



# Trade-Off Between Mental Map and Aesthetic Criteria in Simulated Annealing Based Graph Layout Algorithms

Armin Jörg Slopek<sup>1(✉)</sup>, Carsten Winkelholz<sup>1</sup>, and Margaret Varga<sup>2,3</sup>

<sup>1</sup> Department of Ergonomics and Human-Systems Engineering,  
Fraunhofer Institute for Communication, Information Processing and Ergonomics,  
Fraunhoferstr. 20, 53343 Wachtberg, Germany

[armin.slopek@fkie.fraunhofer.de](mailto:armin.slopek@fkie.fraunhofer.de)

<sup>2</sup> Seetru Ltd., Albion Dockside Works, Bristol BS1 6UT, UK

<sup>3</sup> Department of Zoology, University of Oxford, Oxford OX1 3SY, UK

**Abstract.** Dynamic graph visualization is a key component of interactive graph visualization systems. Whenever a user applies filters or a graph is modified by other reasons, a new visualization of the modified graph should support the user's Mental Map of the previous visualization to facilitate fast reorientation in the new drawing. There exist specialized graph layout algorithms which adopt the concept of Mental Map preservation to create recognizable layouts for similar graphs. In this work we used Simulated Annealing algorithms to calculate layouts which fulfill aesthetic and Mental Map requirements simultaneously. We investigated criteria of both types and conducted an experiment to examine the competition and trade-off between aesthetics and mental map preservation. Our findings show that even without explicitly optimizing Mental Map criteria, recognition can be supported by simply using the previous layout as a starting point, rather than a new layout with randomly allocated vertices. This results in better aesthetic quality as well as lower algorithm runtime. Another finding is that a simple weighted sum between aesthetic and the Mental Map may not be as effective as one might expect, especially if the weight assigned to the Mental Map is higher than the weight for aesthetics. Finally, we propose approaches for changing other aspects of the Simulated Annealing algorithm to obtain better graph layouts.

## 1 Introduction

The visualization of data is an important aspect of today's business and research applications. Datasets can typically be viewed as an information network – for example, when entries within a relational database reference other entries, the whole database can be represented as a graph of vertices and edges.

Graph layout algorithms provide the means to visualize such networks or graphs by assigning positions to vertices, mostly in two or three dimensional

space. Some algorithms may also assign coordinates to edge intersections or a specific trajectory for each edge, however such algorithms are outside the scope of this work.

Changes which may trigger the redrawing of an existing graph can have multiple causes. The dataset which is represented by the graph may have been subject to change, e.g. when new entries were added or deleted or references have been adjusted. Furthermore, the user may set (or unset) filter options in an interactive visualization system or view data in another context, which also leads to a redrawing of a graph.

It is believed that, once a human generates a Mental Map of a graph layout, it is best to maintain the Mental Map for subsequent layouts [1, 4, 8]. This enables humans to reorient themselves in the new updated graph visualization and perceive changes in the graph more quickly. This can be achieved by applying specialized *layout transition algorithms*, which consider a preexisting layout rather than computing a completely new layout, cf. *layout initialization algorithm*. However, too much emphasis on maintaining the Mental Map can lead to unaesthetic layouts, which may severely hamper readability and perception of the graph by humans.

We have investigated criteria for: (1) improving the readability of a graph layout (aesthetic criteria), and (2) maintaining the Mental Map. We use the Simulated Annealing (SA) algorithm [2, 4], a metaheuristic approximation technique to explore the trade-off in terms of aesthetic cost, Mental Map cost and algorithm runtime between the two types of criteria.

Although work has been conducted on the effectiveness of layout transition algorithms compared with layout initialization algorithms [1, 8, 10], there has been no evaluation and comparison of the trade-off between aesthetic and Mental Map criteria in a cost function-based metaheuristic algorithm such as Simulated Annealing. We adapt the SA-based algorithm proposed by Lee et al. [4], whose cost function encompasses five aesthetic and six Mental Map related criteria. For any input, the cost function evaluates all criteria and returns the sum. We modified that behavior to sum up the results for both categories and then return a *weighted sum* instead. Hereby we want to gain new insights into how the results of SA-based graph layout algorithms emerge and how to possibly improve human assimilation and understanding.

This paper is structured as follows: In section two, we go over preliminary definitions and survey optimization criteria for aesthetics and Mental Map preservation. Then we present our methodology in section three and evaluate the results of our work in section four. Finally we conclude by summing up our findings and proposing new approaches for further research in this area in section five.

## 2 Preliminaries

In this section we describe and define the terminology used throughout this paper and give an overview of the algorithm proposed by Lee et al. [4], which is the subject of our investigation.

To visualize a Graph  $G = (V, E)$ , where  $V$  is a set of vertices and  $E \subseteq V \times V$  is a set of edges, we first need to calculate a layout  $L(G)$  for that graph. It provides coordinates for each vertex in the given space – typically two or three dimensions. This is depicted by the function  $pos : V \rightarrow \mathbb{R}^n$ , where  $n \geq 1$  denotes the number of dimensions. The edges  $e \in E$  can be assumed to be straight lines connecting their incident vertices and thus be represented as a vector in  $n$ -dimensional space. If edges are to be represented by more complex figures, such as a sequence of straight lines or splines (Bézier curves), the respective parameters also have to be provided by the layout (position of edge bends, control points for Bézier curves etc.). Sometimes it may also be desirable to provide dedicated positions for placing vertex and edge labels.

Then, in a next step, a layout  $L(G)$  can be used as a skeleton to create the actual visualization. Geometric shapes such as circles or polygons are placed at the vertices' reference coordinates and adjacent vertices are connected using a visual edge representation as discussed above. While the graph  $G$  is a logical view on the data,  $L(G)$  logically represents the visualization.

