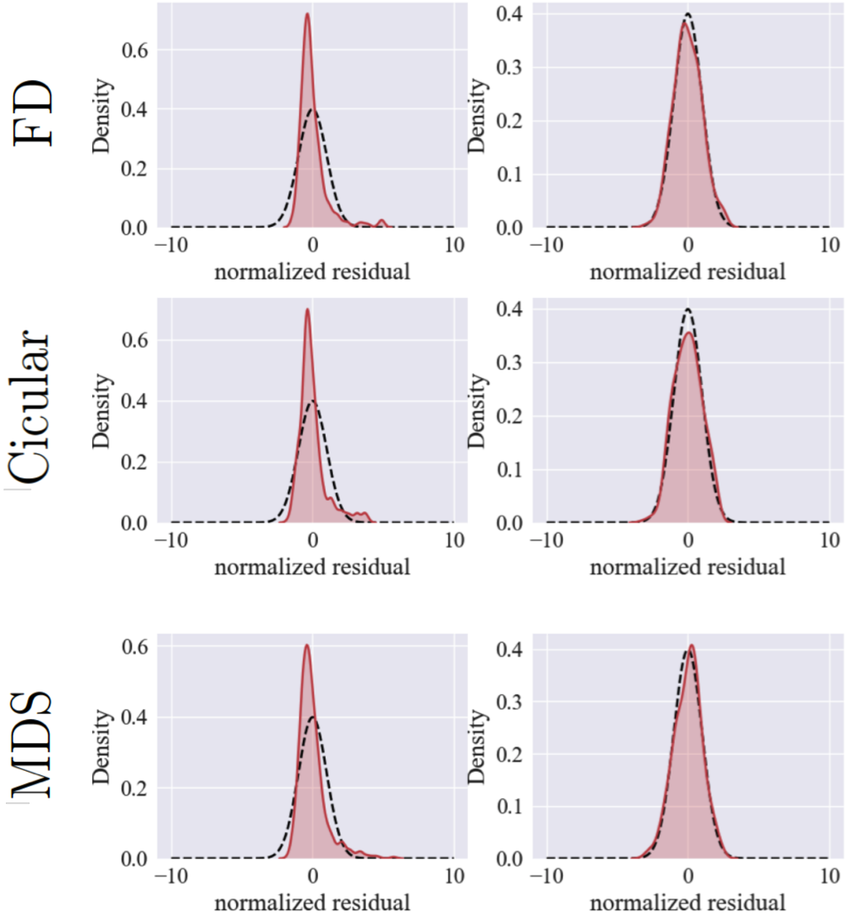


Figure 14: The left shows that residual distributions were skewed before the transformation and the right shows the distribution is more normal after the transformation

Equation 4.6. We then fit the model to the transformed data, Figure 13b.

$$JND_i^{(\lambda)} = \beta_0 + \beta_1 \times D_i + \beta_2 \times a_i + \beta_3 \times a_i \times D_i + \epsilon_i \tag{4.6}$$

$\lambda = -0.5$ is used in the final model and this value is in the 95% confidence interval of the estimated λ for all three layout algorithms. Figure 14 (right) shows that residual distributions after this transformation become more normal. This indicates that the perception of graph density with the drawings given by the FD, Circular, and MDS layout algorithms may not follow an exactly linear relationship to the property value when individual variances are considered; instead, a power transformation may be required.

4.3.3 Comparison Between Layout Algorithms

In the experiment, workers are randomly recruited on AMT for each user study. In this way, subjects are considered as independent between conditions (layout algorithms) and the individual JNDs are independent measures. To compare the effect of the three layout algorithms (FD, Circular, and MDS) with respect to their ability to discriminate on graph density, the individual JNDs (as opposed to the mean JNDs) are used and following the work of Harrison et al. [36], a pairwise Mann-Whitney U test was applied.

A Bonferroni correction was applied for three pairs and set $\alpha = 0.0166$ in each test between two layouts. Results indicate that there are no significant differences in their individual JND distributions. Similarly, when the test is separately applied to the averaged JND, there was no significant difference found.

To further compare the three layout algorithms with respect to the discrimination of graph density, the linear fit for the model of Weber's law (see Figure 13a) was examined. The clear overlap between all the layout algorithms for $JND < 0.55$ confirm the previous statistical findings that the three layout algorithm are not significantly different when perceiving graph density. Finally, the best model fit for all the individual points is shown in Figure 13b where the colored area shows the 95% confidence interval of the model. This plot shows even more overlap across the entire range of D values confirming that all the three layouts are roughly equivalent for discriminating graph density.

Chapter 5

EXPERIMENT 2: AVERAGE LOCAL CLUSTERING COEFFICIENT

While graph density is a relatively straight forward property to visually explain, a primary aim for the second experiment was to begin exploring perception in graph layouts with respect to more complex graph properties. In real-world networks, vertices tend to create tightly connected groups and form clusters. This generates more clustering than random graphs. Along with scale-free property where degree distribution follows a power law, a high clustering coefficient is one critical characteristic of complex networks and plays an important role in graph analysis. As such, our second experiment focused on the perception of clustering in a simple, connected, undirected graph. This experiment studied the perception of a global clustering measure, the average local clustering coefficient (ALCC), which is defined as:

$$C = \frac{1}{|V|} \sum_{i=1}^{|V|} \frac{2\left|\{e_{jk} : v_j, v_k \in V_i, e_{jk} \in E\}\right|}{k_i(k_i - 1)}. \tag{5.1}$$

This is the average of the local clustering coefficients of all the vertices measured by Watts and Strogatz [72] for ‘small-world’ analysis. In this equation, V is the vertex set and E is the edge set. V_i represents the immediately connected neighbors of a vertex v_i, and k_i is the degree of the vertex v_i.

This experiment quantifies how well people perceive clustering given graph drawings with different average local clustering coefficients and different layout algorithms. From an initial inspection among the three layout algorithms (Table 15), the circular layout with random vertex position provides little obvious discrimination between two graph drawings with large differences in ALCC. Therefore, the Circular layout is removed

Algorithm 2: Generate Graph with tunable ALCC

```
Input  : Number of Vertices ($N$), Number of edges for each vertex ($m$),
         Probability of Triad Formation ($P_t$)
Output : Connected simple scale-free graph with ALCC = $C$
Initialize graph $G$ with small number of vertices, $m_0$, and no edges. Let $V$ be
  the set of vertices of $G$, and $k_v$ be the degree of vertex $v$ ;
while the number of vertices in $G < N$ do
    Add a vertex $v$ with $m$ edges to the graph $G$ ;
    Select another vertex $w$ of $G$ with probability $P_w$ which is proportional to its
      degree;
                    $P_w = \frac{k_w}{\sum_{v \in V} k_v}$          (5.2)
    Add an edge connecting $v$ and $w$ ;
    while unattached edges remain in $v$ do
        Perform the following TF step with probability $P_t$ or PA step with
          probability $1 - P_t$;
        TF: add an edge between $v$ and a neighbor of $w$;
        PA: select a new vertex $w'$ other than $v$ and $w$ with probability, $P_{w'}$
          given by Equation 5.2 and add an edge between $v$ and $w'$. Update $w$
          with $w'$. ;
    end
end
```

from this experiment. Two user studies are conducted with the FD and MDS layout algorithms respectively for average local clustering coefficient.

5.1 Graph Generation

For this experiment, graphs of $|V| = 100$ with varying ALCC are generated while keeping the scale-free property. Specifically, the number of edges is kept the same across all graphs generated in order to keep graph density constant. However, by keeping the above properties, only a limited range of ALCC value can be obtained

