On the Role of Aesthetics in Genetic Algorithms Applied to Graph Drawing

Evgheni Polisciuc, António Cruz, Penousal Machado and Joel P. Arrais
CISUC, Department of Informatics Engineering, University of Coimbra, Portugal
evgheni@dei.uc.pt,antonioc@student.dei.uc.pt,machado@dei.uc.pt,jpa@dei.uc.pt

ABSTRACT

Despite the role that aesthetics plays in information visualization, it is often downplayed or ignored in favor of functionality. However, by understanding how graphical representations are perceived it is also possible to improve them and create more comprehensible data visualizations. Meaningful relationships and data patterns can easily get lost among the representation of large and complex datasets. Various methods have been created to reduce visual clutter by either sorting nodes to minimize the number of intersecting edges, or by grouping edges into bundles with clear directions. In information visualization, perception principles have started being integrated into evolutionary computation in order to solve aesthetic problems, as they are capable of looking for solutions that may be found beyond local optima. In this paper we present a study on the importance of aesthetics and how evolutionary approaches can be used to influence visualization. This is supplemented with two case studies involving the design of genetic algorithms for reducing visual clutter through edge crossing minimization and edge bundling parameter optimization.

CCS CONCEPTS

•Human-centered computing → Graph drawings; Empirical studies in visualization; Visualization design and evaluation methods; **•Theory of computation →** *Evolutionary algorithms;*

KEYWORDS

Graph Drawing, Genetic Algorithms, Edge Bundling, Circular Network Layout, Aesthetics

ACM Reference format:
Evgheni Polisciuc, António Cruz, Penousal Machado and Joel P. Arrais. 2017. On the Role of Aesthetics in Genetic Algorithms
Applied to Graph Drawing. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 8 pages.
DOI: http://dx.doi.org/10.1145/3067695.3082552

1 INTRODUCTION

The clarification power and the ease of visual communication are some of the reasons that make graph representation so popular among other visualization techniques. The node-link representation is an efficient way to depict entities and relationship between them. However, aesthetics and perceptual aspects are, to a large extent, an open problem modern in graph drawing. Visual clutter, mainly caused by edge crossings, is one of the main sources of perception problems, making it a main focus of current research. In particular, two techniques of clutter reduction – edge crossing minimization and edge bundling – have been hot topics in graph representation research during the last decade. Graph layout algorithms are also important in network representation, but they are out of the scope of this paper.

In fixed graphs, where nodes have fixed positions, edge bundling is applied to reduce visual clutter, which is the minimization of the "white space" to "ink" ratio. In contrast, in graph composition with flexible nature, the number of edge crossings is minimized by re-configuring the layout. Additionally, edge bundling can also be applied in flexible graphs after the layout has been fixed, when the visual clutter issue still persists. Nevertheless, both techniques are mainly concerned with perception and efficiency issues in graph drawing.

In this paper we present two case studies where evolutionary algorithms are applied to minimize the number of edge crossings and white space to ink ratio. The experimental results demonstrate that these simple measures are not sufficient to guarantee that the resulting visual artifacts possess the functional and aesthetics requirements. Therefore, aesthetics has been identified as one of the key issues, as well as a niche for exploration in designing and applying genetic algorithms in information visualization.

This paper begins by surveying the background of aesthetics in information visualization and related work regarding the use of evolutionary algorithms in this field. Afterwards, two case studies are detailed, along with the experimental results and preliminary conclusions. Finally, we present a critical discussion and our overall conclusions.

2 BACKGROUND AND RELATED WORK

This section provides a detailed discussion on aesthetics in the context of information visualization and related work in the application of evolutionary mechanisms in this field.

2.1 Aesthetics and Functionality in Information Visualization

In the context of information visualization, information aesthetics has only recently become a separate field of study. Lau and Vande Moere came up with a model of information aesthetics as interplay between information visualization, in terms of functionality and effectiveness, and visualization art, in terms of artistic influences and meaningfulness [17]. Authors have identified information aesthetics as a triad of *aesthetics*, *data* and *interaction*, which forms a link between the fields of information visualization and visualization art (e.g. information visualization mainly focuses on representing data using interactive techniques with little concern for aesthetics).

More recently researchers studied the effect of aesthetically pleasing layout on visual search performance, and the findings indicate that response time, but not errors, was strongly affected by the aesthetics level [22]. Similarly, another study by Cawthon and Vande Moere indicated that aesthetics affects the usability of data visualizations [5]. Authors measured aesthetics, efficiency and effectiveness of retrieval tasks amongst 11 different visualization techniques, and found correlation between task execution time, as well as error rate, and perceived aesthetics. Additionally, Borkin et al. concludes that aesthetics directly influence the memorability of visualizations (e.g. the inclusion of human recognizable objects or harmonic color pallets enhance memorability) [3]. Common graphs and charts are less memorable than unique visualizations, independently of the levels of understanding and effectiveness of studied visualizations.

Finally, Kosara proposes a classification of information visualization, which is based not just on technical criteria, but on aesthetic criteria as well [16]. He introduces notions of *artistic* and *pragmatic* visualisation, as well as a discussion on their properties, therefore reducing the gap between design, art and technical/pragmatic visualizations. Finally, regarding graph drawing, the work of Ware et al. on cognitive measurements of graph aesthetics empirically tests the axiomatic notions of aesthetics in graphs (e.g. minimizing edge crossings, minimizing the sum of lengths of the edges, etc.), providing a detailed understanding of aesthetics in graph drawing [25]. The results suggest that the length of the path, continuity, edge crossings and number of branches emerging from nodes are the most important factors in reducing cognitive load in graph reading. To sum up, our analysis of the literature indicates that there is still a need for the recognition of aesthetic value in the context of information visualization, particularly in graph drawing.

