Computational Complexity of Edge Bundling Problems

J.M Ferreira *H.A.D. Nascimento*
Aaron Quingley *Les Foulds*

Technical Report - RT-INF_001-17 - Relatório Técnico
November - 2017 - Novembro

Instituto de Informática
Universidade Federal de Goiás
www.inf.ufg.br

Computational Complexity of Edge Bundling Problems

Joelma de Moura Ferreira [*]
cedric@inf.ufg.br

Hugo A. D. do Nascimento [†]
hadn@inf.ufg.br

Aaron Quigley [‡]
longo@inf.ufg.br

Les Foulds [§]
lesfoulds@inf.ufg.br

Abstract. *Edge bundling is a technique to visually group, align, coordinate and position the depiction of edges in a node-link graph drawing, so that sets of edges appear to be brought together into shared visual structures, i.e. bundles. The ultimate goal is to visually simplify the drawing to improve how it conveys information. Edge bundling has been extensively studied and can successfully reduce visual clutter. However, there is a lack of fundamental theoretical principles that can be used to objectively measure the effectiveness of bundling techniques and to compare various proposed algorithms. In addition, the computational complexity of this type of problem has not been fully studied to date. In this paper, we provide a general formulation for the edge bundling problem, as a formal combinatorial optimization problem. This allows for the definition and conceptual comparison of various types of edge bundling techniques. In addition, we define a simplified edge bundling problem and prove its NP-completeness. We also propose an enhanced edge bundling problem with maximum angle constraint and propose an integer linear programming model for it.*

Keywords: Edge Bundling, Computational Complexity, Graph Drawing, Combinatorial Optimization

Graphs are often drawn as node-link diagrams. However, as the number of vertices and edges grow, the amount of information increases and the ability to effectively and efficiently display graphs with such diagrams becomes challenging. The standard approaches for drawing an edge as a straight-line, polyline or a curve can result in significant visual clutter. This is due to excessive edge crossings and the over-plotting of edges and nodes in the drawing, resulting in a loss of readability [15]. When very large graphs are drawn digitally, the problem of readability is exacerbated by limited screen size and resolution which often tax human cognitive ability [28].

To reduce visual clutter, several approaches have been proposed such as changing the visual representation (e.g. adopting a hybrid representation [7]), rearranging the position of the node-link graph elements [13], filtering the data [27] or clustering the graph structure [26]. Even

[*]Institute of Informatics, Federal University of Goiás, Brazil.
[†]Institute of Informatics, Federal University of Goiás, Brazil.
[‡]SACHI, School of Computer Science, University of St. Andrews, United Kingdom.
[§]Institute of Informatics, Federal University of Goiás, Brazil.

though these approaches can formally simplify a graph drawing, the simplification obtained may not be suitable for certain tasks, because, in general, they change the original graph structure, or abstract and hide parts of it from the observer. Such changes, while visually simplifying the drawing, can diminish how well it conveys information.

It was in this context where the concept of edge bundling first appeared with techniques focused on moving the drawing of graph edges together, as a way to reduce visual clutter and thus, improve the legibility of a graph drawing [8, 25]. As some pairs of edges may not be suitable for bundling, later compatibility metrics were introduced in order to decide which edges should be joined together [4, 11].

New edge bundling methods have emerged over the past decade. Today, there are a number of approaches that can successfully reduce clutter [2, 4, 8, 9]. Beyond visual inspection, comparing the resultant bundled graph of different bundling approaches is a hard task [29]. This is because edge bundling methods use different quality measures (e.g. ratio of ink, edge curvature or sharpness of the edge density map) and are based on distinct graph properties and compatibility metrics. For example, Hurter *et al.* [10] employ an edge density map to emphasize strong concentrations of edges. Such subtle differences in approaches can give rise to significant differences in the impact on the visual attributes used and hence the comparisons which are possible.

There is a multitude of diverse, and often complex, reported edge bundling algorithms, making it desirable to introduce a formal framework to help to advance work in this area. There have been some attempts to formalize the presentation of certain edge bundling methods (e.g. kernel density estimation [11] and image-based methods [30]) or to study them with new measures (e.g. faithfulness [18]), but bundling itself, as a technique, lacks an underlying formalism. There are a number of existing edge bundling problem definitions, each related to a slightly different aspect of the problem. Most of these definitions are vague, in the sense that they are not based on a detailed mathematical formulation. Moreover, to the extent of our knowledge, there is no generic, formal definition that precisely defines edge bundling. Furhermore, a complexity analysis of the subjacent computational problem has not yet been reported.

Given these gaps in the literature, on formally studying bundling techniques, this paper addresses the issues by posing the edge bundling problem as a combinatorial optimization problem and analizing its complexity, so that:

(i) different bundling problems can be compared based on their inherent objective functions and constraints;

(ii) the computational complexity of each specific problem can be investigated from a theoretical point of view; and

(iii) as a consequence, distinct algorithms for the same problem can be compared, based on how effectively or efficiently they solve the given optimization problem.

Another important aspect is that most edge bundling methods combine edges that have distinct endpoints, thus producing bundles with various source or target vertices. The general formulation presented in this paper covers this typical case. Nevertheless, we propose a specialization of the concept of edge bundling as a particular combinatorial optimization problem for edges that have the same source or the same target node (i.e. only adjacent edges are bundled). This is important in order to show that even simple edge bundling problems may be NP-hard. In addition, the specialization does have practical applications such as visualizing flow maps [23] and network data [19] and, more recently, for solving edge ambiguity cases [16].

Following this concept, and using the general edge bundling optimization framework, this paper then presents and analyzes two edge bundling problems, which we refer to as $EB{-}star$ and $EB{-}star_\alpha$. The first is a simplified specialization of the general framework by allowing the bundling of only edges with the same start/end-node, while the second includes a maximum angle that constrains which pairs of adjacent edges can be bundled.

The remainder of this paper is organized as follows: Section 1 provides a brief overview of the existing edge bundling definitions and computational methods. Section 2 introduces a general formulation that helps to define and conceptually compare several edge bundling problems as combinatorial optimization problems. Section 3 illustrates the use of the formulation, presents the edge bundling problem for adjacent edges called $EB{-}star$ and proves its NP-completeness. The more complex edge bundling problem with an angle constraint, $EB{-}star_\alpha$, is introduced and discussed in Section 4. The Section 5 presents an integer linear programming model of $EB{-}star_\alpha$. Finally, Section 6 draws some conclusions and proposes ideas for future work.

1 Background

We now provide a chronological overview of edge bundling definitions from the earliest work to the present date. It includes the primary work in the area, quoting the original definitions and concepts related to bundling where useful. The intention is not to provide an exhaustive survey of the topic. Instead, it is to demonstrate the lack of a formal definition for what "edge bundling" means. Additional related work is introduced later in the paper, in the context of the specific problem of bundling only adjacent edges.

Edge concentration [17], confluent graph drawing [3] and flow map layouts [23] are considered the earliest techniques related to edge bundling.

Edge concentration was introduced by Newbery in 1989 [17] and was applied to bipartite graphs. It consists of replacing complete bipartite subgraphs by star-like structures, with the goal of minimizing the number of edges in the resultant graph, while retaining the perceived graph structural information. The problem was formally defined as a mathematical combinatorial problem.

Confluent graph drawings were first introduced in 2003 by Dickerson *et al.* [3]. Techniques to generate confluent graphs produce visualizations of non-planar graphs in a planar way. The edges are routed and drawn together into "tracks" [9].

