Computers & Graphics (2023)

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

NeatSankey: Sankey Diagrams with Improved Readability Based on Node Positioning and Edge Bundling

ARTICLE INFO

Article history:
Received July 20, 2022

Keywords: Visualization, Sankey diagrams, Node positioning, Edge bundling, Readability

ABSTRACT

Sankey diagrams are widely used to visualize event sequence data. However, when the data volume is large, its readability is affected by dense edge crossings, excessive swing amplitude, and small crossover angles, while it is computationally intensive to obtain an optimal layout. In this paper, we propose NeatSankey, a balanced method that generates Sankey diagrams smoothly. It can be laid out quickly with good readability when Sankey diagrams are very complex. At the same time, to comprehensively evaluate the readability of Sankey diagrams, we use three evaluation metrics: crossing number, swing amplitude, and layout coverage. Firstly, we use a heuristic layout algorithm and a force-directed algorithm to adjust the node layout to minimize the edge crossings and swing amplitude with edge widths considered. Secondly, to better reduce the dense confusion caused by edge crossings, we introduce a edge bundling algorithm based on attribute similarity. We present three evaluations: a comprehensive comparison of our results with state-of-the-art techniques, user studies with thirty volunteers, and a case study of two datasets. Our evaluations demonstrate the effectiveness and practicability of the NeatSankey.

1. INTRODUCTION

Sankey diagrams are beneficial in presenting the trends of data flow [1]. Edges represent flowing data. Widths of edges indicate specific values. Nodes represent different classifications. The width of the edges is proportional to the flow rate. Because showing how data flows is essential in many applications. Sankey diagrams have been used to represent the life cycle assessment (LCA) of products [2], household budgets [3], changes in energy generation [4].

Since Sankey diagrams are directed acyclic graphs, they have some evaluation metrics in common such as the number of edge crossings [5] and the area of edge crossings [6]. Besides, Sankey diagrams are mainly controlled by drawing nodes and edges. Variants of Sankey diagrams have been proposed to reduce confusion by optimizing these two factors. Sugiyama et al. [7] developed a four-stage heuristic algorithm to draw hierarchical networks, aiming to reduce edge crossings. Since nodes belonging to the same class are at the same hierarchy, Sankey diagrams are mostly drawn using a hierarchical network layout. Therefore, the Sugiyama algorithm is often used in the drawing of Sankey diagrams. In addition, to obtain the optimal result, Zarate et al. [6] introduced an integer programming approach and considered the weights of the edges. However, when the amount of data is large and the number of edge crossings is high, Sankey diagrams will inevitably become visually confusing.

From a brief review and previous assessments, we believe that the current Sankey diagrams are able to reduce the number of edge crossings relatively well but still need to be strengthened, while other limitations need further consideration. We define the readability of Sankey diagrams as measured by Crossing Number (CN), Swing Amplitude (SA), and Layout Coverage (LC).

- Edge crossings. Previous approaches treat edges of different widths equally or assign higher weights to thicker edges, which can lead to ignoring the intersection of thin edges. However, we believe that within a dense region, the intersection of thin edges can also create a great deal of

confusion.

- The amplitude of the swing. According to the theory of shortest edges in directed graphs, short edges help improve aesthetics. In drawing Sankey diagrams, if the edge swing is too large, then it will be difficult to track the data flow, which will cause visual confusion.
- The coverage of layout. The information of space needs to be considered in node layout and edge layout to leave large blank areas to enhance the visualization of the diagram. The confusion in Sankey diagrams comes mainly from the crossing of edges and the area of coverage. When the edges are bundled together, there will be less crossover and less ink on the canvas. Therefore, we propose the space utilization metric, the smaller the metric, the better the performance.

In this paper, we propose NeatSankey, a two-step algorithm, which can minimize the effects of the above three limitations. In the first step, a node layout algorithm, including a reordering algorithm and a force-directed algorithm, is proposed to switch the nodes and determine their position. The reordering algorithm is introduced to get the order of nodes with width constraints. Then, a force-directed algorithm is designed to fix the position of the nodes after the ordering. (see Fig 2)

In the second step, after the node layout is completed, edge bundling is performed to further reduce the clutter generated by edge crossings, especially the layout coverage. By hierarchical clustering, edges at the same level are clustered together for bundling. To show the exact connections, only edges with the same source or destination node can be clustered into one class. To accommodate large datasets, we will cluster again based on the directional angle of the edges. One interesting part is that we can bundle edges with different widths by dividing wide edges into thin edges. To our best knowledge, no one has done this while drawing Sankey diagrams. According to the result of the clustering, we bundle them to better use of space. Although it will distort the representation of cardinality, our method can still reflect the flow of an edge by the ends of the edge.

Our primary contributions are as follows.

- The readability of Sankey diagrams is enhanced by designing the node positioning algorithm and edge bundling algorithm to reduce the visual confusion caused by edge crossings with edge widths considered.
- We propose a novel edge bundling method for Sankey diagrams, which can bundle edges with different widths without adding new confusion to the original Sankey diagrams, and can show the exact connections.
- We evaluate the quality of NeatSankey quantitatively and conduct a user study of 30 volunteers to illustrate the effectiveness and practicability of our method.

2. RELATED WORK

Sankey diagrams and their variants have been studied significantly in data visualization. The Sankey diagram is one of the vital diagrams for observing the flow of data, which can help us understand the information. Sankey diagrams appear in multiple disciplines and help explain complex systems to non-experts who may play an essential role in the system's operation [8]. By observing the Sankey diagram, we can better understand the flow of data and provide support for decisions. Sankey diagrams are highly relevant to graphical visualization, and a great deal of work has focused on improving the visual effect.

2.1. Sankey Diagram Visualization

Multilevel digraphs, called hierarchies, constitute an important subclass of digraphs [9]. A hierarchical network diagram is used to represent the dependencies between components that belong to different layers. Hierarchical graph layout is the preferred method for visualizing a general flow direction (e.g., data or information) in relational data [10]. As an important diagram to show data flow, the Sankey diagram can describe multi-level relationships and classify nodes by level. Since nodes belonging to the same class can be grouped into the same hierarchy, Sankey diagrams are mostly drawn using a hierarchical network layout. Sugiyama et al. [7] developed a four-stage heuristic method to draw hierarchical networks. The first two stages of this method handle cycle removal and the assignment of nodes to layers. The third stage, a barycentric method, is mainly used to reduce edge crossings. Based on it, most of the current variants apply different node layout algorithms and several other attempts for domain-specific applications. In this paper, inspired by it, we not only optimize the node layout of the diagram but also reduce the visual clutter.

Sugiyama et al. [7] used a barycentric heuristic algorithm to control ordering of nodes in hierarchical networks. In this heuristic method, the nodes in the free layer are placed in the barycentric coordinates of their neighboring nodes in the previous layer. After determining the order of the nodes in the free layer, the layer is fixed. Then the nodes of the next layer are moved with that fixed layer. Eades and Wormald [11] show an alternative way to set the free nodes' positions by the median of their neighbor nodes rather than the barycentric. However, their works focus mainly on reducing the number of crossings. Still, they do not consider the weights when edges with different widths cross, i.e., the different effects of different widths of edges crossing together on the readability of the graph.

When drawing Sankey diagrams, previously published studies are limited to the number of crossings. To eliminate this limitation, Alemasoom et al. [12] introduced the weights of the edges and developed a two-step criterion. First, they extend the barycentric heuristic algorithm that Sugiyama et al. presents to find a node ordering with fewer edge crossings. Then, to minimize the sum of the distance between two nodes, they apply a linear programming approach to adjust the positions of the nodes, thus minimizing the distance between nodes. However, this layout is not optimal. Minimizing the number of crossings is an NP-hard problem. Garey et al. [13] proved that computing the crossing number of a graph is an NP-hard problem. Jünger et al. [14] computed the sparse instances to obtain exact optima for the problem of two-layer straight-line crossing minimization, even though the problem is NP-hard. In some cases, some NP-hard problems can be optimized easily once modeled as integer linear programmings (ILPs). Zarate et al.[6] created an optimal Sankey diagram using an ILP model, which sets the