For each graph there is an infinite number of layouts, as even slightly moving one of the vertices results in a different layout. However, the solution space can be reduced by imposing conditions, such as a finite area (or volume for three-dimensional layouts) where vertices can be placed. Further restrictions may be implied by drawing conventions [3]:

- Straight-line drawing: Vertices are connected by straight lines.
- Poly-line drawing: A superset of straight-line drawings, where vertices are connected by a sequence of straight lines.
- Octilinear drawing: A subset of poly-line drawings, where two segments must enclose an angle  $\alpha$  where  $\alpha$  is an integer multiple of  $45^\circ$ . Segments may only run horizontally, vertically or at a  $45^\circ$  angle in-between.
- Orthogonal drawing: A subset of octilinear drawings, where the angle  $\alpha$  between two segments must be an integer multiple of  $90^\circ$ . Segments may run only horizontally and vertically.
- Grid drawing: Vertices and intersections of edges must be placed at integer coordinates.
- Upward/Downward drawing: For directed acyclic graphs (DAGs) only, placing vertices and edges in such a way that edges only run in vertically non-decreasing (upward) or non-increasing (downward) directions.

Nevertheless, even after restricting the space for possible layouts, it may still be (infinitely) large. For a given graph, not all of these layouts are equally well suited for visualization. Thus, finding (an approximation of) the best layout according to some criteria is an *optimization problem*.

We define two different classes of algorithms which provide graph layouts. Layout initialization algorithms (Function 1) take as input a graph  $G = (V, E)$  and provide a layout  $L(G)$  as output. Layout transition algorithms (Function 2) on the other hand need two graphs  $G_1 = (V_1, E_1)$  and  $G_2 = (V_2, E_2)$  and a layout  $L(G_1)$  as input to calculate  $L(G_2)$ . Additionally, the algorithms may require

further algorithm-specific parameters, e.g. electrical charge, spring length and stiffness for spring algorithms.

$$\textit{initialize} : G \rightarrow L \quad (1)$$

$$\textit{transition} : G \times G \times L \rightarrow L \quad (2)$$

Layout initialization algorithms generate graph layouts with respect to aesthetic criteria which are intended to support readability of the resulting visualization. Aesthetic criteria include, but are not limited to: Minimizing edge intersections (crossings), minimizing the number of edge bends, minimizing the variance of edge lengths (i.e. provide a layout where all edges are of roughly the same length), maximize the minimum angle between two adjacent edges (i.e. two edges incident to the same vertex) [3,8]. Minimizing the number of edge intersections and bends are believed to be by far the most important criteria to be optimized [6,7]. It is however not possible to optimize all criteria at once. Some may even contradict each another, for example when an edge intersection could be avoided by detouring one of the intersecting edges, thus adding to its length and number of edge bends.

While it is convenient that this kind of algorithm provides layouts without any further information but the graph to be visualized – and possibly algorithm-specific parameters – this is also one of its drawbacks. Solving the graph layout optimization problem is usually done iteratively, starting at a random point with a randomly generated layout which is then adjusted in each iteration until the algorithm halts. Therefore, once a graph has been altered and a new visualization is deemed necessary, it may look completely different from the previous one. This implies two problems. As the user’s Mental Map [5] is destroyed, he must spend time to reorient himself. But even when the user generated a new Mental Map, it may still be hard to identify the changes, i.e. spot newly added or removed vertices and edges.

In order to mitigate the destruction of the Mental Map and the need for reorientation, a second class of algorithms – layout transition algorithms – were proposed. Besides the graph to be visualized, they take another graph and its layout as input. The resulting layout is optimized for both aesthetic appearance – as in a layout initialization algorithm – but also for Mental Map preservation (recognition). To fulfill that task, the algorithm needs the preexisting graph and its layout. As the Mental Map criteria are optimized *in addition* to aesthetic criteria, even more competition among the criteria arises. Especially the two groups of criteria – aesthetic and Mental Map – contradict each other [8,9]. The objective of our work is therefore to investigate this competition.

Let  $G_1 = (V_1, E_1)$  be the preexisting graph,  $L_1 = L(G_1)$  its layout,  $G_2 = (V_2, E_2)$  the new graph and  $L_2 = L(G_2)$  the new layout, i.e. the transition algorithm’s output. Criteria for Mental Map preservation are defined on the set of common vertices  $V_{\textit{common}} = V_1 \cap V_2$  and common edges  $E_{\textit{common}} = E_1 \cap E_2$ . Mental map preserving criteria include [4]:

- Relative vertex positions: Let  $v_1, v_2 \in V_{\textit{common}}$ . If  $v_1$  was placed to the left/right of  $v_2$  in the preceding layout ( $L_1$ ), it should also be placed to the

left/right in the new layout ( $L_2$ ). The same condition should hold for placing vertices above/below each another.

- Average relative distance: Let  $p_{i,old}$  and  $p_{i,new}$  be the positions of vertex  $v_i \in V_{common}$  in the old and new layout respectively. The average euclidean distance between these positions  $d(p_{i,old}, p_{i,new})$  for all common vertices should be as low as possible, as moving them too far from their original position destroys the Mental Map.
- Nearest neighbor between: In addition to the above assumptions, let  $p_{j,new}$  be the position of vertex  $v_j \in V_2, j \neq i$ . The distance  $d(p_{i,old}, p_{i,new})$  should be smaller than any  $d(p_{i,old}, p_{j,new})$ . In other words, each vertex should be its own nearest neighbor and no other vertex should be positioned closer to its original position.
- Nearest neighbor within: Let  $v_i, v_j \in V_{common}$  and  $nn_L : V_{common} \rightarrow V_{common}$  determine a vertex' nearest neighbor in terms of euclidean distance the respective layout. Then, the following condition should hold:  $\forall v \in V_{common} : nn_{L_1}(v) = nn_{L_2}(v)$ . The nearest neighbor relations should be retained.
- Let  $dir_L : E_{common} \rightarrow \{N, NW, W, SW, S, SE, E, NE\}$  determine an edge's direction in either layout. The directions of common edges should be maintained:  $\forall e \in E_{common} : dir_{L_1}(e) = dir_{L_2}(e)$ .