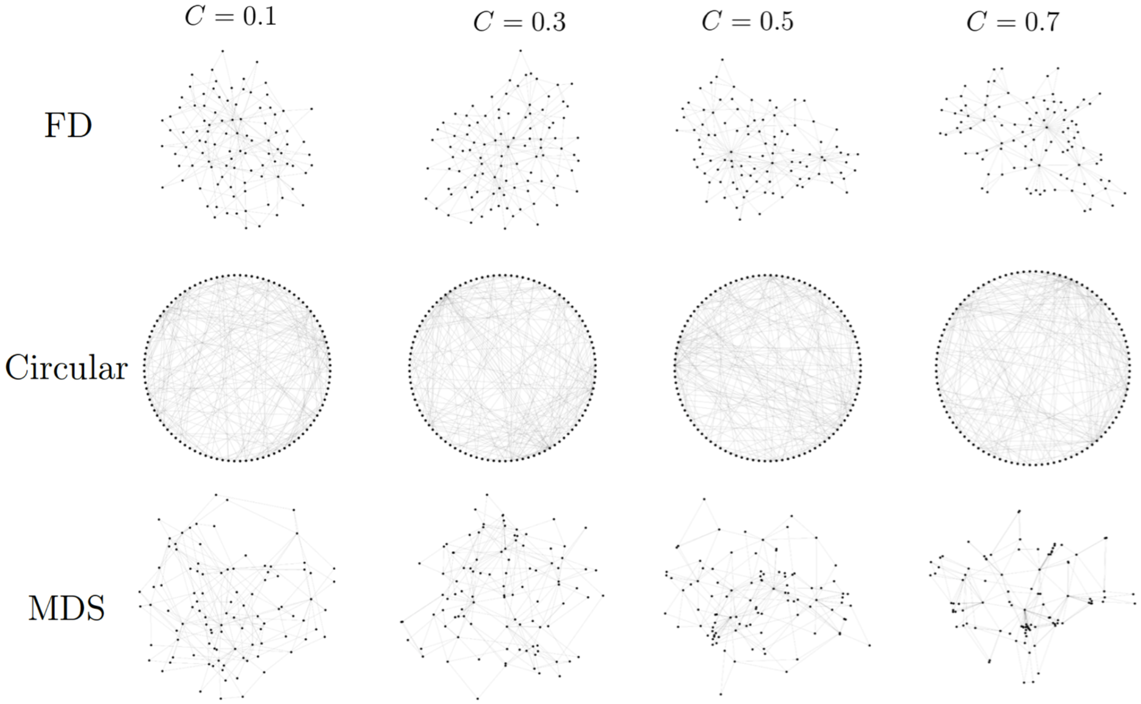


Figure 15: Examples of the graph drawings from the three layout algorithms at several average local clustering coefficient values

as shown by Holme et al. [40]. With $|V| = 100$ and $|E| = 194$, the ALCC value, C ranges from 0.07 to 0.75. Note that the range of ALCC for any graph is 0 to 1.

The graph generation procedure of Holme et al. [40] is used to produces scale-free graphs with tunable clustering. It is an extension of Barábasi and Albert model (BA model) [2]. Algorithm 2 describes the procedure of our graph generation for varying C. In this algorithm, the Preferential Attachment (PA) step comes from the BA model and the Triad Formation (TF) step is an extension by Holme et al. [40]. Note that the PA step ensures the scale-free property of graph G, while the TF step increases C. Thus, by varying the probability P_t of taking a TF step, different values of C can be achieved.

We used Algorithm 2 with $m = 2$, and varied P_t to get graphs with different C.

Similar to the graph generation for graph density, graphs are generated given every ALCC value, C, in the range of [0.07, 0.75] with step equals to 0.01. This results in 69 different average local clustering coefficient values. For each C, 50 graph structures are generated and each graph structure is visualized in 20 different graph drawings using the corresponding layout algorithm. Example graph drawings for varying C are shown in Figure 15.

5.2 Procedure

Similar to Experiment 1, we present participants on AMT an introduction page to describe the meaning of clustering followed by two examples of graph drawings with high clustering and low clustering. Following the introduction, participants are required to take a screening session with 20 judgments which are designed to be highly discriminable. In each judgment, one graph has a low clustering value in [0.1, 0.3], and the other graph has a high clustering value in [0.5, 0.7]. For the first 5 judgments, the participants will receive feedback, and if the participant chooses incorrectly they have to explicitly choose the correct one to move on. For the final 15 judgments, there is no feedback and the participant has to make at least 10 correct choices to continue.

The experimental procedure for the average local clustering coefficient and MDS layout is identical to the procedure for graph density but with a different base value and step-size. This procedure has a 5×2 design in which there are 5 base values (0.2, 0.3, 0.4, 0.5, and 0.6) and two approach conditions (*above* and *below*) for each base value. The initial-difference is 0.1 and the step-size is 0.01, and the maximum number of judgments is set to be 50. For each average local clustering coefficient base value, C, the JND is estimated from *above* and *below* using the methodology presented in section 3.1. Each participant was randomly assigned two base values with both the *above* and the *below* approach.

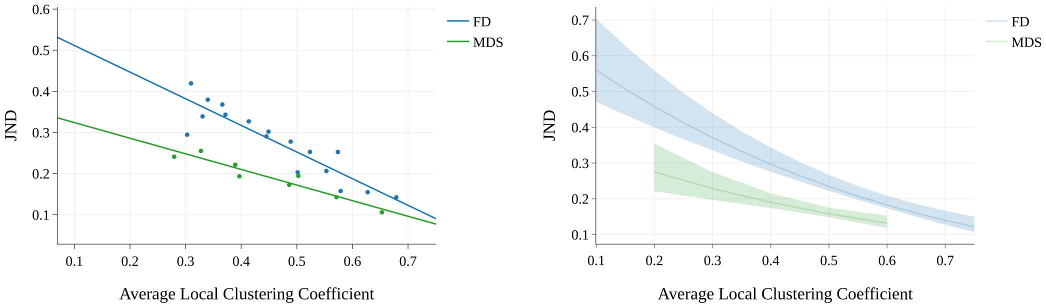
The same procedure was applied for average local clustering coefficient and the FD layout. However, when collecting preliminary data for the FD layout, we found that participants suffered from a severe ceiling effect, which means the JND is bounded by the possible values we can generate. For example, we have $C = 0.07$ as the minimum average local clustering coefficient, and when we run (0.2, *below*), the furthest distance we can achieve is 0.13. This range is not discriminable by participants and prevents us from quantifying the the true JND for small values of C. Furthermore, during our preliminary data collection for the FD layout, we found that many of the participants completed all 50 judgments, which means that the experiment may have ended prior to the participant reaching a stabilized discrimination. Based on this information, the base values and initial-difference was modified for the full study. For the final FD experiment, a 10×2 design was used in which the base values for *above* and *below* are different. For the *above* approach, lower base values (0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, and 0.55) are used, and for the *below* approach, higher base values (0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75) are used. To help participants reach their JND faster, the initial-difference is enlarged to 0.2. This value was chosen because the average JNDs of each base value that was estimated in the Preliminary FD study either reached the ceiling or was larger than 0.2. Finally, the maximum number of judgments was increased to 75 to help participants reach a stable discrimination state.

5.3 Results

For the MDS layout, 75 participants were recruited. Each participant was assigned two base values with both *above* and *below* approaches. This gives 30 data samples for every (value, approach) pair and 300 data samples in total. For the FD layout experiment, 201 participants were recruited. Each participant was assigned two base values, one from each approach. This gives 20 data samples for every (value, approach) pair and 402 samples in total. Among all the participants that shared their gender and age information post study, the MDS layout study had 43 females and 32 males (age varying from 19-69) while the FD study had 110 females and 88 males (age varying from 18-70) respectively.

Before the JNDs for the ALCC were modeled, the distribution of JNDs for each (value, approach) condition are analyzed to identify outliers as well as groups that suffer from the ceiling effect (hit rate $>$ 50%). In the collected JNDs for the MDS layouts, two groups, ($C = 20$, approach $=$ *below*) and ($C = 30$, approach $=$ *below*), had a hit rate greater than 50% (80.42% and 62.08% respectively). These samples were removed from analysis as their true JND is not accurately measured. No outliers (points that fall outside 3 median deviations for each (value, approach) pair) were found.

Following the same procedure for the FD layouts, the observations for $C \in \{30, 35, 40\}$ with approach $=$ *below* were removed. After removing groups due to the ceiling effect, outliers that fall outside 3 median deviation for each (value, approach) pair were also removed. In total, 60 samples were removed because of ceiling effects and 1 sample was removed as an outlier. 341 samples of the 402 samples collected are used in the analysis.



(a)

(b)

Figure 16: Regression results for the average local clustering coefficient user studies. (a) The model fit for the averaged individual JNDs (b) The model fit for individual points after the Box-Cox transformation where the colored area indicates the 95% confidence interval.

5.3.1 The Model of Weber's Law

After the observations were removed under the conditions mentioned, the JNDs in each group (base value × approach) were averaged and adjusted following the same analysis procedure as section 4.3.1. The linear regression model with the adjusted base value C_A was fit for the averaged JNDs for both the FD layout and the MDS layout. The fit coefficients (β_0, β_1), R^2 and RMS error, and the correlation between C_A and JNDs are listed in Table 1. Results indicate that MDS has a better goodness-of-fit ($R^2 \approx 91.1\%$) than FD layout ($R^2 \approx 82.3\%$), while the average perceptions of ALCC displayed by both layout algorithms follow Weber's law with a negative linear relationship of the property value as illustrated in Figure 16a.

