A Fast Approximation Algorithm for the Top-K Group Betweenness Centrality

Wenzheng Xu, *Senior Member, IEEE*, Jing Li, Weifa Liang, *Fellow, IEEE*, Zichuan Xu, *Member, IEEE*, Jian Peng, Pan Zhou, *Member, IEEE*, Cheng Zhao, Binyu Yan, Xiaohua Jia, *Fellow, IEEE*, and Jeffrey Xu Yu, *Senior Member, IEEE*

***Abstract*—Betweenness centrality is one of the key centrality measures in many applications including community detections in biological networks, vulnerability detections in communication networks, misinformation filtering in social networks, etc. The top-K group betweenness centrality problem is to find a group of K nodes from a network so that the total fraction of shortest paths that pass through the K nodes is maximized. Existing studies proposed randomized sampling algorithms for the problem. We notice that the existing studies ensured that, the maximum deviation of the estimated centrality of every group from its expectation is no greater than a small given threshold for all potential groups with no more than K nodes, thereby generating too many samples, as the number of such groups is prohibitively large. In contrast, in this paper we first devise a novel algorithm that enables to estimate the centrality of a tentative group adaptively, and the algorithm immediately stops once the centrality is large enough; otherwise, the algorithm uses more samples to find a better group. We then theoretically show that, even the proposed algorithm uses much less samples, it still can find a performance-guaranteed group with high probability. Experimental results with real-world networks demonstrate that the number of samples used by the proposed algorithm is up to 36 times smaller than the state-of-the-art, while the centrality of the group found by the algorithm is no more than 4.5% smaller than the latter.**

***Index Terms*—Group betweenness centrality, approximation algorithms, randomized algorithms.**

Received 30 September 2025; revised 16 March 2026; accepted 18 March 2026. Date of publication 24 March 2026; date of current version 1 May 2026. The work by Wenzheng Xu was supported by the National Natural Science Foundation of China (NSFC) under Grant 62272328 and in part by Sichuan Science and Technology Program under Grant 2024NSFJQ0026. The work by Weifa Liang was supported by the Research Grants Council of the Hong Kong Special Administrative Region, China, under Grant CityU 11202723 and Grant CityU 11202824. The work by Jian Peng was supported by the Cooperative Program of Sichuan University and Yibin under Grant 2020CDYB-30, in part by the Cooperative Program of Sichuan University and Zigong under Grant 2022CDZG-6, in part by the Key R&D Program of Sichuan Province of China under Grant 22ZDYF3599, and in part by Sichuan Science and Technology Program under Grant 2022ZDZX0011. Recommended for acceptance by R. Akbarinia. *(Corresponding author: Jian Peng.)*

Wenzheng Xu, Jian Peng, and Binyu Yan are with the College of Computer Science, Sichuan University, Chengdu 610065, China (e-mail: wenzheng.xu3@gmail.com; jianpeng@scu.edu.cn; yanby@scu.edu.cn).

Jing Li, Weifa Liang, and Xiaohua Jia are with the Department of Computer Science, City University of Hong Kong, Hong Kong, SAR China (e-mail: jing.li@cityu.edu.hk; weifa.liang@cityu.edu.hk; csjia@cityu.edu.hk).

Zichuan Xu is with the School of Software, Dalian University of Technology, Dalian 116620, China (e-mail: z.xu@dlut.edu.cn).

Pan Zhou is with the School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: panzhou@hust.edu.cnMember).

Cheng Zhao is with Sichuan Province Commercial Investment Group Company, Ltd., Chengdu 610000, China (e-mail: 1404761042@qq.com).

Jeffrey Xu Yu is with the Data Science and Analytics Thrust, The Hong Kong University of Science and Technology, Guangzhou 510530, China (e-mail: jeffreyxuyu@hkust-gz.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TKDE.2026.3677004, provided by the authors.

Digital Object Identifier 10.1109/TKDE.2026.3677004

I. Introduction

Betweenness Centrality (BC) is one of the key measures of central nodes in network analysis, where the betweenness centrality of a node in a network is the total fraction of shortest paths that pass through it [10], [21], [36]. The concept of BC has various applications, including close community detection in social and biological networks [11], [19], [22], [30], vulnerability detection in communication networks or power grid networks [15], [17], [20], misinformation (e.g., rumors) filtering in social networks [5], [12], [13], [16], [29], [32], [34], etc.

In this paper, we study a top-K Group Betweenness Centrality (GBC) problem [26], which is to find a group of K nodes from a network so that the total fraction of shortest paths that pass through at least one of the nodes in the group is maximized. The betweenness centrality of a group can be used to measure the influence of the group over the information flow in the entire network.

Unlike the calculation problem of the betweenness centrality of each node that can be solved in polynomial time, the top-K GBC problem is NP-hard [9], and the best approximation algorithm for it so far can find a $(1-1/e)$-approximate solution by Puzis et al. [26] with a time complexity of $O(n^3)$, where e is the base of the natural logarithm, and n is the number of nodes in a network. Its time complexity however is prohibitively high for real-world large scale networks, such as the Internet, Facebook, Twitter, etc.

To efficiently find a near-optimal solution to the top-K GBC problem in large networks, researchers studied the trade-off between the quality of the solution found and the running time of the proposed algorithms [19], [24]. They randomly sample shortest paths from the network, and find a group of K nodes to cover the maximum number of paths, where a path is covered by the group if a node in the path is contained in the group. They showed that the found group is a $(1-1/e-\epsilon)$-approximate solution with high probability, if the number of sampled shortest paths is sufficiently large, where ϵ is a given error ratio with $0<\epsilon<1-1/e$.

Although the studies in [19], [24] have conducted pioneering researches for the top-K GBC problem, the running times of the algorithms [19], [24] are still long, especially when the network size is large and people want to find the top-K nodes as fast as possible. Therefore, faster yet performance-guaranteed algorithms for the problem are desperately needed.

We notice that, to find a $(1-1/e-\epsilon)$-approximate group with high probability, the studies in [19], [24] conservatively ensured that, the maximum deviation of the estimated centrality of every group from its expectation is no more than a small given threshold $\frac{\epsilon}{2}opt$ for all potential groups with each having no more than K nodes, where opt is the optimal value of the problem. It can be seen that the studies in [19], [24] need to sample many shortest paths, as there are as many as n^K groups with no more than K nodes in a network.

Different from the existing studies in [19], [24], in this paper, we propose a novel approximation algorithm to estimate the centrality of a tentative group, and the algorithm immediately stops once the centrality is no less than the required performance $(1-1/e-\epsilon)opt$; otherwise, the algorithm examines more samples to find a better group. By doing so, the algorithm samples much less numbers of shortest paths than the existing studies. Notice that when the algorithm stops, the maximum deviation of the estimated centrality of every group from its expectation is not necessarily smaller than the threshold $\frac{\epsilon}{2}opt$ required by the existing studies, for all potential groups with no more than K nodes.

The contributions of this paper are as follows. **(i)** Unlike existing studies that posed the stringent maximum deviation of the estimated centrality of every group from its expectation, we propose a novel performance-guaranteed algorithm to estimate the centrality of a tentative group adaptively, and it immediately stops when the centrality is large enough, thereby sampling much less numbers of shortest paths. **(ii)** We theoretically show that, even the proposed algorithm uses much less samples, it still can find a $(1-1/e-\epsilon)$-approximate group with high probability, where ϵ is a given error ratio with $0<\epsilon<1-1/e$. **(iii)** We conduct extensive experiments in real-world networks to validate the effectiveness and accuracy of the solution by the proposed algorithm. Experimental results demonstrate that the number of samples used by the algorithm is up to 36 times smaller than the state-of-the-art [24], while the betweenness centrality of its found group is comparable with the latter, e.g., no more than 4.5% smaller.

The rest of the paper is organized as follows. We review related studies on the topic in Section II. We introduce preliminaries in Section III. We propose a fast randomized algorithm for the problem in Section IV, which finds a $(1-1/e-\epsilon)$-approximate solution with high probability, and we also analyze the performance of the proposed algorithm in Section V. We evaluate the algorithm performance empirically in Section VI and we conclude the paper in Section VII.

II. Related Work

The top-K group betweenness centrality (GBC) problem has been shown to be NP-hard [9], and some pioneering studies have been taken in the past decades. For example, Puzis et al. [26] devised a $(1-1/e)$-approximation algorithm for the problem with a time complexity $O(n^3)$, while Dolev et al. [8] proved the approximation ratio $1-1/e$ in [26]. Fink et al. [9] considered a more generalized case of the GBC problem where the cost of choosing a different node is different, subject to the total cost budget of all chosen nodes.

Both the time complexity $O(n^3)$ and space complexity $O(n^2)$ of the algorithm in [26] are prohibitively high for large networks, and researchers studied non-trivial trades-off between the quality of found solutions and algorithms' running time [19], [24], [35], by proposing $(1-1/e-\epsilon)$-randomized algorithms with high probability $1-\gamma$, where ϵ is given error ratio with $0<\epsilon<1-1/e$, and γ is given error probability. Yoshida [35] used the pair sampling technique [4], in which each sample includes all shortest paths between a randomly chosen pair of nodes. The number of chosen pairs in [35] is $L_1 = O(\frac{\log\frac{1}{\gamma}+\log n^2}{\epsilon^2\mu_{opt}^2})$, where μ_{opt} is the normalization of the optimal value opt with $\mu_{opt}=\frac{opt}{n(n-1)}$ and $0<\mu_{opt}\le 1$. However, Mahmoody et al. [19] pointed out the number L_1 of chosen pairs is inadequate for finding a $(1-1/e-\epsilon)$-approximate solution. On the other hand, both the algorithms in [19], [24] adopted the path sampling technique, in which each sample is a single shortest path between a randomly chosen pair of nodes. The number of samples in [19] is $L_2 = O(\frac{\log\frac{1}{\gamma}+K\log n}{\epsilon^2\mu_{opt}})$, while Pellegrina [24] recently reduced the number of samples to $L_3 = O(\frac{\log\frac{1}{\gamma}+K(\log K)(\log\log n)(\log\frac{1}{\mu_{opt}})}{\epsilon^2\mu_{opt}})$ by utilizing Rademacher averages to estimate the maximum deviation of the estimated centrality of every group from its expectation. In contrast, in this paper, we estimate the centrality of a tentative group adaptively, and our algorithm immediately stops when the centrality is large enough, thereby significantly reducing the number of sampled shortest paths. Notice that this is an improved journal version from a conference paper [33].

We notice that the calculation of the betweenness centrality of each node has attracted lots of attentions in past years. For the exact calculation of node betweenness centrality, Brandes [3] proposed the fastest algorithm with time complexity of $O(nm)$, where n and m are the numbers of nodes and edges, respectively. Furthermore, there are randomized algorithms for the problem. For example, Riondato et al. [27] proposed a rigorous sampling algorithm by the theory of Rademacher averages and pseudodimension. Cousins et al. [7] addressed the limitations of the algorithm in [27] by using the Monte Carlo empirical Rademacher averages and variance-aware tail bounds. Pellegrina et al. [25] adopted non-uniform bounds for different subsets of nodes. Borassi et al. [2] improved the algorithm in [27] by assigning different confidences on the estimated centralities of different nodes.

III. Preliminaries

A. Network Model

We consider a large-scale network $G=(V,E)$, which represents a social network, a computer communication network, or an author citation network. V and E represent the sets of nodes and edges in the network, respectively. Let $V = \{v_1, v_2, \ldots, v_n\}$, where n is the number of nodes in V. In addition, m is the number of edges in E. The edges in the network may be undirected or directed.

For any two nodes s and t in V, the length of a simple path starting from s and ending at t is the number of edges in the path, and denote by $d(s,t)$ the length of a *shortest* path in G from s to t.

B. Group Betweenness Centrality

Denote by σ_{st} the number of shortest paths in G from nodes s to t, where $\sigma_{st} \geq 1$. Especially, we define $\sigma_{st} = 1$ if $s = t$. For any node $v \in V$, denote by $\sigma_{st}(v)$ the number of shortest paths in G from s to t that pass through v, which is defined as

$$\sigma_{st}(v) = \begin{cases} \sigma_{sv} \cdot \sigma_{vt}, & \text{if } d(s,t) = d(s,v) + d(v,t), \\ 0, & \text{otherwise}, \end{cases} \quad (1)$$

where equation $d(s,t) = d(s,v) + d(v,t)$ indicates that v lies on a shortest path in G from s to t.

