

# Exploring Complex Drawings via Edge Stratification

Emilio Di Giacomo<sup>1</sup>, Walter Didimo<sup>1</sup>, Giuseppe Liotta<sup>1</sup>,  
Fabrizio Montecchiani<sup>1</sup>, and Ioannis G. Tollis<sup>2</sup>

<sup>1</sup> Dip. di Ingegneria Elettronica e dell'Informazione, Università degli Studi di Perugia  
{digiacomo, didimo, liotta, montecchiani}@diei.unipg.it

<sup>2</sup> Univ. of Crete and Institute of Computer Science-FORTH, Greece  
tollis@ics.forth.gr

**Abstract.** We propose an approach that allows a user to explore a layout produced by any graph drawing algorithm, in order to reduce the visual complexity and clarify its presentation. Our approach is based on *stratifying* the drawing into *layers* with desired properties; layers can be explored and combined by the user to gradually acquire details. We present stratification heuristics, a user study, and an experimental analysis that evaluates how our stratification heuristics behave on the drawings computed by a variety of popular force-directed algorithms.

## 1 Introduction

Graph drawing algorithms are used in many applications to visualize networked information. Among them, force-directed algorithms are the most popular and are widely adopted to compute drawings in which vertices are represented as small circles and edges are drawn as straight-line segments. Of course, the chosen algorithm is of great importance in creating a readable visualization. However, when the graph is complex (large or locally dense) a high number of edge crossings is typically unavoidable; this is the case, for example, of most small world and scale-free graphs (see, e.g., [13,30]). It is well known that a high number of edge crossings seriously affects the drawing readability [26,27], and makes it hard to perform detailed tasks based on visual inspection. These tasks include finding the shortest path between two given vertices, finding the vertices that are adjacent to both, or even determining the degree of a vertex.

In this paper we propose a new approach to support the user in the visual inspection of complex drawings. Namely, given a drawing  $\Gamma$  of a graph  $G(V, E)$ , we aim at partitioning the set of edges  $E$  into subsets  $E_1, E_2, \dots, E_h$ , such that the subdrawing  $\Gamma_i \subseteq \Gamma$  of each subgraph  $G_i(V, E_i)$  guarantees some desired readability property (in each subdrawing, the vertices remain fixed in their original positions as determined in  $\Gamma$ ). For example, a user could prefer to see  $\Gamma_i$  without any edge crossing, i.e., as planar, or that any two crossing edges form a sufficiently large angle. We say that  $\Gamma$  is *stratified* into a set of *layers*  $\Gamma_i$ , each containing all the vertices of  $\Gamma$  (in their original positions) but only a portion of the edge set. The user can then interact with this edge stratification, by exploring one layer at a time, or by arbitrarily combining multiple layers into a single view. The edges of each layer are assigned the same color and different colors are used for the edges of different layers. The main advantage of this approach is that

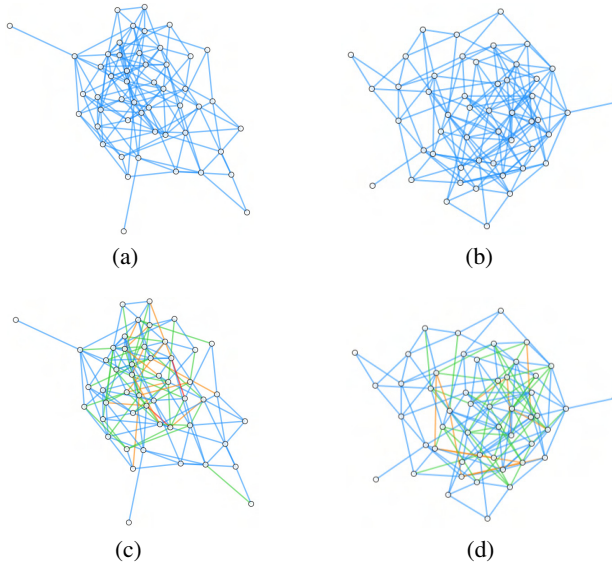
users can get multiple readable views of different portions of the drawing, with the possibility of simplifying the total amount of information, thus allowing them to gradually acquire details by exploring or combining layers. On the negative side, from the cognitive point of view, the user has to face the difficulty of making sense of a distributed information. To deal with this difficulty in practical terms, it is crucial to minimize the number of layers required to stratify the drawing so that the desired readability property is guaranteed for each layer. The main contribution of this paper is as follows:

- (i) We define an edge stratification model and the related optimization problems. Then, we give a general framework to solve these problems for several desired readability properties of the layers, and we describe heuristics within this framework (Section 3).
- (ii) We present the results of a user study aimed at understanding the effectiveness of the proposed approach for executing tasks based on visual inspection (Section 4). These results highlight the usefulness of edge stratification, especially for some of these tasks and for some specific readability properties of the layers.
- (iii) We present an experimental analysis that compares the number of layers required to stratify drawings computed by a variety of popular force-directed algorithms, using our heuristics (Section 5). On one side, these experiments suggest that for some of the computed drawings the number of layers required by some edge stratification is a more reliable measure of the drawing visual complexity with respect to the number of edge crossings. On the other side, the results show that most of the force-directed algorithms that we have considered guarantee a strong correlation between number of crossings and number of layers in the stratifications of their drawings. We interpret this behavior as a positive feature of the drawing algorithms, which witnesses a quite uniform distribution of the crossings in the drawing.