In the work on flow map layout [23], published in 2005, Phan *et al.* presented a flow map layout technique in which edges are routed along the hierarchy of the graph. In this approach a flow tree is created using a binary hierarchical clustering algorithm that has the objective of minimizing the number of edge crossings.

The seminal work of Holten in 2006 [8] evolved the previous ideas and introduced the term edge bundling. This work states that "hierarchical edge bundling is based on the principle of visually bundling adjacent edges together analogous to the way electrical wires and network cables are merged into bundles along their joint paths and fanned out again at the end, in order to make an otherwise tangled web of wires and cables more manageable". The use of a bundling strength parameter and sequence of ordering, blending and rendering steps all induce the perceived visual bundling.

In 2007, in a work on improved circular layouts, Gansner and Koren [5] presented an algorithm that uses a circular layout to decide which sets of edges would be bundled together. The authors alluded to the geometric characteristics of a bundle to reduce density by searching for the best placement of the bundles and other visual characteristics when using a uniquely colored bundle to increase the readability of the graph.

In 2008, in a work on geometry-based edge clustering, Cui *et al.* [2] described bundles as the "grouping of spatially close edges with similar directions" based on a control mesh where common segments in the path of edges are bundled.

Telea and Ersoy in 2010 [30] incorporate a metric d in their definition of edge bundling. The authors state that edge bundling layouts actually spatially group edges by using their metric to estimate pairwise edge closeness in either graph theoretic space, layout space, or both.

In the following year, Ersoy *et al.* [4] extended their own definition of edge bundling to further consider data attributes, not just the graph theoretic or geometric properties. They describe bundling as "edges found to be close in terms of graph structure, geometric position of their endpoints, data attributes, or combinations thereof, are drawn as tightly bundled curves".

In the same year, Čermák, Dokulil and Katreniaková [31] treated edge bundling as "...polylines that consist of one or more connected segments". The authors indicated that "a bundle is a set of polylines that all share one or more segments, i.e., there is a segment s such that each polyline in the bundle contains a segment whose endpoints have exactly the same coordinates as s or s reversed (start- and end-points interchanged)".

Again in 2011, Gansner *et al.* [6] proposed an edge bundling algorithm guided by the principle of saving "ink" in drawings. This concept is important as it raises bundling beyond simply reducing visual clutter, even though as a result it creates significantly reduced cluttering. The authors also provide an informal definition of edge bundling by stating that "compatible edges are [...] combined in a single bundle, sharing part of their routes".

In 2012, Pupyrev *et al.* [24], who studied edge routing with ordered bundles, defined edge bundling as a technique in which "...some edge segments running close are collapsed into bundles to reduce the clutter [...] A set of paths sharing the same edge of G is called a bundle".

In the same year, Hurter, Ersoy and Telea [11] provided a mathematical definition of edge bundling on graph bundling using the concept of kernel density estimation. They stated that "given a graph drawing $G \subset \mathbb{R}^2$ and a point $x \in G$, we can think of bundling as an operator $B : \mathbb{R}^2 \to \mathbb{R}^2$ which displaces x based on the spatial information in $G \cap v_\varepsilon(x)$ where $v_\varepsilon(x)$ is a small neighborhood centered at x. The result $B(G)$ is a new layout whose edges are gathered in dense groups (bundles) separated by low edge density areas (inter-bundle white space) to minimize drawing ink".

Nocaj and Brandes in 2013 [19] presented a new edge bundling method that only bundles edges at their ends, that is, edges that share an endpoint. In addition, they satisfy some maximum edge angle constraints for the bundles, so that only edges roughly in the same direction are merged together. The problem, entitled *stub bundling*, was applied to geographic networks generating confluent spiral drawings.

In the same year, Nguyen, Eades and Hong [18] presented a more mathematical definition of an edge bundling process by stating "...given an input graph $G = (N, E)$, an edge bundling visualization process V partitions E into bundles $E = B_1, B_2, ..., B_k$. Let G_i denote the subgraph of G with edge set B_i and node set N_i consisting of endpoints of edges in B_i. Edge bundling methods ensure that G_i is bipartite".

In 2015, following the discussion initiated by Ersoy *et al.* [4], Peysakhovich, Hurter and Telea [22] extended the definition of edge bundling to incorporate edge properties, such as direction and data attributes. The authors state that, "graph bundling methods attempt to reduce clutter by grouping edges found to be compatible into so-called bundles. Yet, few such methods handle attributed graphs; edge compatibility is mainly based on spatial position and does not use attributes such as edge direction or data values".

Finally, in the work of Zielasko [32] on interactive 3D force-directed edge bundling, the author affirmed the simplicity of the concept of edge bundling by stating that "edge bundling is a method that combines geometrically close edges into bundles, which use much less screen space".

From the definitions above, it is clear that edge bundling is generally stated as a technique that merges or groups "compatible" edges (geometrically close) into so-called bundles. The geometric compatibility measures usually adopted are: angle, scale, visibility and position. Vertices are usually kept fixed, with coordinates defined prior to the bundling process. Edges, on the other hand, can be replaced by deformable curves drawn close together initially along a shared path and which then fan in/out when closer to their destinations.

From the above perspectives, the overall goal of edge bundling methods is been broadly the same: to reduce visual clutter by improving the legibility of dense graph drawings, differing in the way of doing it. In addition, it is noticeable that there is no universal agreement on a precise mathematical formulation of edge bundling that can take into account all the current approaches. Also, we are not aware of a concise account of complexity analysis of edge bundling problems.

Besides the various above mentioned definitions, several edge bundling methods for large graph visualization have been proposed. Examples include hierarchical-based edge bundling [8] that merges edges by routing them along the layout; force-based edge bundling [9] that merges edges by using force-directed interaction and attracting control points on edges close to each other; geometric-based edge bundling [2] that employs a control mesh to route curved edges; image-based edge bundling [11] which is based on calculations of the kernel density estimation; and skeleton-based edge bundling [4] that uses skeletonization to construct bundled layouts. Each method is based on an implicit definition of what an edge bundle is.

As a result of the diversity of methods and definitions, it is difficult to compare edge bundling approaches, since their underlying optimization problems vary significantly.

As far as we know, only two studies have explored the complexity of a related to edge bundling problem. Newbery [17] conjectured that the *edge concentration* problem was NP-complete, and Lin [14] provided the proof of this. The optimization version of the edge concentration problem involves minimizing the number of edges of a bipartite graph by compressing some specific edges (the ones belonging to the same complete bipartite subgraph) into simpler structures. This naturally resembles the edge bundling problem.

The next section presents a general formulation of a particular edge bundling problem as a combinatorial optimization problem. This new formulation helps to overcome the lack of theoretical tools in the edge bundling research area.

2 A general formulation

In an edge bundling process, how to choose the best bundle configuration depends on a variety of aspects such as the current layout, similarity between edges, routing decisions, control points of the spline curves, ambiguity introduced, level of clutter, etc. For example, the placement of routed edges can be affected by the control points assigned to a spline curve. To analyze the effectiveness of an edge bundling method, all aspects of interest should be considered. If, for some aspects, many possible configurations (or solutions) are allowed, then an optimization problem becomes evident.