The algorithm proposed by Lee et al. [4] uses a cost function (Function 3) to evaluate a given layout. It is derived from the cost function of Davidson and Harel's algorithm [2]. Therefore, it can be used in a SA-based algorithm for layout transition and also layout initialization, when the Mental Map related criteria are "deactivated" (do not contribute to the cost function). The function encompasses five aesthetic and six Mental Map criteria, including some of the criteria described in this section.

$$cost(L(G_1), L(G_2)) = aesthetic(L(G_2)) + mm(L(G_1), L(G_2)) \quad (3)$$

For the purpose of our work, we modified the cost function (Function 4) to be parametrized with a mental map preservation factor (mmp-factor). For  $mmp = 0.5$  this resembles the original cost function by Lee et al. [4], but effectively returns half the cost. This may influence the SA algorithm's probability to accept an inferior neighbor layout  $L'$ , which depends on the *absolute* cost difference divided by the current temperature (Eq. 5). This effect can be compensated for by using  $0.5 \cdot T_0$  as the initial temperature, where  $T_0$  is the initial temperature in the original algorithm.

$$cost_{mmp}(L(G_1), L(G_2)) = (1 - mmp) \cdot aesthetic(L(G_2)) + mmp \cdot mm(L(G_1), L(G_2)), 0 \leq mmp \leq 1 \quad (4)$$

$$P_T(\text{accept } L' | cost(L') > cost(L)) = \exp\left(\frac{cost(L') - cost(L)}{T}\right) \quad (5)$$

### 3 Methodology

The evaluation of the trade-off between Mental Map preservation and optimizing aesthetics is achieved by calculating several graph layouts, using Lee et al.'s algorithm [4], with our modified cost function (Function 4) as described in Sect. 2. Then, for each layout we separately evaluate the aesthetic and Mental Map cost as well as the time (in seconds) it took the Simulated Annealing algorithm to calculate the result.

#### 3.1 Random Graph Generation and Modification

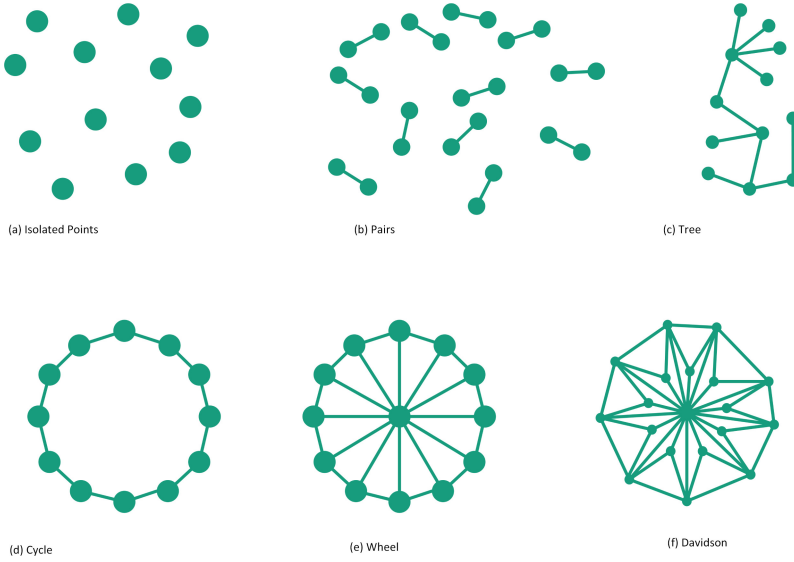
We use six different graph structures (Fig. 1) as a basis for the layouts. They are graphs of roughly the same size in terms of the number of vertices  $|V|$ , but with different levels of connectivity, i.e. different  $|V|$ -to- $|E|$  ratios. This ensures a diverse basis of graph structures for the layout calculations, which is important as the different  $|V|$ -to- $|E|$  ratios lead to varying cost function results. For example, the graph in Fig. 1f will likely have higher aesthetic cost than the graph in Fig. 1d, since it contains considerably more edges which may cause more edge intersections.

For each graph, ten series of random modifications  $S_{i,j} = (G_1, \dots, G_{10})$ ,  $1 \leq i \leq 6$ ,  $1 \leq j \leq 10$  were created, where  $G_1$  is one of the initial graph structures and each consecutive  $G_k$  is determined by the function *modify*:  $G \rightarrow G$  (Function 6). Except for the tree structure (Fig. 1c), all initial structures are always the same. However, the ten tree modification series  $S_{3,j}$  start with different initial trees, as they can be randomly generated.

$$G_{k+1} = (V_{k+1}, E_{k+1}) = \text{modify}(G_k), 1 < k \leq 10 \quad (6)$$

The modification scheme, which infers  $G_{k+1}$  from  $G_k$  works as follows: First, it is decided whether  $G_k$  shall be modified by inserting or deleting vertices, the former occurs with probability  $p_{insert} = 0.7$  and the latter with  $p_{delete} = 0.3$ . The probabilities were chosen because adding vertices is the more interesting use case for layout transition, as removing elements from the graph can be visualized by fading their visual representations. We further decided to *either* delete *or* insert elements from/to the graph, since performing both operations at the same time can be emulated by applying the *modify* function consecutively.

In the case of a delete operation, up to three vertices are randomly selected and together with their incident edges removed from  $G_k$ . The probabilities to select either one, two or three vertices are uniformly distributed ( $p(x = X) = \frac{1}{3}$ ). When vertices are added to the graph, the same probabilities apply to add either one, two or three vertices. If  $V_k$  is not empty, each of the new vertices is connected to one  $v \in V_k$ . Hereby  $G_{k+1}$  is guaranteed to be connected if  $G_k$  was connected and  $G_{k+1}$  is a tree if  $G_k$  was a tree. The latter aspect is important for the graph series  $S_{3,j}$ , which has a tree as initial graph structure and all subsequent  $G_k$ s should remain a tree. To maintain comparability with the series of the other five initial graph structures, this scheme was applied for all random modifications.