Similar to the definition of $\sigma_{st}(v)$, for any group C of nodes in V with $C \subseteq V$, denote by $\sigma_{st}(C)$ the number of shortest paths in G from s to t that pass through at least one node in C. The *group betweenness centrality (GBC)* of a group C is the total fraction of shortest paths in G that pass through the nodes in C. Specifically, the group betweenness centrality $B(C)$ of C is defined as

$$B(C) = \sum_{s \in V} \sum_{t \in V, s \neq t} \frac{\sigma_{st}(C)}{\sigma_{st}}, \quad (2)$$

where $\frac{\sigma_{st}(C)}{\sigma_{st}}$ is the ratio of the number $\sigma_{st}(C)$ of shortest paths passing through as least one node in C to the total number σ_{st} of shortest paths in G from s to t.

Note that, similar to the studies in [1], [19], [24], [26], [35], in the calculation of the group betweenness centrality of C, the shortest paths that start from or end at the nodes in C are included, as the information originating at a node in C is seen by the node and thus it should be counted. Furthermore, the inclusion of the shortest paths with end nodes in C just has an addition of a constant $n(n-1) - (n-K)(n-K-1) = 2Kn - K^2 - K$, and the additional constant $2Kn - K^2 - K$ is much smaller than $B(C) = O(n^2)$ [19] when $K = |C| \ll n$.

C. Problem Definition

In this paper, we consider a *top-K group betweenness centrality problem*. Specifically, given a network $G = (V, E)$ and a positive integer K, the problem is to find a group C of K nodes in G, such that the group betweenness centrality of C is maximized, i.e., $\max_{C \subseteq V, |C| = K}\{\, B(C)\,\}$. The problem is NP-hard [26], implying that it is unlikely to find an optimal solution for the problem in polynomial time unless P=NP.

D. The State-of-the-Art Algorithm for the Problem

We briefly introduce the state-of-the-art algorithm [24] for the problem. The basic idea of the algorithm is to randomly choose L shortest paths from the network, and find a group C of K nodes so that the number of shortest paths 'covered' by group C is maximized, where a path is 'covered' by group C if at least one node in the path is contained in C.

Given the number L of to-be-chosen shortest paths in G, the path sampling procedure is briefly introduced [24], [25] as follows. The starting node s and ending node t of a shortest path are first randomly chosen with $s \neq t$. All shortest paths from nodes s to t then are found by performing a balanced bidirectional BFS (Breadth-First Search) [2], [25], where the balanced bidirectional BFS indicates that, two BFSes from nodes s and t are simultaneously performed in a way that the two BFSes explore approximately the same number of edges, and the search stops when all shortest paths from s to t are found. The time complexity of such a bidirectional BFS is only $O(m^{\frac{1}{2}+o(1)})$ with high probability in many realistic random networks [2], though degrade to $O(m)$ in the worst case. Note that the average time complexity $O(m^{\frac{1}{2}+o(1)})$ is much smaller than the time complexity $O(m)$ of the traditional BFS starting from only node s, where m is the number of edges in a network. Finally, a shortest path from s to t is randomly chosen from all the found shortest paths. It can be seen that the time complexity for randomly sampling L shortest paths is $O(Lm^{\frac{1}{2}+o(1)})$ with high probability in many realistic random networks, and is no greater than $O(Lm)$ in the worst case.

Having obtained L shortest paths, a group C of K nodes can be found by applying the greedy strategy for the problem of covering the maximum number of paths, and the found group C is a $(1 - 1/e)$-approximate solution to the coverage problem [23]. Assume that L' shortest paths are covered by the found group C with $L' \leq L$. The group betweenness centrality $B(C)$ of C can be estimated as

$$\hat{B}_L(C) = \frac{L'}{L} n(n-1). \quad (3)$$

Notice that the estimated centrality $\hat{B}_L(C)$ is **biased** for the centrality expectation $B(C)$, since the found group C highly depends on the L chosen paths. However, the deviation of $\hat{B}_L(C)$ from $B(C)$ becomes smaller with the increase of the number L of sampled paths.

Denote by C^* the optimal group of the problem. Let opt be the value of group C^*, i.e., $opt = B(C^*)$. Denote by $\overline{B_L(C^*)}$ the estimated centrality of group C^* from the L sampled shortest paths. Notice that the estimation $\overline{B_L(C^*)}$ is unbiased, as the group C^* does not depend on the L paths.

The state-of-the-art in [24] samples too many shortest paths to ensure that, the maximum deviation of the estimated centrality of every group from its expectation is no greater than a small given threshold $\frac{\epsilon}{2}opt$ with high probability, for all potential n^K groups with no more than K nodes. Then, the deviation of the biased estimation $\hat{B}_L(C)$ of the found group C from its expectation $B(C)$ is no more than the threshold $\frac{\epsilon}{2}opt$, and the deviation of the unbiased estimation $\overline{B_L(C^*)}$ of the optimal group C^* from its expectation $B(C^*)(= opt)$ is no greater than $\frac{\epsilon}{2}opt$, too. Therefore, group C is a $(1 - 1/e - \epsilon)$-approximate solution with high probability, as $\hat{B}_L(C) \geq (1 - 1/e)\overline{B_L(C^*)}$ [23], [24].

IV. Randomized Algorithm

In this section, we devise a randomized algorithm for the top-K group betweenness centrality problem, which uses much less samples than the state-of-the-art in [24], while still delivers a $(1-1/e-\epsilon)$-approximate solution with a large success probability $1-\gamma$, where ϵ is a given constant with $0<\epsilon<1-1/e$, and γ is a given error probability, e.g., $\epsilon=0.2$ and $\gamma=0.01$.

Recall that *opt* is the optimal value of the top-K GBC problem. Following the definition of the group betweenness centrality in (2) of Section III, the value of *opt* is no larger than $n(n-1)$. Let $Q_{\max}=\lceil\log_b n(n-1)\rceil$, where b is a positive constant with $b>1$ (e.g., $b=1.5$), and we will discuss the choice of b later in Section IV-D. Assume that

$$\frac{n(n-1)}{b^{Q^*}} > opt \geq \frac{n(n-1)}{b^{Q^*+1}}, \tag{4}$$

where $0\leq Q^*\leq Q_{\max}-1$, where the value of Q^* is unknown.

Different from the state-of-the-art in [24] that conservatively ensured that the maximum deviation of the estimated centrality of every group from its expectation is no greater than a small given threshold, the proposed algorithm in this paper first samples some shortest paths and finds a tentative group that covers the maximum number of paths, it then estimates whether the centrality of the group is large enough. If so, the algorithm stops; otherwise, it uses more samples to find a better tentative group. We describe the algorithm as follows.

A. Find a Tentative Group

The proposed algorithm generates two sample sets $\mathcal{S}$ and $\mathcal{T}$ of shortest paths. The first sample set $\mathcal{S}$ is used to find tentative groups, and the second sample set $\mathcal{T}$ is used to calculate unbiased estimated centralities of the groups. Initially, $\mathcal{S}=\emptyset$ and $\mathcal{T}=\emptyset$. Both the numbers of samples in sets $\mathcal{S}$ and $\mathcal{T}$ will grow in the execution of the algorithm.

The algorithm performs iteratively. At the qth iteration with $1\leq q\leq Q_{\max}$, we first obtain a guess g_q on the optimal value *opt* by setting

$$g_q=\frac{n(n-1)}{b^q}. \tag{5}$$

That is, the guesses g_q of *opt* in the $Q_{\max}$ iterations decrease exponentially, which are $\frac{n(n-1)}{b},\frac{n(n-1)}{b^2},\ldots,\frac{n(n-1)}{b^{Q_{\max}}}$, respectively. Let $\alpha=\frac{\epsilon}{2-1/e}$, and $\theta=\frac{0.8+3\epsilon}{\alpha^2}\ln\frac{4}{\gamma}=(2-1/e)^2(\frac{0.8}{\epsilon^2}+\frac{3}{\epsilon})\ln\frac{4}{\gamma}$, which are two constants.

We then randomly sample $L_q-|\mathcal{S}|$ shortest paths by invoking the algorithm in [24], and add the $L_q-|\mathcal{S}|$ shortest paths to $\mathcal{S}$, where

$$L_q = \theta\frac{n(n-1)}{g_q} = \theta b^q. \tag{6}$$

Notice that there are L_q shortest paths in $\mathcal{S}$ after the addition. It can be seen that, the numbers of sampled shortest paths in $\mathcal{S}$ in the $Q_{\max}$ iterations increase exponentially, which are $\theta b,\theta b^2,\ldots,\theta b^{Q_{\max}}$, respectively.

Note that the algorithm in [24] also finds a *tentative* group C_q of K nodes to cover the maximum number of paths in set $\mathcal{S}$, and calculates a *biased* estimated centrality $\hat{B}_{L_q}(C_q)$ of $B(C_q)$, see (3).

We obtain an *unbiased* estimation $\overline{B_{L_q}(C_q)}$ of $B(C_q)$ as follows. We independently sample $L_q-|\mathcal{T}|$ extra shortest paths, and add the $L_q-|\mathcal{T}|$ shortest paths to $\mathcal{T}$. There are L_q shortest paths in $\mathcal{T}$ after the addition. We calculate the number L_q' of paths that are covered by the group C_q. Then, the unbiased estimated centrality of $B(C)$ is

$$\overline{B_{L_q}(C)}=\frac{L_q'}{L_q}n(n-1). \tag{7}$$

We also calculate the relative error β between the *biased* estimated centrality $\hat{B}_{L_q}(C_q)$ and the *unbiased* estimated centrality $\overline{B_{L_q}(C_q)}$ of $B(C_q)$, where $\beta=1-\frac{\overline{B_{L_q}(C_q)}}{\hat{B}_{L_q}(C_q)}$. Then, $\overline{B_{L_q}(C_q)}=(1-\beta)\hat{B}_{L_q}(C_q)$.

B. Estimate Whether the Guess g_q of the Optimal Value opt is Small Enough

When $1\leq q\leq Q^*-1$, it can be seen that the guess $g_q=\frac{n(n-1)}{b^q}>b\cdot opt$ by In (4). We later show that, it is unlikely that the unbiased estimated centrality $\overline{B_{L_q}(C_q)}$ is no less than g_q, i.e., the probability of the random event $\overline{B_{L_q}(C_q)}>g_q$ is very small, since the guess g_q is too large, i.e., larger than $b\cdot opt$. In contrast, if the random event $\overline{B_{L_q}(C_q)}\geq g_q$ occurs, we know that $q\geq Q^*$ with high probability.

We use a counter *cnt* to record the number of times that the random event $\overline{B_{L_q}(C_q)}\geq g_q$ happens. Assume that the random event $\overline{B_{L_q}(C_q)}\geq g_q$ happens at the qth iteration for the first time. Then, the random event will happen in later iterations with high probability, as the value of g_q decreases exponentially when the value of q increases, see the definition of g_q in (5).

The value of *cnt* serves as the following three purposes.

(i) If $cnt=0$, this indicates that the number L_q of sampled shortest paths is not enough, as the guess g_q is larger than $b\cdot opt$ (i.e., $g_q>b\cdot opt$) with high probability. We then need to generate more samples in the next iteration.

(ii) If $cnt\geq 1$, then $q\geq Q^*$ with high probability. Following In (4), we know that the guess $g_q\leq b\cdot opt$.

(iii) The value of *cnt* can be used to estimate the extent to which g_q is smaller than *opt*, and in Section V-B, we will show that

$$g_q\leq\frac{opt}{b^{cnt-2}},\ \forall\ cnt\ \geq 1, \tag{8}$$

and the number of chosen paths now is $L_q=\theta\frac{n(n-1)}{g_q}\geq\theta\frac{n(n-1)}{opt}b^{cnt-2}$ by (6).