2.2 Evolutionary Approaches in Information Visualization

Genetic Algorithms (GAs) are a stochastic search approach to find approximate or even optimal solutions to complex problems, inspired in the process of natural selection [10]. GAs begin with a population of randomly generated candidate solutions and optimize these through iterative processes of mutation, recombination and selection, until a solution with sufficient quality is found. A fitness function is used to evaluate each individual in the population and determine their likelihood of surviving into the next generation. In the field of information visualization, GAs have been used not just in data mining, but also to directly improve the aesthetics of a visualization. Wu et al. presented a GA capable of evolving the placement of graphical elements, such as textual labels and images, and find their optimal position on schematic maps [26]. GAs can also be used to optimize the parameters of a visualization. House et al. used a GA along with user-guided evolution to evolve designs towards perceptually near-optimal visualizations [13].

When dealing with datasets of significant size or complexity, the resulting visualizations can lack comprehensibility due to a lack of perceptible order in both graphical elements and their relationships. Visualizations that consist of structures built using nodes connected with edges, such as graphs and networks, are especially susceptible to visual clutter. At a data mining level, GAs can be used to identify clusters in a dataset [15], which can then be used to organize the resulting visualization by creating perceptible groups of nodes.

Despite this, depicting a large number of relationships across these sorted nodes can significantly impact a network's comprehensibility, as intersecting edges can make their own direction and connections unclear. Some force-based layouts promote self-organizing networks, where nodes will take up positions while taking into account both their edges and their similarity with other nodes. While their initial layouts are often random, Ghassemi et al. presented an alternative in the form of a GA, capable of calculating an initial layout for force-directed graphs that reduces edge crossings [8]. There also exist other GAs that have been developed to optimize the position of nodes through the use of different ordering strategies to minimize edge crossings in both directed [24] and undirected graphs [4].

3 CASE STUDIES

This section provides detailed description of both experiments, which are (i) evolving circular networks and (ii) evolving edge bundling parameters, as well as the preliminary conclusions. Both experiments at their core follow a traditional structure, outlining the experimental setup, fitness function and results.

3.1 Circular Network Edge Reduction

Evolving Circular Networks. Circular layouts are a popular metaphor in information visualization and human-computer interaction research [6]. Circle layouts are typically structured with nodes placed along the circumference of a circle, providing a clear ordered structure at the cost of flexibility in positioning. As such, these types of network models are particularly susceptible to visual clutter caused by edge crossings.

The edge crossing reduction in circular networks problem is not recent and there are various different non-evolutionary approaches specifically aimed at ordering the nodes on the outer circle in order to produce a minimal amount of crossings [2, 18, 23]. A specific example of these heuristics is the Adjacent Vertex with Smallest Degree First (AVSDF) by He and Sykora [9], which presented better results than several older algorithms. While various approaches have been taken to reduce edge crossing in networks using genetic algorithms, the problem has not been properly tackled specifically in circular networks.

Experimental Setup. The objective of this case study was the visualization of circular networks using an evolutionary edge reduction approach to create clear and understandable visualizations. Each individual solution in this problem consists of a cyclical permutation containing a sorted array of nodes IDs which define the order of nodes on the circular network. The fitness of the solution is determined by counting how many edge crossings exist in the resulting graph, where the fittest individual has the least amount of crossings.

The experiments used a dataset containing a list of purchases made in supermarket chain that was used to determine which products were bought together, and how frequently. These relationships are represented in the circular network by dividing the outer circle into segments that represent each of the products in the dataset. The size of these segments is proportional to the amount of times

Table 1: Fitness results, corresponding to the number of edge crossings, from the tests performed on the genetic algorithm, using both the insert and swap mutation operators, and the AVSDF heuristic.

	Insert Mutation			Swap Mutation			AVSDF			AVSDF Evolved		
	Best	Median	Worst	Best	Median	Worst	Best	Median	Worst	Best	Median	Worst
Set 1	728	794	953	742	826	996	795	854	965	734	736	740
Set 2	7	9	14	13	37	56	11	15	30	7	7	8
Set 3	985	1075	1241	1005	1112	1361	1045	1083	1087	995	997	998

that each product was bought, while lines are drawn across the interior to connect products that were purchased at the same time. Analyzing this data can help identify and distinguish the strong and numerous relationships that some products that have with others, as well as the groups of products that have very few relationships.

Selection was performed using a stochastic ranking selection method, which returns a wide spread of individuals by selecting the best individuals multiple times, while also picking some individuals with low fitness. Elitism was considered as an option, as it keeps the best individual from getting lost to the genetic operators, but it was not used. Its absence promotes genetic variety which can prevent the algorithm from getting stuck in local optima, and the best individual may still be preserved by adjusting the genetic operator pressures.

Mutations can affect permutations through the use of techniques such as insertion (one element is moved to a different position), swaps (the position of two elements is swapped), and 2-opt (a permutation segment is reversed). Approaching the optimum requires very small search steps so that the solutions do not stray away from it as they get closer, which means that mutations should have very small impacts. When considering the current problem, insertion should cause the smallest amount of changes as it will only affect a single node and its edges, as opposed to 2-opt which alters the positions of a significant number of nodes. However, given that swapping will only affect two nodes, it was also used in testing so that it could be compared against the insertion mutation operator.

