APPENDIX A
EI, DHT, AND THT

Effective importance (EI) [9] is the degree normalized version of random walk with restart [8], which can be defined as

$$\mathbf{r}_i = \begin{cases} c\sum_{j\in N_i} p_{i,j}\mathbf{r}_j + \frac{1-c}{w_i}, & \text{if } i=q, \\ c\sum_{j\in N_i} p_{i,j}\mathbf{r}_j, & \text{if } i\neq q, \end{cases}$$

where c ($0<c<1$) is a constant decay factor.

Lemma 2. EI has no local maximum.

Proof: Suppose that node i is a local maximum. We have $\mathbf{r}_i = c\sum_{j\in N_i} p_{i,j}\mathbf{r}_j \leq c\sum_{j\in N_i} p_{i,j}\mathbf{r}_i = c\,\mathbf{r}_i < \mathbf{r}_i$. Thus we get a contradiction that $\mathbf{r}_i < \mathbf{r}_i$. □

Discounted hitting time (DHT) [1] is a variant of hitting time [20], which can be defined as

$$\mathbf{r}_i = \begin{cases} 0, & \text{if } i=q, \\ 1+c\sum_{j\in N_i} p_{i,j}\mathbf{r}_j, & \text{if } i\neq q, \end{cases}$$

where c ($0<c<1$) is a constant decay factor.

Lemma 3. DHT has no local minimum.

Proof: The maximum discounted hitting time that a node could have is $\frac{1}{1-c}$, when it cannot reach q. For connected graph, we have $\mathbf{r}_i < \frac{1}{1-c}$. Now suppose that node i is a local minimum. We have that $\mathbf{r}_i = 1+c\sum_{j\in N_i} p_{i,j}\mathbf{r}_j \geq 1+c\sum_{j\in N_i} p_{i,j}\mathbf{r}_i = 1+c\mathbf{r}_i$. Thus $\mathbf{r}_i \geq \frac{1}{1-c}$, which contradicts the fact that $\mathbf{r}_i < \frac{1}{1-c}$. □

Truncated hitting time (THT) [5] is another variant of hitting time [20]. The L-truncated hitting time only considers paths of length less than L. Let $\mathbf{r}_i^{\mathsf{L}}$ and $\mathbf{r}_i^{\mathsf{L}-1}$ be the L- and $(\mathsf{L}-1)$-truncated hitting time of node i respectively when the query is q. The L-truncated hitting time can be defined as

$$\mathbf{r}_i^{\mathsf{L}} = \begin{cases} 0, & \text{if } i=q, \\ 1+\sum_{j\in N_i} p_{i,j}\mathbf{r}_j^{\mathsf{L}-1}, & \text{if } i\neq q. \end{cases}$$

If a node is no less than L hops away from the query node, its proximity is set to L. If node i is within L hops away from the query node, we have that $\mathbf{r}_i^{\mathsf{L}} < \mathsf{L}$.

Lemma 4. THT has no local minimum for the nodes within L hops from the query node.

Proof: Suppose that node i is within L hops from q. We can prove that $\forall j \in N_i$, $1+\mathbf{r}_j^{\mathsf{L}-1} \geq \mathbf{r}_j^{\mathsf{L}}$, and there exists at least one node $j\in N_i$ such that $1+\mathbf{r}_j^{\mathsf{L}-1} > \mathbf{r}_j^{\mathsf{L}}$ by mathematical induction on L. Now suppose that i is a local minimum, i.e., $\mathbf{r}_j^{\mathsf{L}} \geq \mathbf{r}_i^{\mathsf{L}}$ ($\forall j \in N_i$). We have that $\mathbf{r}_i^{\mathsf{L}} = 1+\sum_{j\in N_i} p_{i,j}\mathbf{r}_j^{\mathsf{L}-1} = \sum_{j\in N_i} p_{i,j}(1+\mathbf{r}_j^{\mathsf{L}-1}) > \sum_{j\in N_i} p_{i,j}\mathbf{r}_j^{\mathsf{L}} \geq \sum_{j\in N_i} p_{i,j}\mathbf{r}_i^{\mathsf{L}} = \mathbf{r}_i^{\mathsf{L}}$. We get a contradiction that $\mathbf{r}_i^{\mathsf{L}} > \mathbf{r}_i^{\mathsf{L}}$. □

The proof of Theorem 2 is as follows.

Proof: First, we show that PHP and EI are equivalent. Suppose the decay factors in PHP and EI are both set to be c. Then PHP and EI have the same recursive definition for any node $i\neq q$, and we have $\frac{\text{EI}(i)}{\text{PHP}(i)} = \frac{\text{EI}(q)}{\text{PHP}(q)}$, which is a constant when q is fixed. Thus $\text{PHP}(i)$ is a linear function of $\text{EI}(i)$.

Next, we show that PHP and DHT are equivalent. Suppose the decay factors in PHP and DHT are both set to be c. The relationship between PHP and DHT is $\text{PHP}(i) = 1-(1-c)\cdot\text{DHT}(i)$, which can be proved by substituting it in the recursive definition of PHP. Thus $\text{PHP}(i)$ is a linear function of $\text{DHT}(i)$. □

APPENDIX B
PROOFS OF THEOREMS 3 AND 5 IN SECTION 4.1

The proof of Theorem 3 is as follows.

Proof: Let $\mathbf{P}$ be the original transition probability matrix of PHP. Deleting $\mathbf{P}_{i,j}$ from $\mathbf{P}$ is the same as setting $\mathbf{P}_{i,j}$ to 0. Let $\mathbf{P}'$ represent the resulting matrix. The PHP proximity vector $\mathbf{r}$ is computed based on $\mathbf{P}$, and $\mathbf{r}'$ is based on $\mathbf{P}'$.

Let $\Delta\mathbf{r} = \mathbf{r}-\mathbf{r}'$, and $\Delta\mathbf{P} = \mathbf{P}-\mathbf{P}'$. Note that $\Delta\mathbf{P}$ has only one non-zero element $\Delta\mathbf{P}_{i,j} = \mathbf{P}_{i,j}$. We have $\Delta\mathbf{r} = c(\mathbf{P}'+\Delta\mathbf{P})\mathbf{r} - c\mathbf{P}'\mathbf{r}' = c\mathbf{P}'\Delta\mathbf{r} + c\Delta\mathbf{P}\mathbf{r} = c\mathbf{P}'\Delta\mathbf{r} + \mathbf{e}'$, where $\mathbf{e}'$ is a vector with the only non-zero element $\mathbf{e}'_i = c\mathbf{P}_{i,j}\mathbf{r}_j$. The solution of the previous equation is $\Delta\mathbf{r} = (\mathbf{I}-c\mathbf{P}')^{-1}\mathbf{e}'$. The elements of $c\mathbf{P}'$ are non-negative and $||c\mathbf{P}'||_\infty < 1$. We have that $(\mathbf{I}-c\mathbf{P}')^{-1} = \sum_{l=0}^{\infty}(c\mathbf{P}')^l \geq 0$ [29]. Thus the solution $\Delta\mathbf{r}$ must be non-negative. This completes the proof. □

The proof of Theorem 5 is as follows.

Proof: Let $\mathbf{P}$ be the original transition probability matrix of PHP. Changing the destination of transition probability $p_{i,j}$ from node j to node u is the same as adding $\mathbf{P}_{i,j}$ to $\mathbf{P}_{i,u}$ and setting $\mathbf{P}_{i,j}$ to 0. Let $\mathbf{P}'$ represent the resulting matrix. PHP proximity $\mathbf{r}$ is computed based on $\mathbf{P}$, and $\mathbf{r}'$ is based on $\mathbf{P}'$.

Let $\Delta\mathbf{r} = \mathbf{r}'-\mathbf{r}$, and $\Delta\mathbf{P} = \mathbf{P}-\mathbf{P}'$. Note that $\Delta\mathbf{P}$ has only two non-zero elements $\Delta\mathbf{P}_{i,j} = \mathbf{P}_{i,j}$ and $\Delta\mathbf{P}_{i,u} = -\mathbf{P}_{i,j}$. We have that $\Delta\mathbf{r} = c\mathbf{P}'\mathbf{r}' - c(\mathbf{P}'+\Delta\mathbf{P})\mathbf{r} = c\mathbf{P}'\Delta\mathbf{r} - c\Delta\mathbf{P}\mathbf{r} = c\mathbf{P}'\Delta\mathbf{r} + \mathbf{e}'$, where $\mathbf{e}'$ is a vector with the only non-zero element $\mathbf{e}'_i = c\mathbf{P}_{i,j}(\mathbf{r}_u - \mathbf{r}_j)$. If $\mathbf{r}_u \geq \mathbf{r}_j$, $\Delta\mathbf{r}$ must be non-negative. Otherwise, it is non-positive. This completes the proof. □

APPENDIX C
PROOF OF THEOREM 6 IN SECTION 5.2

The proof of Theorem 6 is as follows.

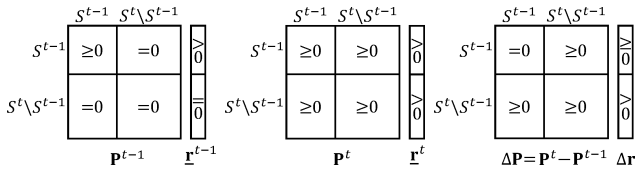
Proof: Let $\mathbf{P}^{t-1}$ and $\mathbf{P}^t$ be the transition probability matrices at the $(t-1)$-th and t-th iterations respectively. The lower bound $\underline{\mathbf{r}}^{t-1}$ is computed based on $\mathbf{P}^{t-1}$, and $\underline{\mathbf{r}}^t$ is computed based on $\mathbf{P}^t$. Since the sizes of matrices $\mathbf{P}^{t-1}$ and $\mathbf{P}^t$ are $|S^{t-1}|\times|S^{t-1}|$ and $|S^t|\times|S^t|$ respectively, we extend $\mathbf{P}^{t-1}$ with $|S^t \setminus S^{t-1}|$ columns and $|S^t \setminus S^{t-1}|$ rows. The newly added elements are set to 0. After extending, $\mathbf{P}^{t-1}$ and $\mathbf{P}^t$ are of the same size. Similarly, we extend $\underline{\mathbf{r}}^{t-1}$ with $|S^t \setminus S^{t-1}|$ elements, which are also set to 0. Thus, $\underline{\mathbf{r}}^{t-1}$ and $\underline{\mathbf{r}}^t$ are of the same size $|S^t|\times 1$ after extending. For any node $i\in S^{t-1}$, we have $\underline{\mathbf{r}}_i^{t-1} > 0$. Figures 20(a) and 20(b) illustrate these matrices and vectors. We have the following two equations.

$$\begin{cases} \underline{\mathbf{r}}^{t-1} = c\mathbf{P}^{t-1}\underline{\mathbf{r}}^{t-1} + \mathbf{e}, \\ \underline{\mathbf{r}}^{t} \quad = c\mathbf{P}^{t}\underline{\mathbf{r}}^{t} \qquad + \mathbf{e}, \end{cases}$$

where $\mathbf{e}_q = 1$ and $\mathbf{e}_i = 0$ if $i \in S^t \setminus \{q\}$.