C. Find a Better Tentative Group

When $cnt\geq 1$, we find a better tentative group C_q' with K nodes, which is to cover the maximum number of paths in set $\mathcal{S}\cup\mathcal{T}$, by invoking the greedy algorithm in [24], where the greedy algorithm finds a $(1-1/e)$-approximate solution C_q'. Note that there are $2L_q$ paths in set $\mathcal{S}\cup\mathcal{T}$. The greedy algorithm also calculates a biased estimated centrality $\hat{B}_{2L_q}(C'_q)$ of $B(C'_q)$, see (3).

We estimate whether the centrality $B(C'_q)$ is no less than $(1-1/e-\epsilon)opt$ as follows. We start by calculating the smallest error ratio ϵ_1 so that the probability that $B(C'_q) \geq \overline{B_{2L_q}(C'_q)} - \epsilon_1 opt$ is no less than $1-\frac{\gamma}{4}$, where $\overline{B_{2L_q}(C'_q)}$ is an unbiased estimated centrality of group C'_q with $2L_q$ sampled paths. In Lemma 4 of Section V-C, we show that

$$\epsilon_1 = \frac{c_1}{3} + \sqrt{\frac{c_1^2}{9} + 2c_1}, \tag{9}$$

where $c_1 = \frac{\ln\frac{4}{\gamma}}{2\theta b^{cnt-2}}$. It must be mentioned that the calculation of ϵ_1 depends on only the value of L_q, see Lemma 4. It can be seen that the value of ϵ_1 reduces when the value of cnt increases.

Recall that C^* is the optimal group and $B(C^*) = opt$. We later show that the probability that $\overline{B_{2L_q}(C^*)} \geq B(C^*) - \epsilon_1 opt$ is no less than $1-\frac{\gamma}{4}$, see Lemma 5 of Section V-D.

We now estimate whether the centrality $B(C'_q)$ is no less than $(1-1/e-\epsilon)opt$ with high probability.

$$\begin{aligned}
&B(C'_q)\\
&\quad\geq \overline{B_{2L_q}(C'_q)} - \epsilon_1 opt\\
&\quad= (1-\beta)\hat{B}_{2L_q}(C'_q) - \epsilon_1 opt \qquad (10)\\
&\quad= (1-\beta)(1-1/e)\overline{B_{2L_q}(C^*)} - \epsilon_1 opt,\\
&\qquad\quad \text{as } \hat{B}_{2L_q}(C'_q) \geq (1-1/e)\overline{B_{2L_q}(C^*)} \text{ [23], [24]}\\
&\quad\geq (1-\beta)(1-1/e)(B(C^*) - \epsilon_1 opt) - \epsilon_1 opt\\
&\quad= (1-1/e-\epsilon_{sum})opt, \text{ as } B(C^*) = opt, \qquad (11)
\end{aligned}$$

where ϵ_{sum} is the accumulative error ratio with $\epsilon_{sum} = (2-1/e)\epsilon_1 + (1-1/e)\beta(1-\epsilon_1)$. It can be seen that if the value of ϵ_{sum} is no greater than ϵ, group C'_q is a $(1-1/e-\epsilon)$-approximate solution with high probability.

We later show that the accumulative error ratio ϵ_{sum} converges to zero at a rate of $O(\frac{1}{\sqrt{L_q}})$, i.e., $\epsilon_{sum} = O(\frac{1}{\sqrt{L_q}})$, see Lemma 7 in Section V-F.

D. The Choice of the Base b

Recall that the number of sampled shortest paths at the qth iteration is $L_q = \theta b^q = \frac{0.8+3\epsilon}{\alpha^2} b^q \ln\frac{4}{\gamma} = c_2 b^q \ln\frac{4}{\gamma}$, where $b > 1$ and $c_2 = \frac{0.8+3\epsilon}{\alpha^2}$.

On one hand, it can be seen that the term b^q increases more slowly if the base b is smaller with $b > 1$, where $L_q = c_2 b^q \ln\frac{4}{\gamma}$, and the algorithm is more likely to use less samples when the algorithm stops. For example, assume that the accumulative error ratio ϵ_{sum} at some iteration is just slightly larger than the given error ratio ϵ. This indicates that we just need slightly more samples to ensure that ϵ_{sum} is no more than ϵ in the next iteration, and a small base b thus is enough. Otherwise, if a larger base b is adopted, this indicates that we will sample many samples in the next iteration.

On the other hand, to ensure that the probability of the random event $\overline{B_{L_q}(C_q)} \geq g_q$ is small in the first Q^*-1 iterations (i.e., $g_q > b\cdot opt$), the value of b should not be too small; otherwise (b is too small), the number L_q of samples must be very large, so that the random event $\overline{B_{L_q}(C_q)} \geq g_q$ occurs with a small probability when $g_q > b\cdot opt$.

We now find the value of b as follows. In later Lemma 3 in Section V-B, we find a lower bound b' on the base b to ensure that the random event $\overline{B_{L_q}(C_q)} \geq g_q$ occurs with a small probability when $g_q > b\cdot opt$, where

$$b' = \frac{3c_2 + 2 + \sqrt{18c_2+4}}{3c_2 - 2}, \tag{12}$$

and $c_2 = \frac{0.8+3\epsilon}{\alpha^2}$.

We also set a minimum value of b, e.g., $b_{\min} = 1.1$. Finally, the value of b is

$$b = \max\{b',\ b_{\min}\}. \tag{13}$$

It can be seen from (12) and (13) that, the base b is larger if a larger error ratio ϵ is adopted. For example, when the error ratio ϵ is 0.5, then $\alpha = \frac{\epsilon}{2-1/e} = 0.3063$, $c_2 = \frac{0.8+3\epsilon}{\alpha^2} = 24.5$, $b' = 1.35$ by following (12), and $b = \max\{b',\ b_{\min}\} = 1.35$. In contrast, when $\epsilon = 0.1$, $\alpha = \frac{\epsilon}{2-1/e} = 0.0613$, $c_2 = \frac{0.8+3\epsilon}{\alpha^2} = 293$, $b' = 1.087$ by following (12), and $b = \max\{b',\ b_{\min}\} = 1.1$.

There is an important property about the growth of the number L_q of sampled paths, where $L_q = \theta b^q$ and $b > 1$. That is, given a large error ratio ϵ, e.g., $\epsilon = 0.5$, the number L_q of sampled paths increases from a relatively small value θ, e.g., $\theta = 146.8$ when $\epsilon = 0.5$, but increases at a high rate (i.e., the base b is large), e.g., $b = 1.351$ when $\epsilon = 0.5$. On the other hand, given a small error ratio ϵ, e.g., $\epsilon = 0.1$, the number L_q of sampled paths increases from a large value θ, e.g., $\theta = 1755.6$ when $\epsilon = 0.1$, but increases at a smaller rate (i.e., the base b is small), e.g., $b = 1.1$ when $\epsilon = 0.1$.

The randomized algorithm for the top-K group betweenness centrality problem is presented in Algorithm 1.

V. Algorithm Analysis

In this section, we first analyze the probability that $\overline{B_L(C)}$ deviates from $B(C)$, see Section V-A.

We then show the following three claims.

i) When $cnt \geq 1$, the probability that the number L_q of chosen shortest paths is at least $\theta\frac{n(n-1)}{opt}b^{cnt-2}$ is no less than $1-0.03\gamma$, i.e., $Pr[L_q \geq \theta\frac{n(n-1)}{opt}b^{cnt-2}] \geq 1-0.03\gamma$, see Section V-B.

ii) The probability that $B(C'_q) \geq \overline{B_{2L_q}(C'_q)} - \epsilon_1 opt$ is no less than $1-\frac{\gamma}{4}$, when $L_q \geq \theta\frac{n(n-1)}{opt}b^{cnt-2}$ and $cnt \geq 1$, where ϵ_1 was defined in (9), see Section V-C.

iii) The probability that $\overline{B_{2L_q}(C^*)} \geq B(C^*) - \epsilon_1 opt$ is also no less than $1-\frac{\gamma}{4}$, when $L_q \geq \theta\frac{n(n-1)}{opt}b^{cnt-2}$ and $cnt \geq 1$, see Section V-D.

The three claims indicate that C'_q is a $(1-1/e-\epsilon)$-approximate solution with high probability, if $cnt \geq 1$ and $\epsilon_{sum} \leq \epsilon$, see Lemma 6 in Section V-E.

Algorithm 1: Algorithm `AdaAlg` for the Top-K Group Betweenness Centrality Problem.

Input: A network $G = (V, E)$, a budget K, an error ratio ϵ with $0 < \epsilon < 1 - 1/e$, and a high success probability $1 - \gamma$
Output: A group C with K nodes
Let $\alpha \leftarrow \frac{\epsilon}{2-1/e}$;
Calculate the base b by (13);
Let $Q_{\max} \leftarrow \lceil \log_b n(n-1) \rceil$; /* number of iterations */
Let $\theta \leftarrow \frac{0.8+3\epsilon}{\alpha^2} \ln \frac{4}{\gamma}$; /* θ is a constant */
Let $cnt \leftarrow 0$; /* count how many times that the random event $\overline{B_{L_q}(C_q)} \geq g_q$ occurs */
Let $\mathcal{S} \leftarrow \emptyset, \mathcal{T} \leftarrow \emptyset$; // two sample sets of shortest paths
for $q \leftarrow 1$ to $Q_{\max}$ **do**
  Let $g_q \leftarrow \frac{n(n-1)}{b^q}$; /* g_q is a guess of opt */
  Let $L_q \leftarrow \theta \frac{n(n-1)}{g_q} = \theta b^q$; /* the number of needed samples in both sets $\mathcal{S}$ and $\mathcal{T}$ */
  Randomly sample $L_q - |\mathcal{S}|$ shortest paths, add the $L_q - |\mathcal{S}|$ shortest paths to $\mathcal{S}$, find a tentative group C_q of K nodes to cover the maximum number of shortest paths in set $\mathcal{S}$, and calculate its *biased* estimated group betweenness centrality $\hat{B}_{L_q}(C_q)$, by invoking the algorithm in [24].
  Independently sample $L_q - |\mathcal{T}|$ extra shortest paths, add the $L_q - |\mathcal{T}|$ shortest paths to $\mathcal{T}$, and calculate the *unbiased* estimated centrality $\overline{B_{L_q}(C_q)}$ of group C_q;
  Let $\beta \leftarrow 1 - \frac{\overline{B_{L_q}(C_q)}}{\hat{B}_{L_q}(C_q)}$; /* the relative error */
  if $\overline{B_{L_q}(C_q)} \geq g_q$ **then**
    $cnt \leftarrow cnt + 1$;
  else
    /* need more samples in the next iteration */
  end if
  if $cnt \geq 1$ **then**
    /* the guess g_q of opt is no more than $\frac{opt}{b^{cnt-2}}$ */
    Calculate the smallest error ratio ϵ_1 so that the probability that $B(C'_q) \geq \overline{B_{2L_q}(C'_q)} - \epsilon_1 opt$ is no less than $1 - \frac{\gamma}{4}$ by (9), if the group C'_q is needed to be found;
    Let $\epsilon_{sum} \leftarrow \beta(1-1/e)(1-\epsilon_1) + (2-1/e)\epsilon_1$;
    if $\epsilon_{sum} \leq \epsilon$ **then**
      Find a better tentative group C'_q with K nodes to cover the maximum number of shortest paths in set $\mathcal{S} \cup \mathcal{T}$, by invoking the algorithm in [24];
      return group C'_q.
    else
      // Sample more shortest paths in the next iteration
    end if
  end if
end for

We finally analyze the convergence rate of Algorithm 1 in Section V-F, and the expected algorithm time complexity in Section V-G.

Notice that both the samples in sets $\mathcal{S}$ and $\mathcal{T}$ of Algorithm 1 are not independent, so we cannot apply the Chernoff bounds, which are applicable to only independent random variables. Instead, we introduce the notion of martingales and a tail probability bound for martingales, which will be used in our probability analysis.