In Fig. 1 we give an example of how the number of layers produced by the stratification heuristics in this paper can be used to measure the readability of different drawings of a same graph. Fig. 1(a) shows a drawing  $T_1$  of a graph with 50 vertices and 150 edges, computed by Fruchterman-Reingold's algorithm [17] and containing 1,395 edge crossings; Fig. 1(b) shows a drawing  $T_2$  of the same graph, computed by Kamada-Kawai's algorithm [21] and having 1,437 edge crossings. Applying our stratification heuristic to compute planar layers,  $T_1$  requires 8 layers while drawing  $T_2$  requires 7 layers; even if we require crossing angles of at least  $\frac{\pi}{4}$  in each layer, our stratification heuristic generates 4 layers for  $T_1$  (Fig. 1(c)) and only 3 layers for  $T_2$  (Fig. 1(d)). Indeed, despite its higher number of crossings, drawing  $T_2$  appears more readable, due to a more uniform distribution of the crossings and a better area. Namely, in the figure the two drawings are scaled to fit in the same bounding box; if we scale the drawings such that they satisfy the same resolution rule, drawing  $T_1$  has twice the area of drawing  $T_2$ .

## 2 Related Work

To reduce the negative impact of edge crossings, different constraints on the type of crossings have been studied. Some of them require that edges cross only at large angles, or that each edge is crossed at most a limited number of times, or even that only few pairwise crossing edges are allowed. A very limited list of papers includes [10,11,12]. Unfortunately, only restricted sub-families of sparse graphs admit drawings that respect

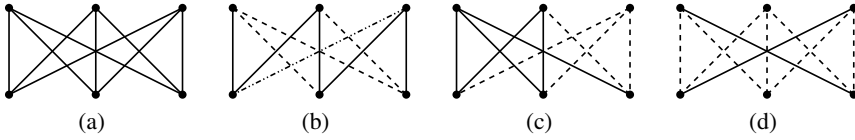


**Fig. 1.** (a-b) Drawings of the same graph computed by two different force-directed algorithms. (c-d) The same drawings in (a-b) stratified with layers having crossing angles of at least  $\frac{\pi}{4}$ ; layers are conveyed with different edge colors.

these constraints. Also, an impressive set of crossing minimization methods are proposed in graph drawing (see [8] for a survey). However, these methods become computationally expensive (or even unfeasible) for large and dense graphs.

Many visualization techniques that compute a *hierarchical clustering* of the vertices and that allow users to interactively explore it are also known (see , e.g., [1,3,30]). The levels of a cluster hierarchy are a sort of vertex stratification, which allows users to control the amount of information displayed in the same view. Other well-studied approaches that facilitate in the visual exploration of networked data are based on node or edge *filtering*, *grouping*, and *motif simplification* (see, e.g., [1,29]).

Our edge stratification approach does not aim at computing drawings of graphs with controlled visual complexity, but rather it starts from a drawing of a graph and aims at supporting its exploration and analysis by distributing the whole drawing information into a set of logical layers with desired edge crossings properties. This idea is somewhat related to the notion of *geometric thickness* of a graph  $G$  [14,15], which is the minimum number of colors that can be assigned to the edges of  $G$ , such that there exists a straight-line drawing of  $G$  where no two edges of the same color cross. Similar to thickness, our stratification defines an edge coloring; the difference is that stratification is executed on a specific drawing, which cannot be changed. Hence, if the geometric thickness can be used as a measure of the graph complexity, the minimum size of a stratification of a straight-line drawing can be used as a measure of the drawing visual complexity in terms of number, types, and distribution of its edge crossings. Clearly, the size of a stratification into planar layers of a drawing  $\Gamma$  of  $G$  cannot be smaller than the geometric thickness of  $G$ .



**Fig. 2.** (a) A straight-line drawing  $\Gamma$  of  $K_{3,3}$ . (b) A stratification  $S(\Gamma, \text{PLANARITY})$ . (c) A stratification  $S(\Gamma, \text{LAC}(\frac{\pi}{3}))$ . (d) A stratification  $S(\Gamma, \text{1-PLANARITY})$ . In each stratification different dash styles for the edges represent different layers.

Other approaches have been proposed for supporting the analysis of a given a drawing. One of the most popular is *edge bundling*, which deforms and groups together edges that are similar according to some metric (see [32] for a survey). Another approach, called *geometric graph generalization*, reduces vertex and/or edge clutter by collapsing groups of vertices that are geometrically close to one another into a single point [7]. Differently from our stratification, edge bundling and geometric graph generalization modify the input drawing, emphasizing its skeletal structure at the expenses of loss of details. We finally mention another recent technique, which aims to visually simplify or remove edge crossings in a given drawing by displaying only portions of the crossing edges; the modified drawings are called *partial edge drawings* [6].

### 3 Stratification: Model and Algorithms

Here we formally describe our edge stratification model and related algorithms.

