Lines: 400

The Turing Test for Graph Drawing Algorithms

Helen C. Purchase[1], Daniel Archambault[2], Stephen Kobourov[3], Martin Nöllenburg[4], Sergey Pupyrev[5], and Hsiang-Yun Wu[4]

[1]University of Glasgow, UK. Helen.Purchase@glasgow.ac.uk
[2]Swansea University, UK. d.w.archambault@swansea.ac.uk
[3]University of Arizona, USA. kobourov@cs.arizona.edu
[4]TU Wien, Austria. noellenburg@ac.tuwien.ac.at, hsiang.yun.wu@acm.org
[5]Facebook, USA. spupyrev@gmail.com

Abstract. Do algorithms for drawing graphs pass the Turing Test? That is, are their outputs indistinguishable from graphs drawn by humans? We address this question through a human-centred experiment, focusing on 'small' graphs, of a size for which it would be reasonable for someone to choose to draw the graph manually. Overall, we find that hand-drawn layouts can be distinguished from those generated by graph drawing algorithms, although this is not always the case for graphs drawn by force-directed or multi-dimensional scaling algorithms, making these good candidates for Turing Test success. We show that, in general, hand-drawn graphs are judged to be of higher quality than automatically generated ones, although this result varies with graph size and algorithm.

Keywords: Empirical studies, Graph Drawing Algorithms, Turing Test

1 Introduction

It is common practice to use node-link diagrams when presenting graphs to an audience (e.g., online, in an article, to support a verbal presentation, or for educational purposes), rather than the alternatives of adjacency matrices or edge lists. Automatic graph layout algorithms replace the need for a human to draw graphs; it is important to determine how well these algorithms fulfil the task of replacing this human activity,

Such algorithms are essential for creating drawings of large graphs; it is less clear that this is the case for drawing smaller graphs. In our experience as graph drawing researchers, it is often preferable to draw a small graph ourselves, how we wish to depict it, than be beholden to the layout criteria of automatic algorithms.

The question therefore arises: are automatic graph layout algorithms any use for small graphs? Indeed, for small graphs, is it even possible to tell the difference? If automatic graph layout algorithms were doing their job properly for small graphs, then they should produce drawings not dissimilar to those we would choose to create by hand.

Distinguishing human and algorithmic graph drawings can be considered a 'Turing Test'; as in Turing's 1950 'Imitation Game' [44], if someone cannot tell the difference between machine output and human output more than half the time, the machine passes the Turing Test. Thus, if someone cannot tell the difference between an algorithmically-drawn graph and a hand-drawn graph more than half the time, the algorithm passes the Turing Test: it is doing as good a job as human graph drawers. Of course, algorithms are useful for non-experts and for large graphs that cannot be drawn by humans effectively, but in the context of experts presenting a small graph, can their creations be distinguished from products from layout algorithms? Turing Tests have never yet been performed on graph layout algorithms.

This paper presents the results of an experiment where participants were asked to distinguish between small hand-drawn graphs and those created by four common graph layout algorithms. Using different algorithms and graphs of different size allows us to investigate under what conditions an algorithm might pass the Turing Test. Our Turing Test results led us to also ask, in common with the *NPR Turing Test* observational study [30], which of the two methods of graph drawing (by hand, or by algorithm) produce better drawings. We find that distinguishing hand-drawn layouts from automatically generated ones depends on the type of the layout algorithm, and that subjectively determined quality depends on graph size and the type of the algorithm.

2 Related Work

2.1 Automatic Graph Layout algorithms

We focus on four popular families of layout algorithms [13, 25]: force-directed, stress-minimisation, circular and orthogonal.

Most general-purpose graph layout algorithms use a force-directed (FD) [15, 19] or stress model [12, 34]. FD works well for small graphs, but does not scale for large networks. Techniques to improve scalability often involve multilevel approaches, where a sequence of progressively coarser graphs is extracted from the graph, followed by a sequence of progressively finer layouts, ending with a layout for the entire graph [8, 21, 26, 28, 29].

Stress minimisation, introduced in the general context of multi-dimensional scaling (MDS) [35] is also frequently used to draw graphs [31, 40]. Simple stress functions can be optimised by exploiting fast algebraic operations such as majorisation. Modifications to the stress model include the strain model (classical scaling) [43], PivotMDS [12], COAST [22], and MaxEnt [23].

Circular layout algorithms [41] place nodes evenly around a circle with edges drawn as straight lines. Layout quality (in particular the number of crossings) is influenced by the order of the nodes on the circle. Crossing minimisation in circular layouts is NP-hard [36], and various heuristics attempt to find good vertex orderings [9, 24, 33].

The orthogonal drawing style [16] is popular in applications requiring a clean and schematic appearance (e.g., in software engineering or database schema).

Edges are drawn as polylines of horizontal and vertical segments only. Orthogonal layouts have been investigated for planar graphs of maximum degree four [42], non-planar graphs [10] and graphs with nodes of higher degree [11, 20].

We seek to understand if drawings produced by these types of algorithms can be distinguished from human-generated diagrams for small networks. We do this by asking experimental participants to identify the hand-drawn layout when it is paired with an algorithmically-created one.

2.2 Studies of Human-Created Graph Layouts

Early user studies [37, 38] confirmed that many of the aesthetic criteria incorporated in layout algorithms (e.g., uniform edge length, crossing minimisation) correlate with user performance in tasks such as path finding. Van Ham and Rogowitz [27] investigated how humans modified given small graph layouts so as to represent the structure of these graphs. They found that force-directed layouts were already good representations of human vertex distribution and cluster separation. Dwyer et al. [14] focused on the suitability of graph drawings for four particular tasks (identifying cliques, cut nodes, long paths and nodes of low degree), and found that the force-based automatic layout received the highest preference ratings, but the best manual drawings could compete with these layouts. Circular and orthogonal layouts were considerably less effective. Purchase et al. [39] presented graph data to participants as adjacency lists and asked them to create drawings by sketching; their findings include that the participants preferred planar layouts with straight-line edges (except for some non-straight edges in the outer face), nodes aligned with an (invisible) grid, and somewhat similar edge lengths. Kieffer et al. [32] focused on orthogonal graph layouts, asking participants to draw a few small graphs (13 or fewer nodes) orthogonally by hand. The human drawings were compared to orthogonal layouts generated by yEd [46] and the best human layouts were consistently ranked better than automatic ones. They then developed an algorithm for creating human-like orthogonal drawings.