A *martingale* is a sequence of random variables $X_1, X_2, \ldots$ with finite expectations (i.e., $|\mathbf{E}[X_i]| < +\infty$), such that the conditional expectation of X_l given $X_1, X_2, \ldots, X_{l-1}$ is equal to X_{l-1}, i.e., $\mathbf{E}[X_l] = X_{l-1}$ for $l \geq 2$.

A martingale has a Chernoff-like tail bound as follows.

Lemma 1: (Theorem 18 in [6]) Let $X_1, X_2, \ldots$ be a martingale, assume that $|X_1| \leq M$ and $|X_l - X_{l-1}| \leq M$, where $2 \leq l \leq L$, M is a given positive constant, and L is a positive integer with $L \geq 2$. In addition, assume that $\mathbf{Var}[X_1] + \sum_{l=2}^{L} \mathbf{Var}[X_l \mid X_1, X_2, \ldots, X_{l-1}] \leq \xi$, where ξ is a given constant, $\mathbf{Var}[X_1]$ is the variance of X_1, and $\mathbf{Var}[X_l \mid X_1, X_2, \ldots, X_{l-1}]$ is the conditional variance of X_l given $X_1, X_2, \ldots, X_{l-1}$. Then,

$$Pr[X_L - \mathbf{E}[X_L] \geq \lambda] \leq exp\left(-\frac{\lambda^2}{2\xi + \frac{2}{3}M\lambda}\right) \tag{14}$$

A. The Probability That $\overline{B_L(C)}$ Deviates From Its Expectation $B(C)$

We analyze the probability that $\overline{B_L(C)}$ deviates from $B(C)$ in the following lemma.

Lemma 2: Given a network $G(V, E)$, a group C with K nodes, and L randomly sampled shortest paths in Algorithm 1, assume that L_c shortest paths are covered by the group C. The group betweenness centrality $B(C)$ of C from the L paths can be estimated as $\overline{B_L(C)} = \frac{L_c}{L} n(n-1)$. For any given positive constant λ, we have

$$Pr\left[\overline{B_L(C)} - B(C) \geq \lambda B(C)\right] \leq exp\left(-L\frac{\lambda^2 B(C)}{(2+\frac{2}{3}\lambda)n(n-1)}\right) \tag{15}$$

$$Pr\left[\overline{B_L(C)} - B(C) \leq -\lambda B(C)\right] \leq exp\left(-L\frac{\lambda^2 B(C)}{(2+\frac{2}{3}\lambda)n(n-1)}\right). \tag{16}$$

Proof: The proof is contained in the supplementary file. ∎

B. The Probability That $L_q \geq \theta \frac{n(n-1)}{opt} b^{cnt-2}$

Lemma 3: Assume that the random event $\overline{B_{L_q}(C_q)} \geq g_q$ occurs at the qth iteration in Algorithm 1, and the accumulative times cnt of such event is no less than one, i.e., $cnt \geq 1$. Then, the probability that the number L_q of sampled shortest paths is at least $\theta \frac{n(n-1)}{opt} b^{cnt-2}$ is no less than $1 - 0.03\gamma$, when $n \leq 10^{10^{10}}$, i.e., $Pr[L_q \geq \theta \frac{n(n-1)}{opt} b^{cnt-2}] \geq 1 - 0.03\gamma$.

Proof: We show that the probability $q \geq Q^* + cnt - 1$ is no less than $1 - 0.03\gamma$, where $\frac{n(n-1)}{b^{Q^*}} > opt \geq \frac{n(n-1)}{b^{Q^*+1}}$ by In (4). Then,

$$\begin{aligned} g_q &= \frac{n(n-1)}{b^q}, \text{ by } Eq. \text{ (5)} \\ &\leq \frac{n(n-1)}{b^{Q^*+cnt-1}}, \text{ as } q \geq Q^* + cnt - 1 \\ &= \frac{n(n-1)}{b^{(Q^*+1)}} \frac{1}{b^{cnt-2}} \\ &\leq \frac{opt}{b^{cnt-2}}, \text{ as } \frac{n(n-1)}{b^{Q^*+1}} \leq opt. \end{aligned} \tag{17}$$

Following the definition of L_q in (6), we have $L_q = \theta \frac{n(n-1)}{g_q} \geq \theta \frac{n(n-1)}{opt} b^{cnt-2}$.

To prove $Pr[q \geq Q^* + cnt - 1] \geq 1 - 0.03\gamma$, we show that $Pr[q < Q^* + cnt - 1] \leq 0.03\gamma$ as follows. Since $q < Q^* + cnt - 1$, we know that $q \leq Q^* + cnt - 2$, as q is an integer. Then, the random event $\overline{B_{L_q}(C_q)} \geq g_q$ occurs at least once in the first $Q^* - 1$ iterations, as the event happens at most $cnt - 1 (= Q^* + cnt - 2 - Q^* + 1)$ times from the Q^*th iteration to the $(Q^* + cnt - 2)$th iteration.

In the following, we show that the probability that the random event $\overline{B_{L_q}(C_q)} \geq g_q$ occurs in the first $Q^* - 1$ iterations is no larger than 0.03γ.

Given the qth iteration, consider the random event $\overline{B_{L_q}(C_q)} \geq g_q$. Recall that $g_q = \frac{n(n-1)}{b^q}$ and $L_q = \theta \frac{n(n-1)}{g_q} = \theta b^q = \frac{0.8+3\epsilon}{\alpha^2} b^q \ln \frac{4}{\gamma}$, where $\theta = \frac{0.8+3\epsilon}{\alpha^2} \ln \frac{4}{\gamma}$, and $\alpha = \frac{\epsilon}{2-1/e}$.

Since $1 \leq q \leq Q^* - 1$ and $\frac{n(n-1)}{b^{Q^*}} > opt \geq \frac{n(n-1)}{b^{Q^*+1}}$, then $g_q \geq b^{Q^*-q} opt$, where $Q^* - q \geq 1$ and $b > 1$.

The *basic idea* behind the proof that the random event $\overline{B_{L_q}(C_q)} \geq g_q$ occurs with a small probability is that, since the guess g_q is at least $b^{Q^*-q} \cdot opt \geq b^{Q^*-q} \cdot B(C_q)$, then it is unlikely that the estimated centrality $\overline{B_{L_q}(C_q)}$ of $B(C_q)$ is larger than $b^{Q^*-q} \cdot B(C_q)$, if the number L_q of samples is sufficiently large, where $b^{Q^*-q} \geq b > 1$. Specifically, given the qth iteration with $1 \leq q \leq Q^* - 1$, we have

$$\begin{aligned} &Pr[\overline{B_{L_q}(C_q)} \geq g_q] \\ &= Pr[\overline{B_{L_q}(C_q)} - B(C_q) \geq g_q - B(C_q)] \\ &\leq Pr[\overline{B_{L_q}(C_q)} - B(C_q) \geq g_q - opt], \\ &\quad \text{as } B(C_q) \leq opt \text{ and } g_q - B(C_q) \geq g_q - opt \\ &= Pr[\overline{B_{L_q}(C_q)} - B(C_q) \geq \frac{g_q - opt}{B(C_q)} B(C_q)] \\ &\leq exp\left(-L_q \frac{\lambda^2 B(C_q)}{(2 + \frac{2}{3}\lambda) n(n-1)}\right) \\ &\quad \text{by } In\ eq. \text{ (15) with } \lambda = \frac{g_q - opt}{B(C_q)} \\ &= exp\left(-L_q \frac{(\frac{g_q - opt}{B(C_q)})^2 B(C_q)}{(2 + \frac{2}{3}\frac{g_q - opt}{B(C_q)}) n(n-1)}\right) \\ &= exp\left(-L_q \frac{(g_q - opt)^2}{(2B(C_q) + \frac{2}{3}(g_q - opt)) n(n-1)}\right) \\ &\leq exp\left(-L_q \frac{(g_q - opt)^2}{(\frac{2}{3}g_q + \frac{4}{3}opt) n(n-1)}\right), \text{ as } B(C_q) \leq opt \\ &= exp\left(-L_q \left(1.5 - \frac{4.5opt}{g_q + 2opt}\right) \frac{g_q - opt}{n(n-1)}\right) \\ &\leq exp\left(-L_q \left(1.5 - \frac{4.5}{b^{Q^*-q} + 2}\right) \frac{g_q - opt}{n(n-1)}\right), \\ &\quad \text{as } g_q \geq b^{Q^*-q} \cdot opt \\ &\leq exp\left(-\theta b^q \left(1.5 - \frac{4.5}{b^{Q^*-q} + 2}\right) \frac{\frac{n(n-1)}{b^q} - opt}{n(n-1)}\right), \\ &\quad \text{as } L_q = \theta b^q \text{ and } g_q = \frac{n(n-1)}{b^q} \\ &= exp\left(-\theta \left(1.5 - \frac{4.5}{b^{Q^*-q} + 2}\right) \left(1 - \frac{b^q opt}{n(n-1)}\right)\right), \\ &\leq exp\left(-\theta \left(1.5 - \frac{4.5}{b^{Q^*-q} + 2}\right) \left(1 - \frac{1}{b^{Q^*-q}}\right)\right), \\ &\quad \text{as } opt \leq \frac{n(n-1)}{b^{Q^*}} \text{ and } q \leq Q^* - 1. \end{aligned} \tag{18}$$

By the union bound, the probability that the random event $\overline{B_{L_q}(C_q)} \geq g_q$ occurs at least once in the first $Q^* - 1$ iterations is no greater than

$$\begin{aligned} &\sum_{q=1}^{Q^*-1} Pr[\overline{B_{L_q}(C_q)} \geq g_q] \\ &= \sum_{j=1}^{Q^*-1} exp\left(-\theta \left(1.5 - \frac{4.5}{b^j + 2}\right) \left(1 - \frac{1}{b^j}\right)\right). \end{aligned} \tag{19}$$

Let function $f(x) = (1.5 - \frac{4.5}{b^x+2})(1 - \frac{1}{b^x}) = 1.5 - \frac{1.5}{b^x} - \frac{4.5}{b^x+2} + \frac{4.5}{b^{2x}+2b^x}$, where b is a constant with $b > 1$ and $x \geq 1$.

To analyze the property of function $f(x)$, we first bound the value of the base b. Recall that $0 < \epsilon < 1 - 1/e$, $\alpha = \frac{\epsilon}{2-1/e}$, and $c_2 = \frac{0.8+3\epsilon}{\alpha^2} = (2 - 1/e)^2(\frac{0.8}{\epsilon^2} + \frac{3}{\epsilon})$. Then, $c_2 > (2 - 1/e)^2(\frac{0.8}{(1-1/e)^2} + \frac{3}{1-1/e}) \geq 17.98$. Following (12), $b' = \frac{3c_2+2+\sqrt{18c_2+4}}{3c_2-2}$. Then, $1 < b' < \frac{3*17.98+2+\sqrt{18*17.98+4}}{3*17.98-2} < 1.426$. On the other hand, following (13), we know $b = \max\{b',\ b_{\min}\}$ and $1.1 \leq b \leq 1.426$, where $b_{\min} = 1.1$.

We also bound the value of θ, where $\theta = c_2 \ln \frac{4}{\gamma}$. Existing studies showed that the number of sampled shortest paths is proportional to $\log \frac{1}{\gamma}$, thus is insensitive to the value of γ [24]. WLOG, we assume that $\gamma \leq 0.01$. Since $c_2 > 17.98$, then $\theta = c_2 \ln \frac{4}{\gamma} \geq 17.98 \ln \frac{4}{0.01} \geq 100$, and $\frac{1}{\theta} \leq 0.01$.

The first derivative of $f(x)$ with respect to x is

$$\begin{aligned} f'(x) &= \frac{1.5 \ln b}{b^x} + \frac{4.5 b^x \ln b}{(b^x + 2)^2} - \frac{4.5(2b^{2x} \ln b + 2b^x \ln b)}{(b^{2x} + 2b^x)^2} \\ &= 1.5 \ln b \frac{4b^{2x} - 2b^x - 2}{b^x(b^x + 2)^2} \end{aligned} \tag{20}$$

Since $b > 1$ and $x \geq 1$, then $b^x > 1$ and $f'(x) > 0$, which indicates that $f(x)$ is increasing with the growth of x.

