ISSN 2464-4617 (print)
ISSN 2464-4625 (online)
Computer Science Research Notes - CSRN 3301
http://www.wscg.eu
WSCG 2023 Proceedings

Evolutionary-Edge Bundling with Concatenation Process of Control Points

Ryosuke Saga[0000-0003-1528-6534]
Osaka Metropolitan University
1-1 Gakuen-cho, Naka-ku
5998531, Sakai, Japan
r.saga@omu.ac.jp

Jaeyoung Baek
Osaka Prefecture University
1-1 Gakuen-cho, Naka-ku
5998531, Sakai, Japan

ABSTRACT

Edge bundling is one of the information visualization techniques, which bundle the edges of a network diagram based on certain rules to increase the visibility of the network diagram and facilitate the analysis of key relationships among nodes. As one of evolutionary-based edge bundling, genetic algorithm-based edge bundling (called GABEB) is proposed which uses a genetic algorithm to optimize the placement of edges based on aesthetic criteria. However, it does not sufficiently consider the bundling between neighboring edges, and thus visual clutter issues still remain. Based on the above background, we propose an improved bundling method that considers the concatenating of control points at each edge using GABEB.

Keywords

Edge Bundling, Genetic Algorithm, Node-link Diagram, Information Visualization, Genetic Algorithm-based Edge Bundling (GABEB).

1. INTRODUCTION

Edge bundling is a method to reduce the visual clutter of a node-link diagram and facilitate intuitive understanding by adjusting the position of nodes and the arrangement of edges in a node-link diagram according to certain rules. Many studies have already proposed various edge-bundling methods, such as Force-directed Edge Bundling (FDEB) [Hol09] (Fig. 1), which is based on the dynamic rules and geometric rule-based methods such as Geometry-based Edge Bundling (GBEB) [Cui08].

On the other hand, Evolutionary-based edge bundling approaches like genetic algorithms (GA), which are evolutionary computations, have been implemented [Fer18][Mei22]. This is approached as an optimization problem to maximize the viewability defined by aesthetic rules, etc. These approaches are expected to provide visualization results that are not expected by humans. Among them, Genetic algorithm-based edge bundling (GABEB) is proposed [Sag20], which treats bundling as an optimization problem of a fitness function based on an evaluation value of aesthetic criteria [Sag16] and tries to optimize edge placement directly by moving control points. However, GABEB does not consider the bundling between edges located in the neighborhood, wherein the neighboring edges does not overlap exactly, while visual clutters remain.

For example, as shown in Fig. 2, there are two parallel edges of equal length and distance 10 apart. In GABEB, the edge bundling is expressed by moving these control points, but in this case, it is desirable that at least the second control point is completely attached to each other. In this case, it is desirable that at least the second control points are completely attached to each other like dashed circles. In this case, it is desirable for v_1 and v_2 to move (5.0,0), (-5.0, 0), but calculating these values is difficult in GA because of the random number factor involved, and errors will inevitably appear. Therefore, some kind of post-processing is necessary. In other words, if the control points are considered to almost overlap, it is necessary to add a process to overlap (merge) them.

In this study, we aim to improve the visibility problem of GABEB by adding a process considering the bundling of multiple edges located in the neighborhood. We focused on bundling edges by

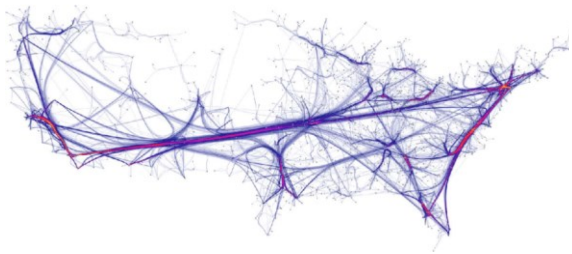

Figure 1. Edge Bundling Example [Hol09]

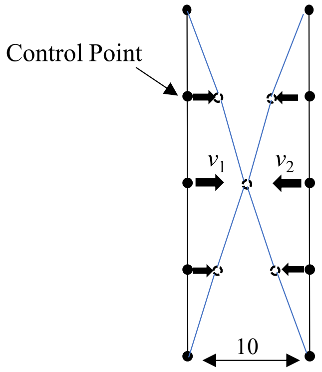

Figure 2. Problem description in GABEB

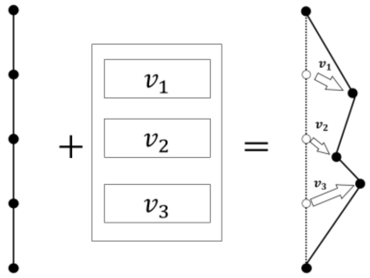

Figure 3. Genetic representation in GABEB

concatenating control points of neighboring edges and propose an improved bundling method that adds the process of concatenating control points of neighboring edges.