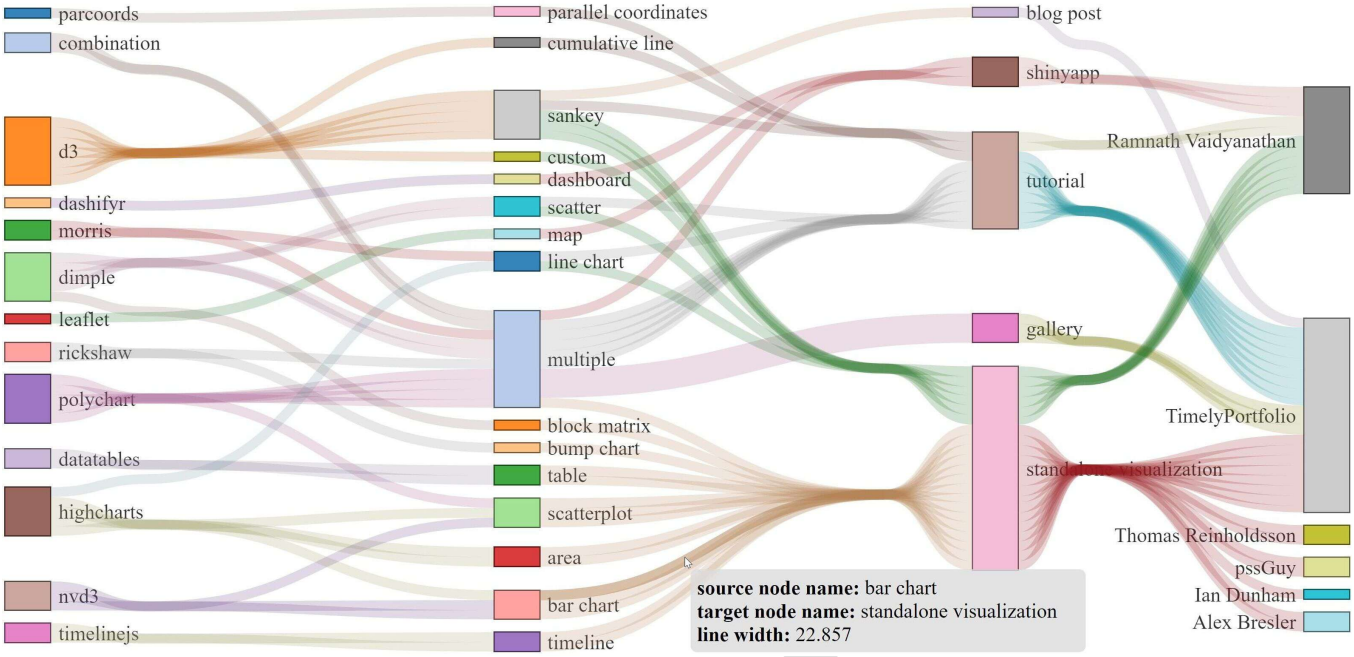


Fig. 1: The dataset has 4 layers and contains 40 nodes, showing the relationship between js chart library, chart types, chart uses, etc. From the diagram we can clearly see that Sankey diagrams are the main view for making standalone visualizations and are widely used, and that Sankey diagrams are mainly drawn by d3.js. Our NeatSankey method employs a two-step algorithm, first controlling the position of nodes through a force-directed node layout algorithm, and then reducing the confusion caused by crossings through an edge-bundling algorithm. Also, we use the hover marker to identify the flow of the edge. The line width represents the data flow from the source node to the destination node.

cross area as the optimization objective. ILP solvers, however, have a high cost in terms of running time, which means it is unsuitable for highly interactive layouts. To balance the runtime and layout effects, NeatSankey uses a heuristic layout algorithm and a force-directed algorithm to minimize the edges' crossings and swing amplitude.

With the increasing applications of Sankey diagrams, especially in energy visualization, the demand for interaction has been promoted. Riehmann et al. [15] presented a system for visualizing electric power that supports various techniques such as overview, detail, and process tracing. However, the edges in the system suffer from the vertical-horizontal illusion [16] and are semi-automatic in layout. Later, Riehmann et al. [12] made further improvements by introducing the EnergyViz system that integrates multiple visualization techniques to explore temporal, spatial, and multi-attribute features. Muh et al. [17] used Sankey diagrams for drug-target visualization, and improved the response time, the number of crossings, and the color. When Mathis et al. [18] explored the design space of Sankey diagrams for the food-energy-water nexus, they chose to make all edges a uniform width and represent the flow magnitude numerically. Validated on tasks designed by experts, the method achieves good results. Porter et al. [19] augment the hierarchical Sankey diagram by simply examining inflow links and levels of detail. Most of these methods, however, are oriented toward specific domains, such as energy and medicine. Our NeatSankey, on the other hand, adds an edge bundling method, allowing the understanding of charts more easily and broader fields of application.

Several empirical evaluations are available for directed or hierarchical diagrams, including Sankey diagrams. Gansner et al. [5] proposed three aesthetic principles in drawing graphs: the same direction, the number of edge crossings and sharp bends, short edges and balance. To draw Sankey diagrams with good readability, Riehmann et al. [12] evaluated several aesthetic criteria, including edge crossing reduction, short edges, and straight edges. Barth et al. [20] calculated the overall blank ratio in the generated layout. These studies show the effectiveness of these evaluation metrics. Accordingly, a large-scale quantitative evaluation was conducted on 25 datasets using aesthetic principles such as the number of edge crossings, swing amplitude, and layout coverage.

2.2. *Edge Bundling*

Network graphs typically suffer from visual clutter caused by edge crossings. The visual clutter caused by dense edges can be used by bundling similar edges together to form edge bundles. Since the introduction of edge bundling by Holten [21] when drawing composite graphs, it has been a general research direction in reducing-edge clutter in graphs. At present, there are various methods to bundle edges, such as hierarchical edge bundling [21], geometry-based methods that use a control mesh [22, 23], force-directed edge bundling [24], and image-based methods [25, 26]. Bundles provide a good overview of connectivity in dense graphs as the confusion caused by edge crossings is greatly reduced.

As the scale and complexity of the network grow, with limited display space, edge-congestion [27, 9, 28] and edge-ambiguity [29, 11, 30] have become trending topics in network visualization research. Luo et al. [31] consider ambiguities in the edge-bundling process by requiring that only edges sharing common nodes can be merged, thus helping the user to trace the edges. However, this edge bundling is still based on visual metrics, such as the closeness of two edges and the degree of bundling may be very limited in dense graphs. To bun-

dle more edges, Toeda et al. [32] group nodes only if edges are connected to the same node. Similarly, the Power Graphs proposed by Bach et al. [33] use the hierarchical aggregation of nodes to reduce edge congestion while displaying the complete set of nodes and conveying precise connectivity between them. Furthermore, to support hierarchical graphs, Toeda et al. [34] bundle edges at their midpoints while converging edges at their endpoints. Wallinger et al. [35] introduced Edge-Path Bundling, which clusters each edge along a weighted, shortest path to limit its deviation from a straight line.

Most of the previous edge bundling methods bundle edges of the same width, so it is a challenge for us to apply edge bundling algorithms to Sankey diagrams in which the edges have width. Inspired by the work of Mathis et al. [18], we unify the widths of the edges and conduct a user study set by experts. Meanwhile, in order to compensate for the influence of edge region distortion caused by edge bundling on maintaining the visual display of data volume, we have made some detailed designs, such as adding hover markers to each edge to show the detailed values of the edges. Both quantitative experiments and user studies show that our method can reflect data flow better.

In summary, since the main function of Sankey diagrams are to show the data flow, we need to provide a clear visual edge pattern [30]. In NeatSankey, the edges in similar directions are clustered first to reduce edge congestion better. Then a baseline is chosen to bundle the other edges in the same cluster.

3. NEATSANKEY

In this paper, a Sankey diagram is defined as a directed graph $G = (V, E)$ with n layers, consisting of a set of nodes $V = \{V^{(1)}, V^{(2)}, ..., V^{(n)}\}$ and a set of edges $E = \{E^{(1)}, E^{(2)}, ..., E^{(n-1)}\}$. Table 1 lists the notations we use in this paper. The following discussion topic is about nodes $V^{(k)} = \{v_1^{(k)}, v_2^{(k)}, ...\}$ and edges $E^{(k)} = \{e_{11}^{(k)}, e_{12}^{(k)}, ...\}$ in layer k. For brevity, we will omit the layer numbers above the right of symbols unless necessary.

In this section, we present a two-step method to draw Sankey diagrams with good readability, which is shown in Fig 1. First, a heuristic node layout and force-directed algorithm are used to determine the node position. Edge bundling is applied to the generated Sankey diagrams, producing less ambiguous edge bundling.

3.1. Node Positioning

We propose a two-stage algorithm based on node positioning. First, the node sequences in every layer are reordered to reduce crossover according to the widths of the edges. Second, to minimize excessive overlap between the intersecting edges, a force-directed model is designed to adjust the distance of nodes, and then the final position will be determined.