Our objective in crossing two permutations is to obtain a new permutation that contains a variable number of similarities in the positions of the nodes from both its parents. To achieve this, we chose geometrical crossover with insertions. In this approach, each node contained in the second permutation is mapped to the positions of the same nodes in the first permutation so that the Longest Increasing Cyclical Subsequence (LICS) [1] can be calculated. The amount of nodes absent from the LICS corresponds to the number of nodes that need to change position in order to transform the second permutation into the first. By performing half of these insertions, we can obtain a new permutation that is equally similar to both its parents.

Fitness Function. Every new individual must be evaluated, unless they are a copy, but counting the number of edge crossings can be demanding. Our edge crossing counting function iterates over every edge in the graph, and as each edge splits the graph into two parts, we can assume that any other edge which contains each of its nodes on a different side will result in an intersection. However, it is not necessary to compare each edge to the remaining others, so instead we iterate only over the edges originating from the half with the least amount of nodes. If one of these edges contains both nodes on that side then these will not result in edge crossings. Furthermore, any edge verified during the first iteration does not need to be considered for following iterations.

Results and Preliminary Conclusions. The tests used three different subsets of the data, which are distinct in terms of structure despite being of similar size. Set 1 contains 150 nodes and 200 edges, Set 2 contains 200 nodes and 198 edges, and Set 3 contains 100 nodes and 187 edges. Test were run 30 times each using populations of 100 individuals for 20000 evaluations. The experimental process was divided into three main steps: compare the mutation operators, validate the best results by comparing them to an existing heuristic, and finally, evolve the results of the existing heuristic in an effort to combine both algorithms.

Regarding the comparison between the insert and swap mutation operators, we were able to observe that for the same conditions in all three sets the insert operator returned solutions with an overall lower amount of edge crossings than the swap operator, as described in Table 1. One of the reasons for this is that the swap operator produces a larger search step due to how it changes the position of two nodes, whereas the insert operator only changes one node. This should cause the fitness to improve faster when using the swap operator during the beginning, as a larger search step is beneficial towards finding the optimum when it is far away. However, when approaching the optimum, the search step should be small as to not overstep the optimum. To further back up this theory, we analyzed the fitness increase over time in each of the tests and discovered that the swap operator generally performs better than the insert operator up to a certain point at the start, after which the insert operator manages to surpass it and maintains this lead as the fitness stabilizes.

To help validate the results, we implemented the AVSDF heuristic [9], and subjected it to the same three datasets and number of runs. The results obtained are described in Table 1, and they show that the solutions obtained are marginally worse compared to those obtained with the stochastic selection using the insert mutation operator. This allows us to conclude that the edge crossing reduction which resulted from using the AVSDF heuristic is not optimal and can still be improved. However, the reduction difference is not large, so it is necessary to verify whether there is an impact on the resulting visualizations.

Regarding the visual results, Figure 1 shows the visualizations generated from the best solutions for each dataset for both heuristics. The visualizations created for Sets 2 and 3 do not appear to have significant visual differences when compared between algorithms, as both seemed to have sorted the nodes in a way that they either are close to the ones they are related with, or that they present clear connections which are not obscured by visual complexity. However,