This paper builds on this prior work by considering drawings of small to medium-sized graphs (up to 108 nodes) and an example from each of four families of standard graph layout algorithms. We address the question of whether people can distinguish between algorithmic and human created drawings, and if so, is this the case for all layout algorithms?

3 Experiment

3.1 Stimuli

The Graphs. Our experiment compares unconstrained hand-drawn graphs with the same graphs laid out using different algorithmic approaches. We considered 24 graphs, from which we selected 9, based on the following criteria:

- A balanced split between real-world graphs and abstract graphs, the abstract graphs being ones of graph-theoretic interest;

Table 1. Characteristics of the experimental graphs. The *size* column indicates how the graphs were divided into sub-sets (small, medium, large) for the purposes of the experiment; (rw): real-world graphs; (ab): abstract graphs.

graph	nodes	edges	density	mean shortest path	clustering coefficient	diam.	planar	size	reference
G_1(rw)	108	156	0.03	5.03	0.11	11	N	L	Causes of obesity [7]
G_2(rw)	22	164	0.71	1.30	0.78	2	N	S	Causes of social problems in Alberta, Canada [4]
G_3(rw)	85	104	0.03	6.05	0.04	13	Y	L	Cross posting users on a newsgroup (final timeslice) [18]
G_4(rw)	34	77	0.14	2.45	0.48	5	N	M	Social network [47]
G_5(ab)	20	30	0.16	2.63	0.00	5	Y	S	Fullerene graph with 20 nodes [3]
G_6(ab)	24	38	0.14	3.41	0.64	6	N	S	A block graph (chordal, every biconnected component is a clique) [2]
G_7(ab)	42	113	0.13	2.55	0.48	5	Y	M	A maximal planar graph [6]
G_8(ab)	37	71	0.11	2.76	0.70	5	Y	M	A planar 2-tree [5]
G_9(ab)	18	27	0.18	2.41	0.00	4	N	S	Pappus graph (bipartite, 3-regular) [1]
mean	43.3	86.7	0.18	3.18	0.36	6.2			
median	34	77	0.14	2.63	0.48	5			

– A balanced split between planar and non-planar graphs;
– A range in the number of nodes between 15 and 108;
– A range in the number of edges (for our graphs, between 27 and 164);
– Connected and undirected graphs only: directionality was removed from the real-world graphs as necessary.

Our graphs exhibit a range of values for other graph characteristics: diameter, density, average shortest path length, and clustering coefficients (Table 1).

The Algorithms. We included examples of major families of graph drawing algorithms (Table 2: force-directed, stress-based, circular, orthogonal), as implemented in yEd [46] and GraphViz [17]. HOLA [32] was considered, but its orthogonal design was deliberately based on human preferences (unlike the other algorithms), and so its inclusion would introduce a bias that could distort human judgements. We considered structure-specific algorithms (e.g., algorithms designed for planar graphs or trees), but for generality used generic algorithms that could handle all nine graphs, leaving specific algorithms for future work.

The Hand-Created Drawings. The process of creating hand-drawn graphs mimicked the context of a graph drawing researcher deciding whether to manually draw a small graph, or to use a well-established graph layout algorithm. Thus, the graphs were drawn in the knowledge they would compete against drawings created by algorithms, making the Turing test as hard as possible. This process was therefore a mini-experiment, with four of the authors (all with

Table 2. The four graph layout algorithms used.

algorithm ID	algorithm type	original name	parameters
A_{FD}	force-directed	Organic [46]	default
A_{MDS}	stress-based	MDS [17]	default
A_C	circular	Circular [46]	default
A_O	orthogonal	Orthogonal [46]	classic, default

graph drawing expertise, called the 'drawers', D_1-D_4) as participants, the context of the study being clear to them. While the drawers might have recognised some of the graphs they were asked to draw, this scenario is comparable to a real-world situation where graph drawing researchers might know the nature of the graph to be drawn.

The first author asked the drawers to lay out the graphs using yEd [46], starting from a random layout (the yEd 'Random' tool). There were no other instructions: it was not specified, for example, that edges needed to be straight lines rather than splines or multiple segments, nor that nodes should not overlap, nor edges cross over nodes. To improve ecological validity, all drawers were told that they could use yEd tools to support their drawing process if they wished (as likely to happen in practice). However, somewhat surprisingly, they all drew the graphs without any yEd tool support (automatic layout or otherwise) (Appendix D). The drawers suggested doing the exercise again on a 'manually-adjusted' basis; that is, using the output from a yEd layout algorithm of their choice as an initial starting point. However, once we paired the algorithmic drawings with their manually-adjusted versions, most of them were visually almost identical. We therefore only used the initial hand-drawn versions.

The mini-experiment output is a set of visual stimuli comprising 9 graphs ($G_1, ..., G_9$), each with four layout algorithms applied G_1A_{FD}, G_1A_{MDS}, $\ldots$, G_2A_{C}, $\ldots$, G_9A_O) and each with four hand-drawn versions (G_1D_1, G_1D_2, $\ldots$, G_2D_1, $\ldots$, G_9D_4), all represented in yEd. All 72 drawings were subject to the same automatic scaling process to ensure the same vertex size and edge thickness. After scaling, all drawings were automatically converted into jpeg images.