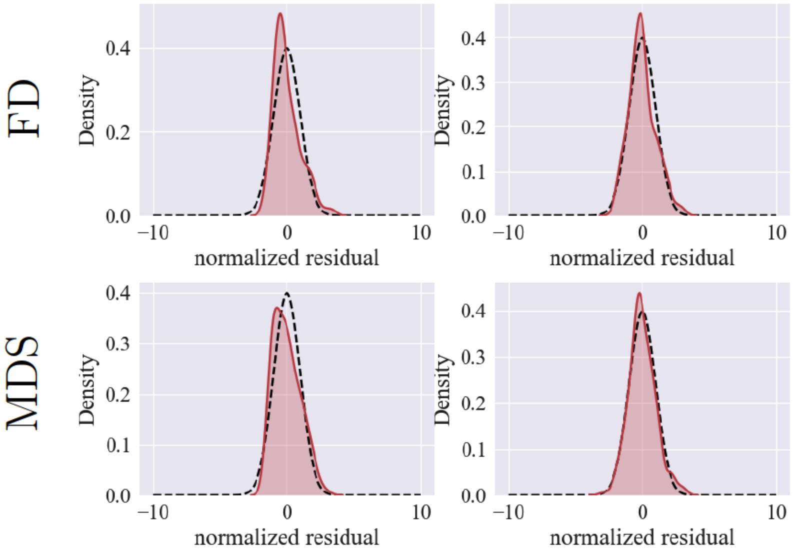


Figure 17: The left shows that residual distributions were skewed before the transformation and the right shows the distribution is more Normal after the transformation.

5.3.2 Fitting Individuals

As in Section 4.3.2, all individual points were modeled without taking the average of the JNDs in each group. First, a linear model with base value C, approach, and their interaction term was fit for individual JNDs. Then, the residuals were analyzed against normal distribution. To correct the skewness of the residual distribution, a Box-Cox transformation was applied ($\lambda = 0.2518$ with Confidence Interval $(0.08, 0.42)$ for FD, and $\lambda = -0.058$ with Confidence Interval $(-0.279, 0.165)$ for MDS). This indicates that a log transformation for MDS and a power transformation with $\lambda \approx 0.24$ could fit the data better than a linear model when individual variance is considered. Figure 17 shows that residual distributions after this transformation become more normal.

5.3.3 Comparison Between Layout Algorithms

Similar to section 4.3.3, a Mann-Whitney U test with $\alpha = 0.05$ was conducted to compare FD and MDS for presenting average local clustering coefficient. With p-value < 0.05 for both the test on individual JNDs and the test on the averaged JNDs, it was found that there is a significant difference between the JNDs observed for the FD and MDS layout algorithm.

To further compare the two layout algorithms with respect to the discrimination of the average local clustering coefficient, their best-fit linear regression models were examined, Figure 16a. Here it can be seen that the MDS layout algorithm provides a better perceptual discrimination of the average local clustering coefficient across the entire range of tested values.

Finally, the best model fit for all the individual points is shown in Figure 16b with the colored area representing the 95% confidence interval of the model. This plot indicates that the MDS layout algorithm performs better then FD for perceiving the average local clustering coefficient.

Chapter 6

CONCLUSION

To my knowledge, this is the first experiment designed to model humans' ability to perceptually discriminate graph properties. Such experiments provide us a means of quantitatively comparing graph layout algorithms with respect to their ability to communicate graph properties. The models and results presented in this work demonstrate that for the two graph properties tested, different layout algorithms can be modeled using Weber's law. This experiment analyzed discriminations from 588 participants. Results in the perception of graph density showed that the three layout algorithms explored (Force Directed - FD, Circular, Multi-Dimensional Scaling - MDS) could be modeled using Weber's law and there was no significant difference between the layout algorithms. Results in the perception of average local clustering coefficient demonstrated that the two algorithms considered (MDS and FD) can be modeled using Weber's law. This time there is a significant difference between the algorithms, as the MDS algorithm is better at discriminating ALCC than the FD algorithm.

The study used Amazon Mechanical Turk to host the experiments. Even with limited control on the user's end, AMT works well for such studies [6]. According to Borgo et al. [6], such crowdsourced studies provide several advantages like larger and diverse samples, easier and faster data collection, and financial effectiveness. They also discuss 4 case studies of successful crowdsourcing-based evaluations among which two compared static visualizations while one used an interactive visualization. The remaining case study was of a crowdsourced-based evaluation that accounted for the effect of user's individual traits on the responses that were collected.

Table 2: Common graph properties

Property	Definition
Triangle Count	Count of complete subgraphs with 3 nodes
Vertex Connectivity	Minimum number of vertices whose removal disconnects the network
Edge Connectivity	Minimum number of edges whose removal disconnects the network
Global Clustering Coefficient	It is given by $3 * n_\Delta / n_\Lambda$ where n_Λ is the number of connected triplets (connected subgraph with 3 vertices and 2 edges) and n_Δ is the number of triangles.
Degree Distribution	Probability distribution of node degrees over the whole graph.
Average Path Length	Average of shortest distance between all pairs of vertices in a graph.
Assortativity Coefficient	Tendency of the vertices in a graph to be connected to other vertices with similar values of some vertex property (e.g., degree distribution).
Network Diameter	Greatest distance between any pair of vertices.
Network Radius	It is defined as the minimum eccentricity over all vertices in the network where eccentricity of a vertex is the maximum distance between the vertex and any other vertex in the network.
Scale-free	A network is scale-free if its degree distribution follows a power law
Small-world	A network is small-world if it has much larger clustering coefficient and almost equal average path length when compared to a random network with same average degree
Efficiency	Measures how well the network is connected such that information is exchanged efficiently between nodes. Networks with short geodesic distance between most pairs of its vertices have high efficiency.

This study explores only two graph properties and instances of graph layouts for three categories of algorithms. However, a large variety of graph properties remain to be studied; see Table 2 for a summary. Future work should explore more such properties and correlations between them. This is especially important for the perception of graph properties like degree assortativity as it does not directly affect the objective function of the layout algorithms unlike the clustering coefficient(a high clustering coefficient introduces cliques in the graph structure that become clumps of vertices in the graph drawing). Moreover, additional layout algorithms need to be tested for perception, especially those belonging to the class of orthogonal layouts which was left in this thesis. An additional limitation is that all graphs tested in this experiment were of fixed order (number of vertices). As graphs become larger, some layout algorithms tend to produce hairball layouts. Future experiments should explore the effects of graph type, graph order, and screen size on perception to determine at what settings the discrimination of graph properties becomes infeasible, and how (or if) perception is affected. Additionally, several multi-level graph drawing methods exists that efficiently produce drawings for large graphs unlike the basic force-directed model. However, the effect of the approximations used in these methods on the perception of properties should be determined.

Furthermore, a comparison of perception within classes of layout algorithms should be done to explore if all layout algorithms that fall within a class (e.g., force directed algorithms) have the same basic underlying perceptual properties. This is especially important for the force-directed algorithms as some of its variants have been specifically designed to showcase clustering in a network. Finally, the correlations between all the graph properties and their effect on graph perception should be investigated. For instance, there is a clear correlation between graph density and average degree, but does this signify that the layouts algorithms that are best to perceive graph density would also be best for average degree. This experiment only scratches the surface of potential combinations of layouts and properties. However, this serves as an initial step in demonstrating that (at least for some) graph properties can be discriminated.

Finally, by identifying the different conditions and classes of algorithms that improve discrimination, future work can inform ideas of new design spaces for graph layout algorithms that not only focus on layouts for graph aesthetics, but also on conditions for graph perception. For instance, are there ways in which the optimization model of the Force-directed or the MDS layout algorithm be informed to manipulate JND properties. Moreover, an entirely new graph drawing algorithms could be developed, based on further findings in the perception of graph properties, that provides a good perception of several(or all) graph properties.

REFERENCES

[1] Shweta Bansal, Shashank Khandelwal, and Lauren Ancel Meyers. "Exploring biological network structure with clustered random networks". In: *BMC bioinformatics* 10.1 (2009), p. 405.

[2] Albert-László Barabási and Réka Albert. "Emergence of scaling in random networks". In: *Science* 286.5439 (1999), pp. 509–512.

[3] Josh Barnes and Piet Hut. "A hierarchical O (N log N) force-calculation algorithm". In: *nature* 324.6096 (1986), p. 446.

[4] Mathieu Bastian, Sebastien Heymann, Mathieu Jacomy, et al. "Gephi: an open source software for exploring and manipulating networks." In: *International AAAI Conference on Weblogs and Social Media* 8 (2009), pp. 361–362.

[5] Vincent D Blondel et al. "Fast unfolding of communities in large networks". In: *Journal of statistical mechanics: theory and experiment* 2008.10 (2008), P10008.

[6] Rita Borgo et al. "Crowdsourcing for Information Visualization: Promises and Pitfalls". In: *Evaluation in the Crowd. Crowdsourcing and Human-Centered Experiments*. Springer, 2017, pp. 96–138.

[7] Ulrik Brandes and Christian Pich. "An experimental study on distance-based graph drawing". In: *Graph Drawing*. Springer. 2009, pp. 218–229.

[8] Ulrik Brandes and Christian Pich. "Eigensolver methods for progressive multidimensional scaling of large data". In: *International Symposium on Graph Drawing*. Springer. 2006, pp. 42–53.