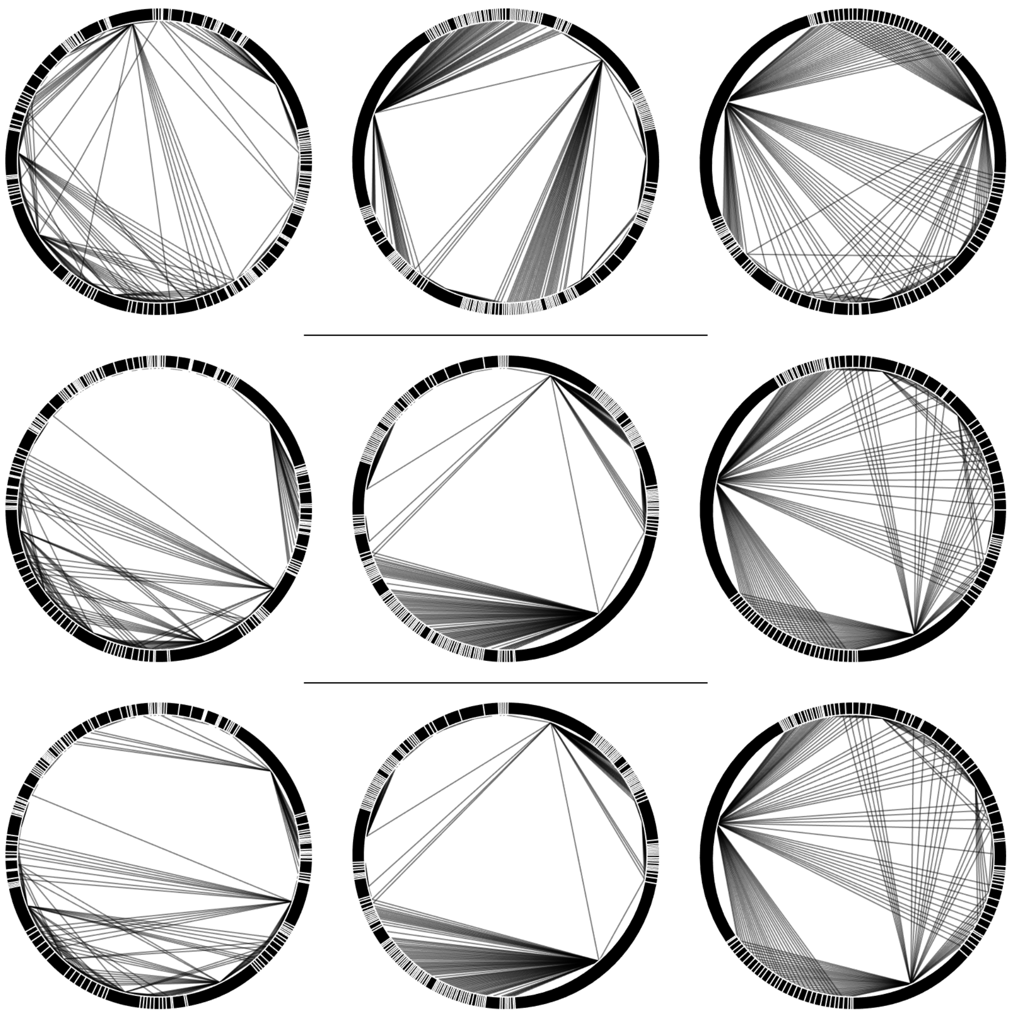

Figure 1: Circular graphs created using the best solutions obtained from both the developed GA using the insert mutation operator (top), the AVSDF heuristic (middle), and the evolved AVSDF solution (bottom), for Sets 1, 2 and 3 (left to right).

despite our evolutionary algorithm producing fewer edge crossings, the resulting graphs appear less organized by comparison, which can be seen primarily in the visualization created for Set 1 in Figure 1 (top row, left), where the edges appear to be more spread out. The reason for this is that the AVSDF heuristic results in node groups that have singular or very few connections close to each other, unlike the genetic algorithm. Having node groups with isolated relationships spread throughout the graph does not affect the number of edge crossings, but it does affect the comprehensibility of the visualization.

As a last step, the best AVSDF solution for each of the previous datasets was added to the initial population of each of the genetic algorithm's runs in order to demonstrate if it was possible to reach further using a combination of both heuristics. Our intent was to maintain the previously mentioned node organization of the AVSDF while allowing the evolutionary algorithm to additionally reduce the final amount of edge crossings. While Table 1 shows that the genetic algorithm was able to reduce the number of edge crossings, the evolution of the previous solutions has broken the order of the node groupings once again, which is most noticeable in the node order of the top section of the solution for Set 1, seen in Figure 1 (bottom row, left). This also happened to some degree in the visualization of Set 2, although it may not be immediately noticeable in Figure 1 (bottom row, center), where some nodes with singular connections between themselves have become lost within others nodes with many relationships. This, in conjunction with the previous tests, demonstrates that the use of a genetic algorithm may be effective on a technical level, as it is able to lower the number of edge crossings, but its impact on the visualization appears to be detrimental when compared to the AVSDF heuristic.

To sum up, in this case study we implemented an evolutionary approach for edge crossing reduction in circular graphs that was shown to be comparable or marginally superior to an existing non-evolutionary heuristic on a technical level. However, while the results did show that a technically-minded evolutionary approach was capable of reducing the amount of edge crossings further than the AVSDF algorithm, it was concluded that it reduced the comprehensibility of the resulting visualizations. The random search nature of the genetic algorithm dispersed the groups of nodes with very few relationships, which was noticeable when compared to the solutions obtained through the AVSDF heuristic, and again when the AVSDF solutions were evolved using the genetic algorithm. This may have been avoided if the visual aspect had been taken into consideration alongside the technical aspect, whereas the fitness function would have been designed to evaluate both the amount of edge crossing and the nodes' neighbourhoods. Evolving the solution towards maintaining groups of nodes with similar number of edges could have had reduced visual complexity in the resulting network visualization alongside the lower level of edge crossings.

3.2 Evolving Edge Bundling Parameters

Edge bundling methods are used in graph representation to reduce the visual clutter, which is unavoidable in complex and large networks. Generally, edge bundling methods consist of the geometrical distortion of edges to draw similar edges on the approximately same path, i.e., edges that are related in geometry are routed along the same path (see e.g. [11, 12, 14, 19]). Consequently, the visualization becomes less cluttered, and the information which is more relevant is revealed (e.g. main streams of flow).

However, the majority of these methods are still difficult to use and apply to real world problems by experts from other areas. This is due to the complexity of the algorithms and the concepts behind them, as well as a strong dependence on their parametrization. We proposed an experimental framework that helps finding near-optimal parameters for edge bundling algorithms, regardless of the configuration of the input graph. Our method is based on evolutionary computation, allowing the users to find edge bundling solutions according to their needs. We performed traditional experiments with automatic fitness functions, as well as with partially user-guided evolution, in order to understand the suitability of the evolutionary algorithm in such kinds of tasks. Additionally, we tested our approach in the optimization of the parameters of two different edge bundling algorithms: (i) Swarm-Based Edge Bundling (SBEB) [20]; (ii) Force-Directed Edge Bundling (FDEB) [12]. Results are compared using objective criteria, and a critical discussion of the graphical solutions is also conducted.

3.2.1 Evolutionary Edge Bundling. The majority of edge bundling algorithms depend on their parameters, which are usually found empirically by trial-and-error or statistically. In general, such parameters have to be fine tuned for different datasets, which depends on graph proprieties such as scale and topology. Additionally, there is no defined criteria that objectively measures the quality of edge bundling, which makes it hard to guarantee the effectiveness and efficiency of bundled drawings. On the one hand, we can measure the ratio between the ink and white space, or the edge curvature. On the other hand, we can measure the effectiveness by including the user in the evaluation cycle, where they are responsible for assessing the quality of the generated solutions.

