# Anonymizing Global Edge Weighted Social Network Graphs

Jiaru Wang[1], Ziyi Wan[1,3], Jiankang Song[1,3], Yanze Huang[1,2], Yuhang Lin[1,3], and Limei Lin[1,3](✉)

[1] College of Computer and Cyber Security, Fujian Normal University,
Fuzhou 350117, Fujian, People's Republic of China
linlimei@fjnu.edu.cn

[2] School of Computer Science and Mathematics, Fujian University of Technology,
Fuzhou 350118, Fujian, People's Republic of China

[3] Key Laboratory of Network Security and Cryptology, Fujian Normal University,
Fuzhou 350117, Fujian, People's Republic of China

**Abstract.** Privacy protection of individual users in social networks is becoming more and more important, thus it requires effective anonymization techniques. In this paper, we use Kruskal and Prim algorithms to model the linear programming of the minimum spanning tree. Finally, we execute the experiments on the number of anonymity solutions and time with different edge weights to analyze the Kruskal algorithm and Prim algorithm to verify their anonymity feasibility.

**Keywords:** Privacy protection · Edge weight anonymization · Social network

## 1 Introduction

Social networks are growing very fast in human society nowadays, and they are attracting the interest of researchers from many disciplines. In social networks, the data might contain sensitive information of individuals, which should not be released. However, disclosing information is a voluntary act for individuals, while they do not know who can access their data and how their data will be used. Thus the data needs to be anonymized before publication to protect the privacy of individuals. There has been increasing concerns on the privacy of individuals in social networks. Siddula et al. [1] proposed a scheme for k-anonymity by using an enhanced equi-cardinal clustering method. In 2020, Huang et al. [2] gave a differential privacy protection scheme based on clustering and noise in social networks. In 2021, Safi et al. [3] provided a framework for privacy protection by public broadcast and attribute-baed encryptions in mobile social networks.

In social networks, individuals can link to or make friends with each other. Moreover, there are rich interactions in social networks, such as joining communities or groups of common interest. Thus we can view a social network as a graph, where individuals are treated as nodes, and the links represent the social ties between individuals. For edge weighted social networks, the weights of edges represent the closeness of social ties

The original version of this chapter was revised: The sections Introduction, Preliminary and References have been revised and updated. The correction to this chapter is available at https://doi.org/10.1007/978-981-16-7913-1_14

L. Lin et al. (Eds.): SocialSec 2021, CCIS 1495, pp. 119–130, 2021.
https://doi.org/10.1007/978-981-16-7913-1_9

between individuals, which should be protected by methods such as anonymization. Das et al. [4, 5] proposed a scheme for protecting the weighted social networks based on linear programming. Also, Zhang and Zhu [6] explored the centrality and cumulative degree distribution of skeletons for weighted social network. The privacy in k-anonymity of shortest paths by modifying edge weights are studied in [7, 8], respectively. Liu et al. [9] proposed an anonymization scheme based on probabilistic indistinguishability, while Dou et al. [10] used weighted noise injection to preserve privacy in multimedia recommendation for social networks. In order to boost the accuracy of differentially private, Wang and Long [11] designed an algorithm for each sub-network of original weighted social network. Moreover, Walia et al. [12] used weighted graph to propose a secure multimodal cancelable biometric system. Furthermore, Zhao et al. [13] gave secure outsourcing schemes based on untrusted cloud server for the min-cut of undirected edge-weighted graphs. Besides, Yin et al. [14] designed local Bayesian differential privacy, and proposed a hybrid method for federal learning.

In this paper, we focus on the anonymization of edge weights. We model the edge weighted graph by preserving the required key property, and replace the original edge weights by other edge weights satisfying the model. In [4, 5], the authors did not give the experimental steps for specific algorithms. On this basis, we first give specific experimental steps, and then further propose anonymous schemes under different algorithms. The contributions of this paper are listed as follows.

- We use the Kruskal and Prim algorithms to model the linear programming of the minimum spanning tree.
- We execute the experiments on the number of anonymity solutions and time with different edge weights to analyze the Kruskal algorithm and Prim algorithm to verify their anonymity feasibility.