[9] Ulrik Brandes and Dorothea Wagner. "Analysis and visualization of social networks". In: *Graph Drawing Software* (2004), pp. 321–340.

[10] Jérôme Callut et al. "Semi-supervised classification from discriminative random walks". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2008, pp. 162–177.

[11] Sara Camacho et al. "Just noticeable differences and weber fraction of oral thickness perception of model beverages". In: *Journal of Food Science* 80.7 (2015).

[12] Deepayan Chakrabarti and Christos Faloutsos. "Graph mining: Laws, generators, and algorithms". In: *ACM Computing Surveys* 38.1 (2006), p. 2.

[13] Lisha Chen and Andreas Buja. "Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis". In: *Journal of the American Statistical Association* 104.485 (2009), pp. 209–219.

[14] Subeh Chowdhury, Avishai Avi Ceder, and Bradley Schwalger. "The effects of travel time and cost savings on commuters' decision to travel on public transport routes involving transfers". In: *Journal of Transport Geography* 43 (2015), pp. 151–159.

[15] William S Cleveland, Charles S Harris, and Robert McGill. "Judgments of circle sizes on statistical maps". In: *Journal of the American Statistical Association* 77.379 (1982), pp. 541–547.

[16] William S Cleveland, Robert McGill, et al. "Graphical perception and graphical methods for analyzing scientific data". In: *Science* 229.4716 (1985), pp. 828–833.

[17] R Dennis Cook. "Detection of influential observation in linear regression". In: *Technometrics* 19.1 (1977), pp. 15–18.

[18] Katri K Cornelissen et al. "Visual biases in judging body weight". In: *British Journal of Health Psychology* 21.3 (2016), pp. 555–569.

[19] Jan De Leeuw. "Applications of convex analysis to multidimensional scaling". In: (2011).

[20] D Varuna SX De Silva et al. "Just noticeable difference in depth model for stereoscopic 3D displays". In: *IEEE International Conference on Multimedia and Expo*. IEEE. 2010, pp. 1219–1224.

[21] Vin De Silva and Joshua B Tenenbaum. *Sparse multidimensional scaling using landmark points*. Tech. rep. Technical report, Stanford University, 2004.

[22] Giuseppe Di Battista et al. "An experimental comparison of four graph drawing algorithms". In: *Computational Geometry* 7.5-6 (1997), pp. 303–325.

[23] Giuseppe Di Battista et al. *Graph Drawing: Algorithms for the Visualization of Graphs*. Englewood Cliffs, NJ: Prentice Hall, 1999.

[24] Uğur Doğrusöz, Brendan Madden, and Patrick Madden. "Circular layout in the graph layout toolkit". In: *International Symposium on Graph Drawing*. Springer. 1996, pp. 92–100.

[25] Markus Eiglsperger, Sándor P Fekete, and Gunnar W Klau. "Orthogonal graph drawing". In: *Drawing Graphs*. Springer, 2001, pp. 121–171.

[26] John Ellson et al. “Graphviz - Open Source Graph Drawing Tools”. In: *International Symposium on Graph Drawing.* 2001, pp. 483–484.

[27] Paul Erds and Alfréd Rényi. “On the evolution of random graphs”. In: *Publ. Math. Inst. Hung. Acad. Sci* 5 (1960), pp. 17–61.

[28] Manuel Freire et al. “ManyNets: an interface for multiple network analysis and visualization”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* ACM. 2010, pp. 213–222.

[29] Thomas MJ Fruchterman and Edward M Reingold. “Graph drawing by force-directed placement”. In: *Software: Practice and experience* 21.11 (1991), pp. 1129–1164.

[30] Severino F Galán and Ole J Mengshoel. “Neighborhood beautification: Graph layout through message passing”. In: *Journal of Visual Languages & Computing* 44 (2018), pp. 72–88.

[31] Helen Gibson, Joe Faith, and Paul Vickers. “A Survey of Two-Dimensional Graph Layout Techniques for Information Visualisation”. In: 12 (July 2012), pp. 324–357.

[32] Louis D Goodfellow. “An empirical comparison of audition, vision, and touch in the discrimination of short intervals of time”. In: *The American Journal of Psychology* 46.2 (1934), pp. 243–258.

[33] Stefan Hachul and Michael Jünger. “An experimental comparison of fast algorithms for drawing general large graphs”. In: *International Symposium on Graph Drawing.* Springer. 2005, pp. 235–250.

[34] Stefan Hachul and Michael Jünger. “Large-Graph Layout Algorithms at Work: An Experimental Study.” In: *J. Graph Algorithms Appl.* 11.2 (2007), pp. 345–369.

[35] David Harel and Yehuda Koren. “Graph drawing by high-dimensional embedding”. In: *International symposium on graph drawing.* Springer. 2002, pp. 207–219.

[36] Lane Harrison et al. “Ranking visualizations of correlation using weber’s law”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 1943–1952.

[37] Jeffrey Heer and danah boyd. "Vizster: Visualizing Online Social Networks". In: *IEEE Information Visualization (InfoVis)*. 2005, pp. 32–39. URL: http://vis.stanford.edu/papers/vizster.

[38] Michael Himsolt. "Comparing and evaluating layout algorithms within GraphEd". In: *Journal of Visual Languages & Computing* 6.3 (1995), pp. 255–273.

[39] Michael Himsolt. "GraphEd: A graphical platform for the implementation of graph algorithms (extended abstract and demo)". In: *International Symposium on Graph Drawing*. Springer. 1994, pp. 182–193.

[40] Petter Holme and Beom Jun Kim. "Growing scale-free networks with tunable clustering". In: *Physical Review E* 65.2 (2002), p. 026107.

[41] Yifan Hu. "Efficient, high-quality force-directed graph drawing". In: *Mathematica Journal* 10.1 (2005), pp. 37–71.

[42] Weidong Huang, Peter Eades, and Seok-Hee Hong. "Larger crossing angles make graphs easier to read". In: *Journal of Visual Languages & Computing* 25.4 (2014), pp. 452–465.

[43] Sanjay Kairam et al. "GraphPrism: compact visualization of network structure". In: *Proceedings of the International Working Conference on Advanced Visual Interfaces*. ACM. 2012, pp. 498–505.

[44] Tomihisa Kamada, Satoru Kawai, et al. "An algorithm for drawing general undirected graphs". In: *Information processing letters* 31.1 (1989), pp. 7–15.

[45] U Kang, Charalampos E Tsourakakis, and Christos Faloutsos. "Pegasus: A peta-scale graph mining system implementation and observations". In: *IEEE International Conference on Data Mining*. IEEE. 2009, pp. 229–238.

[46] Matthew Kay and Jeffrey Heer. "Beyond weber's law: A second look at ranking visualizations of correlation". In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2016), pp. 469–478.

[47] Steve Kieffer et al. "Hola: Human-like orthogonal network layout". In: *IEEE transactions on visualization and computer graphics* 22.1 (2016), pp. 349–358.

[48] S. G. Kobourov. "Force-Directed Drawing Algorithms". In: *Handbook of Graph Drawing and Visualization*. Ed. by R. Tamassia. CRC Press, 2013, pp. 383–408.

[49] Stephen Kobourov, Sergey Pupyrev, and Bahador Saket. "Are Crossings Important for Drawing Large Graphs". In: *International Symposium on Graph Drawing.* 2014, pp. 234–245.

[50] Robert B Lawson, Jean E Graham, and Kristin M Baker. *A history of psychology: Globalization, ideas, and applications.* Routledge, 2015.

[51] Jiafan Li, Yuhua Liu, and Changbo Wang. "Evaluation of graph layout methods based on visual perception". In: *Proceedings of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing.* ACM. 2016, p. 90.

[52] Menglin Li and Colm O'Riordan. "The effect of clustering coefficient and node degree on the robustness of cooperation". In: *Evolutionary Computation (CEC), 2013 IEEE Congress on.* IEEE. 2013, pp. 2833–2839.

[53] Guy Melancon. "Just how dense are dense graphs in the real world?: a methodological note". In: *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization.* ACM. 2006, pp. 1–7.

[54] Tomer Moscovich et al. "Topology-aware navigation in large networks". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* ACM. 2009, pp. 2319–2328.

[55] Mark EJ Newman. "Modularity and community structure in networks". In: *Proceedings of the national academy of sciences* 103.23 (2006), pp. 8577–8582.

[56] Takao Nishizeki and Md Saidur Rahman. *Planar graph drawing.* Vol. 12. World Scientific Publishing Co Inc, 2004.

[57] Adam Perer and Ben Shneiderman. "Integrating statistics and visualization: case studies of gaining clarity during exploratory data analysis". In: *Proceedings of the SIGCHI conference on Human Factors in computing systems.* ACM. 2008, pp. 265–274.

[58] Helen Purchase. "Which aesthetic has the greatest effect on human understanding?" In: *International Symposium on Graph Drawing.* Springer. 1997, pp. 248–261.

