Journal of Global Optimization (2024) 88:199–232
https://doi.org/10.1007/s10898-022-01269-2

# Finding groups with maximum betweenness centrality via integer programming with random path sampling

**Tomás Lagos**[1] · **Oleg A. Prokopyev**[1] · **Alexander Veremyev**[2]

Received: 23 November 2021 / Accepted: 26 December 2022 / Published online: 14 February 2023


**Abstract**
One popular approach to access the importance/influence of a group of nodes in a network is based on the notion of centrality. For a given group, its group betweenness centrality is computed, first, by evaluating a ratio of shortest paths between each node pair in a network that are "covered" by at least one node in the considered group, and then summing all these ratios for all node pairs. In this paper we study the problem of finding the most influential (or central) group of nodes (of some predefined size) in a network based on the concept of betweenness centrality. One known approach to solve this problem exactly relies on using a linear mixed-integer programming (linear MIP) model. However, the size of this MIP model (with respect to the number of variables and constraints) is exponential in the worst case as it requires computing all (or almost all) shortest paths in the network. We address this limitation by considering randomized approaches that solve a single linear MIP (or a series of linear MIPs) of a much smaller size(s) by sampling a sufficiently large number of shortest paths. Some probabilistic estimates of the solution quality provided by our approaches are also discussed. Finally, we illustrate the performance of our methods in a computational study.

**Keywords** Network analysis · Group betweenness centrality · Randomized algorithms · Sample average approximation (SAA) · Linear mixed-integer programming

This research was partially supported by NSF grants CBET-1803527 and CMMI-2002681.

✉ Oleg A. Prokopyev
droleg@pitt.edu

Tomás Lagos
tomaslagos@pitt.edu

Alexander Veremyev
alexander.veremyev@ucf.edu

[1] Department of Industrial Engineering, University of Pittsburgh, Pittsburgh, PA 15261, USA

[2] Department of Industrial Engineering and Management Systems, University of Central Florida, Orlando, FL 32816, USA

## 1 Introduction

One of the key questions in the network analysis area is centered around identifying important (influential) elements (i.e., nodes, edges, subgraphs) of the network [26]. The answer to this question depends on the underlying application setting, and, in particular, on what roles the elements play within the considered network. Over the past several decades a number of more or less sophisticated approaches have been proposed to address this research challenge in the related network analysis and combinatorial optimization literature [5, 11, 20, 26].

One popular approach to access the importance and influence of a graph node is based on the notion of *centrality*, or a *centrality index*. There exist a number of centrality indices that "capture complementary aspects of a node's position" [20] in a network. For example, the *degree centrality* of a node is defined as the degree of the node in a network, possibly, normalized between zero and one. That is, the node is viewed as more influential (or more "central"; hence, the moniker "centrality") if it has a larger size of its immediate "one-hop" neighborhood, e.g., a person is viewed as more important if he/she has more friends (or other types of social connections/interactions) within a social network.

Another interesting and widely used index is *betweenness centrality*, which is considered in this paper. It is defined as follows: to compute betweenness centrality of a given node, for each node pair in the network we compute a proportion of shortest paths (between the pair) that traverse through the given node, and then sum these proportions over all node pairs [11, 26]. In communication and transportation networks, the betweenness centrality of a node can be "interpreted in terms of the potential for controlling flows through the network—that is, playing a gatekeeping or toll-taking role" [5]. We refer the reader to [26] and the references therein for other examples of the centrality indices.

Centrality concept allows us to simply rank all nodes according to their significance to the network structure; see, e.g., [7] for the corresponding polynomial-time approaches. Naturally, the node centrality can be generalized to capture centrality of a group of nodes instead of a single node; see [11]. Formally, given simple undirected graph $G = (V, E)$, where $|V| = n$ and $|E| = m$, we define the betweenness centrality of a group of nodes $S \subset V$ as:

$$C(S) = \frac{2}{n(n-1)} \sum_{i,j \in V: i<j,\, i,j \notin S} \frac{\sigma_{ij}(S)}{\sigma_{ij}}, \tag{1}$$

where $\sigma_{ij}$ denotes the number of shortest paths between nodes $i$ and $j$ in $G$, while $\sigma_{ij}(S)$ denotes the number of shortest paths between the same node pair that traverse through at least one node in $S$.

In (1), we follow the definition similar to the one used in [29] as some of our discussions are based on the results from this study. In particular, we note that the definition of group betweenness centrality by (1) does not take into account shortest paths that have the *endpoints* in $S$. However, if we assume that $G$ is connected (i.e., consists of a single connected component), then:

$$\begin{aligned} \sum_{i,j \in V: i<j} \frac{\sigma_{ij}(S)}{\sigma_{ij}} &= \sum_{i \in S,\ j \in V \setminus S: i<j} \frac{\sigma_{ij}(S)}{\sigma_{ij}} + \sum_{i \in V \setminus S,\ j \in S: i<j} \frac{\sigma_{ij}(S)}{\sigma_{ij}} \\ &\quad + \sum_{i,\ j \in S: i<j} \frac{\sigma_{ij}(S)}{\sigma_{ij}} + \sum_{i,j \in V: i<j,\, i,j \notin S} \frac{\sigma_{ij}(S)}{\sigma_{ij}} \\ &= |S|(n-|S|) + \frac{1}{2}|S|(|S|-1) + \frac{(n-1)n}{2} C(S), \end{aligned} \tag{2}$$

where the first two terms in (2) are constant whenever the cardinality of $S$ is fixed to some positive integer $s \in \mathbb{Z}_{>0}$, i.e., $|S| = s$. In a sense, for a connected graph the first two terms in (2) reflect how many shortest paths are "covered" by any subset of nodes of a given cardinality. Hence, for connected graphs not taking into account the paths with endpoints in $S$ is not a limiting assumption. In the remainder of this paper we assume that $G$ is *connected,* which, in fact, is a standard assumption in various combinatorial optimization settings; see, e.g., a related study in [33].

The main focus of this paper is on the following optimization problem:

$$C^* = \max\{C(S) \,:\, S \subset V,\ |S| = s\}, \tag{3}$$

where $C(S)$ is given by (1) and $s \in \mathbb{Z}_{>0}$ is some given parameter. That is, the decision-maker aims at identifying *the most "central" (important/influential) group of nodes* of some given size (which is typically much smaller than $n$); see, e.g., [33]. There are several application settings, where solving problem (3) is of interest. Some interesting examples include multiple variations of sensor placement and network design [15, 17, 28, 32] as well as data analysis [8, 14] problems.

There are two important observations about the problem given by (3), that need to be pointed out here. *First,* if one wants to estimate the importance (influence) of a group of nodes, $S \subset V$, within a network, then simply summing values of the individual importance of the group members (i.e., their centrality values, $C(i)$, for all $i \in S$) usually does not reflect the importance of a group as a whole; see, e.g., discussions in [11, 12, 23, 33] on this issue. In other words, the top $s$ nodes with respect to their individual betweenness centrality rankings do not provide an optimal solution of (3) in general.

*Second,* finding an optimal solution to problem (3) is an $NP$-hard problem; see, e.g., [10, 13]. In particular, in [13], the authors show that the variant of problem (3) with the objective function as in (2) is APX-hard. One intuitive but relatively simple observation that justifies the computational complexity of (3) from the theoretical perspective is based on the concept of a *vertex cover*. We say that $S$, $S \subseteq V$, forms a vertex cover in $G$ whenever for any $(i, j) \in E$, either $i \in S$, or $j \in S$, or both $i, j \in S$. The problem of checking whether there exits a vertex cover $S$ of cardinality at most $s$ in a given graph $G$, is known to be $NP$-complete [16]; it is typically referred to as the *vertex cover* problem. Then we observe that the value in the right-hand side of (2) is maximum possible whenever $S$ forms a vertex cover. Indeed, if $S$ is not a vertex cover, then there exists $(i, j) \in E$ such that $\{i, j\} \cap S = \emptyset$. Hence, the shortest path between $i$ and $j$ is not "covered" by $S$; recall that $G$ is edge unweighted. Therefore, by finding a group of size $s$ with maximum betweenness centrality, we can verify whether $G$ contains a vertex cover of size $s$. We refer to [13] for some additional discussion on computational complexity of (3) as well as its weighted generalization and polynomially-solvable cases.

In view of the above discussion, it is not surprising that several studies in the related literature explore heuristic and approximate solution algorithms for the optimization problem given by (3). In particular, we note that computing individual betweenness centrality for all nodes in a graph can be done in polynomial time; see, e.g., [7]. Also, for very large graphs there exist randomized polynomial time algorithms that provide estimates of node betweenness centrality with some provable performance guarantees; see [21, 29] and our brief overview in Sect. 2.3. Thus, it is not surprising that the top $s$ nodes with respect to their individual betweenness centrality rankings are often used as a heuristic solution for (3); see examples in [1, 3, 4, 9, 27, 29]. Another related stream of research is focused on finding approximate solutions for problem (3) in polynomial time. In particular, Dolev et al. [10] present a $(1 - \frac{1}{e})$-approximation algorithm. Mahmoody et al. [24] provide a randomized

approach that guarantees a $(1 - \frac{1}{e} - \varepsilon)$-approximate solution with high probability, where $\varepsilon > 0$.

To the best of our knowledge, only the study by Veremyev et al. [33] considers an exact solution approach for problem (3) based on mixed-integer programming (MIP) techniques; see our brief overview in Sects. 2.1 and 2.2. The key idea behind their method is to pre-compute all (or almost all) shortest paths in the graph and then use them for a "set-covering-like" linear mixed-integer programming (linear MIP) model. The main advantage of this approach is its relative simplicity as it relies on using an off-the-shelf MIP solver. Furthermore, this approach can be generalized (via simple linear constraints in the MIP model) to require additional cohesiveness properties for $S$ in (3); for example, the targeted group $S$ should form a clique or a subgraph with some pre-defined structure (e.g., connected by sufficiently short paths). However, the size of the resulting MIP model (with respect to the number of variables and constraints) depends on the the total number of shortest paths in the graph, which can be prohibitively large even for relatively sparse real-life networks (in fact, it is exponential in the worst case). As a result, the computational study presented in [33] demonstrates a somewhat limited applicability of this approach even for moderately sized networks.

The latter observation provides the main motivation behind our study. More specifically, we propose two approaches that combine the random path sampling idea from [29] with the MIP-based techniques from [33]. In our first approach (see Sect. 3), we randomly generate a multi-set (i.e., a *bag*) of shortest paths to form a reasonably sized linear MIP model, which is slightly different from the one proposed in [33]. Let $\varepsilon$ be some pre-defined parameter that controls the quality of the required solution. Then we show that if the cardinality of the generated multi-set is sufficiently large then solving our MIP model provides a solution that is at most $O(\varepsilon)$ away from optimal (with respect to the objective function value) with some prescribed probability. Note that if the value of $\varepsilon$ approaches zero, then the number of shortest paths that need to be generated to ensure the required (theoretical) solution quality becomes somewhat too conservative. However, our computational results indicate is that the proposed randomized approach may provide good quality solutions even when the number of shortest path sampled is fewer than that required by our theoretical derivations. This computational observation provides an interesting avenue for future research.

Our second randomized approach (see Sect. 4) relies on the Sample Average Approximation (SAA) techniques. That is, instead of a single linear MIP model with a large number of randomly generated shortest paths, we solve a series of smaller linear MIPs (i.e., MIPs that are constructed using a smaller number of sampled paths than in the previous approach) until some required solution quality is achieved. The latter is estimated using the arguments that rely on the Central Limit Theorem. In the second set of computational experiments, we compare our two approaches for real-life graphs with up to one million nodes. Finally, in Sect. 5 we provide some concluding remarks and outline promising directions for future research.

## 2 Background

### 2.1 MIPs for finding central groups

Next, we outline the MIP approach proposed in [33] for solving problem (3) and its "bounded-distance" variations. Let $d_{ij}$ be the distance between two distinct nodes $i \in V$ and $j \in V$, $i \neq j$, that is, the minimum number of edges that need to be traversed in order to move

from node $i$ to node $j$ in the network. We denote the diameter of $G$ by $diam(G)$, i.e., $diam(G) = \max\{d_{ij} \mid i, j \in V,\ i \neq j\}$.

Define the *k-bounded-distance group betweenness centrality* as follows:

$$C_{\leq k}(S) = \frac{2}{n(n-1)} \sum_{\substack{i,j \in V:\ i,j \notin S \\ i<j,\ d_{ij} \leq k}} \frac{\sigma_{ij}(S)}{\sigma_{ij}}, \tag{4}$$

which is computed by considering shortest paths with at most $k$ edges, where $k \geq 2$. Clearly, if $k \geq diam(G)$, then $C_{\leq k}(S) = C(S)$.

The centrality concept given by (4) is motivated by the fact that in many real-life settings only sufficiently short paths are practically relevant; see, e.g., brief discussions on this issue in [6, 33]. Hence, when estimating the corresponding "influence" (importance) of the group, $S$, the decision-maker disregards shortest paths longer than some given $k$. Furthermore, for $2 \leq k < diam(G)$, the value of $C_{\leq k}(S)$ can be used to approximate $C(S)$; see further details in Sect. 2.2.

Denote by $\mathcal{P}_{ij}$ a set that contains all shortest paths between two distinct nodes $i$ and $j$, $i \neq j$. Each path $p_t \in \mathcal{P}_{ij}$, $t \in \{1, \ldots, \sigma_{ij}\}$, represents an ordered list of nodes on some shortest path (indexed by $t$) from $i$ to $j$; note that $p_t$ includes both end-nodes $i$ and $j$. That is, if a shortest path between $i$ and $j$ is given by $i$–$i_1$–$\ldots$–$i_{d_{ij}-1}$–$j$, then we denote $p_t$ by $p_t = \langle i, i_1, \ldots, i_{d_{ij}-1}, j\rangle$ and $|p_t| = d_{ij} + 1$. One possible approach for computing sets $\mathcal{P}_{ij}$ for all node pairs $i$ and $j$ is based on a rather simple modification of Brandes algorithm [7]; we refer the reader to further details in [33]. Note that whenever all paths in $\mathcal{P}_{ij}$ are pre-computed in order to formulate our linear MIP model, then the corresponding value of $\sigma_{ij}$ is also readily available.

Next, we introduce the following sets of decision variables. Let $x_i$ be a 0–1 variable indicating whether $i \in V$ belongs to the selected group of nodes $S$, i.e., $x_i = 1$ if and only if $i \in S$. For all node pairs $i, j \in V$, $i \neq j$, $(i, j) \notin E$ and $t \in \{1, \ldots, \sigma_{ij}\}$, we denote by $y_{ij}^t$ a 0–1 variable indicating whether $p_t \in \mathcal{P}_{ij}$ traverses through at least one node in $S$, i.e., $y_{ij}^t = 1$ if and only if there exists $q \in p_t \setminus \{i, j\}$ such that $q \in S$. Recall that in (1) and (4) we do not take into account the endpoints of the path. Given the above notation, we obtain the following MIP proposed in [33]:

$$\textbf{[F1}(k)\textbf{]:} \quad C^*_{\leq k} = \max \ \frac{2}{n(n-1)} \sum_{\substack{i,j \in V:\ i<j \\ 1<d_{ij} \leq k}} \frac{\sum_{p_t \in \mathcal{P}_{ij}} y_{ij}^t}{\sigma_{ij}} \tag{5a}$$

subject to

$$y_{ij}^t \leq \sum_{q \in p_t \setminus \{i,j\}} x_q \quad \forall\, i, j \in V,\ (i,j) \notin E,\ i < j,\ d_{ij} \leq k,\ \forall\, p_t \in \mathcal{P}_{ij}, \tag{5b}$$

$$y_{ij}^t \leq 1 - x_i \quad \forall\, i, j \in V,\ (i,j) \notin E,\ i < j,\ d_{ij} \leq k,\ \forall\, p_t \in \mathcal{P}_{ij}, \tag{5c}$$

$$y_{ij}^t \leq 1 - x_j \quad \forall\, i, j \in V,\ (i,j) \notin E,\ i < j,\ d_{ij} \leq k,\ \forall\, p_t \in \mathcal{P}_{ij}, \tag{5d}$$

$$\sum_{i \in V} x_i = s, \tag{5e}$$

$$x_i \in \{0, 1\} \quad \forall i \in V, \tag{5f}$$

$$y_{ij}^t \geq 0 \quad \forall\, i, j \in V,\ (i,j) \notin E,\ i < j,\ d_{ij} \leq k,\ \forall\, p_t \in \mathcal{P}_{ij}, \tag{5g}$$

where (5b) is a set-covering constraint, which ensures that only the paths "covered" by the nodes in $S$ can contribute to (5a). Constraints (5c) and (5d) guarantee that the shortest paths

with endpoints in $S$ do not contribute to the objective function value; recall (1) and (4) as well as our discussion of (2). Constraint (5e) provides the cardinality restriction for $S$, i.e., $|S| = s$. Constraints (5f) and (5g) represent binary and lower bounding limitations for the decision variables, respectively. Finally, we note that no integrality restrictions for variables $y_{ij}^t$ are needed due to the maximization nature of the objective function in (5a), and the right-hand sides of the constraints in (5c) and (5d).