## 2 Preliminary

Das et al. [4, 5] proposed an abstract modeling technique based on linear programming problems as follows. For an algorithm whose key attributes are expressed as linear combinations of edge weights, it makes decisions based on the actual values $w_i$'s of edge weights and uses variables $x_i$ to model the decisions. In its execution, each step is be expressed by inequalities involving edge weights, which lead to a system of linear inequalities:

$$\begin{bmatrix} p_{11} & p_{12} & \ldots & p_{1m} \\ p_{21} & p_{22} & & p_{2m} \\ \vdots & & \ddots & \vdots \\ p_{nm} & p_{nm} & \cdots & p_{nm} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \leq \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} \tag{1}$$

$$P = \begin{bmatrix} p_{11} & p_{12} & \ldots & p_{1m} \\ p_{21} & p_{22} & & p_{2m} \\ \vdots & & \ddots & \vdots \\ p_{nm} & p_{nm} & \cdots & p_{nm} \end{bmatrix} \tag{2}$$

$$\begin{aligned} \boldsymbol{X^T} &= [x_1, x_2, \ldots, x_n], \\ \boldsymbol{Q^T} &= \left[q_1, q_2, \ldots, q_n\right] \end{aligned} \tag{3}$$

When the edge weights are replaced by any solution of (1), the properties of the graph will be preserved under the modeled algorithm. In general, the model is formulated as a linear programming problem by the above expressions (1), (2) and (3) as follows.

$$\begin{aligned} &\textit{Minimize (or Maximize)} \quad \boldsymbol{F}(x_1, x_2, \ldots, x_m) \\ &\textit{subject to } \boldsymbol{PX} \leq \boldsymbol{Q} \end{aligned} \tag{4}$$

where $\boldsymbol{F}$ is a linear objective function. The model can be obtained by changing the set of inequalities (4) for different algorithms that correspond to different series of inequalities and objective functions. In the following, we use the Kruskal algorithm and Prim algorithm models based on the minimum spanning tree.

## 3 Anonymizing Scheme

Given the original weighted graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{W})$, we use $w_i$ to represent the original weight of the positive edge, where each $w_i$ corresponds to the edge $i = (u, v) \in \boldsymbol{E}$. We use $x_i$ to represent the anonymization edge weight in anonymity edge weighted graph $\boldsymbol{M} = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{X})$, and the component in the matrix $\boldsymbol{X}$, where $\boldsymbol{V}$ and $\boldsymbol{E}$ do not change, and only the edge weight is anonymized.

In order to implement anonymizing models of different algorithms, a concrete original edge weighted graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{W})$ is described as follows (Fig. 1).

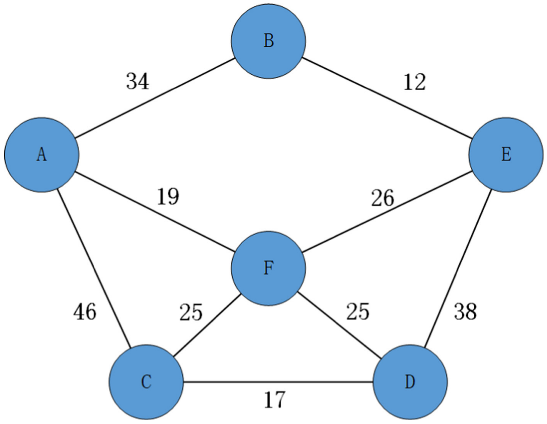

**Fig. 1.** A concrete original edge weighted graph $\boldsymbol{G}$.

$$\boldsymbol{V} = \{A, B, C, D, E, F\},$$

$$\boldsymbol{E} = \{(A, B), (B, E), (A, F), (F, E), (A, C), (F, C), (F, D), (E, D), (C, D)\},$$

$$\boldsymbol{W} = \{w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, w_9\} = \{34, 12, 19, 26, 46, 25, 25, 38, 17\}$$