Most edge bundling methods search for a good solution heuristically with an informal and/or implicit definition of their optimization goals. We embrace a different direction, by providing a general mathematical formulation that can be used to define edge bundling problems as combinatorial optimization problems. This affords us new ways of studying edge bundling, with different tools and techniques to help understand it. Specifically, it allows the formal definition of edge bundling problems, including their computational complexity analysis, and also helps us to propose and compare the effectiveness of different algorithmic approaches. The proposed formulation follows.

Definição 2.1 *Let $G = (V, E)$ be a graph, D an unbundled node-link drawing of G in the plane, and $S = (E_1, E_2, ..., E_n)$ a partition of E (not necessarily disjoint), $n \in \mathbb{N}^+$. Further, let $R()$ be a function that takes G, D and S, and renders a bundled graph drawing version of D called D^r, given some extra necessary information, such as rules for routing the edges. The general edge bundling problem (EB) is hence to determine the partition S (here called bundles), with $E = \cup_{i=1}^{n} E_i$, so that:*

- *a set F of objective functions, representing aesthetic bundling measurements, are optimized (minimized or maximized), and*
- *a set P of constraints, defining what can be bundled and how they are satisfied.*

The functions in F and the constraints in P can be defined based on G and D, as well as on the dynamic subsets $E_i \in S$ being computed. The generation of a bundled drawing D^r is made through $R()$. In D^r, edges of G that originally were drawn as independent lines in D are now bundled together (that is, routed along a shared path) if they belong to the same subset E_i. Figure 1 shows the framework for defining and solving edge bundling problems using the EB formulation.

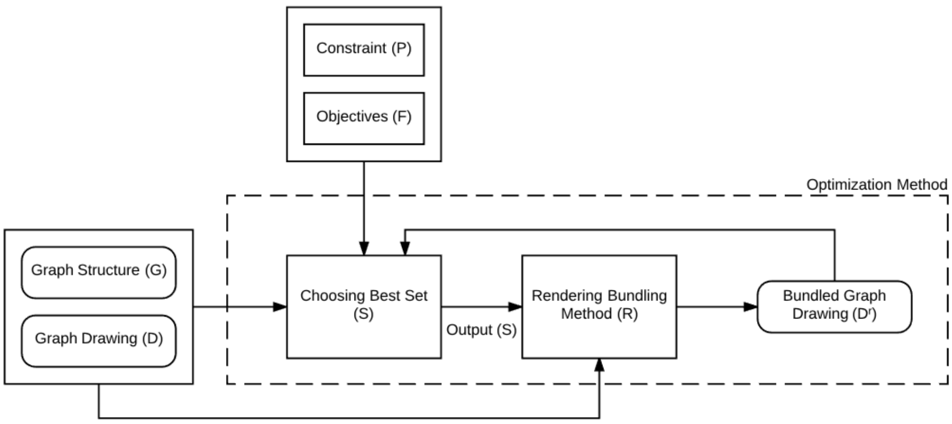


Figura 1: General Edge Bundling Optimization Framework.

The decision variables of the optimization problem are restricted to the choice of how to partition E into S. That is, the edge bundling problem is primarily to find the subsets $S = E_1$, $E_2, \ldots E_n$ so that the corresponding drawing D^r created by R is optimal with respect to F and P.

Clearly, EB is not an actual edge bundling problem, but a template for specifying such types of problems. Therefore, for defining a real edge bundling problem, EB has to be rewritten with a more precise definition of G, D, R, P and F. For example, G can be an undirected or directed graph, D might or might not be a node-link representation with straight-line drawings for the edges, and R can be a rendering method that takes the edge positions from D and the set S and uses them to draw bundles with splines routed along evenly spaced control points. The set P might include one or more compatibility constraints, such as a similarity constraint or an ambiguity constraint that forbids any bundle to be routed near its unrelated nodes. This condition might also be a local constraint embedded in $R()$, but we may wish to have $R()$ as a more general function and insert the constraint in P. In addition, the set F might be associated with any visual aspect of the resultant graph drawing produced by R, for example, minimizing the amount of ink for drawing the bundles or minimizing the number of edge crossings in D^r. Also, F might have more than one objective function.

Next, we show how some of the main existing edge bundling approaches can be expressed using the EB formulation.

- **Hierarchical-based edge bundling:** Edges are routed along the shortest paths in the grid over an hierarchy, for graphs that have an hierarchical structure. For graphs without an associated pre-defined hierarchy, a structure of the tree is used to join the edges. In this case, the community structure of nodes is extracted and organized into a community hierarchy, using affinity metrics to determine the similarity between any two nodes. (Affinity metrics are measures that can be used for computing the affinity between nodes: Jaccard's coefficient, Pearson's coefficient, Euclidean distance, cosine distance, etc.) In an optimization hierarchical bundling approach the affinity metrics can be maximized or minimized as the goal in F. The set P contains properties that have to be satisfied for the control hierarchy to be defined, such as the distance between linked nodes or bundling strength.

- **Force-directed edge bundling:** A self-organizing approach to bundling edges modeled as flexible springs. A physical system is modeled using attracting spring force and electrostatic force. Metrics measure the compatibility between edges. In the optimization model of this approach, the objective function in F calculates the amount of energy related to the spring forces and electrostatic forces of the modeled force-directed system. The set P defines edge compatibility constraints regarding scale, position, visibility and angle measures.

- **Geometric-based edge bundling:** An approach that employs a control mesh, generated based on the distribution pattern of nodes and edges, to route curved edges and form the bundles. Triangulation is usually used, and to maximize the quality of this, and the length and direction of the edges could be considered in an optimization approach. An optimization model of geometric bundling could have an objective function in F that computes the optimal triangulation, for example, with minimum total edge length, or average distance between the edges inside the bundles generated. The set P contains rules to define the kind of grid used to guide the edge-bundling process and adjustments for merging the control points for bundling.

The EB formulation, however, has limitations as it does not cover all possible edge bundling problems. In particular, it does not consider cases where another bundling aspect has to be treated as a decision variable and optimized at the same level of importance as the choice of the set S. To exemplify our argument, one such situation is to choose the position of control points to route the bundles simultaneously along with the specification of S, as equally important variables. If changing a routing may imply reorganizing the partition of E in S, then EB does not take this factor into account. Recall that R, in the current formulation, is a given function; so, edge bundling routing and other visual aspects of the drawing will be embedded in R and influenced by S, and not the other way around.

In order to overcome such a limitation, the EB formulation could be extended to include other types of decision variables. Nevertheless, as the current formulation is general enough to express several edge bundling problems, we adopt it in the remainder of this paper.

3 The star-edge bundling problem

An interesting problem that can be obtained from a simple specialization of *EB* has received little attention in the edge bundling literature. It consists of bundling only edges linked to a same end-point. We first review previously reported work on this particular bundling case, and then present a specialization of *EB* for it.

3.1 Adjacent edge bundles

In general, traditional edge bundling methods join edges with different endpoints, generating bundles that have a common interior part but that link multiple source and destination nodes [19]. However, this is not the only possible configuration. Another approach is to join only adjacent edges, i.e., edges that share a same endpoint (source or destination node).

Figure 2 (a) illustrates an example of a bundle with three sources and three destinations; Figure 2 (b) shows a bundle with one common node as either source or destination.

(a) (b)

Figura 2: Type of bundles: (a) multiple source and destination nodes; (b) one common ending node.