2. RELATED WORK

Edge Bundling and Evolutionary-based Edge Bundling

Edge bundling is a well researched research topic. Most works in this area define a model to express edge bundling with one of the best known methods being Holten's work where they proposed Hierarchical Edge Bundling for a graph based on a tree structure [Hol06].

Geometry-Based Edge Bundling (GBEB) proposed by Cui et al.[Cui08] realises edge bundling so as to bend edges based on meshes generated through a Delaunay triangulation, although this approach sometimes leads to some extreme bends. On the other hand, Holten et al. [Hol09] proposed FDEB which performs bundling based on Hooke's law. Also, Selassie et al. introduced Divided Edge Bundling by improving FDEB to apply to directed graph[Sel11], while Hurter et al. proposed Kernel Density Estimation Edge Bundling based on image-based visualisation [Hur12].

On the other hand, the approaches categorised into evolutionary-based edge bundling are proposed. Many graph layout algorithms using genetic algorithm [Bar00] [Bra96] [Elo96] [Net12] [Vra06] [Zha05] have been proposed since the last century. These methods aim to place nodes in a plausible way by optimizing some evaluation value, and have been proposed for not only directed graph but also undirected graphs, orthogonal graphs and so on. The evolutionary-based approaches are based on the idea of graph layout algorithms, which view edge bundling as an optimization problem and attempt to implement edge bundling by solving the optimization problem. Ferreira et al. [Fer18] proposed a bundling method by solving the edge combination optimization problem. The method is useful but FDEB is necessary to bundling in real. Saga et al. [Sag20] proposed a method by solving the placement problem of each control point on edges. Although this method produces a bundling result as a result of the evolutionary computation, it has the problem of leaving visual clutter if the optimization is not successful. This paper proposes a method to solve the problem.

3. Genetic algorithm-based Edge Bundling

Genetic algorithm

Genetic algorithms (GAs), which belong to the family of evolutionary algorithms, simulate Darwin's theory of evolution [Gol89]. GAs are employed to solve difficult, often NP-hard, optimization problems. The genetic representation and fitness function depend on the problem and domain to solve. After these are defined, a GA proceeds iteratively through stages of selection, crossover, and mutation to improve a population of individuals that expresses candidate solutions to the problem.

GABEB is one of the algorithms based on the GA which treated bundling as a placement optimization problem of edge control points based on aesthetic criteria.

Genetic Representation

The genetic representation of GABEB is based on control-based approaches. The approach employed in FDEB divides an edge uniformly by c control points. By moving these control points the edges can be controlled for edge bundling. In our algorithm, edges in the input graph are also divided based on c uniformly spaced points as shown in Fig. 3. Then, for each control point, GABEB stores a distance-limited displacement vector v (as (x, y) coordinates) (where the limited distance is called maximum movement distance). Thus, for n edges and using c control points per edge, we encode $2*n*c$ parameters.

Fitness Function

An appropriate fitness function is key to a successful GA. Here, there are also some general accepted aesthetic rules. The data-ink ratio[Tuf01] is one of the most widely used ones to evaluate visualization results quantitatively in all of visualization problems. It is based on the ink amount required for drawing a visualized figure. The path quality, proposed by Cui in GBEB, is also useful to evaluate the degree of zig-zag in edge bundling. Furthermore, Saga proposed three quantitative criteria to evaluate edge bundling which are formulated from the difference of edge length, area illustrated by edges (which is similar to data-ink ratio), and density of edges [Sag16].

GABEB adopts these three criteria together with the path quality by Cui, and uses the four criteria separately and perform multi-objective optimization.

