Hyperparameter Optimization for Graph Drawing

Fuga Takata [*] Nihon University

Yosuke Onoue [†] Nihon University

ABSTRACT

Hyperparameter tuning for graph drawing algorithms yields enhanced drawing results; however, it is a difficult and time-consuming task. Therefore, we propose an approach to apply a hyperparameter optimization technique to graph drawing algorithms. In this paper, we report on the results of three computational experiments to show the effectiveness of our approach using a graph drawing algorithm, Sparse SGD, as a trial of concept.

Index Terms: Human-centered computing—Visualization—Visualization techniques—Graph drawings; Applied computing—Operations research—Decision analysis—Multi-criterion optimization and decision-making;

1 INTRODUCTION

Graph drawing algorithms are widely used to visualize network data [4]. Various graph drawing algorithms have been proposed, and most of them require several parameters (e.g., the number of iterations and cooling schedule for simulated annealing) to adjust the drawing results. These parameters are called hyperparameters and have a significant effect on the drawing quality. Graph drawing users tune hyperparameters to obtain better drawings; however, this is a time-consuming and difficult task.

In machine learning (ML), hyperparameter optimization (HPO) techniques that automatically adjust the hyperparameters of an ML model are often used to improve the prediction accuracy of the model. Inspired by HPO in the ML field, we introduce HPO in graph drawing. To optimize the quality of drawing results, it is necessary to represent the quality of the drawing results by quantitative metrics. We employ several quantitative quality metrics used in the quantitative evaluation of drawing results (e.g., stress and the number of edge crossings) and formalize a multiobjective HPO problem for graph drawing.

We performed computational experiments of HPO on benchmark network data for one graph drawing algorithm, Sparse SGD [5], as a trial of concept. In this paper, we present the results, which indicate the effectiveness of our proposed approach of applying an HPO technique to graph drawing algorithms.

2 HPO FOR GRAPH DRAWING

First, we introduce some notations for formalizing HPO problems for graph drawing. Let $G=(V,E)$ be a graph (V,E are set of vertices and edges, respectively) and D_G be a drawing of G. A layout algorithm f takes a set of hyperparameters $P=(P_1,P_2,\cdots,P_{n_f})$ where n_f is the number of hyperparameters for the algorithm f. Let $S=S_1\times S_2\times\cdots\times S_{n_f}$ be a domain of hyperparameters. A drawing D_G is determined as $D_G=f(G,P)$ where $P\in S$. The quality of D_G is described as $Q(D_G)=(Q_1(D_G),Q_2(D_G),\cdots,Q_m(D_G))$ where m is the number of quality metrics. Notably, some quality metrics $Q'_i(D_G)$ are desirable to be low; however, by setting $Q_i(D_G)=-Q'_i(D_G)$ we can assume that all quality metrics are desirable to be high. In reality, many graph drawing algorithms have randomness, so the same graph and hyperparameters do not always produce the same drawing. Therefore, the statistical value of the results of a sufficient number of trials is adopted as the value of each quality metric.

An HPO problem is formalized as follows:

$$\underset{P\in S}{\arg\max}.Q(f(G,P)) \tag{1}$$

Generally, there are tradeoffs among quality metrics, and not all metrics can be maximized simultaneously. Therefore, our goal is to find a Pareto frontier F of hyperparameters. The Pareto optimal solutions $P^s,P^t\in F(P^s\neq P^t)$ satisfies $\exists i\in\{1,2,\cdots,m\}:Q_i(f(G,P^s))>Q_i(f(G,P^t))$. In this study, we used Optuna [2], an HPO framework, to find the Pareto frontier.

3 COMPUTATIONAL EXPERIMENTS

We performed computational experiments to confirm that the proposed approach generates parameters that obtain good values of the quality metrics.

3.1 Hypotheses

We describe three hypotheses about the proposed approach.