**Model.** Let  $G(V, E)$  be a graph and let  $\Gamma$  be a straight-line drawing of  $G$ . Given a subset  $E' \subseteq E$ ,  $G[E']$  denotes the subgraph  $G'(V, E')$  of  $G$ , and  $\Gamma[E']$  is the subdrawing of  $G[E']$  in  $\Gamma$ . Also, let  $\mathcal{P}$  denote a desired geometric property of a drawing of a graph. An *edge stratification* (or simply a *stratification*) of  $\Gamma$  with respect to  $\mathcal{P}$ , also denoted as  $S(\Gamma, \mathcal{P})$ , is a partition of the edges of  $G$  into  $h$  subsets  $E_1, E_2, \dots, E_h$  such that, for every  $i \in \{1, \dots, h\}$ , property  $\mathcal{P}$  holds for  $\Gamma[E_i]$ . Each subdrawing  $\Gamma[E_i]$  is called a *layer* of  $S(\Gamma, \mathcal{P})$ , and the *size* of  $S(\Gamma, \mathcal{P})$  is the number  $h$  of its layers. We study the following general optimization problem.

*Problem 1.* – **MINGENERALSTRATIFICATION:** Given a straight-line drawing  $\Gamma$  and a geometric property  $\mathcal{P}$ , find a stratification  $S(\Gamma, \mathcal{P})$  of minimum size.

In particular, we focus on the following geometric properties  $\mathcal{P}$ : (i) **PLANARITY:** the drawing is crossing free; (ii) **LAC( $\alpha$ ):** any two crossing edges of the drawing form an angle of at least  $\alpha$  radians (LAC stands for *large angle crossing*); (iii)  **$k$ -PLANARITY:** each edge of the drawing is crossed at most by  $k$  edges ( $k \geq 1$ ). Each property gives rise to a specialized version of Problem 1 (see Fig. 2 for an example of the different types of stratification):

*Problem 2.* – **MINPLANARSTRATIFICATION:** Given a straight-line drawing  $\Gamma$ , find a stratification  $S(\Gamma, \text{PLANARITY})$  of minimum size.

**Problem 3.** – MINLACSTRATIFICATION: Given a straight-line drawing  $\Gamma$  and a constant  $\alpha \in (0, \frac{\pi}{2}]$ , find a stratification  $S(\Gamma, \text{LAC}(\alpha))$  of minimum size.

**Problem 4.** – MIN $k$ -PLANARSTRATIFICATION: Given a straight-line drawing  $\Gamma$  and a constant  $k > 0$ , find a stratification  $S(\Gamma, k\text{-PLANARITY})$  of minimum size.

It is natural to ask what is the complexity of the stratification problems defined above. We prove that they are difficult at least as the well-known problem called *classification*, restricted to planar graphs with maximum vertex degree 4 or 5; this problem is conjectured to be NP-hard [9]. Namely, an *edge coloring* of a graph  $G(V, E)$  is an assignment of edge colors such that adjacent edges have different colors. The minimum number of colors of an edge coloring of  $G$  is called the *chromatic index* of  $G$ . It is known that the chromatic index of a graph is either  $\Delta(G)$  or  $\Delta(G) + 1$ , where  $\Delta(G)$  is the maximum vertex degree of  $G$  [31]. The classification problem is the problem of deciding whether a graph  $G$  has chromatic index  $\Delta(G)$  or  $\Delta(G) + 1$ , and it is NP-complete in general [20]. Restricting the input graph to be planar, the classification problem can be reduced to our stratification problems (we omit details due to space limitations).

**Algorithms.** Let  $\Gamma$  be a drawing of  $G(V, E)$ . To solve our different stratification problems on  $\Gamma$  we provide heuristics based on a common unified framework. It exploits an enhanced version of the *crossing graph* of  $\Gamma$ , which is a graph  $\chi_\Gamma(V_\chi, E_\chi)$  having a vertex for each edge of  $\Gamma$ , i.e.,  $V_\chi = E$ , and an edge for each pair of crossing edges of  $\Gamma$ , i.e.,  $E_\chi = \{(e_1, e_2) | e_1, e_2 \in E \text{ and } e_1 \text{ and } e_2 \text{ cross in } \Gamma\}$ . In our enhanced version of  $\chi_\Gamma$ , we add a weight to each edge  $(e_1, e_2) \in E_\chi$ , equal to the minimum angle formed by  $e_1$  and  $e_2$  at their crossing point in  $\Gamma$ . Given the one-to-one correspondence between the edges of  $\Gamma$  and the vertices of  $\chi_\Gamma$ , an edge stratification  $S(\Gamma, \mathcal{P}) = \{E_1, \dots, E_h\}$  corresponds to coloring the vertices of  $\chi_\Gamma$  such that the subgraph induced by all vertices with the same color satisfies a property  $\mathcal{P}'$  that is the “translation” of  $\mathcal{P}$  to the crossing graph. Namely, if  $V_\chi^i \subseteq V_\chi$  is the color class associated with  $E_i$  ( $1 \leq i \leq h$ ), we have that: (i)  $\mathcal{P} = \text{PLANARITY}$  translates into  $\mathcal{P}' = \text{INDEPENDENTSET}$ : the vertices of  $V_\chi^i$  form an independent set in  $\chi_\Gamma$ ; (ii)  $\mathcal{P} = k\text{-PLANARITY}$  translates into  $\mathcal{P}' = \text{MAXDEGREE-}k$ : the subgraph of  $\chi_\Gamma$  induced by  $V_\chi^i$  has vertex degree at most  $k$ ; (iii)  $\mathcal{P} = \text{LAC}(\alpha)$  translates into  $\mathcal{P}' = \text{EDGEWEIGHT}(\alpha)$ : the subgraph of  $\chi_\Gamma$  induced by  $V_\chi^i$  has no edge weight less than  $\alpha$ .