One of the problems of traditional edge bundling methods is that the identification of source and target nodes are obscured, decreasing the traceability of the edges. Joining only adjacent edges is likely to be a more efficient approach when it is important to identify which pairs of node are actually connected. Earlier flow maps and geographic networks have already employed that strategy, in order to produce drawings that generate star-subgraphs [19]. However, to our knowledge, only a few studies have effectively investigated it in the context of edge bundling. These studies are described next.

An edge bundling approach for adjacent edges only was first introduced in Beck *et al.* [1]. They proposed a method that preserves the traceability of directed graphs using the notion of an implicit, or a computed hierarchy, to create the bundles restricted to edges that start or end at the same node.

Čermák, Dokulil e Katreniaková [31] proposed and compared four edge bundling algorithms for adjacent edges based on preventing edges and nodes from overlapping by using an edge-routing algorithm to add bends to edges incident with a bundle. The choice of the edges that would be part of a bundle is dictated by the following two strategies. The first strategy splits adjacent edges into groups, considering a critical angle, and the second approach uses a node clustering algorithm to choose the edges to be joined, considering the adjacency of group of nodes.

Luo *et al.* [16] have suggested the simple idea of joining only adjacent edges to address the ambiguity problem. They presented a method called ambiguity-free edge-bundling that adds ambiguity resolution into the process of bundling. The technique bundles edges together with same target/source nodes that are also geometrically close, applying a quadtree structure (a tree in which each internal node has exactly four children) to choose the edges in each bundle. A bundled curved-edge layout is automatically generated to avoid ambiguity. The "edge ambiguity" problem is illustrated in the Figure 3, where in (a) it is hard to identify the relational patterns between nodes A, B, C and D, but (b), on the other hand, clearly shows two possible configurations (AC, BD) and (AD, BC).

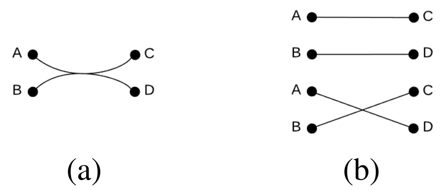


Figura 3: Edge Ambiguity: (a) ambiguity bundle, (b) some possible interpretations.

Peng *et al.* [21] proposed a method called the node-based bundling method that clusters and knot edges. The authors noted that edge bundling is akin to knotting edges, and traditional techniques knot the edges in the middle of the bundle while, by contrast, their approach moves the knot nearer to the common node of adjacent edges, arranged based on their orientation, i.e, the knot is placed near the central node and not in the middle of the bundle.

Nocaj and Brandes [19] proposed a method called stub bundling to visualize unambiguous geographic networks based on bundling adjacent edges. The authors assume that joining edges at their endpoint ensures the edges in the bundles do not deviate from the line connecting the endpoints and induces a general direction of destinations. They refined the approach of Peng *et al.* [21] and used the angle between consecutive edges as the criterion to choose the edges that are joined together. Another contribution of Nocaj and Brandes is called a confluent spiral drawing in which the intersection angle between two spirals is zero, thus producing a drawing with a star-configuration.

As these examples demonstrate, bundling only adjacent edges is relevant in practice, with concrete bundling algorithms falling into this class of problem. The algorithms produce bundles that are, in fact, *star subgraphs*, whose common node between the edges is identified as the central node of the bundle.

In the following section we introduce edge bundling of adjacent-only edges as an optimization process. We call this problem *star-edge bundling*, or $EB-star$ for short. This allows us to begin the formal discussion about edge bundling problems and to show the applicability of the EB formulation.

3.2 Formulation of the star-edge bundling problem

A compact definition of $EB-star$ decision problem is as follows:

Definição 3.1 *Given a graph $G = (V, E)$, a drawing D of G with straight-line edges and a positive integer $k \leq |E|$, the decision version of the star-edge bundling problem is to determine if there is a partition of E into disjoint subsets $E_1, E_2, ..., E_n$, with $E = \cup_{i=1}^{n} E_i$, $n \leq k$, and each E_i, for $i = 1, 2, ..., n$, inducing a star subgraph G_i (i.e. all edges in E_i share the same vertex v_i).*

The star-edge bundling problem $EB-star$ is a specialization of EB that represents a simpler and potentially useful possible case of edge bundling. This investigation assumes that the function R is a simple method that draws each $E_i \in S$ as an edge bundle, rendering it using a spline routing structure. In our model $EB-star$, visual aspects of the problem are not addressed, and the approach is only focused on finding the optimal set S. So R is a post-processing function that can be omitted from the definition, since it does not contribute to the resolution of the optimization problem.

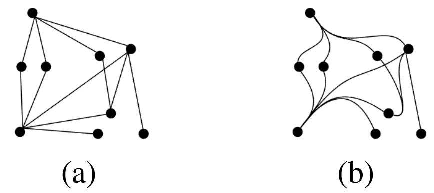


Figura 4: Solving the star-edge bundling problem: (a) a drawing of a graph and (b) an edge bundled representation.

Figure 4 illustrates a graph and its bundled graph drawing, obtained by solving the optimization of the star-edge bundling problem.

Next, a formal proof of the NP-completeness of the decision version of the star-edge bundling problem is presented. Note that, the minimum vertex cover problem VC, is stated as follows: given a graph $G = (V, E)$ and a positive integer k, decide if there is a vertex cover $A \subseteq V$ with size $|A| \leq k$.

Teorema 3.2 *The decision problem $EB-star$ is NP-complete.*

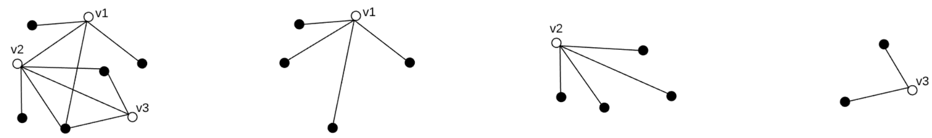
Prova. In order to prove that $EB-star$ is NP-complete, it is necessary to establish that it is in both NP and NP-hard. The first part of the proof is straightforward: given a graph $G = (V, E)$, an integer $k \leq |E|$ and a certificate $S = \{E_1, E_2, ..., E_l\}$ that represents a partition of E in subsets, it is possible to verify in polynomial time that S has at most k subsets ($l \leq k$), if $E_1, E_2, ..., E_l$ are disjoint, if $E = \cup_{i=1}^{l}(E_i)$ and if each E_i induces a star subgraph. This shows that $EB-star$ is in NP.

The NP-hardness of $EB-star$ can be demonstrated by a reduction from the minimum vertex cover problem VC. Let $G = (V, E)$ be a graph. $A \subseteq V$ is a *vertex cover*, if for all $\{u, v\} \in E, u \in A$ or $v \in A$. The NP-hard proof of $EB-star$ is based on the observation that a valid vertex cover for VC can be mapped to a correct partition for $EB-star$. Assume there is a vertex cover $A = \{v_1, v_2, \ldots v_p\}$ with size at most k. For every v_i in A, $i = 1, 2, \ldots p$, let D_i be the set of edges linked to v_i. Note that every v_i "covers" the edges in D_i, and that $E = \cup_{i=1}^{p} D_i$. Note as well that every D_i induces a star subgraph of G with central node v_i. The sets D_i are not necessarily disjoint. So, let $E_i = \{e \in D_i \mid e \text{ does not appear in } D_j \text{ for } j \neq i\}$. The sets E_i are now disjoint subsets of E that induce star subgraphs each, forming a valid partition for $EB-star$ with $p \leq k$ subsets. Conversely, a solution to $EB-star$ provides a solution to VC. First, assume that there is a partition $S = \{E_1, E_2, ..., E_l\}$ of E with at most k subsets, $l \leq k$, that is, a solution for $EB-star$. Each E_i, $1 \leq i \leq l$, induces a star subgraph of G with a central node v_i. Since every central node v_i "covers" the edges in E_i, and S contains all edges of E, there is a vertex cover $A = \cup_{i=1}^{l} v_i$ with size at most k.