[59] Helen C Purchase. "Performance of layout algorithms: Comprehension, not computation". In: *Journal of Visual Languages & Computing* 9.6 (1998), pp. 647–657.

[60] Saif Ur Rehman, Asmat Ullah Khan, and Simon Fong. “Graph mining: A survey of graph mining techniques”. In: *Seventh International Conference on Digital Information Management*. IEEE. 2012, pp. 88–92.

[61] Edward M. Reingold and John S. Tilford. “Tidier drawings of trees”. In: *IEEE Transactions on Software Engineering* 2 (1981), pp. 223–228.

[62] Ronald A Rensink and Gideon Baldridge. “The perception of correlation in scatterplots”. In: *Computer Graphics Forum*. Vol. 29. 3. Wiley Online Library. 2010, pp. 1203–1210.

[63] Manojit Sarkar and Marc H Brown. “Graphical fisheye views”. In: *Communications of the ACM* 37.12 (1994), pp. 73–83.

[64] Rudolf Seisenberger. “Termgraph: Ein system zur zeichnerischen darstellung von strukturierten agenten und petrinetzen”. PhD thesis. 1990.

[65] John Stasko and Eugene Zhang. “Focus+ context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations”. In: *Information Visualization, 2000. InfoVis 2000. IEEE Symposium on*. IEEE. 2000, pp. 57–65.

[66] GK Stephen and G Pawel. “Graph drawing with intelligent placement”. In: *8th Symposium on Graph Drawing (GD)*. 2000, pp. 222–228.

[67] Stanley S Stevens. “On the psychophysical law.” In: *Psychological Review* 64.3 (1957), p. 153.

[68] Ying Tu and Han-Wei Shen. “Balloon focus: a seamless multi-focus+ context method for treemaps”. In: *IEEE Transactions on Visualization and Computer Graphics* 14.6 (2008).

[69] Daniel Tunkelang. *A practical approach to drawing undirected graphs*. Tech. rep. CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE, 1994.

[70] Frank Van Ham, Hans-Jörg Schulz, and Joan M Dimicco. “Honeycomb: Visual analysis of large scale social networks”. In: *IFIP Conference on Human-Computer Interaction*. Springer. 2009, pp. 429–442.

[71] Tatiana Von Landesberger et al. “Visual analysis of large graphs: state-of-the-art and future research challenges”. In: 30.6 (2011), pp. 1719–1749.

[72] Duncan J Watts and Steven H Strogatz. “Collective dynamics of ‘small-world’ networks”. In: *Nature* 393.6684 (1998), pp. 440–442.

[73] Graham Wilson et al. “Some like it hot: thermal feedback for mobile devices”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2011, pp. 2555–2564.

APPENDIX A

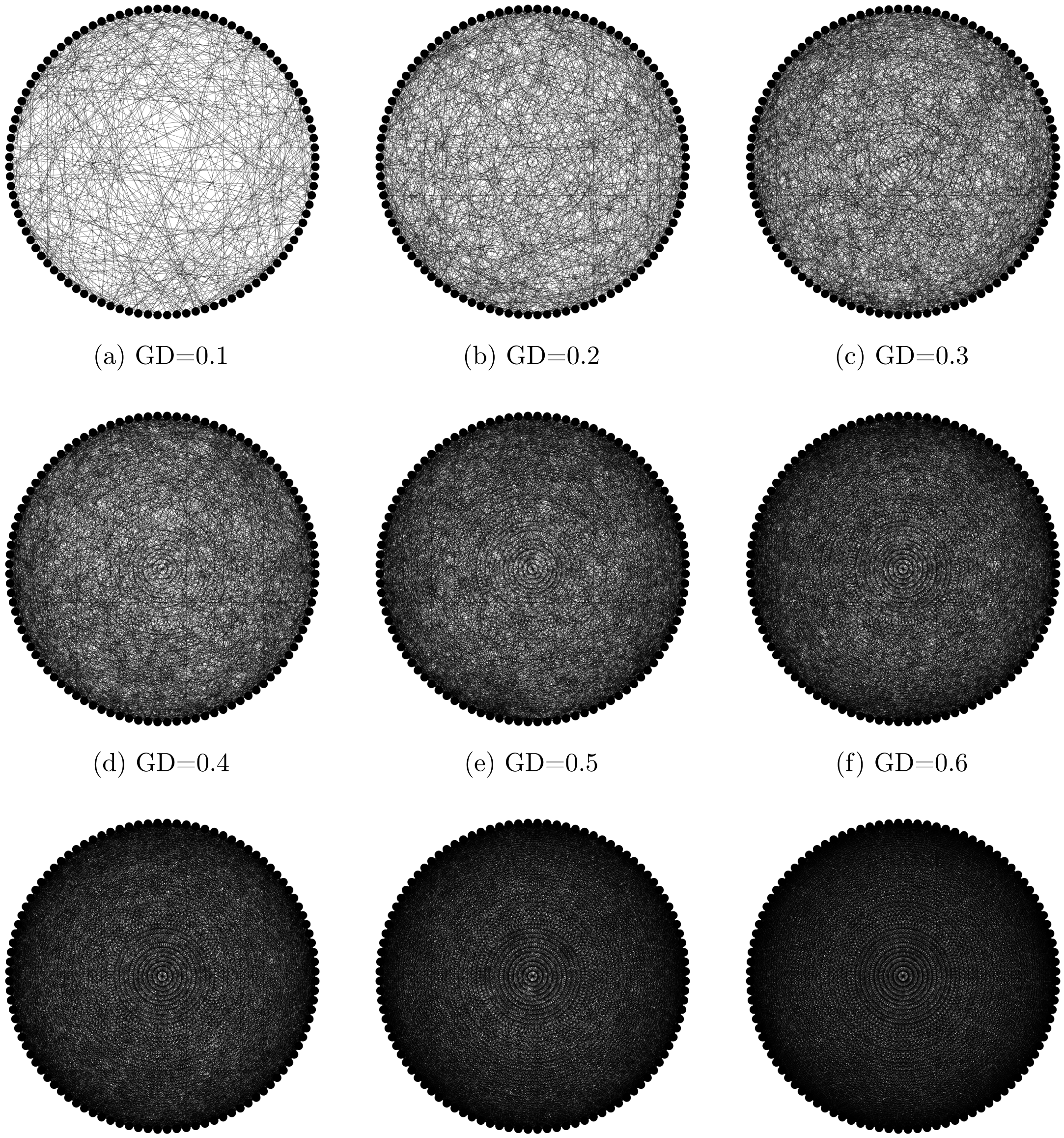
STIMULI USED FOR THE EXPERIMENTS

A sample of the graphs generated for experiment 1 (section 4.1) and experiment 2 (section 5.1) is presented in this section.

A.1 Graphs generated for experiment 1 (graph density)

For the graph density experiment, a total of 99000 graph drawings were generated for each layout. For each graph density value from 0.02 to 1, with step equals to 0.01, 50 different non-isomorphic graphs were generated. Then, each graph was drawn 20 times using the layout algorithm to produce 20 graph drawings that were all unique because of the stochastic nature of the graph layout algorithms. The following section shows one graph sampled for each density value from 0.1 to 0.9 (steps of 0.1), for each layout algorithm.