**Fig. 1.** The initial graph structures

Of course, the initial graph structures and the way they are modified cover only a tiny fraction of all possible graphs and graph visualizations. We aimed to emulate a small database system, where the vertices are table entries and edges foreign keys which hold a reference to other entries. Since the time for layout calculation depends heavily on the graph size ( $|V|$ ), we chose relatively small initial graph structures and accordingly, relatively small changes for each step of the *modify* function.

### 3.2 Evaluation

For each of the dynamic graph series  $S_{i,j} = (G_1, \dots, G_{10})$ , six series of layouts  $(L(G_1), \dots, L(G_{10}))$  were calculated. Five of them are the result of Lee et al.'s algorithm [4] using our modified cost function (Function 4) with  $mmp = 0, 0.25, 0.5, 0.75$  and 1. However the first layout  $L(G_1)$  in each series is an initialized layout, where from a random layout as starting point, only aesthetic cost are optimized. The sixth series serves as a control group where each  $L(G_k)$  is an initialized layout. The only difference between layout initialization and layout transition with  $mmp = 0$  is that the former uses a random layout as starting point and the latter assigns each  $v \in V_{common}$  its previous position and random positions for all remaining (new) vertices.

Then, aesthetic cost, Mental Map cost and algorithm runtime for each layout are separately evaluated. The cost are compared by dividing the transitioned layouts' cost by the initialized layouts' cost to determine the relative quality of both layout transition and initialization (Functions 7 and 8). Thus, a quotient

greater than 1.0 indicates the layout initialization algorithm obtained a better result, a quotient lower than 1.0 on the other hand implies layout transition generated a better result. We also investigate the quotient of layout transition runtime divided by layout initialization runtime.

$$Q_{aesthetic}(L_{transitioned}, L_{initialized}) = \frac{aesthetic(L_{transitioned})}{aesthetic(L_{initialized})} \quad (7)$$

$$Q_{mm}(L_{transitioned}, L_{initialized}) = \frac{mm(L_{transitioned})}{mm(L_{initialized})} \quad (8)$$

### 3.3 Limitations

The aesthetic part of Lee et al.’s algorithm [4], which is based on the work of Davidson and Harel [2], requires four parameters which provide relative weights between the aesthetic criteria. Finding an optimal set of parameters is not a trivial task. Parameters which are suitable to create a layout for one graph may be completely improper when visualizing another graph. We decided to find parameters which draw the Davidson graph (Fig. 1f) nicely and apply them to each of the layout calculations. Therefore, some of the resulting graph layouts do not look nicely in terms of human readability. However, this approach ensures comparability between our results. It is furthermore easy to see that not all aesthetic criteria actually lead to a nice looking graph layout. For instance, equal edge lengths and evenly spread out vertices may be desirable to draw trees (Fig. 1b, c) but are entirely negligible for graph structures as presented in Fig. 1e, f.

During a short preliminary experiment we found the parameters  $\lambda_1 = 0.2$ ,  $\lambda_2 = 0$ ,  $\lambda_3 = 1$  and  $\lambda_5 = 200$  to produce layouts for the Davidson graph similar to the one presented in Fig. 1f. The parameter  $\lambda_4$  depends on  $\lambda_5$  and the minimal vertex-to-edge distance. Setting  $\lambda_2 = 0$  effectively turns off the Borderlines criterion, which ensures vertices do not come too close to any of the borders of the specified rectangle, in which vertices may be placed. Our preliminary experiment showed that using this criterion prevented the layout algorithm from *efficiently* using the available space, which lead to otherwise not nice-looking layouts, e.g. by inserting unnecessary edge intersections or generating an overall cluttered view of the graph. The values for  $\lambda_1$  and  $\lambda_3$  for node distribution and equal edge lengths respectively are relatively low, as they do not contribute to a nice drawing of the Davidson graph.

The rectangular area in which vertices are placed is a  $1000 \times 1000$  raster. Initial temperature for the Simulated Annealing algorithm was  $10^5$  and a geometric temperature reduction schedule with  $\gamma = 0.75$  and the polynomial to determine the stage size was  $p(n) = 30 \cdot n$  [4], where  $n$  is the number of vertices. The algorithm halts after the same layout occurred for three consecutive stages.

All results presented in Sect. 4 must be interpreted in the context of the parameters used for the algorithm and also the initial graph structures and random modification scheme. Furthermore, as we investigate the Simulated Annealing algorithm’s ability to optimize aesthetic and Mental Map related criteria by



using a white-box approach, the results do not necessarily have to reflect a human’s perception. This means, it has to be verified to which extent the cost functions for aesthetics and Mental Map reflect a human’s perception. Finding a good set of parameters is a complex task which greatly depends on the graph to be visualized and is outside the scope of our work.

## 4 Results

### 4.1 Hypotheses

Before we started the layout calculation and evaluation, we have come up with hypotheses regarding the experiment’s outcome.

1. Except for 0% mental map preservation factor (mmp-factor), the initialized layouts are expected to have less aesthetic cost, since the layout initialization algorithm solely optimizes this type of criteria. However, when using a mmp-factor of 0%, the layout transition algorithm will also only optimize aesthetics. It is thus difficult to make a prediction for this case, but possibly the layout transition algorithm’s results have even less aesthetic cost, since the algorithm starts from an already good layout rather than random positions.
2. The aesthetic quotient  $Q_{aesthetic}$  will negatively correlate with the mmp-factor.
3. Transitioned layouts are expected to always have less Mental Map cost than the initialized layouts. Even with 0% mmp-factor, the randomness is eliminated and instead, vertices are assigned their previous locations as a starting point for subsequent layout calculations.
4. The Mental Map quotient  $Q_{mm}$  will positively correlate with the mmp-factor.
5. For 0% mmp-factor we expect the runtime for layout initialization and transition to be equal as in this case the algorithms are equivalent.
6. For  $0\% < \text{mmp-factor} < 100\%$  the runtime of layout transition is likely higher than for layout initialization. This is because in each iteration, more criteria need to be evaluated by the cost function, thus prolonging the time to complete one iteration. Furthermore, there it may take more iterations until a trade-off between all criteria is found.