□

The Theorem 3.2 is important as from its conclusion we can deduce that an existing algorithm for VC can be used for producing a solution to $EB-star$. Figure 5 shows an example of star edge bundling created by a vertex cover solution. The vertex cover $V^r = \{v_1, v_2, v_3\}$ can be used to determine the bundles.

One limitation of $EB-star$ is that it is somewhat of an artificial problem to solve because the solutions to it can contain edges bundled together that have a high angular relation between them. For example, a solution to $EB-star$ could result in a bundle with edges that are 180° apart in the underlying graph drawing. Simply joining all edges of a star graph may be unrealistic, in particular if some of them are wide apart.



(a) (b) (c) (d)

Figura 5: Example of star edge bundling: (a) original graph with vertex cover $V^r = \{v_1, v_2, v_3\}$; (b), (c) and (d) induced star subgraph by vertex cover

In the next section, an extended formulation of $EB\!-\!star$ is presented that includes an angular constraint. The aim is to avoid inconvenient bundles, by bundling pairs of adjacent edges only when the angle between them is less than or equal to a given maximum value.

4 Star edge bundling problem with maximum angle constraint

Consider a maximum angle constraint between all pairs of adjacent edges, as a compatibility criterion. This criterion can be taken into account when performing edge bundling, and can be included in the problem formulation as a new constraint as a part of the constraint set P. We define γ_{ab} as the smallest of the two angles, clockwise and counterclockwise, between adjacent edges a and b of a graph. As can be seen, $\gamma_{ab} \leq 180°$, since there is always one orientation in which the angle between a and b is less than or equal to 180°.

The new problem definition, denoted as $EB\!-\!star_\alpha$, is presented below.

Definição 4.1 *Given a graph $G = (V, E)$, a drawing D of G with straight-line edges, a positive integer $k \leq |E|$ and an angle $0 \leq \alpha \leq 180°$, the decision version of the star-edge bundling problem with maximum angle constraint ($EB\!-\!star_\alpha$) is to determine if there is a partition of E into disjoint subsets $E_1, E_2, ..., E_n$, with $E = \cup_{i=1}^{n} E_i$, $n \leq k$, and each E_i, for $i = 1, 2, ..., n$, inducing a star subgraph G_i with $\gamma_{xy} \leq \alpha$ for every two edges x and y in E_i, where γ_{xy} is the smallest angle between the adjacent edges x and y.*

Figure 6 compares possible solutions to $EB\!-\!star$ (b) and $EB\!-\!star_\alpha$ (c) and (d) for the original graph in (a). Observe that some of the bundles created by $EB\!-\!star$ in (b) are split in the new formulation into smaller and more legible bundles in (c) and (d) when the angular constraint was applied.

(a) (b) (c) (d)

Figura 6: Comparing solutions by $EB\!-\!star$ and $EB\!-\!star_\alpha$: (a) a drawing of a graph; (b) a solution to $EB\!-\!star$ with four bundles; (c) and (d) two different solutions to $EB\!-\!star_\alpha$ with angle constraint $\alpha \leq 30°$ and five bundles.

It is clear that, the minimum number of subsets in any solution partition of E will, in general, vary with α.

In order to analyze the complexity of the $EB{-}star_{\alpha}$ problem, we define min_{γ} and max_{γ} of a graph $G = (V, E)$ as, respectively, the smallest and the largest angle between all pairs of adjacent edges of G in D. And as attempt to clarify some issues concerning angular constraints, we establish the following three properties, namely Properties 1, 2 and 3.

Property 1 *When $0 \leq \alpha < min_{\gamma}$, the length of the optimal solution S for $EB{-}star_{\alpha}$ is $|E|$.*

Prova. Setting $0 \leq \alpha < min_{\gamma}$ makes all pairs of edge incompatible. This results in a trivial solution for $EB{-}star_{\alpha}$, where E is partitioned into disjoint subsets $E_1, E_2, \ldots E_{|E|}$ with every $|E_i| = 1$. □

Property 2 *When $max_{\gamma} \leq \alpha \leq 180°$, $EB{-}star_{\alpha}$ has the same solution as that for the $EB{-}star$ model.*

Prova. If $max_{\gamma} \leq \alpha \leq 180°$, then all adjacent edges of G can be bundled together, which makes the $EB{-}star_{\alpha}$ equivalent to $EB{-}star$. So, for that range of values of α, $EB{-}star_{\alpha}$ becomes NP-hard (NP-complete for the decision-version problem). □

Property 3 *Let S^r be the disjoint subsets (bundles) in the solution space for the optimization $EB{-}star$ problem and S be the disjoint subsets in the solution space of $EB{-}star_{\alpha}$ for a graph G and a drawing D of it. Then $|S^r| \leq |S|$.*

Prova. This property implies that the number of bundles in the optimum solution for $EB{-}star$ is a lower bound on the optimal value of $EB{-}star_{\alpha}$. Considering the graph G as input for both problems and assuming they have the same set of nodes as the centers of the induced star subgraphs. As the optimal solution for $EB{-}star_{\alpha}$ is restricted by an angular constraint, it tends to have a number of edges lower than $EB{-}star$ for each center node which increases the number of bundles in S. □

The challenge is to establish the complexity of $EB{-}star_{\alpha}$ when $min_{\gamma} \leq \alpha < max_{\gamma}$. We expect the problem complexity to reduce, and the number of bundles to increase, as α decreases, for certain types of graphs. However, this has yet to be investigated, and requires a more elaborate proof. An analysis of these aspects is an interesting topic for future research.

4.1 Considerations about maximum overall bundle angle

The $EB{-}star_{\alpha}$ formulation does not explicitly consider a limit on the overall angle of the bundles [1]. On the contrary, it defines only a constraint for joining two edges based on their edge-to-edge angle γ – the smallest angle (clockwise or counterclockwise) between them –, what is simpler and less expensive to calculate. As a consequence, in some cases, adjacent edges may be joined together causing the overall bundle angle to be higher than α. This problem occurs when γ is measured clockwise for some pairs of edges included in the bundle, and counterclockwise for other pairs.

Figure 7(a) illustrates the problem. In it, the graph has only three adjacent edges (A, B and C), and the edge-to-edge angle is the same for all of them ($\gamma_{AB} = \gamma_{BC} = \gamma_{CA} = 120°$).

(a) (b)

Figura 7: Star-edge bundling with maximum angle constraint: (a) case in which all edges will be joined together if $\alpha = 135°$; and (b) case in which small bundles will be created if $\alpha = 90°$.

Considering a maximum angle constraint $\alpha = 135°$, the optimum solution for $EB{-}star_\alpha$ is to join all three edges, leading to an overall bundle angle of 240°.

The same problem will not occur for small values of α. For instance, in Figure 7(b), the graph has sixteen edges equally spaced and connected to a central vertex. Considering a maximum angle constraint $\alpha = 90°$, the value of α implicitly imposes a limit on the overall bundle angle, which will not exceed 90°.