For subsequent verification of the anonymizing model, the minimum spanning trees of the original edge weighted graph $\boldsymbol{G}$ based on the Kruskal algorithm and Prim algorithm are respectively listed as follows (Fig. 2):

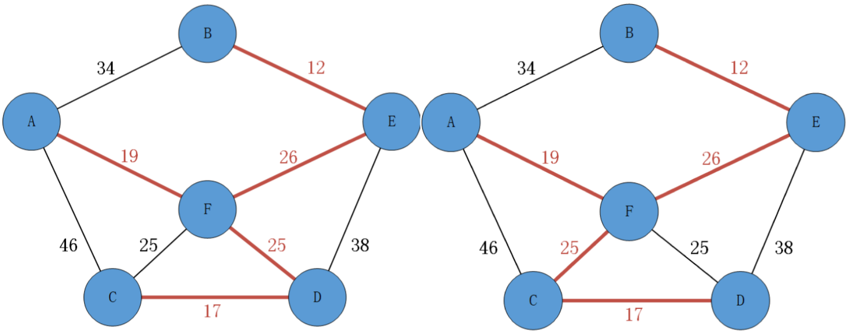

**Fig. 2.** Two minimum spanning tree based on the original edge weighted graph.

### 3.1 An Anonymity Model of Edge Weight Based on Kruskal Algorithm

#### A. Abstract model based on Kruskal algorithm

***Constraint*** **I**: At every step of Kruskal's algorithm for the MST, the edge with the minimum weight amongst the set of remaining edges is selected, and if this edge does not result in a cycle, it is added to the MST. Let $(u, v)$ be the edge selected in the $i^{th}$ iteration, and $(u', v')$ be the edge selected in the $(i + 1)^{th}$ iteration, then this implies that $w[u, v] \leq w[u', v']$. If $x_{(u,v)}$ and $x_{(u',v')}$ are the variables representing these edges in the model, then this outcome is modeled by the inequality $x_{(u,v)} \leq x_{(u',v')}$. Therefore, for every pair of edges $(u, v)$ and $(u', v')$ selected in consecutive iterations, the inequality $x_{(u,v)} \leq x_{(u',v')}$ can be added to the model whenever the given weights satisfy $w[u, v] \leq w[u', v']$.

***Constraint*** **II:** All the edges $(u, v)$ that are added to the minimum spanning tree, and the minor edges $\left(u', v'\right)$ that can be added but do not form a ring when the last edge is not added to the MST. There exists $w[u, v] \leq w\left[u', v'\right]$, correspondingly there is a constraint $x_{(u,v)} \leq x_{(u',v')}$.

***Objective function***:

$$\begin{aligned} &Minimize\ \boldsymbol{F}(x_1, x_2, \ldots, x_m) = c_1x_1 + c_2x_2 + \ldots + c_mx_m, \\ &Num_{c_m=1} = n - 1, Num_{c_m=0} = m - n + 1 \end{aligned} \tag{5}$$

where $Num_{c_m=1}$ represents the number of $C_m = 1$ and $Num_{c_m=0}$ represents the number of $C_m = 0$.

Based on the Kruskal algorithm, we establish a concrete anonymity model as follows:

First, add the edge $(B, E)$ with the smallest edge weight, whose edge weight is $x_2$. Second, add edge $(C, D)$ with edge weight is $x_9$, then $x_2 \leq x_9$. If edge $(A, F)$ is added, whose edge weight is $x_3$, then $x_9 \leq x_3$. Fourth, add edge $(C, F)$ with edge weight is $x_6$, then $x_3 \leq x_6$. Finally, add edge $(E, F)$ with edge weight is $x_4$, then $x_6 \leq x_4$.

Therefore:

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{6}$$

$$\begin{aligned} \boldsymbol{X^T} &= [x_1, x_2, \ldots, x_9], \\ \boldsymbol{Q^T} &= \left[q_1, q_2, \ldots, q_n\right] = [0, 0, \ \ldots, 0] \end{aligned} \tag{7}$$

Based on constraint II, for any $x_i$, $1 \leq x_9 \leq 38$.