3.3.1 *Mean Edge Length Difference*

Mean Edge Length Difference (MELD) is a criterion to express the difference from the original edges after edge bundling. A smaller change of edge lengths indicates superior edge bundling because of over-bundling, whereas a large change often leads to a loss of the meaning of the original network. MELD is calculated as

$$MELD = \frac{1}{n}\sum_{e\in E}|L'(e) - L(e)| \qquad (1)$$

where n is the number of edges, E is the edge set, and $L(e)$ and $L'(e)$ are the lengths of edge e before and after edge bundling, respectively. Employing this criterion, we can prevent edges from over-bending and over-bundling. MELD can be normalized to [0;1] by

$$MELD = \frac{1}{n}\sum_{e\in E}|1 - L'(e)/L(e)|$$

GABEB aims to minimize the MELD.

3.3.2 *Mean of Occupation Area*

Mean of Occupation Area (MOA) indicates the degree among the compressed areas before and after edge bundling. Based on the idea that better bundling can compress the area occupied by the edges, MOA is calculated as

$$MOA = \frac{1}{N}\left|\bigcup_{e\in E} O(e)\right| \qquad (2)$$

where N is the number of total areas, $O(e)$ is the set of areas occupied by edge e based on an occupation degree (we use 5% of unit area), and | | indicates the number of elements contained by a set. Minimizing the MOA is one of optimization goals of GABEB.

3.3.3 *Edge Density Distribution*

Edge Density Distribution (EDD) is rooted in the idea that a better edge bundling method can gather edges within a unit area and that the density per unit is high. EDD is calculated as

$$EDD = \frac{1}{|P|}\sum_{p\in P}(H(p) - H)^2 \qquad (4)$$

where P is a set of pixels, $H(p)$ is the number of edges pathing pixel p, and H is the average of $H(p)$. GABEB aims to minimize the EDD.

3.3.4 *Path Quality*

Path Quality (PQ) expresses the degree of zig-zag. The higher the PQ, the better the edge bundling. PQ is calculated by the summation of angle differences between neighbors as

$$PQ = \sum_{e\in E}(-\sum_{i=3}^{m}\gamma_i|\Delta_i|) \qquad (5)$$

with

$$\Delta_i = \begin{cases} A_i - A_{i-1} & \text{if } -\pi < |A_i - A_{i-1}| < \pi \\ |A_i - A_{i-1}| - 2\pi & \text{if } |A_i - A_{i-1}| > \pi \\ 2\pi + |A_i - A_{i-1}| & \text{if } |A_i - A_{i-1}| < -\pi \end{cases} \qquad (6)$$

and

$$\gamma_i = \begin{cases} 0 & \text{if } \operatorname{sign}(\Delta_i) = \operatorname{sign}(\Delta_{i-1}) \\ 1 & \text{if } \operatorname{sign}(\Delta_i) \neq \operatorname{sign}(\Delta_{i-1}) \end{cases} \qquad (7)$$

, where m is the number of segments divided by control points+1, and A_i is the angle between the original edge and the segment edge. GABEB tries to maximize PQ.

Genetic Operations

The main process of the proposed method follows NSGA-II [Deb et al., 2002] which is a method for multi-objective optimizations. Also, the genetic representation consists of real value for each gene, so the process uses BLX-α [Eshelman and Schaffer, 1993] for crossover. The overall of this process is as follows.

1. Initial population generation and evaluation

2. Selection, crossover by BLX-α and random mutation

3. Evaluation

4. Generation updating

5. Repeat 2. to 4. until the termination condition is satisfied.

Here, a generation is regarded as the process from step 2 to step 4. And BLX-α crossover operates to randomly generate a child from an extended area of the hyper-rectangle composed of the two parents, as shown in the following equation. From the parental genes p and q of dimension D, the child gene x is generated by the formula

$$x_i = r_i p_i + (1 - r_i)q_i \qquad (8)$$

where i is an index of dimension. Also, the termination condition is configured by the number of generations. Using this process, the vector of each control point in the gene is changed in order to ensure that the edges are well bundled. However, it is quite difficult for control points to overlap and bundle with each other, etc., since GABEB are dealing with real values of vectors. In particular, when the amount of movement of v is large, control points of adjacent edges rarely overlap.