Hence, computing a stratification  $S(\Gamma, \mathcal{P}) = \{E_1, \dots, E_h\}$  is equivalent to computing a coloring  $C(\chi_\Gamma, \mathcal{P}') = \{V_\chi^1, \dots, V_\chi^h\}$  of the vertices of  $\chi$ , such that the subgraph induced by each  $V_\chi^i$  satisfies property  $\mathcal{P}'$ . In particular, Problem MINPLANARSTRATIFICATION equals to the classical *minimum vertex coloring* problem on  $\chi_\Gamma$ , which consists of coloring the vertices of  $\chi_\Gamma$  with the minimum number of colors, such that no vertices with the same color are adjacent. Problems MINLACSTRATIFICATION can be reduced to a minimum vertex coloring problem on  $\chi_\Gamma$  by applying a pre-processing step that removes from  $\chi_\Gamma$  all the edges whose weight is at least  $\alpha$ . Problem MIN $k$ -PLANARSTRATIFICATION corresponds to a generalization of the minimum vertex coloring on  $\chi_\Gamma$ , which allows each vertex to have at most  $k$  adjacent vertices of its same color. Given this strong correlation among all problems, we solve them with a unified framework that is an adaptation of a heuristic for the minimum vertex coloring problem, called *sequential coloring* [5]. It has been shown to be more effective with respect to

other heuristics for the minimum vertex coloring, and can be easily adapted to all our variants of this problem. Our unified heuristic framework works as follows.

Let  $\mathcal{P}'$  be the desired property for the subgraph induced by each color class. The vertices  $\{v_1, \dots, v_{|E|}\}$  of  $\chi_\Gamma$  are processed one per time; the first vertex is assigned to color class  $V_\chi^1$ . If vertices  $v_1, v_2, \dots, v_{i-1}$  have been assigned to the color classes  $V_\chi^1, V_\chi^2, \dots, V_\chi^k$ , the next vertex  $v_i$  is assigned to the color class  $V_\chi^j$ , where  $j$  is the minimum value for which  $V_\chi^j \cup \{v_i\}$  satisfies property  $\mathcal{P}'$ ; if no such  $j$  exists,  $v_i$  is assigned to a new color class  $V_\chi^{k+1}$ . Several different criteria can be used to choose the next vertex  $v_i$  to be processed. We choose vertex  $v_i$  with the highest *degree of saturation*, which is the number of different colors assigned to the neighbors of  $v_i$ . This strategy has been experimentally proven to give good performance in terms of number of colors used [5]. The time complexity of our heuristic can be evaluated as follows. By using a brute-force approach the crossing graph can be computed in  $O(m^2)$ , where  $m$  is the number of edges of  $\Gamma$ . Using the degree of saturation as the criterium for the vertex selection, the time complexity of the sequential coloring heuristic is  $O(N^2 \log N)$ , where  $N$  is the number of vertices of the graph to be colored. Since in our case  $N = m$ , the overall time complexity of our heuristic is  $O(m^2 \log n)$ .

## 4 User Study

To evaluate the usefulness of our approach, we performed a user study where different interfaces based on edge stratification are compared with an interface where drawings are not stratified. The stratifications were computed with the heuristic framework described in Section 3 and the geometric properties considered were PLANARITY and  $\text{LAC}(\frac{\pi}{4})$ . In particular, we chose  $\frac{\pi}{4}$  as minimum crossing angular resolution, because we observed that this value gives rise to limited number of layers without affecting too much the readability of each layer (see also [12]). Also, in the experiment we decided not to evaluate stratifications obtained for  $k$ -PLANARITY for two reasons: (i) comparing too many interfaces would have taken to the users a very long time to complete their test; (ii) there were not significant differences between the sizes of the stratifications  $S(\Gamma, \text{PLANARITY})$  and  $S(\Gamma, k\text{-PLANARITY})$  (considering small values of  $k$ ) for the drawings  $\Gamma$  of our benchmark, thus there was no clear advantage in using  $k$ -PLANARITY with respect to PLANARITY from the practical point of view. Clearly, this last observation motivates the study of more effective heuristics to compute a stratification  $S(G, k\text{-PLANARITY})$ , when  $k > 0$ . Alternatively, we could consider large values of  $k$ , however, this would strongly reduce the readability of the layers.

The goal of our study was to address the following two research questions: **(Q1)**. Given a straight-line drawing of a graph, does stratification assist in the reading of the relational information represented by the graph? **(Q2)**. If the first question is settled in the affirmative, is one of the two considered geometric properties (PLANARITY and  $\text{LAC}(\frac{\pi}{4})$ ) more effective in assisting the reading of such relational information?