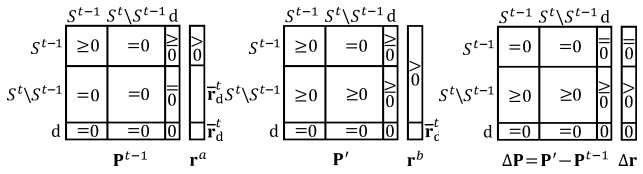
Let $\Delta\mathbf{r} = \underline{\mathbf{r}}^t - \underline{\mathbf{r}}^{t-1}$ and $\Delta\mathbf{P} = \mathbf{P}^t - \mathbf{P}^{t-1}$. Note that $\Delta\mathbf{P}_{i,j} = 0$ if $i,j \in S^{t-1}$ and $\Delta\mathbf{P}_{i,j} = \mathbf{P}^t_{i,j}$ otherwise. Figure 20(c) illustrates the matrix $\Delta\mathbf{P}$ and vector $\Delta\mathbf{r}$. We have that $\Delta\mathbf{r} = c\mathbf{P}^t\underline{\mathbf{r}}^t - c\mathbf{P}^{t-1}\underline{\mathbf{r}}^{t-1} = c\mathbf{P}^t\Delta\mathbf{r} + c\Delta\mathbf{P}\underline{\mathbf{r}}^{t-1} = c\mathbf{P}^t\Delta\mathbf{r} + \mathbf{e}'$, where $\mathbf{e}' = c\Delta\mathbf{P}\underline{\mathbf{r}}^{t-1}$.

Note that $\Delta\mathbf{P}_{i,j} = 0$ if $i,j \in S^{t-1}$. We also have that $\underline{\mathbf{r}}_j^{t-1} = 0$ if $j \in S^t \setminus S^{t-1}$. Thus, $\mathbf{e}'_i = 0$ if $i \in S^{t-1}$. For any node $i \in S^t \setminus S^{t-1}$, it must connect to at least one node in S^{t-1}. Thus, for each node $i \in S^t \setminus S^{t-1}$, $\Delta\mathbf{P}_{i,j} > 0$ for some node $j \in S^{t-1}$.



(a) $(t-1)$-th iteration (b) t-th iteration (c) difference

Fig. 20. Transition probability matrices between two adjacent iterations (lower bound)



(a) after step 1 (b) after step 2 (c) difference

Fig. 21. Transition probability matrices between two adjacent iterations (upper bound, step 2)

We also have that $\underline{\mathbf{r}}_j^{t-1} > 0$ if $j \in S^{t-1}$. Therefore, $\mathbf{e}'_i > 0$ if $i \in S^t \setminus S^{t-1}$. Thus, we have

$$\begin{cases} \mathbf{e}'_i = 0, & \text{if } i \in S^{t-1}, \\ \mathbf{e}'_i > 0, & \text{if } i \in S^t \setminus S^{t-1}. \end{cases}$$

We can get that $\Delta\mathbf{r} = (\mathbf{I} - c\mathbf{P}^t)^{-1}\mathbf{e}'$. The elements of $c\mathbf{P}^t$ are non-negative and $||c\mathbf{P}^t||_\infty < 1$. We have that $(\mathbf{I} - c\mathbf{P}^t)^{-1} = \sum_{l=0}^{\infty} c^l (\mathbf{P}^t)^l$, where $(\mathbf{P}^t)^l$ represents the matrix $\mathbf{P}^t$ to the power of l. Each element $[(\mathbf{P}^t)^l]_{i,j}$ represents the probability of transiting from node i to j in the l-th step.

Consider a node $i \in S^{t-1}$. If $i \rightsquigarrow S^t \setminus S^{t-1}$, we have $[(\mathbf{P}^t)^l]_{i,j} > 0$ for some l and some node $j \in S^t \setminus S^{t-1}$. So, $[(\mathbf{P}^t)^l \mathbf{e}']_i > 0$ and $\Delta\mathbf{r}_i > 0$. If $i \not\rightsquigarrow S^t \setminus S^{t-1}$, we have $[(\mathbf{P}^t)^l]_{i,j} = 0$ for any l and any node $j \in S^t \setminus S^{t-1}$. So, $[(\mathbf{P}^t)^l \mathbf{e}']_i = 0$ and $\Delta\mathbf{r}_i = 0$. □

APPENDIX D
PROOF OF THEOREM 7 IN SECTION 5.3

In this subsection, we will prove Theorem 7, which shows the monotonicity of the upper bounds. In Section 5.3, from iteration $(t-1)$ to t, the transition graph is modified in three steps. We will analyze the three steps rigidly in Lemmas 5, 6, and 7 respectively. Only the process from iteration $(t-1)$ to t is considered in the three lemmas. After the discussion of the three lemmas, we will use mathematical induction on each iteration to prove the monotonicity of the upper bound at all the iterations. Thus, in Lemmas 5, 6, and 7, we assume that $t \geq 2$ and $\overline{\mathbf{r}}_{\mathrm{d}}^{t-1} > \overline{\mathbf{r}}_i^{t-1}$ for any node $i \in \delta S^{t-1}$. The idea is that the condition is first assumed to be satisfied at the $(t-1)$-th iteration, and we will prove that it is also satisfied at the t-th iteration.

In Lemmas 5, 6, and 7, we use the following symbols. At the $(t-1)$-th iteration, we use $\mathbf{P}^{t-1}$ to denote the transition probability matrix, and $\overline{\mathbf{r}}^{t-1}$ to denote the upper bound vector. After applying step 1, the transition probability matrix does not change and we use $\mathbf{r}^a$ to represent the new upper bound vector. After applying step 2, the new transition probability matrix is denoted as $\mathbf{P}'$ and the new upper bound vector is denoted as $\mathbf{r}^b$. After applying step 3, we use $\mathbf{P}^t$ to denote the transition probability matrix, and $\overline{\mathbf{r}}^t$ to denote the upper bound vector.

Lemma 5. *(Step 1)* For any node $i \in S^{t-1}$, we have

$$\begin{cases} \overline{\mathbf{r}}_i^{t-1} = \mathbf{r}_i^a, & \text{if } i \not\rightsquigarrow \mathrm{d}, \\ \overline{\mathbf{r}}_i^{t-1} > \mathbf{r}_i^a, & \text{if } i \rightsquigarrow \mathrm{d}, \end{cases}$$

where $\rightsquigarrow$ and $\not\rightsquigarrow$ represent the reachability in the transition graph at the $(t-1)$-th iteration.

Proof: In step 1, we decrease the proximity value of the dummy node from $\overline{\mathbf{r}}_{\mathrm{d}}^{t-1}$ to $\overline{\mathbf{r}}_{\mathrm{d}}^t = \max_{i \in \delta S^{t-1}} \overline{\mathbf{r}}_i^{t-1}$. By the assumption, $\overline{\mathbf{r}}_{\mathrm{d}}^{t-1} > \overline{\mathbf{r}}_i^{t-1}$ for any node $i \in \delta S^{t-1}$. Thus, $\overline{\mathbf{r}}_{\mathrm{d}}^{t-1} > \overline{\mathbf{r}}_{\mathrm{d}}^t$.

The upper bound vectors $\overline{\mathbf{r}}^{t-1}$ and $\mathbf{r}^a$ both are computed based on $\mathbf{P}^{t-1}$ but with different $\mathbf{e}$ vectors. The size of matrix $\mathbf{P}^{t-1}$ is $(|S^{t-1}|+1)\times(|S^{t-1}|+1)$. The sizes of vectors $\overline{\mathbf{r}}^{t-1}$ and $\mathbf{r}^a$ both are $(|S^{t-1}|+1)\times 1$. We have the following two equations.

$$\begin{cases} \overline{\mathbf{r}}^{t-1} = c\mathbf{P}^{t-1}\overline{\mathbf{r}}^{t-1} + \mathbf{e}^{t-1}, \\ \mathbf{r}^a \quad = c\mathbf{P}^{t-1}\mathbf{r}^a \quad + \mathbf{e}^a. \end{cases}$$

In the first equation, $\mathbf{e}_q^{t-1} = 1$, $\mathbf{e}_{\mathrm{d}}^{t-1} = \overline{\mathbf{r}}_{\mathrm{d}}^{t-1}$, and $\mathbf{e}_i^{t-1} = 0$ if $i \in S^{t-1} \setminus \{q\}$. In the second equation, $\mathbf{e}_q^a = 1$, $\mathbf{e}_{\mathrm{d}}^a = \overline{\mathbf{r}}_{\mathrm{d}}^t$, and $\mathbf{e}_i^a = 0$ if $i \in S^{t-1} \setminus \{q\}$.

Let $\Delta\mathbf{r} = \overline{\mathbf{r}}^{t-1} - \mathbf{r}^a$. We have that $\Delta\mathbf{r} = c\mathbf{P}^{t-1}\Delta\mathbf{r} + \mathbf{e}'$, where $\mathbf{e}' = \mathbf{e}^{t-1} - \mathbf{e}^a$. Since $\overline{\mathbf{r}}_{\mathrm{d}}^{t-1} > \overline{\mathbf{r}}_{\mathrm{d}}^t$, we have that

$$\begin{cases} \mathbf{e}'_i = 0, & \text{if } i \in S^{t-1}, \\ \mathbf{e}'_i > 0, & \text{if } i = \mathrm{d}. \end{cases}$$

We can get that $\Delta\mathbf{r} = (\mathbf{I} - c\mathbf{P}^{t-1})^{-1}\mathbf{e}'$. The elements of $c\mathbf{P}^{t-1}$ are non-negative and $||c\mathbf{P}^{t-1}||_\infty < 1$. We have that $(\mathbf{I} - c\mathbf{P}^{t-1})^{-1} = \sum_{l=0}^{\infty} c^l (\mathbf{P}^{t-1})^l$, where $(\mathbf{P}^{t-1})^l$ denotes matrix $\mathbf{P}^{t-1}$ to the power of l. Each element $[(\mathbf{P}^{t-1})^l]_{i,\mathrm{d}}$ represents the probability of transiting from node i to d in the l-th step.

Consider a node $i \in S^{t-1}$. If $i \rightsquigarrow \mathrm{d}$, we have that $[(\mathbf{P}^{t-1})^l]_{i,\mathrm{d}} > 0$ for some l. Thus, $\Delta\mathbf{r}_i > 0$, i.e., $\overline{\mathbf{r}}_i^{t-1} > \mathbf{r}_i^a$. If $i \not\rightsquigarrow \mathrm{d}$, we have that $[(\mathbf{P}^{t-1})^l]_{i,\mathrm{d}} = 0$ for any l. Thus, $\Delta\mathbf{r}_i = 0$. □

Lemma 6. *(Step 2)*

$$\begin{cases} \mathbf{r}_i^a = \mathbf{r}_i^b, & \text{if } i \in S^{t-1}, \\ \overline{\mathbf{r}}_{\mathrm{d}}^t > \mathbf{r}_i^b, & \text{if } i \in S^t \setminus S^{t-1}. \end{cases}$$

Proof: In step 2, we add transition probabilities $\{p_{i,j}\}$ from the newly added nodes $i (\in S^t \setminus S^{t-1})$ to nodes $j (\in S^t)$, and $\{p_{i,\mathrm{d}}\}$ from i to the dummy node d.