- Hypothesis 1: Quality metrics can be optimized using the proposed approach.
- Hypothesis 2: Optimized parameters using the proposed approach can obtain good drawing quality.
- Hypothesis 3: Optimized parameters using the proposed approach can be applied to any graph to obtain a good drawing quality.

Graph drawing algorithms attempt to generate good drawings according to their criteria. This does not necessarily optimize external metrics, such as quality metrics. Therefore, it is not obvious whether tuning hyperparameters improves quality metrics. Hypothesis 1 was set up to clarify this. As aforementioned, quality metrics have tradeoffs in various manners. Considering multiple quality metrics while tuning is important; therefore, we formulated Hypothesis 2. In the proposed approach, a particular graph is drawn many times and the qualities are measured. Consequently, it takes a long time to generate parameters for large-scale graphs. Therefore, applying the parameters generated using a relatively small graph to a relatively large graph reduces the time taken for optimization. In addition, parameters that show good quality across different graphs are considered good parameters for that layout algorithm. Based on this, we set up Hypothesis 3.

3.2 Experimental Design

In the experiments, we use the following quality metrics described in Ahmed et al. [1]: angular resolution (ANR), aspect ratio (AR), crossing angle (CA), crossing number (CN), Gabriel graph property (GB), ideal edge length (IE), node resolution (NR), shape-based metrics (SB), and stress (ST). For the layout algorithm, we use the Sparse SGD, which minimizes the stress function based on the stochastic gradient descent approach. To confirm the effectiveness of the HPO, we compare the distributions of quality metrics using

[*]e-mail: chfu22019@g.nihon-u.ac.jp

[†]e-mail: onoue.yousuke@nihon-u.ac.jp

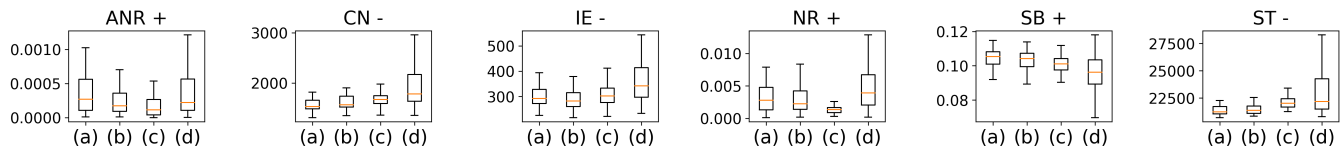

Figure 1: Distributions of quality metrics on 1138-bus graph with (a) Pareto frontier optimized with 1138-bus graph, (b) Pareto frontier optimized with lesmis graph, (c) empirical parameters, and (d) randomly generated parameters.

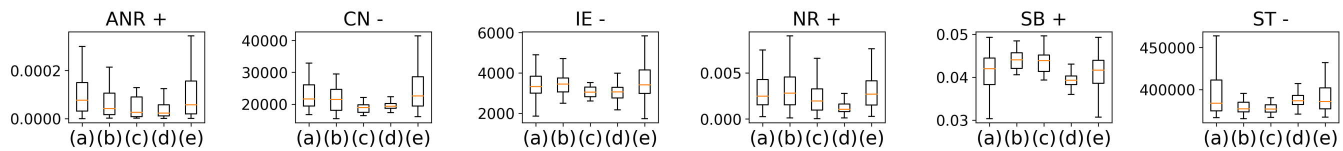

Figure 2: Distributions of quality metrics on US-Grid graph with (a) Pareto frontier optimized with US-Grid graph, (b) Pareto frontier optimized with 1138-bus graph, (c) Pareto frontier optimized with lesmis graph, (d) empirical parameters, and (e) randomly generated parameters.

empirically used parameters and randomly generated parameters. The domains of the hyperparameters were set to $[1,100]$ for the number of pivots, $[1,200]$ for the number of iterations, and $[0.01,1]$ for the ε used to control the step size for each iteration. We used 50 for the number of pivots, 100 for the number of iterations, and 0.1 for the ε as a set of empirical parameters.