We performed a within-subjects experiment involving 40 participants. We used 5 different drawings; for each drawing, the participants had to solve 3 different tasks, using 3 different user interfaces. Thus, a trial was represented by the triple (drawing, task, interface) and the number of trials for each participant was  $5 \times 3 \times 3 = 45$ .

**Table 1.** The graphs of the user study

<i>graph</i>	<i>vertices</i>	<i>edges</i>	<i>density</i>	<i>stratification layers</i>	
				PLANARITY	LAC( $\frac{\pi}{4}$ )
lesmis	77	2,148	27.9	4	3
football	115	613	5.33	7	4
gd01	249	635	2.55	6	4
organization	165	726	4.4	7	4
scalefree	204	803	3.94	9	7

**Drawings.** We chose 5 different complex graphs modeling both real and artificial networks of different type, and we drew them using the OGDF<sup>1</sup> implementation of the multi-level force-directed algorithm FMMM [19]. The chosen graphs are: *lesmis*, a coappearance graph of characters in the novel “Les Miserables” [22]; *football*, a graph of American football games [18]; *gd01*, a graph drawing self-reference network used in the GD 2001 contest [25]; *organization*, a social network modeling the relationships among employees in a private company [24]; *scalefree*, a scale-free network generated using the Barabási-Albert model [2]. For each graph, Table 1 reports the number of vertices, the number of edges, the density (ratio between number of edges and number of vertices), and the number of stratification layers for properties PLANARITY and LAC( $\frac{\pi}{4}$ ).

**Tasks.** We chose 3 tasks as representative of the possible tasks involving graph reading. We aimed at having tasks significantly different one to another, easy to understand by non-expert users, and requiring both local and global explorations of the drawing. We considered: the *Shortest Path (SP)* task, which asks “How long is the shortest path between the two highlighted vertices?”; the *Common Adjacent (CO)* task, which asks “What is the number of adjacent vertices shared by the two highlighted vertices?”; and the *Degree (DE)* task, which asks “What is the degree of the highlighted vertex?”. Similar tasks are used in other experiments on graph reading (see, e.g., [28]). The vertices highlighted for each task were chosen without looking at the stratification layers.

**User interfaces.** The three user interfaces differ from one another only by the geometric property used to compute a stratification. The *Planar Stratification (PS)* interface uses PLANARITY; the *LAC Stratification (LS)* interface uses LAC( $\frac{\pi}{4}$ ); the *Overview (OV)* interface does not use any stratification (there is only one layer). To limit the number of layers in PS and LS, we halted the edge partitioning process when 80% of the edges of the drawing were placed in some layer. The remaining edges were all put in an additional layer, for which no geometric property is guaranteed. In every interface the edges of each layer were displayed with the same color and we used a different color for the edges of each layer. We chose the colors so to maximize the readability over a black background and so that different colors can be easily distinguished. Every interface allowed the user to select any combination of the available layers, so that only the edges in the selected layers were displayed on the screen; the edges in the non-selected

<sup>1</sup> <http://www.ogdf.net>

layers were sketched as transparent light gray segments ( $\alpha=0.2$ ). In this way, the user could focus on the selected layers only, still keeping in mind that the remaining edges were part of the drawing and should possibly be considered to properly solve the task. Thus, the user's strategy to explore the drawing varied from looking at one single layer per time to looking at the drawing as a whole. Also, the vertices were always drawn as white circles except the highlighted vertices of each trial, which were drawn red and slightly larger. Such a consistent environment did not require to the user any cognitive shift to move from one interface to another. Finally, the participants were not aware of the criteria used to stratify the drawings.

**Experimental Procedure.** Before starting the experiments, the participants received a brief tutorial introducing the basic concepts on graphs. Also, an explanation of the tasks and of the user interfaces was given with practical examples. The 45 trials were preceded by 4 training trials whose results were not taken into account, although the participants were not aware of it. Regardless of the task, participants had to answer each question by entering a number in a text box, with no time limit. However, they were asked to spend at most 3 minutes for each question. In order to counter the learning effect, in each experiment the 45 stimuli appeared in a randomized order, and the system randomly flipped and rotated each drawing. Between a question and the next one, participants could take a short break, without the possibility of exchanging information.

**Results.** 40 volunteering students (with age from 19 to 25) in Computer Engineering took part in the experiments. We recorded their answers and the time spent for each question. We compared the performance of OV, PS, and LS in terms of absolute error rate (the absolute value of the difference between the user answer and the correct answer) and response time. First of all, we performed a Shapiro-Wilk test ( $\alpha=0.05$ ) to determine whether the data was normally distributed or not. We found that none of the considered populations was normally distributed. Thus, we performed a non-parametric analysis exploiting repeated measures Friedman tests ( $\alpha=0.05$ ), with post-hoc pairwise comparisons. We applied Bonferroni corrections on the pairwise comparisons setting  $\alpha < 0.017$ . We obtained these results (see Table 2). (i) Considering all the tasks, both PS and LS significantly outperformed OV in terms of absolute error rate. This improvement in the accuracy comes together with a slower response time: PS and LS show slower performance than OV, and LS is faster than PS. (ii) For task SP, the mean absolute error rate of LS is smaller than the one of OV and PS, although such a difference turns out to be not statistically significant. In terms of response time, again, OV is faster than PS and LS, while LS is faster than PS. (iii) For task CA, the three interfaces led to comparable performance in terms of absolute error rate. In terms of response time, the situation is similar to the previous cases, OV is faster than PS and LS, while LS is faster than PS. (iv) For task DE, both PS and LS outperformed OV in terms of absolute error rate, and the difference turns out to be statistically significant. Also, PS led to more accurate results than LS, still with statistical significance. About the response time, OV and LS behave similarly and slightly faster than PS.