3.2 Experimental Design

Each experimental trial (Fig. 1) comprises two versions of the same graph, one hand-drawn, and one created by a layout algorithm. For each graph, we firstly paired the four algorithmic versions (on the left) with the four hand-drawn versions (right) (16 pairs). We then flipped the algorithmic versions along the y axis (reducing the possibility of participants remembering the algorithm drawings), and paired the flipped versions (right) with the four drawn versions (left) (32 pairs for each graph). Putting all graphs in one experiment means 288 trials, an unreasonably long experiment. The alternative of running a separate experiment for each graph means several very small experiments, greatly increasing

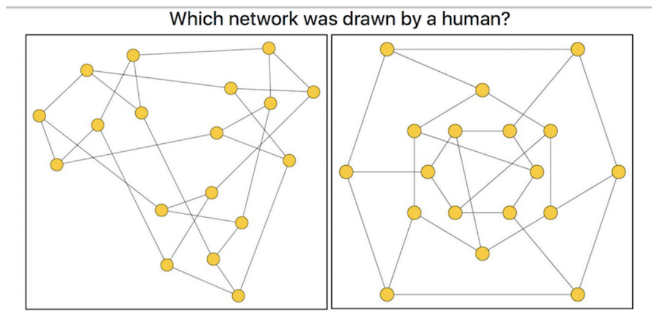


Fig. 1. Screen shot of the experimental system.

the number of participants needed. As a compromise, we divided our 9 graphs into three sets, (loosely 'small', 'medium' and 'large' (Table 1)), a convenience decision so as to reduce the duration of each experiment while ensuring we would be able to recruit enough participants. We thus had three sub-experiments, one 'small' (128 trials), one 'medium' (96 trials) and one 'large' (64 trials).

Using a custom-built online experimental system, participants read instructions and information about graphs (referred to as 'networks') and indicated consent before proceeding. They were told it would always be the case that the two drawings presented were the same graph. Twelve practice trials used a different graph of similar size for familiarisation purposes. Experimental trials were presented in random order, with no distinction between graphs. Participants took a self-timed break every 20 trials, and demographic data was collected.

4 Results and Data Analysis

The experimental link was distributed to authors' colleagues, students, family and friends. Participants were considered outliers if their mean time over all trials was unreasonably low (less than 1 second, $n = 2$), or if they consistently responded one side for a large number of consecutive trials (e.g., always left, $n = 1$). No participants consistently alternated left and right. We removed the data from one participant who used a very small screen (198×332 pixels), unconvinced that the stimuli could be perceived sufficiently well. Although some participants did not complete the experiment, since the answer to each trial is a data point in its own right (i.e., it is independent and its value to the experiment does not depend on answers to any other trial), we retained all data for participants who completed at least 3/4 of the trials, inferring that those who did not do so ($n = 20$) might not have taken the experiment seriously.

Data from 46 participants was analysed; a total of 4364 independent decisions. We categorised participants as expert ($n = 21$) if their self-declared knowledge of network drawings was 'expert', 'highly knowledgeable', or 'knowledgeable', and novice ($n = 22$) for 'somewhat knowledgeable' or 'no knowledge'. Three participants did not provide full demographic details (Appendix C).

4.1 Data Analysis Methods

Our data was analysed in three parts: Part 1 investigates the extent to which 'human' was chosen over 'algorithm', comparing the proportion of responses with random selection. We look at overall responses, responses for each algorithm, for each graph size, for novice and expert participants, for planar and non-planar graphs, and consider the combination of graph size and algorithm. The Binomial distribution test compares observed proportion against the 'random' proportion of 0.5, where each trial is independent; its calculated p-value represents the probability that the mean of the population distribution (based on the observed samples) is equal to 0.5. A p-value < 0.05 indicates a significant result: that is, the observed choice proportion is so much greater than 0.5 that there is a very low probability that the hand-drawn and algorithmically drawn graphs cannot be distinguished; statistically, this means there is insufficient evidence to indicate Turing Test success. A p-value > 0.05 is a high probability that hand-drawn and algorithmically drawn graphs cannot be distinguished: thus, Turing Test success. We apply p-value Bonferroni corrections when dividing the data sets.

Part 2 considers response times with respect to different algorithms, sizes, expertise, and planarity, using non-parametric tests since our data is not normally distributed. Response time is considered as a proxy for the perception of difficulty of the task: participants will take longer if they find the task difficult.

Part 3 identifies trials with extreme responses (high or low response time, or extreme proportional choice).

A choice for a hand-drawn graph is scored as 1; a choice for an algorithmic drawings is 0. Thus, proportions > 0.5 indicate that the human drawing was selected more often on average. Proportions < 0.5 indicate that the algorithmic drawing was (incorrectly) selected with greater frequency.

4.2 Results

Choice of drawing. Our hypotheses are:

- H_0: It is not possible to distinguish algorithmic drawings from hand-drawn ones; thus, the true proportion $= 0.5$; the algorithm passes the Turing test. This hypothesis is accepted if the Binominal p-value > 0.05.
- H_1: It is possible to distinguish algorithmic drawings from hand-drawn ones; thus, the true proportion $\neq 0.5$.

Binomial test results over all 4364 data points are shown in Table 3. Accepting H_0 means it is not possible to distinguish between hand-drawn and algorithmic drawings: the Turing Tests succeeds. Rejecting it means that there is insufficient support for the hypothesis; we infer that telling the difference is possible. There are no proportions < 0.5, so no cases where, on average, algorithmically-drawn graphs were incorrectly selected more often than hand-drawn ones.

The results indicate that people can distinguish between algorithmic and hand-drawn graphs (over all graphs and algorithms), correctly choosing the hand-drawn graph 56% of the time ($p < 0.001$). This result applies equally well

Table 3. Binomial test results for 'Which network was drawn by a human?' Accepting H_0 indicates Turing Test 'pass'. Although $0.049 < 0.05$, statistical correction means the MDS p-value threshold is $0.05/4 = 0.0125$. The corrected Novice p-value threshold is $0.05/2 = 0.025$, a significant result.

	Number of samples	Mean response time (s)	Observed proportion	Binomial p-value	Result
All trials	4364	3.14	0.56	$p < 0.001$	reject H_0
Force-Directed (A_{FD})	1094	4.26	0.51	$p = 0.566$	accept H_0
MDS (A_{MDS})	1090	3.32	0.53	$p = 0.049$	reject H_0
Circular (A_C)	1090	2.85	0.56	$p < 0.001$	reject H_0
Orthogonal (A_O)	1090	2.79	0.65	$p < 0.001$	reject H_0
Small graphs (G_2, G_5, G_6, G_9)	1656	2.58	0.55	$p < 0.001$	reject H_0
Medium graphs (G_4, G_7, G_8)	1817	3.08	0.55	$p < 0.001$	reject H_0
Large graphs (G_1, G_3)	891	4.28	0.62	$p < 0.001$	reject H_0
Expert participants	1915	3.99	0.63	$p < 0.001$	reject H_0
Novice participants	2101	2.74	0.53	$p = 0.016$	reject H_0
Planar graphs	2069	3.15	0.55	$p < 0.001$	reject H_0
Non-planar graphs	2295	3.49	0.58	$p < 0.001$	reject H_0

Table 4. Binomial test results by graph size and algorithm; * indicates responses sufficiently close to random for Turing Test 'pass'.

