An Adaptive Sampling Algorithm for the Top-K Group Betweenness Centrality

Wenzheng Xu[a,b], Honglin Mao[a], Heng Shao[a], Weifa Liang[c], Jian Peng[a], Wen Huang[a], Zichuan Xu[d], Pan Zhou[e], Jeffrey Xu Yu[f]
[a]College of Computer Science, Sichuan University, Chengdu, 610065, P. R. China
[b]National Key Laboratory of Fundamental Algorithms and Models for Engineering Simulation, Sichuan University
[c]Department of Computer Science, City University of Hong Kong, Hong Kong
[d]School of Software, Dalian University of Technology, Dalian, P. R. China
[e]School of Cyber Science and Engineering, Huazhong University of Science & Technology, Wuhan, P. R. China
[f]Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong
Corresponding authors: Jian Peng and Wen Huang
Email: wenzheng.xu3@gmail.com, 2023223040080@stu.scu.edu.cn, 2022223040029@stu.scu.edu.cn, weifa.liang@cityu.edu.hk
jianpeng@scu.edu.cn, wen@scu.edu.cn, z.xu@dlut.edu.cn, panzhou@hust.edu.cn, yu@se.cuhk.edu.hk

***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 algorithm uses much less samples, it still can find a performance-guaranteed group with a large success probability. Experimental results with real-world networks demonstrate that the number of samples used by the proposed algorithm is from 2 to 18 times smaller than the state-of-the-art, while the centrality of the group found by the algorithm is no more than 4% smaller than the latter.**

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

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 [11], [23], [37]. The concept of BC has various applications, including close community detection in social and biological networks [12], [20], [24], [32], vulnerability detection in communication networks or power grid networks [16], [18], [21], misinformation (e.g., rumors) filtering in social networks [5], [13], [14], [17], [31], [34], [35], etc.

In this paper, we study a top-K Group Betweenness Centrality (GBC) problem [28], 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 [10], and the best approximation algorithm for it so far can find a $(1-1/e)$-approximate solution by Puzis *et al.* [28] 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 algorithm [20], [26]. 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 a large success 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 [20], [26] have conducted pioneering researches for the top-K GBC problem, the running times of the algorithms [20], [26] 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 a large success probability, the studies in [20], [26] 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

2375-026X/25/$31.00 ©2025 IEEE
DOI 10.1109/ICDE65448.2025.00020

is the optimal value of the problem. It can be seen that the studies in [20], [26] 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 [20], [26], 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.

- 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.
- We theoretically show that, even the proposed algorithm uses much less samples, it still can find a $(1-1/e-\epsilon)$-approximate group with a large success probability, where ϵ is a given error ratio with $0 < \epsilon < 1-1/e$.
- 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 from 2 to 18 times smaller than the state-of-the-art [26], while the betweenness centrality of its found group is comparable with the latter, e.g., no more than 4% 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 a large success 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 [10], and some pioneering studies have been taken in the past decades. For example, Puzis *et al.* [28] 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 [28]. Fink *et al.* [10] 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 [28] are prohibitively high for large networks, and researchers studied non-trivial trades-off between the quality of found solutions and algorithms' running time [20], [26], [36], by proposing $(1-1/e-\epsilon)$-randomized algorithms with a large success probability $1-\gamma$, where ϵ is given error ratio with $0 < \epsilon < 1-1/e$, and γ is given error probability. Yoshida [36] 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 [36] 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 $0 < \mu_{opt} \leq 1$. However, Mahmoody *et al.* [20] 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 [20], [26] 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 [20] is $L_2 = O(\frac{\log\frac{1}{\gamma}+K\log n}{\epsilon^2\mu_{opt}})$, while Pellegrina [26] 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.

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.* [29] proposed a rigorous sampling algorithm by the theory of Rademacher averages and pseudodimension. Cousins *et al.* [7] addressed the limitations of the algorithm in [29] by using the Monte Carlo empirical Rademacher averages and variance-aware tail bounds. Pellegrina *et al.* [27] adopted non-uniform bounds for different subsets of nodes. Borassi *et al.* [2] improved the algorithm in [29] by assigning different confidences on the estimated centralities of different nodes.

III. Preliminaries

In this section, we first introduce the network model and notion of group betweenness centrality, then formally define the top-K group betweenness centrality problem, and briefly introduce the state-of-the-art algorithm for the problem.

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], [20], [26], [28], [36], 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 $B(C) = O(n^2)$ [20] 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.,

$$\max_{C \subseteq V, |C| = K} \{ B(C) \}. \quad (3)$$

The problem is NP-hard [28], 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 Top-K GBC Problem

We briefly introduce the state-of-the-art algorithm [26] 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 [26], [27] 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], [27], 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 [25]. 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

$$\hat{B}_L(C) = \frac{L'}{L} n(n-1). \quad (4)$$

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 [26] 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 a large success 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 a large success probability, as $\hat{B}_L(C) \geq (1-1/e)\overline{B_L(C^*)}$ [25], [26].

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 [26], 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 Eq. (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-C. Assume that

$$\frac{n(n-1)}{b^{Q^*}} \;\geq\; opt \;\geq\; \frac{n(n-1)}{b^{Q^*+1}}, \tag{5}$$

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

Different from the state-of-the-art in [26] 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 samples more shortest paths 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

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

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 $\theta = (\ln\frac{2}{\gamma} + \ln Q_{max})\frac{2+\alpha}{\alpha^2}$, which are two constants.

We then randomly sample $L_q - |\mathcal{S}|$ shortest paths by invoking the algorithm in [26], and add the $L_q - |\mathcal{S}|$ shortest paths to $\mathcal{S}$, where

$$L_q \;=\; \theta\frac{n(n-1)}{g_q} \;=\; \theta b^q. \tag{7}$$

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 [26] 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 Eq. (4).

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

$$\overline{B_L(C)} = \frac{L'_q}{L} n(n-1). \tag{8}$$

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 Centrality of the Tentative Group is Large Enough

When $1 \leq q \leq Q^* - 1$, it can be seen that the guess $g_q = \frac{n(n-1)}{b^q} \geq b \cdot opt$ by Ineq. (5). 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 $\overline{B_{L_q}(C_q)} \geq g_q$ is very small, since the guess g_q is too large, i.e., no less than $b \cdot opt$. On the other hand, if the random event $\overline{B_{L_q}(C_q)} \geq g_q$ occurs, we know that $q \geq Q^*$ with a large success 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 a large success probability, as the value of g_q decreases exponentially when the value of q increases, see the definition of g_q in Eq. (6).

The value of cnt is very important to estimate whether the group betweenness centrality $B(C_q)$ of C_q is no less than $(1-1/e-\epsilon)opt$, where opt is the optimal value. Specifically, the value of cnt serves as the following three purposes.

(i) If $cnt \leq 1$, this indicates that the number L_q of sampled shortest paths is not enough, as the guess g_q is no less than opt (i.e., $g_q \geq opt$) with a large probability. We then need to generate more samples in the next iteration.

(ii) If $cnt \geq 2$, then $q \geq Q^* + 1$ with a large probability. Following Ineq. (5), we know that the guess $g_q \leq opt$.