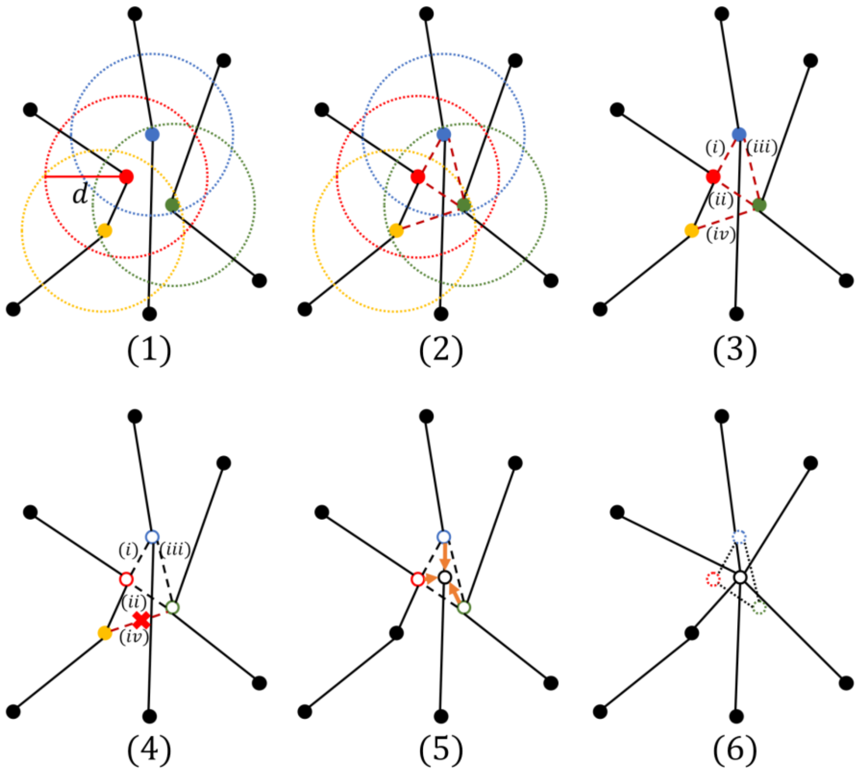

Figure 4. Concatenation Process

In order to improve the results of edge bundling, we add the steps related to concatenation process we propose before and after crossover and mutation steps. Hereafter, we describe the algorithm for concatenation and deconcatenation processes in detail.

4. GABEB WITH CONCATENATION PROCESS

In this paper, we propose a bundling method that considers concatenating of control points at neighboring edges to improve the visual clutter problem in GABEB. The overall of the improved process is as follows.

1. Initial population generation and evaluation
2. Deconcatenation of control points
3. Selection, crossover by BLX-α and random mutation
4. Concatenation of control points
5. Evaluation
6. Generation updating
7. Repeat 2. to 6. until the termination condition is satisfied.

Hereafter, we describe the algorithm for concatenation and deconcatenation processes in detail.

Control Point Concatenation and Deconcatenation Process

In this paper, we propose a bundling method that considers concatenating of control points at neighboring edges to improve the visual clutter problem in GABEB.

4.1.1 *Concatenation Process*

After the crossover and mutation process, the concatenating process of control points is performed. The control point merging process is performed as follows. An example figure of the concatenation process is shown in Fig. 4.

1. For all control points belonging to each edge, find the neighbouring control points where the distance is less than *d* (called maximum concatenating distance) and there is no control point belonging common edge in the combined set of control points (Fig. 4 (1), (2)).
2. Determine concatenating pairs in order of shorter distance between control points. If a control point included in a common edge is newly added to the set of control points that have already been joined by a control point pair that has already been decided to be concatenated, no concatenating is performed (Fig. 4 (3)). For example, when considering concatenation of the pair of control points shown in (*iv*), after the control point pairs (*i*), (*ii*), and (*iii*) have already been decided to be joined, the control point pair (iv) is judged that they are belonging to common edge due to the pair (*iii*), thus the control point pair (*iv*) is not joined.