### 2.2 Bounded-distance betweenness and betweenness centralities

As the network size increases the total number of all-pairs shortest paths in the network may grow very fast; in fact, the growth is exponential in the worst case. Moreover, the study in [33] with real-life graphs demonstrates that even if the total number of shortest paths in a graph is approximately quadratic (with respect to $n$), then the number of variables and constraints in $\mathbf{F1}(k)$ may become prohibitively large for an off-the-shelf linear MIP solver, whenever $k = diam(G)$. Recall that the latter condition ensures that the MIP-based approach outlined in Sect. 2.1, i.e., model $\mathbf{F1}(k)$, provides an exact solution for (3). Therefore, it is natural to explore whether solving $\mathbf{F1}(k)$ with $k < diam(G)$ provides a near-optimal solution of (3) for reasonably large values of $k$.

Next, we briefly describe the results and observations from [33], where the posed research question is considered. Define:

$$C_{>k}(S) = \frac{2}{n(n-1)} \sum_{\substack{i,j \in V:\, i,j \notin S \\ i<j,\ d_{ij}>k}} \frac{\sigma_{ij}(S)}{\sigma_{ij}}, \tag{6}$$

which is the betweenness centrality of $S$ considering all paths longer than $k$. Thus, we have:

$$\begin{aligned} &C(S) = C_{\leq k}(S) + C_{>k}(S), \qquad \text{where} \\ &C(S) = C_{\leq \mathrm{diam}(G)}(S) \quad \text{and} \quad C_{>k}(S) = 0 \text{ for all } k > \mathrm{diam}(G). \end{aligned} \tag{7}$$

Furthermore, denote by $S^*$ and $S_k^*$ the most influential groups of nodes according to $C(S)$ and $C_{\leq k}(S)$, respectively, for some given $k$ and the group size $s$, i.e., $|S^*| = s$ and $|S_k^*| = s$. Therefore,

$$\begin{aligned} &S^* \in \arg\max\{C(S) \,:\, S \subset V,\ |S| = s\} \qquad \text{and} \\ &S_k^* \in \arg\max\{C_{\leq k}(S) \,:\, S \subset V,\ |S| = s\}. \end{aligned} \tag{8}$$

Then it is shown in [33] that:

**Proposition 1** *Let $S^*$ and $S_k^*$ be given by (8). Then:*

$$C(S^*) - C(S_k^*) \leq \Delta_k(S_k^*) := \frac{2}{n(n-1)} \sum_{i,j \in V:\, i<j} \mathbb{1}_{\{d_{ij}>k\}} - C_{>k}(S_k^*), \tag{9}$$

*where $\mathbb{1}$ denotes the standard indicator function, i.e., in (9) it returns 1, if $d_{ij} > k$, and 0, otherwise.*

The value of $\Delta_k(S_k^*)$ in the right-hand side of (9) can be computed rather effectively whenever sets $S_k^*$ and $\mathcal{P}_{ij}$ for all node pairs $i, j \in V$ are known, i.e., by explicitly counting the number of shortest paths of length larger than $k$ that traverse through $S_k^*$. Hence, by increasing the value of $k$ and solving $\mathbf{F1}(k)$ we can achieve the desired quality of approximation. The upper bound for the latter is estimated via (9) for an optimal solution of $\mathbf{F1}(k)$ given that

all shortest paths in the graph are pre-computed. This approach is considered in [33] and its pseudo-code is summarized in Algorithm 1.

**Algorithm 1:** $\Delta$-Approximation Algorithm ($\Delta$-**AA**); see [33]

**Input:** A graph $G = (V, E)$, the required solution quality guarantee $\Delta$ and the group size $s$;
**Result**: Subset $S^{\Delta} \subseteq V$ such that $|S^{\Delta}| = s$ and $\Delta_k(S^{\Delta}) \leq \Delta$
$k \leftarrow 1$;
**repeat**
  $k \leftarrow k + 1$;
  $S^{\Delta} \leftarrow S_k^* \in \text{argmax}\{C_{\leq k}(S) \,:\, S \subseteq V,\ |S| = s\}$ via MIP $\mathbf{F1}(k)$ in (5) ;
  $\Delta_k(S^{\Delta}) \leftarrow 2/(n(n-1)) \sum_{i,j \in V,\ i<j} \mathbb{1}_{k<d_{ij}} - C_{>k}(S^{\Delta})$
**until** $\Delta_k(S^{\Delta}) \leq \Delta$;
**Return:** $S^{\Delta}$

Finally, it should be pointed out that (7) and (9) in Proposition 1 provide an approach for computing an upper bound for $C(S^*)$. That is:

$$\begin{aligned} C(S^*) \leq UB_k :&= \frac{2}{n(n-1)} \sum_{i,j \in V,\ i<j} \mathbb{1}_{\{d_{ij}>k\}} - C_{>k}(S_k^*) + C(S_k^*) \\ &= \frac{2}{n(n-1)} \sum_{i,j \in V,\ i<j} \mathbb{1}_{\{d_{ij}>k\}} - C_{>k}(S_k^*) + C_{\leq k}(S_k^*) + C_{>k}(S_k^*) \\ &= \frac{2}{n(n-1)} \sum_{i,j \in V,\ i<j} \mathbb{1}_{\{d_{ij}>k\}} + C_{\leq k}^*, \end{aligned} \tag{10}$$

where the second term in (10) is the optimal objective function of $\mathbf{F1}(k)$; see (5).

**Algorithm 2:** Randomized approximation of betweenness centrality for each $v \in V$; see [29]

**Input:** Graph $G = (V, E)$, parameter $r$;
**Result**: Multiset of shortest paths $X$ such that $r = |X|$, and $C(v, X)$ for all $v \in V$
Let $C(v) \leftarrow 0$ for all $v \in V$;
Let $X \leftarrow \emptyset$;
**for** $\ell = 1, \ldots, r$ **do**
  Sample uniformly at random two distinct nodes from $i, j \in V$;
  Compute all shortest paths $\mathcal{P}_{ij}$ using Brandes subroutine for $i, j$;
  Let $p_\ell \leftarrow \{j\}$;
  Let $t \leftarrow j$;
  **while** $t \neq i$ **do**
    Sample $z$ from the predecessors of $t$ in the path from $i$ to $t$, with probability $\frac{\sigma_{iz}}{\sigma_{it}}$;
    $p_\ell \leftarrow \{z\} \cup p_\ell$;
    **if** $z \neq i$ **then**
      $C(z) \leftarrow C(z) + \frac{1}{r}$;
    **end**
    $t \leftarrow z$;
  **end**
  $X \leftarrow X \cup \{p_\ell\}$ ;
**end**
**Return:** $X$ and $C(v, X)$:=$C(v)$ for all $v \in V$

### 2.3 Randomized $(\varepsilon, \delta)$-approximation of node betweenness centrality

The study in [29] describes a randomized algorithm for approximating betweenness centrality of nodes in a network. Specifically, the key idea is to randomly sample a *multiset* (a bag[1]) $X$ of shortest paths in a given network and then approximate $C(v)$ for each $v \in V$ as:

$$C(v, X) = \frac{1}{|X|} \sum_{p \in X} \mathbb{1}_{\{v \in \text{Int}(p)\}}, \tag{11}$$

where $\text{Int}(p)$ denotes the *interior vertices* of path $p$ (see Algorithm 2). Formally, given $p = \langle i, i_1, \ldots, i_{|p|-2}, j \rangle \in \mathcal{P}_{ij}$ we define:

$$\text{Int}(p) := \{i_1, \ldots, i_{|p|-2}\}, \tag{12}$$

where $|p| = d_{ij} + 1$. Each path in $X$ is an element of $\mathcal{P} := \cup_{i,j \in V} \mathcal{P}_{ij}$. However, as $X$ is constructed using a randomized approach, then $X$ may contain multiple copies of the same shortest path. That is, $X$ is a *multiset*.

In [29], the authors show that if $X$ is sufficiently large, then it is possible to derive some probabilistic guarantees on the quality of $C(v, X)$. Formally, given $\varepsilon, \delta \in (0, 1)$ pair, let

$$r \geq \frac{c}{\varepsilon^2} \left( \lfloor \log_2(\text{diam}(G) - 2) \rfloor + 1 + \ln \frac{1}{\delta} \right), \tag{13}$$

where $c \approx 0.5$. The lower bound for $r$ in (13) is constructed using the arguments based on a finite upper bound for a particular Vapnik-Chervonenkis (VC) dimension. Loosely speaking, "a finite bound on the VC-dimension of the class of subsets implies a bound on the number of random samples required to approximate the probability of each subset in the class with its empirical average"; see [29]. Then a randomized approach outlined in Algorithm 2 with $|X| \geq r$ has the following property:

**Proposition 2** [29] *If* (13) *holds, then the approximations of* $C(v)$*, for all* $v \in V$*, computed by Algorithm* 2 *are within* $\varepsilon$ *from their real values with probability at least* $1 - \delta$*, i.e.:*

$$Pr\{|C(v) - C(v, \mathbf{X})| \leq \varepsilon, \ \forall v \in V\} \geq 1 - \delta. \tag{14}$$

We refer to $C(v, X)$ as an $(\varepsilon, \delta)$-approximation of $C(v)$ whenever (14) holds. Note also, that in (14) and in the remainder of the paper, we use symbols in bold to indicate random variables. Specifically, $\mathbf{X}$ denotes a random variable that corresponds to a randomly generated multiset (or bag) of shortest paths, and $X$ denotes a realization of $\mathbf{X}$.

Furthermore, it is worth pointing out that the values of $\sigma_{iz}$ and $\sigma_{it}$ in line 10 of Algorithm 2 are available due to the computations performed in line 6; see further details in [29]. The pseudo-code outlined in Algorithm 2 is a slightly modified version of the one presented in [29]. Specifically, our version, in addition to betweenness centrality approximations, also returns a multiset $X$ of shortest paths; this feature is exploited in our approach discussed in the next section.

## 3 Randomized approximation of $C^*$

Next, we discuss a randomized approximation approach that combines the ideas behind the methods discussed in Sects. 2.2 and 2.3. Our approach generates at random a multiset of

[1] A multiset is a generalization of a set and allows multiple instances of each of its elements.

shortest paths and then solves a linear MIP to find a group that provides the maximum betweenness centrality estimate with respect to the considered multiset of shortest paths. It can be shown that if the cardinality of the generated multiset of shortest paths is sufficiently large, then there exists some probabilistic guarantees on the quality of the obtained solution.

### 3.1 Estimator of $C(S)$

Given a multiset of shortest paths $X$ we estimate the group betweenness centrality of $S$ as

$$C(S, X) = \frac{1}{|X|} \sum_{p=\langle i,\ldots,j\rangle \in X:\ i,j \notin S} \mathbb{1}_{\{|\mathrm{Int}(p)\cap S|\geq 1\}}, \tag{15}$$

which is a generalization of (11) for a subset of vertices. That is, if $S = \{v\}$, then $C(S, X)$ reduces to $C(v, X)$ given by (11).

Recall that $\mathcal{P}$ is the set of all shortest paths in $G$, and that for any path $p \in \mathcal{P}$, $\mathrm{Int}(p)$ denotes the set that consists of its interior vertices. Then given a group of vertices $S \subseteq V$, we define

$$\mathcal{T}_S = \{p = \langle i, \ldots, j\rangle \in \mathcal{P} \ :\ |\mathrm{Int}(p) \cap S| \geq 1,\ i, j \notin S\}, \tag{16}$$

which is simply a subset of shortest paths such that each path within the subset contains at least one node from $S$ in its interior.

Recall that $\mathbf{X}$ denotes a random variable that corresponds to a randomly generated multiset of shortest paths. Whenever each path in $X$, $r = |X|$, is sampled uniformly at random from $\mathcal{P}$ (e.g., based on Algorithm 2) we observe that the expectation of $C(S, \mathbf{X})$ is given by:

$$\begin{aligned}
\mathbb{E}[C(S,\mathbf{X})] &= \frac{1}{r}\mathbb{E}[|\mathbf{X}\cap\mathcal{T}_S|] = \frac{1}{r}\sum_{p\in\mathcal{P}} \mathbb{1}_{\{p\in\mathcal{T}_S\}}\mathbb{E}\left[\sum_{\ell=1}^{r}\mathbb{1}_{\{\mathbf{p}_\ell = p\}}\right] \\
&= \frac{1}{r}\sum_{p\in\mathcal{P}} \mathbb{1}_{\{p\in\mathcal{T}_S\}}\sum_{\ell=1}^{r} Pr\{\mathbf{p}_\ell = p\} = \frac{1}{r}\sum_{p\in\mathcal{P}} \mathbb{1}_{\{p\in\mathcal{T}_S\}} r Pr\{\mathbf{p}_1 = p\} \\
&= \sum_{i,j\in V: i\neq j}\ \sum_{p\in\mathcal{P}_{ij}} \mathbb{1}_{\{p\in\mathcal{T}_S\}}\frac{1}{n(n-1)}\frac{1}{\sigma_{ij}} = \sum_{i,j\in V: i\neq j}\frac{1}{n(n-1)\sigma_{ij}}\sum_{p\in\mathcal{P}_{ij}}\mathbb{1}_{\{p\in\mathcal{T}_S\}} \\
&= \sum_{i,j\in V: i\neq j, i,j\notin S}\frac{1}{n(n-1)\sigma_{ij}}\sigma_{ij}(S) = C(S),
\end{aligned} \tag{17}$$

and therefore, $C(S, \mathbf{X})$ provides an *unbiased estimator* of $C(S)$. Furthermore, using the same arguments it can be shown that the *variance* of this estimator is given by:

$$\mathrm{Var}[C(S,\mathbf{X})] = \frac{C(S) - C(S)^2}{r}, \tag{18}$$

and we conclude that the quality of the estimator, $C(S, X)$, should improve as $r \to +\infty$.