(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

$$g_q \leq \frac{opt}{b^{cnt-2}},\ \forall\ cnt\ \geq 2, \tag{9}$$

and the number L_q of chosen paths now is $\theta\frac{n(n-1)}{g_q} \geq \theta\frac{n(n-1)}{opt}b^{cnt-2}$ by Eq. (7).

When $cnt \geq 2$, we estimate whether the centrality $B(C_q)$ is no less than $(1-1/e-\epsilon)opt$ as follows. We start by calculate the smallest error ratio ϵ_1 so that the probability that $B(C_q) \geq \overline{B_{L_q}(C_q)} - \epsilon_1 opt$ is no less than $1-\frac{\gamma}{4}$. In Lemma 4 of Section V-C, we show that

$$\epsilon_1 = \frac{\frac{2c_1}{3} + \sqrt{\frac{4c_1^2}{9} + 8c_1}}{2}, \tag{10}$$

where $c_1 = \frac{\ln\frac{4}{\gamma}}{\theta b^{cnt-2}}$. 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 $\overline{B_{L_q}(C^*)} \geq B(C^*) - \epsilon_1 opt$ 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 a large success probability.

$$\begin{aligned}
& B(C_q) \\
\geq\ & \overline{B_{L_q}(C_q)} - \epsilon_1 opt \\
=\ & (1-\beta)\hat{B}_{L_q}(C_q) - \epsilon_1 opt \\
=\ & (1-\beta)(1-1/e)\overline{B_{L_q}(C^*)} - \epsilon_1 opt, \\
& \text{as } \hat{B}_{L_q}(C_q) \geq (1-1/e)\overline{B_{L_q}(C^*)} \text{ [25], [26]} \\
\geq\ & (1-\beta)(1-1/e)(B(C^*) - \epsilon_1 opt) - \epsilon_1 opt \\
=\ & (1-1/e-\epsilon_{sum})opt,\ \text{as } B(C^*) = opt,
\end{aligned} \tag{11}$$

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

Remark: Let $\epsilon = \beta_{max}(1-1/e)(1-\epsilon_1) + (2-1/e)\epsilon_1$. After re-writing the equation, we have $\beta_{max} = 1 - \frac{1-1/e-\epsilon+\epsilon_1}{(1-1/e)(1-\epsilon_1)}$. Since ϵ_1 decreases with the growth of cnt when $cnt \geq 2$, ϵ_1 decreases with the number q of iterations performed in the algorithm if $q \geq Q^* + 1$. Then, the value of β_{max} increases with q, which indicates that the algorithm allows a larger maximum relative error β_{max} between the *biased* estimation $\hat{B}_{L_q}(C_q)$ and the *unbiased* estimation $\overline{B_{L_q}(C_q)}$ of $B(C_q)$.

On the other hand, the real relative error β between $\hat{B}_{L_q}(C_q)$ and $\overline{B_{L_q}(C_q)}$ becomes smaller with the increase on the number L_q of chosen paths. In fact, our later experimental results confirm that the relative error β decreases exponentially with the increase of L_q. Then, β becomes smaller with the increase of q. Therefore, the algorithm is very likely to terminate, i.e., the probability that $\beta \leq \beta_{max}$ increases with the growth of q.

C. The Choice of the Base b

Recall that the number of sampled shortest paths at the qth iteration is $L_q = \theta b^q = (\ln\frac{2}{\gamma} + \ln Q_{max})\frac{2+\alpha}{\alpha^2}b^q = c_2(\ln\frac{2}{\gamma} + \ln Q_{max})b^q$, where $b > 1$, $c_2 = \frac{2+\alpha}{\alpha^2}$ and $Q_{max} = \lceil \log_b n(n-1) \rceil$.

On one hand, it can be seen that the term b^q increases more slower if the base b is smaller with $b > 1$, where $L_q = c_2(\ln\frac{2}{\gamma} + \ln Q_{max})b^q$, 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, if the base b is too small, the number Q_{max} of iterations will increase as $Q_{max} = \lceil \log_b n(n-1) \rceil$. Then, the number of samples in each iteration will increase, as the term $(\ln\frac{2}{\gamma} + \ln Q_{max})$ in L_q increases. In addition, 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., $g_q \geq b \cdot opt$), the value of b should not be too small; otherwise (b is too small), the number L_q of samples should be very large to ensure that the random event $\overline{B_{L_q}(C_q)} \geq g_q$ occurs with a small probability when $g_q \geq b \cdot opt$.

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 $g_q \geq b \cdot opt$, where

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

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

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

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

For example, assume that the error ratio ϵ is 0.5. Then, $\alpha = \frac{\epsilon}{2-1/e} = 0.3063$, and $c_2 = \frac{2+\alpha}{\alpha^2} = 24.57$. Following Eq. (12), $b' = 1.35$. Then, $b = \max\{b',\ b_{min}\} = 1.35$.

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

(ii) The probability that $B(C_q) \geq \overline{B_{L_q}(C_q)} - \epsilon_1 opt$ is no less than $1-\frac{\gamma}{4}$, when $L_q \geq \theta\frac{n(n-1)}{opt}2^{cnt-2}$ and $cnt \geq 2$, where ϵ_1 was defined in Eq. (10), see Section V-C.

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 probability $1 - \gamma$

Output: A group C with K nodes

Let $\alpha \leftarrow \frac{\epsilon}{2-1/e}$;
Calculate the base b by Eq. (13);
Let $Q_{max} \leftarrow \lceil \log_b n(n-1) \rceil$; /* number of iterations */
Let $\theta \leftarrow (\ln \frac{2}{\gamma} + \ln Q_{max}) \frac{2+\alpha}{\alpha^2}$; /* 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 in the algorithm */
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 [26].
  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;
  if $\overline{B_{L_q}(C_q)} \geq g_q$ **then**
    $cnt \leftarrow cnt + 1$;
  else
    /* need more samples in the next iteration, as the guess g_q is at least $b \cdot opt$ with $b > 1$ */
  end if
  if $cnt \geq 2$ **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 $B(C_q) \geq \overline{B_{L_q}(C_q)} - \epsilon_1 opt$ is no less than $1 - \frac{\gamma}{4}$ by Eq. (10);
    Let $\beta \leftarrow 1 - \frac{\overline{B_{L_q}(C_q)}}{\hat{B}_{L_q}(C_q)}$; /* the relative error */
    /* The probability that $\overline{B_{L_q}(C*)} \geq B(C^*) - \epsilon_1 opt$ is no less than $1 - \frac{\gamma}{4}$; */
    Let $\epsilon_{sum} \leftarrow \beta(1-1/e)(1-\epsilon_1) + (2-1/e)\epsilon_1$;
    if $\epsilon_{sum} \leq \epsilon$ **then**
      return group C_q and its estimated centrality $\overline{B_{L_q}(C_q)}$.
    else
      /* the algorithm needs to sample more shortest paths in the next iteration, as the accumulative error ratio ϵ_{sum} is larger than ϵ */
    end if
  end if
end for

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

The three claims indicate that C_q is a $(1 - 1/e - \epsilon)$-approximate solution with a large success probability, if $cnt \geq 2$ and $\epsilon_{sum} \leq \epsilon$, see Lemma 6 in Section V-E.