### 4.2 Development of Aesthetic and Mental Map Criteria

Each line in Table 1 represents the arithmetic mean values of the aesthetic quotients  $Q_{aesthetic}$  (Function 7). For mmp-factor 100% all aesthetic quotients are in the order of magnitude  $10^{11}$  to  $10^{15}$  (Table 1). Also, at a first glance, the aesthetics of the Davidson and Wheel graphs degrade a lot when using the layout transition algorithm, even if the mmp-factor is 0%. However, when looking at the variance in Table 2 it is evident that further investigation of the parameters is needed.

Comparing the mean aesthetic quotients of Table 1 to their variance in Table 2, shows that all aesthetic quotients which are considerably larger than

**Table 1.** Experimental results – arithmetic mean values of  $Q_{aesthetic}$ 

mmp-factor	0%	25%	50%	75%	100%
Davidson	$1.87 \times 10^{10}$	$1.85 \times 10^9$	$6.01 \times 10^{11}$	$1.05 \times 10^8$	$1.40 \times 10^{11}$
Isolated	1.18	3.21	6.56	4.11	$5.39 \times 10^{12}$
Pairs	1.18	1.12	1.20	1.51	$9.75 \times 10^{14}$
Ring	1.11	1.13	1.18	1.46	$1.37 \times 10^{15}$
Wheel	$9.08 \times 10^{12}$	$1.89 \times 10^{11}$	$5.66 \times 10^{11}$	1.22	$1.10 \times 10^{15}$
Tree	1.07	3.77	4.99	6.23	$3.28 \times 10^{15}$

**Table 2.** Experimental results – variance of  $Q_{aesthetic}$ 

mmp-factor	0%	25%	50%	75%	100%
Davidson	$1.44 \times 10^{21}$	$1.37 \times 10^{19}$	$3.25 \times 10^{24}$	$9.84 \times 10^{16}$	$2.47 \times 10^{22}$
Isolated	$4.71 \times 10^{-2}$	8.16	$1.12 \times 10^2$	$1.05 \times 10^1$	$1.40 \times 10^{25}$
Pairs	$1.61 \times 10^{-2}$	$3.10 \times 10^{-2}$	$1.56 \times 10^{-2}$	$7.26 \times 10^{-2}$	$6.29 \times 10^{29}$
Ring	$3.79 \times 10^{-2}$	$2.80 \times 10^{-2}$	$2.49 \times 10^{-2}$	$4.08 \times 10^{-2}$	$7.30 \times 10^{29}$
Wheel	$5.13 \times 10^{26}$	$3.23 \times 10^{23}$	$2.88 \times 10^{24}$	$3.25 \times 10^{-1}$	$6.48 \times 10^{29}$
Tree	$3.98 \times 10^{-2}$	$3.51 \times 10^1$	$9.13 \times 10^1$	$1.47 \times 10^2$	$6.14 \times 10^{30}$

one have an even higher variance, roughly the mean value squared. For further investigation, the mean aesthetic quotient for the ten graph series  $S_{6,j}$  (i.e. the Davidson graph series) are detailed in Table 3. The bottom row of Table 3 indicates the probability that a given transitioned layout has less aesthetic cost than

**Table 3.** Experimental results – all aesthetic quotients  $Q_{aesthetic}$  of the Davidson graph

mmp-factor	0%	25%	50%	75%	100%
1	0.339	2.05	0.481	0.245	2.83
2	$7.98 \times 10^{10}$	0.0182	0.118	$1.05 \times 10^9$	$6.83 \times 10^9$
3	$1.77 \times 10^{-12}$	0.0404	0.0273	0.227	$1.15 \times 10^1$
4	0.370	$9.08 \times 10^9$	0.0833	0.00768	3.30
5	0.364	0.399	$6.01 \times 10^{12}$	0.171	$4.90 \times 10^{10}$
6	$1.08 \times 10^{11}$	0.757	0.533	$1.58 \times 10^1$	$4.37 \times 10^{11}$
7	0.799	0.912	1.01	0.832	$2.99 \times 10^{11}$
8	0.369	$9.45 \times 10^9$	0.545	0.808	$2.27 \times 10^{11}$
9	0.543	0.659	0.939	0.588	$3.25 \times 10^{11}$
10	0.896	0.767	0.992	0.527	$5.68 \times 10^{10}$
$p(Q_{aesthetic} < 1)$	0.8	0.7	0.8	0.8	0

its newly initialized counterpart. This probability is inferred from the experimental results. The finding shows that – except for 100% mmp-factor – the layout transition algorithm was able to produce layouts for the Davidson graph series with better aesthetic cost compared to the initialization algorithm.

The aesthetic quotients of the graphs Isolated Points, Pairs, Ring and Tree – which are graphs without runaway values – are within the same order of magnitude and plotted in Fig. 2a. The quotients of the Davidson and Wheel graphs are also included in this Figure, however the values differ from Table 1, since runaway values had to be excluded to be able to plot the quotients.

The Ring and Pairs graphs have equivalent aesthetic quotients. Up to 50% mmp-factor they are slightly above 1 with an increase to 1.5 if using 75% mmp-factor. The Isolated Points graphs’ aesthetic quotient rises noticeably as the mmp-factor increases. Yet the transitioned and initialized layout look equally well in terms of human readability. That is due to the lack of edges in this graph: Effectively, only the vertex distribution criterion dictates the aesthetic cost, as the contribution of uniform edge length, edge crossings and vertex-edge distances are close to zero. The Tree graphs’ aesthetic quotient scales almost linearly with the mmp-factor.