**Discussion.** The results of the user study show an improvement in terms of accuracy in the reading of the displayed graphs when using stratification. The PS interface outperformed both LS and OV for most of the considered tasks. We conclude that stratifying



**Table 2.** Results of the user study. The mean values and the pairwise significance between each user interface are shown for absolute error rate and time

		Overall	SP	CA	DE			Overall	SP	CA	DE
Abs. Error Rate	mean OV	2.74	1.84	1.93	4.43	Time (sec.)	mean OV	54.36	60.47	61.23	41.36
	mean PS	1.87	1.79	1.96	1.86		mean PS	80.01	91.78	98.26	50.01
	mean LS	1.92	1.57	1.98	2.22		mean LS	68.73	77.46	85.71	43.00
	OV vs PS	< .001	n.s.	n.s.	< .001		OV vs PS	< .001	< .001	< .001	.001
	OV vs LS	< .001	n.s.	n.s.	< .001		OV vs LS	< .001	< .001	< .001	n.s.
	PS vs LS	n.s.	n.s.	n.s.	.010		PS vs LS	< .001	.002	.001	< .001

the drawing into planar layers gave a significant help to the participants. The task that received more significant advantage from both PS and LS is task DE. Indeed, counting the degree of a vertex can be quite hard when the drawing is cluttered around the highlighted vertex; on the other hand, by selecting a layer per time, one can effectively cope with such clutter and the partial degree of the vertex can be easily counted, at the expenses of a negligible increase of the response time (less than 10 seconds in the average). Moreover, according to this strategy, having planar layers guarantees that the region of the drawing around the highlighted vertex is crossing free, and hence clearer.

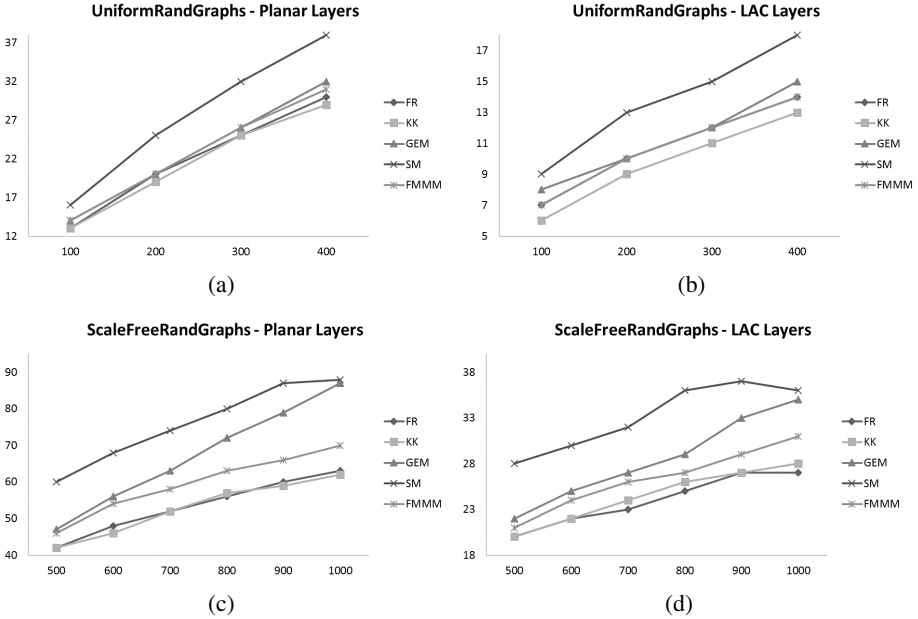
## 5 Comparison of Graph Drawing Algorithms

We describe a second experiment, which compares several force-directed algorithms to answer the following research questions: **(Q3)**. What is the force-directed drawing algorithm for which the computed layouts require the smallest number of layers with properties PLANARITY and  $LAC(\frac{\pi}{4})$ , when stratified with our heuristics? **(Q4)**. How are the number of crossings and the number of layers correlated?

We tested 5 different force-directed algorithms on a benchmark of 105 graphs, thus collecting 525 drawings; for each drawing  $\Gamma$ , we measured its number of crossings, the size of a stratification  $S(\Gamma, \text{PLANARITY})$ , and the size of a stratification  $S(\Gamma, LAC(\frac{\pi}{4}))$  computed by our heuristics. Observe that, since the size of the crossing graph can be  $\Omega(n^4)$ , coloring this graph exactly is often prohibitive even for small graphs.