We finally analyze the worst-case expected time complexity of `Algorithm 1` in Section V-F.

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,

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

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

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

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

Proof: Let $\mu = \frac{B(C)}{n(n-1)}$. Let random variable $Y_l = 1$, if the lth shortest path is covered by group C; otherwise, $Y_l = 0$, where $1 \leq l \leq L$. Then, the expectation and variance of Y_l are $\mathbf{E}[Y_l] = \mu$ and $\mathbf{Var}[Y_l] = \mu(1-\mu)$, respectively.

Let $X_l = \sum_{j=1}^{l} (Y_j - \mu)$ with $1 \leq l \leq L$. Then, the expectation of X_l is $\mathbf{E}[X_l] = \sum_{j=1}^{l} (\mathbf{E}[Y_j] - \mu) = 0$.

Notice that the sampling process of the lth shortest path is independent with the previous $(l-1)$ shortest paths, though the decision of sampling the lth shortest path depends on the previous $(l-1)$ shortest paths. Since the conditional expectation of $Y_l - \mu$ given $Y_1, Y_2, \ldots, Y_{l-1}$ is

$\mathbf{E}[Y_l - \mu \mid Y_1, Y_2, \ldots, Y_{l-1}] = \mu - \mu = 0$, the conditional expectation of X_l given $X_1, X_2, \ldots, X_{l-1}$ then is

$$\begin{aligned} & \mathbf{E}[X_l \mid X_1, X_2, \ldots, X_{l-1}] \\ = & \sum_{j=1}^{l-1}(Y_j - \mu) + \mathbf{E}[(Y_l - \mu) \mid X_1, X_2, \ldots, X_{l-1}] \\ = & X_{l-1} + \mathbf{E}[(Y_l - \mu) \mid Y_1, Y_2, \ldots, Y_{l-1}] \\ = & X_{l-1}. \end{aligned} \tag{17}$$

Therefore, the random variables $X_1, X_2, \ldots, X_L$ form a martingale. In addition, the conditional variance of X_l given $X_1, X_2, \ldots, X_{l-1}$ is

$$\begin{aligned} & \mathbf{Var}[X_l \mid X_1, X_2, \ldots, X_{l-1}] \\ = & \mathbf{E}[(X_l - \mathbf{E}[X_l])^2 \mid X_1, X_2, \ldots, X_{l-1}] \\ = & \mathbf{E}[(X_l - X_{l-1})^2 \mid X_1, X_2, \ldots, X_{l-1}], \text{ by Eq. (17)} \\ = & \mathbf{E}[(Y_l - \mu)^2 \mid X_1, X_2, \ldots, X_{l-1}] \\ = & \mathbf{E}[(Y_l - \mu)^2 \mid Y_1, Y_2, \ldots, Y_{l-1}] \\ = & \mathbf{E}[(Y_l - \mu)^2] = \mathbf{Var}[Y_l] = \mu(1-\mu). \end{aligned} \tag{18}$$

Then, $\xi = \mathbf{Var}[X_1] + \sum_{l=1}^{L} \mathbf{Var}[X_l \mid X_1, X_2, \ldots, X_{l-1}] = \mathbf{Var}[Y_1] + \sum_{l=1}^{L} \mathbf{Var}[Y_l] = \mu(1-\mu) + (L-1)\mu(1-\mu) = L\mu(1-\mu)$.

Since $|X_1| \leq 1 = M$ and $|X_l - X_{l-1}| \leq 1 = M$, we now prove Ineq. (15) as follows.

$$\begin{aligned} & Pr[\overline{B_L(C)} - B(C) \geq \lambda B(C)] \\ = & Pr[L\frac{\overline{B_L(C)}}{n(n-1)} - L\frac{B(C)}{n(n-1)} \geq \lambda L\frac{B(C)}{n(n-1)}] \\ = & Pr[\sum_{l=1}^{L} Y_l - L\mu \geq \lambda L\mu] \\ = & Pr[X_L \geq \lambda L\mu] \\ = & Pr[X_L - \mathbf{E}[X_L] \geq \lambda L\mu], \text{ as } \mathbf{E}[X_L] = 0 \\ \leq & exp(-\frac{(\lambda L\mu)^2}{2\xi + \frac{2}{3}M\lambda L\mu}), \text{ by Ineq. (14)} \\ = & exp(-\frac{(\lambda L\mu)^2}{2\xi + \frac{2}{3}\lambda L\mu}), \text{ as } M = 1 \\ = & exp(-\frac{(\lambda L\mu)^2}{2L\mu(1-\mu) + \frac{2}{3}\lambda L\mu}), \text{ as } \xi = L\mu(1-\mu) \\ = & exp(-L\frac{\lambda^2\mu}{2(1-\mu) + \frac{2}{3}\lambda}) \\ \leq & exp(-L\frac{\lambda^2}{2 + \frac{2}{3}\lambda}\mu), \text{ as } 0 \leq 1-\mu \leq 1 \\ = & exp(-L\frac{\lambda^2 B(C)}{(2 + \frac{2}{3}\lambda)n(n-1)}). \end{aligned} \tag{19}$$

On the other hand, random variables $-X_1, -X_2, \ldots, -X_L$ also form a martingale, and Ineq. (16) can be shown similarly, omitted. ■

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 two, i.e., $cnt \geq 2$. 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 $1 - \frac{\gamma}{2}$, i.e., $Pr[L_q \geq \theta\frac{n(n-1)}{opt}b^{cnt-2}] \geq 1 - \frac{\gamma}{2}$.

Proof: We show that the probability $q \geq Q^* + cnt - 1$ is no less than $1 - \frac{\gamma}{2}$, where $\frac{n(n-1)}{b^{Q^*}} \geq opt \geq \frac{n(n-1)}{b^{Q^*+1}}$ by Ineq. (5). Then,

$$\begin{aligned} g_q = & \frac{n(n-1)}{b^q}, \text{ by Eq. (6)} \\ \leq & \frac{n(n-1)}{b^{Q^*+cnt-1}}, \text{ as } q \geq Q^* + cnt - 1 \\ = & \frac{n(n-1)}{b^{(Q^*+1)+cnt-2}} \\ = & \frac{n(n-1)}{b^{(Q^*+1)}}\frac{1}{b^{cnt-2}} \\ \leq & \frac{opt}{b^{cnt-2}}, \text{ as } \frac{n(n-1)}{b^{Q^*+1}} \leq opt. \end{aligned} \tag{20}$$

Following the definition of L_q in Eq. (7), we have $L_q = \theta\frac{n(n-1)}{g_q} \geq \theta\frac{n(n-1)}{opt}b^{cnt-2}$.

To prove $Pr[q \geq Q^* + cnt - 1] \geq 1 - \frac{\gamma}{2}$, we show that $Pr[q < Q^* + cnt - 1] \leq \frac{\gamma}{2}$ 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.