Node Reordering. Taking account of the edges with different widths in Sankey diagrams, we propose a heuristic ordering algorithm. To reduce crossing between edges with similar widths, we introduce the function $\varphi\left(w_{iq}^{(k)}, w_{jp}^{(k)}\right)$ shown in Eq.1. If there is edge $e_{iq}^{(k)}$ connecting node $v_i^{(k)}$ and node $v_q^{(k+1)}$ as well as edge $e_{jp}^{(k)}$ connecting node $v_j^{(k)}$ and node $v_p^{(k+1)}$, the function will have a positive value otherwise zero. To prevent division by zero, we limit the maximum value of $\varphi\left(w_{iq}^{(k)}, w_{jp}^{(k)}\right)$.

Table 1: Key notations in this paper

Symbol	Meaning
$e_{ij}^{(k)}$	Edge connecting node $v_i^{(k)}$ in layer k and node $v_j^{(k+1)}$ in layer $k+1$
$v_{ij}^{(k)}$	Meta-node (see Fig 3) connected to the edge $e_{ij}^{(k)}$
$w_{ij}^{(k)}$	Width of the edge $e_{ij}^{(k)}$, fixed by how much flow it represents
$\alpha_{ij}^{(k)}$	Angle of the edge $e_{ij}^{(k)}$
$W_i^{(k)}$	Sum of the widths of all edges incident on the node $v_i^{(k)}$ in $E^{(k)}$
$f_{i,j}^{(k)}$	Force between node $v_i^{(k)}$ and node $v_j^{(k)}$ in layer k
$d_{i,j}^{(k)}$	Distance between node $v_i^{(k)}$ and node $v_j^{(k)}$ in layer k
$s^{(k)}$	Distance between layer k and layer $k+1$
X	Width of canvas
Y	Height of canvas
$\lvert\cdot\rvert$	Size of a set or absolute value

$$\varphi\left(w_{iq}^{(k)}, w_{jp}^{(k)}\right) = \begin{cases} \frac{1}{\left|w_{iq}^{(k)} - w_{jp}^{(k)}\right|} & e_{iq}^{(k)}, e_{jp}^{(k)} \in E^{(k)} \\ 0 & otherwise. \end{cases} \tag{1}$$

From the perspective of the visual effect, the crossing between edges of similar width will lead to severe visual clutter, while different width edges can help alleviate it. Although the width of the edge is considered by Riehmann et al. [15], the weight is directly assigned according to the edge width, which may result in a large number of overlapping thin edges. Therefore, an objective function Eq.2 is defined considering the edge width to find the crossing and reduce the crossing of the edges with the similar width. As shown in Fig 4, we need to find the appropriate sequence of nodes by switching the order of nodes so that the value of the objective function is as small as possible. The simulated annealing method [36] is used to optimize the order and reduce time consumption compared with the iterative algorithm [37].

$$min \sum_{i=1}^{|V^{(k)}|-1} \sum_{j=i+1}^{|V^{(k)}|} \left(\sum_{p=1}^{|V^{(k+1)}|-1} \sum_{q=p+1}^{|V^{(k+1)}|} \varphi\left(w_{iq}^{(k)}, w_{jp}^{(k)}\right) \right) \tag{2}$$

Force Direction. The order is determined after Node Reordering. We employ the force-directed algorithm to determine the final position (x, y coordinates) in the canvas and reduce the overlap crossing edges. The distance of nodes and layers should be considered. Inspired by the Fruchterman-Reingold algorithm [38] which reflects the connection between nodes by introducing attraction and repulsion, we present our two substages force-directed algorithm. As shown in Fig 4, the distance between nodes in each layer is adjusted first and then the distance between each layer.

First, the y coordinate of nodes in the same layer will be determined. Nodes with similar widths are kept as far away as possible to alleviate visual clutter. Initial y coordinates of nodes are assigned to each layer according to the size and the sequence of nodes. Since the order of nodes has been settled, we limit the activity boundary of nodes to a specific range according to the

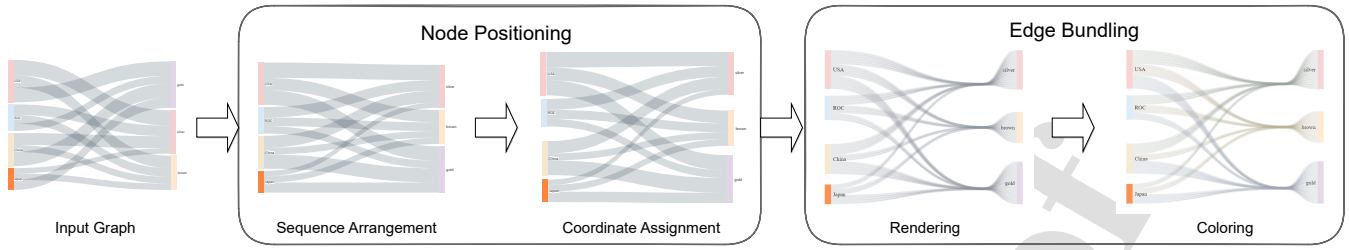


Fig. 2: An overview of our method. NeatSankey is invented to generate a neater graph when data is complex. It divides the Sankey diagrams drawing into two steps. The first step is to use a heuristic layout algorithm and a force-directed algorithm to minimize the crossings and swing amplitude of the edges. The second step is to apply an edge bundling algorithm without ambiguity based on hierarchical clustering to make edge crossings less confusing.

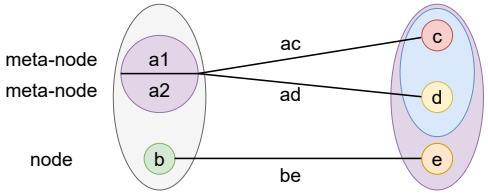


Fig. 3: The illustration of nodes, meta-nodes and edges. Node b is a node, while node a is divided into meta-node a1 and a2 which are the source nodes of edge

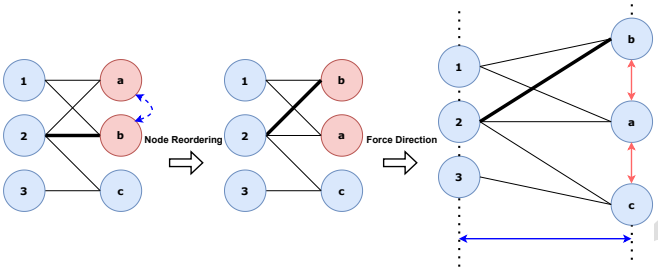


Fig. 4: The Node Reordering algorithm takes account of the edges with different widths in Sankey diagrams to reduce the crossing of a similar width. The Force-directed algorithm introduces the attraction and repulsion to determine the final position and reduce the overlap of the crossed edges.

size of nodes and ensure that there is no overlap between the activity boundary of nodes. The activity boundary is the fixed scope that a node can be placed in. That is, the node can't be placed beyond the boundary to maintain order and avoid overlapping. For nodes of the same layer, the repulsive force will be apparent between nodes with similar widths, while the attractive force will be evident between nodes with a significant difference in width. For node v_i and node v_j, the edge width difference is defined as

$$\Delta W_{i,j} = \Delta W_{j,i} = \left| \frac{W_i - W_j}{W_i + W_j} \right| \tag{3}$$

To describe it, we define attractive forces and repulsive forces as

$$fa_{i,j}\left(d_{i,j}\right) = \frac{d_{i,j}^2}{\rho_{i,j}}, \; fr_{i,j}\left(d_{i,j}\right) = \frac{-\rho_{i,j}^2}{d_{i,j}} \tag{4}$$

where $\rho_{i,j}$ is

$$\rho_{i,j} = \omega_{i,j} \sqrt{\frac{Y}{|V^{(k)}|}} \tag{5}$$

and $\omega_{i,j}$ is

$$\omega_{i,j} = \vartheta \frac{1}{\Delta W_{i,j}} \tag{6}$$

related to the edge width and the distance between nodes, ϑ can be adjusted by experiment. To prevent division by zero, we limit the maximum value of ω.

The median of all $\Delta W_{i,j}$ in layer k is chosen as ΔW_* to distinguish attraction or repulsion, then

$$f_{i,j}\left(d_{i,j}\right) = \begin{cases} fa_{i,j}\left(d_{i,j}\right) & \Delta W_{i,j} \geq W_* \\ fr_{i,j}\left(d_{i,j}\right) & \Delta W_{i,j} < W_* \end{cases} \tag{7}$$

Note that when the difference between the two edge widths goes larger, the value of ω approaches 1, and the equation is translated into the basic Fruchterman-Reingold equation [38]. When the difference is small, ω goes to infinity, and ρ also goes to infinity. For attraction, it will approach zero, and for repulsion, it will approach infinity, as expected. Based on it, we use the simulated annealing method for several iterations to determine the y coordinates of all nodes in each layer and reduce time consumption.