Based on the above constraints, the linear model is solved. Assume that the publisher further restricts the edge weight to $10 \leq x_9 \leq 38$. By the exhaustive method, it is found that there are 53,130 anonymity solutions for the publisher to choose. The time taken to solve the linear model is 62.003 s.

### B. Validation based on Kruskal specific model

In this subsection, we choose one solution for verifying the anonymity as follows (Fig. 3).

**Scheme I:** $x_2 = 16, x_3 = 21, x_4 = 27, x_6 = 26, x_9 = 19$.

The anonymity graph is listed as follows:

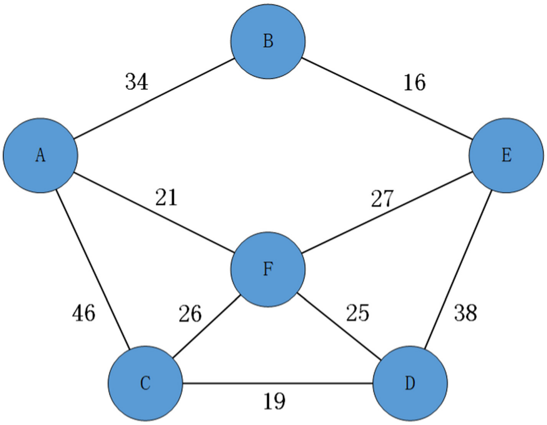

**Fig. 3.** Anonymity graph based on Scheme I.

Through the Kruskal algorithm, a minimum spanning tree of the anonymity graph of Scheme I is obtained as follows, where the red edges represent the edges of the anonymity graph spanning tree (Fig. 4).

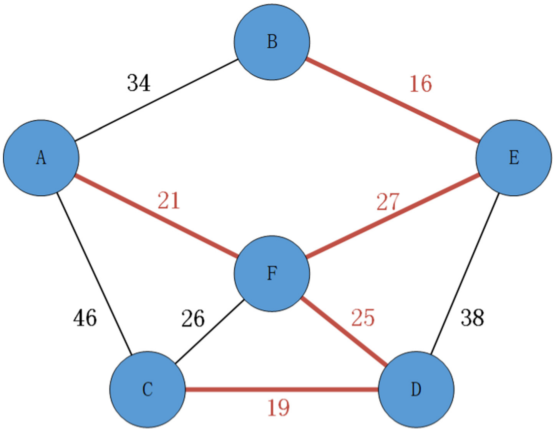

**Fig. 4.** The minimum spanning tree of the anonymity graph based on Scheme I.

It can be found that the minimum spanning tree of the anonymity graph is the same as the minimum spanning tree of the original weighted graph. This shows that the linear model based on Kruskal algorithm is feasible.

## 3.2 An Anonymity Model of Edge Weight Based on Prim Algorithm

### A. Abstract model based on Prim algorithm

Assume that $\boldsymbol{TE}$ is the set of edges in the minimum spanning tree on $\boldsymbol{G}$. The algorithm starts from $\boldsymbol{U} = \{u_0\}(u_0 \in \boldsymbol{V}), \boldsymbol{TE} = \{\}$ and repeats the following operations: Find an edge with the least cost among all the edges $(u, v) \in \boldsymbol{E}$ with $u \in \boldsymbol{U}, v \in \boldsymbol{V} - \boldsymbol{U}$. $(u_0, v_0)$ is merged into the set $\boldsymbol{TE}$, and $v_0$ is merged into the set $\boldsymbol{U}$ at the same time, until $\boldsymbol{U} = \boldsymbol{V}$. At this time, there must be $n - 1$ edges in $\boldsymbol{TE}$, then $\boldsymbol{T} = (\boldsymbol{V}, \boldsymbol{TE})$ is the minimum spanning tree of $\boldsymbol{G}$.

***Constraint*** **I**: In the Prim algorithm, $(u, v)$ becomes the edge selected for the $i^{th}$ generation, $(u', v')$ is the edge that can be selected but not selected for the $i^{th}$ generation, and the inequality is in the selection $x_{(u,v)} \leq x_{(u',v')}$ will be added to this model when the given weight satisfies $w[u, v] \leq w[u', v']$.