For the particular example in Figure 7(b), consider $G = (V, E)$ the graph, and $S = \{E_1, E_2, ..., E_n\}$ an optimal solution for $EB{-}star_\alpha$ with $\alpha = 90°$. Now let E_i, $i \in \{1, \ldots, n\}$, be an arbitrarily chosen bundle from S, and A and B the pair of edges in E_i with highest edge-to-edge angle (γ_{AB}). Note that, because of the maximum angle constraint, $\gamma_{AB} \leq \alpha$. It is easy to show that any other edge $C \in E_i$, with a distinct position, appears between A and B in the orientation defined by γ_{AB}, what guarantees that γ_{AB} is also the overall bundle angle. If C appears before A, then $\gamma_{CB} = \gamma_{CA} + \gamma_{AB}$, what contradicts the proposition that γ_{AB} is the highest edge-to-edge angle in the bundle. A similar conclusion can be reached when considering that C could appear after B in the orientation from A to B.

This reasoning is not valid, however, for Figure 7(a), not even for 7(b), if $\alpha = 135°$, because C may appear before A and still have an angle $\gamma_{CB} \leq \alpha$, taken in the opposite orientation (going from C directly to B without passing by A).

In fact, it is possible to devise a general rule for when the maximum angle constraint α always works as a limit on the maximum overall bundle angle.

Rule 1 *If $\alpha < 120°$, then the overall bundle angle of every bundle in a solution of $EB{-}star_\alpha$ is less or equal to α.*

A demonstration of this rule can be obtained by extending the initial analysis presented above. When α is sufficiently small, any adjacent edge X outside the orientation given by the maximum edge-to-edge angle γ_{AB} is too far away either from A or from B, even when taken in the opposite direction. Thus, an external edge X (before A or after B) would never be included in the same bundle with A and B. Such condition is valid when the maximum allowed angle (α) between two edges A and B, plus the same angle before A, added to that angle after B does not make a complete turn, that is, it is less than 360°. This can be formulated as $3\alpha < 360°$, which implies in $\alpha < 120°$.

This observation is useful because it shows that it is possible to limit the overall angle of the bundles by only restricting the angles between the bundled edges when $\alpha < 120°$. No extra calculations, like sorting adjacent edges or summing up the angles of consecutive edges in a bundle, are necessary. In addition, the computation of the γ values and of which pairs of adjacent edges may be bundled together can be done as a pre-processing stage, saving computational time later in the main edge bundling optimization process.

[1] The overall angle of a bundle can be obtained by summing up the angles between all two consecutive edges in it, in the orientation (clockwise or counterclockwise) that provides the smallest value.

5 An integer linear programming for edge bundling problem

In order to investigate the effect of the complexity of the problem $EB-star_\alpha$, we propose an integer linear programming model for solving the $EB-star_\alpha$ problem with a linear objective and linear constraints. Numerical experiments show that the model produces the global optimal solution for certain instances using the angles $\alpha = 30°$, $\alpha = 45°$, $\alpha = 70°$, $\alpha = 90°$ and $\alpha = 110°$.

Necessary theoretic notation and terminology are given as follows. For a given angle α, let $EB-star_\alpha$ denote the star edge bundling problem for an embedding (a drawing) of a given G in the plane with a given maximum angle constraint of α, as defined in the previous section. Let $G^r = (V, E^r)$ denote the embedding of G. We say two edges in E^r are adjacent if their equivalent edges in E are adjacent. A pair of edges in E^r is termed *compatible* if the edges can be bundled with respect to α. Note, any two edges can be compatible only if they are adjacent. A pair of edges in E^r that are not compatible are said to be *incompatible*. A *compatibility matrix* $\delta_{\alpha G^r}$ for G^r with respect to α is defined on all pairs of edges $i, j \in E^r$ as

$\delta_{\alpha G^r}(i, j) = 1$, if i and j are compatible,

$\delta_{\alpha G^r}(i, j) = 0$, otherwise.

A partition of E^r into subsets $E_k, k = 1, \dots, p$; for some $p \leq m$ is called an *$\alpha-$edge bundling plan*. Note that the choice of p is not intuitive and in general nontrivial instances it might have to be found by experimentation.

5.1 An ILP model of $EB-star_\alpha$

The following proposed model is defined on a given embedded graph drawing $G_r = (V, E_r)$ and a maximum angle α, that induces an edge compatibility matrix $\delta_{\alpha G^r}$. The aim of the model is to find an $\alpha-$edge bundling plan with an edge partition that has the minimum number of subsets.

MODEL IP

Minimize:

$Z_{EB-star_\alpha} = \sum_{i \in E} \sum_{k=1}^{m} k x_{ik}$

subject to

$\sum_{k=1}^{m} x_{ik} = 1, i \in E,$

$x_{ik} + x_{jk} \leq 1, i, j \in E, \delta_{\alpha G'}(i, j) = 0, k = 1, \dots, m,$

where $x_{ik} = 1$, if $i \in E_k, x_{ik} = 0$, otherwise, $i \in E, k = 1, \dots, m.$

Constraint (2), ensures that each edge is assigned to exactly one subset in the partition. Constraint (3) guarantees that only compatible edges are assigned to the same subset. Constraints (4) is the usual binary condition reflecting the fact that each edge is either assigned to a particular subset or it is not.

5.2 Experimental results and discussion

In this section, we present numerical results of the performance of the proposed model **MODEL IP**. We used two graphs, the first graph is synthetic consisting of 20 vertices and 28 edges, the second graph is the PlanarGD2015 provided by the ISGCI [12] consisting of 66 vertices and 101 edges. The solver was implemented using Gurobi [20] and the algorithm was run on a server DELL M630 with 128 GB of RAM and 2 processors with 10 cores each (40 visible cores) of 3-3.6Ghz. The results for the graph instances are summarized in Table 1 and Table 2.

Max. Angle	Variables	Constraints	Time (s)	Bundles
110	784	9044	0.166445	10
90	784	9268	0.161957	11
70	784	9548	0.354999	13
45	784	9856	0.396842	16
30	784	9996	0.144982	17

Tabela 1: Optimal solutions for the Synthetic graph.

Max. Angle	Variables	Constraints	Time (s)	Bundles
110	10201	487123	-	-
90	10201	492880	1275.27	50
70	10201	497021	1210.29	58
45	10201	502475	656.139	69
30	10201	504899	125.654	74

Tabela 2: Optimal solutions for the PlanarGD2015.

As can be seen in Tables 1 and 2, the process of implementing **MODEL IP** generated optimal solutions for the instances of the smaller problem, but took an inordinate amount of time to solve some of the instances of the larger problem and failed to solve instances with higher maximum angles. This underline the complexity of the $EB{-}star_\alpha$ problem as discussed in the section 4.

Figure 8 and Figure 9 illustrate the graph drawing layouts for the optimal solution produced by the implementation of **MODEL IP** for the Synthetic and PlanarGD2015 graphs respectively. Bundled edges are rendered as spline combined with degraded red-to-green, red color represents source and green the destination. In the single-edge size bundles, the edge is drawn straight-line and colored in gray. The red nodes correspond to a vertex cover set. Note that, the nodes of the vertex cover are not always the center of the bundles, what underline the differences between the problems $EB{-}star$ and $EB{-}star_\alpha$.