After the y coordinates of nodes in each layer are determined, all nodes in each layer are regarded as one big node then to determine the distance between layers. At this point, the edges of each layer are kept as flat as possible, especially if there are too many edges. Similar to the previous work, we restrict the nodes to a certain range according to the fixed node order and adopt the force-directed algorithm. Since the relationship between nodes in this sub-stage is relatively simple, we only introduce the effect of attraction to avoid the complex calculation. The number of edges connected between two big nodes $|E^{(k)}|$ is set to be a parameter. The greater it is, the smaller the attractive force is. Let $\omega^{(k)} = \vartheta|E^{(k)}|$ in Eq.6. Obviously, the larger $|E^{(k)}|$ is, the less attraction is, and vice versa.

3.2. Edge Bundling

In order to understand the difficulties existing in our method better, the ambiguities incurred in edge bundling will be further described. The two ambiguities have occurred in some methods and were concretely defined by Wallinger et al. [35].

Path Endpoint Ambiguity. Fig 6 shows this ambiguity. Edge bundling is used to reduce the confusion caused by many edges crossing. However, after bundling edges, it can be difficult to tell whether there is a path existing between the nodes, for example, c does not connect d and f.

Edge Crossing Ambiguity. Fig 7 shows this ambiguity. When two edges cross, ambiguity may exist if the crossing angle is shallow [39]. Edges that cross at nearly 90° are less likely to be confusing than those crossing at acute angles.

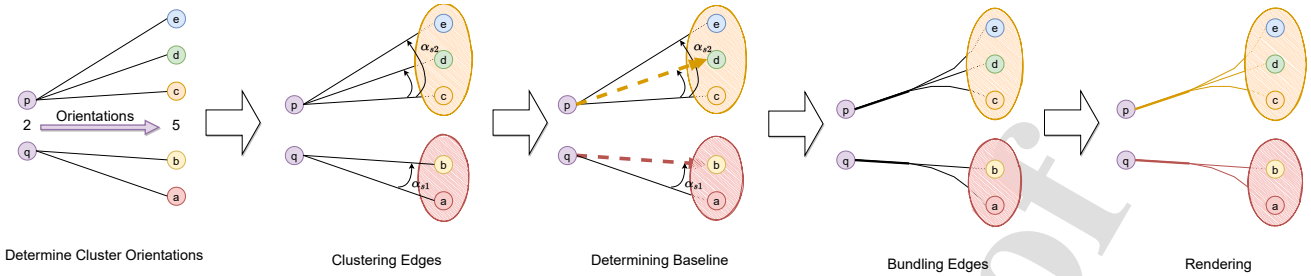


Fig. 5: An overview of our edge bundling method. Our edge bundling method has five steps: Determining cluster orientations, Clustering edges, Determining baseline, Bundling edges, and Rendering.

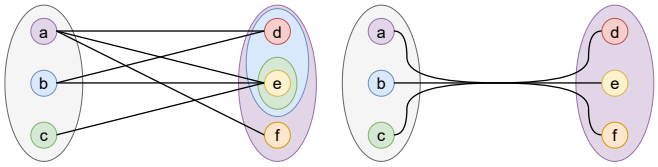


Fig. 6: Path endpoint ambiguity. A connection does exist between (b, f), (c, e) and (c, f) in (a). When edges are strongly bundled, we might get false perceptions.

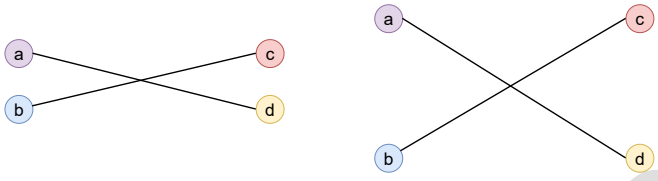


Fig. 7: According to Huanget al. [39], the left is more confusing than the left when two edges cross.

In graph visualization, edge-ambiguity problems usually lead users to perceive incorrect relations between nodes or require intensive effort to read the graph. For Sankey diagrams, the exact information of each edge is needed. Compared to the conventional methods, the main idea of our method is to bundle edges that share at least one endpoint. Meanwhile, since edges in Sankey diagrams have different widths and are not working well when using previous edge bundling methods directly, the granularity of the edge drawing is adjusted. Fig 5 illustrates an overview of our method. We propose a specific edge bundling technique for Sankey diagrams, which has the following five steps: Determining cluster orientations, Clustering edges, Determining baseline, Bundling edges, and Rendering.

The determining clustering orientations step chooses one direction by calculating the relationship between the number of source nodes and destination nodes. The clustering edges step is performed by grouping edges with the same source node or destination node into one cluster. In this process, the maximum of the directional angular deviation is limited in a cluster. The determining baseline step is to select a wide and flat line in each cluster. The bundling edges step is based on the selected baseline bundling the other edges in that cluster. The rendering step is to adjust the granularity of the edge drawing and choose the same color for each cluster to provide an excellent visual perception for the user.

Determining Cluster Orientations. Before clustering is performed, we determine the direction of clustering first, i.e., whether to bundle near the source node (from the source node to the destination node) or the inverse. In NeatSankey, we specify that the bundle direction is consistent between layers aiming to have a better aesthetic result. The cluster direction is determined by comparing $|V^{(k)}|$ and $|V^{(k+1)}|$. If $|V^{(k)}| \leq |V^{(k+1)}|$, the clustering direction is from the source node to the destination node. Otherwise, the clustering direction is from the destination node to the source node. The clustering direction we discuss is from the source node to the destination node below.

Clustering Edges. The bundling points are usually in two places, one at the midpoint of the edge and the other at the endpoint. Usually, bundling at the midpoint of the edges can display the main structure of the graph better, but some edges can be difficult to identify. To avoid the ambiguity that occurred in edge bundling, only edges with the same source or destination node can be bundled together. Our method is an edge direction clustering method based on angular neighborhood, which is able to deal with direction clustering problems with a small computational cost.

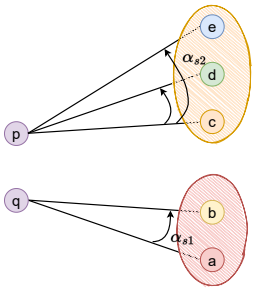


Fig. 8: Edge direction clustering based on angular neighborhoods.

Each edge has two endpoints corresponding to a source meta-node and a destination node. To minimize ambiguity, only edges with the same source meta-node or destination node can be bundled. The core of edge direction clustering is to process the directional angles of edges so that the edges with less difference in directional angles are classified into the same cluster. As shown in Fig 8, for the edge in set $E^{(k)}$ connecting the node $v_i^{(k)}$, the direction angle $\alpha_{ij}^{(k)}$ of each edge $e_{ij}^{(k)}$ is calculated.

Through the direction angle $\alpha_{ij}^{(k)}$, the edges are clustered again. Since the order of the nodes is fixed (determined by the node positioning), the angle $\Delta\alpha_{p,p+1}$ between the adjacent edge in the set $E^{(k)}$ is calculated according to the order. The edges are classified satisfying Eq.8 as one cluster and obtain a bundling cluster $c_i^{(k)} \in C^{(k)}$ and $c_i^{(k)} = \{e_{ij}^{(k)}, e_{ij+1}^{(k)}, ...\}$.

$$A_i = \sum_{p=1}^{|c_i|-1} \Delta\alpha_{p,p+1} = \sum_{p=1}^{|c_i|-1} |\alpha_{ip} - \alpha_{ip+1}| \leq \Upsilon \tag{8}$$

where Υ denotes the restricted angular neighborhood and $\Delta\alpha_{p,p+1}$ is the angle between e_{ij} and e_{ij+1}. In accordance with

node positioning, $\Upsilon = 30°$ is used as a constraint for the edges to join the bundling clusters, which is suitable to create a bundle.

Determining Baseline. When clustering, direction angles are used to limit the range of clustering in order to avoid too many edges contained in one cluster. An edge is chosen to represent the main direction in a bundling cluster to avoid time-consuming calculations for better interactivity. Then, the other edges will be bound in the same bundling cluster to the baseline in the bundling edges step. For the orientation of the baseline, we want it to preferably be in the middle of the cluster so that the bending of the other edges can be minimized when bundling. The edges are added to the candidate edge set $c_{i*} = \{e_{iq}, e_{iq+1}, ...\}$ when their direction angles meet Eq.9

$$\frac{\overline{\alpha_i} + \underline{\alpha_i}}{2} - \delta \leq \alpha_* \leq \frac{\overline{\alpha_i} + \underline{\alpha_i}}{2} + \delta \tag{9}$$

where $\overline{\alpha_i}$ is the maximum of the edge angle in bundling cluster c_i while $\underline{\alpha_i}$ is the minimum, and δ is set 10° through experiments.