***Constraint*** **II**: The difference between Kruskal algorithm and Prim algorithm is that Prim algorithm is based on all edge weights. The publisher can model based on constraint I and set the anonymity range of the side rights by himself.

***Objective function***: Prim algorithm and Kruskal algorithm are both MST algorithms, so the objective function is the same.

Based on the Prim algorithm, we establish a concrete anonymity model as follows:

First, select $(A, F)$ and add vertex $F$, then

$$x_3 \leq x_1, x_3 \leq x_5,$$

$$U = \{A, F\},\ \boldsymbol{TE} = \{(A, F)\}.$$

Second, select $(C, F)$ and add vertex $C$, then

$$x_6 \le x_1, x_6 \le x_4, x_6 \le x_5, x_6 \le x_7,$$

$$\boldsymbol{U} = \{A, F, C\},\ \boldsymbol{TE} = \{(A, F), (C, F)\}$$

Third, select $(C, D)$ and add vertex $D$, then

$$x_9 \le x_1, x_9 \le x_4, x_9 \le x_7,$$

$$\boldsymbol{U} = \{A, F, C, D\},\ \boldsymbol{TE} = \{(A, F), (C, F), (C, D)\}$$

Fourth, select $(F, E)$ and add vertex $E$, then

$$x_4 \le x_1, x_4 \le x_8,$$

$$\boldsymbol{U} = \{A, F, C, D, E\},\ \boldsymbol{TE} = \{(A, F), (C, F), (C, D), (F, E)\}$$

Fifth, select $(B, E)$ and add vertex $B$, then

$$x_2 \le x_1,$$

$$\boldsymbol{U} = \{A, F, C, D, E, B\},\ \boldsymbol{TE} = \{(A, F), (C, F), (C, D), (F, E), (B, E)\}$$

Based on constraint II, assume that the publisher further restricts the edge weight to $1 \le x_i \le 10$. The linear model is solved by the exhaustive method, and there are 12,855,337 anonymity solutions for the publisher to choose. The time required to solve the linear model is 16594.8 s.

**B. Validation based on Prim specific model.**
In this subsection, we choose one solution for verifying the anonymity as follows.

**Scheme II:** $x_1 = 7, x_2 = 1, x_3 = 7, x_4 = 5, x_5 = 10, x_6 = 1, x_7 = 7, x_8 = 10, x_9 = 2.$

The anonymity graph is listed as follows (Fig. 5):

Through the Prim algorithm, a minimum spanning tree of the anonymity graph of Scheme II is obtained as follows, where the red edges represent the edges of the anonymity graph spanning tree (Fig. 6).

It can be found that the minimum spanning tree of the anonymity graph is the same as the minimum spanning tree of the original weighted graph. This shows that the linear model based on Prim algorithm is feasible.

## 4 Experiments

Both Kruskal algorithm and Prim algorithm are used for finding the minimum spanning tree. Based on the linear anonymity of MST, the range of anonymous edge weights changes with respect to the time (in seconds), number of anonymity solutions (No. of solutions), and degree of anonymity $\boldsymbol{\alpha}$ (the quotient of number of anonymous edges and number of edges of the original edge weight graph). The experimental results are shown in Tables 1, 2, and Figs. 7, 8, 9, 10.