All Mental Map quotients  $Q_{mm}$  are plotted in Fig. 2b. With 0% mmp-factor the results show large differences between the graph structures, since the Mental Map quotients are widely spread. The minimum improvement was about 5% cost reduction for the Isolated Points graph and up to 40% less Mental Map cost for the Davidson graph.

All graphs’ Mental Map quotients scale with the mmp-factor, with a large drop from 75% to 100% weight. Interestingly, the Pairs and Wheel graphs have the exact same results for all five mmp-factor.

At 25% mmp-factor all but the Davidson graph Mental Map quotients drop. The most significant drop occurs for the Isolated Points graph, which had the highest quotient at 0% mmp-factor and is now on par with the Davidson graph’s quotient (both 0.6). Between 25% and 75% mmp-factor the Pairs, Ring Wheel and Tree graphs’ Mental Map quotient decrease almost linearly. The Davidson graph has a noticeable drop at 50% mmp-factor but remains stable until the next step.

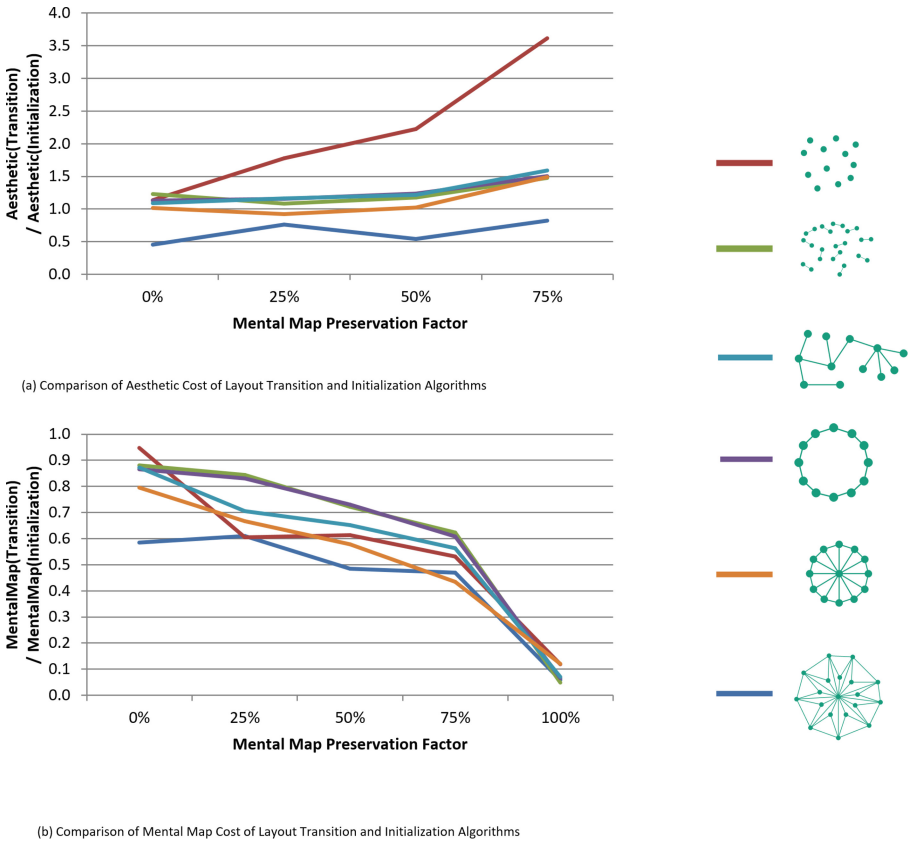
To further evaluate the overall quality of layout transition vs. initialization, we merge the results of aesthetic and Mental Map analysis. Therefore,  $Q_{aesthetic}$ ’s and  $Q_{mm}$ ’s median values are multiplied and the product is plotted in Fig. 3a. The idea behind this approach is that, for instance, five times lower Mental Map cost even out five times higher aesthetic cost. Except for an offset, the trajectories are the same when compared to the aesthetic quotients (Fig. 2a). This implies that, whatever the actual mmp-factor may be, aesthetics dictate the overall quality of a layout.

However, layout transition performs better in most cases as the overall quality is slightly better for the Pairs, Ring and Tree graphs and considerably better for the Davidson and Wheel graphs. Only the Isolated Points graph has a better overall quality when using layout initialization.