A.1.1 Graph drawings using the circular layout



(a) GD=0.1

(b) GD=0.2

(c) GD=0.3

(d) GD=0.4

(e) GD=0.5

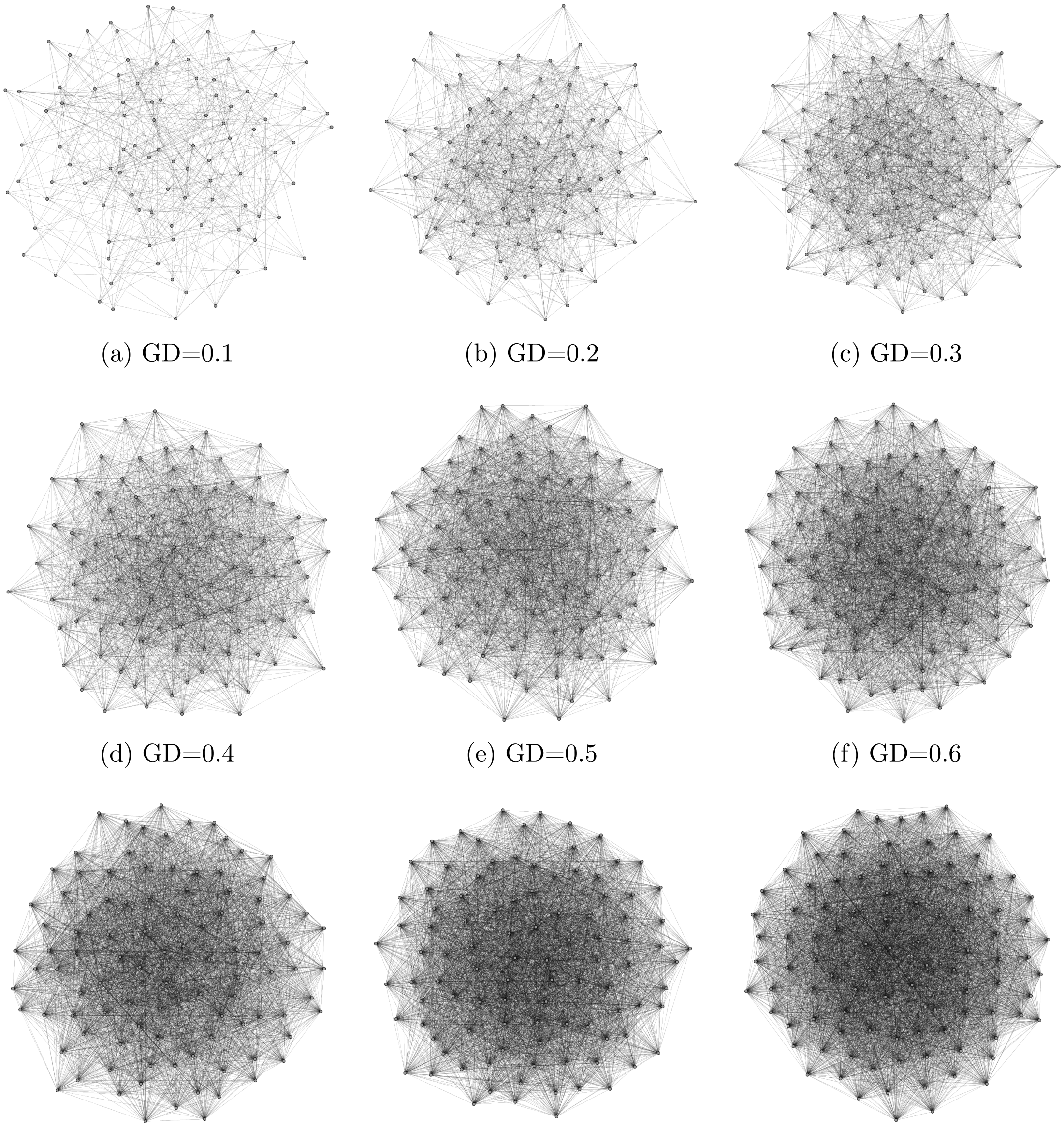
(f) GD=0.6

(g) GD=0.7

(h) GD=0.8

(i) GD=0.9

A.1.2 Graph drawings using the FD layout



(a) GD=0.1

(b) GD=0.2

(c) GD=0.3

(d) GD=0.4

(e) GD=0.5

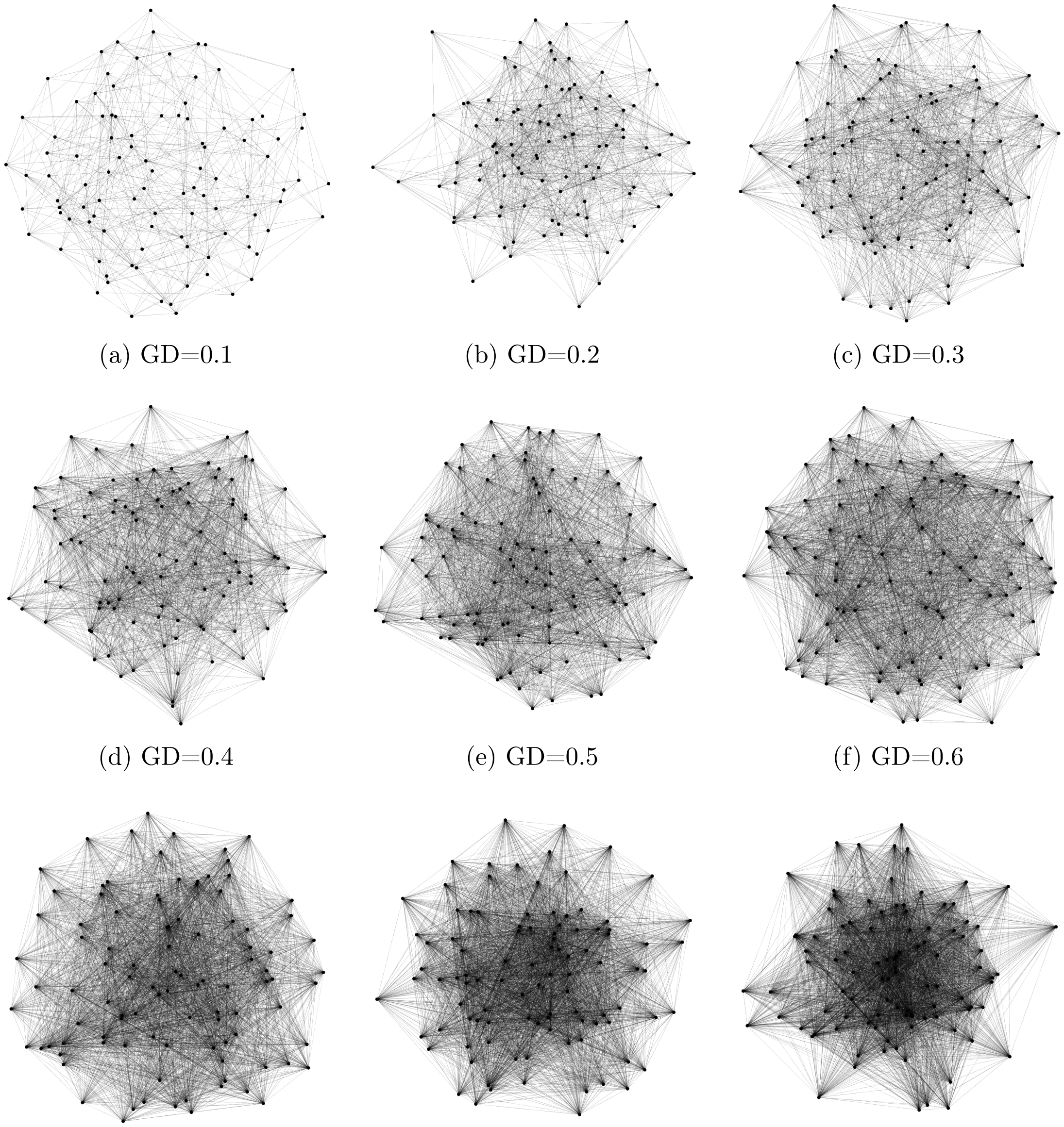
(f) GD=0.6

(g) GD=0.7

(h) GD=0.8

(i) GD=0.9

A.1.3 Graph drawings using the MDS layout



(a) GD=0.1

(b) GD=0.2

(c) GD=0.3

(d) GD=0.4

(e) GD=0.5

(f) GD=0.6

(g) GD=0.7

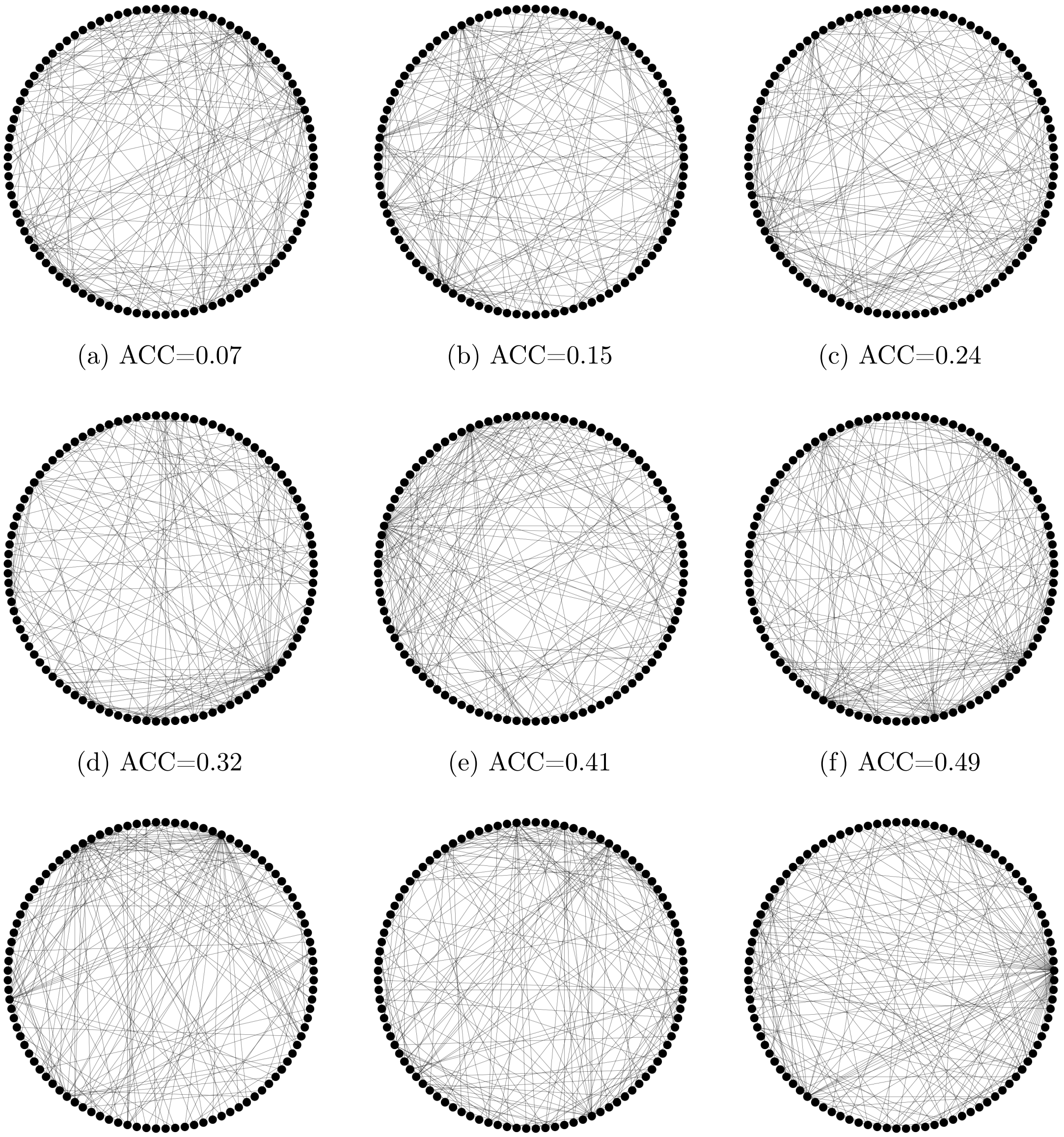
(h) GD=0.8

(i) GD=0.9

A.2 Graphs generated for experiment 2 (average clustering coefficient)

For the average clustering coefficient experiment, a total of 69000 graph drawings were generated for each layout. For each clustering coefficient value from 0.07 to 0.75, with step equals to 0.01, 50 different non-isomorphic graphs were generated. Then, each graph was drawn 20 times using the layout algorithm to produce 20 graph drawings that were all unique because of the stochastic nature of the graph layout algorithms. The following section shows one graph sampled for 9 density value from 0.07 to 0.75 (steps of ≈ 0.08), for each layout algorithm.