We show that the probability that the random event $\overline{B_{L_q}(C_q)} \geq g_q$ occurs at a given qth iteration with $1 \leq q \leq Q^* - 1$ is no larger than $\frac{\gamma}{2Q_{max}}$. Then, by the union bound, the random event $\overline{B_{L_q}(C_q)} \geq g_q$ occurs in the first $Q^* - 1$ iterations in no more than $\frac{(Q^*-1)\gamma}{2Q_{max}} \leq \frac{\gamma}{2}$, as $Q^* - 1 \leq Q_{max}$.

Given the qth iteration, we now prove that $Pr[\overline{B_{L_q}(C_q)} \geq g_q] \leq \frac{\gamma}{2Q_{max}}$. Recall that $g_q = \frac{n(n-1)}{b^q}$ and $L_q = \theta\frac{n(n-1)}{g_q} = \theta b^q = (\ln\frac{2}{\gamma} + \ln Q_{max})\frac{2+\alpha}{\alpha^2}b^q$, where $\theta = (\ln\frac{2}{\gamma} + \ln Q_{max})\frac{2+\alpha}{\alpha^2}$, and $\alpha = \frac{\epsilon}{2-1/e}$.

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

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 $b \cdot opt \geq b \cdot B(C_q)$, then it is unlikely that the estimated centrality $\overline{B_{L_q}(C_q)}$ of $B(C_q)$ is larger than $b \cdot B(C_q)$, if the number L_q of samples is sufficiently large, where $b > 1$. Specifically, we have

$$\begin{aligned} & Pr[\overline{B_{L_q}(C_q)} \geq g_q] \\ = & Pr[\overline{B_{L_q}(C_q)} - B(C_q) \geq g_q - B(C_q)] \\ \leq & Pr[\overline{B_{L_q}(C_q)} - B(C_q) \geq g_q - opt], \\ & \text{as } B(C_q) \leq opt \text{ and } g_q - B(C_q) \geq g_q - opt \\ = & Pr[\overline{B_{L_q}(C_q)} - B(C_q) \geq \frac{g_q - opt}{B(C_q)}B(C_q)] \end{aligned}$$

$$
\begin{aligned}
&\leq exp(-L_q\frac{\lambda^2 B(C_q)}{(2+\frac{2}{3}\lambda)n(n-1)}) \\
&\quad \text{by Ineq. (15) with } \lambda = \frac{g_q - opt}{B(C_q)} \\
&= exp(-L_q\frac{(\frac{g_q-opt}{B(C_q)})^2 B(C_q)}{(2+\frac{2}{3}\frac{g_q-opt}{B(C_q)})n(n-1)}) \\
&= exp(-L_q\frac{(g_q-opt)^2}{(2B(C_q)+\frac{2}{3}(g_q-opt))n(n-1)}) \\
&\leq exp(-L_q\frac{(g_q-opt)^2}{(\frac{2}{3}g_q+\frac{4}{3}opt)n(n-1)}), \text{ as } B(C_q)\leq opt \\
&= exp(-L_q(\frac{3}{2}-\frac{9opt}{2g_q+4opt})\frac{g_q-opt}{n(n-1)}) \\
&\leq exp(-L_q(\frac{3}{2}-\frac{9}{2b+4})\frac{g_q-opt}{n(n-1)}), \text{ as } g_q \geq b\cdot opt \\
&\leq exp(-\theta b^q(\frac{3}{2}-\frac{9}{2b+4})\frac{\frac{n(n-1)}{b^q}-opt}{n(n-1)}), \\
&\quad \text{as } L_q=\theta b^q \text{ and } g_q = \frac{n(n-1)}{b^q} \\
&= exp(-\theta(\frac{3}{2}-\frac{9}{2b+4})(1-\frac{b^q opt}{n(n-1)})), \\
&\leq exp(-\theta(\frac{3}{2}-\frac{9}{2b+4})(1-\frac{1}{b})), \\
&\quad \text{as } opt \leq \frac{n(n-1)}{b^{Q^*}} \leq \frac{n(n-1)}{b^{q+1}} \text{ and } q \leq Q^*-1 \\
&= exp(-(\ln\frac{2}{\gamma}+\ln Q_{max})c_2(\frac{3}{2}-\frac{9}{2b+4})(1-\frac{1}{b})), \\
&\quad \text{as } \theta = (\ln\frac{2}{\gamma}+\ln Q_{max})\frac{2+\alpha}{\alpha^2} \text{ and } c_2=\frac{2+\alpha}{\alpha^2} \\
&\leq exp(-(\ln\frac{2}{\gamma}+\ln Q_{max})c_2(\frac{3}{2}-\frac{9}{2b'+4})(1-\frac{1}{b'})), \\
&\quad \text{as } b=\max\{b',\ b_{min}\}\geq b' \text{ by Eq. (13)} \\
&= exp(-(\ln\frac{2}{\gamma}+\ln Q_{max})), \text{ as } b' \text{ is a root of the} \\
&\quad \text{equation } c_2(\frac{3}{2}-\frac{9}{2x+4})(1-\frac{1}{x})=1 \text{ with} \\
&\quad b' = \frac{3c_2+2+\sqrt{18c_2+4}}{3c_2-2} \\
&= \frac{\gamma}{2Q_{max}}. \qquad (21)
\end{aligned}
$$

The lemma then follows. ■

C. The Probability that $B(C_q) \geq \overline{B_{L_q}(C_q)} - \epsilon_1 opt$

Lemma 4: When $L_q \geq \theta\frac{n(n-1)}{opt}b^{cnt-2}$ and $cnt \geq 2$, the probability that $B(C_q) \geq \overline{B_{L_q}(C_q)} - \epsilon_1 opt$ is no less than $1-\frac{\gamma}{4}$, where ϵ_1 was defined in Eq. (10).

Proof: It is sufficient to show that $Pr[\overline{B_{L_q}(C_q)} - B(C_q) \geq \epsilon_1 opt] \leq \frac{\gamma}{4}$. We have

$$
\begin{aligned}
&Pr[\overline{B_{L_q}(C_q)} - B(C) \geq \epsilon_1\cdot opt] \\
=\ &Pr[\overline{B_{L_q}(C_q)} - B(C) \geq \frac{\epsilon_1\cdot opt}{B(C)}B(C)] \\
\leq\ &exp(-L_q\frac{\lambda^2 B(C)}{(2+\frac{2}{3}\lambda)n(n-1)}), \\
&\quad \text{by Ineq. (15) with } \lambda = \frac{\epsilon_1\cdot opt}{B(C)} \\
=\ &exp(-L_q\frac{(\frac{\epsilon_1\cdot opt}{B(C)})^2 B(C)}{(2+\frac{2}{3}\frac{\epsilon_1\cdot opt}{B(C)})n(n-1)}) \\
=\ &exp(-L_q\frac{(\epsilon_1\cdot opt)^2}{(2B(C)+\frac{2}{3}\epsilon_1\cdot opt)n(n-1)}) \\
\leq\ &exp(-L_q\frac{\epsilon_1^2\cdot opt}{(2+\frac{2}{3}\epsilon_1)n(n-1)}), \text{ as } B(C)\leq opt \\
=\ &exp(-\theta b^{cnt-2}\frac{\epsilon_1^2}{2+\frac{2}{3}\epsilon_1}), \\
&\quad \text{as } L_q \geq \theta\frac{n(n-1)}{opt}b^{cnt-2} \\
=\ &exp(-\frac{\ln\frac{4}{\gamma}}{c_1}\frac{\epsilon_1^2}{2+\frac{2}{3}\epsilon_1}), \text{ as } c_1=\frac{\ln\frac{4}{\gamma}}{\theta b^{cnt-2}} \\
=\ &exp(-\ln\frac{4}{\gamma}), \text{ as } \epsilon_1 = \frac{\frac{2c_1}{3}+\sqrt{\frac{4c_1^2}{9}+8c_1}}{2} \text{ by Eq. (10)} \\
=\ &\frac{\gamma}{4}. \qquad (22)
\end{aligned}
$$