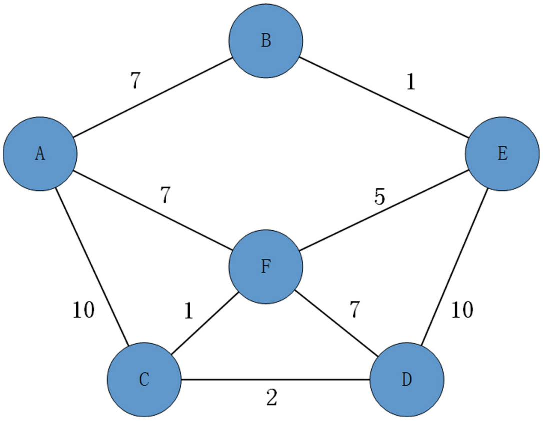

**Fig. 5.** Anonymity graph based on Scheme II.

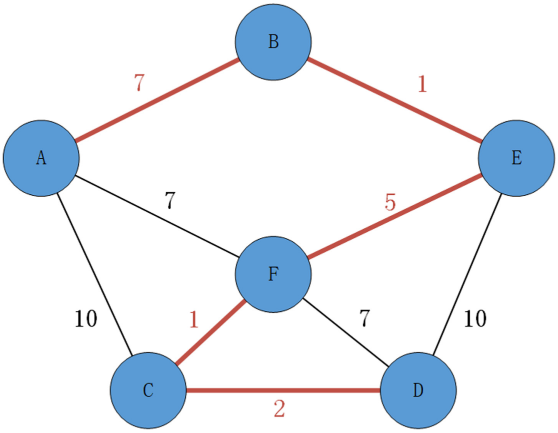

**Fig. 6.** The minimum spanning tree of the anonymity graph based on scheme II.

**Table 1.** Anonymity results based on Kruskal algorithm.

| Edge weight | No. of solutions | Time | $\alpha$ |
|---|---|---|---|
| 1–5 | 126 | 0.409 | 0.56 |
| 2–6 | 126 | 0.417 | 0.56 |
| 3–7 | 126 | 0.579 | 0.56 |
| 4–8 | 126 | 0.6 | 0.56 |
| 5–9 | 126 | 0.501 | 0.56 |
| 6–10 | 126 | 0.707 | 0.56 |

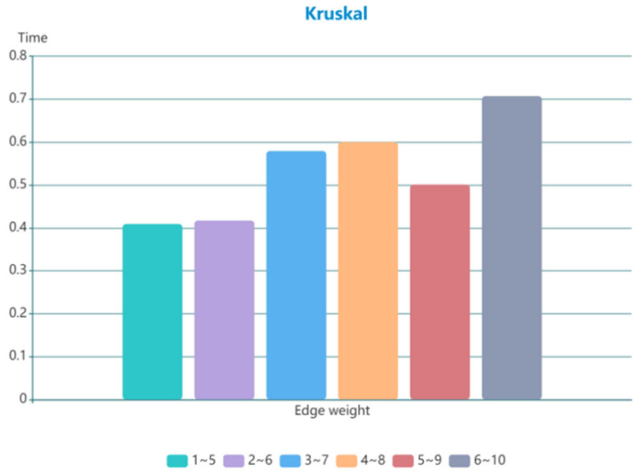

**Fig. 7.** Anonymity time with different edge weights based on Kruskal algorithm.

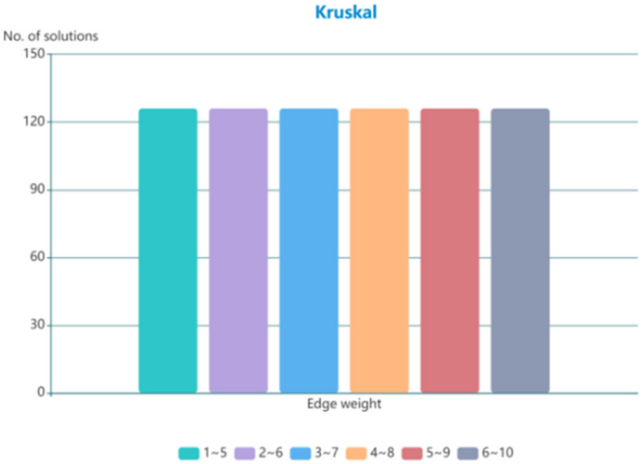

**Fig. 8.** Number of anonymity solutions based on Kruskal algorithm with different edge weights.

In Table 1, Fig. 7 and Fig. 8, we can get that the number of anonymity solutions based on Kruskal algorithm is 126 for all choices of the edge weights. However, the anonymity time with different edge weights are different. When the edge weights are in 1–5, the anonymity time is 0.409 s. When the edge weights are in 2–6, the anonymity time is 0.417 s. When the edge weights are in 3–7, the anonymity time is 0.579 s. When the edge weights are in 4–8, the anonymity time is 0.6 s. When the edge weights are in 5–9, the anonymity time is 0.501 s. When the edge weights are in 6–10, the anonymity time is 0.707 s.