Since the sizes of matrices $\mathbf{P}^{t-1}$ and $\mathbf{P}'$ are $(|S^{t-1}|+1)\times(|S^{t-1}|+1)$ and $(|S^t|+1)\times(|S^t|+1)$ respectively, we extend $\mathbf{P}^{t-1}$ with $|S^t \setminus S^{t-1}|$ columns and $|S^t \setminus S^{t-1}|$ rows as shown in Figure 21(a). The newly added elements are set to 0. Thus, matrices $\mathbf{P}^{t-1}$ and $\mathbf{P}'$ are of the same size after extension. Similarly, we extend $\mathbf{r}^a$ with $|S^t \setminus S^{t-1}|$ elements, which are set to $\overline{\mathbf{r}}_{\mathrm{d}}^t$. Thus, $\mathbf{r}^a$ and $\mathbf{r}^b$ are of the same size $(|S^t|+1)\times 1$ after extension. For any node $i \in S^{t-1}$, we must have that $\mathbf{r}_i^a > 0$. Figures 21(a) and 21(b) illustrate these matrices and vectors. We have the following two equations.

$$\begin{cases} \mathbf{r}^a = c\mathbf{P}^{t-1}\mathbf{r}^a + \mathbf{e}^a, \\ \mathbf{r}^b = c\mathbf{P}'\mathbf{r}^b \quad + \mathbf{e}^t. \end{cases}$$

In the first equation, $\mathbf{e}_q^a = 1$, $\mathbf{e}_i^a = 0$ if $i \in S^{t-1} \setminus \{q\}$ and $\mathbf{e}_i^a = \overline{\mathbf{r}}_{\mathrm{d}}^t$ if $i \in \left(S^t \setminus S^{t-1}\right) \cup \{\mathrm{d}\}$. In the second equation, $\mathbf{e}_q^t = 1$, $\mathbf{e}_{\mathrm{d}}^t = \overline{\mathbf{r}}_{\mathrm{d}}^t$, and $\mathbf{e}_i^t = 0$ if $i \in S^t \setminus \{q\}$.

Let $\Delta\mathbf{r} = \mathbf{r}^a - \mathbf{r}^b$ and $\Delta\mathbf{P} = \mathbf{P}' - \mathbf{P}^{t-1}$. Note that $\Delta\mathbf{P}_{i,j} = 0$ if $i \in S^{t-1} \cup \{\mathrm{d}\}$ and $\Delta\mathbf{P}_{i,j} = \mathbf{P}'_{i,j}$ otherwise. Figure 21(c) illustrates the matrix $\Delta\mathbf{P}$ and vector $\Delta\mathbf{r}$. We have that $\Delta\mathbf{r} = c\mathbf{P}^{t-1}\mathbf{r}^a - c\mathbf{P}'\mathbf{r}^b + \mathbf{e}^a - \mathbf{e}^t = c\mathbf{P}'\Delta\mathbf{r} - c\Delta\mathbf{P}\mathbf{r}^a + \mathbf{e}^a - \mathbf{e}^t = c\mathbf{P}'\Delta\mathbf{r} + \mathbf{e}'$, where $\mathbf{e}' = \mathbf{e}^a - \mathbf{e}^t - c\Delta\mathbf{P}\mathbf{r}^a$.

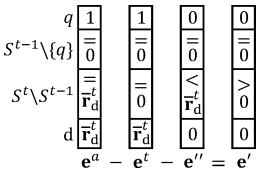


Fig. 22. The vectors $\mathbf{e}^a$, $\mathbf{e}^t$, $\mathbf{e}''$, and $\mathbf{e}'$

Let $\mathbf{e}''=c\Delta\mathbf{P}\mathbf{r}^a$. Note that $\Delta\mathbf{P}_{i,j}=0$ if $i\in S^{t-1}\cup\{\mathrm{d}\}$. Thus, $\mathbf{e}''_i=0$ if $i\in S^{t-1}\cup\{\mathrm{d}\}$. We have that $\Delta\mathbf{P}_{i,j}=0$ if $i\in S^t\backslash S^{t-1}$ and $j\in S^{t-1}\backslash\delta S^{t-1}$ and $\Delta\mathbf{P}_{i,j}\geq 0$ if $i\in S^t\backslash S^{t-1}$ and $j\in\delta S^{t-1}$. For any node $j\in\delta S^{t-1}$, $j\nrightarrow\mathrm{d}$. By Lemma 5, we have that $\bar{\mathbf{r}}^{t-1}_j>\mathbf{r}^a_j$. Since $\bar{\mathbf{r}}^t_{\mathrm{d}}=\max_{i\in\delta S^{t-1}}\bar{\mathbf{r}}^{t-1}_i$, $\mathbf{r}^a_j<\bar{\mathbf{r}}^t_{\mathrm{d}}$ for any node $j\in\delta S^{t-1}$. We also have that $\mathbf{r}^a_j=\bar{\mathbf{r}}^t_{\mathrm{d}}$ if $j\in\big(S^t\backslash S^{t-1}\big)\cup\{\mathrm{d}\}$. The sum of the elements in the i-th ($i\in S^t\backslash S^{t-1}$) row of $\Delta\mathbf{P}$ equals 1. Thus, $\mathbf{e}''_i<\bar{\mathbf{r}}^t_{\mathrm{d}}$ if $i\in S^t\setminus S^{t-1}$. Since $\mathbf{e}'=\mathbf{e}^a-\mathbf{e}^t-\mathbf{e}''$, we have that

$$\begin{cases}\mathbf{e}'_i=0, & \text{if } i\in S^{t-1}\cup\{\mathrm{d}\},\\ \mathbf{e}'_i>0, & \text{if } i\in S^t\setminus S^{t-1}.\end{cases}$$

Figure 22 illustrates the vectors $\mathbf{e}^a$, $\mathbf{e}^t$, $\mathbf{e}''$, and $\mathbf{e}'$.

We can get that $\Delta\mathbf{r}=(\mathbf{I}-c\mathbf{P}')^{-1}\mathbf{e}'$. The elements of $c\mathbf{P}'$ are non-negative and $||c\mathbf{P}'||_\infty<1$. We have that $(\mathbf{I}-c\mathbf{P}')^{-1}=\sum_{l=0}^\infty c^l(\mathbf{P}')^l$, where $(\mathbf{P}')^l$ represents the matrix $\mathbf{P}'$ to the power of l. We have that $\Delta\mathbf{r}=(\mathbf{I}+\sum_{l=1}^\infty c^l(\mathbf{P}')^l)\mathbf{e}'=\mathbf{e}'+\sum_{l=1}^\infty c^l(\mathbf{P}')^l\mathbf{e}'$. Since all the elements in matrix $\mathbf{P}'$ are non-negative, we have that $[(\mathbf{P}')^l]_{i,j}\geq 0$ for any l and any nodes $i,j\in S^t\cup\{\mathrm{d}\}$.

For any node $i\in S^t\backslash S^{t-1}$, $\Delta\mathbf{r}_i>0$, i.e., $\mathbf{r}^a_i=\bar{\mathbf{r}}^t_{\mathrm{d}}>\mathbf{r}^b_i$.

For any node $i\in S^{t-1}$, $i\nrightarrow S^t\backslash S^{t-1}$ in the transition graph corresponding to $\mathbf{P}'$. Thus, $[(\mathbf{P}')^l]_{i,j}=0$ for any l and any node $j\in S^t\backslash S^{t-1}$. So, $\Delta\mathbf{r}_i=0$. □

Lemma 7. *(Step 3)* For any node $i\in S^t$, we have that

$$\begin{cases}\mathbf{r}^b_i=\bar{\mathbf{r}}^t_i, & \text{if } i\nrightarrow S^t\setminus S^{t-1},\\ \mathbf{r}^b_i>\bar{\mathbf{r}}^t_i, & \text{if } i\rightsquigarrow S^t\setminus S^{t-1},\end{cases}$$

where $\rightsquigarrow$ and $\nrightarrow$ represent the reachability in the transition graph at the t-th iteration.

Proof: In step 3, we add the transition probabilities $\{p_{j,i}\}$ from nodes $j\in\delta S^{t-1}$ to the newly added nodes $i\in S^t\backslash S^{t-1}$, and remove their correspondences in $\{p_{j,\mathrm{d}}\}$.

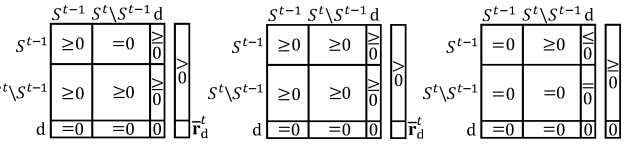
The sizes of matrices $\mathbf{P}'$ and $\mathbf{P}^t$ both are $(|S^t|+1)\times(|S^t|+1)$. The sizes of vectors $\mathbf{r}^b$ and $\bar{\mathbf{r}}^t$ both are $(|S^t|+1)\times 1$. Figures 23(a) and 23(b) illustrate these matrices and vectors. We have the following two equations.

$$\begin{cases}\mathbf{r}^b=c\mathbf{P}'\mathbf{r}^b+\mathbf{e}^t,\\ \bar{\mathbf{r}}^t=c\mathbf{P}^t\bar{\mathbf{r}}^t+\mathbf{e}^t,\end{cases}$$

where $\mathbf{e}^t_q=1$, $\mathbf{e}^t_{\mathrm{d}}=\bar{\mathbf{r}}^t_{\mathrm{d}}$, and $\mathbf{e}^t_i=0$ if $i\in S^t\setminus\{q\}$.

Let $\Delta\mathbf{r}=\mathbf{r}^b-\bar{\mathbf{r}}^t$ and $\Delta\mathbf{P}=\mathbf{P}^t-\mathbf{P}'$. Note that $\Delta\mathbf{P}_{i,j}=0$ if $i\in\big(S^t\backslash S^{t-1}\big)\cup\{\mathrm{d}\}$ or $j\in S^{t-1}$, $\Delta\mathbf{P}_{i,j}\geq 0$ if $i\in S^{t-1}$ and $j\in S^t\backslash S^{t-1}$, and $\Delta\mathbf{P}_{i,\mathrm{d}}\leq 0$ if $i\in S^{t-1}$. The sum of elements in each row of $\Delta\mathbf{P}$ equals 0. Figure 23(c) illustrates the matrix $\Delta\mathbf{P}$ and vector $\Delta\mathbf{r}$. We have that $\Delta\mathbf{r}=c\mathbf{P}'\mathbf{r}^b-c\mathbf{P}^t\bar{\mathbf{r}}^t=c\mathbf{P}^t\Delta\mathbf{r}-c\Delta\mathbf{P}\mathbf{r}^b=c\mathbf{P}^t\Delta\mathbf{r}+\mathbf{e}'$, where $\mathbf{e}'=-c\Delta\mathbf{P}\mathbf{r}^b$.