### 3.2 Finding cardinality of $X$

Given (17) and (18), it is natural to derive an appropriate lower bound for the value of $r$, i.e., the cardinality of $\mathbf{X}$, to ensure that $C(S, \mathbf{X})$ provides an approximation for $C(S)$, similar to Proposition 2. We need to exploit the following classical result:

**Lemma 1** (Hoeffding Bound, Theorem 4.12 [25]) *Let* $\mathbf{Y}_1, \ldots, \mathbf{Y}_r$ *be independent random variables such that for all* $1 \le i \le r$, $\mathbb{E}[\mathbf{Y}_i] = \mu$ *and* $Pr\{0 \le \mathbf{Y}_i \le 1\} = 1$. *Then*

$$Pr\left\{\left|\frac{1}{r}\sum_{i=1}^{r}\mathbf{Y}_i - \mu\right| \ge \varepsilon\right\} \le 2e^{-2r\varepsilon^2}. \tag{19}$$

Then we obtain a lower bound for $r$:

**Lemma 2** *If Algorithm 2 is used to sample multiset* $X$ *such that* $r = |X|$ *and*

$$r \ge \frac{1}{2\varepsilon^2}\ln\frac{2}{\delta}, \tag{20}$$

*then we have:*

$$Pr\{|C(S, \mathbf{X}) - C(S)| \le \varepsilon\} \ge 1 - \delta, \quad \forall S \subset V. \tag{21}$$

***Proof*** Given $S \subset V$, we select a shortest path $p$ from $\mathcal{P}$ uniformly at random. Define random variable $\mathbf{Y}_i$ to be 1 if one of the interior nodes of $p$ belongs to $S$ and 0, otherwise. Taking into account (15) and (17), we apply Lemma 1 and obtain the necessary result.

It is important to point out that in both (13) and (20), we have $r \ge O(\frac{1}{\varepsilon^2}\ln\frac{1}{\delta})$. However, in contrast to (13), the bound for $r$ in (20) does not depend on the graph diameter. Next, we exploit Lemma 2 and a modified version of $\mathbf{F1}(k)$ to construct our randomized approach.

### 3.3 Algorithm and its performance bound

We introduce a modified version $\mathbf{F1}(k)$ that seeks the most central group with respect to (15) for a given multiset of shortest paths $X$. That is, we consider the problem of the form:

$$C^*_{\mathbf{X}} = \max\{C(S, \mathbf{X}) \ : \ S \subset V, \ |S| = s\}, \tag{22}$$

and for a realization of $\mathbf{X}$, the corresponding MIP is given as follows:

$$\textbf{[F2(X)]:} \qquad C^*_X = \max\left[\sum_{i,j\in V,\, i\ne j} \frac{\sum_{p_t=\langle i,\ldots,j\rangle\in X} y^t_{ij}}{|X|}\right] \tag{23a}$$

subject to

$$y^t_{ij} \le \sum_{q\in p_t\setminus\{i,j\}} x_q, \ \forall\, p_t = \langle i, i_1, \ldots, i_{|p_t|-2}, j\rangle \in X, \tag{23b}$$

$$y^t_{ij} \le 1 - x_i, \ \forall\, p_t = \langle i, i_1, \ldots, i_{|p_t|-2}, j\rangle \in X, \tag{23c}$$

$$y^t_{ij} \le 1 - x_j, \ \forall\, p_t = \langle i, i_1, \ldots, i_{|p_t|-2}, j\rangle \in X, \tag{23d}$$

$$\sum_{i\in V} x_i = s, \tag{23e}$$

$$x_i \in \{0, 1\} \qquad \forall i \in V, \tag{23f}$$

$$y^t_{ij} \ge 0, \qquad \forall\, p_t = \langle i, i_1, \ldots, i_{|p_t|-2}, j\rangle \in X, \tag{23g}$$

where we also assume that each shortest path $p \in X$ contains at least two edges. The key ideas behind $\mathbf{F2}(X)$ are somewhat similar to those behind $\mathbf{F1}(k)$. In particular, we use $p_t$ to index shortest paths in $X$ between a node pair $i$ and $j$; see (23b), (23c), (23d) and (23g). Hence, the

feasible regions of $\mathbf{F2}(X)$ with $X = \mathcal{P}$ and $\mathbf{F1}(k)$ with $k \geq \text{diam}(G)$ coincide. To confirm this observation, it is sufficient to compare the constraints and variables $y_{ij}^t$ corresponding to each shortest path $p_t$ between these two formulations.

However, when comparing the denominators in (5a) and (23a) we observe that $\mathbf{F1}(k)$ and $\mathbf{F2}(X)$ differ in their coefficients in the respective objective functions. For the latter MIP model, the required approximation quality is achieved by appropriate sampling of a sufficiently large number of shortest paths with possibly multiple copies of some paths; recall our discussion in Sect. 3.1.

Our randomized approach, see its pseudo-code in Algorithm 3, combines MIP $\mathbf{F2}(X)$, Algorithm 2 from Sect. 2.3 and the theoretical observations in Sects. 3.1 and 3.2. In particular, the latter results, namely, Lemma 2, are exploited to derive the required cardinality of $X$ in order to provide the probabilistic performance bound.

**Proposition 3** *If Algorithm 2 is used to sample multiset $X$ such that $r = |X|$ and*

$$r \geq \frac{1}{2\varepsilon^2}\left(\ln\binom{n}{s} + \ln\frac{2}{\delta}\right), \tag{24}$$

*then:*

$$Pr\left\{\max_{S\subset V\,:\,|S|=s} |C(S,\mathbf{X}) - C(S)| \leq \varepsilon\right\} \geq 1-\delta. \tag{25}$$

***Proof*** Define $\mathcal{S} := \{S \subset V : |S| = s\}$. By the union bound, e.g., see Lemma 1.2 in [25],

$$\begin{aligned} Pr\left\{\max_{S\in\mathcal{S}} |C(S,\mathbf{X}) - C(S)| \leq \varepsilon\right\} &\geq 1 - Pr\left\{\bigcup_{S\in\mathcal{S}} |C(S,\mathbf{X}) - C(S)| \geq \varepsilon\right\} \\ &\geq 1 - \sum_{S\in\mathcal{S}} Pr\{|C(S,\mathbf{X}) - C(S)| \geq \varepsilon\} \\ &\geq 1 - |\mathcal{S}|2e^{-2r\varepsilon^2} = 1 - 2\binom{n}{s}e^{-2r\varepsilon^2}, \end{aligned} \tag{26}$$

where (26) follows from Lemma 1. Next, we apply (24) to obtain the following:

$$\begin{aligned} Pr\left\{\max_{S\in\mathcal{S}} |C(S,\mathbf{X}) - C(S)| \leq \varepsilon\right\} &\geq 1 - 2|\mathcal{S}|e^{-2\varepsilon^2\left(\frac{1}{2\varepsilon^2}\left(\ln|\mathcal{S}|+\ln\frac{2}{\delta}\right)\right)} = 1 - 2|\mathcal{S}|e^{-\left(\ln|\mathcal{S}|+\ln\frac{2}{\delta}\right)} \\ &= 1-\delta, \end{aligned}$$

which completes the proof.

**Algorithm 3:** Probabilistic-Approximation Algorithm **[PAA]**

1 **Input:** Graph $G = (V, E)$, the group size $s$, the number of shortest paths $r$;
**Result**: An estimate of $C^*$ and the corresponding group of $s$ nodes given by $C_X^*$ and $S_X^*$, respectively
2 Generate multiset $X$, where $r = |X|$, using Algorithm 2 with input $(G, r)$;
3 Solve $\mathbf{F2}(X)$; let $x^*$, $C_X^*$ be an optimal solution and the optimal objective function value of $\mathbf{F2}(X)$, respectively;
4 Let $S_X^* = \{i \in V : x_i^* = 1\}$;
5 **Return:** $C_X^*$ and $S_X^*$

**Theorem 1** *Let $C^*$ be given by (3) and $C_X^*$ be computed by Algorithm 3. If $r = |X|$ satisfies (20) then:*

$$Pr\{C^* - C_{\mathbf{X}}^* \leq \varepsilon\} \geq 1 - \delta. \tag{27}$$

*Moreover, if $r$ satisfies (24), then:*

$$Pr\{|C^* - C_{\mathbf{X}}^*| \leq \varepsilon\} \geq 1 - \delta, \quad \text{and} \tag{28}$$

$$Pr\left\{C^* - C(S_{\mathbf{X}}^*) \leq \left(1 + \sqrt{\frac{\ln\frac{2}{\delta}}{\ln\binom{n}{s} + \ln\frac{2}{\delta}}}\right)\varepsilon\right\} \geq (1-\delta)^2.$$

***Proof*** Recall that $S^*$ denotes an optimal solution of (3), i.e., $C^* = C(S^*)$, and $S_X^*$ is the solution of (23), i.e., $C_X^* = C(S_X^*, X)$. From the latter we conclude that $C(S^*, X) \leq C_X^*$, which can be used to obtain (27). Specifically, we apply the condition in (20) and Lemma 2. Then we observe that:

$$Pr\{C^* - C_{\mathbf{X}}^* \leq \varepsilon\} \geq Pr\{C^* - C(S^*, \mathbf{X}) \leq \varepsilon\} \geq 1 - \delta.$$

which implies that the inequality in (27) holds.

To establish (28), we first observe that if (24) holds, then by Proposition 3:

$$Pr\{|C_{\mathbf{X}}^* - C(S_{\mathbf{X}}^*)| \leq \varepsilon\} \geq 1 - \delta. \tag{29}$$

To prove the left inequality in (28), note that in view of (27), it is sufficient to consider the case of $C_X^* > C^*$. Given that $C^* \geq C(S_X^*)$, we conclude

$$C_X^* - C^* \leq C_X^* - C(S_X^*) \leq |C_X^* - C(S_X^*)|,$$

which, in view of of (29) implies the left inequality in (28).

Furthermore, by Lemma 1 (using the same arguments as in the proof in Lemma 2), we have:

$$Pr\left\{|C^* - C(S^*, \mathbf{X})| \leq \varepsilon\sqrt{\frac{\ln\frac{2}{\delta}}{\ln\binom{n}{s} + \ln\frac{2}{\delta}}}\right\} \geq 1 - 2e^{-2r\varepsilon^2\frac{\ln\frac{2}{\delta}}{\ln\binom{n}{s}+\ln\frac{2}{\delta}}}$$

$$\geq 1 - 2e^{-2\frac{1}{2\varepsilon^2}\left(\ln\binom{n}{s}+\ln\frac{2}{\delta}\right)\varepsilon^2\frac{\ln\frac{2}{\delta}}{\ln\binom{n}{s}+\ln\frac{2}{\delta}}} = 1 - \delta.$$

Hence, the right inequality in (28) can be shown to hold using the following sequence of inequalities:

$$\begin{aligned} C^* - C(S_X^*) &= C^* - C(S^*, X) + C(S^*, X) - C(S_X^*) \\ &\leq C^* - C(S^*, X) + C(S_X^*, X) - C(S_X^*) \\ &\leq |C^* - C(S^*, X)| + |C(S_X^*, X) - C(S_X^*)| \leq \varepsilon\sqrt{\frac{\ln\frac{2}{\delta}}{\ln\binom{n}{s} + \ln\frac{2}{\delta}}} + \varepsilon, \end{aligned}$$

where the last inequality holds with probability at least $(1-\delta)^2$, whenever $r$ is sufficiently large. This observation completes the proof.

We should point out that **PAA** is not a polynomial-time algorithm as it requires solution of a linear MIP problem. However, if the parameters $\varepsilon$ and $\delta$ are sufficiently large, then the size of the corresponding MIP should be smaller than the size of the MIP required to solve in order

to obtain an exact solution of our problem. Finally, we note that (24) is rather conservative. Hence, in our computations we explore weaker requirements; see the discussion next.

### 3.4 Computational Study

In this section we report the results for the first set of our computational experiments; further experiments are provided in Sect. 4. Our goal is to compare the following three approaches: (i) $\Delta$-**AA** from [33] outlined in Sect. 2.2, (ii) our approach **PAA** proposed in Sect. 3.3 with different choices of the parameter $r$, and (iii) the Top-$s$ approach from [4]. The Top-$s$ approach simply picks $s$ nodes with the largest values of their individual betweenness centrality estimates using Proposition 2. The primary advantage of the Top-$s$ approach is that from the theoretical computational complexity perspective, it is a polynomial-time algorithm [4, 7]; in contrast, recall that (3) is a difficult problem as outlined in Sect. 1.

In view of (24), (13) and (20), for the parameter $r$ in the **PAA** algorithm we consider:

$$r^{(1)} = \left\lceil \frac{1}{2\varepsilon^2}\left(\ln\binom{n}{s} + \ln\frac{2}{\delta}\right)\right\rceil, \tag{30}$$

$$r^{(2)} = \left\lceil \frac{1}{2\varepsilon^2}\left(\lfloor \log_2(\mathrm{diam}(G)-2)\rfloor + 1 + \ln\frac{2}{\delta}\right)\right\rceil, \tag{31}$$

$$r^{(3)} = \left\lceil \frac{1}{2\varepsilon^2}\ln\frac{2}{\delta}\right\rceil, \tag{32}$$

respectively, and we set $\delta = 10^{-3}$ for both **PAA** and Top-$s$ in our experiments throughout the paper. We use $r := r^{(2)}$ in the Top-$s$ approach as we use its implementation from [4]. This setting provides guarantees on the estimates of the node betweenness centrality; see Proposition 2.

Given Theorem 1, by setting $r := r^{(1)}$ in **PAA** we obtain the strongest solution quality guarantees, namely, the probabilistic bounds given by (28). In contrast to $r^{(1)}$, by setting either $r := r^{(2)}$ or $r := r^{(3)}$ we obtain a weaker performance guarantee for **PAA**; see (27). However, either $r := r^{(2)}$ or $r := r^{(3)}$ should imply an improved running time performance of **PAA** in comparison to $r := r^{(1)}$. Hence, it is of interest to explore the quality of the corresponding solutions obtained by **PAA**.

In this section we use **PAA** only with $r := r^{(1)}$ and $r := r^{(2)}$ as our focus is on the comparison with $\Delta$-**AA**. The setting $r := r^{(3)}$ results in the smallest number of paths that need to be sampled; we report the results for **PAA** with $r := r^{(3)}$ in Sect. 4, where we explore networks of larger sizes than those in this section.

All experiments were performed on an HP Z220 CMT Workstation with an Intel(R) Core(TM) i5-3470 CPU 3.20GHz processor, 22GB of RAM and Windows 10 platform. All algorithms throughout the paper are implemented in C++. All MIPs are solved using Gurobi 9.0.1 [18] with the default parameter settings.

**Test instances**. Our test bed includes both real-life and randomly generated instances that are publicly available.[2] This set of experiments is based on the following five networks from [30]:

- **ieeebus_118** ($|V| = 118$, $|E| = 179$, $diam(G) = 14$): The IEEE 118-bus test case serving as an approximation of the American Electric Power system (in the U.S. Midwest) in 1962;

[2] The additional datasets generated during and/or analysed during the current study are available from the corresponding author on request.

- **494_bus** ($|V| = 494,\ |E| = 586, diam(G) = 26$) and **662_bus** ($|V| = 662,\ |E| = 906,\ diam(G) = 25$): power system networks;
- **USAir97** ($|V| = 332,\ |E| = 2126, diam(G) = 6$): a network of direct flight connections between a subset of US airports; **fb-pages-food** ($|V| = 620,\ |E| = 2096,\ diam(G) = 17$): a social network of mutually liked Facebook pages; each node represents a Facebook page and edges correspond to mutual likes.

Also, the following networks were randomly generated using NetworkX library for Python [19]:

- *Watts-Strogatz graphs* (labelled as **watts_strogatz** in our tables) are constructed based on the model proposed by Watts and Strogatz [34]; the sampled graphs can be "highly clustered, like regular lattices, yet have small characteristic path lengths" [34]. We consider two possible sizes $|V| = 300,\ |E| = 600$ and $|V| = 500,\ |E| = 1000$; the number of direct neighbors in the original ring topology and the edge rewiring probability are set to 4 and 0.21, respectively.
- *Barabási-Albert graphs* (labelled as **barabasi_albert** in our tables) are constructed according to Barabási-Albert preferential attachment mechanism [2], which is known to result in scale-free networks. We consider two possible graph sizes $|V| = 300,\ |E| = 596$ and $|V| = 500,\ |E| = 996$; the number of edges attached to any new vertex is set to 2.

The parameters for the randomly generated networks are selected to ensure that their edge densities are similar.