**Algorithms.** We tested the whole set of force-directed algorithms available in the OGDF library: Kamada-Kawai (KK) [21], Fruchterman-Reingold (FR) [17], GEM (GEM) [16], FM<sup>3</sup> (FMMM) [19], and Stress majorization (SM) [4]. Since tuning is a critical issue for force-directed techniques, it is worth remarking that we initialized the algorithms by using the default parameters set by their implementations in the library.

**Benchmark.** We ran the drawing algorithms on a benchmark of 105 complex graphs, organized in three groups: `UniformRandGraphs`, containing 40 random graphs generated with a uniform probability distribution; for each  $n \in \{100, 200, \dots, 400\}$  we generated 10 graphs with density in the interval  $[2, 6]$ . `ScaleFreeRandGraphs`, containing 60 small-world and scale-free graphs generated with the LFR algorithm [23], already used to generate graphs in previous extensive experimental works [13]; for each



**Fig. 3.** Mean number of planar and LAC( $\frac{\pi}{4}$ ) layers on UniformRandGraphs (a-b) and ScaleFreeRandGraphs (c-d). The  $x$ -axis reports the number of vertices.

$n \in \{500, 600, \dots, 1000\}$  we generated 10 graphs with density in the interval  $[4, 8]$ . UserStudyGraphs, which contains the 5 graphs used in the user study.

**Results.** (i) On UniformRandGraphs, the layouts of FR, GEM, and FMMM require a comparable number of layers regardless the geometric property. KK outperforms the other algorithms for LAC( $\frac{\pi}{4}$ ) layers, while SM always led to the highest number of layers. See Figs. 3(a) and 3(b). (ii) On ScaleFreeRandGraphs, the layouts of FR and KK most frequently require the smallest number of layers, regardless the geometric property; FMMM led to slightly bigger numbers, while again SM led to the largest number of layers. Finally, GEM behaves similarly to FMMM for small values of  $n$ , and approaches SM as  $n$  grows. See Figs. 3(c) and 3(d). (iii) On UserStudyGraphs, the layouts computed by FR are those that more often require the smallest number of layers. KK and FMMM behave similarly, GEM is slightly worse, and SM most frequently led to the largest number of layers. We omit the charts due to space limitations.

To evaluate the correlation between layers and crossings, we executed a Kendall's  $\tau$  test ( $\alpha < 0.01$ ). The test showed a strong correlation ( $0.9 < r < 0.96$ ) between number of crossings and number of layers (both planar and LAC( $\frac{\pi}{4}$ )), for all the algorithms.

**Discussion.** Concerning question Q3, the results show that FR and KK have the best performance in terms of number of layers required to stratify their layouts (both planar and LAC( $\frac{\pi}{4}$ )), while SM is always the worst for the graphs in our benchmark. About question Q4, we interpret the strong correlation between the number of crossings and

the number of layers as a positive feature of the drawing algorithms, which witnesses a quite uniform distribution of the crossings in the drawing. We also observe that, for some instances, there are pairs of drawings  $\langle \Gamma_1, \Gamma_2 \rangle$  computed by different algorithms such that  $\Gamma_1$  has more crossings but less layers than  $\Gamma_2$ . By visually inspecting these instances, the crossings in the drawings with fewer layers appear to be more evenly distributed, resulting in a more readable drawing in spite of the greater number of crossings. This seems to confirm our intuition that the number of layers is also related to the distribution of the crossings in the drawing area, and suggests that the number of layers could be a more reliable measure of the drawing visual complexity with respect to the number of crossings. Note that just measuring the crossing spatial distribution in a drawing does not necessarily yield to similar conclusions; indeed, a drawing with low average crossing spatial distribution may still contain very cluttered spots surrounded by crossing-free regions, which might cause a high number of layers.

## 6 Conclusions and Future Research Directions

Our framework, based on the use of the crossing graph, is not suited for very large graphs. The number of crossings can be  $\Omega(n^4)$  and hence, even for relatively small complex drawings, the crossing graph can be so large that the performance of our heuristic degrades both in terms of space and time. It would be useful to devise more efficient heuristics that do not make use of crossing graphs. Also, in our user interface we presented the different layers using colors and allowing users to select any subset of layers. It would be interesting to design new visualization paradigms to effectively present stratified drawings. A 2.5D visualization could be an interesting option to explore.

## References

1. Auber, D., Chiricota, Y., Jourdan, F., Melançon, G.: Multiscale visualization of small world networks. In: *InfoVis 2003*, pp. 75–81. IEEE (2003)
2. Barabasi, A.-L., Albert, R.: Emergence of Scaling in Random Networks. *Science* 286(5439), 509–512 (1999)
3. Batagelj, V., Brandenburg, F., Didimo, W., Liotta, G., Palladino, P., Patrignani, M.: Visual analysis of large graphs using (X,Y)-clustering and hybrid visualizations. *IEEE TVCG* 17(11), 1587–1598 (2011)
4. Brandes, U., Pich, C.: More flexible radial layout. In: Eppstein, D., Gansner, E.R. (eds.) *GD 2009*. LNCS, vol. 5849, pp. 107–118. Springer, Heidelberg (2010)
5. Brélez, D.: New methods to color the vertices of a graph. *Comm. ACM* 22, 251–256 (1979)
6. Bruckdorfer, T., Cornelsen, S., Gutwenger, C., Kaufmann, M., Montecchiani, F., Nöllenburg, M., Wolff, A.: Progress on partial edge drawings. In: Didimo, W., Patrignani, M. (eds.) *GD 2012*. LNCS, vol. 7704, pp. 67–78. Springer, Heidelberg (2013)
7. Brunel, E., Gamsa, A., Krug, M., Rutter, I., Wagner, D.: Generalizing geometric graphs. In: Speckmann, B. (ed.) *GD 2011*. LNCS, vol. 7034, pp. 179–190. Springer, Heidelberg (2011)
8. Buchheim, C., Chimani, M., Gutwenger, C., Jünger, M., Mutzel, P.: Crossings and planarization. In: Tamassia, R. (ed.) *Handbook of Graph Drawing and Visualization*. CRC Press (2013)
9. Chrobak, M., Nishizeki, T.: Improved edge-coloring algorithms for planar graphs. *J. Algo.* 11(1), 102–116 (1990)