Consider the set of nodes $S'=\{i\,|\,i\in\delta S^{t-1}\text{ and }\mathbf{P}^t_{i,j}>0\text{ for some }j\in S^t\setminus S^{t-1}\}$. S' comprises the set of nodes in δS^{t-1} that are adjacent to the newly added nodes. If node $i\notin S'$, the transition probability $p_{i,j}$ does not change during step 3. Thus, we have that $\Delta\mathbf{P}_{i,j}=0$ and $\mathbf{e}'_i=0$ for any node $i\notin S'$ and $j\in S^t\cup\{\mathrm{d}\}$. If node $i\in S'$, the sum of elements in the i-th row of $\Delta\mathbf{P}$ is equal to 0. By Lemma 6, we have that



(a) after step 2 (b) after step 3 (c) difference

Fig. 23. Transition probability matrices between two adjacent iterations (upper bound, step 3)

$\mathbf{r}^b_j<\bar{\mathbf{r}}^t_{\mathrm{d}}$ for any node $j\in S^t\setminus S^{t-1}$. Thus, we have that $\mathbf{e}'_i>0$. In conclusion, we have that

$$\begin{cases}\mathbf{e}'_i=0, & \text{if } i\in(S^t\setminus S')\cup\{\mathrm{d}\},\\ \mathbf{e}'_i>0, & \text{if } i\in S'.\end{cases}$$

We can get that $\Delta\mathbf{r}=(\mathbf{I}-c\mathbf{P}^t)^{-1}\mathbf{e}'$. The elements of $c\mathbf{P}^t$ are non-negative and $||c\mathbf{P}^t||_\infty<1$. We have that $(\mathbf{I}-c\mathbf{P}^t)^{-1}=\sum_{l=0}^\infty c^l(\mathbf{P}^t)^l$, where $(\mathbf{P}^t)^l$ represents the matrix $\mathbf{P}^t$ to the power of l. We have that $\Delta\mathbf{r}=\sum_{l=0}^\infty c^l(\mathbf{P}^t)^l\mathbf{e}'$. Since all the elements in matrix $\mathbf{P}^t$ are non-negative, we have that $[(\mathbf{P}^t)^l]_{i,j}\geq 0$ for any l and any nodes $i,j\in S^t\cup\{\mathrm{d}\}$.

Consider a node $i\in S^t$. If $i\rightsquigarrow S'$, $[(\mathbf{P}^t)^l]_{i,j}>0$ for some l and some node $j\in S'$. Thus, $\Delta\mathbf{r}_i>0$ and $\mathbf{r}^b_i>\bar{\mathbf{r}}^t_i$. If $i\nrightarrow S'$, $[(\mathbf{P}^t)^l]_{i,j}=0$ for any l and any $j\in S'$. Thus, $\Delta\mathbf{r}_i=0$ and $\mathbf{r}^b_i=\bar{\mathbf{r}}^t_i$. Since each node in S' is adjacent to at least one node in $S^t\backslash S^{t-1}$ and the graph is undirected, $i\rightsquigarrow S'$ if and only if $i\rightsquigarrow S^t\backslash S^{t-1}$. This completes the proof. □

Theorem 8. For any node $i\in\delta S^t$, we have that $\bar{\mathbf{r}}^t_{\mathrm{d}}>\bar{\mathbf{r}}^t_i$.

Proof: The theorem is proved by mathematical induction on the iteration. At the first iteration, for any node $i\in\delta S^1$, we have that $\bar{\mathbf{r}}^1_i<1$ and $\bar{\mathbf{r}}^1_{\mathrm{d}}=1$. So, the theorem is trivially satisfied.

Suppose that the theorem is satisfied at the $(t-1)$-th iteration with $t\geq 2$. That is, $\bar{\mathbf{r}}^{t-1}_{\mathrm{d}}>\bar{\mathbf{r}}^{t-1}_i$, for any node $i\in\delta S^{t-1}$. Next, we will prove that this theorem is still satisfied at the t-th iteration.

Consider a node $i\in\delta S^{t-1}$. Note that $i\rightsquigarrow\mathrm{d}$ in the transition graph corresponding to $\mathbf{P}^{t-1}$. By Lemma 5, $\bar{\mathbf{r}}^{t-1}_i>\mathbf{r}^a_i$. By Lemma 6, $\mathbf{r}^a_i=\mathbf{r}^b_i$. By Lemma 7, $\mathbf{r}^b_i\geq\bar{\mathbf{r}}^t_i$. Thus, $\bar{\mathbf{r}}^{t-1}_i>\bar{\mathbf{r}}^t_i$. Since $\bar{\mathbf{r}}^t_{\mathrm{d}}=\max_{i\in\delta S^{t-1}}\bar{\mathbf{r}}^{t-1}_i$, we have that $\bar{\mathbf{r}}^t_{\mathrm{d}}>\bar{\mathbf{r}}^t_i$.

Consider a node $i\in S^t\backslash S^{t-1}$. By Lemma 6, $\bar{\mathbf{r}}^t_{\mathrm{d}}>\mathbf{r}^b_i$. By Lemma 7, $\mathbf{r}^b_i>\bar{\mathbf{r}}^t_i$. Thus, we have that $\bar{\mathbf{r}}^t_{\mathrm{d}}>\bar{\mathbf{r}}^t_i$.

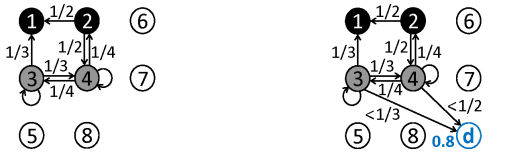
Since $\delta S^t\subset\delta S^{t-1}\cup(S^t\backslash S^{t-1})$, we have that $\bar{\mathbf{r}}^t_{\mathrm{d}}>\bar{\mathbf{r}}^t_i$ for any node $i\in\delta S^t$. This completes the proof. □

The proof of Theorem 7 is as follows.

Proof: For any t ($t\geq 1$) and any node $i\in\delta S^t$, we have that $\bar{\mathbf{r}}^t_{\mathrm{d}}>\bar{\mathbf{r}}^t_i$ by Theorem 8. Thus, the assumption for Lemmas 5, 6, and 7 is satisfied. Consider a node $i\in S^{t-1}$. Suppose that $i\rightsquigarrow\mathrm{d}$. By Lemma 5, $\bar{\mathbf{r}}^{t-1}_i>\mathbf{r}^a_i$. By Lemma 6, $\mathbf{r}^a_i=\mathbf{r}^b_i$. By Lemma 7, $\mathbf{r}^b_i\geq\bar{\mathbf{r}}^t_i$. Thus, we have that $\bar{\mathbf{r}}^{t-1}_i>\bar{\mathbf{r}}^t_i$. Suppose that $i\nrightarrow\mathrm{d}$. By Lemma 5, $\bar{\mathbf{r}}^{t-1}_i=\mathbf{r}^a_i$. By Lemma 6, $\mathbf{r}^a_i=\mathbf{r}^b_i$. By Lemma 7, $\mathbf{r}^b_i=\bar{\mathbf{r}}^t_i$. Thus, we have that $\bar{\mathbf{r}}^{t-1}_i=\bar{\mathbf{r}}^t_i$. □

APPENDIX E
TIGHTENING THE BOUNDS

The bounds used in FLoS can be further tightened by adding self-loop transition probabilities to the nodes in δS. We will still use PHP as an example to illustrate the process. We first define the star-to-mesh transformation on the transition graph, which is inspired by the star-mesh transformation in circuit theory [30].



(a) adding self-loop for LB $\underline{\mathbf{r}}$ (b) adding self-loop for UB $\bar{\mathbf{r}}$

Fig. 24. Transition graphs with self-loops

Definition 4. *[Star-to-mesh transformation]* (1) Delete a node $u \in V \setminus \{q\}$ and its incident transition probabilities; (2) For any pair of nodes $i, j \in N_u$, add transition probabilities $p'_{i,j} = cp_{i,u}p_{u,j}$.

Note that if $i = j$, it becomes the self-loop transition probability $p'_{i,i} = cp_{i,u}p_{u,i}$. Applying the star-to-mesh transformation for a node will not change the PHP proximity values of the remaining nodes.

Lemma 8. Applying the star-to-mesh transformation of node u will not change the PHP proximity values of nodes $V \setminus \{q, u\}$.

Proof: This can be proved by plugging the equation $\mathbf{r}_u = c\sum_{i \in N_u} p_{u,i}\mathbf{r}_i$ into the recursive equations of neighbor nodes $i \in N_u$ in the original transition graph. □

The self-loop transition probabilities generated in the star-to-mesh transformation can be used to further tighten the lower and upper bounds. Lemmas 9 and 10 analyze the tighter lower and upper bounds respectively. Figure 24 shows the transition graphs with self-loop transition probabilities for computing the lower and upper bounds. The original ones are shown in Figure 3.

Lemma 9. Adding self-loop transition probability $p_{i,i} = c\sum_{j \in N_i \cap \delta\bar{S}} p_{i,j}p_{j,i}$ $(\forall i \in \delta S)$ will tighten the lower bound.

Proof: First, we show the new bound values are still lower bounds. For the nodes in $\delta\bar{S}$, we apply star-to-mesh transformation sequentially in any order. After the star-to-mesh transformation of one node $j \in \delta\bar{S}$, we delete all the newly added transition probabilities except the self-loop transition probabilities of nodes in δS. After all the nodes in $\delta\bar{S}$ have been deleted, the self-loop transition probability of a node $i \in \delta S$ is $p_{i,i} = c\sum_{j \in N_i \cap \delta\bar{S}} p_{i,j}p_{j,i}$. During this process, we only apply star-to-mesh transformation and transition probability deletion. Therefore, the new bound values are still lower bounds.

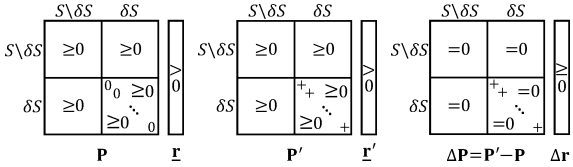
Next, we prove that the new lower bound is tighter.

Let $\mathbf{P}$ be the transition probability matrix for computing the original lower bound vector $\underline{\mathbf{r}}$. Since we add some self-loop transition probabilities to the nodes in δS, we use $\mathbf{P}'$ to denote the new transition probability matrix and $\underline{\mathbf{r}}'$ to denote the corresponding lower bound vector. Figures 25(a) and 25(b) illustrate the matrices and vectors. We have the following two equations

$$\begin{cases} \underline{\mathbf{r}} = c\mathbf{P}\underline{\mathbf{r}} + \mathbf{e}, \\ \underline{\mathbf{r}}' = c\mathbf{P}'\underline{\mathbf{r}}' + \mathbf{e}. \end{cases}$$

Let $\Delta\mathbf{r} = \underline{\mathbf{r}}' - \underline{\mathbf{r}}$ and $\Delta\mathbf{P} = \mathbf{P}' - \mathbf{P}$. Note that $\Delta\mathbf{P}_{i,i} > 0$ if $i \in \delta S$ and $\Delta\mathbf{P}_{i,j} = 0$ otherwise. Figure 25(c) illustrates the matrix $\Delta\mathbf{P}$ and vector $\Delta\mathbf{r}$. We have that $\Delta\mathbf{r} = c\mathbf{P}'\underline{\mathbf{r}}' - c\mathbf{P}\underline{\mathbf{r}} = c\mathbf{P}'\Delta\mathbf{r} + c\Delta\mathbf{P}\underline{\mathbf{r}} = c\mathbf{P}'\Delta\mathbf{r} + \mathbf{e}'$, where $\mathbf{e}' = c\Delta\mathbf{P}\underline{\mathbf{r}}$. We have that

$$\begin{cases} \mathbf{e}'_i = 0, & \text{if } i \in S \setminus \delta S, \\ \mathbf{e}'_i > 0, & \text{if } i \in \delta S. \end{cases}$$



(a) original (b) with self-loop (c) difference

Fig. 25. Transition probability matrices (lower bound)

We can get that $\Delta\mathbf{r} = (\mathbf{I} - c\mathbf{P}')^{-1}\mathbf{e}'$. The elements of $c\mathbf{P}'$ are non-negative and $||c\mathbf{P}'||_\infty < 1$. We have that $(\mathbf{I} - c\mathbf{P}')^{-1} = \sum_{l=0}^{\infty} c^l(\mathbf{P}')^l$, where $(\mathbf{P}')^l$ represents the matrix $\mathbf{P}'$ to the power of l. Each element $[(\mathbf{P}')^l]_{i,j}$ represents the probability of transiting from node i to j in the l-th step.