Note that ϵ_1 is a root of the quadratic equation that $\frac{x^2}{2+\frac{2}{3}x} = c_1$. The lemma then follows. ■

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

Lemma 5: When $L_q \geq \theta\frac{n(n-1)}{opt}b^{cnt-2}$ and $cnt \geq 2$, the probability that $\overline{B_{L_q}(C^*)} \geq B(C^*) - \epsilon_1 opt$ is no less than $1-\frac{\gamma}{4}$, where ϵ_2 is defined in Eq. (10).

Proof: It is sufficient to show that $Pr[\overline{B_{L_q}(C^*)} - B(C^*) \leq -\epsilon_1\cdot opt] \leq \frac{\gamma}{4}$. We have

$$
\begin{aligned}
&Pr[\overline{B_{L_q}(C^*)} - B(C^*) \leq -\epsilon_1\cdot opt] \\
=\ &Pr[\overline{B_{L_q}(C^*)} - B(C^*) \leq -\epsilon_1\cdot B(C^*)], \\
&\quad \text{as } opt = B(C^*) \\
\leq\ &exp(-L_q\frac{\epsilon_1^2 B(C^*)}{(2+\frac{2}{3}\epsilon_1)n(n-1)}), \text{ by Ineq. (16)} \\
=\ &exp(-\theta\frac{\epsilon_1^2}{2}b^{cnt-2}), \\
&\quad \text{as } L_q = \theta\frac{n(n-1)}{opt}b^{cnt-2} \text{ and } B(C^*)=opt \\
=\ &\frac{\gamma}{4}, \text{ by Ineq. (22).} \qquad (23)
\end{aligned}
$$

The lemma then follows. ■

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, $cnt \geq 2$, and $\epsilon_{sum} \leq \epsilon$. Then, C_q is a $(1-1/e-\epsilon)$-approximate solution with a probability $1-\gamma$.

Proof: Assume that $L_q \geq \theta\frac{n(n-1)}{opt}2^{cnt-2}$ and $cnt \geq 2$. Since $Pr[B(C_q) \leq \overline{B_{L_q}(C_q)} - \epsilon_1 opt] \leq \frac{\gamma}{4}$

and $Pr[\overline{B_{L_q}(C^*)} \leq B(C^*) - \epsilon_2 opt] \leq \frac{\gamma}{4}$ by Ineqs. (22) and (23), following the union bound, the probability that the two random events $B(C_q) \geq \overline{B_{L_q}(C_q)} - \epsilon_1 opt$ and $\overline{B_{L_q}(C^*)} \geq B(C^*) - \epsilon_2 opt$ happen simultaneously is no less than $1 - (\frac{\gamma}{4} + \frac{\gamma}{4}) = 1 - \frac{\gamma}{2}$

By combining Ineq. (11), the probability that $B(C_q) \geq (1 - 1/e - \epsilon_{sum})opt$ is no less than $1 - \frac{\gamma}{2}$.

On the other hand, since the probability of the assumption $L_q \geq \theta \frac{n(n-1)}{opt} 2^{cnt-2}$ is no less than $1 - \frac{\gamma}{2}$ by Lemma 3, the probability that $B(C_q) \geq (1 - 1/e - \epsilon_{sum})opt$ is at least $(1 - \frac{\gamma}{2})(1 - \frac{\gamma}{2}) > 1 - \gamma$. When $\epsilon_{sum} \leq \epsilon$, C_q is a $(1 - 1/e - \epsilon)$-approximate solution with a probability $1 - \gamma$. ■

F. 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, its worst-case expected time complexity is no more than $O(\frac{\log \frac{1}{\gamma} + K(\log K)(\log\log n)(\log \frac{1}{\mu_{opt}})}{\epsilon^2 \mu_{opt}} m^{\frac{1}{2}+o(1)})$, 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 approximation ratio and the success probability have been shown in Lemma 6. The rest is to analyze the time complexity of `Algorithm 1`.

Since the termination of `Algorithm 1` depends on the estimated centralities of found tentative groups, one may want to know whether the worst-case time complexity of `Algorithm 1` is larger than the time complexity of the state-of-the-art in [26]. We show that the worst-case expected time complexity of `Algorithm 1` is no greater than that in [26].

Following the work in [26], if the number L_q of sampled paths at qth iteration of `Algorithm 1` is $O(\frac{\log \frac{1}{\gamma} + K(\log K)(\log\log n)(\log \frac{1}{\mu_{opt}})}{\epsilon^2 \mu_{opt}})$, then it is very likely that the deviation of the estimated centrality $\hat{B}_{L_q}(C)$ of a found tentative group C_q from its expectation $B(C_q)$ is no more than a small given threshold $\frac{\epsilon}{2} opt$, e.g., $Pr[\ |\hat{B}_{L_q}(C_q) - B(C_q)| \leq \frac{\epsilon}{2} opt\ ] \geq 1 - \frac{\gamma}{2}$. In addition, for the optimal group C^* with $B(C^*) = opt$, we still have $Pr[\ |\overline{B_{L_q}(C^*)} - B(C^*)| \leq \frac{\epsilon}{2} opt\ ] \geq 1 - \frac{\gamma}{2}$. Therefore, the centrality of the group C_q is $B(C_q) \geq \hat{B}_{L_q}(C_q) - \frac{\epsilon}{2} opt \geq (1 - 1/e)\overline{B_{L_q}(C^*)} - \frac{\epsilon}{2} opt \geq (1 - 1/e)(B(C^*) - \frac{\epsilon}{2} opt) - \frac{\epsilon}{2} opt \geq (1 - 1/e - \epsilon) opt$, and the probability of the random event $C_q \geq (1 - 1/e - \epsilon) opt$ is no less than $1 - \gamma$. Therefore, the algorithm terminates at the qth iteration with high probability.