	Force-Directed		MDS		Circular		Orthogonal	
	proportion	p-value	proportion	p-value	proportion	p-value	proportion	p-value
small	0.52*	0.432	0.57*	0.006	0.51*	0.786	0.62	< 0.001
medium	0.49*	0.851	0.52*	0.542	0.53*	0.205	0.64	< 0.001
large	0.52*	0.640	0.49*	0.789	0.73	< 0.001	0.74	< 0.001

regardless of graph size, viewer expertise, or graph planarity: the tests all reveal significant difference between the observed proportion and 0.5. Thus, overall, the Turing test fails.

There is a difference, however, when the algorithm is taken into account: the observed proportion for Force-Directed algorithm trials was 0.51, sufficiently close to the random response proportion of 0.50 that we can accept H_0, and state that this algorithm passes the Turing Test. The proportion of 0.53 for MDS is very close (but not really close enough in statistical terms), and we clearly reject H_0 for circular and orthogonal algorithms.

The size/algorithm combination (threshold p-value $= 0.05/12 = 0.0042$) reveals additional results according to the size of the graph (Table 4). As expected, the Force-Directed algorithm gives proportions close to 0.5 for all graph sizes. The MDS results suggest Turing Test success for all three sizes when analysed separately (albeit a marginal result for the smallest graphs), even though the overall MDS result reported above (at $p = 0.049$) indicates rejection of the null hypothesis. The MDS result is therefore clearly on the boundary of success. There are Turing Test passes for small and medium graphs for the Circular algorithm.

Response Time. Non-parametric tests on response time for algorithm and graph size (Table 3) reveals that MDS decisions were slower than orthogonal ones (adjusted pairwise comparison after repeated measures Freidman, $p = 0.022$), decisions on large graphs were slower than on small graphs (adjusted pairwise comparison after independent measures Kruskal Wallis, $p = 0.039$), and experts made slower decisions than novices (independent measures Mann-Whitney, $p = 0.014$). There was no statistical difference between response times with respect to graph planarity.

Extreme Examples Extreme trials (response time: Figure 2; proportion: Figure 3) are identified as G_iA_j and G_iD_k: G_i (graph), A_j (algorithm), D_k(drawer). All experimental stimuli jpeg files can be found in the supplementary material included with the paper submission.

Three slow trials relate to a particular FD graph, suggesting that this form of drawing was seen by participants as possibly hand-drawn – it shows clusters and symmetry, while the drawers all attempted to remove crosses. The combinations of $G_4A_{\mathrm{MDS}}/G_4D_4$ and G_7A_C/G_7D_4 (top row of Figure 2) are interesting because, for each, the overall shape of the human-drawn graph is similar to that produced by the algorithm: it is not hard to see why participants found this choice difficult. Three quick responses ($G_5A_{\mathrm{FD}}/G_5D_3$, G_5A_C/G_5D_4, $G_9A_{\mathrm{MDS}}/G_9D_1$, bottom row of Figure 2) demonstrate effort on the part of the drawer to depict symmetry that is not highlighted by the algorithms; the other two relate to the orthogonal algorithm, which, as noted above, produced worst performance in making a human *vs* algorithm judgements.

Of the four combinations where participants gave mostly correct responses, it is not hard to see why for G_1A_C/G_1D_2 and G_1A_C/G_1D_1 (top row of Figure 3), since the human-drawn graphs lack any clear structure or visual elegance in comparison with those created by the circular algorithm. The fact that G_5A_{MDS} is geometrically precise in its node positioning (while G_5D_2 has slight mis-positionings) can explain the 0.92 accuracy for this combination, although we note that this decision still took above average time (32.4 seconds). More difficult to explain is the high proportion associated with $G_6A_{\mathrm{FD}}/G_6D_3$, since the human drawing is highly structured and symmetrical. Of the combinations where the average accuracy is low, three algorithmic drawings depict some extent of symmetry (G_3A_{MDS}, G_9A_C, G_5A_{FD}, bottom row of Figure 3), while the fourth is compared against a human drawing which used an approach that, if adopted by an algorithm, would have resulted in a more geometrically precise diagram. The examples in Figure 3 (top and bottom rows) suggest that regular node and edge placements (that is, grid-like or evenly spaced on a circle), indicate an algorithmically-drawn graph.

Key factors affecting the human *vs* algorithm choice were thus depiction of symmetry (even if only approximate), and geometric precision (i.e. very precise node placement, with regular spacing or grid-like).