We not only select some edges in the more intermediate part of the node cluster but also choose the edges with wider widths. The edges in the candidate edge set c_{i*} are sorted in descending order by width w, and the edge with the largest width among them is selected as the baseline. Iterate over all clusters, and we will get a baseline in each bundling cluster.

Bundling Edges. Compared with the classic method like force-directed edge bundling and image-based edge bundling, our technique is based on the geometrical structure. The Sankey diagram is a hierarchical network structure that is very regular in structure and contains fewer edges in a bundling cluster. Based on the simplicity in structure, we only need to converge the other edges in the node clusters toward the baseline and then flow into the corresponding nodes separately without massive calculation.

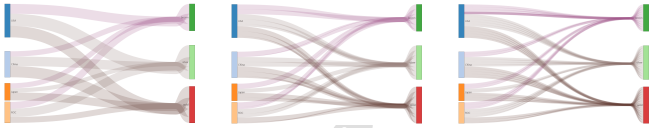
In each layer, the distance of edge expansion is consistent to keep the symmetrical beauty and structural similarity as much as possible. A good expansion point contributes a lot to the reduction of edge crossing ambiguity for it increases the crossover angle. In different layers, the distance is related to the direction angles and widths of the edges. Let $\eta_i^{(k)}$ be the expansion coefficients. It is defined as follows.

$$\eta_i = \frac{1}{\lambda} \cdot \sum_{q=1}^{|c_i|} w_{iq} \cdot |\Delta\alpha_{q,*}| \tag{10}$$

$\Delta\alpha_{q,*}$ is the angle between edge e_q and the baseline in bundling cluster c_i. Because the symbol of the angles only indicates the direction, but not the magnitude, the absolute value needs to be calculated. Here we consider the influence of weights w_{iq} on the performance and compute the weighted sum. λ is set to 5 through experiments to adjust the sensitivity.

The final expansion distance from left to right, is defined as $x_*^{(k)} = \eta^{(k)} \cdot s^{(k)}$. Finally, we use Bezier's method to draw the edges in D3.js.

Rendering. After bundling edges, the result seems a little unnatural since edges with different widths are bundled together and the overlapping region is not regular enough. For this reason, we propose a new method to bundle edges that are different in widths. Instead of drawing edges all at once with the original width, one wide edge is replaced with a combination of multiple thin edges of the same width. When the width of the edge is less than the division width, no division is performed. The width of the thin edge d_γ is set as a constant quantity and can be edited by the users depending on their datasets. Typically, we set $d_\gamma = 8$ while d_γ is measured in pixel units. Fig 9 provides some illustrations of the edges divided into different levels. Typically, we suggest d_γ set close to the minimum edge width like (c).



(a) $d_\gamma = max$ (b) $d_\gamma = 16$ (c) $d_\gamma = 8$

Fig. 9: Examples of Sankey diagrams generated on the edge bundling using different parameter d_γ.

With the clusters divided above, colors are added simply. In the same layer, edges in the same cluster are drawn with the same color. But the clusters of different layers can have the same color, and there is no correlation between the colors of different layers.

4. EVALUATION

In this section, we compare NeatSankey with other Sankey diagram algorithms through quantitative analysis, a user study, and a case study. The experiment data is available at https://github.com/NeatSankey/NeatSankey-supp.

4.1. Quantitative Analysis

Metrics. The following three metrics are used to evaluate the readability of the generated Sankey diagram quantitatively:

- Crossing Number(CN) is an aesthetic metric of diagrams proposed by Gansner et al. [5] for measuring the number of edge crossings in a generated layout. Whether the two edges cross is judged between the coordinate of the endpoint of the edge and count the number of all the crossed edges. The smaller the value of CN is, the fewer edges intersect in the layout. To some extent, it reflects the clutter.
- Swing Amplitude(SA). Gansner et al. [5] proposed the shortest edge as the metric. Swing amplitude further extends the shortest edge. The swing amplitude of the edge can be obtained by calculating the slope of the edge. We approximate the slope as the swing amplitude.
- Layout Coverage(LC). Barth et al. [20] proposed this metric in order to measure the total proportion of blank space in the generated layout. The generated images are converted without text into grayscale images and obtained LC by calculating the number of pixels in the images. Therefore, a smaller LC often means clearer visuals.

Data & Settings. A total of 25 datasets were used for the overall assessment, including 10 real-world (see Table 3 top 5)

and 15 randomly generated datasets (see Table 3 last 5). Multiple datasets were selected with different layers and node numbers. We implemented BC, ILP, and our algorithm for each dataset. Three quantitative metrics (CN, SA, and LC) were calculated to measure the result. In the experiment, the BC algorithm and our algorithm are implemented in Python, and the ILP algorithm is implemented according to the ILP solver provided by its author. We then render the resulting chart with d3.js. All experiments were conducted on a computer with an i7-9750H kernel, 2.60GHz, 16.0G RAM, and Windows 10 operating system.

Result. For the metrics CN, SA, and LC, we quantitatively calculate the corresponding values. Boxplots show the comparison of our algorithm with the BC and ILP algorithms in Fig 10. For the metric CN, a smaller CN means fewer edge crossings. The results of our method are similar to OptimSankey, which has the best performance in terms of the median. For the metric SA, we believe it is more representative of the overall confusion than CN. A smaller SA value means that the slope of the edge is smaller. Due to the effectiveness of our node positioning algorithm, our results in SA outperform the other two techniques. Additionally, LC measures spatial coverage. When the value of LC is small, it means that there are many blank spaces. If there are many edge crossings in a thin space, it may lead to serious visual clutter. Thanks to our edge bundling method, the blank areas in the layout are significantly increased, and the visual clutter is greatly reduced.

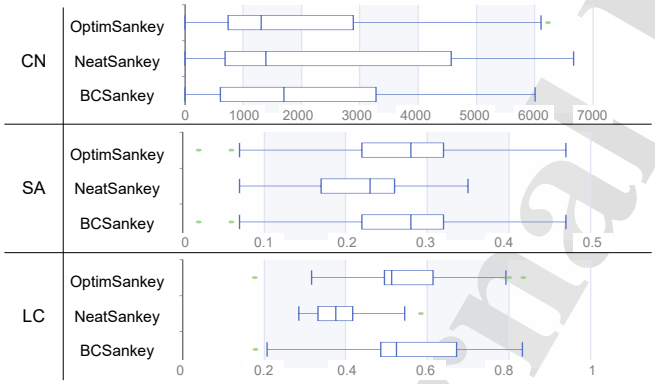


Fig. 10: Comparison of three metrics (crossing number, swing amplitude and layout coverage) and three methods (OptimSankey, NeatSankey and BCSankey) on 25 datasets, small values are better.

In experiments for measuring running time (see Table 2), the results show that NeatSankey takes longer to obtain results than the traditional method by Sugiyama et al. However, even in that case, we can still get the result by NeatSankey within 1 second in most cases. The time cost of the ILP method, on the other hand, is much higher in most cases, taking more than 10 seconds to obtain the results.

4.2. user study

We conducted a user study comparing the readability of NeatSankey with other Sankey diagrams implemented by different algorithms.

Participants and Apparatus. A total of 30 volunteers were invited. Among them, 25 students (17 males and 8 females, aged 19-21, 15 majored in computer science, 5 majored in economics, and 5 majored in mechanisms) and 5 teachers (4 males and 1 female, aged 25-40) from a university participated in the survey. The teachers had Sankey diagram drawing experience, and the rest didn't.

Table 2: Running time of the three algorithms

dataset	BCSankey cost(s)	OptimSankey cost(s)	NeatSankey cost(s)
1	0.035	3.00	0.371
2	0.051	13.00	0.336
3	0.070	16.00	0.438
4	0.238	69.00	0.750
5	0.297	20.00	0.387
6	0.129	11.00	0.563
7	0.051	24.00	0.430
8	0.043	7.00	0.223
9	0.680	24.00	0.934
10	0.063	1.00	0.211

Conditions. The readability of NeatSankey was compared with the integer linear programming algorithm [6] and the barycentric heuristic algorithm [7]. To reduce the influence of coloring on the results, all Sankey diagrams were drawn in gray. The captions of BC and ILP were used only for recording and analyzing results. To avoid user bias, the captions of the algorithm used for Sankey diagrams wouldn't be shown during the study. The diagram drawn by different algorithms appeared at random.