**Results and discussion**. The results for the considered real-life networks are presented in Tables 1, 2 and 3. In particular, for the $\Delta$-**AA** approach recall from Sect. 2.2, that solving MIP **F1**$(k)$ provides an upper bound, $\Delta_k(S^\Delta)$, on the quality of approximating $C^*$ by $C(S_k^*)$. Hence, to have a fair comparison for $\Delta$-**AA** we report the results provided by **F1**$(k)$ for each $k$. Then we set

$$\varepsilon = \Delta_k(S^\Delta) \cdot \left(1 + \sqrt{\frac{\ln \frac{2}{\delta}}{\ln \binom{n}{s} + \ln \frac{2}{\delta}}}\right)^{-1}$$

to compute $r := r^{(1)}$ and $r := r^{(2)}$ for **PAA**, which is executed 15 times. Loosely speaking, for $\Delta := \Delta_k(S^\Delta)$ and $r := r^{(1)}$ both algorithms achieve similar approximation quality guarantees; one should compare Proposition 1 and Theorem 1.

For the considered graphs the number of shortest paths of length at most $k$ grows rapidly despite the graphs' sparsity. Hence, as $k$ increases the running time performance of $\Delta$-**AA**, i.e., the time needed for solving **F1**$(k)$, deteriorates significantly. Similarly, as the values of $r^{(1)}$ and $r^{(2)}$ increase (and the value of $\varepsilon$ decreases), the running time performance of **PAA** becomes less competitive, in particular, when comparing to Top-$s$. This observation is not surprising given that the required cardinality, $r = |X|$, and the corresponding number of variables **F2**$(X)$, become relatively large.

As expected the quality of $C(S_k^*)$ computed by $\Delta$-**AA** is better for graphs with smaller diameters. More importantly, by comparing the values of $C(S^\Delta)$ and $C(S_X^*)$ we observe that **PAA** provides either the same solution quality or outperforms $\Delta$-**AA**. When comparing these results to those provided by $C(s$-Top$)$, it is clear that the main advantage of the Top-$s$ approach is its fast running time. However, the quality of the Top-$s$ solutions is typically worse than those provided by $\Delta$-**AA** and **PAA** for reasonably large values of $k$ and small values of $\varepsilon$, respectively, in particular, for graphs with larger diameters.

Furthermore, it is important to highlight the following two observations from our experiments. First, using the sample size $r := r^{(2)}$ for **PAA** achieves the desired accuracy without

the computational burden observed for $r := r^{(1)}$. This result suggests that the sample size outlined in Proposition 2 from [29] can be used to compute reasonably good solutions when solving (3). Establishing this result theoretically with a stronger result than (27) could be an interesting avenue for future research. Secondly, the running time gains from using **PAA** when comparing to the exact MIP method, increase with respect to the network size. In particular, note that **ieeebus_118** with $s = 5$ is the only instance where solving the full MIP is faster than our proposed approach.

The results for the randomly generated networks (Watts-Strogatz and Barabási-Albert graphs) are given in Table 4. Our observations for these networks are fairly consistent with those reported for the considered real-life graphs. That is, **PAA** is very good as long as $\varepsilon$ is relatively large i.e., the value of $r$ is relatively small.

To summarize our discussion, we conclude from Tables 1, 2, 3 and 4 that as the value of $\varepsilon$ decreases the corresponding required cardinality of $X$, in particular, the value of $r^{(1)}$, may become relatively large for **F2**$(X)$ and **PAA**. In fact, the number of paths, $r^{(1)}$, that need to be generated in **PAA** for sufficiently small values of $\varepsilon$ may be even larger than the number of paths generated by $\Delta$-**AA**. This observation provides the main motivation for our second approach outlined in the discussion next.

## 4 SAA-based approach

The key idea behind our method considered in Sect. 3 is to solve an MIP model given by **F2**$(X)$ using a sufficiently large number of shortest paths in a multiset $X$. Naturally, this approach is favorable as long as the cardinality of $X$ is not too large. However, in our computational experiments (see Sect. 3.4) we observe that if the required worst-case solution quality guarantee is sufficiently small (recall Theorem 1), then the scalability of the approach may deteriorate due to a rather large number of shortest paths (i.e., the cardinality of $X$) that need to be sampled for **F2**$(X)$.

In this section we outline an alternative solution scheme based on the SAA-based ideas, and instead of solving one "big" MIP with a large number of shortest paths, we solve a series of substantially smaller MIPs. Loosely speaking, this approach aims at exploiting the Central Limit Theorem and the fact that $C^* \leq \mathbb{E}[C^*_X]$ (see the discussion below). To achieve the required performance bound, we iteratively increase the cardinality of $X$ that is, the sizes of the corresponding MIP models, **F2**$(X)$, solved in each iteration until some required solution quality is achieved. We note that the quality of the candidate solution is verified without solving an MIP. Finally, we refer the reader to [22, 31] and the references therein, which provides a detailed discussion on sample average approximation (SAA) in stochastic optimization.

### 4.1 Algorithm description

Recall that $C^*_X$ denotes the optimal objective function value of **F2**$(X)$ for a given $X$. Hence, for any $S \subset V$ we have $C(S, X) \leq C^*_X$. By taking expectations in both sides of this inequality and setting $S := S^*$, we obtain the following *sandwich*-like result:

$$\mathbb{E}[C(S^*, \mathbf{X})] = C^* \leq \mathbb{E}[C^*_{\mathbf{X}}], \tag{33}$$

where the equality follows from (17).

**Table 1** Comparing Δ-**AA**, **PAA** and Top-$s$ for **ieeebus_118** and **USAir97** with $s \in \{5, 10\}$

| Δ-**AA** (Algorithm 1) | | | | | | **PAA** (Algorithm 3) | | | | | | | | | | | Top-$s$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | | |
| $k$ | $\Delta_k(S^\Delta)$ | $\text{UB}_k$ | $C(S^\Delta)$ | #paths | Time | $\varepsilon$ | Avg $C^*_X$ | | $C(S^*_X)$ | | | | Avg time | | #paths | | $C$(Top-$s$) | |
| **ieeebus_118**: $diam(G) = 14$, Number of shortest paths= 15600, Brandes algorithm Time = 0.75 | | | | | | | | | | | | | | | | | | |
| $s = 5$: $C^* = 0.714$, MIP Time= 29.4, Top-$s$ max Avg Time= 1.4 | | | | | | | | | Avg | | max | | | | | | Avg | max |
| 2 | 0.48 | 1.31 | 0.82 | 440 | 0.03 | 0.48 | 1.04 | 1.21 | 0.97 | 0.89 | 1.0 | 0.95 | 0.06 | 0.01 | 265 | 48 | 0.79 | 0.85 |
| 3 | 0.29 | 1.23 | 0.94 | 1246 | 0.11 | 0.29 | 0.99 | 1.08 | 0.99 | 0.94 | 1.0 | 0.98 | 0.24 | 0.03 | 717 | 130 | 0.86 | 0.93 |
| 4 | 0.23 | 1.17 | 0.94 | 2399 | 0.39 | 0.23 | 1.0 | 1.06 | 0.99 | 0.98 | 1.0 | 1.0 | 0.39 | 0.03 | 1200 | 217 | 0.88 | 0.92 |
| 5 | 0.17 | 1.11 | 0.94 | 3830 | 1.07 | 0.17 | 1.0 | 1.03 | 1.0 | 0.97 | 1.0 | 1.0 | 1.12 | 0.07 | 2168 | 392 | 0.87 | 0.92 |
| 6 | 0.06 | 1.06 | 1.0 | 5541 | 1.51 | $0.07^\dagger$ | 0.99 | 1.0 | 1.0 | 0.99 | 1.0 | 1.0 | 37.1 | 0.43 | 12523 | 2261 | 0.86 | 0.89 |
| $s = 10$: $C^* = 0.759$, MIP Time= 180.6, Top-$s$ max Avg Time= 0.044 | | | | | | | | | Avg | | max | | | | | | Avg | max |
| 2 | 0.33 | 1.24 | 0.9 | 440 | 0.03 | 0.33 | 1.02 | 1.18 | 0.98 | 0.89 | 0.99 | 0.91 | 0.1 | 0.01 | 642 | 89 | 0.87 | 0.88 |
| 3 | 0.25 | 1.19 | 0.94 | 1246 | 0.13 | 0.25 | 1.02 | 1.15 | 0.99 | 0.92 | 1.0 | 0.99 | 0.37 | 0.02 | 1189 | 164 | 0.87 | 0.9 |
| 4 | 0.16 | 1.14 | 0.98 | 2399 | 0.58 | 0.16 | 1.01 | 1.08 | 0.99 | 0.95 | 1.0 | 0.97 | 2.4 | 0.03 | 2806 | 386 | 0.88 | 0.92 |
| 5 | 0.12 | 1.1 | 0.99 | 3830 | 2.5 | 0.12 | 1.0 | 1.03 | 1.0 | 0.97 | 1.0 | 0.99 | 13.8 | 0.08 | 5242 | 721 | 0.87 | 0.88 |
| 6 | 0.07 | 1.07 | 1.0 | 5541 | 3.6 | 0.07 | 0.99 | 1.01 | 1.0 | 0.99 | 1.0 | 1.0 | 79.9 | 0.26 | 12888 | 1772 | 0.88 | 0.92 |
| 7 | 0.05 | 1.05 | 1.0 | 7414 | 11.3 | $0.07^\dagger$ | 0.99 | 1.01 | 1.0 | 0.99 | 1.0 | 1.0 | 143.3 | 0.5 | 16442 | 2261 | 0.88 | 0.88 |

**Table 1** continued

| Δ-**AA** (Algorithm 1) | | | | | | **PAA** (Algorithm 3) | | | | | | | | | | | Top-$s$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | | |
| $k$ | $\Delta_k(S^\Delta)$ | $\text{UB}_k$ | $C(S^\Delta)$ | #paths | Time | $\varepsilon$ | Avg $C^*_X$ | | $C(S^*_X)$ | | | | Avg time | | #paths | | $C$(Top-$s$) | |
| **USAir97**: $diam(G) = 6$, Number of shortest paths= 304356, Brandes algorithm Time = 5.5 | | | | | | | | | | | | | | | | | | |
| $s = 5$: $C^* = 0.583$, MIP Time= 793.7, Top-$s$ max Avg Time= 0.111 | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | Avg | | max | | | | | | Avg | max |
| 2 | 0.44 | 1.36 | 0.92 | 55646 | 60.5 | 0.44 | 1.02 | 1.16 | 1.0 | 0.9 | 1.0 | 1.0 | 0.04 | 0.01 | 535 | 76 | 0.93 | 1.0 |
| 3 | 0.12 | 1.06 | 0.95 | 219491 | 1267.9 | 0.12 | 0.99 | 1.01 | 1.0 | 1.0 | 1.0 | 1.0 | 2.5 | 0.06 | 7806 | 1099 | 0.99 | 1.0 |
| 4 | 0.006 | 1.01 | 1.0 | 285545 | 2068.8 | 0.09† | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 11.1 | 0.14 | 14103 | 1985 | 1.0 | 1.0 |
| $s = 10$: $C^* = 0.779$, MIP Time= 664.5, Top-$s$ max Avg Time= 0.116 | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | Avg | | max | | | | | | Avg | max |
| 2 | 0.13 | 1.09 | 0.96 | 55646 | 51.5 | 0.13 | 0.99 | 1.0 | 1.0 | 0.98 | 1.0 | 1.0 | 0.77 | 0.02 | 4596 | 470 | 0.93 | 0.98 |
| 3 | 0.009 | 1.01 | 1.0 | 219491 | 1236.3 | 0.06† | 0.99 | 1.0 | 1.0 | 0.99 | 1.0 | 1.0 | 11.0 | 0.11 | 19433 | 1985 | 0.95 | 0.97 |

All running times are in seconds. All values of betweenness centrality $C$ (except $C^*$ itself) and their approximations as well as the error bounds (i.e., $\Delta_k(S^\Delta)$, $UB_k$ and $\varepsilon$) are scaled with respect to $C^*$. The value of $C^*$ is computed using an MIP solver with **F1**($k$), where $k = diam(G)$. For Δ-**AA** we report the MIP running time **F1**($k$) for each $k$. **PAA** and Top-$s$ are executed 15 times. The average running time of these runs is reported for **PAA**; for Top-$s$ we report only the maximum average running times over the corresponding rows. To avoid out of memory error for the MIP solver, we set $\varepsilon = 0.05$ for the rows marked with †

**Table 2** Comparing Δ-**AA**, **PAA** and Top-$s$ for **fb-pages-food** and **494_bus** with $s \in \{5, 10\}$; see the captions of Table 1 for the details of the computational settings and the parameters used

| Δ-**AA** (Algorithm 1) | | | | | | **PAA** (Algorithm 3) | | | | | | | | | | | Top-$s$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | | |
| $k$ | $\Delta_k(S^\Delta)$ | $\text{UB}_k$ | $C(S^\Delta)$ | #paths | Time | $\varepsilon$ | Avg $C^*_X$ | | $C(S^*_X)$ | | | | Avg time | | #paths | | $C$(Top-$s$) | |
| **fb-pages-food**: $diam(G) = 17$, Number of shortest paths= 1724951, Brandes algorithm Time = 70.1 | | | | | | | | | | | | | | | | | | |
| $s = 5$: $C^* = 0.653$, MIP Time= 39299.1, Top-$s$ max Avg Time= 0.128 | | | | | | | | | Avg | | max | | | | | | Avg | max |
| 2 | 0.58 | 1.45 | 0.87 | 30743 | 10.7 | 0.58 | 1.02 | 1.14 | 0.99 | 0.82 | 1.0 | 0.98 | 0.03 | 0.01 | 266 | 42 | 0.82 | 0.88 |
| 3 | 0.48 | 1.35 | 0.87 | 150704 | 331.3 | 0.48 | 1.05 | 1.17 | 0.99 | 0.86 | 1.0 | 0.94 | 0.04 | 0.01 | 385 | 60 | 0.88 | 0.93 |
| 4 | 0.29 | 1.24 | 0.95 | 348313 | 1442.9 | 0.29 | 1.0 | 1.04 | 0.99 | 0.93 | 1.0 | 0.98 | 0.12 | 0.02 | 1032 | 160 | 0.91 | 0.92 |
| 5 | 0.2 | 1.15 | 0.95 | 652326 | 6325.4 | 0.2 | 1.0 | 1.03 | 1.0 | 0.97 | 1.0 | 1.0 | 0.32 | 0.02 | 2139 | 331 | 0.91 | 1.0 |
| 6 | 0.08 | 1.08 | 1.0 | 945479 | 8943.2 | 0.08 | 1.0 | 1.0 | 1.0 | 0.99 | 1.0 | 1.0 | 5.3 | 0.12 | 13373 | 2065 | 0.93 | 1.0 |
| 7 | 0.04 | 1.04 | 1.0 | 1189770 | 9488.7 | 0.08$^\dagger$ | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 6.6 | 0.14 | 15030 | 2321 | 0.93 | 1.0 |
| $s = 10$: $C^* = 0.771$, MIP Time= 31622.6, Top-$s$ max Avg Time= 0.126 | | | | | | | | | Avg | | max | | | | | | Avg | max |
| 2 | 0.36 | 1.24 | 0.88 | 30743 | 11.3 | 0.36 | 1.01 | 1.19 | 0.99 | 0.83 | 1.0 | 0.9 | 0.07 | 0.01 | 688 | 76 | 0.77 | 0.88 |
| 3 | 0.25 | 1.17 | 0.92 | 150704 | 339.5 | 0.25 | 1.01 | 1.1 | 0.99 | 0.91 | 1.0 | 0.96 | 0.15 | 0.02 | 1378 | 151 | 0.86 | 0.91 |
| 4 | 0.19 | 1.11 | 0.92 | 348313 | 1056.6 | 0.19 | 1.01 | 1.06 | 1.0 | 0.96 | 1.0 | 0.99 | 0.33 | 0.02 | 2460 | 270 | 0.87 | 0.91 |
| 5 | 0.07 | 1.06 | 0.99 | 652326 | 8623.5 | 0.07 | 1.0 | 1.01 | 1.0 | 1.0 | 1.0 | 1.0 | 16.4 | 0.13 | 21073 | 2310 | 0.92 | 0.95 |
| 6 | 0.03 | 1.03 | 1.0 | 945479 | 19591.7 | 0.06$^\dagger$ | 1.0 | 1.01 | 1.0 | 0.99 | 1.0 | 1.0 | 15.7 | 0.14 | 21171 | 2321 | 0.91 | 0.95 |