We calculate the second order derivative of $f(x)$.

$$f''(x) = 3b^x(b^x+2)(\ln b)^2 \frac{-2b^{3x}+6b^{2x}+3b^x+2}{b^{2x}(b^x+2)^4} \tag{21}$$

Let $f''(x) = 0$. Then, $b^x = 3.5$ and $x = \log_b 3.5$. This indicates that the increasing rate $f'(x)$ becomes faster when x increases from 1 to $\log_b 3.5$, while the increasing rate $f'(x)$ becomes slower when x is larger than $\log_b 3.5$.

For function $f(x)$, we find the largest integer J, so that $f(2) - f(1) \geq \frac{1}{\theta}, f(3) - f(2) \geq \frac{1}{\theta}, \ldots, f(J) - f(J-1) \geq \frac{1}{\theta}$, while $f(J+1) - f(J) < \frac{1}{\theta}$. We now prove that $J \geq \log_b 50 - 1$. First, when $b = b_{\min} = 1.1$, it can be seen that $f(2) - f(1) = 0.01263 \geq \frac{1}{\theta} = 0.01$. Therefore, for any base b with $b_{\min} = 1.1 \leq b \leq 1.426$, we know that $f(2) - f(1) \geq 0.01263 \geq \frac{1}{\theta}$ as $\log_b 3.5 \geq 2$ when $1.1 \leq b \leq 1.426$. Then, it can be seen that when $b^x = 50$, the increasing rate $f'(x)$ is $1.5 \ln b \frac{4b^{2x}-2b^x-2}{b^x(b^x+2)^2} = 1.5 \ln b \frac{4*50^2-2*50-2}{50(50+2)^2} \geq 0.109 \ln b \geq 0.109 * \ln 1.1 \geq 0.01 = \frac{1}{\theta}$ by (20) and $b \geq 1.1$. Since function $f(x)$ is increasing (i.e., $f'(x) > 0$), $f(2) - f(1) \geq \frac{1}{\theta}$, the increasing rate $f'(x)$ first becomes faster, then becomes slower, and $f'(\log_b 50) \geq \frac{1}{\theta}$, we know that $f(2) - f(1) \geq \frac{1}{\theta}, f(3) - f(2) \geq \frac{1}{\theta}, \ldots, f(J') - f(J'-1) \geq \frac{1}{\theta}$, where $J' = \lfloor \log_b 50 \rfloor \leq \log_b 50$. Then, $J \geq J' \geq \lfloor \log_b 50 \rfloor \geq \log_b 50 - 1$.

We now bound the In (19) as follows.

$$\begin{aligned}
&\sum_{j=1}^{Q^*-1} exp\left(-\theta\left(1.5 - \frac{4.5}{b^j+2}\right)\left(1 - \frac{1}{b^j}\right)\right) \\
&= \sum_{j=1}^{Q^*-1} exp(-\theta \cdot f(j)), \text{ by the definition of } f(x) \\
&= exp(-\theta \cdot f(1)) + \sum_{j=2}^{J} exp(-\theta \cdot (f(j) \\
&\quad - f(j-1) + f(j-1)) \\
&\quad + \sum_{j=J+1}^{Q^*-1} exp(-\theta \cdot f(j)) \\
&\leq exp(-\theta \cdot f(1)) + \sum_{j=2}^{J} exp\left(-\theta \cdot \left(\frac{1}{\theta} + f(j-1)\right)\right) \\
&\quad + Q^* exp(-\theta \cdot f(J+1)), \text{ as } f(j) - f(j-1) \geq \frac{1}{\theta} \\
&\ldots \\
&\leq exp(-\theta \cdot f(1)) + \sum_{j=2}^{J} exp\left(-\theta \cdot \left(\frac{j-1}{\theta} + f(1)\right)\right) \\
&\quad + Q^* exp(-\theta \cdot f(J+1)) \\
&\leq \sum_{j=1}^{J} exp(-\theta \cdot f(1) - (j-1)) + Q^* exp(-\theta \cdot f(J+1)) \\
&\leq exp(-\theta \cdot f(1)) \frac{1}{1-1/e} + Q^* exp(-\theta \cdot f(J+1))
\end{aligned} \tag{22}$$

Notice that $\theta \cdot f(1) = c_2 \ln \frac{4}{\gamma}(1.5 - \frac{4.5}{b+2})(1 - \frac{1}{b}) \geq c_2 \ln \frac{4}{\gamma}(1.5 - \frac{4.5}{b'+2})(1 - \frac{1}{b'}) = \frac{4}{\gamma}$, as $b \geq b'$ and b' is the root of the equation $c_2(1.5 - \frac{4.5}{b'+2})(1 - \frac{1}{b'}) = 1$. On the other hand, since $J \geq \log_b 50 - 1$, then $J + 1 \geq \log_b 50$, $b^{J+1} \geq 50$, and $f(J+1) = (1.5 - \frac{4.5}{b^{J+1}+2})(1 - \frac{1}{b^{J+1}}) \geq (1.5 - \frac{4.5}{50+2})(1 - \frac{1}{50}) \geq 1.38$. Then, $Q^* exp(-\theta \cdot f(J+1)) \leq Q^* exp(-1.38\theta) = Q^* exp(-1.38 c_2 \ln \frac{4}{\gamma}) \leq Q^* exp(-24.81 \ln \frac{4}{\gamma})$, as $\theta = c_2 \ln \frac{4}{\gamma}$ and $c_2 \geq 17.98$. Since $0 < \gamma \leq 1$, then $Q^* exp(-24.81 \ln \frac{4}{\gamma}) = Q^* exp(-\ln \frac{4}{\gamma} - 23.81 \ln \frac{4}{\gamma}) \leq Q^* exp(-\ln \frac{4}{\gamma} - 33) = \frac{\gamma}{4}\frac{Q^*}{e^{33}} \leq \frac{\gamma}{1000}$, as $Q^* \leq Q_{\max} = \log_b n(n-1) \leq \log_b n^2 \leq \frac{4 \cdot e^{33}}{1000}$ for any reasonable network size n, e.g., $n \leq 10^{10^{10}} < 1.1^{e^{33}/500} \leq b^{e^{33}/500}$, as $1.1 \leq b$.

We finally bound the In (22) as follows.

$$\begin{aligned}
&exp(-\theta \cdot f(1)) \frac{1}{1-1/e} + Q^* exp(-\theta \cdot f(J+1)) \\
&\leq exp\left(-\frac{4}{\gamma}\right) \frac{1}{1-1/e} + \frac{\gamma}{1000} \\
&\quad \text{as } \theta \cdot \mathrm{f}(1) \geq \frac{4}{\gamma} \mathrm{and} \mathrm{Q}^* \exp(-\theta \cdot \mathrm{f}(\mathrm{J}+1)) \leq \frac{\gamma}{1000} \\
&< 0.029\gamma + \frac{\gamma}{1000}, \text{ as } \mathrm{e}^{\mathrm{y}} \geq 13.649\mathrm{y} \text{ with } \mathrm{y} = \frac{4}{\gamma} \geq 4 \\
&= 0.03\gamma
\end{aligned} \tag{23}$$

By combining In (19), (22), and (23), we know that the probability that the random event $\overline{B_{L_q}(C_q)} \geq g_q$ occurs at least once in the first $Q^* - 1$ iterations is no greater than 0.03γ. The lemma then follows. ∎

C. The Probability That $B(C'_q) \geq \overline{B_{2L_q}(C'_q)} - \epsilon_1 opt$

Lemma 4: When $L_q \geq \theta \frac{n(n-1)}{opt} b^{cnt-2}$ and $cnt \geq 1$, the probability that $B(C'_q) \geq \overline{B_{2L_q}(C'_q)} - \epsilon_1 opt$ is no less than $1 - \frac{\gamma}{4}$, where ϵ_1 was defined in (9).

Proof: The proof is in the supplementary file. ∎

D. The Probability That $\overline{B_{2L_q}(C^*)} \geq B(C^*) - \epsilon_1 opt$

Lemma 5: When $L_q \geq \theta \frac{n(n-1)}{opt} b^{cnt-2}$ and $cnt \geq 1$, the probability that $\overline{B_{2L_q}(C^*)} \geq B(C^*) - \epsilon_1 opt$ is no less than $1 - \frac{\gamma}{4}$, where ϵ_1 was defined in (9).

Proof: The proof is in the supplementary file. ∎

E. Approximation Ratio Analysis

Lemma 6: Assume that at the random event $\overline{B_{L_q}(C_q)} \geq g_q$ occurs at the qth iteration in Algorithm 1, $cnt \geq 1$, and $\epsilon_{sum} \leq \epsilon$. Also, assume that the probability that $\beta' \leq \beta$ is no less than $1 - 0.47\gamma$ when Algorithm 1 terminates, where β' is the relative

TABLE I
EIGHT REAL-WORLD NETWORKS IN THE EXPERIMENTS

Dataset	$\|V\|$	$\|E\|$	Type
GrQc [28]	5,244	14,496	undirected
Facebook [31]	63,731	817,090	undirected
Coauthor [18]	53,442	127,968	undirected
DBLP-2011 [14]	986,324	3,353,618	undirected
Epinions [28]	75,879	508,837	directed
Twitter [18]	92,180	377,942	directed
Email-euAll [28]	265,214	420,045	directed
LiveJournal [14]	5,363,260	54,880,888	directed

error between the biased estimated centrality $\hat{B}_{2L_q}(C'_q)$ and the unbiased estimated centrality $\overline{B_{2L_q}(C'_q)}$ of group C'_q, and β is the relative error between the biased estimated centrality $\hat{B}_{L_q}(C_q)$ and the unbiased estimated centrality $\overline{B_{L_q}(C_q)}$ of group C_q. Then, the group C'_q found by the algorithm is a $(1-1/e-\epsilon)$-approximate solution with a probability $1-\gamma$, when $n \leq 10^{10^{10}}$.

Proof: The proof is in the supplementary file. ■

F. Convergence Rate Analysis

Lemma 7: Assume that at the random event $\overline{B_{L_q}(C_q)} \geq g_q$ occurs at the qth iteration in Algorithm 1 and $cnt \geq 1$. Then, the accumulative error ratio ϵ_{sum} of the group C'_q found by Algorithm 1 converges to zero at a rate of $O(\frac{1}{\sqrt{L_q}})$, i.e., $\epsilon_{sum} = O(\frac{1}{\sqrt{L_q}})$.

Proof: The proof is in the supplementary file. ■

G. Time Complexity Analysis

Theorem 1: Given an error ratio ϵ and an error probability γ, Algorithm 1 can find a $(1-1/e-\epsilon)$-approximate solution with a probability $1-\gamma$ for the top-K group betweenness centrality problem. In addition, an upper bound on its expected time complexity is $O(\frac{\log\frac{1}{\gamma}+K(\log K)(\log\log n)(\log\frac{1}{\mu_{opt}})}{\epsilon^2\mu_{opt}} m^{\frac{1}{2}+o(1)})$, where μ_{opt} is the normalization of the optimal value *opt* with $\mu_{opt} = \frac{opt}{n(n-1)}$, n and m are numbers of nodes and edges in the network, respectively.

Proof: The proof is in the supplementary file. ■

VI. PERFORMANCE EVALUATION

A. Experimental Environment Settings

We adopt eight real-world networks, see Table I. The number K of to-be-found nodes for the top-K group betweenness centrality problem is from 20 to 100. The error ratio ϵ is from 0.1 to 0.5, which is less than $1-1/e \approx 0.632$. The error probability is γ is 1%. Then, the success probability $1-\gamma$ is 99%. Notice that we do not vary the value of γ, as the number of sampled shortest paths is proportional to $\log\frac{1}{\gamma}$, thus is insensitive to the value of γ [24].