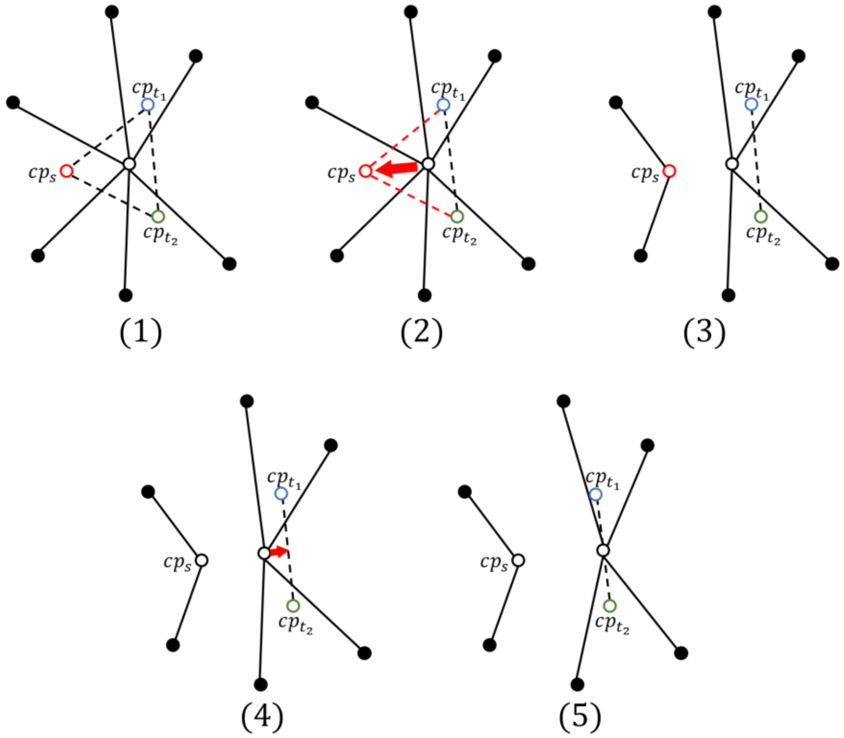

Figure 5. Deconcatenation Process

3. Stores the *pos* positions of each control point for which a join has been determined to be concatenating (Fig. 4 (4)).
4. Calculate the average position of each set of combined control points and assign them as the sharing position of the control points in the set shown as Fig. 4 (5), (6).

4.1.2 *Deconcatenation Process*

Before the crossover and mutation process, deconcatenation is performed when a bound control points within an individual becomes operation targets of crossover or mutation. The following procedure is used for deconcatenation (Fig. 5).

1. In the crossover and mutation process, check the presence of the control points that are bound to the target gene (Fig. 5 (1)).
2. If a bound control point cp_s is included in the control point set CP_S, remove the control point cp_s from CP_S and assign the position pos_s of the control point cp_s as the new position of the control point (Fig. 5 (2), (3)).
3. Perform the unbinding process for control points cp_t ($\forall cp_t \in CP_S$). In the case that cp_t is only bound to cp_s, assign the position post of the control point cp_t as the position of the control point. If there are other control points bound to cp_s, calculate the average position of CP_S without cp_s and assign it to CP_S (Fig. 5 (4), (5)).

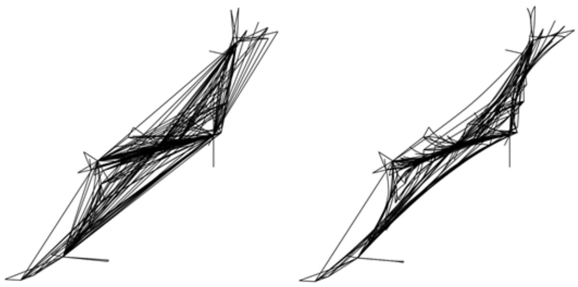

Figure 6. Original Japan Aerial Map(left) and FDEB result(right)

5. EXPERIMENTS

Goal, Dataset, Parameters and criteria

To check the effectiveness of the proposed method, we performed the experiments by applying proposed method to the node-link diagrams.

In this experiment, we used the node-link diagrams of the aerial map in Japan. The node-link diagram consists of 79 nodes (airports) and 233 edges (routes) in total. Bundled graph of the aerial map in Japan by FDEB is shown in Fig. 6 as example.