**Table 2** continued

| Δ-**AA** (Algorithm 1) | | | | | | **PAA** (Algorithm 3) | | | | | | | | | | | Top-$s$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | | |
| $k$ | $\Delta_k(S^{\Delta})$ | $\text{UB}_k$ | $C(S^{\Delta})$ | #paths | Time | $\varepsilon$ | Avg $C^*_X$ | | $C(S^*_X)$ | | | | Avg time | | #paths | | $C$(Top-$s$) | |
| **494_bus**: $diam(G) = 26$, Number of shortest paths= 278828, Brandes algorithm Time = 22.5 | | | | | | | | | | | | | | | | | | |
| $s = 5$: $C^* = 0.737$, MIP Time= 6719.4, Top-$s$ max Avg Time= 0.042 | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | Avg | | max | | | | | | Avg | max |
| 2 | 0.91 | 1.34 | 0.43 | 1238 | 0.08 | 0.91 | 1.09 | 1.36 | 0.96 | 0.51 | 0.99 | 0.73 | 0.02 | 0.01 | 83 | 14 | 0.62 | 0.8 |
| 3 | 0.55 | 1.32 | 0.77 | 3474 | 0.37 | 0.55 | 1.06 | 1.31 | 0.98 | 0.83 | 0.99 | 0.94 | 0.04 | 0.01 | 222 | 37 | 0.74 | 0.89 |
| 4 | 0.33 | 1.3 | 0.96 | 7248 | 1.19 | 0.33 | 1.01 | 1.1 | 0.99 | 0.9 | 1.0 | 0.98 | 0.13 | 0.02 | 616 | 103 | 0.72 | 0.98 |
| 5 | 0.3 | 1.27 | 0.96 | 13113 | 3.6 | 0.3 | 1.0 | 1.1 | 0.99 | 0.94 | 1.0 | 0.98 | 0.2 | 0.03 | 742 | 124 | 0.75 | 0.92 |
| 6 | 0.27 | 1.23 | 0.96 | 21998 | 13.3 | 0.27 | 1.01 | 1.09 | 0.99 | 0.94 | 1.0 | 0.98 | 0.21 | 0.03 | 946 | 158 | 0.79 | 0.93 |
| 7 | 0.2 | 1.19 | 0.99 | 34844 | 41.8 | 0.2 | 1.0 | 1.04 | 0.99 | 0.97 | 1.0 | 1.0 | 0.47 | 0.03 | 1684 | 281 | 0.84 | 0.93 |
| 8 | 0.16 | 1.15 | 0.99 | 52007 | 98.8 | 0.16 | 1.0 | 1.02 | 1.0 | 0.97 | 1.0 | 1.0 | 1.14 | 0.04 | 2563 | 428 | 0.82 | 0.93 |
| 9 | 0.13 | 1.12 | 0.99 | 72995 | 208.3 | 0.13 | 0.99 | 1.02 | 1.0 | 0.98 | 1.0 | 1.0 | 2.4 | 0.06 | 4185 | 698 | 0.8 | 0.93 |
| 10 | 0.09 | 1.09 | 0.99 | 97006 | 349.8 | 0.09 | 0.99 | 0.99 | 1.0 | 0.98 | 1.0 | 1.0 | 6.4 | 0.13 | 7515 | 1253 | 0.74 | 0.93 |
| 11 | 0.07 | 1.06 | 0.99 | 123003 | 622.4 | 0.07 | 1.0 | 1.01 | 1.0 | 0.99 | 1.0 | 1.0 | 28.3 | 0.26 | 14694 | 2450 | 0.8 | 0.93 |
| $s = 10$: $C^* = 0.853$, MIP Time= 913.4, Top-$s$ max Avg Time= 0.086 | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | Avg | | max | | | | | | Avg | max |
| 2 | 0.34 | 1.16 | 0.81 | 1238 | 0.09 | 0.34 | 1.01 | 1.15 | 0.98 | 0.88 | 0.99 | 0.91 | 0.16 | 0.02 | 598 | 72 | 0.78 | 0.89 |
| 3 | 0.21 | 1.14 | 0.93 | 3474 | 0.47 | 0.21 | 1.0 | 1.08 | 0.99 | 0.92 | 1.0 | 0.95 | 0.34 | 0.02 | 1528 | 183 | 0.86 | 0.91 |
| 4 | 0.15 | 1.13 | 0.98 | 7248 | 1.56 | 0.15 | 1.0 | 1.04 | 1.0 | 0.95 | 1.0 | 0.98 | 0.79 | 0.03 | 3159 | 377 | 0.87 | 0.91 |
| 5 | 0.13 | 1.11 | 0.98 | 13113 | 5.0 | 0.13 | 1.0 | 1.03 | 1.0 | 0.96 | 1.0 | 0.99 | 1.35 | 0.04 | 4103 | 490 | 0.87 | 0.9 |
| 6 | 0.11 | 1.09 | 0.98 | 21998 | 21.0 | 0.11 | 1.0 | 1.03 | 1.0 | 0.97 | 1.0 | 0.99 | 1.84 | 0.06 | 5786 | 690 | 0.86 | 0.92 |
| 7 | 0.07 | 1.07 | 1.0 | 34844 | 49.2 | 0.07 | 1.0 | 1.0 | 1.0 | 0.99 | 1.0 | 1.0 | 11.9 | 0.16 | 14444 | 1722 | 0.87 | 0.91 |
| 8 | 0.05 | 1.05 | 1.0 | 52007 | 112.4 | 0.06$^{\dagger}$ | 1.0 | 1.0 | 1.0 | 0.99 | 1.0 | 1.0 | 28.0 | 0.21 | 20543 | 2450 | 0.86 | 0.9 |

**Table 3** Comparing Δ-**AA**, **PAA** and Top-$s$ for **662_bus** with $s \in \{5, 10\}$; see the captions of Table 1 for the details of the computational settings and the parameters used

| Δ-**AA** (Algorithm 1) | | | | | | **PAA** (Algorithm 3) | | | | | | | | | | | Top-$s$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | |
| $k$ | $\Delta_k(S^\Delta)$ | $\mathrm{UB}_k$ | $C(S^\Delta)$ | #paths | Time | $\varepsilon$ | Avg $C^*_X$ | | $C(S^*_X)$ | | | | Avg time | | #paths | | $C$(Top-$s$) |
| **662_bus**: $diam(G) = 25$, Number of shortest paths= 534654, Brandes algorithm Time = 41.0 | | | | | | | | | | | | | | | | | |
| $s = 5$: $C^* = 0.609$, MIP Time= 47048.3, Top-$s$ max Avg Time= 0.063 | | | | | | | | | | | | | | | | | |
| | | | | | | | | | Avg | | max | | | | | | Avg | max |
| 2 | 1.01 | 1.62 | 0.61 | 2126 | 0.14 | 1.01 | 1.08 | 1.64 | 0.91 | 0.63 | 0.95 | 0.72 | 0.02 | 0.01 | 101 | 17 | 0.71 | 0.89 |
| 3 | 0.92 | 1.6 | 0.68 | 6125 | 0.64 | 0.92 | 1.06 | 1.46 | 0.91 | 0.7 | 0.97 | 0.84 | 0.03 | 0.01 | 122 | 20 | 0.68 | 0.79 |
| 4 | 0.88 | 1.56 | 0.68 | 13020 | 2.3 | 0.88 | 1.05 | 1.57 | 0.93 | 0.68 | 0.96 | 0.82 | 0.04 | 0.01 | 131 | 22 | 0.77 | 0.8 |
| 5 | 0.72 | 1.52 | 0.79 | 24103 | 6.4 | 0.72 | 1.07 | 1.51 | 0.98 | 0.78 | 1.0 | 0.85 | 0.05 | 0.01 | 194 | 32 | 0.7 | 0.83 |
| 6 | 0.56 | 1.46 | 0.9 | 41254 | 31.6 | 0.56 | 1.02 | 1.24 | 0.97 | 0.79 | 0.99 | 0.85 | 0.08 | 0.01 | 322 | 52 | 0.77 | 0.84 |
| 7 | 0.5 | 1.4 | 0.9 | 66507 | 134.6 | 0.5 | 1.01 | 1.2 | 0.97 | 0.86 | 0.99 | 0.94 | 0.1 | 0.02 | 412 | 67 | 0.79 | 0.86 |
| 8 | 0.4 | 1.32 | 0.93 | 101885 | 457.0 | 0.4 | 0.99 | 1.12 | 0.98 | 0.86 | 1.0 | 0.91 | 0.2 | 0.02 | 641 | 104 | 0.8 | 0.88 |
| 9 | 0.32 | 1.25 | 0.93 | 148367 | 5014.3 | 0.32 | 1.01 | 1.12 | 0.99 | 0.94 | 0.99 | 0.99 | 0.35 | 0.02 | 966 | 156 | 0.82 | 0.9 |
| 10 | 0.19 | 1.18 | 1.0 | 205100 | 9599.3 | 0.19 | 1.01 | 1.04 | 0.99 | 0.97 | 1.0 | 0.99 | 1.66 | 0.06 | 2914 | 470 | 0.83 | 0.88 |
| 11 | 0.13 | 1.13 | 1.0 | 266321 | 15049.8 | 0.13 | 1.01 | 1.03 | 1.0 | 0.98 | 1.0 | 1.0 | 4.2 | 0.13 | 5840 | 941 | 0.83 | 0.88 |
| 12 | 0.09 | 1.08 | 0.99 | 325561 | 30408.9 | 0.09 | 1.0 | 1.01 | 1.0 | 0.99 | 1.0 | 1.0 | 22.6 | 0.35 | 12473 | 2010 | 0.85 | 0.88 |
| 13 | 0.06 | 1.05 | 0.99 | 377713 | 42850.8 | $0.08^\dagger$ | 1.0 | 1.01 | 1.0 | 0.99 | 1.0 | 1.0 | 33.5 | 0.48 | 15126 | 2438 | 0.85 | 0.85 |

**Table 3** continued

| Δ-**AA** (Algorithm 1) | | | | | | **PAA** (Algorithm 3) | | | | | | | | | | | Top-$s$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | | |
| $k$ | $\Delta_k(S^\Delta)$ | $\mathrm{UB}_k$ | $C(S^\Delta)$ | #paths | Time | $\varepsilon$ | Avg $C^*_X$ | | $C(S^*_X)$ | | | | Avg time | | #paths | | $C$(Top-$s$) | |
| $s = 10$: $C^* = 0.77$, MIP Time= 129571.0, Top-$s$ max Avg Time= 0.106 | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | Avg | | max | | | | | | Avg | max |
| 2 | 0.65 | 1.28 | 0.63 | 2126 | 0.15 | 0.65 | 1.08 | 1.27 | 0.94 | 0.67 | 0.96 | 0.79 | 0.04 | 0.01 | 211 | 25 | 0.79 | 0.83 |
| 3 | 0.57 | 1.27 | 0.7 | 6125 | 0.91 | 0.57 | 1.04 | 1.25 | 0.96 | 0.6 | 0.97 | 0.75 | 0.07 | 0.01 | 282 | 33 | 0.69 | 0.79 |
| 4 | 0.45 | 1.24 | 0.8 | 13020 | 2.7 | 0.45 | 1.02 | 1.29 | 0.97 | 0.74 | 0.98 | 0.83 | 0.1 | 0.01 | 453 | 52 | 0.79 | 0.83 |
| 5 | 0.32 | 1.22 | 0.9 | 24103 | 11.1 | 0.32 | 1.01 | 1.16 | 0.99 | 0.85 | 1.0 | 0.91 | 0.33 | 0.02 | 898 | 103 | 0.78 | 0.85 |
| 6 | 0.26 | 1.18 | 0.92 | 41254 | 47.0 | 0.26 | 1.01 | 1.14 | 0.99 | 0.88 | 1.0 | 0.95 | 0.53 | 0.02 | 1356 | 155 | 0.8 | 0.83 |
| 7 | 0.17 | 1.15 | 0.98 | 66507 | 204.6 | 0.17 | 1.0 | 1.07 | 1.0 | 0.95 | 1.0 | 0.99 | 2.2 | 0.04 | 3206 | 366 | 0.8 | 0.85 |
| 8 | 0.13 | 1.11 | 0.98 | 101885 | 509.9 | 0.13 | 1.0 | 1.05 | 1.0 | 0.97 | 1.0 | 1.0 | 5.8 | 0.06 | 5006 | 572 | 0.81 | 0.86 |
| 9 | 0.09 | 1.08 | 1.0 | 148367 | 12418.5 | 0.09 | 1.0 | 1.02 | 1.0 | 0.99 | 1.0 | 1.0 | 33.4 | 0.23 | 12310 | 1406 | 0.81 | 0.84 |
| 10 | 0.06 | 1.05 | 1.0 | 205100 | 26750.0 | 0.06 | 1.0 | 1.01 | 1.0 | 0.99 | 1.0 | 1.0 | 127.4 | 0.45 | 21351 | 2438 | 0.82 | 0.85 |

**Table 4** Comparing Δ-**AA**, **PAA** and Top-$s$ for 10 randomly generated networks: Watts-Strogatz and Barabási-Albert graphs with $|V| = 300$ and $|V| = 500$

| Δ-**AA** (Algorithm 1) | | | | | | **PAA** (Algorithm 3) | | | | | | | | | | | Top-$s$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | | |
| $k$ | $\Delta_k(S^\Delta)$ | $\text{UB}_k$ | $C(S^\Delta)$ | #paths | Time | $\varepsilon$ | Avg $C^*_X$ | | $C(S^*_X)$ | | | | Avg time | | #paths | | $C$(Top-$s$) | |
| **watts_strogatz_300**: $diam(G) = 5.0$, Number of shortest paths= 178145.3, Brandes algorithm Time = 3.8 | | | | | | | | | | | | | | | | | | |
| $s = 10$: $C^* = 0.176$, MIP Time= 610.0, Top-$s$ max Avg Time= 0.284 | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | Avg | | max | | | | | | Avg | max |
| 2 | 3.8 | 4.8 | 0.96 | 9277 | 0.78 | 3.8 | 1.83 | 5.5 | 0.66 | 0.44 | 0.77 | 0.53 | 0.04 | 0.01 | 109 | 11 | 0.37 | 0.48 |
| 3 | 1.4 | 2.4 | 0.99 | 57039 | 26.5 | 1.4 | 1.15 | 2.4 | 0.88 | 0.59 | 0.94 | 0.7 | 0.13 | 0.03 | 806 | 81 | 0.49 | 0.64 |
| 4 | 0.03 | 1.03 | 1.0 | 168369 | 536.5 | 0.29† | 1.0 | 1.1 | 0.99 | 0.91 | 1.0 | 0.97 | 18.8 | 0.21 | 19148 | 1921 | 0.84 | 0.92 |
| **watts_strogatz_500**: $diam(G) = 5.4$, Number of shortest paths= 519558.7, Brandes algorithm Time = 12.3 | | | | | | | | | | | | | | | | | | |
| $s = 10$: $C^* = 0.128$, MIP Time= 1196.5, Top-$s$ max Avg Time= 0.216 | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | Avg | | max | | | | | | Avg | max |
| 2 | 6.1 | 7.0 | 0.95 | 15319 | 1.29 | 6.1 | 2.4 | 7.6 | 0.61 | 0.41 | 0.72 | 0.51 | 0.03 | 0.01 | 86 | 8 | 0.36 | 0.47 |
| 3 | 3.4 | 4.4 | 0.99 | 98414 | 29.3 | 3.4 | 1.65 | 5.5 | 0.72 | 0.49 | 0.81 | 0.6 | 0.06 | 0.01 | 266 | 25 | 0.4 | 0.5 |
| 4 | 0.28 | 1.28 | 1.0 | 392983 | 414.4 | 0.39† | 1.01 | 1.17 | 0.99 | 0.87 | 1.0 | 0.94 | 10.6 | 0.19 | 20577 | 1946 | 0.8 | 0.9 |