Datasets. Our user study will be carried out on 5 datasets. In order to accurately judge the practicability of Sankey diagrams, real-world datasets (see Table 3 top 5) are used. Our dataset is about energy flows, government budgets, etc., all of which are common problems in life. The parameter max / min width represents the ratio of the maximum width to the minimum width of the edge.

Table 3: Five real-world datasets and typical five generated datasets

dataset	layers	nodes number	links number	max / min width
City of Oakland Budget	3	40	67	5810.41
Canadian Energy Flows	4	40	85	230.0
Production of Products	4	19	21	7.97
Icon Made	4	40	58	8.00
Energy Source	4	17	42	1300.0
auto-generated-1	4	34	108	93.0
auto-generated-2	5	95	191	37.0
auto-generated-3	6	97	219	78.0
auto-generated-4	7	146	319	421.0
auto-generated-5	8	153	340	193.0

Tasks. Mathis et al. [18] proposed 7 tasks through reviewing Sankey diagrams literature and expert discussions. To fully evaluate the readability of Sankey diagrams, 3 tasks are selected to compare the performance between different algorithms. All participants were asked to complete three tasks designed to assess the readability of Sankey diagrams. The tasks and reasons

are as follows.

T_{flow} It evaluates the observing ability of the Sankey diagrams data flow. The flow is analyzed based on the work of Lee et al. [40]. A data flow path is given, and the participants are asked to judge whether the path exists in the Sankey diagrams.

T_{rel} It assesses the understanding of participants to the node relationships. Link analysis is performed based on the work of Lee et al.[40]. Given a node and options, participants are asked to select the data flow of these nodes.

T_{data} It is mainly about flow volume in Sankey diagrams. Based on the work of Amar et al. [9], extreme value identification is performed. For different layers of the Sankey diagrams, participants are required to select the node of the maximum value in each layer.

Procedure. The independent variables are task, Sankey diagram algorithm, and dataset, while the dependent variables are accuracy and completion time. Each participant must complete tasks for all independent variables and finish in 45 trials (3 tasks, 3 Sankey diagram algorithms, and 5 datasets). The user study involved 30 participants and 1,350 trials. To ensure that participants did not identify the algorithms used in the diagrams, the appeared order of Sankey diagrams with different algorithms was shuffled.

We did not train users before the trial (the effect of experience on the results will be analyzed later). Before starting the task, we record the user's age and major. Major can evaluate whether our Sankey diagram is friendly for non-specialists and whether it can be used with the same ease as a computer professional. In each task, participants answered 3 questions covering the Sankey diagrams with 3 different algorithms and 5 datasets. T_{flow}, T_{rel} and T_{data} are multiple choice questions. The chart was at the top of the page for each task, with four options below, and participants were asked to view the chart without any additional tool. When they have answered a question, they click a button to go to the next question. We record the participants' answers and the completion time of each question.

In addition, NeatSankey is tested by the five-point SUS system availability table. Participants were asked to rate the Sankey diagrams generated by NeatSankey based on their aesthetic preferences and perceived readability. The user study took about 40 minutes for each participant.

Results & Analysis. We evaluated the readability of each Sankey diagram by analyzing the completion time and accuracy of the three tasks proposed above. Fig 11 shows the average time taken to complete the task and its accuracy. NeatSankey outperforms the BC algorithm and OptimSankey in both accuracy and completion time. Overall, the accuracy of the three tasks is not very high, because the dataset selected is relatively complex. T_{flow} has the lowest average accuracy of the three tasks, indicating that the task is the most difficult. The performance of NeatSankey and OptimSankey is similar in terms of completion time, and the Sankey diagram designed by the BC algorithm takes the most time to observe. T_{data} takes the shortest time to complete because it only examines the ability to perceive the data size and only needs to judge the maximum value, which is relatively simple.

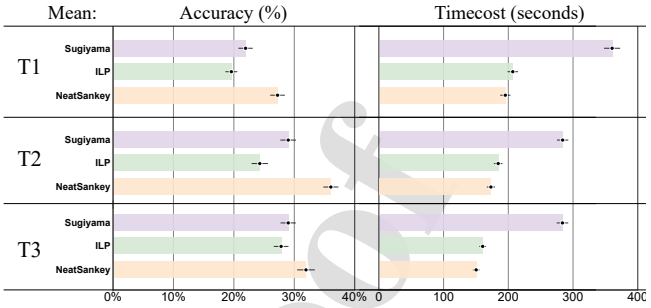


Fig. 11: For the three tasks of determining the readability of Sankey diagrams, the accuracy and time to complete the three algorithms are compared.

For readability tasks, NeatSankey performs better on average than BC and ILP. Among the tasks, the ILP, considered the optimal layout, had the lowest accuracy. We believe that the reason may be that ILP thinks the number of edge crossings and the high weight of edges, but ignores the intersections of thin edges. Our algorithm has the best performance because the effects of swing amplitude and layout coverage on visual clutter are considered. The overall average accuracy is low for T_{flow}. We suspect it is due to the complexity of the task, data flow across multiple layers, and the accumulation of numerous clutter. The overall accuracy is relatively high for T_{rel} because node relationships can be obtained without analysis of multiple layers and reducing-edge crossings improve the accuracy of the task. For T_{data}, we think that the task can be easily finished whichever the algorithms, so the accuracy of the three algorithms is similar, and all are high.

For readability tasks, NeatSankey took the least time on average. We found that though ILP was less accurate than BC, it was better in terms of completion time. We think that ILP can obtain the optimal layout through the ILP algorithm, but there will be some ambiguity. For example, a large number of edges crossed at a small angle will lead to ambiguity, and this problem is solved through an edge bundling algorithm. For T_{flow}, the task is complicated, so the overall completion time is long. For T_{rel}, ILP and NeatSankey are close in completion time. We believe that ILP also has a similar effect of reducing-edge swing amplitude through node positioning. For T_{data}, ILP and NeatSankey times are close, indicating that users are sure of the answer.

In addition, people in different majors were analyzed to test the three algorithms. As can be seen from Fig 12(a), for teachers with Sankey diagrams design experience, the accuracy of the ILP algorithm is the lowest, and the accuracy of the BC algorithm is similar to our algorithm at a high level, indicating that our algorithm can improve the accuracy for experienced people. For people in computer science majors, their average score is the highest. Our algorithm has also achieved the highest accuracy. We believe that it is because computer students are often in touch with charts similar to the Sankey diagram, which can arouse resonance. Economics students and mechanical students have the same accuracy rate, with the highest accuracy rate of our algorithm, which shows that our algorithm is not limited to people in computer major. From Fig 12(b), we notice that the teacher takes the shortest time, probably because they are familiar with the Sankey diagrams. On the whole, the

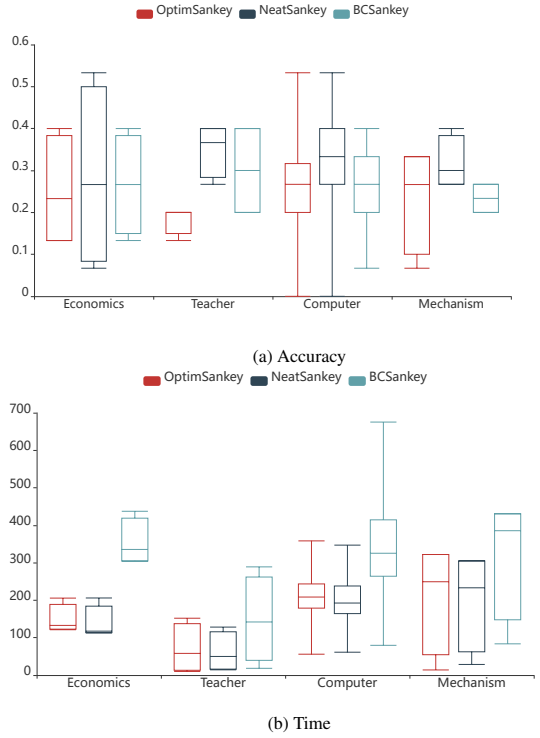


Fig. 12: The accuracy rate and completion time of the users of different majors were answered in three algorithms.

BC algorithm takes the longest time, while the ILP algorithm can improve the efficiency of judgment. However, our algorithm has the shortest completion time, which can well display data flow and help the task completion of the Sankey diagrams. Furthermore, there seems to be some relationship between accuracy and majors. Whether between professional scholars and students, or between different majors, the accuracy rates differ less, which means that our Sankey diagrams are easy to understand.