6 Conclusion

This paper presented a general mathematical formulation, called EB, for edge bundling as an optimization problem. The formulation considers objective functions to be optimized, representing quality aspects of the bundling, and a set of constraints to be satisfied, such as compatibility criteria. The objective functions and the constraints can be defined based on aspects of the graph, of its drawing or on the output of a rendering procedure, among other parameters. A simple specialization of the edge bundling formulation were provided, focusing on bundling only adjacent edges and on minimizing the number of bundles, which was proved to be NP-complete. In addition, an extended version of this problem, that included a maximum-angle compatibility constraint, was proposed and analyzed. An integer linear programming model was formulated for it. Simulations on graph instances showed the effectiveness of the model, but with a high computational time underlined the complexity of problem. The EB formulation and the specific problems investigated in the paper show a great potential for modeling and studying edge bundling problems as optimization problems.

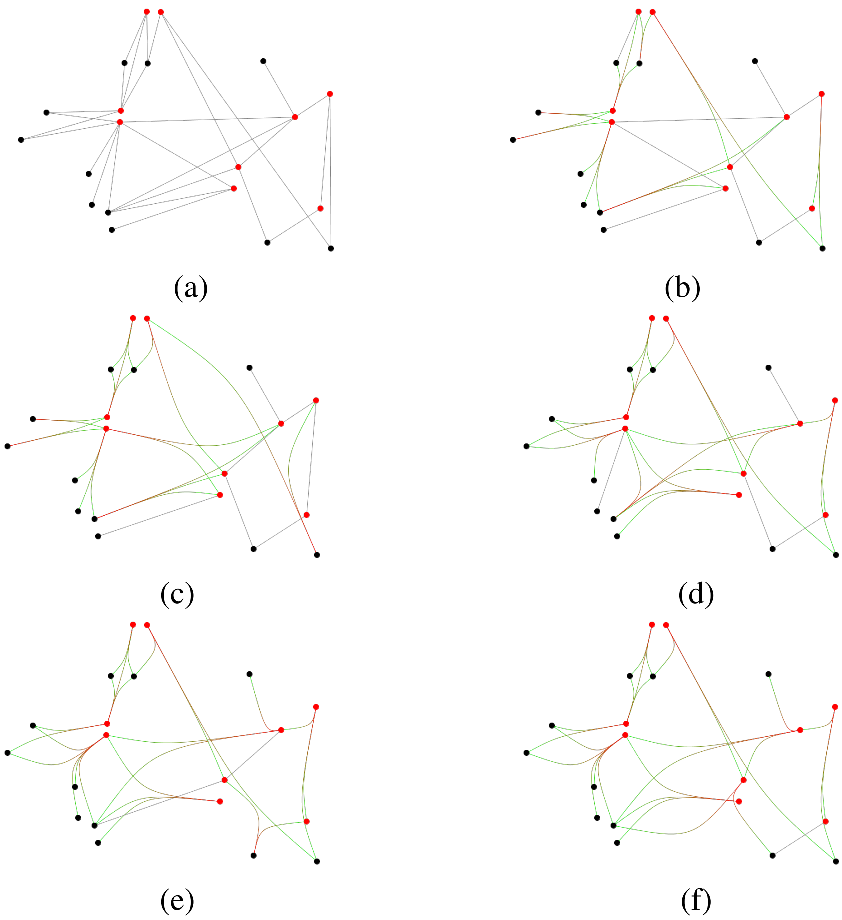


Figura 8: Optimal solutions for the Synthetic graph: (a) original graph, (b) optimal solution for $\alpha = 30°$,(c) optimal solution for $\alpha = 45°$, (d) optimal solution for $\alpha = 70°$, (e) optimal solution for $\alpha = 90°$ and (f) optimal solution for $\alpha = 110°$.

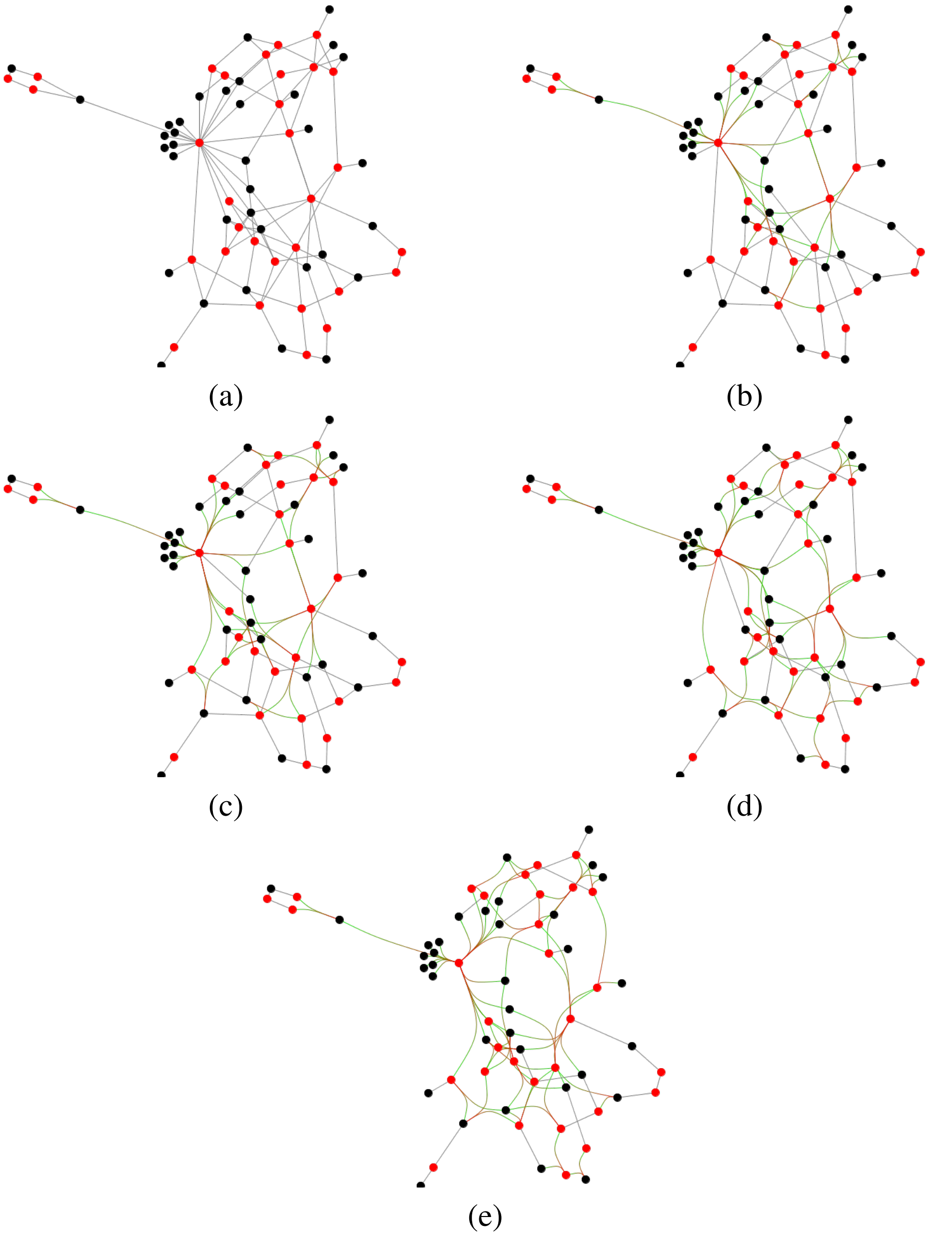


Figura 9: Optimal solutions for the PlanarGD2015 graph: (a) original graph, (b) optimal solution for $\alpha = 30°$, (c) optimal solution for $\alpha = 45°$, (d) optimal solution for $\alpha = 70°$ and (e) optimal solution for $\alpha = 90°$.