Consider a node $i \in S$. If $i \rightsquigarrow \delta S$ in the transition graph corresponding to $\mathbf{P}'$, $[(\mathbf{P}')^l]_{i,j} > 0$ for some l and some node $j \in \delta S$. So, $[(\mathbf{P}')^l\mathbf{e}']_i > 0$ and $\Delta\mathbf{r}_i > 0$. If $i \not\rightsquigarrow \delta S$ in the transition graph corresponding to $\mathbf{P}'$, $[(\mathbf{P}')^l]_{i,j} = 0$ for any l and any node $j \in \delta S$. So, $[(\mathbf{P}')^l\mathbf{e}']_i = 0$ and $\Delta\mathbf{r}_i = 0$. In conclusion, we have that $\underline{\mathbf{r}}_i \leq \underline{\mathbf{r}}'_i$ for any node $i \in S$. This completes the proof. □

Lemma 10. Adding self-loop transition probability $p_{i,i} = c\sum_{j \in N_i \cap \delta\bar{S}} p_{i,j}p_{j,i}$ and setting the transition probability to dummy node as $p_{i,\mathrm{d}} = c\sum_{j \in N_i \cap \delta\bar{S}} p_{i,j}(1 - p_{j,i})(\forall i \in \delta S)$ will tighten the upper bound.

Proof: First, we show the new bound values are still upper bounds. For the nodes in $\delta\bar{S}$, we apply star-to-mesh transformation sequentially in any order. After the star-to-mesh transformation of one node $j \in \delta\bar{S}$, we change the destination of all the newly added transition probabilities except the self-loop transition probabilities of nodes in δS to the dummy node d. After all the nodes in $\delta\bar{S}$ have been deleted, the self-loop transition probability of a node $i \in \delta S$ is $p_{i,i} = c\sum_{j \in N_i \cap \delta\bar{S}} p_{i,j}p_{j,i}$, and the transition probability from a node $i \in \delta S$ to the dummy node is $p_{i,\mathrm{d}} = c\sum_{j \in N_i \cap \delta\bar{S}} p_{i,j}(1 - p_{j,i})$. During this process, we only apply star-to-mesh transformation and change the destination of transition probabilities from a node $i \in \delta S$ to the dummy node. The proximity value of the dummy node is set as the maximum upper bound value of the nodes in the boundary node, i.e., $\bar{\mathbf{r}}_\mathrm{d} = \max_{i \in \delta S} \bar{\mathbf{r}}_i$. Thus, we have that $\bar{\mathbf{r}}_\mathrm{d} > \mathbf{r}_i$, for any node $i \in \delta S$. Therefore, the new bound values are still upper bounds.

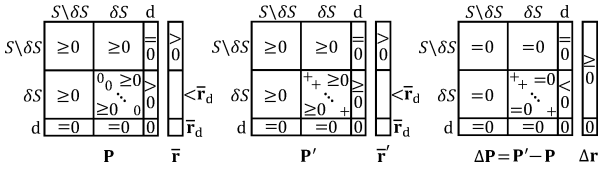
Next, we prove that the new upper bound is tighter.

Let $\mathbf{P}$ be the transition probability matrix for computing the original upper bound vector $\bar{\mathbf{r}}$. Let $\mathbf{P}'$ denote the new transition probability matrix and $\bar{\mathbf{r}}'$ denote the corresponding upper bound vector. Figures 26(a) and 26(b) illustrate the matrices and vectors. We have the following two equations

$$\begin{cases} \bar{\mathbf{r}} = c\mathbf{P}\bar{\mathbf{r}} + \mathbf{e}', \\ \bar{\mathbf{r}}' = c\mathbf{P}'\bar{\mathbf{r}}' + \mathbf{e}', \end{cases}$$

where $\mathbf{e}'_q = 1$, $\mathbf{e}'_\mathrm{d} = \bar{\mathbf{r}}_\mathrm{d}$, and $\mathbf{e}'_i = 0$ if $i \in S \setminus \{q\}$.

Let $\Delta\mathbf{r} = \bar{\mathbf{r}} - \bar{\mathbf{r}}'$ and $\Delta\mathbf{P} = \mathbf{P}' - \mathbf{P}$. Comparing $\mathbf{P}$ and $\mathbf{P}'$, we add some self-loop transition probabilities to the nodes in δS and change the destination of transition probabilities from node $i (\in \delta S)$ to the dummy node d. Note that $\Delta\mathbf{P}_{i,i} = p_{i,i}$ if $i \in \delta S$, $\Delta\mathbf{P}_{i,\mathrm{d}} < 0$ if $i \in \delta S$ and $\Delta\mathbf{P}_{i,j} = 0$ otherwise. Moreover, $-\Delta\mathbf{P}_{i,\mathrm{d}} = \sum_{j \in N_i \cap \delta\bar{S}} p_{i,j} - c\sum_{j \in N_i \cap \delta\bar{S}} p_{i,j}(1 - p_{j,i}) > p_{i,i}$. Thus, for any node $i \in \delta S$, $\Delta\mathbf{P}_{i,i} + \Delta\mathbf{P}_{i,\mathrm{d}} < 0$. Figure 26(c) illustrates the matrix $\Delta\mathbf{P}$ and vector $\Delta\mathbf{r}$.



(a) original (b) with self-loop (c) difference

Fig. 26. Transition probability matrices (upper bound)

We have that $\Delta\mathbf{r} = c\mathbf{P}\bar{\mathbf{r}} - c\mathbf{P}'\bar{\mathbf{r}}' = c\mathbf{P}'\Delta\mathbf{r} - c\Delta\mathbf{P}\bar{\mathbf{r}} = c\mathbf{P}'\Delta\mathbf{r} + \mathbf{e}''$, where $\mathbf{e}'' = -c\Delta\mathbf{P}\bar{\mathbf{r}}$. Based on Theorem 8, we have that for any node $i \in \delta S^t$, $\bar{\mathbf{r}}^t_{\mathrm{d}} > \bar{\mathbf{r}}^t_i$. We get that

$$\begin{cases} \mathbf{e}''_i = 0, & \text{if } i \in (S \setminus \delta S) \cup \{\mathrm{d}\}, \\ \mathbf{e}''_i > 0, & \text{if } i \in \delta S. \end{cases}$$

We can get that $\Delta\mathbf{r} = (\mathbf{I} - c\mathbf{P}')^{-1}\mathbf{e}''$. The elements of $c\mathbf{P}'$ are non-negative and $||c\mathbf{P}'||_\infty < 1$. We have that $(\mathbf{I} - c\mathbf{P}')^{-1} = \sum_{l=0}^{\infty} c^l(\mathbf{P}')^l$, where $(\mathbf{P}')^l$ represents the matrix $\mathbf{P}'$ to the power of l. Each element $[(\mathbf{P}')^l]_{i,j}$ represents the probability of transiting from node i to j in the l-th step.

Consider a node $i \in S$. If $i \rightsquigarrow \delta S$ in the transition graph corresponding to $\mathbf{P}'$, $[(\mathbf{P}')^l]_{i,j} > 0$ for some l and some node $j \in \delta S$. So, $[(\mathbf{P}')^l\mathbf{e}'']_i > 0$ and $\Delta\mathbf{r}_i > 0$. If $i \nrightsquigarrow \delta S$ in the transition graph corresponding to $\mathbf{P}'$, $[(\mathbf{P}')^l]_{i,j} = 0$ for any l and any node $j \in \delta S$. So, $[(\mathbf{P}')^l\mathbf{e}'']_i = 0$ and $\Delta\mathbf{r}_i = 0$. In conclusion, we have that $\bar{\mathbf{r}}_i \geq \bar{\mathbf{r}}'_i$ for any node $i \in S$. This completes the proof. □

APPENDIX F
ANALYSIS ON THE NUMBER OF VISITED NODES

In this subsection, we analyze the number of visited nodes. Let h be the average number of neighbors of a node. Suppose that the nodes in the boundary of visited nodes are ρ hops away from the query node q. We assume that the number of visited nodes equals h^ρ. Note that this is an upper bound of the number of visited nodes.

The proximity of one node will decrease by a factor of c when it is one hop farther away from the query node. The nodes in the boundary of the visited nodes have proximity less than c^ρ because they are ρ hops away from the query.

What is the distribution of the gaps between the upper and lower bounds? The upper bound $\bar{\mathbf{r}}$ is computed based on the recursive equation $\bar{\mathbf{r}} = c\mathbf{P}\bar{\mathbf{r}} + \mathbf{e}''$, where vector $\mathbf{e}''$ has only two non-zero elements $\mathbf{e}''_q = 1$ and $\mathbf{e}''_{\mathrm{d}} = \mathbf{r}_{\mathrm{d}}$. When we set the proximity value of dummy node to 0, i.e., $\mathbf{e}''_{\mathrm{d}} = 0$, we will get the recursive equation $\underline{\mathbf{r}} = c\mathbf{P}\underline{\mathbf{r}} + \mathbf{e}$ for the lower bound $\underline{\mathbf{r}}$. Let $\mathbf{r}' = \bar{\mathbf{r}} - \underline{\mathbf{r}}$ be the gaps between the upper and lower bounds. We have that $\mathbf{r}' = c\mathbf{P}\mathbf{r}' + \mathbf{e}'$, where vector $\mathbf{e}'$ has only one non-zero element $\mathbf{e}'_{\mathrm{d}} = \mathbf{r}_{\mathrm{d}}$. Thus the gaps $\mathbf{r}'$ can be interpreted as the PHP proximity values when the query node is the dummy node d with constant proximity value $\mathbf{r}_{\mathrm{d}}$.

Based on this observation, the gap $\mathbf{r}'_u$ of one node u will decrease by a factor of c when it is one hop farther away from the boundary. Let ρ_u denote the number of hops that u is away from the query node. Node u is $\rho - \rho_u$ hops away from the boundary. So, the gap $\mathbf{r}'_u$ is less than $c^{\rho-\rho_u} \cdot c^\rho$, i.e., $\mathbf{r}'_u < c^{2\rho-\rho_u}$.

For any two nodes u and v, we can distinguish their rankings if $\mathbf{r}'_u + \mathbf{r}'_v < \epsilon$, where ϵ is the difference of the exact proximity values of nodes u and v. We already derive that $\mathbf{r}'_u < c^{2\rho-\rho_u}$ and $\mathbf{r}'_v < c^{2\rho-\rho_v}$. Thus if we have that $\rho > \frac{1}{2}\log_c \epsilon - \frac{1}{2}\log_c(c^{-\rho_u} + c^{-\rho_v})$, then the pair of nodes u and v can be distinguished.