The SUS system availability table questionnaire is a standardized questionnaire developed by Brook et al. [19]. It is a famous usability test questionnaire for final subjective evaluation and can be used as a measure of usability. Fig 13 shows the SUS questionnaire scores of the participants, including total score, usability score, and learnability score, respectively. Our method outperforms OK for readability scores and comes close to GOOD. However, the learnability needs to be strengthened. We think that the complex dataset and the complicated questions may also be the factors leading to a low learnability score.

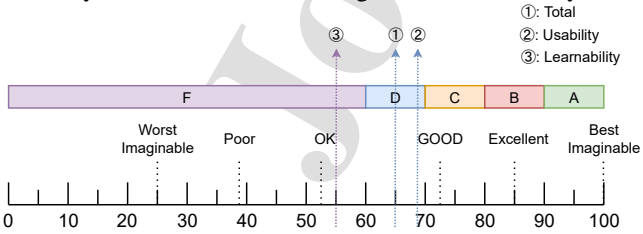


Fig. 13: The total score, usability score and learnability score of NeatSankey were obtained by the SUS questionnaire. Scores are counted on a percentage scale.

Based on the results from user study, we can find out these phenomena below: With NeatSankey, users can find the correct answer more quickly; NeatSankey brings the change that users can sense; The influence of NeatSankey is stable with low variance.

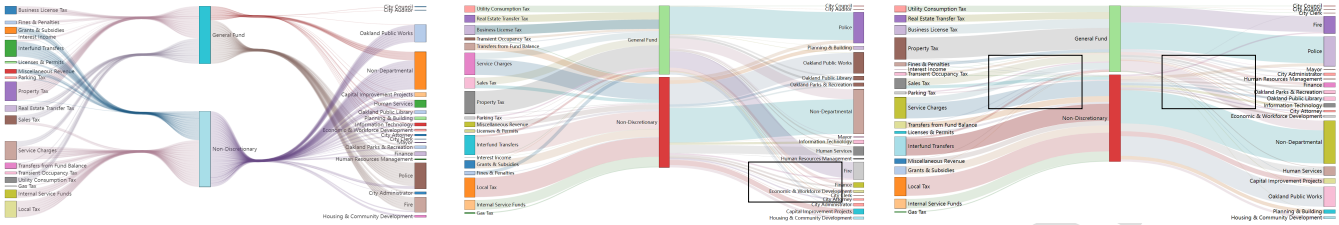
4.3. Case Study

In this section, our results are compared with state-of-the-art techniques such as BCSankey and OptimSankey. A case study on two datasets is conducted to further demonstrate the effectiveness and tractability of NeatSankey.

Fig 14 shows the City of Oakland budget dataset through Sankey diagrams generated by our method, the BC method, and the ILP method, respectively. Compared with the BC method (b) and the ILP method (c), our method can show the clearer budget flow. The BC method (b) has dense edge crossings in the black-boxed area marked in the bottom right corner. There is also a doubt about whether the edges flow from the node General Fund to the node Finance, City Clerk, City Attorney, and others. Also, the multiple thick edge crossings bring a destructive visual impact. This may be caused by an indiscriminate node reordering algorithm, which doesn't take into account the edge widths. Although the ILP method (c) solves the clutter caused by the large area of edge crossings, it generates new confusion simultaneously. On the left side (see left black box) and right side (see right black box) of figure (c), multiple edges cross in a dense space so that the destination of node General Fund is difficult to determine, such as Mayor, City Administrator, Finance, Information Technology, and others.

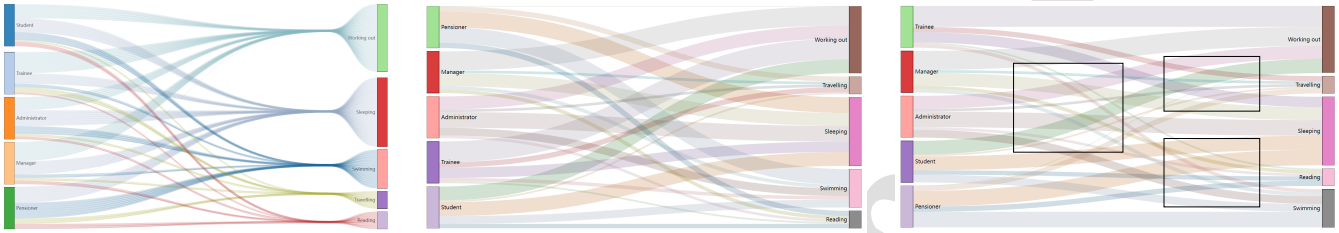
Meanwhile, some sources of the node General Fund are also diffcult to recognize. The edges cross slightly, which may lead to edge ambiguity. The more it is, the harder it is to determine the data flow direction. But in our algorithm (a), the angle of every edge crossing is close to 90°, preventing this problem. With the force-directed algorithm, the nodes' relative positions can be optimized. In this case, a better spatial layout is obtained. From node General Fund to node Finance or City Clerk, we can always see a straighter edge connecting two nodes. In addition, the figures generated by our method are more readable because the improved node layout and edge bundling algorithm make the whole graph less spatial coverage, larger white space, and less visual clutter. Though, it also shows a problem with NeatSankey. In figure (b), we can see how much percent of node General Fund comes in from the various elements on the leftmost (input) row. In figure (a), however, it is impossible to see this since the bundling aggregates node Business License Tax with several other nodes in the leftmost input layer. All we can see is a split of node General Fund into two groups of inputs. Assigning different colors to different types of flow can solve this problem.

The Holidays dataset mainly describes the activities of people with different occupations during holidays and is displayed in Fig 15. This dataset is relatively simple, with only two layers, but it can illustrate many potential problems of the Sankey diagram. Here, both the BC algorithm (b) and the ILP algorithm (c) suffer from serious ambiguity. The ILP algorithm considers the weights of the edges and solves the optimal solution by integer programming, mainly reducing the crossover of thick edges with thick ones. However, from the black box in Fig 15,



(a) NeatSankey (b) BCSankey (c) OptimSankey

Fig. 14: The City of Oakland through Sankey diagrams which are generated by our method, the BC method and ILP method, respectively



(a) NeatSankey (b) BCSankey (c) OptimSankey

Fig. 15: Holidays through Sankey diagrams which are generated by our method, the BC method and ILP method, respectively

many edges cross each other at a smaller angle which causes local confusion. Meanwhile, the excessive aggregation of edges also causes the same visual confusion. Our algorithm adjusts the order as well as the spacing of nodes and bundles the edges from the same source node or the same destination node. The improved Sankey diagram not only has a better field of view with less spatial coverage but also obtains edges crossing close to 90°, dramatically reducing the possibility of visual clutter.

From the three studies above, we can see that our method reduces clutter visually. Quantitative evaluation metrics (CN, SA, LC) demonstrate the effectiveness of our method. user study shows volunteers can finish three tasks well with NeatSankey in a short completion time with high accuracy. Our generated Sankey diagram (Fig 14, Fig 15) in the Case Study is more clear. However, we also find some limitations.

Many Sankey diagrams are drawn with the width of the edges indicating the volume of the data low, which is the primary way the Sankey diagrams convey cardinality. Our NeatSankey greatly improves readability by using the edge bundling algorithm. In order to compensate for the influence of edge region distortion caused by edge bundling on maintaining the visual display of data volume, we have made some detailed designs. In this paper, our method can reflect the flow volume by the width of the edge endpoints. At both ends of the edge, that is, the part close to the nodes, the width of the edge doesn't change, presenting the amount of data flow. Meanwhile, in order to achieve a balance between aesthetics and the original characteristics of Sankey diagrams, we add hover markers to show the flow of the edges. Fortunately, through tasks ($T_{flow}, T_{rel}, T_{data}$) proposed by previous users and experts in the user study, we found that volunteers could still accomplish these tasks well with NeatSankey despite the change in the width of the middle part. The results show that it outperforms the other two methods in accuracy and time, proving that our method can effectively reflect data flow. Meanwhile, we can see that for the 10 datasets shown in Table 2, our method is slightly less computationally efficient than the initial method BCSankey but substantially better compared to OptimSankey. It isn't a significant limitation because understanding the graph accurately should be more important than a short response time [20].