For future work, many interesting topics should be investigated. Among them, we suggest: defining new specializations of *EB*, by including other objective functions and constraints; establishing the complexity of these problems, proving the complexity of $EB{-}star_\alpha$ for particular graphs and values of α; developing heuristics for the proposed problems; and classifying some existing edge bundling methods based on the formal *EB* optimization definition.

Referências

[1] Beck, F.; Puppe, M.; Braun, P.; Burch, M.; Diehl, S. **Edge bundling without reducing the source to target traceability**. In: *Proc. InfoVis Poster (2011)*, p. 298–305, 2011.

[2] Cui, W.; Zhou, H.; Qu, H.; Wong, P. C.; Li, X. **Geometry-based edge clustering for graph visualization**. *IEEE Trans. Vis. Comput. Graph*, 14(6):1277–1284, 2008.

[3] Dickerson, M. T.; Eppsteing, D.; Goodrich, M. T.; Meng, J. Y. **Confluent drawings: visualizing nonplanar diagrams in a planar way**. In: *Proc. 11th International Symposium on Graph Drawing*, p. 1–12, 2003.

[4] Ersoy, O.; Hurter, C.; Paulovich, F. V.; Cantareiro, G.; Telea, A. **Skeleton-based edge bundling for graph visualization**. *IEEE Trans. Vis. Comput. Graph*, 12(17):2364–2373, 2011.

[5] Gansner, E. R.; Koreng, Y. **Improved circular layouts**. In: *Proc. 14th International Symposium on Graph Drawing*, p. 386–398, 2007.

[6] Gansner, E. R.; Y. Hu, S. N.; Scheidegger, C. **Multilevel agglomerative edge bundling for visualizing large graphs**. In: *Proc. IEEE Pacific Visualization Symposium*, p. 187–194, 2011.

[7] Henry, N.; Fekete, J.; McGuffin, M. J. **Nodetrix: a hybrid visualization of social networks**. *IEEE Trans. Vis. Comput. Graph*, 13(6):1302–1309, 2007.

[8] Holten, D. **Hierarchical edge bundles: visualization of adjacency relations in hierarchical data**. *IEEE Trans. Vis. Comput. Graph*, 12(5):741–748, 2006.

[9] Holten, D.; van Wijk, J. J. **Force-directed edge bundling for graph visualization**. *Comput. Graph. Forum*, 28(3):983–990, 2009.

[10] Hurter, C.; Conversy, S.; Gianazza, D.; Telea, A. C. **Interactive image-based information visualization for aircraft trajectory analysis**. *Transportation Research Part C: Emerging Technologies*, 47:207–227, 2014.

[11] Hurter, C.; Ersoy, O.; Telea, A. **Graph bundling by kernel density estimation**. *Comput. Graph. Forum*, 31(3):865–874, 2012.

[12] ISGCI. **Information system on graph classes and their inclusions**. `http://www.csun.edu/gd2015/topics.htm`, 2015. [Online; accessed 19-May-2017].

[13] Landesberger, T. V.; Kuijper, A.; Schreck, T.; Kohlhammer, J.; Wijk, J. V. **Visual analysis of large graphs: State-of-the-art and future research challenges**. *Comput. Graph. Forum*, 30(6):1719–1749, 2011.

[14] Lin, X. **On the computational complexity of edge concentration**. *DAMATH: Discrete Applied Mathem. and Combin. Operat. Research*, 101:1–3, 2000.

[15] Lovasz, L. **Large networks and graph limits**. American Mathematical Society, 2012.

[16] Luo, S. J.; Liu, C. L.; Chen, B. Y.; Ma, K. L. **Ambiguity-free edge-bundling for interactive graph visualization**. *IEEE Trans. Vis. Comput. Graph*, 18(5):810–821, 2012.

[17] Newbery, F. J. **Edge concentration: a method for clustering directed graphs**. *SACM SIGSOFT Software Engineering Notes*, 14(7):76–85, 1989.

[18] Nguyen, Q.; Eades, P.; Hong, S. H. **On the faithfulness of graph visualizations**. In: *Proc. IEEE Pacific Visualization Symposium*, p. 209–216, 2013.

[19] Nocaj, A.; Brandes, U. **Stub bundling and confluent spirals for geographic networks**. In: *Proc. 21st International Symposium on Graph Drawing*, p. 388–399, 2013.

[20] Optimization, G. **Gurobi optimizer quick start guide**. `http://www.gurobi.com/documentation/7.0/quickstart_linux.pdf`, 2017. [Online; accessed 14-May-2017].

[21] Peng, D.; N. Lu, W. C.; Peng, Q. **Sideknot: Revealing relation patterns for graph visualization**. In: *Proc. IEEE Pacific Visualization Symposium*, p. 65–72, 2012.

[22] Peysakhovich, V.; Hurter, C.; Telea, A. **Attribute-driven edge bundling for general graphs with applications in trail analysis**. In: *Proc. IEEE Pacific Visualization Symposium*, p. 39–46, 2015.

[23] Phan, D.; Xiao, L.; Yeh, R.; Hanrahan, P.; Winograd, T. **Flow map layout**. In: *Proc. IEEE Information Visualization (InfoVis)*, p. 219–224, 2005.

[24] Pupyrev, S.; Nachmanson, L.; Bereg, S.; Holroyd, A. L. **Edge routing with ordered bundles**. In: *Proc. 19th International Conference on Graph Drawing*, p. 136–147, 2011.

[25] Qu, H.; Zhou, H.; Wu, Y. **Controllable and progressive edge clustering for large networks**. In: *Proc. 14th International Conference on Graph Drawing*, p. 399–404, 2006.

[26] Quigley, A. **Large Scale Relational Information Visualization, Clustering, and Abstraction**. PhD thesis, Computer Science, Univeristy of Newcastle, 2001.

[27] Shneiderman, B.; Dunne, C. **Interactive network exploration to derive insights: Filtering, clustering, grouping, and simplification**. In: *Proc. 20th International Conference on Graph Drawing*, p. 2–18, 2012.

[28] Tarawaneh, R. M.; Keller, P.; Ebert, A. **A general introduction to graph visualization techniques**. In: *Proc. IRTG 1131 Workshop 2011*, p. 151–164, 2011.

[29] Telea, A. C. **Data visualization, principles and practice**. CRC Press, 2015.

[30] Telea, A. C.; Ersoy, O. **Image-based edge bundles: simplified visualization of large graphs**. *Comput. Graph. Forum*, 29(3):843–852, 2010.

[31] Čermák, M.; Dokulil, J.; Katreniaková, J. **Edge routing and bundling for graphs with fixed node positions**. In: *Proc. 15th International Conference on Information Visualization*, p. 475–481, 2011.

[32] Zielasko, D.; Weyers, B.; Hentschel, B.; Kuhlen, T. W. **Interactive 3d force-directed edge bundling**. *Comput. Graph. Forum*, 35(3):51–60, 2016.