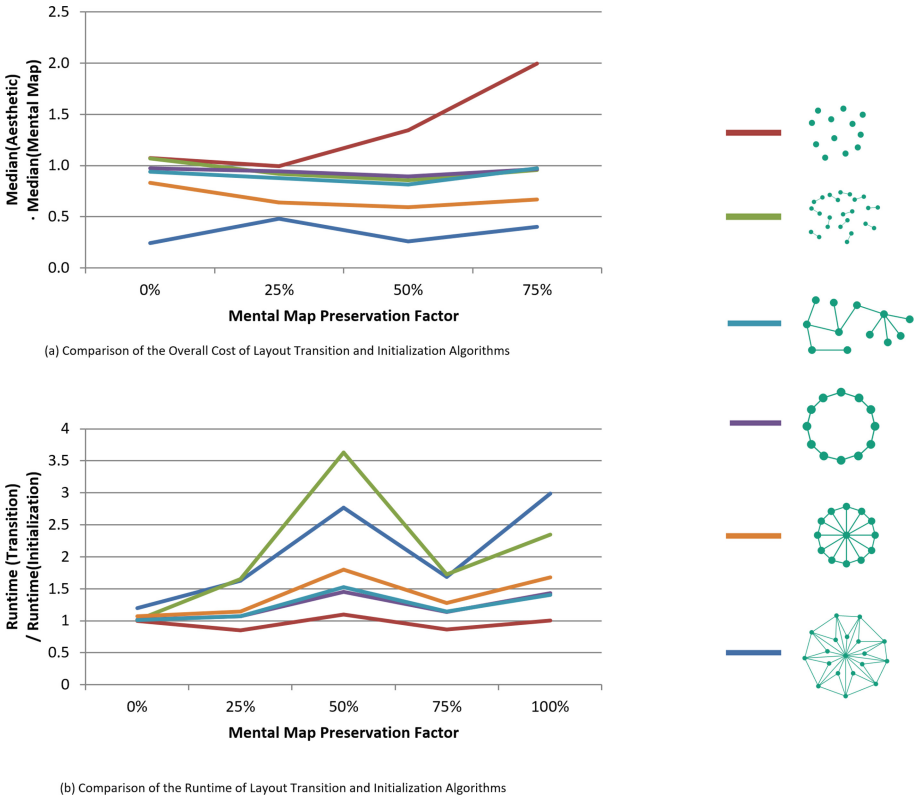
### 4.3 Algorithm Runtime

The runtime quotients which compare the runtime of layout transition to layout initialization are illustrated in Fig. 3b. As a basis for the runtime analysis we chose the actual runtimes measured in seconds rather than iterations of the Simulated Annealing algorithm. This is because just analyzing the iterations would not account for runtime differences per iteration. This is however an important aspect, because some of the additional Mental Map criteria require  $\mathcal{O}(|V|^2)$  time in each iteration to be evaluated. Hence, measuring the relative runtime in terms of seconds instead of iterations allows for a more realistic assessment of layout transition’s feasibility for interactive graph visualization systems.

Since all quotients compare the respective runtime of layout transition to the same runtime of layout initialization, one can compare the quotients for each graph. For instance, the Davidson graph’s  $Q_{runtime}$  for 100% mmp-factor is 3



**Fig. 2.** Experimental results – evolution of  $Q_{aesthetic}$ ’s mean value (top) and  $Q_{mm}$ ’s mean value (bottom)



**Fig. 3.** Experimental results – evolution of  $Q_{total}$ 's median value (top) and the runtime comparison (bottom)

and for 0% roughly 1.2 (Fig. 3b). Thus, layout transition with 100% mmp-factor takes about 2.5 times longer than with 0%.

All quotients share a w-like trajectory with a peak at 50% mmp-factor. While the Pair and Ring graphs had similar aesthetic and Mental Map quotients, their runtime quotients differ a lot: The Pair graph has (for the most part) the highest relative runtime and the Ring graph the second lowest (together with the Tree graph).

Setting the mmp-factor to 50% – which is equivalent to Lee et al.'s original algorithm – yields the overall highest relative runtime for all graphs. The fact that with 100% mmp-factor runtimes are always higher than for 0%, shows that Mental Map criteria take more time to converge than aesthetic criteria.

#### 4.4 Result Interpretation

The first hypothesis, which states that initialized layouts have better aesthetic cost than their transitioned counterparts for mmp-factor > 0% is true. When

using 0% mmp-factor, i.e. using the initialization algorithm with a non-random starting point, aesthetic cost improved in some cases, but for the most part aesthetic cost were slightly increased. On the other hand, even though Mental Map related criteria were not explicitly optimized in this case, the cost in this category improved between 5% and 40%. This shows that when using a previous layout rather than a randomized one, existing layout initialization algorithms are also capable of implicitly maintaining the Mental Map.

However, in some instances of the Davidson and Wheel graphs, aesthetic cost of transitioned layout increased more than  $10^{10}$  times, even when only aesthetic cost were optimized. Due to the limited scope of our experiment it is not feasible to make a general statement as to why this occurs.

When increasing the mmp-factor, the quotient  $Q_{aesthetic}$  increases a bit, but remains relatively stable, hence the results confirm the second hypothesis. When optimizing only Mental Map criteria and ignoring aesthetics, i.e. setting mmp-factor to 100%, all transitioned layouts had more than  $10^{10}$  times higher aesthetic cost than initialized layouts. When only Mental Map cost dictate the cost function result, the newly added vertices can be arbitrarily placed. This is because new vertices are only associated to aesthetic cost and thus, moving them to an arbitrary position does not change the cost function result. On the other hand, moving an old vertex inevitably increases the cost function, since any movement is directly translated to aesthetic cost. In summary, the modification of the cost function proposed in this paper (Function 4) is not suitable for high degrees of Mental Map preservation (mmp-factor  $> 0\%$ ). An alternative approach would be to apply the mmp-factor only to old vertices, but not to new ones.

Hypotheses three and four could be confirmed, as all transitioned layouts had lower Mental Map cost compared to initialized layouts and with rising mmp-factor, the relative cost were further reduced. When analyzing the algorithm runtime, hypothesis five can be partially confirmed. With 0% mmp-factor, runtimes of layout transition and initialization were mostly equivalent, thus confirming hypothesis five. However, it took about 1.2 times longer when transitioning layouts based on the Davidson graph.

When the mmp-factor is set between 0% and 100%, layout transition takes longer than initialization, the only exception being graphs based on the Isolated Points structure. At 100% mmp-factor relative runtimes vary between one (initialization and transition take equal time) for the Isolated Points graph structure and three for the Davidson graph structure. From this it could be inferred that Mental Map criteria take longer to converge than aesthetic criteria. However, we believe that the prolonged runtime is another result of entirely neglecting aesthetic cost of new vertices. In the first stages, the underlying Simulated Annealing algorithm has high acceptance rates of non-improving layouts. In this case, such non-improving layouts are layouts, where old vertices have been displaced from their original location. This displacement then provides the potential for reducing cost in later iterations, as the old vertices are moved back to their original locations. Furthermore, the displacement per iteration is higher at the beginning, since vertices move less in later iterations. These unnecessary steps –

displacing old vertices and then moving them back to their original locations – may be considered a crucial factor for the prolonged runtime.