**Table 2.** Anonymity results based on Prim algorithm.

| Edge weight | No. of solutions | Time | $\alpha$ |
|---|---|---|---|
| 1–5 | 45261 | 195.288 | 1 |
| 2–6 | 45261 | 137.863 | 1 |
| 3–7 | 45261 | 183.463 | 1 |
| 4–8 | 45261 | 181.371 | 1 |
| 5–9 | 45261 | 172.721 | 1 |
| 6–10 | 45261 | 154.81 | 1 |

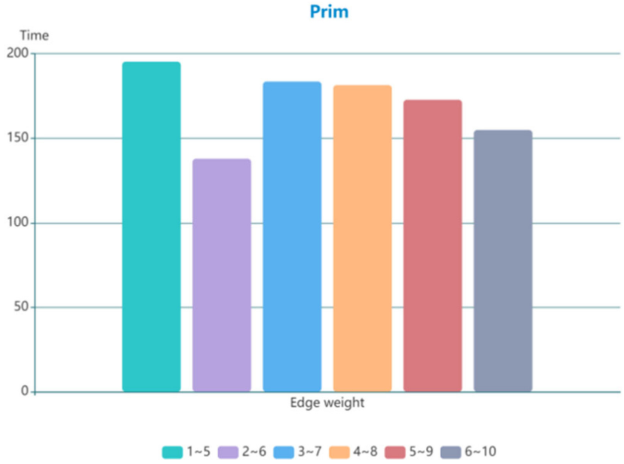

**Fig. 9.** Anonymity time with different edge weights based on Prim algorithm.

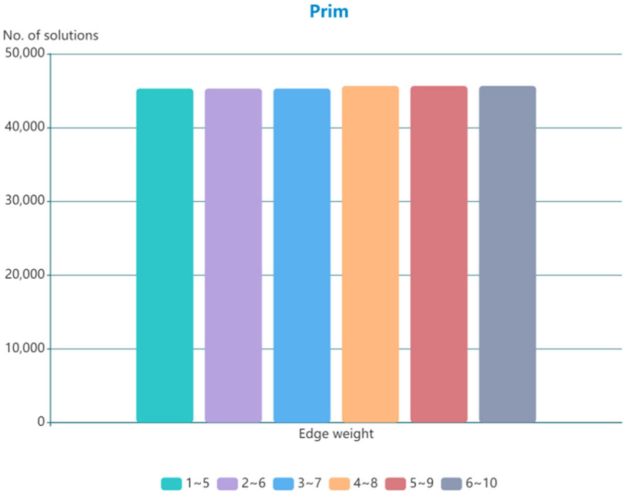

**Fig. 10.** Number of anonymity solutions based on Prim algorithm with different edge weights.

In Table 2, Fig. 9 and Fig. 10, we can get that the number of anonymity solutions based on Prim algorithm is 45261 for all choices of the edge weights. However, the anonymity time with different edge weights are different. When the edge weights are in 1–5, the anonymity time is 195.288 s. When the edge weights are in 2–6, the anonymity time is 137.863 s. When the edge weights are in 3–7, the anonymity time is 183.463 s. When the edge weights are in 4–8, the anonymity time is 181.371 s. When the edge weights are in 5–9, the anonymity time is 172.721 s. When the edge weights are in 6 ~ 10, the anonymity time is 154.81 s.

It can be seen that the Prim algorithm has significant advantages in the number of anonymity solutions. This gives web publishers a better choice. Although Kruskal algorithm has more advantages in the anonymity time, the Prim algorithm does not cost much more time than that of Kruskal algorithm. To a certain extent, anonymity based on Prim algorithm is better.