In the experiments, we check the four evaluation criteria used in fitness function, MELD, MOA, EDD, and PQ explained in Section 3.3. Also, we use Hypervolume which is widely used as an evaluation indicator for the non-dominated solutions in multi-objective optimization problems [Zit98][Li19]. The hypervolume is calculated from the area formed by the reference point and the solution set. And the larger this Hypervolume, the better the solution set is considered. We used reference point for Hypervolume to the worst value of each objective function.

Also, as parameters, the maximum movement distance and the maximum concatenating distance *d* are set to 10, 20, 30, 50. And the other parameters of the experiments are shown in Table 1.

Experiment Results

Bundled graph of the edge bundling results are shown in Fig. 6 and Fig. 7. Fig. 6 is the results of pareto solutions by GABEB and Fig. 7 is the results of proposed results. Also, bundled results with the change of the connection distance are shown in Fig. 8. We first compared the result figures of the bundling between Fig. 7 and Fig. 8. Because of the large d, the edges are basically hard to coalesce in GABEB. However, the results of the bundling in Fig. 8 are improved by the concatenation of the control points by the connection of the nearest neighbors. The comparison of the results of the bundling by the distance of the concatenation in Fig. 9 also shows that the more control points are aggregated as *d* increases. As a result, aggregation of the wide-area edges is well performed in bigger concatenating distances.

Next, we compared the evaluation value. The average evaluation value of the population is shown in Table 2, which shows proposed method archived better values in the two or three evaluation values by GABEB. Also, Hypervolume value of the non-dominated solutions in the whole population is shown in Table 3, and it indicates proposed method acquires more diverse solutions.

On the other hand, the computation time of proposed method shown in Table 5 is worse than GABEB. Moreover, the more longer movement distance increases computation time significantly compared to GABEB.

Initial Population Size	1000
Max Population Size	2000
Crossover Probability	0.9
Mutation Probability	0.05
α for BLX-α	0.5
Termination of generation	1000
MOA Unit Size	5
Control Point	3

Table 1. Parameters of Experiments

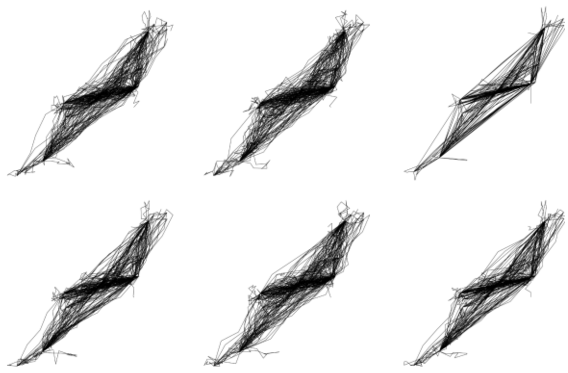

Figure 7. Examples of GABEB

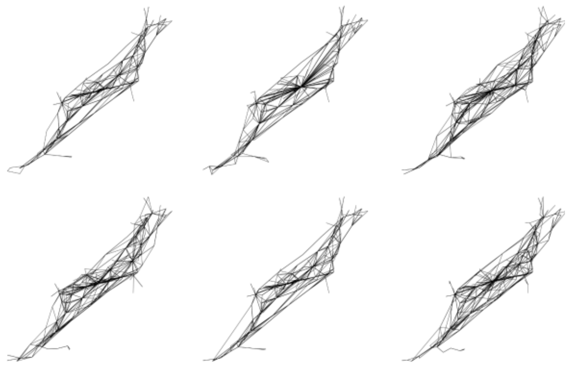

Figure 8. Sample Pareto Solutions of Proposed Method

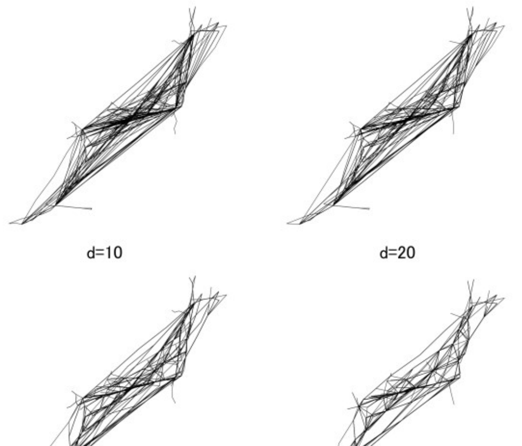

Figure 9. Example of Proposed Method Results with Different Concatenation Distance