To study the performance of the proposed algorithm `AdaAlg`, we compare with the following three benchmarks. (i) Algorithm `HEDGE` [19] finds a $(1-1/e-\epsilon)$-approximate solution with a probability $1-\gamma$, and the number of sampled shortest paths is $O(\frac{\log\frac{1}{\gamma}+K\log n}{\epsilon^2\mu_{opt}})$, where μ_{opt} is the normalization of the optimal value *opt* with $\mu_{opt} = \frac{opt}{n(n-1)}$ and $0 < \mu_{opt} \leq 1$. (ii) Algorithm `CentRa` [24] recently reduced the number of samples in [19] to $O(\frac{\log\frac{1}{\gamma}+K\log K}{\epsilon^2\mu_{opt}})$. (iii) Algorithm `EXHAUST` finds an approximate solution with its value very close to $(1-1/e)opt$, by applying Algorithm `HEDGE` with a small error ratio ϵ (e.g., 0.03) and a small error probability γ (e.g., 0.01%). Algorithm `EXHAUST` can be used to show how good or bad the solutions found by the three comparison algorithms `AdaAlg`, `HEDGE`, and `CentRa` are. (iv) Algorithm `Determ` [26] finds a deterministic $(1-1/e)$-approximate solution in time $O(n^3)$. We compared with algorithm `Determ` only in the medium-sized network `GrQc`, due to its sky-high time complexity.

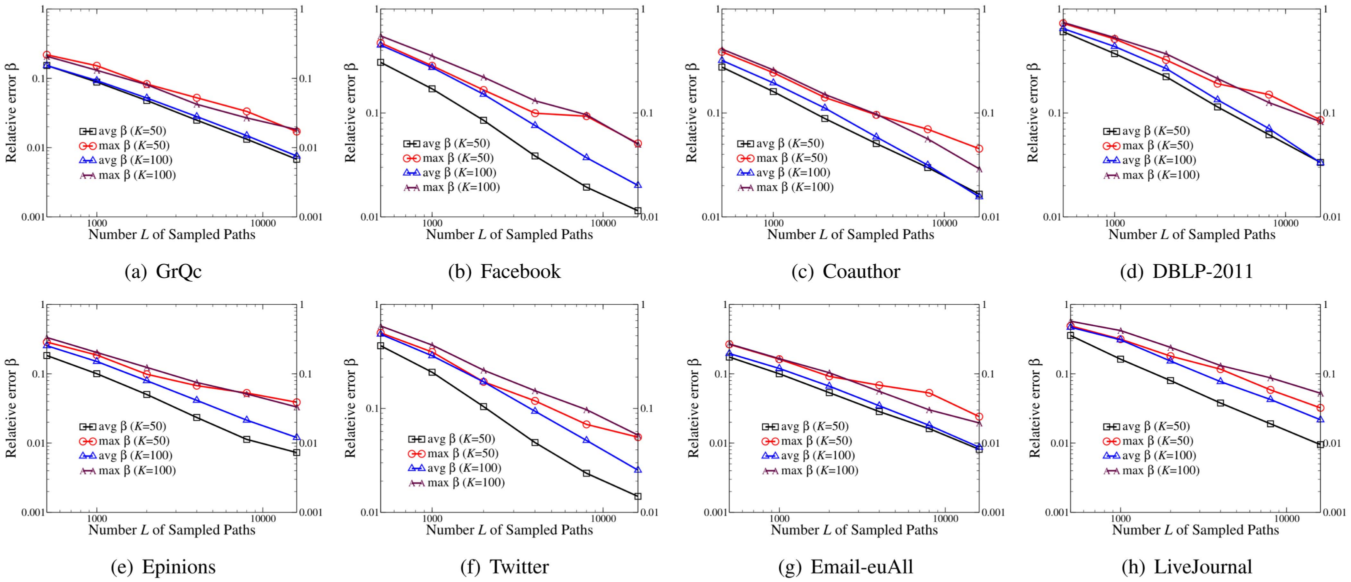
All algorithms are implemented by the programming language C++, and their source codes are publicly available at the website https://github.com/Yu-Huai-M/maxGBC-AdaptiveSamplingAlgorithm. The algorithms are run on a server with an Intel i9-9900 K CPU. The frequency of the CPU is between 3.6 GHz and 5 GHz. The Memory in the server is a 32 GB RAM with DDR4 2,666 MHz. Each algorithm is run 20 times and its average result is shown.

B. The Convergence of the Relative Error β

The proposed algorithm works only when the relative error β between the *biased* and *unbiased* estimated centralities $\hat{B}_L(C)$ and $\overline{B_L(C)}$ converges to zero with the growth of the number L of sampled shortest paths, where $\beta = 1 - \frac{\overline{B_L(C)}}{\hat{B}_L(C)}$. Fig. 1 shows that both the average and maximum relative errors β in 100 simulations significantly become smaller in each of the eight networks, when the number L of sampled shortest paths increases from 500 to 16,000. It can be seen from Fig. 1 that both the average and maximum relative errors β decrease approximately by half, when the number L of sampled paths increases by twice. Fig. 1 also shows that the average relative error β with $K=100$ is larger than that with $K=50$ in all the eight networks. The rationale behind this is that the found top-100 group C_{100} covers more paths than the found top-50 group C_{50} in the total number L of sampled paths, and the estimated centrality $\hat{B}_L(C_{100})$ thus is more biased than $\hat{B}_L(C_{50})$.

C. The Performance of Different Algorithms by Varying the Number K of Nodes

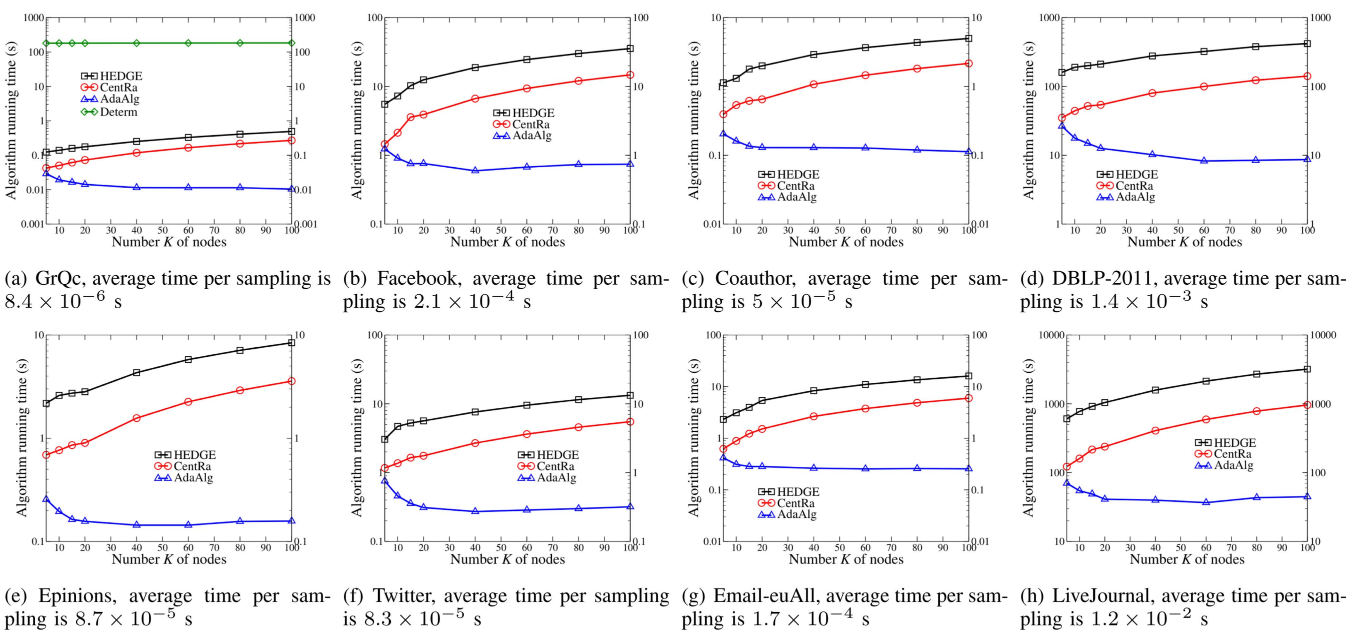
We then evaluate the running times of different algorithms, by increasing the number K of nodes from 20 to 100, when the error ratio ϵ is set at 0.3 and the error probability γ is set at 1%. Notice that the algorithms are run one by one and only a single thread is used, though there are 16 threads in the CPU of our server. Fig. 2 shows that both the running times of algorithms `HEDGE` and `CentRa` increase with K, as the both algorithms need to ensure that the maximum deviation of the estimated



(a) GrQc (b) Facebook (c) Coauthor (d) DBLP-2011

(e) Epinions (f) Twitter (g) Email-euAll (h) LiveJournal

Fig. 1. The average and maximum relative error β between the *biased* and *unbiased* estimated centralities $\hat{B}_L(C)$ and $\overline{B_L(C)}$ in 100 simulations, by varying the number L of sampled shortest paths from 500 to 16,000, where $\beta = 1 - \frac{\overline{B_L(C)}}{\hat{B}_L(C)}$.

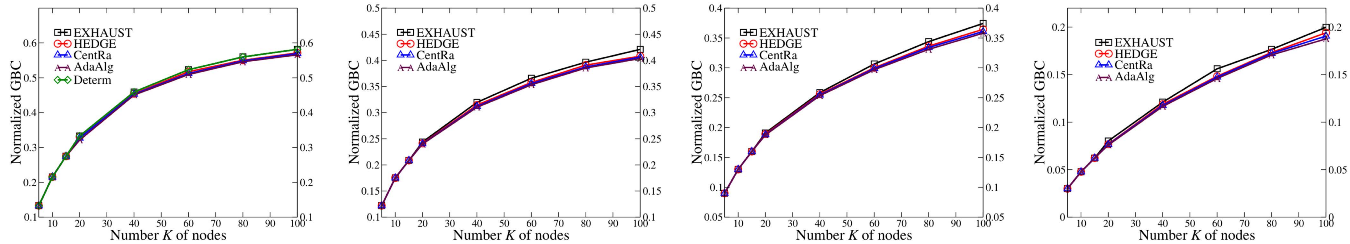


(a) GrQc, average time per sampling is 8.4×10^{-6} s (b) Facebook, average time per sampling is 2.1×10^{-4} s (c) Coauthor, average time per sampling is 5×10^{-5} s (d) DBLP-2011, average time per sampling is 1.4×10^{-3} s

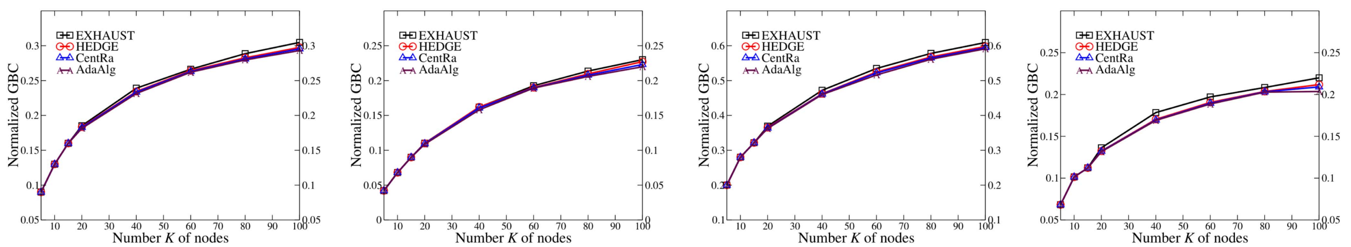
(e) Epinions, average time per sampling is 8.7×10^{-5} s (f) Twitter, average time per sampling is 8.3×10^{-5} s (g) Email-euAll, average time per sampling is 1.7×10^{-4} s (h) LiveJournal, average time per sampling is 1.2×10^{-2} s

Fig. 2. The running times of different algorithms, by increasing the number K of nodes from 5 to 100, when the error ratio ϵ is 0.3 and the error probability γ is 1%.