Now we consider the case when u and v are the k-th and $(k+1)$-th nodes in the exact ranking list respectively. We have $h^{\rho_u} \geq k$ and $h^{\rho_v} \geq k+1$. Thus, the inequality $\rho > \frac{1}{2}\log_c \frac{\epsilon}{2} + \frac{1}{2}\log_h k$ should be satisfied. Therefore, we need to visit $h^\rho = O((kh^{\log_c(\epsilon/2)})^{\frac{1}{2}})$ nodes to distinguish the k-th and $(k+1)$-th nodes.

APPENDIX G
EXTENSIONS OF FLOS TO THE PROXIMITY MEASURES HAVING LOCAL MAXIMUM

This subsection shows the details about how to extend the FLoS method to random walk with restart, RoundTripRank, Katz score, and absorption probability.

G.1 Extension to Random Walk with Restart

In this subsection, we discuss how to apply the FLoS method to random walk with restart (RWR) [8] which has local optimum. The key idea is to utilize the relationship between RWR and PHP. Utilizing such relationship, we can easily derive the lower and upper bounds for RWR based on the lower and upper bounds for PHP.

Random walk with restart (also known as personalized PageRank) [8] is a widely used proximity measure. RWR can be described as follows. From a node i, the random walker can walk to its neighbors with probabilities proportional to the edge weights. In each step, it has a probability of $(1-c)$ to return to the query node q, where c is a constant. The proximity of node i w.r.t. q is defined as the stationary probability that the random walker will finally stay at i. RWR can be defined recursively as

$$\mathbf{r}_i = \begin{cases} c\sum_{j \in N_i} p_{j,i}\mathbf{r}_j + (1-c), & \text{if } i = q, \\ c\sum_{j \in N_i} p_{j,i}\mathbf{r}_j, & \text{if } i \neq q, \end{cases}$$

where c ($0 < c < 1$) is a constant decay factor, and $p_{j,i} = \frac{w_{j,i}}{w_j}$ is the transition probability from node j to i.

Lemma 11. RWR has local maximum.

Proof: Examples can be constructed to show that RWR has local maximum, which are omitted. □

Let $\mathrm{RWR}(i)$ and $\mathrm{PHP}(i)$ represent the RWR and PHP proximity values of node i respectively. We first show that RWR and PHP have the following relationship.

Theorem 9. $\mathrm{RWR}(i) \propto w_i \cdot \mathrm{PHP}(i)$

Proof: Based on the recursive definition of RWR, we have $\frac{\mathrm{RWR}(i)}{w_i} = c\sum_{j \in N_i} \frac{w_{j,i}}{w_j} \cdot \frac{\mathrm{RWR}(j)}{w_i} = c\sum_{j \in N_i} \frac{w_{i,j}}{w_i} \cdot \frac{\mathrm{RWR}(j)}{w_j}$, for any node $i \neq q$. Thus, for any node $i \neq q$, we have that $\frac{\mathrm{RWR}(i)}{w_i} = c\sum_{j \in N_i} p_{i,j} \cdot \frac{\mathrm{RWR}(j)}{w_j}$. This degree normalized RWR, $\frac{\mathrm{RWR}(i)}{w_i}$, has the same recursive equation as PHP with decay factor c. So we have that $\frac{\mathrm{RWR}(i)}{w_i \cdot \mathrm{PHP}(i)} = \frac{\mathrm{RWR}(q)}{w_q \cdot \mathrm{PHP}(q)}$. When the query node q is fixed, we have that $\mathrm{RWR}(i) \propto w_i \cdot \mathrm{PHP}(i)$. □

How to extend FLoS to RWR by using this relationship is already discussed in Section 6.

Here, we provide more details about how to maintain the maximum degree of unvisited nodes. In the implementation of the algorithm, we use the vector container in C++. The

program can access any element in the vector in $O(1)$ time, and can delete the last element in $O(1)$ time. Each element in the vector is of structure type and contains two members, one represents the node degree and the other represents the node color. If a node is visited, its node color is black; otherwise, its node color is white.

When we pre-process the graph, we sort the elements in the vector by the node degree in ascending order, and set the initial node color of each element to white.

The last element always contains the node with the maximum degree of unvisited nodes. When we newly visit a node, we set its node color to black. If this newly visited node corresponds to the last element in the vector, we remove this element from the vector. And then we start removing the elements from the end of the vector until the last element's node color is white.

To maintain the maximum degree of unvisited nodes, the amortized time complexity is $O(1)$ for each newly visited node.

G.2 Extension to RoundTripRank

RoundTripRank (RT) [10] considers round trip paths between nodes. RT of node i is defined as the probability that a round trip from q contains node i. The round trip can be decoupled into two types of trips, i.e., the forward trip from node q to i and the backward trip from node i to q. Let $\mathbf{r}_i$ denote the RT proximity value of node i with regard to the query node q. Let $\mathbf{x}_i$ denote the proximity value defined in the forward direction from q to i, and $\mathbf{y}_i$ denote the proximity value defined in the backward direction from i to q. $\mathbf{x}_i$ and $\mathbf{y}_i$ can be defined as

$$\mathbf{x}_i = \begin{cases} c\sum_{j\in N_i} p_{j,i}\mathbf{x}_j + (1-c), & \text{if } i = q, \\ c\sum_{j\in N_i} p_{j,i}\mathbf{x}_j, & \text{if } i \neq q, \end{cases}$$

$$\mathbf{y}_i = \begin{cases} c\sum_{j\in N_i} p_{i,j}\mathbf{y}_j + (1-c), & \text{if } i = q, \\ c\sum_{j\in N_i} p_{i,j}\mathbf{y}_j, & \text{if } i \neq q, \end{cases}$$

where c $(0<c<1)$ is a constant decay factor, and $p_{i,j} = \frac{w_{i,j}}{w_i}$ is the transition probability from node i to j.

RT can be decomposed as the multiplication of these two values, i.e.,

$$\mathbf{r}_i \propto \mathbf{x}_i^{\beta} \cdot \mathbf{y}_i^{1-\beta},$$

where β $(0\leq\beta\leq 1)$ is a constant and used to tune the trade-off between the forward and backward directions.

The relationship between RT and PHP is shown in the following theorem.

Theorem 10. $\text{RT}(i) \propto w_i^{\beta} \cdot \text{PHP}(i)$

Proof: We can see that $\mathbf{x}_i$ is exactly the RWR proximity value, and $\mathbf{y}_i$ has a linear relationship with the EI proximity value, i.e., $\mathbf{y}_i = w_q \cdot \text{EI}(i)$. Thus, we have that $\text{RT}(i) \propto \text{RWR}(i)^{\beta} \cdot \text{EI}(i)^{1-\beta}$. From Theorem 2, we have that $\text{EI}(i) \propto \text{PHP}(i)$. From Theorem 9, we have that $\text{RWR}(i) \propto w_i \cdot \text{PHP}(i)$. Thus, $\text{RT}(i) \propto w_i^{\beta} \cdot \text{PHP}(i)$. □

Lemma 12. RT has no local maximum when $\beta = 0$, and has local maximum when $0 < \beta \leq 1$.

Proof: Based on Theorem 10, when $\beta = 0$, we have that $\text{RT}(i) \propto \text{PHP}(i)$. Since PHP has no local maximum, RT has no local maximum. Examples can be constructed to show that RT has local maximum when $0 < \beta \leq 1$, which are omitted. □

Suppose that node $v \in \delta S^t$ has the largest PHP proximity value. Based on Theorem 1, for any $i \in \bar{S}^t$, we have that $\text{PHP}(i) \leq \text{PHP}(v)$. Let $w(\bar{S}^t)$ denote the maximum degree of unvisited nodes in $\bar{S}^t$. We have that $w_i^{\beta} \cdot \text{PHP}(i) \leq w(\bar{S}^t)^{\beta} \cdot \text{PHP}(i) \leq w(\bar{S}^t)^{\beta} \cdot \text{PHP}(v)$. Therefore, if we maintain the maximum degree of unvisited nodes, we can develop the upper bound for the proximity values of unvisited nodes.

Specifically, we can apply FLoS to RT as follows. In Algorithm 3, we can change line 1 to the following line.

1: $u \leftarrow \text{argmax}_{i\in\delta S^{t-1}} w_i^{\beta} \cdot (\underline{\mathbf{r}}_i^{t-1} + \overline{\mathbf{r}}_i^{t-1})$;

In Algorithm 6, we can change line 2 and 3 to the following two lines.

2: $K \leftarrow k$ nodes in $S^t \setminus (\delta S^t \cup \{q\})$ with largest $w_i^{\beta} \cdot \underline{\mathbf{r}}_i^t$ values;
3: **if** $\min_{i\in K} w_i^{\beta} \cdot \underline{\mathbf{r}}_i^t \geq \max_{i\in S^t\setminus(K\cup\{q\})} w_i^{\beta} \cdot \overline{\mathbf{r}}_i^t$ **and** $\min_{i\in K} w_i^{\beta} \cdot \underline{\mathbf{r}}_i^t \geq w(\bar{S}^t)^{\beta} \cdot \max_{i\in\delta S^t} \overline{\mathbf{r}}_i^t$ **then** bStop $\leftarrow$ true;

All other processes remain the same.

G.3 Extension to the Katz Score

The Katz score (KZ) [18] measures the proximity between nodes via a weighted sum of the length of paths between them. Let $\mathbf{R}$ denote the matrix containing all pairwise KZ proximity values (with $\mathbf{R}_{i,j}$ denoting the KZ proximity value between nodes i and j). Let $\mathbf{W}$ be the symmetric adjacency matrix of an undirected and connected graph. We have that

$$\mathbf{R} = \kappa\mathbf{W} + \kappa^2\mathbf{W}^2 + \cdots = (\mathbf{I} - \kappa\mathbf{W})^{-1} - \mathbf{I},$$

where κ $(0<\kappa<1)$ is the decay factor.

In general, KZ has local maximum. But when κ is small, it does not have local maximum. To prove this, we first define a new proximity measure PHP′ as follows

$$\mathbf{r}_i = \begin{cases} 1, & \text{if } i = q, \\ c\sum_{j\in N_i} p_{i,j}\mathbf{r}_j, & \text{if } i \neq q, \end{cases}$$

where c $(0<c<1)$ is the decay factor in the random walk process, $p_{i,j} = w_{i,j}/w_{\max}$ denotes the transition probability from node i to j, and $w_{\max}$ denotes the maximum degree among all the nodes. Different from PHP, the transition probability in PHP′ is normalized by the maximum degree.

Theorem 11. If $\kappa < 1/w_{\max}$, $\text{KZ}(i) \propto \text{PHP}'(i)$.