A.2.1 Graph drawings using the circular layout



(a) ACC=0.07

(b) ACC=0.15

(c) ACC=0.24

(d) ACC=0.32

(e) ACC=0.41

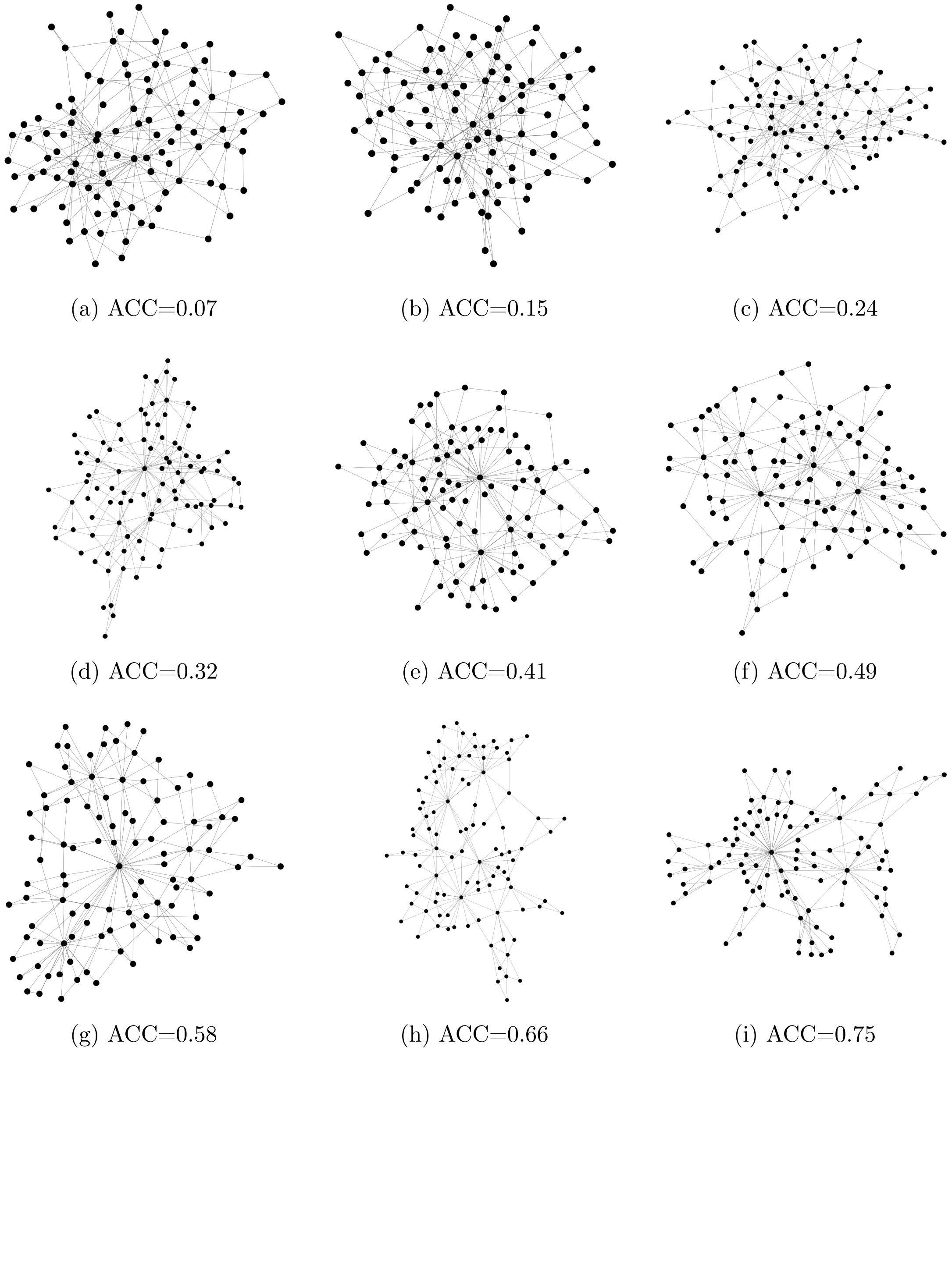
(f) ACC=0.49

(g) ACC=0.58

(h) ACC=0.66

(i) ACC=0.75

A.2.2 Graph drawings using the FD layout



(a) ACC=0.07

(b) ACC=0.15

(c) ACC=0.24

(d) ACC=0.32

(e) ACC=0.41

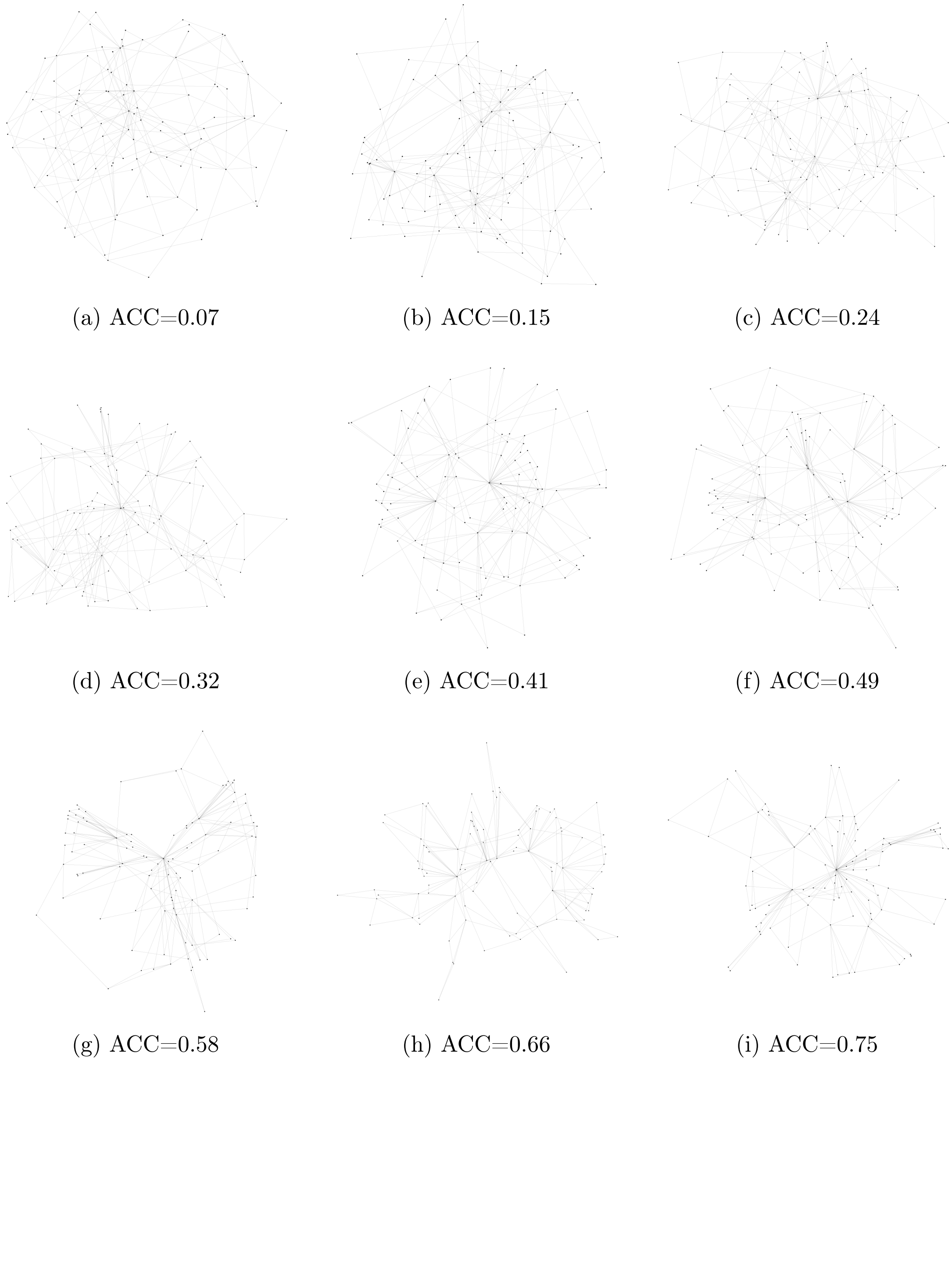
(f) ACC=0.49

(g) ACC=0.58

(h) ACC=0.66

(i) ACC=0.75

A.2.3 Graph drawings using the MDS layout



(a) ACC=0.07

(b) ACC=0.15

(c) ACC=0.24

(d) ACC=0.32

(e) ACC=0.41

(f) ACC=0.49

(g) ACC=0.58

(h) ACC=0.66

(i) ACC=0.75

APPENDIX B

RESPONSE OBTAINED IN THE EXPERIMENTS

A sample of responses obtained for experiment 1 (section 4.3) and experiment 2 (section 5.3) is presented in this section. For each base value and approach pair in the user study, the experiment system recorded the following:

1. the Amazon turk worker ID of the participant
2. the computed JND for the participant
3. the number of selections performed till the convergence of the staircase procedure
4. the average time taken per selection in seconds

At the end of the user study, the participants were also asked to optionally provide their demographic information.

B.1 Response obtained for experiment 1 (average local clustering coefficient)

B.1.1 User study of the circular layout

A total of 105 participants were recruited for the user study of circular layout. Among all the participants that shared their gender and age information post study, 60 were females while 42 were males. The age of the participants varied from 18 to 68. Table 3 shows a sample of 10 responses obtained from the user study of circular layout.

Table 3: Responses collected for the user study of the circular layout for perception of graph density

WorkerID	Target	Direction	JND	abs JND	Number of selections	Average time per selection
A298X7UE3B93PL	20	below	-2.66	2.66	50	1.01
A1H60UAPD5M113	70	below	-2.58	2.58	26	2.03
A195LN7C3VWZI8	40	above	5.83	5.83	46	1.51
A28VLQAH7JLSN9	20	above	3.08	3.08	26	1.55
A4397UDZV79P0	30	below	-2.25	2.25	35	3.26
AB76Q5DREW15V	50	above	6.5	6.5	24	1.42
A31JM9RECQGYEX	40	below	-7	7	49	0.99
A1EWV27J9TWQ54	60	above	6	6	24	1.48
A2RHJT0OMA09YH	70	below	-5.5	5.5	36	1.43
A4LIJVRU6DG61	20	above	2	2	29	3.76

B.1.2 User study of the FD layout

A total of 102 participants were recruited for the user study of FD layout. Among all the participants that shared their gender and age information post study, 62 were females while 34 were males. The age of the participants varied from 19 to 71. Table 4 shows a sample of 10 responses obtained from the user study of FD layout.

Table 4: Responses collected for the user study of the FD layout for perception of graph density

WorkerID	Target	Direction	JND	abs JND	Number of selections	Average time per selection
A7N3J27F3IL6M	30	below	-17.5	17.5	35	0.4
AS7ZK30LXDC46	40	above	3.375	3.375	26	2.1
A3FDLHGTF8I256	30	below	-4	4	40	1.3
A1S9EY8YUIGDYV	80	above	4.625	4.625	28	6.7
A15QGLWS8CNJFU	70	below	-11.5	11.5	26	1.3
A1669EVSVOHN54	80	above	3.25	3.25	25	2.4
A20E3U52YFBLHJ	20	above	3.5	3.5	46	0.6