## 5 Conclusions and Future Work

In this work we have examined how aesthetic and Mental Map criteria compete and influence one another. To achieve this goal, formal definitions to distinguish between layout initialization and transition algorithms were introduced and optimization criteria for aesthetic and Mental Map preservation surveyed.

The main contribution is an experiment based on the layout transition algorithm proposed by Lee et al. [4]. A modification of the original cost function allows to weight the aesthetic and Mental Map criteria within the cost function, making it more flexible and allowing for a greater insight in how the Simulated Annealing algorithm’s results emerge. The experiment compares the aesthetic and Mental Map cost of transitioned layouts to the according cost of initialized layouts.

Three major aspects can be derived from the experiment. Firstly, an explicit optimization of Mental Map criteria is not necessary in order to improve Mental Map cost. When calculating a layout for a modified graph, rather than using a random layout in the first iteration of the Simulated Annealing procedure, the positions of old vertices can be used. By using the previous layout as a starting point for all subsequent calculations, the Mental Map is implicitly maintained and the associated cost improve by 5% to 40%, depending on the concrete graph structure. Thus, existing layout initialization algorithms may suffice when they do not start with a random layout.

Furthermore, weighing aesthetic and Mental Map criteria equally – as in the original algorithm by Lee et al. [4] – has the highest runtime. When focusing on either aesthetic or Mental Map, the relative runtime between transition and initialization reduces compared to mmp-factor 50%. This is because the competition between the two types of criteria is reduced, as one type of criteria dominates the cost function.

Lastly, the extreme increase in aesthetic cost when only optimizing Mental Map cost but ignoring aesthetics shows a disadvantage of our modification to the cost function. The more focus is on Mental Map preservation, i.e. if the mental map preservation factor (mmp-factor) is greater than 50%, the placement of new vertices contributes less to the cost function result, since new vertices only have aesthetic but no Mental Map related cost. Thus, if a weighted sum-based cost function is desired, the weight should only apply to costs caused by old vertices.

The focus of our work lies on the Simulated Annealing algorithm’s ability to handle the trade-off and competition between aesthetic and Mental Map criteria. Therefore, further research should study how the cost function results correlate with human perception of aesthetics and Mental Map preservation and if so, which weight(s) is/are optimal.

Further research can be conducted with alternative approaches for weighing the Mental Map. For instance, transitioned layouts may look nicer when applying

the weight factor between aesthetics and Mental Map only to old vertices and take the new vertices' aesthetic cost fully into account to achieve optimal placing of new vertices while preserving the Mental Map criteria for old ones.

The effects of changes to other aspects of the Simulated Annealing algorithm are also subject to further investigation. It may be possible to obtain better results when adjusting the probability to select a certain vertex to be moved, e.g. by first deciding whether to move an old or new vertex and then selecting a random vertex from the given group. Then, when a graph with 99 vertices receives one new vertex, its probability to be moved in each iteration is 50% rather than 1%, possibly allowing the algorithm to place the new vertex in an optimal position with respect to the old vertices, while not moving the old vertices too much around and thus, destroying the Mental Map.

## References

1. Bridgeman, S., Tamassia, R.: A user study in similarity measures for graph drawing. In: Marks, J. (ed.) GD 2000. LNCS, vol. 1984, pp. 19–30. Springer, Heidelberg (2001). [https://doi.org/10.1007/3-540-44541-2\\_3](https://doi.org/10.1007/3-540-44541-2_3)
2. Davidson, R., Harel, D.: Drawing graphs nicely using simulated annealing. *ACM Trans. Graph.* **15**(4), 301–331 (1996)
3. Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G.: *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, Englewood Cliffs (1999)
4. Lee, Y.-Y., Lin, C.-C., Yen, H.-C.: Mental map preserving graph drawing using simulated annealing. In: *Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation, APVis 2006, Darlinghurst, Australia*, vol. 60, pp. 179–188. Australian Computer Society Inc. (2006)
5. Misue, K., Eades, P., Lai, W., Sugiyama, K.: Layout adjustment and the mental map. *J. Vis. Lang. Comput.* **6**, 183–210 (1995)
6. Purchase, H.C., Cohen, R.F., James, M.I.: An experimental study of the basis for graph drawing algorithms. *J. Exp. Algorithmics* **2** (1997)
7. Purchase, H.: Which aesthetic has the greatest effect on human understanding? In: DiBattista, G. (ed.) GD 1997. LNCS, vol. 1353, pp. 248–261. Springer, Heidelberg (1997). [https://doi.org/10.1007/3-540-63938-1\\_67](https://doi.org/10.1007/3-540-63938-1_67)
8. Purchase, H.C., Hoggan, E., Görg, C.: How important is the “mental map”? – an empirical investigation of a dynamic graph layout algorithm. In: Kaufmann, M., Wagner, D. (eds.) GD 2006. LNCS, vol. 4372, pp. 184–195. Springer, Heidelberg (2007). [https://doi.org/10.1007/978-3-540-70904-6\\_19](https://doi.org/10.1007/978-3-540-70904-6_19)
9. Purchase, H.C., Samra, A.: Extremes are better: investigating mental map preservation in dynamic graphs. In: Stapleton, G., Howse, J., Lee, J. (eds.) *Diagrams 2008*. LNCS (LNAI), vol. 5223, pp. 60–73. Springer, Heidelberg (2008). [https://doi.org/10.1007/978-3-540-87730-1\\_9](https://doi.org/10.1007/978-3-540-87730-1_9)
10. Saffrey, P., Purchase, H.C.: The “mental map” versus “static aesthetic” compromise in dynamic graphs: a user study. In: *Proceedings of the Ninth Conference on Australasian User Interface, AUIC 2008, Darlinghurst, Australia*, vol. 76, pp. 85–93. Australian Computer Society Inc. (2008)