Proof: For the KZ proximity measure, we already have that $\mathbf{R} = (\mathbf{I} - \kappa\mathbf{W})^{-1} - \mathbf{I}$. The elements in the q-th column of $\mathbf{R}$ represent the KZ proximity values when the query node is q. Thus, the KZ proximity vector is $\mathbf{r} = (\mathbf{I} - \kappa\mathbf{W})^{-1}\mathbf{e} - \mathbf{e}$. Then, we have that

$$\mathbf{r} + \mathbf{e} = \kappa\mathbf{W}(\mathbf{r} + \mathbf{e}) + \mathbf{e}.$$

To derive the relationship between KZ and PHP′, we define a new proximity measure KZ′ as

$$\text{KZ}'(i) = \begin{cases} \frac{\text{KZ}(i)+1}{w_{\max}}, & \text{if } i = q, \\ \frac{\text{KZ}(i)}{w_{\max}}, & \text{if } i \neq q. \end{cases}$$

Let $\mathbf{r}'$ denote the proximity vector of KZ′ when the query node is q. The relationship between KZ and KZ′ can be expressed in a matrix form as

$$\mathbf{r}' = (\mathbf{r} + \mathbf{e})/w_{\max}.$$

Then, the proximity vector of KZ′ satisfies that

$$\mathbf{r}' = \kappa\mathbf{W}\mathbf{r}' + \mathbf{e}/w_{\max}.$$

Based on this matrix form, KZ′ has the following recursive definition

$$\mathbf{r}'_i = \begin{cases} \kappa \cdot w_{\max} \sum_{j \in N_i} p_{i,j} \mathbf{r}'_j + \frac{1}{w_{\max}}, & \text{if } i = q; \\ \kappa \cdot w_{\max} \sum_{j \in N_i} p_{i,j} \mathbf{r}'_j, & \text{if } i \neq q. \end{cases}$$

Since $\kappa < 1/w_{\max}$, we have that $\kappa \cdot w_{\max} < 1$. If we set the decay factor in PHP′ as $c = \kappa \cdot w_{\max}$, KZ′ and PHP′ have the same recursive definition for any node $i \neq q$, and we have that $\frac{\text{PHP}'(i)}{\text{KZ}'(i)} = \frac{\text{PHP}'(q)}{\text{KZ}'(q)}$. Thus, $\frac{\text{PHP}'(i)}{\text{KZ}(i)} = \frac{\text{PHP}'(q)}{\text{KZ}(q)+1}$. When the query node q is fixed, we have that $\text{KZ}(i) \propto \text{PHP}'(i)$. □

Lemma 13. KZ does not have local maximum when $\kappa < 1/w_{\max}$, and has local maximum when $\kappa \geq 1/w_{\max}$.

Proof: Based on Theorem 11, when $\kappa < 1/w_{\max}$, we have that $\text{KZ}(i) \propto \text{PHP}'(i)$. We can prove that PHP′ has no local maximum. The proof is similar to that in Lemma 1. Thus KZ has no local maximum. Examples can be constructed to show that KZ has local maximum when $\kappa \geq 1/w_{\max}$, which are omitted. □

Theorem 11 says that $\text{KZ}(i)$ is proportional to $\text{PHP}'(i)$ when κ is small. Therefore, ranking by KZ is equivalent to ranking by PHP′. Comparing with PHP, PHP′ only has different transition probabilities, which are normalized by the maximum degree instead of the degree of each node as in PHP. Therefore, the FLoS method is readily applicable to PHP′.

G.4 Extension to Absorption Probability

In the absorption probability (AP) [11], with probability $p_{i,i}$, the random walker will be absorbed at node i, and with probability $1 - p_{i,i}$, the random walker will follow a random edge out of it. Let $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)$ be a diagonal matrix, where $\lambda_i > 0$ is a constant for any node $i = 1, \cdots, n$. The transition probability in AP is defined as

$$p_{i,j} = \begin{cases} \frac{\lambda_i}{\lambda_i + w_i}, & \text{if } i = j, \\ \frac{w_{i,j}}{\lambda_i + w_i}, & \text{if } i \neq j. \end{cases}$$

The AP proximity $r_{i,j}$ of node j with regard to node i is defined as the probability that a random walker starting from node i is absorbed at node j. AP can be defined recursively as

$$r_{i,j} = \begin{cases} \sum_{k \in N_i} p_{i,k} r_{k,j} + p_{i,i}, & \text{if } i = j, \\ \sum_{k \in N_i} p_{i,k} r_{k,j}, & \text{if } i \neq j. \end{cases}$$

Let $\mathbf{R}$ be the matrix of AP proximity values, i.e., $\mathbf{R}_{i,j} = r_{i,j}$. Let $\mathbf{W}$ be the adjacency matrix, and $\mathbf{D}$ be the diagonal matrix where each element $\mathbf{D}_{i,i}$ equals the degree w_i of node i. The above equations can be expressed in a matrix form

$$\mathbf{R} = (\mathbf{\Lambda} + \mathbf{D})^{-1} \mathbf{W} \mathbf{R} + (\mathbf{\Lambda} + \mathbf{D})^{-1} \mathbf{\Lambda}.$$

Thus, we have that

$$\mathbf{R} = (\mathbf{\Lambda} + \mathbf{D} - \mathbf{W})^{-1} \mathbf{\Lambda}.$$

$\mathbf{R}$ is a non-negative matrix with each row summing up to 1 [11]. The elements in the q-th row of $\mathbf{R}$ represent the AP proximity values with regard to the query q.

To extend FLoS to AP, we consider a variant of PHP, which is referred to as PHP″. PHP″ is defined as

$$\mathbf{r}_i = \begin{cases} 1, & \text{if } i = q, \\ \sum_{j \in N_i} p_{i,j} \mathbf{r}_j, & \text{if } i \neq q, \end{cases}$$

where $p_{i,j}$ is the transition probability defined in AP.

Compared with PHP, the transition probability in PHP″ is changed. In PHP, the transition probability is normalized by degree w_i. In PHP″, the transition probability is normalized by $\lambda_i + w_i$. Another difference is that there is no decay factor in PHP″. Even though we do not have decay factor in PHP″, FLoS is still applicable to PHP″. Some key observations are shown as follows.

Lemma 14. PHP″ has no local maximum.

Proof: Suppose that node i is a local maximum. We have that $\mathbf{r}_i = \sum_{j \in N_i} p_{i,j} \mathbf{r}_j \leq \sum_{j \in N_i} p_{i,j} \mathbf{r}_i = \mathbf{r}_i \cdot \frac{w_i}{\lambda_i + w_i} < \mathbf{r}_i$. We get a contradiction that $\mathbf{r}_i < \mathbf{r}_i$. □

We still use $\mathbf{P}$ to denote the transition probability matrix with

$$\mathbf{P}_{i,j} = \begin{cases} 0, & \text{if } i = q \text{ or } i = j, \\ p_{i,j}, & \text{if } i \neq q \text{ and } i \neq j. \end{cases}$$

Then the proximity $\mathbf{r}$ based on PHP″ can be written in the following matrix form

$$\mathbf{r} = \mathbf{P}\mathbf{r} + \mathbf{e}.$$

We can see that the sum of each row of $\mathbf{P}$ is smaller than 1, i.e., $\sum_j \mathbf{P}_{i,j} = \sum_j p_{i,j} = \frac{w_i}{\lambda_i + w_i} < 1$. Thus, we have that $||\mathbf{P}||_\infty < 1$. Based on this, Theorems 3, 4 and 5 are still satisfied. Thus, we can develop the lower and upper bounds in a similar way as that for PHP. Therefore, the FLoS method is applicable to PHP″.

AP and PHP″ have the following relationship.

Theorem 12. $\text{AP}(i) \propto \lambda_i \cdot \text{PHP}''(i)$

Proof: The elements in the q-th row of $\mathbf{R}$ represent the AP proximity values when the query node is q. Thus, the AP proximity vector is $\mathbf{r} = \mathbf{R}^\top \mathbf{e} = \mathbf{\Lambda}(\mathbf{\Lambda} + \mathbf{D} - \mathbf{W})^{-1} \mathbf{e}$, where $\mathbf{R}^\top$ represents the transpose of $\mathbf{R}$. Then, $(\mathbf{\Lambda} + \mathbf{D} - \mathbf{W}) \mathbf{\Lambda}^{-1} \mathbf{r} = \mathbf{e}$. Thus, the proximity vector of AP satisfies that

$$\mathbf{\Lambda}^{-1} \mathbf{r} = (\mathbf{\Lambda} + \mathbf{D})^{-1} \mathbf{W} \mathbf{\Lambda}^{-1} \mathbf{r} + (\mathbf{\Lambda} + \mathbf{D})^{-1} \mathbf{e}.$$

We define a new proximity measure AP′ as

$$\text{AP}'(i) = \frac{\text{AP}(i)}{\lambda_i},$$

for any node $i \in V$.

Let $\mathbf{r}'$ denote the proximity vector of AP′ when the query is q. The relationship between AP and AP′ can be expressed in a matrix form as $\mathbf{r}' = \mathbf{\Lambda}^{-1} \mathbf{r}$. Thus, the proximity vector of AP′ satisfies the following equation

$$\mathbf{r}' = (\mathbf{\Lambda} + \mathbf{D})^{-1} \mathbf{W} \mathbf{r}' + (\mathbf{\Lambda} + \mathbf{D})^{-1} \mathbf{e}.$$

Based on this matrix form, AP′ has the following recursive definition

$$\mathbf{r}'_i = \begin{cases} \sum_{j \in N_i} p_{i,j} \mathbf{r}'_j + \frac{1}{\lambda_i + w_i}, & \text{if } i = q, \\ \sum_{j \in N_i} p_{i,j} \mathbf{r}'_j, & \text{if } i \neq q. \end{cases}$$

AP′ and PHP″ have the same recursive definition for any node $i \neq q$. Thus, $\frac{\text{AP}'(i)}{\text{PHP}''(i)} = \frac{\text{AP}'(q)}{\text{PHP}''(q)}$. Since $\text{AP}'(i) = \frac{\text{AP}(i)}{\lambda_i}$, we have that $\frac{\text{AP}(i)}{\lambda_i \cdot \text{PHP}''(i)} = \frac{\text{AP}(q)}{\lambda_q \cdot \text{PHP}''(q)}$. When q is fixed, we have that $\text{AP}(i) \propto \lambda_i \cdot \text{PHP}''(i)$. □

Lemma 15. AP has local maximum.

Proof: Examples can be constructed to show that AP has local maximum and are omitted here. □

Suppose that node $v \in \delta S^t$ has the largest PHP″ proximity value. Based on Theorem 1, for any node $i \in \bar{S}^t$, $\text{PHP}''(i) \leq \text{PHP}''(v)$. Let $\lambda(\bar{S}^t)$ denote the maximum λ value of unvisited nodes in $\bar{S}^t$. We have that $\lambda_i \cdot \text{PHP}''(i) \leq \lambda(\bar{S}^t) \cdot \text{PHP}''(i) \leq \lambda(\bar{S}^t) \cdot \text{PHP}''(v)$. Therefore, if we maintain the

maximum λ value of unvisited nodes, we can develop the upper bound for the proximity values of unvisited nodes.

Specifically, we can apply FLoS to AP as follows. In Algorithm 3, we can change line 1 to the following line.

1: $u \leftarrow \text{argmax}_{i \in \delta S^{t-1}} \lambda_i \cdot (\underline{\mathbf{r}}_i^{t-1} + \overline{\mathbf{r}}_i^{t-1})$;

In Algorithm 6, we can change line 2 and 3 to the following two lines.

2: $K \leftarrow k$ nodes in $S^t \setminus (\delta S^t \cup \{q\})$ with largest $\lambda_i \cdot \underline{\mathbf{r}}_i^t$ values;
3: **if** $\min_{i \in K} \lambda_i \cdot \underline{\mathbf{r}}_i^t \geq \max_{i \in S^t \setminus (K \cup \{q\})} \lambda_i \cdot \overline{\mathbf{r}}_i^t$ **and** $\min_{i \in K} \lambda_i \cdot \underline{\mathbf{r}}_i^t \geq \lambda(\bar{S}^t) \cdot \max_{i \in \delta S^t} \overline{\mathbf{r}}_i^t$ **then** bStop $\leftarrow$ true;

All other processes remain the same.