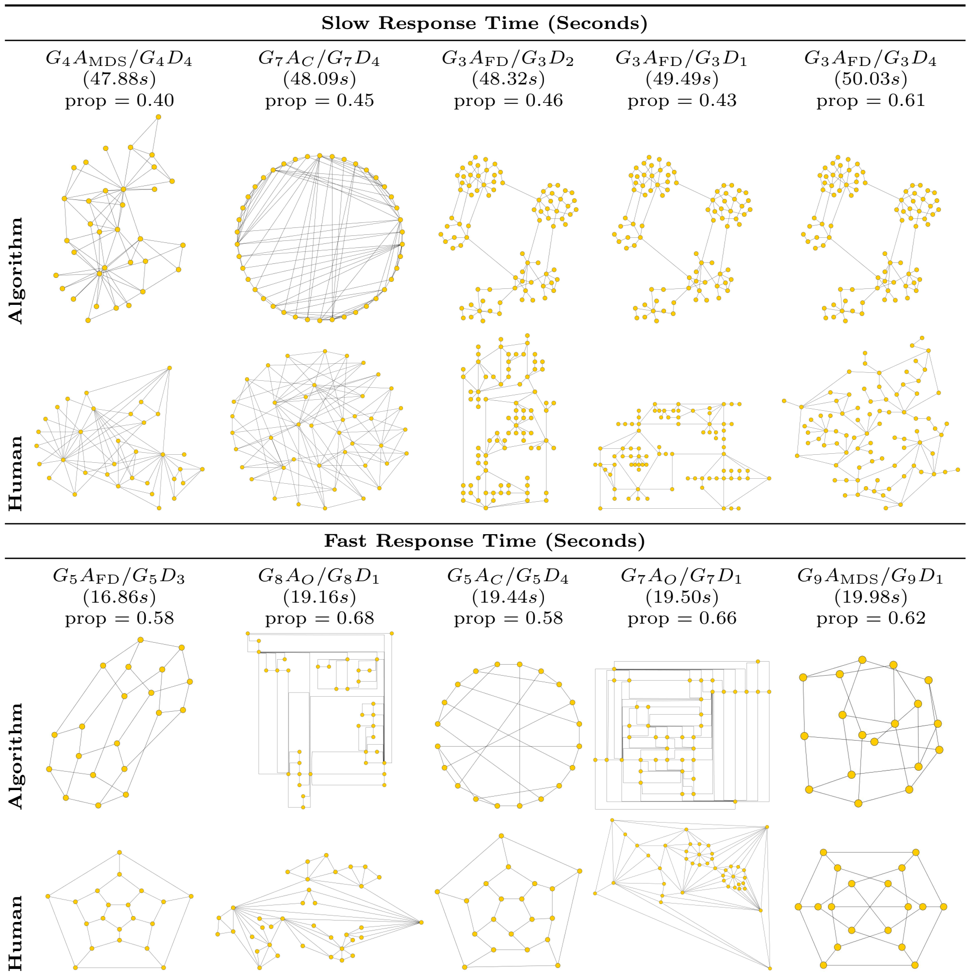


Fig. 2. Trials with slow response times (top) and quick response times (bottom). Time in seconds, and human-selection proportion shown.

5 Discussion

In general, over all graphs and algorithms, participants can correctly distinguish hand-drawn layouts from algorithmically created ones: graph drawing algorithms (in general) effectively fail the Turing Test. The only exception is the Force-Directed algorithm, where we did not find evidence that participants could reliably distinguish between the algorithmic and hand-drawn layouts. We speculate this might be because our drawers (consciously or unconsciously) created drawings with similar FD layout principles in mind: separating unconnected nodes, and clustering connected ones together. The MDS algorithm provided

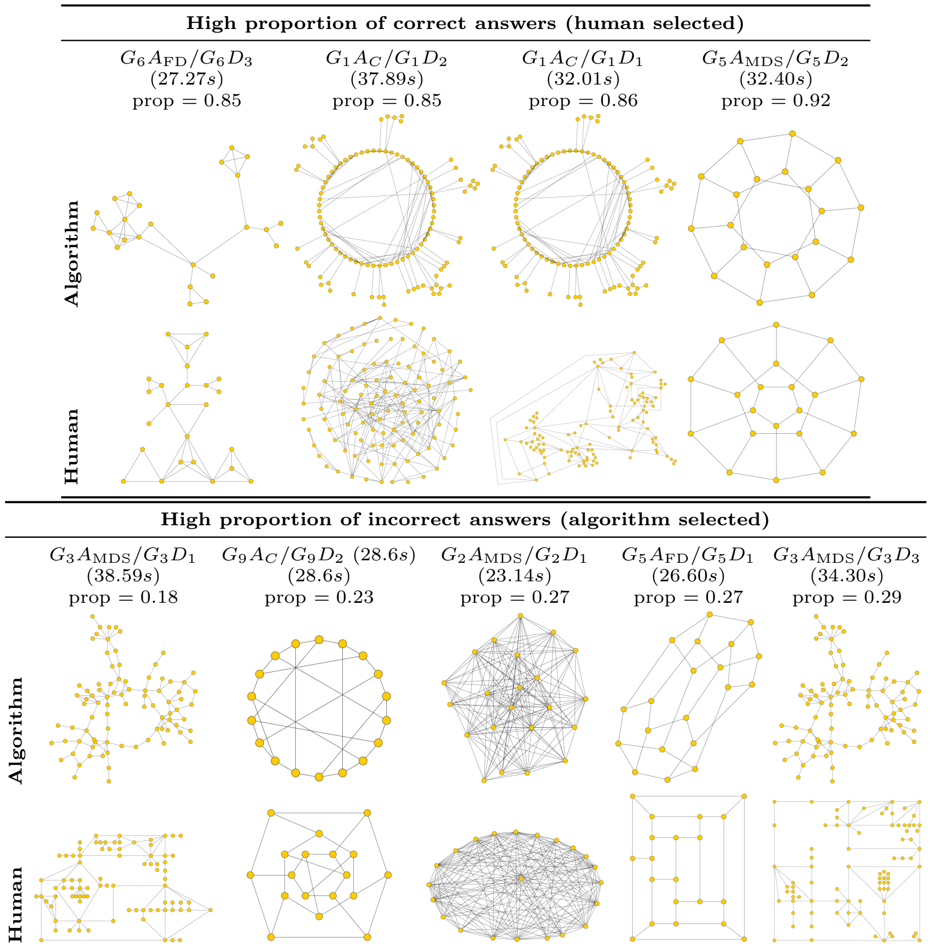


Fig. 3. Trials with a high proportion of correct (human drawing chosen, upper) and incorrect (algorithm drawing chosen, lower) answers.

some evidence of passing the test (in particular for medium and large graphs); it produces similar shapes to FD.