In addition, when the random event $C_q \geq (1 - 1/e - \epsilon) opt$ occurs, the accumulative error ratio ϵ_{sum} in `Algorithm 1` is no greater than ϵ, where $\epsilon_{sum} = \beta(1 - 1/e)(1 - \epsilon_1) + (2 - 1/e)\epsilon_1$. Then, the relative error β is no greater than $\frac{\epsilon_{sum} - (2 - 1/e)\epsilon_1}{(1 - 1/e)(1 - \epsilon_1)} \leq \frac{\epsilon - (2 - 1/e)\epsilon_1}{(1 - 1/e)(1 - \epsilon_1)}$, where ϵ_1 is proportional to the value of ϵ. Notice that when the value of ϵ becomes smaller, more samples L_q are needed to ensure that the relative error β is no greater than $\frac{\epsilon - (2 - 1/e)\epsilon_1}{(1 - 1/e)(1 - \epsilon_1)}$, where the value of $\frac{\epsilon - (2 - 1/e)\epsilon_1}{(1 - 1/e)(1 - \epsilon_1)}$ is smaller when ϵ becomes smaller. That is, the relative error β converges to zero with high probability when the number of samples becomes larger. ■

VI. Performance Evaluation

In this section, we evaluate the performance of the proposed algorithm, including the change of the relative error β, the betweenness centralities of found groups, and its running times.

A. Experimental Environment Settings

We adopt ten networks, see Table I, including eight real-world networks and two synthetic networks generated by the software NetworkX [22]. The first synthetic network SyntheticNetwork-BA follows the well-known Barabási-Albert (BA) model and the second network SyntheticNetwork-WS follows the Watts-Strogatz (WS) model [9]. 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 [26].

TABLE I: Eight real-world networks in the experiments

Dataset	$\|V\|$	$\|E\|$	Type
GrQc [30]	5,244	14,496	undirected
Facebook [33]	63,731	817,090	undirected
Coauthor [19]	53,442	127,968	undirected
DBLP-2011 [15]	986,324	3,353,618	undirected
Epinions [30]	75,879	508,837	directed
Twitter [19]	92,180	377,942	directed
Email-euAll [30]	265,214	420,045	directed
LiveJournal [15]	5,363,260	54,880,888	directed
SyntheticNetwork-BA [9]	100,000	800,000	undirected
SyntheticNetwork-WS [9]	100,000	800,000	undirected

To study the performance of the proposed algorithm `AdaAlg`, we compare with the following three benchmarks.

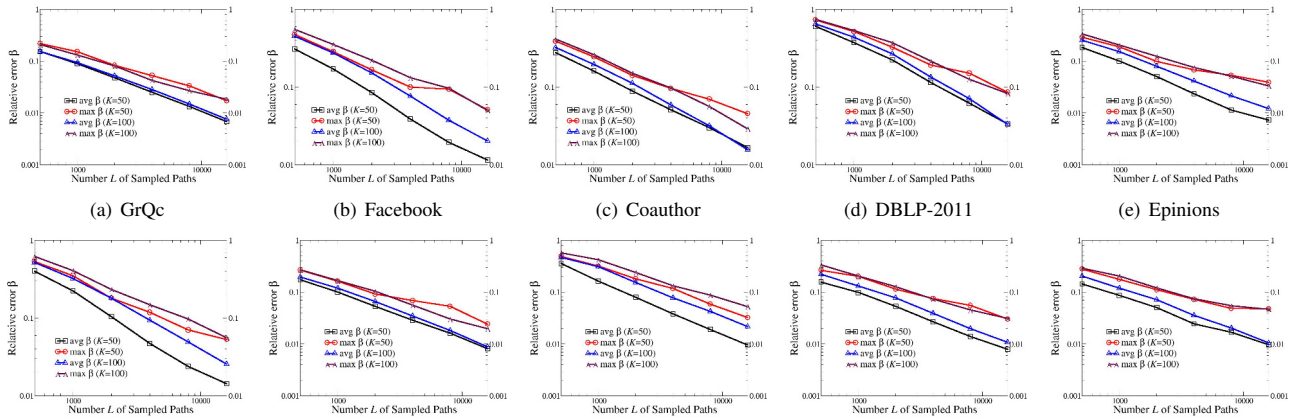
(i) Algorithm `HEDGE` [20] finds a $(1 - 1/e - \epsilon)$-approximate solution with a probability $1 - \gamma$, and the number of sampled shortest paths is $O(\frac{\log \frac{1}{\gamma} + K \log n}{\epsilon^2 \mu_{opt}})$, where μ_{opt} is the normalization of the optimal value opt with $\mu_{opt} = \frac{opt}{n(n-1)}$ and $0 < \mu_{opt} \leq 1$.

(ii) Algorithm `CentRa` [26] recently reduced the number of samples in [20] 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.

All algorithms are implemented by the programming language C++, and their source codes are publicly available[1]. The algorithms are run on a server with an Intel i9-9900K CPU. The frequency of the CPU is between 3.6 GHz and

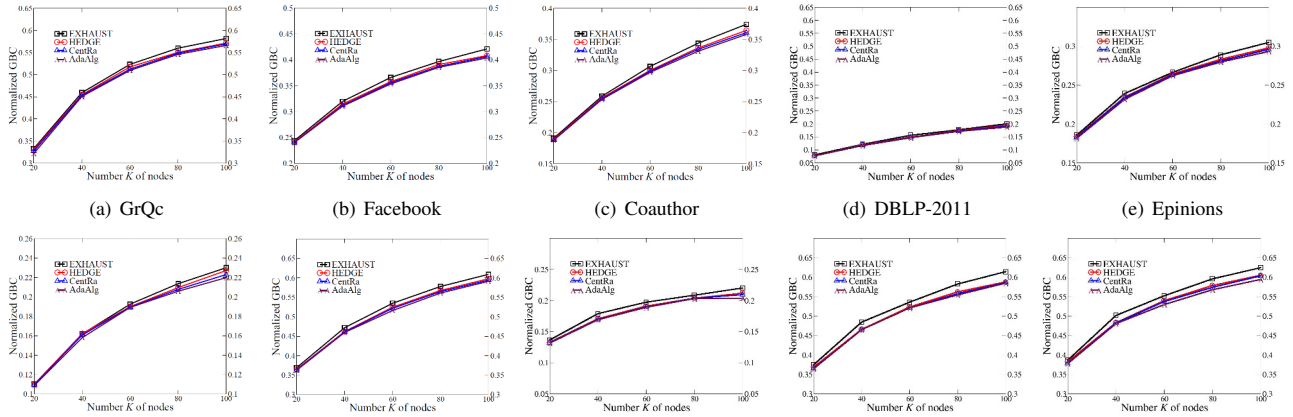
[1] https://github.com/Yu-Huai-M/maxGBC-AdaptiveSamplingAlgorithm



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

(f) Twitter (g) Email-euAll (h) LiveJournal (i) SyntheticNetwork-BA (j) SyntheticNetwork-WS

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 (b) Facebook (c) Coauthor (d) DBLP-2011 (e) Epinions

(f) Twitter (g) Email-euAll (h) LiveJournal (i) SyntheticNetwork-BA (j) SyntheticNetwork-WS

Fig. 2: 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%.