Framework. With that said, we introduce a framework based on GAs. Individuals encode the edge bundling algorithm parameters, and are therefore represented as ordered linear sequences where each position has a value in the range allowed for that specific parameter. The proposed method automatically assists in finding parameters for given edge bundling algorithms that generate near-optimal solutions using the proposed metrics. These metrics consist of measuring the ink to white space ratio and the edge curvature of bundled drawings. By maximizing the white space and minimizing the edge curvature the gain is two fold: on one hand, the generated solutions are clutter free and space efficient; and on the other hand, solutions present smooth and easy-to-follow edges and flow streams.

To promote evolution and the exploration of the problem's domain we use uniform crossover and mutation. Regarding the mutation operator, we apply a per gene mutation to the candidate solutions, which allows the algorithm to change, from generation to generation, a percentage of the values to other valid ones.

Evolution is generational, which means that from one generation to the next we keep the offspring. Additionally, an elite of 1 (i.e., the best individual from the previous generation) is kept. However, the method that generates the graphical edge bundling results is not deterministic and, as such, can result in a loss in the population quality.

Fitness Function. Defining a metric that automatically assesses the quality of edge bundling algorithms is difficult. On the one hand, we can measure the ratio of the ink to white space [7], or the edge curvature in a bundled drawing. On the other hand, we can measure the effectiveness by performing a user-guided evolution, which is strongly dependent on the task and dataset at hand. Traditional approaches, such as the minimization of edge crossings, are not applicable because edge bundling favors such crossings. With that said, we consider experimenting with (i) the ink-white-space ratio, (ii) a combination of edge curvature and white space, and (iii) the user in the loop to assist the evaluation of bundled drawings. The first two measures are used as automatic fitness functions, and the latter one is partially interactive.

White space is a ratio that defines the percentage of white pixels in the drawing canvas. For a bundle to be considered good we want to maximize the white space so that the visual clutter is reduced, increasing the readability of the representation. That said, the fitness function consists of the number of white pixels in the canvas, which is then normalized using the maximum and minimum number of pixels. The maximum number of pixels is obtained by multiplying the width and height of the canvas. Since the canvas is never painted in total, the minimum value was empirically determined, in order to obtain values in the $[0, 1]$ interval. This is not relevant for the evolutionary process when used on its own, although it is crucial when combined with the curvature component. Otherwise, the white space component would not have enough impact for the final fitness value.

Curvature is a measure that comes from differential geometry and that describes a curve in the Euclidean space [21]. To compute the curvature, the edges (E) are defined as a sequence of vertices, $E = \{v_1, v_2, \ldots, v_n\}$. The curvature, $k_i = d\theta/dS$, is computed at each vertex v_i, starting at v_2 and finishing at v_{n-1}, where $d\theta$ is the angle between vectors $||v_{i-1} - v_i||$ and $||v_i - v_{i+1}||$, and dS is the Euclidean distance between points v_{i-1} and v_{i+1}. So, the final curvature is the average of all the curvatures of an edge, and is equal to $C = {}^{1}/_{(n-2)} \sum_{i=2}^{n-1} k_i$, where n is the number of vertices. In simple words, the curvature consists of the average of the angular velocities along the trace, and it is natural that a straight line has a curvature of zero. So, in our case the curvature should not be too high, since smooth curves are easier to follow visually. Therefore, we want to minimize the curvature, but retain some degree of curviness. Finally, the total curvature of a bundled graph is the sum of all the edge curvatures, and since the number of edges is the same within a dataset it is fair to compare different bundled drawings.

Both components are used to enhance the results obtained by the evolutionary framework. By minimizing the curvature, the algorithm tends to straighten lines and the space becomes more filled with ink, therefore reducing white space. As such, by maximizing the white space and minimizing the curvature, we expect to achieve a balanced bundled drawing, where the edges are smooth enough to be easy to follow, and the ink to white space ratio is low, so that the visualization is clean and easy to understand. For that, we define a fitness function in the form of $fitness\_both = W/(1 + C)$, where W is white space and C is curvature. We add 1 to the curvature, because the values of C can be in the $[0, 1]$ interval. The GA algorithm is guided towards the maximization of the fitness function.

3.2.2 Experimental Setup and Preliminary Conclusions. The experimentation is divided in two independent sub-experiments: (i) automatic and (ii) user-guided evolutions.

Automatic Evolution. Figure 2 depicts the evolution of the best individuals across 100 generations. Results are averages of 30 independent runs for each experiment and each chart component is normalized by the maximum value found for that component throughout all performed experiments. In all charts it is possible to conclude that the algorithm is able to promote evolution, reaching areas of the search space that contain desired solutions. More precisely, in the experiment where the white space is used to guide the evolution of the SBEB algorithm, the convergence is faster than when combining white space with curvature. This result is expected, because when promoting evolution towards multiple objectives the search space is greater and less flat.

Focusing on the best solutions found with both approaches when the evolution is guided by the white space only (first two charts), it is observable that in both of them the evolutionary process takes around 40 generations to converge. Moreover, the attained solutions are of similar quality (in terms of fitness), although those that resulted from SBEB are slightly better than those obtained by FDEB. When promoting the evolution of the SBEB using both fitness components (last chart) the evolutionary process seems to still be evolving at a slow pace. Additionally, the white space appears to be the fitness component that most contributes to evolution, as it is the one that best accompanies fitness evolution, with less erratic behaviour. That is, the curvature is less stable than the white space during the evolution process, which may indicate that curvature is an objective that is harder to achieve.