5. DISCUSSION AND FUTURE WORK

We propose a method to reduce visual clutter in Sankey diagrams when the data volume is large, proposing a newly invented method to generate neater diagrams. Compared to previous methods, our approach achieves a balance between effectiveness and time, improving the readability observably. The major contribution of this technique is that it minimizes the cross area as well as swing amplitude of the edges and makes edge crossings less confusing by two-step algorithm. It is worth noting that our method can bundle edges with different widths. To the best of our knowledge, no previous studies have applied edge bundling method in the field of Sankey diagrams.After bundling, we can still find the edge from the source node to the destination node, because we only bundle the edges which have the same source node or the destination node. It is worth noting that our edge bundling algorithm will not cause secondary ambiguity. A quantitative evaluation, user studies and a case study demonstrate the effectiveness and practicability of NeatSankey.

Although NeatSankey has improved the readability of Sankey diagrams, it still has several limitations and we are ready to solve it in future work. First, in node positioning, we improve the result by adding constraints for the force-directed algorithm. The time complexity increases after adding restriction optimization, requiring more time to reach the optimal result. Besides, our approach draws Sankey diagrams through D3.js, which lacks some customizability. Additionally, our method may cause partial confusion by distorting the width of the middle side. Finally, while our method is visually more readable than other methods, cognitive research is needed to explore the deep perception of visualization quality in Sankey diagrams.

References

[1] Sankey, M. Introductory note on the thermal efficiency of steam-engines. In: Minutes of proceedings of the institution of civil engineers; vol. 134. 1898,.

[2] Schmidt, M. The sankey diagram in energy and material flow management: part ii: methodology and current applications. Journal of industrial ecology 2008;12(2):173–185.

[3] J.D, R. Visualize your budget flows with a sankey diagram. 2022. https://www.getrichslowly.org/sankey-diagrams/, Accessed:2022.9.25.

[4] Policy, CE. Us energy history visualization. 2022. https://us-sankey.rcc.uchicago.edu/, Accessed:2022.9.25.

[5] R, GE, Koutsofios E, NSC. A technique for drawing directed graphs. IEEE Transactions on Software Engineering 1993;19:214–230.

[6] Zarate, DC, Le Bodic, P, Dwyer, T, Gange, G, Stuckey, P. Optimal sankey diagrams via integer programming. In: 2018 IEEE pacific visualization symposium (PacificVis). IEEE; 2018, p. 135–139.

[7] Sugiyama, K, Tagawa, S, Toda, M. Methods for visual understanding of hierarchical system structures. IEEE Transactions on Systems, Man, and Cybernetics 1981;11(2):109–125.

[8] Brandon Mathis, YMMRM. Exploring the design space of sankey diagrams for the food-energy-water nexus. IEEE computer graphics and applications 2019;.

[9] Von Landesberger, T, Kuijper, A, Schreck, T, Kohlhammer, J, van Wijk, JJ, Fekete, JD, et al. Visual analysis of large graphs: state-of-the-art and future research challenges. In: Computer graphics forum; vol. 30. Wiley Online Library; 2011, p. 1719–1749.

[10] Bachmaier, C. A radial adaptation of the sugiyama framework for visualizing hierarchical information. IEEE Transactions on Visualization and Computer Graphics 2007;13(3):583–594.

[11] Nguyen, Q, Eades, P, Hong, SH. On the faithfulness of graph visualizations. In: International Symposium on Graph Drawing. Springer; 2012, p. 566–568.

[12] Alemasoom, H, Samavati, FF, Brosz, J, Layzell, D. Interactive visualization of energy system. In: 2014 International Conference on Cyberworlds. IEEE; 2014, p. 229–236.

[13] Garey, MR, Johnson, DS. Crossing number is np-complete. SIAM Journal on Algebraic and Discrete Methods 1983;4(3):312–316.

[14] Jünger, M, Mutzel, P. 2-Layer Straightline Crossing Minimization: Performance of Exact and Heuristic Algorithms. Graph Algorithms And Applications I; 1998.

[15] Riehmann, P, Hanfler, M, Froehlich, B. Interactive sankey diagrams. In: IEEE Symposium on Information Visualization, 2005. INFOVIS 2005. IEEE; 2005, p. 233–240.

[16] Fineman, M. The nature of visual illusion. Courier Corporation; 2012.

[17] Ginoga, MFAH, Trisminingsih, R, Kusuma, WA. Drug-target visualization on ijah analytics using sankey diagram. In: 2020 International Conference on Computer Science and Its Application in Agriculture (ICOSICA). IEEE; 2020, p. 1–6.

[18] Mathis, B, Ma, Y, Mancenido, M, Maciejewski, R. Exploring the design space of sankey diagrams for the food-energy-water nexus. IEEE computer graphics and applications 2019;41(2):25–34.

[19] Brooke, J. Sus: a retrospective. Journal of usability studies 2013;8(2):29–40.

[20] Barth, L, Fabrikant, SI, Kobourov, SG, Lubiw, A, Nöllenburg, M, Okamoto, Y, et al. Semantic word cloud representations: Hardness and approximation algorithms. In: Latin American Symposium on Theoretical Informatics. Springer; 2014, p. 514–525.

[21] Holten, D. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. IEEE Transactions on visualization and computer graphics 2006;12(5):741–748.

[22] Cui, W, Zhou, H, Qu, H, Wong, PC, Li, X. Geometry-based edge clustering for graph visualization. IEEE transactions on visualization and computer graphics 2008;14(6):1277–1284.

[23] Lambert, A, Bourqui, R, Auber, D. Winding roads: Routing edges into bundles. In: Computer graphics forum; vol. 29. Wiley Online Library; 2010, p. 853–862.

[24] Holten, D, Van Wijk, JJ. Force-directed edge bundling for graph visualization. In: Computer graphics forum; vol. 28. Wiley Online Library; 2009, p. 983–990.

[25] Ersoy, O, Hurter, C, Paulovich, F, Cantareiro, G, Telea, A. Skeleton-based edge bundling for graph visualization. IEEE transactions on visualization and computer graphics 2011;17(12):2364–2373.

[26] Hurter, C, Ersoy, O, Telea, A. Graph bundling by kernel density estimation. In: Computer graphics forum; vol. 31. Wiley Online Library; 2012, p. 865–874.

[27] Wong, N, Carpendale, S, Greenberg, S. Edgelens: An interactive method for managing edge congestion in graphs. In: IEEE Symposium on Information Visualization 2003 (IEEE Cat. No. 03TH8714). IEEE; 2003, p. 51–58.

[28] Ellis, G, Dix, A. A taxonomy of clutter reduction for information visualisation. IEEE transactions on visualization and computer graphics 2007;13(6):1216–1223.

[29] Burch, M, Vehlow, C, Konevtsova, N, Weiskopf, D. Evaluating partially drawn links for directed graph edges. In: International Symposium on Graph Drawing. Springer; 2011, p. 226–237.

[30] Luo, SJ, Liu, CL, Chen, BY, Ma, KL. Ambiguity-free edge-bundling for interactive graph visualization. IEEE transactions on visualization and computer graphics 2011;18(5):810–821.

[31] Bruls, M, Huizing, K, Wijk, JJv. Squarified treemaps. In: Data visualization 2000. Springer; 2000, p. 33–42.

[32] Toeda, N, Nakazawa, R, Itoh, T, Saito, T, Archambault, DW. On edge bundling and node layout for mutually connected directed graphs. In: 2016 20th International Conference Information Visualisation (IV). IEEE; 2016, p. 94–99.

[33] Bach, B, Riche, NH, Hurter, C, Marriott, K, Dwyer, T. Towards unambiguous edge bundling: Investigating confluent drawings for network visualization. IEEE transactions on visualization and computer graphics 2016;23(1):541–550.

[34] Toeda, N, Nakazawa, R, Itoh, T, Saito, T, Archambault, D. Convergent drawing for mutually connected directed graphs. Journal of Visual Languages & Computing 2017;43:83–90.

[35] Wallinger, M, Archambault, D, Auber, D, Nöllenburg, M, Peltonen, J. Edge-path bundling: A less ambiguous edge bundling approach. IEEE Transactions on Visualization and Computer Graphics 2021;28(1):313–323.

[36] Steinbrunn, M, Moerkotte, G, Kemper, A. Heuristic and randomized optimization for the join ordering problem. Vldb Journal 1997;6(3):191–208.

[37] Optimization of large join queries 1988;17(3):8–17. doi:10.1145/971701.50203.

[38] Fruchterman, T, Reingold, EM. Graph drawing by force-directed placement. Software: Practice and Experience 1991;.

[39] Huang, W, Hong, SH, Eades, P. Effects of crossing angles. In: 2008 IEEE Pacific Visualization Symposium. IEEE; 2008, p. 41–46.

[40] Lee, B, Plaisant, C, Parr, CS, Fekete, JD, Henry, N. Task taxonomy for graph visualization. In Proceedings of BEyond time and errors: novel evaLuation methods for Information Visualization 2006;:82–86.