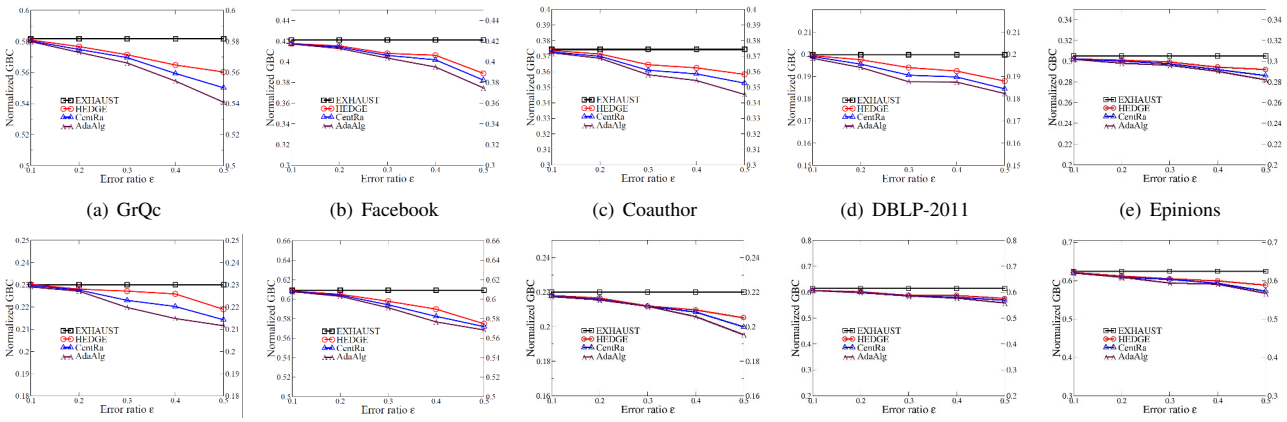
5 GHz. The Memory in the server is a 32 GB RAM with DDR4 2,666 MHz. Each algorithm is run 20 times and its average result is shown.

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 Betweenness Centralities of Found Groups by Different Algorithms

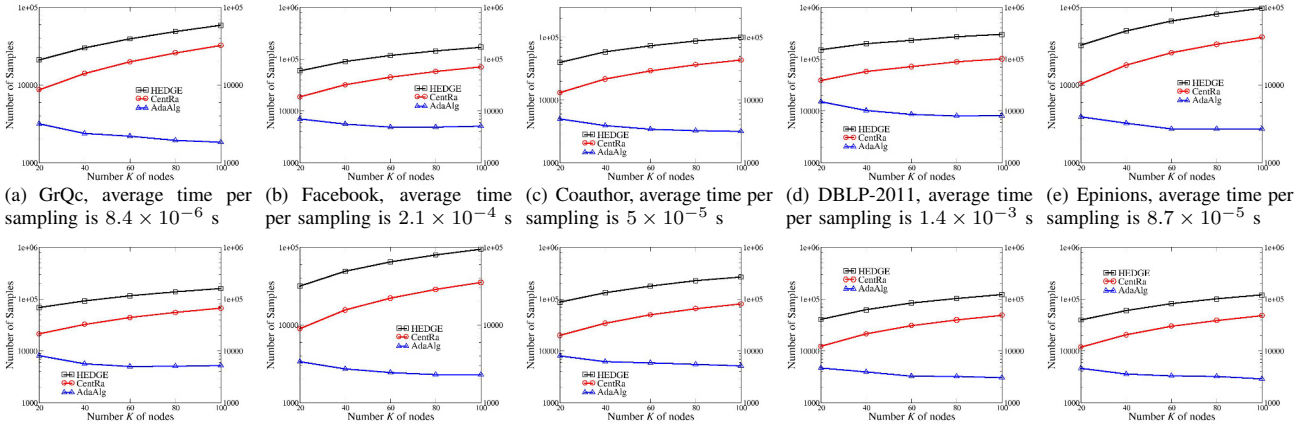
We then investigate the normalized GBCs (Group Betweenness Centralities) of the solutions found by different algorithms, by increasing the number K of nodes from 20 to 100, when the error ratio ϵ is set at 0.3 and the error probability γ is set at 1%. Fig. 2 shows that the normalized GBC by each of the 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`,



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

(f) Twitter (g) Email-euAll (h) LiveJournal (i) SyntheticNetwork-BA (j) SyntheticNetwork-WS

Fig. 3: 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%.



(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 (i) SyntheticNetwork-BA, average time per sampling is 5.9×10^{-5} s (j) SyntheticNetwork-WS, average time per sampling is 6.03×10^{-5} s

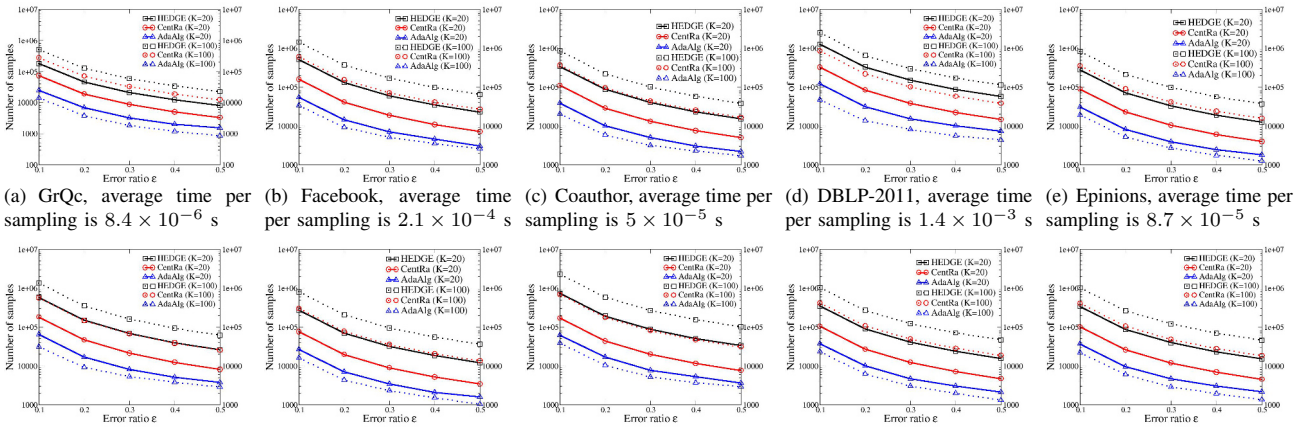
Fig. 4: The numbers of samples used by different algorithms, by increasing the number K of nodes from 20 to 100, when the error ratio ϵ is 0.3 and the error probability γ is 1%.

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 93% of that by algorithm EXHAUST. Denote by ϵ_e the empirical error ratio of algorithm AdaAlg. That is, the 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 a large success probability, the value of the solution then is at least $(1-1/e-0.03)opt$. We thus have $\frac{(1-1/e-\epsilon_e)opt}{(1-1/e-0.03)opt} \geq 93\%$ and $\epsilon_e \leq 7.3\%$, which is much smaller than its theoretical error ratio $\epsilon = 0.3$.