In Experiment 1, we assessed whether each quality metric could be optimized using the proposed approach. Specifically, we optimized hyperparameters using a single quality metric as an objective function and compared the optimized parameters with 20 randomly generated parameters. A quality metric is considered to be optimized if the quality metric using the optimized parameter is in the 75th percentile of the quality metric using the randomly generated parameters. In Experiment 2, we compare the distributions of quality metrics with parameters contained in the optimized Pareto frontier, empirical parameters, and randomly generated parameters. In Experiment 3, we measure the quality metrics by applying parameters optimized on a smaller graph to a bigger graph. This would suggest the generality of the effect of hyperparameters on quality metrics. For Experiments 2 and 3, 100 empirical parameters and 100 randomly generated parameters were used.

We chose three graphs, lesmis($|V|=77, |E|=254$), 1138-bus ($|V|=1138, |E|=1458$), and US-Grid ($|V|=4941, |E|=6594$) from well-known benchmark datasets [3] to be drawn. In addition, we empirically set the number of trials for exploring optimal hyperparameters in Optuna to 100 for Experiment 1 and 200 for Experiments 2 and 3.

3.3 Results

From Experiment 1, we confirmed that optimized parameters outperformed randomly generated parameters on all three graphs for quality metrics ANR, CN, IE, NR, SB, and ST. There were no significant differences from the randomly generated parameters in other quality metrics (AR, CA, and GB). Therefore, we used only six quality metrics ANR, CN, IE, NR, SB, and ST for the subsequent experiments.

Figures 1 and 2 show the distributions of quality metrics for each condition on 1138-bus and US-Grid data, respectively. Outliers have been omitted from the boxplots to avoid irrelevant results. Figures 1 (a), (c), and (d) and Figures 2 (a), (d), and (e) depict the results of Experiment 2. We confirmed that the parameters contained in the optimized Pareto frontier outperformed the empirical and randomly generated parameters in most quality metrics. Figures 1 (a) and (b) and Figures 2 (a), (b), and (c) depict the results of Experiment 3. The results suggest no significant difference when the parameters optimized in one graph are applied to other graphs in most cases. Although omitted due to space limitations, similar results were confirmed for the lesmis graph.

3.4 Discussion

Through experiments, optimizing the hyperparameters of Sparse SGD can improve the values of several quality metrics compared with empirical parameters and randomly generated parameters. From the results of Experiment 1, Hypothesis 1 was partially supported, and some quality metrics were confirmed to be optimizable. From the results of Experiment 2, the proposed approach can tune multiple quality metrics simultaneously, which is difficult even for experts. In addition, from the results of Experiment 3, Hypotheses 2 and 3 were suggested to be supported.

4 Conclusion and Future Work

In this study, we proposed an approach for optimizing the hyperparameters of graph drawing and performed computational experiments to show the effectiveness of the proposed approach. In future work, we will optimize the parameters of various graph drawing algorithms and conduct a user study to analyze user preferences.

References

[1] R. Ahmed, F. De Luca, S. Devkota, S. Kobourov, and M. Li. Graph drawing via gradient descent,(gd)ˆ 2 (gd) 2. In *Graph Drawing and Network Visualization: 28th International Symposium, GD 2020, Vancouver, BC, Canada, September 16–18, 2020, Revised Selected Papers 28*, pp. 3–17. Springer, 2020.

[2] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2623–2631, 2019.

[3] R. Rossi and N. Ahmed. The network data repository with interactive graph analytics and visualization. In *Proceedings of the AAAI conference on artificial intelligence*, vol. 29, 2015.

[4] R. Tamassia. *Handbook of graph drawing and visualization*. CRC press, 2013.

[5] J. X. Zheng, S. Pawar, and D. F. Goodman. Graph drawing by stochastic gradient descent. *IEEE transactions on visualization and computer graphics*, 25(9):2738–2748, 2018.