10. Di Giacomo, E., Didimo, W., Liotta, G., Montecchiani, F.: *h*-quasi planar drawings of bounded treewidth graphs in linear area. In: Golubic, M.C., Stern, M., Levy, A., Morgenstern, G. (eds.) WG 2012. LNCS, vol. 7551, pp. 91–102. Springer, Heidelberg (2012)
11. Di Giacomo, E., Didimo, W., Liotta, G., Montecchiani, F.: Area requirement of graph drawings with few crossings per edge. *Comp. Geom.* 46(8), 909–916 (2013)
12. Didimo, W., Liotta, G.: The crossing angle resolution in graph drawing. In: Pach, J. (ed.) *Thirty Essays on Geometric Graph Theory*. Springer (2012)
13. Didimo, W., Montecchiani, F.: Fast layout computation of hierarchically clustered networks: Algorithmic advances and experimental analysis. In: IV 2012, pp. 18–23 (2012)
14. Dillencourt, M.B., Eppstein, D., Hirschberg, D.S.: Geometric thickness of complete graphs. *Jour. Graph. Alg. and Appl.* 4(3), 5–17 (2000)
15. Duncan, C.A., Eppstein, D., Kobourov, S.G.: The geometric thickness of low degree graphs. In: SoCG 2004, pp. 340–346. ACM (2004)
16. Frick, A., Ludwig, A., Mehldau, H.: A fast adaptive layout algorithm for undirected graphs. In: Tamassia, R., Tollis, I.G. (eds.) GD 1994. LNCS, vol. 894, pp. 388–403. Springer, Heidelberg (1995)
17. Fruchterman, T.M.J., Reingold, E.M.: Graph drawing by force-directed placement. *Softw. Pract. Exper.* 21(11), 1129–1164 (1991)
18. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. *PNAS* 99(12), 7821–7826 (2002)
19. Hachul, S., Jünger, M.: Drawing large graphs with a potential-field-based multilevel algorithm. In: Pach, J. (ed.) GD 2004. LNCS, vol. 3383, pp. 285–295. Springer, Heidelberg (2005)
20. Holyer, I.: The NP-completeness of edge-coloring. *SIAM J. Comput.* 10(4), 718–720 (1981)
21. Kamada, T., Kawai, S.: An algorithm for drawing general undirected graphs. *Inf. Process. Lett.* 31(1), 7–15 (1989)
22. Knuth, D.E.: *The Stanford Graphbase: A Platform for Combinatorial Computing*. Addison-Wesley Professional (1993)
23. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* 78(4) (2008)
24. Michael Fire, Y.E., Puzis, R.: Organization mining using online social networks (2012), <http://proj.ise.bgu.ac.il/sns>
25. Mutzel, P., Jünger, M., Leipert, S. (eds.): GD 2001. LNCS, vol. 2265. Springer, Heidelberg (2002)
26. Purchase, H.C.: Effective information visualisation: a study of graph drawing aesthetics and algorithms. *Interact. Comput.* 13(2), 147–162 (2000)
27. Purchase, H.C., Carrington, D.A., Alder, J.-A.: Empirical evaluation of aesthetics-based graph layout. *Empir. Softw. Eng.* 7(3), 233–255 (2002)
28. Purchase, H.C., Hamer, J., Nöllenburg, M., Kobourov, S.G.: On the usability of lombardi graph drawings. In: Didimo, W., Patrignani, M. (eds.) GD 2012. LNCS, vol. 7704, pp. 451–462. Springer, Heidelberg (2013)
29. Shneiderman, B., Dunne, C.: Interactive network exploration to derive insights: Filtering, clustering, grouping, and simplification. In: Didimo, W., Patrignani, M. (eds.) GD 2012. LNCS, vol. 7704, pp. 2–18. Springer, Heidelberg (2013)
30. van Ham, F., van Wijk, J.J.: Interactive visualization of small world graphs. In: *InfoVis 2004*, pp. 199–206. IEEE (2004)
31. Vizing, V.G.: On an estimate of the chromatic class of a  $p$ -graph. *Diskret. Analiz* No. 3, 25–30 (1964)
32. Zhou, H., Xu, P., Yuan, X., Qu, H.: Edge bundling in information visualization. *Tsinghua Science and Technology* 18(2), 145–156 (2013)