We also study 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. 3 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. 3 demonstrates that the empirical ratio of the solution delivered by algorithm AdaAlg to the solution by algorithm EXHAUST is at least 98%, 97%, 93%, 92%, and 89%, 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 4.3%, 4.9%, 7.3%, 7.9%, and 9.7%, respectively, which are much less than their theoretical error ratios 0.1, 0.2, 0.3, 0.4, and 0.5, respectively.

D. The Numbers of Samples Used by Different Algorithms

We further evaluate the numbers of samples used 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%. Notice that the algorithms



(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

(i) SyntheticNetwork-BA, average time per sampling is 5.9×10^{-5} s

(j) SyntheticNetwork-WS, average time per sampling is 6.03×10^{-5} s

Fig. 5: 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 numbers K of found nodes are 20 and 100, respectively.

are run one by one and only a single thread is used, though there are 16 threads in the CPU of our server. Fig. 4 shows that both the numbers of samples used by algorithms `HEDGE` and `CentRa` increase with K, as both algorithms need to ensure that the maximum deviation of the estimated centrality of every group from its expectation is no greater than a small given threshold $\frac{\epsilon}{2}opt$ 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 number of samples used by algorithm `AdaAlg` first slightly 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 proportional to the value of β. Fig. 4 also demonstrates that the gap between the numbers of samples used by algorithms `CentRa` and `AdaAlg` becomes larger when K increases, and the number of samples used by algorithm `AdaAlg` is from 2.5 to 17 times smaller than that of the state-of-the-art algorithm `CentRa` when K increases from 20 to 100.

We finally study the numbers of samples used by different algorithms, by increasing the error ratio ϵ from 0.1 to 0.5. Fig. 5 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. 5 plots that the number of samples used by algorithm `AdaAlg` is about from 2 to 18 times smaller than that of the state-of-the-art algorithm `CentRa`.

VII. Conclusions

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 a large success probability. Furthermore, experimental results with real-world large-scale networks showed that, the number of samples used by the proposed algorithm is from 2 to 18 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% smaller.

Acknowledgement

The work by Wenzheng Xu was supported by the National Natural Science Foundation of China (NSFC) with grant number 62272328 and Sichuan Science and Technology Program with grant number 2024NSFJQ0026. The work by Jian Peng was supported by the Cooperative Program of Sichuan University and Yibin (2020CDYB-30), the Cooperative Program of Sichuan University and Zigong (2022CDZG-6), the Key R&D Program of Sichuan Province of China (22ZDYF3599), and Sichuan Science and Technology Program under Grant 2022ZDZX0011.

References

[1] M. Barthélemy, "Betweenness centrality in large complex networks," *European Physical 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. Sociology*, 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, article no. 10424, 2021.

[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. Discovery Data*, vol. 17, no. 6, article no. 78, 2023.

[8] S. Dolev, Y. Elovici, R. Puzis, and P. Zilberman, "Incremental deployment of network monitors based on group betweenness centrality," *Inf. Processing Lett.*, vol. 109, no. 20, pp. 1172–1176, 2009.

[9] Allen B. Downey and Franklin W. Olin, *Think Complexity: Exploring Complexity Science with Python – 2e*, Green Tea Press, 2012

[10] M. Fink and J. Spoerhase, "Maximum betweenness centrality: approximability and tractable cases," in *Proc. Int. Workshop Algorithms Comput.*, 2011, pp. 9–20.

[11] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, pp. 35–41, 1977.

[12] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proc. National Academy Sci. (PNAS)*, vol. 99, no. 12, pp. 7821–7826, 2002.

[13] 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. (ICDE)*, 2017, pp. 771–782.

[14] 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. Discovery Data Mining (KDD)*, 2016, pp. 875–884.

[15] K. Huang, J. Tang, K. Han, X. Xiao, W. Chen, A. Sun, X. Tang, and A. Lim, "Efficient approximation algorithms for adaptive influence maximization," *VLDB J.*, vol. 29, pp. 1385–1406, 2020.

[16] S. Iyer, T. Killingback, B. Sundaram, and Z. Wang, "Attack robustness and centrality of complex networks," *PloS one*, vol. 8, no. 4, article no. e59613, 2013.

[17] M. Li, J. Peng, S. Ju, Q. Liu, H. Li, W. Liang, J. X. Yu, abd W. Xu, "Efficient algorithms for finding diversified top-k structural hole spanners in social networks," *Inf. Sci.*, vol. 602, pp. 236–258, 2022.

[18] 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 Express Briefs*, vol. 65, no. 3, pp. 346–350, March 2018.

[19] T. Lou and J. Tang, "Mining structural hole spanners through information diffusion in social networks", in *Proc. ACM Int. Conf. World Wide Web (WWW)*, 2013, pp. 825–836.

[20] A. Mahmoody, C. E. Tsourakakis, and E. Upfal, "Scalable betweenness centrality maximization via sampling," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2016, pp. 1765–1773.

[21] A. Maulana and M. Atzmueller, "Many-objective optimization for anomaly detection on multi-layer complex interaction networks," *Applied Sciences*, vol. 11, no. 9, article no. 4005, 2021.

[22] NetworkX, https://networkx.org/

[23] M. Newman, *Networks*, Oxford University Press, 2018.

[24] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, article no. 026113, 2004.

[25] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions-I," *Math. Program.*, vol. 14, pp. 265–294, Dec. 1978.

[26] L. Pellegrina, "Efficient centrality maximization with Rademacher averages," in *Proc. 29th ACM SIGKDD Conf. Knowl. Discovery Data Mining (KDD)*, 2023, pp. 1872–1884.

[27] L. Pellegrina and F. Vandin, "SILVAN: estimating betweenness centralities with progressive sampling and non-uniform Rademacher bounds," *ACM Trans. Knowl. Discovery Data (TKDD)*, vol. 18, no. 3, article no. 52, pp. 1–55, 2023.

[28] R. Puzis, Y. Elovici, and S. Dolev, "Fast algorithm for successive computation of group betweenness centrality," *Physical Review E*, vol. 76, no. 5, article no. 056709, 2007.

[29] M. Riondato and E. Upfal, "ABRA: approximating betweenness centrality in static and dynamic graphs with Rademacher averages," *ACM Trans. Knowl. Discovery Data*, vol. 12, no. 5, article no. 61, 2018.

[30] Stanford large network dataset collection, http://snap.stanford.edu/data/index.html

[31] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," *Sci.*, vol. 359, pp. 1146–1151, 2018.

[32] 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. (ICDE)*, 2023, pp. 2415–2428.

[33] WOSN 2009 data sets, https://socialnetworks.mpi-sws.org/data-wosn2009.html.

[34] 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.

[35] W. Xu, M. Rezvani, W. Liang, J. X. Yu, C. Liu, "Efficient algorithms for the identification of top-k structural hole spanners in large social networks," *IEEE Trans. Knowl. Data Eng. (TKDE)*, vol. 29, no. 5, pp. 1017–1030, 2017.

[36] Y. Yoshida, "Almost linear-time algorithms for adaptive betweenness centrality using hypergraph sketches," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2014, pp. 1416–1425.

[37] 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. (ICDE)*, 2022, pp. 380–392.