centrality of every group from its expectation is no greater than a small given threshold $\frac{\epsilon}{2}opt$ for all the groups with no more than K nodes, and the number n^K of such groups grows very quickly with the increase on the value of K, thereby sampling more paths. In contrast, the running time of algorithm `AdaAlg` first decreases, and perhaps slightly increases in some networks with the increase on K. The rationale behind the phenomenon is as follows. On one hand, since the optimal value opt increases with the growth of K, then the random event $\overline{B_{L_q}(C_q)} \geq g_q$ in algorithm `AdaAlg` is more likely to happen in an earlier iteration of the algorithm, i.e., a smaller value of q, thereby sampling less shortest paths before the random event. On the other hand, since the relative error β between the biased and unbiased estimated centralities $\hat{B}_L(C)$ and $\overline{B_L(C)}$ becomes larger when the number K of nodes increases, see Fig. 1, algorithm `AdaAlg` needs to sample more shortest paths to ensure that the accumulative error ratio ϵ_{sum} is no more than the given error ratio ϵ, where β decreases with the number L of sampled paths, and ϵ_{sum} is

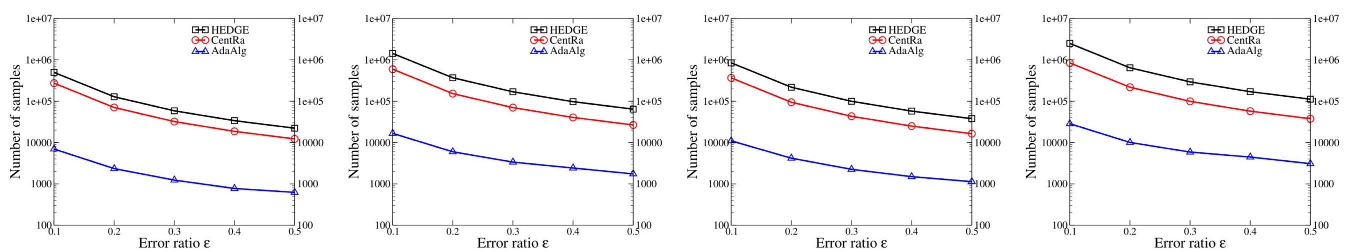


(a) GrQc (b) Facebook (c) Coauthor (d) DBLP-2011

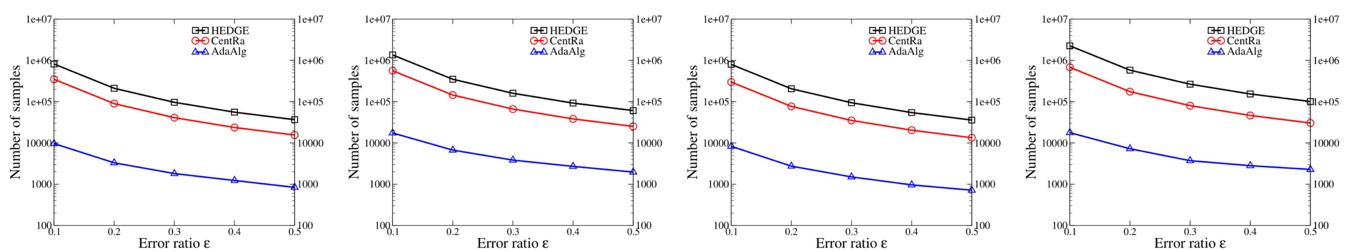


(e) Epinions (f) Twitter (g) Email-euAll (h) LiveJournal

Fig. 3. The normalized GBCs (Group Betweenness Centralities) of different algorithms, by increasing the number K of nodes from 20 to 100, when the error ratio ϵ is 0.3 and the error probability γ is 1%.



(a) GrQc, average time per sampling is 8.4×10^{-6} s (b) Facebook, average time per sampling is 2.1×10^{-4} s (c) Coauthor, average time per sampling is 5×10^{-5} s (d) DBLP-2011, average time per sampling is 1.4×10^{-3} s

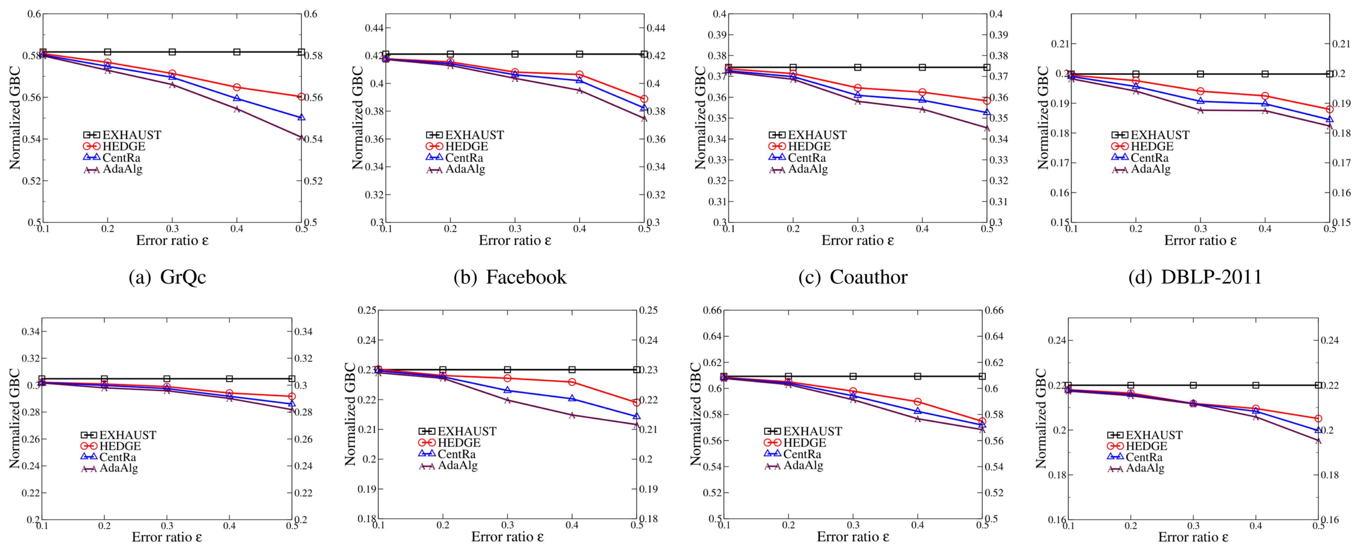


(e) Epinions, average time per sampling is 8.7×10^{-5} s (f) Twitter, average time per sampling is 8.3×10^{-5} s (g) Email-euAll, average time per sampling is 1.7×10^{-4} s (h) LiveJournal, average time per sampling is 1.2×10^{-2} s

Fig. 4. The numbers of samples used by different algorithms, by increasing the error ratio ϵ from 0.1 to 0.5, when the error probability γ is 1% and the number K of found nodes is 100.

proportional to the value of β. Fig. 2 also demonstrates that the gap between the running times of algorithms `CentRa` and `AdaAlg` becomes larger when K increases, and the number of samples used by algorithm `AdaAlg` is from 1.3 to 26 times smaller than that of the state-of-the-art algorithm `CentRa` when K increases from 5 to 100.

We also investigate the normalized GBCs (Group Betweenness Centralities) of the solutions found by different algorithms, by increasing the number K of nodes from 20 to 100, when the error ratio ϵ is set at 0.3 and the error probability γ is set at 1%. Fig. 3 shows that the normalized GBC by each of the four mentioned algorithms `EXHAUST`, `HEDGE`, `CentRa`, and `AdaAlg` increases with the growth of the value of K, as more shortest paths will pass through the nodes in a group when the group size increases. In addition, the normalized GBCs of the three algorithms `HEDGE`, `CentRa`, and `AdaAlg` are very close to that of algorithm `EXHAUST`. Although the normalized GBC by algorithm `AdaAlg` is the smallest among the comparison algorithms, its value is at least 92% of that by algorithm `EXHAUST`. Denote by ϵ_e the empirical error ratio of algorithm `AdaAlg`. That is, the



(a) GrQc (b) Facebook (c) Coauthor (d) DBLP-2011

(e) Epinions (f) Twitter (g) Email-euAll (h) LiveJournal

Fig. 5. The normalized GBCs (Group Betweenness Centralities) of different algorithms, by increasing the error ratio ϵ from 0.1 to 0.5, when the number K of found nodes is 100 and the error probability γ is 1%.

value of the solution by the algorithm is $(1-1/e-\epsilon_e)opt$. To estimate the value of ϵ_e, notice that algorithm EXHAUST finds a $(1-1/e-0.03)$-approximate solution with high probability, the value of the solution then is at least $(1-1/e-0.03)opt$. We thus have $\frac{(1-1/e-\epsilon_e)opt}{(1-1/e-0.03)opt} \geq 92\%$ and $\epsilon_e \leq 7.9\%$, which is much smaller than its theoretical error ratio $\epsilon = 0.3$.

D. The Performance of Different Algorithms by Varying the Error Ratio ϵ

We further study the numbers of samples used by different algorithms, by increasing the error ratio ϵ from 0.1 to 0.5. Fig. 4 shows that the numbers of samples used by the three algorithms HEDGE, CentRa, and AdaAlg decrease with the increase on the error ratio ϵ, as less numbers of shortest paths need to be sampled for a larger error ratio. Fig. 4 plots that the number of samples used by algorithm AdaAlg is about from 12 to 36 times smaller than that of the state-of-the-art algorithm CentRa.

We finally investigate the normalized GBCs of different algorithms, by increasing the error ratio ϵ from 0.1 to 0.5, when the number K of found nodes is 100 and the error probability γ is 1%. Fig. 5 shows that the normalized GBC by each of the three algorithms HEDGE, CentRa, and AdaAlg decreases when the error ratio ϵ increases, as less numbers of shortest paths are sampled in the three algorithms, thereby resulting in weaker solutions. Fig. 5 demonstrates that the empirical ratio of the solution delivered by algorithm AdaAlg to the solution by algorithm EXHAUST is at least 98%, 96%, 92%, 90%, and 88%, respectively, in the eight networks, when the error ratio ϵ is 0.1, 0.2, 0.3, 0.4, and 0.5, respectively. Since the value of $\frac{(1-1/e-\epsilon_e)opt}{(1-1/e-0.03)opt}$ is no less than the empirical ratio, we know that the empirical error ratio ϵ_e of algorithm AdaAlg is no more than 4.5%, 5.1%, 7.5%, 8.2%, and 10%, respectively, which are much less than their theoretical error ratios 0.1, 0.2, 0.3, 0.4, and 0.5, respectively.

VII. Conclusion

Unlike existing randomized algorithms for the top-K group betweenness centrality problem that ensured that, the maximum deviation of the estimated centrality of every group with no more than K nodes from its expectation is no greater than a small given threshold, in this paper we proposed a novel algorithm to estimate the centrality of a tentative group adaptively, and the proposed algorithm immediately stops once the centrality is large enough, thereby sampling much less numbers of shortest paths. We theoretically showed that, even the proposed algorithm used much less samples, it still can find a $(1-1/e-\epsilon)$-approximate solution with high probability. Furthermore, experimental results with real-world large-scale networks showed that, the number of samples used by the proposed algorithm is up to 36 times smaller than the state-of-the-art, while the centrality of the group found by the algorithm is comparable with the baseline, e.g., no more than 4.5% smaller.

References

[1] M. Barthélemy, "Betweenness centrality in large complex networks," *Eur. Phys. J. B*, vol. 38, no. 2, pp. 163–168, 2004.

[2] M. Borassi and E. Natale, "KADABRA is an adaptive algorithm for betweenness via random approximation," *J. Exp. Algorithmics*, vol. 24, pp. 1–35, 2019.

[3] U. Brandes, "A faster algorithm for betweenness centrality," *J. Math. Sociol.*, vol. 25, no. 2, pp. 163–177, 2001.

[4] M. H. Chehreghani, A. Bifet, and T. Abdessalem, "An in-depth comparison of group betweenness centrality estimation algorithms," in *Proc. IEEE Int. Conf. Big Data*, 2018, pp. 2104–2113.

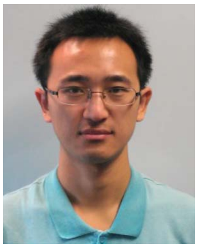
[5] M. Cheng, C. Yin, S. Nazarian, and P. Bogdan, "Deciphering the laws of social network-transcendent COVID-19 misinformation dynamics and implications for combating misinformation phenomena," *Sci. Rep.*, vol. 11, 2021, Art. no. 10424.