In the proposed method, the calculation of the distance between the control points in the concatenating process requires large amount of computation time. Thus, the alternative method of the calculation of the

d	Method	MELD	MOA	EDD	PQ
10	GABEB	**0.9**	0.133	2.135	19.208
	Proposed	98.57	**0.023**	**0.248**	**23.626**
20	GABEB	**3.594**	0.134	2.251	33.555
	Proposed	29.296	**0.055**	**0.514**	**35.13**
30	GABEB	**7.816**	0.135	2.328	46.623
	Proposed	13.255	**0.104**	**0.827**	**68.274**
50	GABEB	**19.788**	**0.139**	2.463	67.727
	Proposed	28.596	0.141	**0.86**	**117.526**

Table 2. Evaluation of Edge Bundling Result (Average of values)

d	GABEB	Proposed
10	1679.129	**4780.816**
20	2626.932	**7534.682**
30	3523.665	**13293.775**
50	4755.462	**22836.456**

Table 3. Hypervolume value of Non-dominated Solutions

d	GABEB	Proposed
10	134.322	199.701
20	144.838	242.154
30	145.207	273.804
50	147.366	336.035

Table 4. Average computation time for an generation(sec)

distance of the control points such as approximation neighborhood search method needs to be considered.

6. CONCLUSION

GABEB is a method of bundling using a genetic algorithm as an optimization problem for edge placement based on aesthetic criteria, but GABEB does not sufficiently consider bundling process between neighboring edges, which causes the result of leaving visual clutter in the bundling results. We proposed an improved bundling method based on GABEB by considering the concatenation of control points of neighboring edges. By concatenating neighboring control points that satisfy certain conditions and proceeding with optimization with shared positions, which enabled to aggregating many control points and improving visual clutters.

In the experiment, proposed method performed bundling on Japan aerial map, and the results were compared with GABEB. Experiment results showed that the proposed method obtained a better evaluation values in some evaluation values and a more diverse solution set.

As future works, we believe it is necessary to solve the computational speed problem that makes application to large-scale node-link diagrams difficult, with faster concatenation processing. For this purpose, we plan to incorporate techniques such as Local Sensitive Hashing [Ind98] and SketchSort [Tab10] which are fast Nearest-Neighbor methods. Also, the results in this paper are shown using GA, but we would like to verify whether other optimization methods based on computational intelligence (such as meta heuristic algorithms like firefly algorithm [Yan08] can also be applied in Edge-Bundling, which aims for optimal placement of control points. Other possibilities include hardware acceleration (e.g., using GPGPU [Nak12]) rather than algorithms. Also, the proposed algorithm is implemented based on GABEB, which does not move nodes. Therefore, since graph drawing which is an algorithm to place the nodes properly needs to be considered separately, it is necessary to implement an algorithm that takes node placement into account as well.

7. ACKNOWLEDGMENTS

We appreciate KAKENHI 22K12116 that supports this research.

8. REFERENCES

[Bar00] Barreto, A. and Barbosa, H. Graph layout using a genetic algorithm. In Proc. of Sixth Brazilian Symposium on Neural Networks, pp. 179–184. 2000.

[Bra96] Branke, J., Bucher, F., and Schmeck, H.. Using Genetic Algorithms for Drawing Undirected Graphs. In The Third Nordic Workshop on Genetic Algorithms and their Applications, pp. 193–206, 1996

[Cui08] Cui, W., Zhou, H., Qu, H., Wong, P. C., and Li, X. Geometry-based edge clustering for graph visualization. In IEEE Transactions on Visualization and Computer Graphics, volume 14, pp. 1277–1284, 2008

[Deb02] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, Vol. 6, No. 2, pp. 182–197, 2002.

[Elo96] Eloranta, T., Eloranta, T., and M ̈akinen, E. Timga - a genetic algorithm for drawing undirected graphs. Technical report, Divulgaciones Matematicas, 1996

[Esh93] Eshelman, L. J. and Schaffer, J. D. Real-Coded Genetic Algorithms and Interval-Schemata, Foundations of Genetic Algorithms, Vol. 2, pp. 187–202, 1993.