**Table 4** continued

| $\Delta$-**AA** (Algorithm 1) | | | | | | **PAA** (Algorithm 3) | | | | | | | | | | | | Top-$s$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | | |
| $k$ | $\Delta_k(S^\Delta)$ | $\text{UB}_k$ | $C(S^\Delta)$ | #paths | Time | $\varepsilon$ | Avg $C^*_X$ | | $C(S^*_X)$ | | | | Avg time | | #paths | | $C$(Top-$s$) | |
| **barabasi_albert_300**: $diam(G) = 4.0$, Number of shortest paths= 195814.5, Brandes algorithm Time = 3.6 | | | | | | | | | | | | | | | | | | | |
| $s = 10$: $C^* = 0.535$, MIP Time= 865.8, Top-$s$ max Avg Time= 0.608 | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | Avg | | max | | | | | | Avg | max |
| 2 | 0.43 | 1.43 | 1.0 | 22066 | 9.9 | 0.43 | 1.01 | 1.17 | 0.99 | 0.82 | 1.0 | 0.91 | 0.11 | 0.02 | 938 | 90 | 0.84 | 0.94 |
| 3 | 0.02 | 1.02 | 1.0 | 164710 | 594.9 | 0.09$^\dagger$ | 1.0 | 1.01 | 1.0 | 0.99 | 1.0 | 1.0 | 16.4 | 0.14 | 19148 | 1838 | 0.99 | 1.0 |
| **barabasi_albert_500**: $diam(G) = 4.3$, Number of shortest paths= 583527.4, Brandes algorithm Time = 11.0 | | | | | | | | | | | | | | | | | | | |
| $s = 10$: $C^* = 0.513$, MIP Time= 5929.3, Top-$s$ max Avg Time= 0.152 | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | Avg | | max | | | | | | Avg | max |
| 2 | 0.61 | 1.6 | 1.0 | 41133 | 23.0 | 0.61 | 1.02 | 1.31 | 0.98 | 0.69 | 1.0 | 0.82 | 0.07 | 0.02 | 542 | 49 | 0.71 | 0.84 |
| 3 | 0.05 | 1.05 | 1.0 | 395942 | 1476 | 0.1$^\dagger$ | 1.0 | 1.01 | 1.0 | 0.99 | 1.0 | 1.0 | 17.8 | 0.15 | 20577 | 1862 | 0.99 | 1.0 |

All values of betweenness centrality and their approximations are averaged over 10 instances; see the captions of Table 1 for the details of the computational settings and the parameters used

**Table 5** Comparing **PAA** and **SAA** for the minimum value of $\varepsilon$ achieved in Tables 1, 2 and 3 and $\delta = 10^{-3}$. All values of betweenness centrality $C$ (except for $C^*$ itself) and their approximations are scaled with respect to $C^*$

| **SAA** (Algorithm 4) | | | | | | | | | **PAA** (Algorithm 3) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ |
| $\beta$ | # iter | $\varepsilon_{\text{SAA}}$ | $\bar{\mu}_M$ | $\widetilde{C}$ | max $\lvert X_\ell \rvert$ | $\widetilde{r} = \lvert\widetilde{X}\rvert$ | Time | | Avg $C^*_X$ | | $C(S^*_X)$ | | | | Avg. time | | #paths | |
| | | | | | | | Ev. | Sol. | | | Avg | | max | | | | | |
| **ieeebus_118**: $\varepsilon = 0.07$, $\delta = 0.001$ | | | | | | | | | | | | | | | | | | |
| $s = 5$, $C(\text{Top-}s, \widetilde{X}) = 0.907$, Top-$s$ running time= 0.02, $C(S^*) = 0.714$ | | | | | | | | | | | | | | | | | | |
| 0 | 8 | 0.07 | 1.03 | 1.0 | 249 | 15600 | 10.8 | 2.2 | 0.99 | 1.0 | 1.0 | 0.99 | 1.0 | 1.0 | 27.5 | 0.38 | 12523 | 2261 |
| 1 | 7 | 0.07 | 1.02 | 0.99 | 343 | 2062 | **0.11** | **4.1** | | | | | | | | | | |
| **ieeebus_118**: $\varepsilon = 0.066$, $\delta = 0.001$ | | | | | | | | | | | | | | | | | | |
| $s = 10$, $C(\text{Top-}s, \widetilde{X}) = 0.859$, Top-$s$ running time= 0.04, $C(S^*) = 0.759$ | | | | | | | | | | | | | | | | | | |
| 0 | 9 | 0.04 | 1.02 | 0.99 | 970 | 15600 | 12.5 | 8.1 | 0.99 | 1.01 | 1.0 | 0.99 | 1.0 | 1.0 | 143.3 | 0.5 | 16442 | 2261 |
| 1 | 9 | 0.06 | 1.03 | 1.02 | 602 | 2062 | **0.04** | **7.2** | | | | | | | | | | |
| **USAir97**: $\varepsilon = 0.086$, $\delta = 0.001$ | | | | | | | | | | | | | | | | | | |
| $s = 5$, $C(\text{Top-}s, \widetilde{X}) = 0.97$, Top-$s$ running time= 0.12, $C(S^*) = 0.583$ | | | | | | | | | | | | | | | | | | |
| 0 | 7 | 0.08 | 1.03 | 1.0 | 225 | 304356 | 115.9 | 4.1 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 11.1 | 0.14 | 14103 | 1985 |
| 1 | 6 | 0.06 | 1.01 | 1.02 | 218 | 2062 | **0.03** | **3.5** | | | | | | | | | | |
| **USAir97**: $\varepsilon = 0.064$, $\delta = 0.001$ | | | | | | | | | | | | | | | | | | |
| $s = 10$, $C(\text{Top-}s, \widetilde{X}) = 0.96$, Top-$s$ running time= 0.13, $C(S^*) = 0.779$ | | | | | | | | | | | | | | | | | | |
| 0 | 6 | 0.06 | 1.02 | 1.0 | 132 | 304356 | 108.1 | 2.9 | 0.99 | 1.0 | 1.0 | 0.99 | 1.0 | 1.0 | 11.0 | 0.11 | 19433 | 1985 |
| 1 | 5 | 0.04 | 0.99 | 1.01 | 307 | 2062 | **0.03** | **3.3** | | | | | | | | | | |

**Table 5** continued

| **SAA** (Algorithm 4) | | | | | | | | | **PAA** (Algorithm 3) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ | $r^{(1)}$ | $r^{(2)}$ |
| **494_bus**: $\varepsilon = 0.068$, $\delta = 0.001$ | | | | | | | | | | | | | | | | | | |
| $s = 5$, $C(\text{Top-}s, \widetilde{X}) = 0.945$, Top-$s$ running time= 0.05, $C(S^*) = 0.737$ | | | | | | | | | | | | | | | | | | |
| 1 | 7 | 0.03 | 0.98 | 1.0 | 446 | 2062 | **0.05** | **6.0** | 1.0 | 1.01 | 1.0 | 0.99 | 1.0 | 1.0 | 28.3 | 0.26 | 14694 | 2450 |
| **494_bus**: $\varepsilon = 0.059$, $\delta = 0.001$ | | | | | | | | | | | | | | | | | | |
| $s = 10$, $C(\text{Top-}s, \widetilde{X}) = 0.834$, Top-$s$ running time= 0.09, $C(S^*) = 0.853$ | | | | | | | | | | | | | | | | | | |
| 1 | 7 | 0.06 | 1.01 | 1.0 | 304 | 2062 | **0.06** | **4.7** | 1.0 | 1.0 | 1.0 | 0.99 | 1.0 | 1.0 | 28.0 | 0.21 | 20543 | 2450 |
| **fb-pages-food**: $\varepsilon = 0.077$, $\delta = 0.001$ | | | | | | | | | | | | | | | | | | |
| $s = 5$, $C(\text{Top-}s, \widetilde{X}) = 0.959$, Top-$s$ running time= 0.14, $C(S^*) = 0.653$ | | | | | | | | | | | | | | | | | | |
| 1 | 6 | 0.04 | 1.01 | 1.03 | 179 | 2062 | **0.07** | **3.0** | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 6.6 | 0.14 | 15030 | 2321 |
| **fb-pages-food**: $\varepsilon = 0.065$, $\delta = 0.001$ | | | | | | | | | | | | | | | | | | |
| $s = 10$, $C(\text{Top-}s, \widetilde{X}) = 0.881$, Top-$s$ running time= 0.13, $C(S^*) = 0.771$ | | | | | | | | | | | | | | | | | | |
| 1 | 7 | 0.05 | 1.01 | 1.0 | 196 | 2062 | **0.06** | **3.3** | 1.0 | 1.01 | 1.0 | 0.99 | 1.0 | 1.0 | 15.7 | 0.14 | 21171 | 2321 |
| **662_bus**: $\varepsilon = 0.082$, $\delta = 0.001$ | | | | | | | | | | | | | | | | | | |
| $s = 5$, $C(\text{Top-}s, \widetilde{X}) = 0.866$, Top-$s$ running time= 0.06, $C(S^*) = 0.609$ | | | | | | | | | | | | | | | | | | |
| 1 | 6 | 0.08 | 1.0 | 1.03 | 499 | 2062 | **0.06** | **5.5** | 1.0 | 1.01 | 1.0 | 0.99 | 1.0 | 1.0 | 33.5 | 0.48 | 15126 | 2438 |
| **662_bus**: $\varepsilon = 0.065$, $\delta = 0.001$ | | | | | | | | | | | | | | | | | | |
| $s = 10$, $C(\text{Top-}s, \widetilde{X}) = 0.792$, Top-$s$ running time= 0.11, $C(S^*) = 0.77$ | | | | | | | | | | | | | | | | | | |
| 1 | 7 | 0.06 | 1.03 | 1.01 | 259 | 2062 | **0.07** | **4.7** | 1.0 | 1.01 | 1.0 | 0.99 | 1.0 | 1.0 | 127.4 | 0.45 | 21351 | 2438 |

All running times are in seconds. **PAA** is executed 15 times and the average results are reported. For **SAA**, we set $M = 15$ and report the total time used for solving MIPs (in lines 5–8) over all iterations in column "Sol. Time"; the total time used for estimating the solution quality (in lines 10–18) over all iterations is reported in column "Ev. Time". The best total running time results are in **bold** . That is, we need to compare the times given by the sum of those in columns "Sol. Time" and "Ev. Time" against the times given in column "Avg time"

**Table 6** Comparing **SAA** ($\beta = 1$) and **PAA** for large graphs

| **SAA** (Algorithm 4) | | | | | | | | **PAA** (Algorithm 3) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | $r^{(2)}$ | $r^{(3)}$ | $r^{(2)}$ | $r^{(3)}$ | $r^{(2)}$ | $r^{(3)}$ | $r^{(2)}$ | $r^{(3)}$ | $r^{(2)}$ | $r^{(3)}$ |
| # iter | $\varepsilon_{\text{SAA}}$ | $\bar{\mu}_M$ | $\widetilde{C}$ | max $\lvert X_\ell \rvert$ | $\widetilde{r}$ | Time | | Avg $C^*_X$ | | $C(S^*_X, \widetilde{X})$ | | | | Avg. time | | #paths | |
| | | | | | | Ev. | Sol. | | | Avg | | max | | | | | |
| **opsahl-powergrid**: $\varepsilon = 0.01, \delta = 0.001$ | | | | | | | | | | | | | | | | | |
| $s = 10, C(\text{Top-}s, \widetilde{X}) = 0.663$, Top-$s$ running time= 2.51 | | | | | | | | | | | | | | | | | |
| 13 | 0.006 | 0.83 | 0.84 | 2867 | 51545 | **12.4** | **28.6** | 0.84 | 0.84 | 0.83 | 0.83 | 0.83 | 0.83 | 116.8 | 387.0 | 67891 | 38005 |
| $s = 30, C(\text{Top-}s, \widetilde{X}) = 0.784$, Top-$s$ running time= 2.5 | | | | | | | | | | | | | | | | | |
| 15 | 0.006 | 0.93 | 0.93 | 13170 | 51545 | 15.1 | 309.9 | 0.93 | 0.93 | 0.93 | 0.93 | 0.93 | 0.93 | **131.5** | 376.3 | 67891 | 38005 |
| **pgp**: $\varepsilon = 0.01, \delta = 0.001$ | | | | | | | | | | | | | | | | | |
| $s = 10, C(\text{Top-}s, \widetilde{X}) = 0.576$, Top-$s$ running time= 1.39 | | | | | | | | | | | | | | | | | |
| 8 | 0.009 | 0.6 | 0.6 | 1634 | 51545 | **4.4** | **15.9** | 0.6 | 0.6 | 0.6 | 0.6 | 0.6 | 0.6 | 49.1 | 129.8 | 60623 | 38005 |
| $s = 30, C(\text{Top-}s, \widetilde{X}) = 0.809$, Top-$s$ running time= 1.49 | | | | | | | | | | | | | | | | | |
| 13 | 0.008 | 0.84 | 0.84 | 5837 | 51545 | 7.8 | 74.6 | 0.83 | 0.83 | 0.83 | 0.83 | 0.83 | 0.83 | **70.6** | 121.3 | 60623 | 38005 |
| **ia-enron-large**: $\varepsilon = 0.01, \delta = 0.001$ | | | | | | | | | | | | | | | | | |
| $s = 10, C(\text{Top-}s, \widetilde{X}) = 0.374$, Top-$s$ running time= 2.88 | | | | | | | | | | | | | | | | | |
| 9 | 0.005 | 0.4 | 0.4 | 2514 | 51545 | 9.7 | 39.0 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | **16.2** | 24.1 | 57539 | 38005 |
| $s = 30, C(\text{Top-}s, \widetilde{X}) = 0.66$, Top-$s$ running time= 2.98 | | | | | | | | | | | | | | | | | |
| 11 | 0.01 | 0.66 | 0.66 | 11774 | 51545 | 12.2 | 107.3 | 0.66 | 0.66 | 0.66 | 0.66 | 0.66 | 0.66 | **15.6** | 25.9 | 57539 | 38005 |
| **soc-gowalla**: $\varepsilon = 0.01, \delta = 0.001$ | | | | | | | | | | | | | | | | | |
| $s = 10, C(\text{Top-}s, \widetilde{X}) = 0.762$, Top-$s$ running time= 11.62 | | | | | | | | | | | | | | | | | |
| 10 | 0.009 | 0.76 | 0.76 | 926 | 51545 | 43.1 | 165.6 | 0.76 | 0.76 | 0.76 | 0.76 | 0.76 | 0.76 | **9.0** | 11.8 | 58442 | 38005 |
| $s = 30, C(\text{Top-}s, \widetilde{X}) = 0.836$, Top-$s$ running time= 13.6 | | | | | | | | | | | | | | | | | |
| 14 | 0.003 | 0.84 | 0.84 | 7613 | 51545 | 60.7 | 299.0 | 0.84 | 0.84 | 0.84 | 0.84 | 0.84 | 0.84 | **9.3** | 11.7 | 58442 | 38005 |