## 5 Conclusion

In this paper, we provided an effective solution to the anonymization of global edge weighted social network graphs. We first proposed a linear programming model to effectively preserve properties of the graph. At the same time, we considered the minimum spanning tree and the shortest path between multiple sources, and showed how to use the Kruskal algorithm and Prim algorithm to anonymize the original edge weighted graph. Finally, we executed the experiments on the number of anonymity solutions and time with different edge weights to analyze the Kruskal algorithm and Prim algorithm to verify their anonymity feasibility.

**Acknowledgment.** The authors declare that there is no conflict of interest regarding the publication of this paper. This work was supported in part by the National Natural Science Foundation of China (No. 62171132, No. 62102088, No. U1905211, No. 61771140), Fok Ying Tung Education Foundation (No. 171061), Natural Science Foundation of Fujian Province (No. 2021J05228), and Fujian University of Technology (No. GJ-YB-20-06).

## References

1. Siddula, M., Li, Y., Cheng, X., Tian, Z., Cai, Z.: Anonymization in online social networks based on enhanced equi-cardinal clustering. IEEE Trans. Comput. Soc. Syst. **6**(4), 809–820 (2019)
2. Huang, H., Zhang, D.J., Xiao, F., Wang K., Wang, R.: Privacy-preserving approach PBCN in social network with differential privacy. IEEE Trans. Netw. Serv. Manage. **17**(2), 931–945 (2020)
3. Safi, S.M., Movaghar, A., Mahmoodzadeh, K.S.: A framework for protecting privacy on mobile social networks. Mobile Netw. Appl. **26**, 1281–1299 (2021)
4. Das, S., Eğecioğlu, Ö., Abbadi, A.E.: Anonimos: an LP based approach for anonymizing weighted social network graphs. IEEE Trans. Knowl. Data Eng. **24**(4), 590–604 (2010)
5. Das, S., Eğecioğlu, Ö., Abbadi, A.E.: Anonymizing weighted social network graphs. In: IEEE International Conference on Data Engineering, pp. 904–907. IEEE, Long Beach (2010)

6. Zhang, X., Zhu, J.: Skeleton of weighted social network. Phys. A Stat. Mech. Appl. **392**(6), 1547–1556 (2013)
7. Wang, S.-L., Tsai, Y.-C., Kao, H.-Y., Ting, I.-H., Hong, T.-P.: Shortest paths anonymization on weighted graphs. Int. J. Softw. Eng. Know. Eng. **23**(01), 65–79 (2013)
8. Tsai, Y.-C., Wang, S.-L., Kao, H.-Y., Hong, T.-P.: Edge types vs privacy in k-anonymization of shortest paths. Appl. Soft Comput. **31**, 348–359 (2015)
9. Liu, Q., Li, F., Yang, S., Wu, J.: Preserving privacy with probabilistic indistinguishability in weighted social networks. IEEE Trans. Parallel Distrib. Syst. **28**(5), 1417–1429 (2017)
10. Dou, K., Gou, B., Kuang, L.: A privacy-preserving multimedia recommendation in the context of social network based on weighted noise injection. Multimedia Tools Appl. **78**(6), 26907–26926 (2019)
11. Wang, D., Long, S.: Boosting the accuracy of differentially private in weighted social networks. Multimedia Tools Appl. **78**(24), 34801–34817 (2019)
12. Walia, G.S., Jain, G., Bansal, N., Singh, K.: Adaptive weighted graph approach to generate multimodal cancelable biometric templates. IEEE Trans. Inf. Forensics Secur. 15, 1945–1958 (2020)
13. Zhao, P., Yu, J., Zhang, H., Qin, Z., Wang, C.: How to securely outsource finding the min-cut of undirected edge-weighted graphs. IEEE Trans. Inf. Forensics and Secur. **15**, 315–328 (2020)
14. Yin, L., Feng, J., Xun, H., Sun, Z., Cheng, X.: A privacy-preserving federated learning for multiparty data sharing in social IoTs. IEEE Trans. Netw. Sci. Eng. **8**(3), 2706–2718 (2021)