Regarding the evolved parameters for the edge bundling algorithms we have found that, in general, these yield efficient, easy to read solutions. Figure 3 displays the best and worst individuals for each experiment using the automatic fitness function. The very first observation shows that the best solutions (top row) are distinct from the worst ones (bottom row). In general, the worst solutions result in straight lines (first image of the bottom row) or hairy drawings (last two images of the bottom row). In what regards spacial efficiency, the best results are clean and clutter-free. The main streams of flow are distinguishable and well defined. However, in the case of SBEB with the composite fitness (last image in the top row) the result seems to be the most cluttered among the three solutions. This is because the curvature fitness function pushes the process to solutions with straight lines, and thus, more filled drawings.

In contrast, using SBEB solely with white space fitness (second image in the top row) tends to result in schematic-looking representations, favoring abrupt changes in trace directions and fewer main streams. This can be disadvantageous in cases where large complex graphs are to be visualized, since it is harder to follow paths with many changes in direction. Finally, the application of the composite fitness to SBEB results in drawings with smoother bundles and shorter branches, but generates more main streams in comparison with the other approaches. Ultimately, we have observed that it is sufficient to use only white space fitness with edge bundling algorithms that generate smooth bundles naturally, like FDEB (first image in the top row). The GA finds near-optimal solutions for spacial efficiency, while the FDEB algorithm ensures smoothness of the lines.

User-Guided Evolution. Another aspect of this work is the fact that we added users in the evolutionary process, so that it is also guided according to their preferences. This way, the evaluation cycle is divided into two steps. First, all the individuals are evaluated using one of the automatic fitness functions. Then, the user is given 12 solutions to evaluate: the 7 best solutions of the population, 3 from the middle, and the 2 worst solutions. The score that is given by the user lies in the range $[1, 5]$. These evaluations are then normalized and combined with automatic fitness, weighting both components. In order to give higher impact to the user evaluations and less to automatic fitness the components are weighted 80% and 20%, respectively.

Finally, the user evaluation is performed through a graphical interface, in which the twelve individuals are presented in a 4×3 grid, and they are evaluated using five radio buttons on top of each picture. By pressing "Next generation" the evolutionary process proceeds to the next generation. The process ends when the user is satisfied with the obtained results.

3.2.3 User Testing Results. Ten users with diverse backgrounds and areas of specialization participated in the test. Their expertise in information visualization field varies from 2 (passing knowledge) to 5 (expert), and the majority are 3 (knowledgeable). Table 2 summarizes the results of the user testing experiments. The values presented in the table are averages (μ) and standard deviations (σ) of ten independent testings, for each of the measured items (described above).

Starting with the task execution time, the results indicate that through the direct manipulation of parameters the users find the solution in half of the time of the user-guided system. Since the parametric search happens in real time the users find a solution in shorter time. Also, having no reference image the choices are based on previous solutions, while using the user-guided system the users take more time to think, since they need to compare 12 images in one visual instance. Also, it is important to note that through direct manipulation all the users conclude the task in approximately the

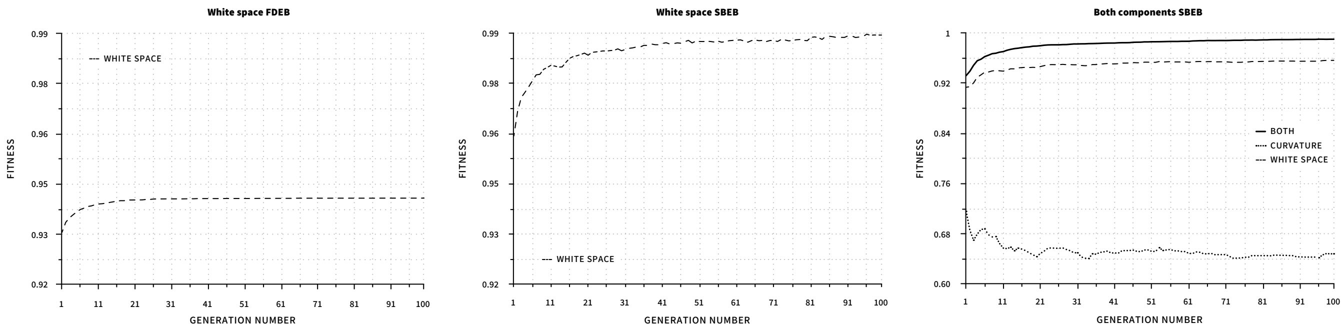

Figure 2: From left to right, evolution of the fitness of the best individuals for the FDEB with white space, SBEB with white space and SBEB with both components. The values of each component are normalized.