[6] F. Chung and L. Lu, “Concentration inequalities and martingale inequalities: A survey,” *Internet Math.*, vol. 3, no. 1, pp. 79–127, 2006.
[7] C. Cousins, C. Wohlgemuth, and M. Riondato, “Bavarian: Betweenness centrality approximation with variance-aware rademacher averages,” *ACM Trans. Knowl. Discov. Data*, vol. 17, no. 6, 2023, Art. no. 78.
[8] S. Dolev, Y. Elovici, R. Puzis, and P. Zilberman, “Incremental deployment of network monitors based on group betweenness centrality,” *Inf. Process. Lett.*, vol. 109, no. 20, pp. 1172–1176, 2009.
[9] M. Fink and J. Spoerhase, “Maximum betweenness centrality: Approximability and tractable cases,” in *Proc. Int. Workshop Algorithms Comput.*, 2011, pp. 9–20.
[10] L. C. Freeman, “A set of measures of centrality based on betweenness,” *Sociometry*, vol. 40, pp. 35–41, 1977.
[11] M. Girvan and M. E. Newman, “Community structure in social and biological networks,” *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 12, pp. 7821–7826, 2002.
[12] P. V. Gupte, B. Ravindran, and S. Parthasarathy, “Role discovery in graphs using global features: Algorithms, applications and a novel evaluation strategy,” in *Proc. IEEE 33rd Int. Conf. Data Eng.*, 2017, pp. 771–782.
[13] L. He, C. Lu, J. Ma, J. Cao, L. Shen, and P. S. Yu, “Joint community and structural hole spanner detection via harmonic modularity,” in *Proc. 22nd ACM Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 875–884.
[14] K. Huang et al., “Efficient approximation algorithms for adaptive influence maximization,” *VLDB J.*, vol. 29, pp. 1385–1406, 2020.
[15] S. Iyer, T. Killingback, B. Sundaram, and Z. Wang, “Attack robustness and centrality of complex networks,” *PLoS One*, vol. 8, no. 4, 2013, Art. no. e59613.
[16] M. Li et al., “Efficient algorithms for finding diversified top-k structural hole spanners in social networks,” *Inf. Sci.*, vol. 602, pp. 236–258, 2022.
[17] B. Liu, Z. Li, X. Chen, Y. Huang, and X. Liu, “Recognition and vulnerability analysis of key nodes in power grid based on complex network centrality,” *IEEE Trans. Circuits Syst. II Exp. Briefs*, vol. 65, no. 3, pp. 346–350, Mar. 2018.
[18] T. Lou and J. Tang, “Mining structural hole spanners through information diffusion in social networks,” in *Proc. ACM Int. Conf. World Wide Web*, 2013, pp. 825–836.
[19] A. Mahmoody, C. E. Tsourakakis, and E. Upfal, “Scalable betweenness centrality maximization via sampling,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 1765–1773.
[20] A. Maulana and M. Atzmueller, “Many-objective optimization for anomaly detection on multi-layer complex interaction networks,” *Appl. Sci.*, vol. 11, no. 9, 2021, Art. no. 4005.
[21] M. Newman, *Networks*, Oxford, U.K.: Oxford Univ. Press, 2018.
[22] M. E. J. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Phys. Rev. E*, vol. 69, 2004, Art. no. 026113.
[23] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, “An analysis of approximations for maximizing submodular set functions-I,” *Math. Prog.*, vol. 14, pp. 265–294, Dec. 1978.
[24] L. Pellegrina, “Efficient centrality maximization with rademacher averages,” in *Proc. 29th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2023, pp. 1872–1884.
[25] L. Pellegrina and F. Vandin, “SILVAN: Estimating betweenness centralities with progressive sampling and non-uniform rademacher bounds,” *ACM Trans. Knowl. Discov. Data*, vol. 18, no. 3, 2023, Art. no. 52.
[26] R. Puzis, Y. Elovici, and S. Dolev, “Fast algorithm for successive computation of group betweenness centrality,” *Phys. Rev. E*, vol. 76, no. 5, 2007, Art. no. 056709.
[27] M. Riondato and E. Upfal, “ABRA: Approximating betweenness centrality in static and dynamic graphs with Rademacher averages,” *ACM Trans. Knowl. Discov. Data*, vol. 12, no. 5, 2018, Art. no. 61.
[28] Stanford large network dataset collection, Accessed: Mar. 30, 2024. [Online]. Available: http://snap.stanford.edu/data/index.html
[29] S. Vosoughi, D. Roy, and S. Aral, “The spread of true and false news online,” *Sci.*, vol. 359, pp. 1146–1151, 2018.
[30] T. Weng, X. Zhou, Y. Fang, K. L. Tan, and K. Li, “Finding top-k important edges on bipartite graphs: Ego-betweenness centrality-based approaches,” in *Proc. IEEE 39th Int. Conf. Data Eng.*, 2023, pp. 2415–2428.
[31] WOSN 2009 data sets, Accessed: Mar. 30, 2024. [Online]. Available: https://socialnetworks.mpi-sws.org/data-wosn2009.html
[32] W. Xu, T. Li, W. Liang, J. Xu Yu, N. Yang, and S. Gao, “Identifying structural hole spanners to maximally block information propagation,” *Inf. Sci.*, vol. 505, pp. 100–126, 2019.
[33] W. Xu et al., “An adaptive sampling algorithm for the top-K group betweenness centrality,” in *Proc. 41st IEEE Int. Conf. Data Eng.*, 2025, pp. 170–182.
[34] W. Xu, M. Rezvani, W. Liang, J. X. Yu, and C. Liu, “Efficient algorithms for the identification of top-k structural hole spanners in large social networks,” *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 5, pp. 1017–1030, May 2017.
[35] Y. Yoshida, “Almost linear-time algorithms for adaptive betweenness centrality using hypergraph sketches,” in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2014, pp. 1416–1425.
[36] Q. Zhang, R. H. Li, M. Pan, Y. Dai, G. Wang, and Y. Yuan, “Efficient top-k ego-betweenness search,” in *Proc. IEEE 38th Int. Conf. Data Eng.*, 2022, pp. 380–392.

Wenzheng Xu (Senior Member, IEEE) received the BSc, ME, and PhD degrees in computer science from Sun Yat-Sen University, Guangzhou, China, in 2008, 2010, and 2015, respectively. He currently is a full professor with Sichuan University. He was a visitor with both the Australian National University and the Chinese University of Hong Kong. He has authored or coauthored more than 100 papers on prestigious journals and conferences, such as *IEEE Transactions on Knowledge and Data Engineering*, *IEEE/ACM Transactions on Networking*, *IEEE Transactions on Computers*, *IEEE Transactions on Mobile Computing*, *IEEE Transactions on Parallel and Distributed Systems*, ICDE, INFOCOM, CIKM. His research interests include online social networks, Internet of Things, UAV networks, mobile computing, approximation algorithms, combinatorial optimization, and graph theory. He is a senior member of the IEEE.

Jing Li received BSc (first class Honours) and PhD degrees in computer science from The Australian National University, in 2018 and 2022, respectively. He is currently a postdoctoral fellow with the City University of Hong Kong. His research interests include combinatorial optimization, edge computing, Internet of Things, digital twin, and network function virtualization.

Weifa Liang (Fellow, IEEE) received the BSc degree in computer science from Wuhan University, China, in 1984, the ME degree in computer science from the University of Science and Technology of China in 1989, and the PhD degree in computer science from the Australian National University, in 1998. He currently is a professor with the Department of Computer Science, City University of Hong Kong. Prior to the current position, he was a professor with the Australian National University. His research interests include design and analysis of energy efficient routing protocols for wireless ad hoc and sensor networks, Internet of Things, edge and cloud computing, network function virtualization and software-defined networking, design and analysis of parallel and distributed algorithms, approximation algorithms, combinatorial optimization, and graph theory. He was an associate editor for *IEEE Transactions on Communications*.



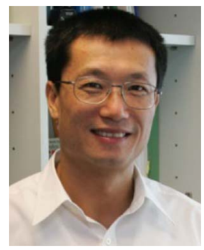
Zichuan Xu (Member, IEEE) received the ME and BSc degrees in computer science from the Dalian University of Technology, China, in 2011 and 2008, respectively, and the PhD degree in computer science from the Australian National University, in 2016. He was a research associate with the University College London. He is currently a professor with the School of Software, Dalian University of Technology. His research interests include cloud computing, software-defined networking, wireless sensor networks, algorithmic game theory, and optimization problems.

Jian Peng received the BA and PhD degrees from the University of Electronic Science and Technology of China (UESTC), in 1992 and 2004, respectively. He is currently a professor with the College of Computer Science, Sichuan University. His recent research interests include social networks, wireless sensor networks, Big Data, cloud computing, and combinatorial optimization.

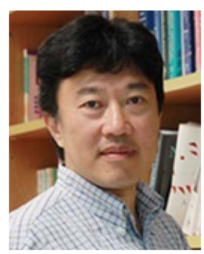
Pan Zhou (Member, IEEE) received the BS degree with the Advanced Class of Huazhong University of Science and Technology (HUST), Wuhan, China, in 2006, and the PhD degree from the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA, in 2011. He is currently a professor with the School of Electronic Information and Communications, HUST, Wuhan, China. He was a senior technical member with Oracle, Inc., American, from 2011 to 2013, Boston, MA, USA. His current research interests include security and privacy, machine learning and Big Data analytic, social networks, and information networks.

Cheng Zhao received the the master's degree from Shanghai Jiao Tong University. He is currently with Sichuan Province Commercial Investment Group Company, Ltd., Chengdu, China. His research interests include optimization, randomized algorithms, and UAV networks.

Binyu Yan is currently an associate professor with the College of Computer Science, Sichuan University, Chengdu, China. His research interests include combinatorial optimization, computer vision and software engineering.



Xiaohua Jia (Fellow, IEEE) received the BSc and MEng degrees from the University of Science and Technology of China, in 1984 and 1987, respectively, and the DSc degree in information science from the University of Tokyo, in 1991. He is currently a chair professor with the Department of Computer Science, City University of Hong Kong. His research interests include cloud computing and distributed systems, social networks, computer networks, wireless sensor networks and mobile wireless networks. He is also an editor of *IEEE Transactions on Parallel and Distributed Systems* from 2006 to 2009, *Journal of World Wide Web*, *Wireless Networks*, *Journal of Combinatorial Optimization*. He is also the general chair of ACM MobiHoc 2008, TPC co-chair of IEEE MASS 2009, area-chair of IEEE INFOCOM 2010, TPC co-chair of IEEE GlobeCom 2010, Ad Hoc and Sensor Networking Symposium, and Panel co-chair of IEEE INFOCOM 2011.



Jeffrey Xu Yu (Senior_Member, IEEE) received the BE, ME, and PhD degrees in computer science, from the University of Tsukuba, Tsukuba, Japan, in 1985, 1987, and 1990, respectively. He is currently a professor of data science and analytics thrust, the Hong Kong University of Science and Technology (Guangzhou). He was a professor with the Chinese University of Hong Kong, Hong Kong. His main research interests include graph algorithms and systems, graph neural networks, vector databases, query processing and optimization. He was more than 300 organization committees and program committees in international conferences/workshops including the PC Co-chair of APWeb'04, WAIM'06, APWeb/WAIM'07, WISE'09, PAKDD'10, DASFAA'11, ICDM'12, NDBC'13, ADMA'14, CIKM'15, Bigcomp'17, DSAA'19 and CIKM'19, and conference general co-chair of APWeb'13 and ICDM'18. He was an information director and a member in ACM SIGMOD executive committee from 2007 to 2011, an associate editor of *IEEE Transactions on Knowledge and Data Engineering* from 2004 to 2008, an associate editor in *VLDB Journal* from 2007to 2013, and the chair of the steering committee in Asia Pacific Web Conference from 2013to 2016. He is an associate editor in ACM Transactions on Database Systems, *WWW Journal*, the *International Journal of Cooperative Information Systems*, the *Journal of Information Processing*, and Journal on *Health Information Science and Systems*. He is a member of ACM, a senior member of IEEE, and a member of IEEE Computer Society.