We were not surprised that it was easy to distinguish circular (especially large circular) and orthogonal graph drawings from hand-drawn ones, since they make use of precise node placement: equal separation around the circle circumference, placement on equally-spaced horizontal lines or on an underlying unit grid. While the human drawers sometimes used such placements (G_2D_1 and G_5D_1 in Figure 3), in many cases (G_8D_1 in Figure 2, G_5D_2 in Figure 3) they did not. We were also not surprised to find that larger graphs took more time than the smaller ones, but were surprised that experts took longer than novices –

Table 5. Results for the 'Which is better' question, by graph size and algorithm. * indicates statistically significant results ($p < 0.05/12 = 0.0042$)

	Force-Directed		MDS		Circular		Orthogonal	
	proportion	p-value	proportion	p-value	proportion	p-value	proportion	p-value
small	0.83*	< 0.001	0.68*	< 0.001	0.55	0.040	0.62*	< 0.001
medium	0.44	0.006	0.42*	0.001	0.62*	< 0.001	0.74*	< 0.001
large	0.19*	< 0.001	0.42	0.009	0.41*	0.002	0.63*	< 0.001

we had expected the converse; perhaps experts made more considered analytical decisions as opposed to novices' more spontaneous ones.

6 The Quality of the Drawings

Our study shows that some graph drawing algorithms produce diagrams that are obviously perceived as different from those drawn by graph drawing experts. This raises the question: if algorithmic drawings are perceived as being different from hand-drawn ones, are they any better? And even if they are not perceived as different, is there a perceived difference in quality?

We followed our Turing experiment with a supplementary, almost identical study, using the same paired stimuli and experimental system. The only difference was the question asked: 'Which drawing is better?'. We deliberately did not give a definition for 'better', since (at least for this initial study), we wished to get an overall judgement, rather than, for example, one based on a particular task or defined aesthetic. 52 participants took part, producing a total of 4887 data points. As before, hand-drawn graphs are scored 1, and algorithmic drawings 0. Thus, proportions > 0.5 indicate the human drawing was, on average, considered better. Over all graphs and algorithms, the vote was for hand-drawn graphs (proportion=0.57, $p < 0.001$). However, size and algorithm data show variations within this overall result (Table 5). Hand-drawn graphs were always preferred over orthogonal drawings; FD and MDS were only preferred for medium and large graphs, and circular only for the large graphs.

Thus, even when it is not possible to distinguish between hand-drawn and algorithmic drawings (as for FD and MDS), subjective judgement determines that algorithmic ones are 'better', especially for the larger graphs. The orthogonal algorithm had no wins: it did not pass the Turing Test, and was always considered worse than the hand-drawn versions. There were mixed results for the circular algorithm: easy to distinguish from hand-drawn layouts when small or medium, and only preferred when large.

7 Conclusions and Future Work

This is the first experiment that compares graphs drawn by graph drawing researchers to those produced by graph drawing algorithms as a Turing Test. Overall, we found that hand-drawn graphs could be reliably distinguished from those generated by algorithms – thus, on average, Turing Test failure. However, we did not find evidence that force-directed and (marginally) MDS algorithms could be reliably distinguished from hand-drawn layouts – they therefore effectively 'pass' the Turing Test. We speculate that this is the case because of the prevalence of these algorithms in the popular media (e.g., for depicting social networks); further studies could establish exactly why these two algorithms perform differently from the others.

The generalisability of our conclusions is, of course, limited by our experimental scope. While we used a good range of real-world and abstract graphs, differently sized graphs, planar and non-planar graphs, and good coverage of various graph metrics, our data set comprises nine experimental graphs. Using only 'small' graphs (15 to 108 nodes) was an obvious design decision when considering the feasibility of creating hand-drawn layouts. We chose four common layout algorithms representing different approaches, and four human drawers (experts in graph drawing research). Despite these experimental limitations, our results represent the first empirical attempt to compare perception of a range of hand-drawn versus algorithmic graph layouts as a 'Turing Test'.

Our motivation for these studies arose from a desire to determine whether algorithms depicting small graphs produce results that are similar to human efforts. Our results show that, in general, people notice when a graph has been hand-drawn. This result must, of course, be weighed against the length of time that it takes to draw a graph: we found that it takes much longer than we had anticipated to create drawings by hand. We also need to consider that, when considering the algorithmic approaches separately, some algorithmic versions were considered 'better' than the hand-drawn ones – the notable exception being the orthogonal algorithm.

Graph drawing algorithms are often inspired by assumptions about what a human would do in generating a drawing. Therefore, understanding what makes a drawing human-like will help inform future algorithm designers to make algorithms of higher quality. In future work, we would like to explore whether we get similar results if we explicitly match graph structure with graph algorithm (e.g., tree algorithms for trees, planar algorithms for planar graphs), use other less common algorithms (e.g., HOLA [32], Wang et al. [45]), and use graphs drawn by a wider range of people (including non-experts). In addition, gathering both quantitative and qualitative data in future studies will help determine those attributes of a graph drawing that suggest that it is human-like or machine-like.

References

1. Bipartite 3-regular undirected graph (2019), `http://reference.wolfram.com/language/ref/GraphData.html`, accessed: 2019-08-08
2. Block graph. `https://en.wikipedia.org/wiki/Block_graph` (2019), accessed: 2019-08-08
3. Fullerene graphs. `https://commons.wikimedia.org/wiki/Fullerene_graphs/` (2019), accessed: 2019-08-08
4. G_2. `http://haikuanalytics.com/portfolio/` (2019), accessed: 2019-08-08