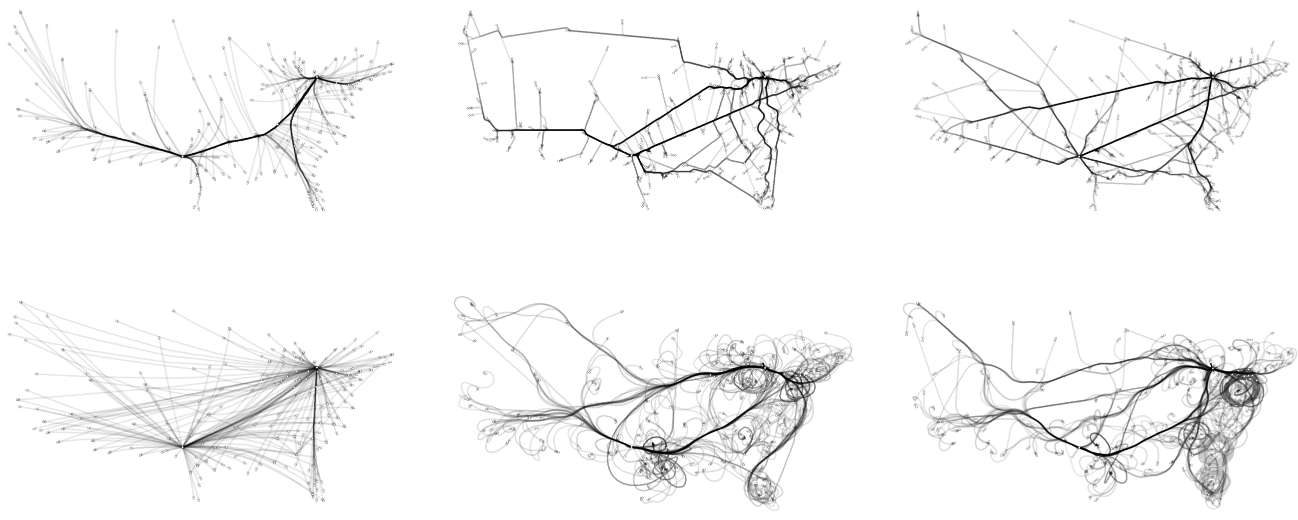

Figure 3: Best (top row) and worst (bottom row) solutions at the end of each experiment. From left to right – results for FDEB with white space, SBEB with white space and SBEB with both components.

Table 2: User testing mean (μ) and standard deviation (σ) results. DM and UG correspond to direct manipulation and user-guided, respectively. Time values are in minutes and seconds, and other values are in the range $[1, 5]$.

	DM (μ)	DM (σ)	UG (μ)	UG (σ)
Useful Tm.	6m 13s	2m 36s	12m 47s	9m 47s
Total Tm.	6m 14s	3m 2s	15m 47s	10m 47s
Accuracy	3.61	0.87	4.0	0.7
Preference	2.7	0.9	4.8	0.4

same time, which is perceptible through a low standard deviation. In contrast, the duration of the task performance using the user-guided system varies drastically. Finally, the useful time for interaction with the user-guided system is smaller than the total time, since the system takes some time to generate solutions, which depend on the edge bundling algorithm's performance.

Regarding the accuracy of the found solutions, the user-guided system performs slightly better than direct manipulation, and the output from the user-guided system is closer to the average in comparison to those that come from direct manipulation, which result in more disperse solutions. Moreover, the user-guided system outperforms the direct manipulation in terms of user preference. The majority of the users find the user-guided system encouraging for exploration, and easier to use and understand. However, user feedback indicates that the user-guided system should preserve at least one evaluated solution in the next generation, or provide an archive, so the user can compare the current solutions with the ones they preferred.

Our observations indicate that in general the users do not prefer the best solution according to the automatic fitness function component. Curiously, they opt for the solutions that have the poorest automatic fitness. In particular, users prioritize the solutions that generate unwanted visual artifacts, such as curly endings, spirals, etc. Regarding user interaction with the interactive-guided system, some users evaluate all the solutions, and other evaluate only a few, they like most. Also, we observe that after finding a solution some users continue exploring the system, justifying it by curiosity of what may come next.

4 DISCUSSION AND CONCLUSIONS

While current research for graph drawings still prioritizes technical aspects, aesthetics play a major role in the creation of clear and effective visualizations. GAs have been used across diverse fields, including information visualization, to solve problems that deal with complex data. By taking into account how graphical representations are perceived, techniques such as edge crossing minimization and edge bundling can be evolved to further reduce visual clutter and emphasize meaningful relationships.

To ascertain the importance of aesthetics and the potential role of evolutionary algorithms we presented two case studies: an evolutionary approach to edge crossing reduction on circular layouts, and a user-guided evolutionary approach for edge bundling parametrization. The first case study describes a GA designed to re-order nodes on a circular layout in order to minimize the number of edge intersections, with the goal of improving the visualization's legibility. The second experiment consists of evolving parameters of an edge bundling algorithm to find near-optimal bundled graph drawings. In order to measure the quality of bundles we use the white space to ink ratio, curvature and combination of both. Furthermore, we resort to interactive evolution, evolving the user in the assessment of the bundles.

The analysis of literature and our experimentation indicate that aesthetics should be recognized as one of the key issues, as well as a niche in designing and applying genetic algorithms in modern information visualization. The first experimentation demonstrates that the GA was capable of surpassing a non-evolutionary heuristic on a technical level. However, the visual groupings present in the non-evolutionary solutions were lost. Thus, the minimization of edge crossing should not always be taken axiomatically, since it may lead to a decrease of legibility. It is therefore necessary to consider higher order principles, that assess the how the artifact is perceived, when assigning fitness. Although taking into account perceptual principles may hinder the minimization of edge crossings, the clarity of the visualization should be prioritized.

Similarly, the second case study shows that the results obtained by automatic evolution are near-optimal, at least in terms of spacial efficiency. However, in some of the cases, the excessive overlap of the bundles decreases legibility. Furthermore, the trade-off between aesthetics and functionally should be taken into account when involving user in the loop. A fitness function should guarantee the correctness of representation, while leaving enough room for user's preferences.

The results presented herein indicate that future research is necessary in order to develop and incorporate new measurements for edge bundling quality, both in terms of technical characteristics and in terms of aesthetic and perceptual principles.

ACKNOWLEDGMENTS

This project is partially founded by Fundação para a Ciência e Tecnologia (FCT), Portugal, under the grants SFRH/BD/109745/2015 and SFRH/BD/124538/2016.