AVJ2J2LH1WMQB	70	above	4.04	4.04	30	3.1
A5I4Y0GXJM69D	70	above	28.375	28.375	36	0.9
A2C5UH05QY2GZH	20	above	2.33	2.33	27	3.5

B.1.3 User study of the MDS layout

A total of 105 participants were recruited for the user study of MDS layout. Among all the participants that shared their gender and age information post study, 64 were females while 39 were males. The age of the participants varied from 18 to 76. Table 5 shows a sample of 10 responses obtained from the user study of MDS layout.

Table 5: Responses collected for the user study of the FD layout for perception of graph density

WorkerID	Target	Direction	JND	abs JND	Number of selections	Average time per selection
ASHYATOD3J5Z9	50	above	3	3	39	1.34
AG2YM9OWQP690	50	above	7.66	7.66	33	3.70
A15HRBYM3REOEV	30	below	-12.66	12.66	50	1.81
A1NITBXDX8TN7T	20	above	2.25	2.25	31	2.39
A3AD7HRMCBVVRZ	20	below	-4.5	4.5	29	1.09
A3OVF9XI01U9W7	40	above	3.25	3.25	25	2.54
A36KDWI1CGJFFA	40	above	3.875	3.875	44	2.71
A2CSV75E3JT58Y	20	below	-7	7	30	0.95
A1KYOQ0CHD4VUT	30	above	2	2	29	1.01
A2CWYA82P6BE09	50	below	-3.125	3.125	30	1.41

B.2 Response obtained for experiment 2 (average local clustering coefficient)

B.2.1 User study of the FD layout

A total of 201 participants were recruited for the user study of FD layout. Among all the participants that shared their gender and age information post study, 110 were females while 88 were males. The age of the participants varied from 18 to 70. Table 6 shows a sample of 10 responses obtained from the user study of FD layout.

Table 6: Responses collected for the user study of the FD layout for perception of ACC

WorkerID	Target	Direction	JND	abs JND	Number of selections	Average time per selection
AMXSOUSW1WJ6N	50	above	16.83	16.83	29	4.00
A3PB0KSWXYUEA	40	below	-32.375	32.375	44	0.39
A2VL807897JLT9	40	above	11.33	11.33	50	1.47
ASHABZG2VI0KN	20	below	-11.125	11.125	50	2.33
A3GOPXU6AK0K0Y	50	below	-41.625	41.625	39	0.27
AVCTTVFVLG90I	40	above	21.83	21.83	50	2.63
A39P7K4TGGHBUY	60	below	-9	9	27	1.90
A11IGE3ORQP5WI	20	above	26	26	38	1.97
A11IGE3ORQP5WI	50	below	-18.83	18.83	30	2.08
ATB2PODALQX0A	40	above	21	21	50	1.57

B.2.2 User study of the MDS layout

A total of 75 participants were recruited for the user study of MDS layout. Among all the participants that shared their gender and age information post study, 43 were females while 32 were males. The age of the participants varied from 19 to 69. Table 7 shows a sample of 10 responses obtained from the user study of FD layout.

Table 7: Responses collected for the user study of the MDS layout for perception of ACC

WorkerID	Target	Direction	JND	abs JND	Number of selections	Average time per selection
A1R37XA9QPSIAN	45	below	-35.375	35.375	46	1.68
A19HJF1JGJL7EY	15	above	39.83	39.83	59	2.76
ADRINUO0QTC33	35	below	-27.33	27.33	26	2.55
A2ZVWCVZO273CD	40	above	33.875	33.875	29	2.6
A2BEH9YQPHKP6A	40	below	-32.16	32.16	42	1.37
A15340BRCER2UO	20	above	54.5	54.5	54	2.43
A38G99MUB98LXL	60	below	-26.83	26.83	24	2.23
A195LN7C3VWZI8	45	above	20	20	27	5.01
A3ATZTGLM6AK05	70	below	-33.5	33.5	75	0.79
A32A08M12Z9V7K	25	above	38.66	38.66	75	4.41