**Table 6** continued

| **SAA** (Algorithm 4) | | | | | | | | **PAA** (Algorithm 3) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | $r^{(2)}$ | $r^{(3)}$ | $r^{(2)}$ | $r^{(3)}$ | $r^{(2)}$ | $r^{(3)}$ | $r^{(2)}$ | $r^{(3)}$ | $r^{(2)}$ | $r^{(3)}$ |
| **ca-citeseer**: $\varepsilon = 0.01$, $\delta = 0.001$ | | | | | | | | | | | | | | | | | |
| $s = 10$, $C(\text{Top-}s, \widetilde{X}) = 0.229$, Top-$s$ running time= 21.66 | | | | | | | | | | | | | | | | | |
| 5 | 0.004 | 0.23 | 0.23 | 5099 | 51545 | 40.2 | 150.7 | 0.23 | 0.23 | 0.24 | 0.24 | 0.24 | 0.24 | **15.0** | 20.4 | 64793 | 38005 |
| $s = 30$, $C(\text{Top-}s, \widetilde{X}) = 0.33$, Top-$s$ running time= 21.8 | | | | | | | | | | | | | | | | | |
| 9 | 0.007 | 0.34 | 0.34 | 38025 | 51545 | 73.0 | 747.4 | 0.34 | 0.34 | 0.34 | 0.34 | 0.34 | 0.34 | **17.4** | 23.6 | 64793 | 38005 |
| **soc-twitter-higgs**: $\varepsilon = 0.01$, $\delta = 0.001$ | | | | | | | | | | | | | | | | | |
| $s = 10$, $C(\text{Top-}s, \widetilde{X}) = 0.37$, Top-$s$ running time= 325.23 | | | | | | | | | | | | | | | | | |
| 8 | 0.004 | 0.37 | 0.38 | 4998 | 51545 | 1043.6 | 1148.3 | 0.38 | 0.38 | 0.38 | 0.37 | 0.38 | 0.38 | **129.1** | 157.9 | 56507 | 38005 |
| $s = 30$, $C(\text{Top-}s, \widetilde{X}) = 0.627$, Top-$s$ running time= 336.03 | | | | | | | | | | | | | | | | | |
| 11 | 0.001 | 0.63 | 0.63 | 26871 | 51545 | 1442.4 | 3726.4 | 0.63 | 0.63 | 0.62 | 0.62 | 0.62 | 0.62 | **122.7** | 149.4 | 56507 | 38005 |
| **inf-roadNet-PA**: $\varepsilon = 0.01$, $\delta = 0.001$ | | | | | | | | | | | | | | | | | |
| $s = 10$, $C(\text{Top-}s, \widetilde{X}) = 0.266$, Top-$s$ running time= 2654.72 | | | | | | | | | | | | | | | | | |
| 13 | 0.005 | 0.7 | 0.71 | 3737 | 51545 | 12769.2 | 6317.2 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | **1250.0** | 1651.2 | 89046 | 38005 |
| $s = 30$, $C(\text{Top-}s, \widetilde{X}) = 0.394$, Top-$s$ running time= 2645.59 | | | | | | | | | | | | | | | | | |
| 15 | 0.006 | 0.86 | 0.86 | 18736 | 51545 | 14784.8 | 30846.7 | 0.86 | 0.85 | 0.85 | 0.85 | 0.86 | 0.86 | **4374.1** | 7037.5 | 89046 | 38005 |
| **soc-youtube-snap**: $\varepsilon = 0.01$, $\delta = 0.001$ | | | | | | | | | | | | | | | | | |
| $s = 10$, $C(\text{Top-}s, \widetilde{X}) = 0.534$, Top-$s$ running time= 55.59 | | | | | | | | | | | | | | | | | |
| 11 | 0.006 | 0.53 | 0.54 | 5137 | 51545 | 250.3 | 1473.8 | 0.53 | 0.53 | 0.53 | 0.53 | 0.53 | 0.53 | **30.9** | 37.7 | 60623 | 38005 |
| $s = 30$, $C(\text{Top-}s, \widetilde{X}) = 0.669$, Top-$s$ running time= 59.57 | | | | | | | | | | | | | | | | | |
| 11 | 0.006 | 0.68 | 0.68 | 5027 | 51545 | 251.5 | 1525.8 | 0.67 | 0.67 | 0.67 | 0.67 | 0.67 | 0.67 | **30.9** | 38.1 | 60623 | 38005 |

The graphs are too large to compute $C$ exactly and execute **PAA** with $r := r^{(1)}$. Hence, no scaling is used and we report only their approximations $\widetilde{C}$ for **SAA**; for **PAA** we report the results only for $r := r^{(2)}$ and $r := r^{(3)}$. To estimate the actual centrality of solutions, $S_X^*$, obtained by **PAA** we report $C(S_X^*, \widetilde{X})$, where $\widetilde{X}$ is randomly sampled with the same cardinality as in **SAA**, i.e., $\widetilde{r} = |\widetilde{X}|$. We refer to the captions of Table 5 for further details

**Algorithm 4:** Sample Average Approximation Algorithm **(SAA)**

1 **Input:** Graph $G = (V, E)$, the group size $s$, number of SAA problems to solve $M$, the numbers of path to be sampled with the strictly increasing sequence $r_k$ and parameter $\widetilde{r}$, parameter $\beta \in \{0, 1\}$ and error bound $\varepsilon > 0$;
**Result**: An estimate of $C^*$ and the corresponding group of $s$ nodes given by $\widetilde{C}$ and $\widetilde{S}$, respectively
2 **Initialize:** $k \leftarrow 0$;
3 **repeat**
4 $k \leftarrow k + 1$;
5 **for** $\ell = 1, \ldots, M$ **do**
6 Generate $X_\ell$, where $|X_\ell| = r_k$, using Algorithm 2 with input $(G, r_k)$;
7 Let $S^*_{X_\ell}$ and $C^*_{X_\ell}$ be an optimal solution and the optimal objective function value of $\mathbf{F2}(X)$ with $X = X_\ell$, respectively;
8 **end**
9 Compute $\bar{\mu}_M \leftarrow \frac{1}{M}\sum_{\ell=1}^{M} C^*_{X_\ell}$, and $\bar{\zeta}^2_M \leftarrow \frac{1}{M(M-1)}\sum_{\ell=1}^{M}(C^*_{X_\ell} - \bar{\mu}_M)^2$;
10 **if** $\beta = 0$ **then**
`// for β = 0 we compute betweenness centrality using all shortest paths`
11 $\widetilde{S} \in \operatorname{argmax}\{C(S) : S \in \{S^*_{X_1}, \ldots, S^*_{X_M}\}\}$;
12 $\widetilde{C} \leftarrow C(\widetilde{S})$;
13 **else**
`// for β = 1 we estimate betweenness centrality using X̃`
14 Generate $\widetilde{X}$, where $|\widetilde{X}| = \widetilde{r}$, using Algorithm 2 with input $(G, \widetilde{r})$;
15 $\widetilde{S} \in \operatorname{argmax}\{C(S, \widetilde{X}) : S \in \{S^*_{X_1}, \ldots, S^*_{X_M}\}\}$;
16 $\widetilde{C} \leftarrow C(\widetilde{S}, \widetilde{X})$; Compute $\widetilde{\zeta}^2$ using (37);
17 **end**
`// Stopping criterion`
18 **until** $\varepsilon \geq \varepsilon_{SAA} = \bar{\mu}_M - \widetilde{C} + z_\delta\sqrt{\bar{\zeta}^2_M + \beta \cdot \widetilde{\zeta}^2}$;
19 **Return:** $\widetilde{C}$ and $\widetilde{S}$;

Next, suppose we have $M$ multisets of randomly constructed shortest paths $X_1, \ldots, X_M$, $\ell \in \{1, \ldots, M\}$. Then we solve $M$ instances of $\mathbf{F2}(X)$, see (23), for each $X_\ell$, and define:

$$\bar{\mu}_M = \frac{1}{M}\sum_{\ell=1}^{M} C^*_{X_\ell}, \tag{34}$$

which is an unbiased estimator of $\mathbb{E}[C^*_X]$. We can also estimate variance of $\bar{\mu}_M$ by:

$$\bar{\zeta}^2_M = \frac{1}{M(M-1)}\sum_{\ell=1}^{M}(C^*_{X_\ell} - \bar{\mu}_M)^2. \tag{35}$$

By the Central Limit Theorem (CLT) the probability distribution of $\bar{\mu}_M$ becomes approximately normal as $M$ increases. The key idea of our approach (its formal pseudo-code is given in Aglorithm 4) is to exploit this fact.

Specifically, we solve $M$ problems $\mathbf{F2}(X)$; see lines 5-8 of Algorithm 4. We assume that $r = |X_\ell|$, $\ell \in \{1, \ldots, M\}$, is not required to satisfy conditions used in Sect. 3; recall inequality (19). Hence, the value of $r$ can be chosen sufficiently small (at least initially), which allows us to limit the sizes of the corresponding MIPs $\mathbf{F2}(X)$. Note that in Algorithm 4 the value of $r = |X_\ell|$, $\ell \in \{1, \ldots, M\}$, is controlled by some pre-defined strictly increasing sequence $r_k$, where $k$ is the iteration counter; see our brief discussion on this issue in Sect. 4.2.

By solving these $M$ models $\mathbf{F2}(X)$, we also obtain $M$ optimal solutions $S_1, \ldots, S_M$, and each of them is also a feasible solution for (3). To estimate the quality of these solutions, there are two possible options in Algorithm 4 that are controlled by parameter $\beta \in \{0, 1\}$:

- If $\beta = 0$, then we use all shortest paths in the graph and estimate $C^*$ using $C(\widetilde{S})$, where $\widetilde{S}$ is the best node subset out of $M$ available; see lines 10-12. This option is reasonable whenever the graph is sufficiently small and/or sparse; hence, we can pre-compute and store all shortest paths in the memory.
- Otherwise, i.e., $\beta = 1$, we sample a sufficiently large subset of paths, denoted as $\widetilde{X}$ using Algorithm 2; see line 14 in Algorithm 4. The number of these paths is controlled by parameter $\widetilde{r}$, where $|\widetilde{X}| = \widetilde{r}$. Consequently, we use these paths in (15) to pick the best node subset out of $M$ available and estimate $C^*$; see lines 13-17. One important remark that needs to be highlighted here is that we do not solve $\mathbf{F2}(X)$ with $X := \widetilde{X}$ but rather evaluate the quality of the candidate solutions, which are obtained by solving $\mathbf{F2}(X)$ with $X := X_\ell$. Hence, to achieve some computational advantage we need $|X_\ell| \ll |\widetilde{X}|$.

We exploit (34), (35) and the CLT to estimate the quality of the obtained solution via (33) and verify our stopping criteria in line 18 of Algorithm 4. If the stopping criteria is not satisfied, then we increase the sizes of randomly sampled path $X_1, \ldots, X_M$ (by increasing $k$ and hence, the value of $r := r_k$; see line 4) and the outlined steps are repeated. In view of our discussion above, we conclude that the following result holds under the appropriate settings of the algorithm input parameters.

**Proposition 4** *SAA terminates with probability 1, as $k \to \infty$.*

Next, we estimate the quality of the obtained solution given by the condition $\varepsilon \geq \varepsilon_{\text{SAA}}$ in line 18 of Algorithm 4. Note that in our discussion we follow the notation and the key concepts behind SAA outlined in [31].

First, suppose $\beta = 1$ and define $\text{gap}(\widetilde{S})$ as follows:

$$\text{gap}(\widetilde{S}) := C^* - C(\widetilde{S}) \leq \mathbb{E}[C^*_{\mathbf{X}}] - C(\widetilde{S}) = \mathbb{E}[\bar{\mu}_M - C(\widetilde{S}, \widetilde{\mathbf{X}})], \tag{36}$$

where the first inequality follows from (33) and the last equality is due to (17).

Define $\widetilde{C} = C(\widetilde{S}, \widetilde{X})$; see line 16 in Algorithm 4. The last equality in (36) implies that the value of $\bar{\mu}_M - \widetilde{C}$ can be used as an estimator for the upper bound on $\text{gap}(\widetilde{S})$.

Furthermore, given the definition in (36), observe that the variance of $\bar{\mu}_M - C(\widetilde{S}, \widetilde{X})$ is equal to the sum of the variances of $\bar{\mu}_M$ and $C(\widetilde{S}, \widetilde{X})$. Note that the empirical variance of $\bar{\mu}_M$ is $\bar{\zeta}^2_M$; recall (34) and (35). Also, recall that paths in $\widetilde{X}$ are generated independently and uniformly at random. Hence, we can estimate the corresponding sample variance as:

$$\widetilde{\zeta}^2 := \frac{1}{\widetilde{r}(\widetilde{r}-1)} \sum_{p \in \widetilde{X}} (\mathbb{1}_{\{p \in \mathcal{T}_S\}} - \widetilde{C})^2 = \frac{1}{\widetilde{r}-1}(\widetilde{C} - 2\widetilde{C}^2 + \widetilde{C}^2) = \frac{\widetilde{C} - \widetilde{C}^2}{\widetilde{r}-1}, \tag{37}$$

where we use the fact that $\widetilde{r} = |\widetilde{X}|$, i.e., $\widetilde{r}$ shortest paths are sampled; see line 16 in Algorithm 4.

Summarizing the above discussion, by the CLT, for sufficiently large values of $M$ and $\widetilde{r}$, we conclude that:

$$\varepsilon_{\text{SAA}} := \bar{\mu}_M - \widetilde{C} + z_\delta \sqrt{\bar{\zeta}^2_M + \widetilde{\zeta}^2}$$

estimates a $(1-\delta) \cdot 100\%$ confidence upper bound for $\text{gap}(\widetilde{S})$ in (36); here, we follow [31] and use critical value $z_\alpha$ from the standard normal distribution.

Note that there are two cases:

- If $C^* \geq \widetilde{C}$, then $\bar{\mu}_M + z_\delta \sqrt{\bar{\zeta}_M^2 + \widetilde{\zeta}^2} - C^* \geq 0$ can be estimated to hold with probability $1-\delta$ by our construction. Hence, with the same probability we estimate:

$$\begin{aligned} |C^* - \widetilde{C}| &\leq C^* - \widetilde{C} + \bar{\mu}_M + z_\delta \sqrt{\bar{\zeta}_M^2 + \widetilde{\zeta}^2} - C^* \\ &= \bar{\mu}_M - \widetilde{C} + z_\delta \sqrt{\bar{\zeta}_M^2 + \widetilde{\zeta}^2} = \varepsilon_{\text{SAA}} \leq \varepsilon, \end{aligned} \tag{38}$$

  where the last inequality follows from the stopping criterion in Algorithm 4; see line 18.
- If $C^* < \widetilde{C}$, then

$$|C^* - \widetilde{C}| = \widetilde{C} - C^* \leq \widetilde{C} - C^* + \text{gap}(\widetilde{S}) = \widetilde{C} - C(\widetilde{S}) \leq |\widetilde{C} - C(\widetilde{S})| \leq \varepsilon, \tag{39}$$

  where the first inequality holds by the definition of $\text{gap}(\widetilde{S})$, and the last inequality can be estimated to hold with probability at least $1-\delta$ as long as we sample a sufficiently large number of paths, $\widetilde{r}$.
  In particular, in our experiment we use

$$\widetilde{r} = \left\lceil \frac{1}{2\varepsilon^2} \left( \ln(M) + \ln \frac{2}{\delta} \right) \right\rceil, \tag{40}$$

  which can be justified as follows. Recall the proof of Proposition 3 and the derivation of inequality (26). Then we observe that the required bound in (39) can be shown to hold by using the similar arguments when replacing $|\mathcal{S}|$ by $M$ in (26). Then (24) is replaced by (40).