5. K-tree. `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.71.3875` (2019), page 5. Accessed: 2019-08-08
6. Planar graph. `https://arxiv.org/pdf/1707.08833.pdf` (2019), page 18. Accessed: 2019-08-08
7. Allender, S., Owen, B., Kuhlberg, J., Lowe, J., Nagorcka-Smith, P., Whelan, J., Bell, C.: A community based systems diagram of obesity causes. PLoS ONE **10**(7), e0129683 (2015), `https://doi.org/10.1371/journal.pone.0129683`
8. Bartel, G., Gutwenger, C., Klein, K., Mutzel, P.: An experimental evaluation of multilevel layout methods. In: Brandes, U., Cornelsen, S. (eds.) Proc. of Graph Drawing (GD '10). LNCS, vol. 6502, pp. 80–91 (2011). https://doi.org/https://doi.org/10.1007/978-3-642-18469-7_8
9. Baur, M., Brandes, U.: Crossing reduction in circular layouts. In: Hromkovič, J., Nagl, M., Westfechtel, B. (eds.) Graph-Theoretic Concepts in Computer Science (WG'04). LNCS, vol. 3353, pp. 332–343. Springer Berlin Heidelberg (2004). https://doi.org/10.1007/978-3-540-30559-0_28
10. Biedl, T., Kant, G.: A better heuristic for orthogonal graph drawings. Comput. Geom. Theory Appl. **9**(3), 159–180 (1998). https://doi.org/10.1016/S0925-7721(97)00026-6
11. Biedl, T.C., Madden, B.P., Tollis, I.G.: The three-phase method: A unified approach to orthogonal graph drawing. International Journal of Computational Geometry and Applications **10**(6), 553–580 (2000). https://doi.org/10.1142/S0218195900000310
12. Brandes, U., Pich, C.: Eigensolver methods for progressive multidimensional scaling of large data. In: Proc. of Graph Drawing. LNCS, vol. 4372, pp. 42–53. Springer (2007). https://doi.org/https://doi.org/10.1007/978-3-540-70904-6_6
13. Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing: Algorithms for the Visualization of Graphs. 1st edn. (1998)
14. Dwyer, T., Lee, B., Fisher, D., Quinn, K.I., Isenberg, P., Robertson, G., North, C.: A comparison of user-generated and automatic graph layouts. IEEE Trans. Visualization and Computer Graphics **15**(6), 961–968 (2009). https://doi.org/10.1109/TVCG.2009.109
15. Eades, P.: A heuristic for graph drawing. Congressus Numerantium **42**, 149–160 (1984)
16. Eiglsperger, M., Fekete, S.P., Klau, G.W.: Orthogonal graph drawing. In: Kaufmann, M., Wagner, D. (eds.) Drawing Graphs: Methods and Models, LNCS, vol. 2025, chap. 6, pp. 121–171. Springer-Verlag (2001). https://doi.org/10.1007/3-540-44969-8_6
17. Ellson, J., Gansner, E.R., Koutsofios, E., North, S.C., Woodhull, G.: Graphviz - open source graph drawing tools. In: Proc. of Graph Drawing (GD '01). LNCS, vol. 2265, pp. 483–484 (2002). https://doi.org/https://doi.org/10.1007/3-540-45848-4_57
18. Frishman, Y., Tal, A.: Online dynamic graph drawing. IEEE Trans. on Visualization and Computer Graphics **14**(4), 727–740 (2008)
19. Fruchterman, T.M.J., Reingold, E.M.: Graph drawing by force directed placement. Software - Practice and Experience **21**, 1129–1164 (1991)
20. Fößmeier, U., Kaufmann, M.: Algorithms and area bounds for nonplanar orthogonal drawings. In: Di Battista, G. (ed.) Proc. of Graph Drawing (GD'97). LNCS, vol. 1353, pp. 134–145. Springer (1997). https://doi.org/10.1007/3-540-63938-1_57
21. Gajer, P., Kobourov, S.: GRIP: Graph dRawing with Intelligent Placement. Journal of Graph Algorithms and Applications **6**(3), 203–224 (2002)

22. Gansner, E.R., Hu, Y., Krishnan, S.: COAST: A convex optimization approach to stress-based embedding. In: Graph Drawing (GD '13). LNCS, vol. 8242, pp. 268–279. Springer (2013). https://doi.org/https://doi.org/10.1007/978-3-319-03841-4_24
23. Gansner, E.R., Hu, Y., North, S.: A maxent-stress model for graph layout. IEEE Trans. on Visualization and Computer Graphics **19**(6), 927–940 (2013)
24. Gansner, E.R., Koren, Y.: Improved circular layouts. In: Kaufmann, M., Wagner, D. (eds.) Proc. of Graph Drawing (GD'06). LNCS, vol. 4372, pp. 386–398. Springer Berlin Heidelberg (2006). https://doi.org/10.1007/978-3-540-70904-6_37
25. Gibson, H., Faith, J., Vickers, P.: A survey of two-dimensional graph layout techniques for information visualisation. Information Visualization **12**(3-4), 324–357 (2013). https://doi.org/10.1177/1473871612455749
26. Hadany, R., Harel, D.: A multi-scale algorithm for drawing graphs nicely. Discrete Applied Mathematics **113**(1), 3–21 (2001)
27. van Ham, F., Rogowitz, B.E.: Perceptual organization in user-generated graph layouts. IEEE Trans. Visualization and Computer Graphics **14**(6), 1333–1339 (2008). https://doi.org/10.1109/TVCG.2008.155
28. Harel, D., Koren, Y.: A fast multi-scale method for drawing large graphs. Journal of Graph Algorithms and Applications **6**, 179–202 (2002)
29. Hu, Y.: Efficient, high-quality force-directed graph drawing. Mathematica journal **10**(1), 37–71 (2006)
30. Isenberg, T., Neumann, P., Carpendale, S., Sousa, M.C., Jorge, J.A.: Non-photorealistic rendering in context: An observational study. In: Non-Photorealistic Animation and Rendering (NPAR'06). pp. 115–126. ACM (2006). https://doi.org/10.1145/1124728.1124747
31. Kamada, T., Kawai, S.: An algorithm for drawing general undirected graphs. Information Processing Letters **31**(1), 7 – 15 (1989)
32. Kieffer, S., Dwyer, T., Marriott, K., Wybrow, M.: HOLA: Human-like orthogonal network layout. IEEE Trans. Visualization and Computer Graphics **22**(1), 349–358 (2016). https://doi.org/10.1109/TVCG.2015.2467451
33. Klawitter, J., Mchedlidze, T., Nöllenburg, M.: Experimental evaluation of book drawing algorithms. In: Frati, F., Ma, K.L. (eds.) Graph Drawing and Network Visualization (GD'17). LNCS, vol. 10692, pp. 224–238. Springer (2018). https://doi.org/10.1007/978-3-319-73915-1_19
34. Koren, Y., Carmel, L., Harel, D.: ACE: A fast multiscale eigenvectors computation for drawing huge graphs. In: Proc. of IEEE Symposium on Information Visualization. pp. 137–144. IEEE (2002)
35. Kruskal, J.B., Wish, M.: Multidimensional Scaling. Sage Press (1978)
36. Masuda, S., Nakajima, K., Kashiwabara, T., Fujisawa, T.: Crossing minimization in linear embeddings of graphs. IEEE Trans. Computers **39**(1), 124–127 (1990). https://doi.org/10.1109/12.46286
37. Purchase, H.C., Cohen, R.F., James, M.: Validating graph drawing aesthetics. In: Brandenburg, F.J. (ed.) Proc. of Graph Drawing (GD '95). LNCS, vol. 1027, pp. 435–446. Springer-Verlag (1996). https://doi.org/https://doi.org/10.1007/BFb0021827
38. Purchase, H.: Which aesthetic has the greatest effect on human understanding? In: Di Battista, G. (ed.) Proc. of Graph Drawing (GD '97). LNCS, vol. 1353, pp. 248–261. Springer-Verlag (1998). https://doi.org/https://doi.org/10.1007/3-540-63938-1_67