[Fer18] Ferreira, J. d. M., do Nascimento, H. A., and Foulds, L. R. An evolutionary algorithm for an optimization model of edge bundling. Information (Switzerland), Vol. 9, No. 7, pp. 1–27, 2018.

[Gol89] Goldberg, D. E. Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Longman Publishing Co., Inc., 1989.

[Hol06] Holten, D. Hierarchical edge bundles: visualization of adjacency relations in hierarchical data. IEEE Transactions on Visualization and Computer Graphics, Vol.12, No. 5, pp. 741–748, 2006

[Hol09] Holten, D. and Van Wijk, J. J. Force-Directed edge bundling for graph visualization. Computer Graphics Forum, Vol. 28, No. 3, pp. 983–990, 2009

[Hur12] Hurter, C., Ersoy, O., Telea, A. Graph bundling by kernel density estimation. Computer Graphics Forum, Vol. 31, No. 3, pp. 865–874, 2012.

[Ind98] Indyk, P. and Motwani, R. Approximate nearest neighbors: Towards removing the curse of dimensionality. In Proc. of the Thirtieth Annual ACM Symposium on Theory of Computing, STOC ’98, pp. 604-613, 1998.

[Li19] Li, M. and Yao, X. Quality evaluation of solution sets in multi-objective optimisation: A survey. ACM Computing Surveys, Vol. 52, No. 2, 2019.

[Mei22] Meikari, J. and Saga, R. Evolutionary node layout and edge bundling. In Proc. of 2022 IEEE Congress on Evolutionary Computation (CEC), pp. 1-6, 2022.

[Nak12] Nakashima, T., Tanaka, K., Fujimoto, N. and Saga, R. GPGPU Implementation of fuzzy rule-based classifiers, Smart Innovation, Systems and Technologies Vol. 16, pp. 323-332, 2012

[Net12] Neta, B., Ara ́ujo, G., Guimar ̃aes, F., Mesquita, R., and Ekel, P. A fuzzy genetic algorithm for automatic orthogonal graph drawing. Applied Soft Computing, Vol. 12, pp. 1379–1389, 2012.

[Sag16] Saga, R. Quantitative Evaluation for Edge Bundling Based on Structural Aesthetics. EuroVis’16: In Proc. of the Eurographics /IEEE VGTC Conf. on Visualization, pp. 1–3, 2016.

[Sag20] Saga, R., Yoshikawa, T., Wakita, K., Sakamoto, K., Schaefer, G., and Nakashima, T. A genetic algorithm optimising control point placement for edge bundling. In VISIGRAPP 2020 – Proc. of the 15th International Joint Conf. on Computer Vision, Imaging and Computer Graphics Theory and Applications, Vol. 3, pp. 217–222, 2020

[Sel11] Selassie, D., Heller, B., Heer, J. Divided edge bundling for directional network data. IEEE Transaction Visualization & Computer Graphics, Vol. 17, No. 12, pp. 2354–2363, 2011.

[Tab10] Tabei, Y., Uno, T., Sugiyama, M. and Tsuda, K. Single versus multiple sorting in all pairs similarity search. In Proc. of ACML2010, pp. 145–160, 2010.

[Tuf01] Tufte, E. The Visual Display of Quantitative Information, Graphics Press USA, 2001.

[Vra06] Vrajitoru, D. and El-Gamil, B. R. Genetic algorithms for graph layouts with geometric constraints. In Proc. of International Conference on Climate Informatics, 2006.

[Yan08] Yang, X.-S. Nature-Inspired Metaheuristic Algorithms, Luniver Press, 2008.

[Zha05] Zhang, Q.-G., Liu, H.-Y., Zhang, W., and Guo, Y.-J. Drawing undirected graphs with genetic algorithms. In International Conference on Natural Computation, pp. 28–36. Springer.

[Zit98] Zitzler, E. and Thiele, L. Multiobjective optimization using evolutionary algorithms - A comparative case study. In Lecture Notes in Computer Science, Vol. 1498, pp. 292–301, 1998.