Next, by feasibility of $\widetilde{S}$ for the optimization problem in (3) we have that $C^* \geq C(\widetilde{S})$. Therefore, combining the latter with (36), (38) and (39), we can estimate the quality of $C(\widetilde{S})$ as:

$$Pr\{C^* - C(\widetilde{S}) \leq \varepsilon\} \geq 1 - \delta. \tag{41}$$

which holds for sufficiently large values $r_k$ and $M$. Similarly, we can estimate the quality of $\widetilde{C}$ with:

$$Pr\{|C^* - \widetilde{C}| \leq \varepsilon\} \geq 1 - \delta. \tag{42}$$

Recall that the above discussion and derivations in (41) and (42) assume that $\beta = 1$. On the other hand, if $\beta = 0$, then $\widetilde{C} = C(\widetilde{S})$; see Algorithm 4. Then (41) and (42) coincide. Moreover, $\bar{\mu}_M + z_\delta \bar{\zeta}_M - C^* \geq 0$ holds with probability $1-\delta$ by the Central Limit Theorem and the fact $M$ is sufficiently large (for $M < 30$ we use a $t$-distribution with $M-1$ degrees of freedom; see [31]). Thus, we obtain

$$\begin{aligned} |C^* - \widetilde{C}| &\leq C^* - \widetilde{C} + \bar{\mu}_M + z_\delta \bar{\zeta}_M - C^* \\ &= \bar{\mu}_M - \widetilde{C} + z_\delta \bar{\zeta}_M = \varepsilon_{\text{SAA}} \leq \varepsilon, \end{aligned}$$

where the last inequality follows from the stopping criterion in Algorithm 4.

Finally, we note that the computational performance of Algorithm 4 depends on the appropriate choices of its parameters that is, the values of $M$ and the sequence $r_k$, $k \geq 1$. In particular, the right-hand side in the stopping criteria (see line 18) decreases as $r_k$ increases and/or $M$ increases. Therefore, there is an inherent trade-off as for the larger values of $r_k$ and $M$ the sizes of the corresponding MIPs that need to be solved (see line 7) and the number of these MIPs, respectively, increase.

### 4.2 Computational results

In this section we present the second set of our computational experiments. Specifically, we compare the scalability and performance of **PAA** (see Algorithm 3) against **SAA** (see Algorithm 4). Similar to Sect. 3.4 we also provide the results for the Top-$s$ procedure.

For Algorithm 4 we compute the sequence $r_k$ as follows:

$$\varepsilon_0 = \left(\frac{\log\left(\frac{2}{\delta}\right)}{10s}\right)^{1/2}, \quad \varepsilon_k = \varepsilon_{k-1} \cdot \begin{cases} 1 - \frac{\varepsilon_{\text{SAA}}^{k-1}}{\bar{\mu}_M}, & \text{if } k \geq 2,\ \varepsilon_{\text{SAA}}^{k-2} - \varepsilon_{\text{SAA}}^{k-1} > 0.01, \\ \left(\frac{1}{\sqrt{2}}\right), & \text{otherwise,} \end{cases}$$

$$r_0 = 5s, \quad r_k = \frac{1}{2\varepsilon_k^2} \ln \frac{2}{\delta},$$

which implies that $r_1 = 10s$. The intuition behind our parameter settings is as follows. If $\varepsilon_{\text{SAA}}^{k-1}$ is sufficiently large relative to $\bar{\mu}_M$, then we scale $\varepsilon_k$ to be considerably smaller than $\varepsilon_{k-1}$. On the other hand, if we observe that increasing $r_k$ produces little improvement in $\varepsilon_{\text{SAA}}^k$ (specifically, $\varepsilon_{\text{SAA}}^{k-2} - \varepsilon_{\text{SAA}}^{k-1} \leq 0.1$), then we use $r_k = 2r_{k-1}$. Finally, we use $\delta = 10^{-3}$ for both algorithms.

**Additional test instances.** In addition to the real-life instances outlined in Sect. 3.4, we consider several larger real-life networks from [30]:

- **opsahl-powergrid** ($|V| = 4941$, $|E| = 6594$): an energy network representing the topology of the Western States Power Grid of the United States.
- **pgp** ($|V| = 10681$, $|E| = 24316$): a technological network, where nodes represent users that communicate via the Pretty-Good-Privacy protocol.
- **enron-large** ($|V| = 33697$, $|E| = 180811$): a social network formed by emails generated by employees of Enron Corporation.
- **citeseer** ($|V| = 227321$, $|E| = 814134$): a collaboration network from Citeseer repository.
- **soc-gowalla** ($|V| = 196591$, $|E| = 950327$): a social network, where users share their locations.
- **soc-twitter-higgs** ($|V| = 456631$, $|E| = 12508453$): a social network used to study the spreading processes on Twitter.
- **roadNet-PA** ($|V| = 1087563$, $|E| = 1541514$): a transportation network representing the roads in Pennsylvania.
- **soc-youtube-snap** ($|V| = 1134890$, $|E| = 2987624$): a social network from a video-sharing web site.

**Results and discussion**. Table 5 presents the results for the same instances considered in Tables 1, 2 and 3. For the stopping criteria in **SAA** we use the best performance guarantee, $\varepsilon$, that is achieved by Algorithm 3 in Tables 1, 2 and 3 (i.e., the last row for each network instance and the specified value of $s$). For **PAA**, we need to specify $r = |X|$, which is computed using the parameter $\varepsilon$; see (30) and (31), which correspond to $r := r^{(1)}$ and $r := r^{(2)}$, respectively. In view of (28) and (41), to have a fair comparison (i.e., the same performance quality guarantees), we compute $r^{(1)}$ and $r^{(2)}$ using $\varepsilon := \varepsilon_{PAA}$ in (30) and (31), where

$$\varepsilon_{PAA} := \varepsilon \cdot \left(1 + \sqrt{\frac{\ln \frac{2}{\delta}}{\ln\binom{n}{s} + \ln \frac{2}{\delta}}}\right)^{-1}, \tag{43}$$

and $\varepsilon$ in (43) corresponds to the stopping criteria of **SAA**.

As in Sect. 3.4, Algorithm **PAA** is executed 15 times and the average results are reported. For **SAA**, we set $M = 15$ and report the total time used for solving MIPs (in lines 5-8) over all iterations in column "Sol. Time"; the total time used for estimating the solution quality (in lines 10-18) over all iterations is reported in column "Ev. Time". The best total running time results are in **bold**; that is, we compare the time given in column "Avg time" for **PAA** against the times given by the sum of those in columns "Sol. Time" and "Ev. Time", i.e., the total time for **SAA**. All betweenness centrality values in Table 5 (including the errors $\varepsilon$ and $\varepsilon_{SAA}$) are scaled with respect to $C^*$ (except for $C^*$ itself). In view of the latter, we note that some values of $\widetilde{C}$ in Table 5 are larger than 1.

From the results in Table 5, we observe that **SAA** provides good quality solutions faster than **PAA**. For example, consider the results for **662_bus** with $s \in \{5, 10\}$ in Table 5. We observe that **SAA** provides a smaller error estimate (recall that the stopping criteria is $\varepsilon_{SAA} \leq \varepsilon$ in Algorithm 4), while its total running time is $0.06 + 5.5 \approx 5.6$ vs. 33.5 seconds on average for **PAA**, when $s = 5$, and 4.8 vs. 127.4, when $s = 10$. We also observe, when comparing the results between Tables 3 and 5, that for the same instance (**662_bus** with $s \in \{5, 10\}$), our procedure $(\varepsilon, \delta)$-**SAA** provides better running times and error estimates than $\Delta$-**AA**.

In Table 6 we present the results for networks that are larger than those in Table 5. Note that setting $r := r^{(1)}$ in **PAA** is too prohibitive for the MIP solver, as the resulting number of paths is too large. Thus, we consider only $r := r^{(2)}$ and $r := r^{(3)}$; see (31) and (32), respectively. Recall from our discussion above that $\varepsilon$ is set to the value in (43). Also, neither $C(S)$, nor $C^*$ can be computed exactly; hence, the reported values of the betweenness centrality are not scaled in Table 6.

First, we observe that in contrast to the results in Table 5, **PAA** can be competitive with respect to its running time performance. This observation is not surprising given that we do not use $r := r^{(1)}$ in our computations for the instances in Table 6. However, neither $r := r^{(2)}$ nor $r := r^{(3)}$ provide strong quality performance guarantees; recall Theorem 1. In fact, only (27) holds, which can be viewed as a relatively weak bound. However, **PAA** with $r := r^{(2)}$ and $r := r^{(3)}$ results in very good solutions, and this observation opens an interesting avenue for future research. That is, it could be of interest to relax the requirement on the cardinality, $r$, of the shortest paths that need to be sampled. In particular, in view of the results in [29] (recall Proposition 2) we conjecture that Theorem 1 could be substantially strengthened.

As a final comment, we note that the MIP solution times within **PAA** on every instance from Table 6, are faster for $r^{(2)}$ than for $r^{(3)}$, despite the fact that $r^{(2)} > r^{(3)}$. This behaviour could be explained by a high degree of symmetry in the feasible regions of the respective MIPs, when the number of the sampled shortest paths is not sufficiently large. This observation provides yet another avenue for future research.

## 5 Concluding remarks

In this paper we describe simple randomized approaches for finding most central group of nodes based on the notion of betweenness centrality. Our methods combine the ideas behind randomized path sampling and the exact linear MIP model. We also provide some probabilistic estimates for the solution quality obtained by our approaches and illustrate their performance in a computational study. Clearly, our bounds (see Proposition 3 and Theorem 1)

are very loose and could be strengthened (e.g., by reducing the requirement on the number of shortest paths needed).

Note that the base MIP model for computing the most central groups can be modified to capture some pre-specified cohesiveness properties of the group of interest, e.g., it should form a clique, or a star. Furthermore, the betweenness centrality concept can be generalized for groups of graph elements, which contain nodes and edges simultaneously; see the corresponding discussion and numerical illustrations in [33] for both of these generalizations. Clearly, our methods can be extended to handle these meaningful generalizations in a relatively straightforward manner as only the corresponding MIP needs to be modified. Also, it could be of interest to extend our approaches for weighted graphs and other centrality concepts, in particular, those that involve paths, which are not necessarily shortest. These research directions provide interesting avenues for future studies.

**Acknowledgements** The authors are grateful to Dr. Oliver Hinder for pointing out an issue in the early version of the paper as well as his subsequent comments. Also, we would like to thank the anonymous referees for the comments and suggestions, which helped us to improve the paper.

## References

1. Akgün, M.K., Tural, M.K.: k-step betweenness centrality. Comput. Math. Organ. Theory **26**(1), 55–87 (2020)
2. Albert, R., Barabási, A.-L.: Statistical mechanics of complex networks. Rev. Mod. Phys. **74**(1), 47–97 (2002)
3. Angriman, E., van der Grinten, A., Bojchevski, A., Zügner, D., Günnemann, S., Meyerhenke, H.: Group centrality maximization for large-scale graphs. In: 2020 Proceedings of the Twenty-Second Workshop on Algorithm Engineering and Experiments (ALENEX). SIAM, pp. 56–69 (2020)
4. Borassi, M., Natale, E.: Kadabra is an adaptive algorithm for betweenness via random approximation. ACM J. Exp. Algorithm. **24**(1.2), 1–35 (2019)
5. Borgatti, S., Everett, M., Johnson, J.: Analyzing Social Networks. SAGE Publications Limited (2013)
6. Borgatti, S.P., Everett, M.G.: A graph-theoretic perspective on centrality. Social Netw. **28**(4), 466–484 (2006)
7. Brandes, U.: A faster algorithm for betweenness centrality. J. Math. Sociol. **25**(2), 163–177 (2001)
8. Chou, C.-H., Sheu, P., Hayakawa, M., Kitazawa, A.: Querying large graphs in biomedicine with colored graphs and decomposition. J. Biomed. Inform. **108**, 103503 (2020)
9. Dinler, D., Tural, M.K.: Faster computation of successive bounds on the group betweenness centrality. Networks **71**(4), 358–380 (2018)
10. Dolev, S., Elovici, Y., Puzis, R., Zilberman, P.: Incremental deployment of network monitors based on group betweenness centrality. Inf. Process. Lett. **109**(20), 1172–1176 (2009)
11. Everett, M., Borgatti, S.: The centrality of groups and classes. J. Math. Sociol. **23**(3), 181–201 (1999)
12. Everett, M.G., Borgatti, S.P.: Extending centrality. Models Methods Soc. Netw. Anal. **35**(1), 57–76 (2005)
13. Fink, M., Spoerhase, J.: Maximum betweenness centrality: approximability and tractable cases. In: International Workshop on Algorithms and Computation. Springer, pp. 9–20 (2011)
14. Fronzetti Colladon, A., Guardabascio, B., Innarella, R.: Using social network and semantic analysis to analyze online travel forums and forecast tourism demand. Decis. Support Syst. **123**, 113075 (2019)
15. Ganesana, B., Ramanb, S., Ramalingamb, S., Turanc, M.E., Bacak-Turanc, G.: Vulnerability of sewer network-graph theoretic approach. Desalination Water Treat. **196**, 370–376 (2020)
16. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, New York (2002)
17. Guan, J., Yan, Z., Yao, S., Xu, C., Zhang, H.: GBC-based caching function group selection algorithm for SINET. J. Netw. Comput. Appl. **85**, 56–63 (2017)
18. Gurobi Optimization, L.: Gurobi Optimizer Reference Manual (2021)
19. Hagberg, A., Swart, P., Chult, D.S.: Exploring network structure, dynamics, and function using network. Technical Report, Los Alamos National Lab. (LANL), Los Alamos (2008)
20. Jackson, M.: Social and Economic Networks. Princeton University Press (2010)

21. Jacob, R., Koschützki, D., Lehmann, K.A., Peeters, L., Tenfelde-Podehl, D.: Algorithms for Centrality Indices, chapter 4, Springer, Berlin, pp. 62–82 (2005)
22. Kleywegt, A.J., Shapiro, A., Homem-de Mello, T.: The sample average approximation method for stochastic discrete optimization. SIAM J. Optim. **12**(2), 479–502 (2002)
23. Kolaczyk, E.D., Chua, D.B., Barthélemy, M.: Group betweenness and co-betweenness: inter-related notions of coalition centrality. Soc. Netw. **31**(3), 190–203 (2009)
24. Mahmoody, A., Tsourakakis, C.E., Upfal, E.: Scalable betweenness centrality maximization via sampling. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1765–1773 (2016)
25. Mitzenmacher, M., Upfal, E.: Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis. Cambridge University Press (2017)
26. Newman, M.: Networks: An introduction. Oxford University Press (2010)
27. Puzis, R., Elovici, Y., Dolev, S.: Fast algorithm for successive computation of group betweenness centrality. Phys. Rev. E **76**, 056709 (2007)
28. Puzis, R., Tubi, M., Elovici, Y., Glezer, C., Dolev, S.: A decision support system for placement of intrusion detection and prevention devices in large-scale networks. ACM Trans. Model. Comput. Simul. **22**(1), 1–26 (2011)
29. Riondato, M., Kornaropoulos, E.M.: Fast approximation of betweenness centrality through sampling. Data Min. Knowl. Discov. **30**(2), 438–475 (2016)
30. Rossi, R.A., Ahmed, N.K.: The network data repository with interactive graph analytics and visualization. In: AAAI (2015). http://networkrepository.com. Accessed 2 Nov 2021
31. Shapiro, A., Dentcheva, D., Ruszczyński, A.: Lectures on Stochastic Programming: Modeling and Theory. SIAM (2014)
32. Tubi, M., Puzis, R., Elovici, Y.: Deployment of DNIDS in social networks. In: 2007 IEEE Intelligence and Security Informatics. IEEE, pp. 59–65 (2007)
33. Veremyev, A., Prokopyev, O.A., Pasiliao, E.L.: Finding groups with maximum betweenness centrality. Optim. Methods Softw. **32**(2), 369–399 (2017)
34. Watts, D., Strogatz, S.: Collective dynamics of "small-world" networks. Nature **393**(6684), 440–442 (1998)