39. Purchase, H.C., Pilcher, C., Plimmer, B.: Graph drawing aesthetics created by users, not algorithms. IEEE Trans. Visualization and Computer Graphics **18**(1), 81–92 (2012). https://doi.org/10.1109/TVCG.2010.269
40. Seery, J.B.: Designing network diagrams. In: General Conf. Social Graphics. vol. 49, p. 22 (1980)
41. Six, J.M., Tollis, I.G.: Circular drawing algorithms. In: Tamassia, R. (ed.) Handbook of Graph Drawing and Visualization, chap. 9, pp. 285–315. CRC Press (2013)
42. Tamassia, R.: On embedding a graph in the grid with the minimum number of bends. SIAM J Computing **16**(3), 421–444 (1987). https://doi.org/10.1137/0216030
43. Torgerson, W.S.: Multidimensional scaling: I. theory and method. Psychometrika **17**(4), 401–419 (1952)
44. Turing, A.: Computing machinery and intelligence. Mind **59**(236), 433–460 (1950)
45. Wang, Y., Wang, Y., Sun, Y., Zhu, L., Lu, K., Fu, C., Sedlmair, M., Deussen, O., Chen, B.: Revisiting stress majorization as a unified framework for interactive constrained graph visualization. IEEE Trans. on Visualization and Computer Graphics **24**(1), 489–499 (2018). https://doi.org/10.1109/TVCG.2017.2745919
46. Wiese, R., Eiglsperger, M., Kaufmann, M.: yFiles: Visualization and automatic layout of graphs. In: Proc. of Graph Drawing (GD '01). pp. 453–454. LNCS (2002)
47. Zachary, W.W.: An information flow model for conflict and fission in small groups. Journal of Anthropological Research **33**, 452–473 (1977), `https://www.cise.ufl.edu/research/sparse/matrices/Newman/karate.html`

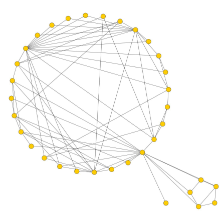
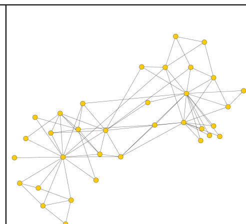
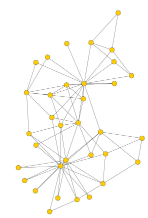
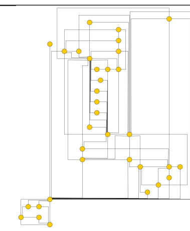
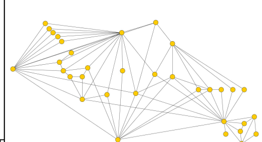
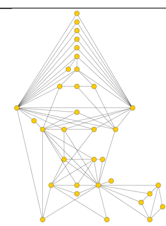
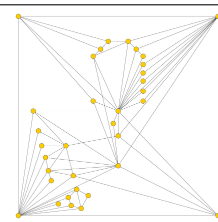
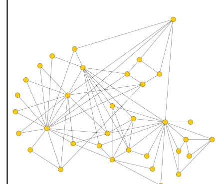
A Acknowledgements

We are grateful to all the experimental participants, to Drew Sheets who assisted with creating the graphs in yEd, and to John Hamer who implemented the online experimental system. Ethical approval was given by the University of Arizona Institutional Review Board (ref: 1712113015).

B Example graph in all eight versions

Graph number 4 (G_4) in the experiment shown below in all eight versions. All the experimental stimuli can be found in the supplementary material included with the submission.

Table 6. Graph number 4 in all eight versions.

			
Circular	Force Directed	Multi-dimensional Scaling	Orthogonal
			
Human Drawer 1	Human Drawer 2	Human Drawer 3	Human Drawer 4

C Demographics

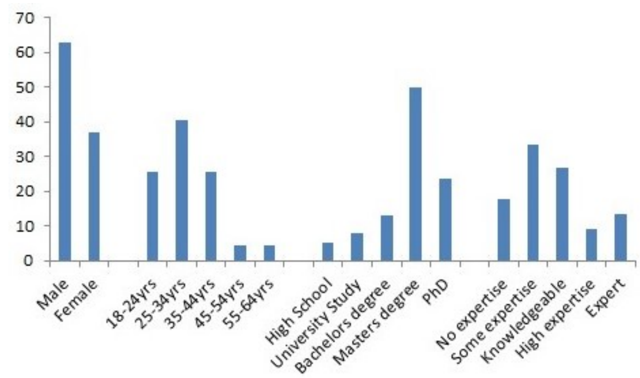


Fig. 4. Distribution of demographic information of our participants in the experiment.

D Time Taken for Human Drawing

The drawers were asked to note the length of time taken to draw each graph; one drawer, D_3, did not note the length of time, but said that drawing all nine graphs took over 24 hours.

Table 7. Length of time taken to draw the graphs, in minutes

	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9
D_1	42	9	27	15	12	5	17	9	12
D_2	74	5	53	37	10	12	23	20	33
D_4	36	50	40	19	18	4	15	12	34
mean	50.7	21.3	40.0	23.7	13.3	7.0	18.3	13.7	26.3