REFERENCES

[1] Michael H. Albert, Mike D. Atkinson, Doron Nussbaum, Jörg-Rüdiger Sack, and Nicola Santoro. 2007. On the longest increasing subsequence of a circular list. *Inform. Process. Lett.* 101, 2 (2007), 55–59.

[2] Michael Baur and Ulrik Brandes. 2005. *Crossing Reduction in Circular Layouts.* Springer Berlin Heidelberg, Berlin, Heidelberg, 332–343.

[3] Michelle A. Borkin, Azalea A. Vo, Zoya Bylinskii, Phillip Isola, Shashank Sunkavalli, Aude Oliva, and Hanspeter Pfister. 2013. What Makes a Visualization Memorable? *IEEE Trans. Vis. Comput. Graph.* 19, 12 (2013), 2306–2315.

[4] Jürgen Branke, Frank Bucher, and Hartmut Schmeck. 1996. Using genetic algorithms for drawing undirected graphs. In *The Third Nordic Workshop on Genetic Algorithms and their Applications.* Citeseer, 193–206.

[5] Nick Cawthon and Andrew Vande Moere. 2007. The Effect of Aesthetic on the Usability of Data Visualization. In *11th International Conference on Information Visualisation, IV 2007, 2-6 July 2007, Zürich, Switzerland.* 637–648.

[6] Geoffrey M Draper, Yarden Livnat, and Richard F Riesenfeld. 2009. A survey of radial methods for information visualization. *IEEE transactions on visualization and computer graphics* 15, 5 (2009), 759–776.

[7] Emden R. Gansner and Yehuda Koren. 2006. Improved Circular Layouts. In *Graph Drawing, 14th International Symposium, GD 2006, Karlsruhe, Germany, September 18-20, 2006. Revised Papers.* 386–398.

[8] Farshad Ghassemi Toosi, Nikola S Nikolov, and Malachy Eaton. 2016. A GA-Inspired Approach to the Reduction of Edge Crossings in Force-Directed Layouts. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion.* ACM, 89–90.

[9] Hongmei He and Ondrej Sykora. 2004. New circular drawing algorithms. In *Workshop on Information Technologies-Applications and Theory, Slovakia.*

[10] John H Holland. 1992. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* MIT press.

[11] Danny Holten. 2006. Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data. *IEEE Trans. Vis. Comput. Graph.* 12, 5 (2006), 741–748.

[12] Danny Holten and Jarke J. van Wijk. 2009. Force-Directed Edge Bundling for Graph Visualization. *Comput. Graph. Forum* 28, 3 (2009), 983–990.

[13] Donald H. House, Alethea S. Bair, and Colin Ware. 2006. An approach to the perceptual optimization of complex visualizations. *IEEE Transactions on Visualization and Computer Graphics* 12, 4 (2006), 509–521.

[14] Christophe Hurter, Ozan Ersoy, and Alexandru Telea. 2012. Graph Bundling by Kernel Density Estimation. *Comput. Graph. Forum* 31, 3 (2012), 865–874.

[15] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. 1999. Data clustering: a review. *ACM computing surveys (CSUR)* 31, 3 (1999), 264–323.

[16] Robert Kosara. 2007. Visualization Criticism - The Missing Link Between Information Visualization and Art. In *11th International Conference on Information Visualisation, IV 2007, 2-6 July 2007, Zürich, Switzerland.* 631–636.

[17] Andrea Lau and Andrew Vande Moere. 2007. Towards a Model of Information Aesthetics in Information Visualization. In *11th International Conference on Information Visualisation, IV 2007, 2-6 July 2007, Zürich, Switzerland.* 87–92.

[18] Erkki Mäkinen. 1988. On circular layoutsfk. *International Journal of Computer Mathematics* 24, 1 (1988), 29–37.

[19] Vsevolod Peysakhovich, Christophe Hurter, and Alexandru Telea. 2015. Attribute-driven edge bundling for general graphs with applications in trail analysis. In *2015 IEEE Pacific Visualization Symposium, PacificVis 2015, Hangzhou, China, April 14-17, 2015.* 39–46.

[20] Evgheni Polisciuc, Pedro Cruz, Hugo Amaro, Catarina Maçãs, and Penousal Machado. 2016. Flow Map of Products Transported among Warehouses and Supermarkets. In *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2016) - Volume 2: IVAPP, Rome, Italy, February 27-29, 2016.* 179–188.

[21] Andrew Pressley. 2010. *Elementary Differential Geometry.* Springer London.

[22] Carolyn Salimun, Helen C. Purchase, David R. Simmons, and Stephen A. Brewster. 2010. The effect of aesthetically pleasing composition on visual search performance. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction 2010, Reykjavik, Iceland, October 16-20, 2010.* 422–431.

[23] Janet M Six and Ioannis G Tollis. 1999. Circular drawings of biconnected graphs. In *Workshop on Algorithm Engineering and Experimentation.* Springer, 57–73.

[24] J Utech, J Branke, H Schmeck, and P Eades. 1998. An evolutionary algorithm for drawing directed graphs. In *Proc. of the Int. Conf. on Imaging Science, Systems and Technology.* Las Vegas, Nevada, USA, 154–160.

[25] Colin Ware, Helen Purchase, Linda Colpoys, and Matthew McGill. 2002. Cognitive Measurements of Graph Aesthetics. *Information Visualization* 1, 2 (June 2002), 103–110.

[26] Hsiang-Yun Wu, Shigeo Takahashi, Chun-Cheng Lin, and Hsu-Chun Yen. 2011. A zone-based approach for placing annotation labels on metro maps. In *International Symposium on Smart Graphics.* Springer, 91–102.