APPENDIX H
EXTENSIONS TO THE REVERSE PROXIMITY MEASURES

In this subsection, we show how to apply the FLoS method to solve the top-k reverse-proximity query problem. Let $\text{RWR}_i(j)$, $\text{EI}_i(j)$, $\text{PHP}_i(j)$, $\text{DHT}_i(j)$, $\text{RT}_i(j)$, and $\text{AP}_i(j)$ denote the RWR, EI, PHP, DHT, RT, and AP proximity values of node j when the query is node i.

H.1 Extension to Reverse RWR

In this subsection, we show that ranking by the reverse RWR proximity is the same as ranking by the PHP proximity. Thus the FLoS method for PHP can be directly applied to find the top-k nodes for reverse RWR.

Theorem 13. PHP and reverse RWR give the same ranking results.

Proof: From Theorem 2, we have that PHP and EI give the same ranking results. Thus, it is equivalent to prove that EI and reverse RWR give the same ranking results.

EI is defined as the degree normalized RWR, i.e., $\text{EI}_q(i) = \frac{\text{RWR}_q(i)}{w_i}$. Thus, we have that $\text{RWR}_q(i) = w_i \cdot \text{EI}_q(i)$. Therefore, we have that $\text{RWR}_i(q) = w_q \cdot \text{EI}_i(q) = w_q \cdot \text{EI}_q(i)$, where we use the fact that EI is symmetric, i.e., $\text{EI}_i(q) = \text{EI}_q(i)$. Thus, $\text{RWR}_i(q) \propto \text{EI}_q(i)$, when the query q is fixed. This completes the proof. □

H.2 Extension to Reverse PHP and Reverse DHT

In this subsection, we first show how to solve the top-k reverse PHP problem. Then, we prove that reverse PHP and reverse DHT give the same ranking results.

To find the top-k nodes for reverse PHP, we use the following relationship between PHP and reverse PHP.

Theorem 14. $\text{PHP}_i(q) \propto \frac{\text{PHP}_q(i)}{\text{EI}_i(i)}$

Proof: From Theorem 2, we have that $\text{PHP}_q(i) = \frac{\text{EI}_q(i)}{\text{EI}_q(q)}$. Thus, $\text{PHP}_i(q) = \frac{\text{EI}_i(q)}{\text{EI}_i(i)} = \frac{\text{EI}_q(i)}{\text{EI}_i(i)} = \text{EI}_q(q) \cdot \frac{\text{PHP}_q(i)}{\text{EI}_i(i)}$, where we use the fact that EI is symmetric, i.e., $\text{EI}_i(q) = \text{EI}_q(i)$. Therefore, we have that $\text{PHP}_i(q) \propto \frac{\text{PHP}_q(i)}{\text{EI}_i(i)}$ when the query node q is fixed. □

Suppose that we already pre-compute the values $\text{EI}_i(i)$ for each node i, and node $v \in \delta S^t$ has the largest PHP proximity value. Based on Theorem 1, for any node $i \in \bar{S}^t$, $\text{PHP}_q(i) \leq \text{PHP}_q(v)$. Let $\text{EI}(\bar{S}^t)$ denote the minimum $\text{EI}_i(i)$ value of unvisited nodes i in $\bar{S}^t$. We have that $\frac{\text{PHP}_q(i)}{\text{EI}_i(i)} \leq \frac{\text{PHP}_q(i)}{\text{EI}(\bar{S}^t)} \leq \frac{\text{PHP}_q(v)}{\text{EI}(\bar{S}^t)}$. Therefore, if we maintain the minimum $\text{EI}_i(i)$ value of unvisited nodes, we can develop the upper bound for the proximity values of unvisited nodes.

Specifically, we can apply FLoS to reverse PHP as follows. In Algorithm 3, we can change line 1 to the following line.

1: $u \leftarrow \text{argmax}_{i \in \delta S^{t-1}} \frac{1}{\text{EI}_i(i)} \cdot (\underline{\mathbf{r}}_i^{t-1} + \overline{\mathbf{r}}_i^{t-1})$;

In Algorithm 6, we can change line 2 and 3 to the following two lines.

2: $K \leftarrow k$ nodes in $S^t \setminus (\delta S^t \cup \{q\})$ with largest $\frac{\underline{\mathbf{r}}_i^t}{\text{EI}_i(i)}$ values;
3: **if** $\min_{i \in K} \frac{\underline{\mathbf{r}}_i^t}{\text{EI}_i(i)} \geq \max_{i \in S^t \setminus (K \cup \{q\})} \frac{\overline{\mathbf{r}}_i^t}{\text{EI}_i(i)}$ **and** $\min_{i \in K} \frac{\underline{\mathbf{r}}_i^t}{\text{EI}_i(i)} \geq (\text{EI}(\bar{S}^t))^{-1} \cdot \max_{i \in \delta S^t} \overline{\mathbf{r}}_i^t$ **then** bStop $\leftarrow$ true;

All other processes remain the same.

Theorem 15. Reverse PHP and reverse DHT give the same ranking results.

Proof: From Theorem 2, DHT and PHP have the relationship $\text{DHT}_q(i) = \frac{1}{1-c}(1 - \text{PHP}_q(i))$. Thus, $\text{DHT}_i(q) = \frac{1}{1-c}(1 - \text{PHP}_i(q))$. It means that reverse DHT is a linear function of reverse PHP. Thus reverse DHT and reverse PHP give the same ranking results. □

H.3 Extension to Reverse RT

In this subsection, we show how to find the top-k nodes for reverse RT.

Theorem 16. $\text{RT}_i(q) \propto w_i^{1-\beta} \cdot \text{PHP}_q(i)$

Proof: Let $\text{RTF}_i(j)$ represent the forward proximity value defined in the direction from i to j. Thus, we have that $\text{RTF}_q(i) = \mathbf{x}_i$, where $\mathbf{x}_i$ is defined in Appendix G.2. $\text{RTF}_q(i)$ is exactly the RWR proximity value, i.e., $\text{RTF}_q(i) = \text{RWR}_q(i)$. Let $\text{RTB}_i(j)$ represent the backward proximity value defined in the direction from j to i. Thus, we have that $\text{RTB}_q(i) = \mathbf{y}_i$, where $\mathbf{y}_i$ is defined in Appendix G.2. $\text{RTB}_q(i)$ has a linear relationship with the EI proximity value, i.e., $\text{RTB}_q(i) = w_q \cdot \text{EI}_q(i)$.

RT can be decomposed as

$$\text{RT}_q(i) \propto \text{RTF}_q(i)^{\beta} \cdot \text{RTB}_q(i)^{1-\beta}.$$

Reverse RT can be decomposed as

$$\begin{aligned} \text{RT}_i(q) &\propto \text{RTF}_i(q)^{\beta} \cdot \text{RTB}_i(q)^{1-\beta} \\ &\propto \text{RWR}_i(q)^{\beta} \cdot w_i^{1-\beta} \cdot \text{EI}_i(q)^{1-\beta}. \end{aligned}$$

From Theorem 13, we have that $\text{RWR}_i(q) \propto \text{EI}_q(i)$. Since EI is symmetric, we have that $\text{EI}_i(q) = \text{EI}_q(i)$. From Theorem 2, we have that $\text{EI}_q(i) \propto \text{PHP}_q(i)$. Thus, we have that $\text{RT}_i(q) \propto w_i^{1-\beta} \cdot \text{PHP}_q(i)$. □

The relationship between PHP and reverse RT in Theorem 16 is quite similar to the relationship between PHP and RT in Theorem 10. We only need to change β to $(1-\beta)$ in the FLoS method for RT to apply it for reverse RT.

H.4 Extension to Reverse AP

In this subsection, we show how to find the top-k nodes for reverse AP.

Theorem 17. PHP″ and reverse AP give the same ranking results.

Proof: In Appendix G.4, we have that the proximity matrix of AP is $\mathbf{R} = (\mathbf{\Lambda} + \mathbf{D} - \mathbf{W})^{-1}\mathbf{\Lambda}$. The elements in the q-th column of $\mathbf{R}$ represent the reverse AP proximity values when the query node is q. Thus, the reverse AP proximity

vector is $\mathbf{r}=\mathbf{R}\mathbf{e}=(\mathbf{\Lambda}+\mathbf{D}-\mathbf{W})^{-1}\mathbf{\Lambda}\mathbf{e}$. Then, $(\mathbf{\Lambda}+\mathbf{D}-\mathbf{W})\mathbf{r}=\mathbf{\Lambda}\mathbf{e}$. Thus, the proximity vector of reverse AP satisfies the following equation

$$\mathbf{r}=(\mathbf{\Lambda}+\mathbf{D})^{-1}\mathbf{W}\mathbf{r}+(\mathbf{\Lambda}+\mathbf{D})^{-1}\mathbf{\Lambda}\mathbf{e}.$$

Based on this matrix form, reverse AP has the following recursive definition

$$\mathbf{r}_i=\begin{cases}\sum_{j\in N_i}p_{i,j}\mathbf{r}_j+\frac{\lambda_i}{\lambda_i+w_i}, & \text{if } i=q,\\ \sum_{j\in N_i}p_{i,j}\mathbf{r}_j, & \text{if } i\neq q,\end{cases}$$

where $p_{i,j}=\frac{w_{i,j}}{\lambda_i+w_i}$ is the transition probability from node i to j.

PHP″ and reverse AP have the same recursive definition for any node $i\neq q$, thus $\frac{\mathrm{AP}_i(q)}{\mathrm{PHP}''_q(i)}=\frac{\mathrm{AP}_q(q)}{\mathrm{PHP}''_q(q)}$. When the query node q is fixed, $\mathrm{AP}_i(q)\propto\mathrm{PHP}''_q(i)$. This completes the proof. □

EI and KZ both are symmetric, thus the top-k results for the reverse proximity are the same as the top-k results for the original proximity.

APPENDIX I
COMPARISON OF CASTANET AND FLOS

Castanet [4] is an improved global iteration method, which needs to iterate over the entire graph and compute the exact proximity of each node. Castanet uses the developed bounds to identify the top-k nodes earlier thus reduce the number of iterations required by the global iteration method.

The bounds in Castanet are based on the following probability distributions. Starting from the query node, the random surfer randomly walks to the neighbor nodes following the transition probabilities. Let p_i^t represent the probability that the random surfer is at node i at time t. We have that $\sum_{i\in V}p_i^t=1$ for any t. Initially, $p_q^0=1$ and $p_i^0=0$ if $i\neq q$.

At each iteration t, Castanet computes the probability p_i^t and uses p_i^t to refine the lower and upper bounds of node i. We can see that, at each iteration t, Castanet only uses the probability distribution at time t to refine the lower and upper bounds. However, at each iteration t, FLoS uses the probability distributions at all time points to refine the bounds. This is the key difference between the bounds in Castanet and FLoS and the reason why the bounds in FLoS are more effective than the bounds in Castanet.

APPENDIX J
SIMRANK

SimRank is a classic random walk based proximity measure [31]. It is based on the intuition that two nodes are similar if their neighbors are similar. The SimRank value between two nodes measures the expected number of steps required before two random surfers, one starting from each node, meet at the same node if they walk in lock-step.

Lemma 16. SimRank has local maximum.

Proof: Examples can be constructed to show that SimRank has local maximum, which are omitted. □

The random walk processes in SimRank and PHP are quite different. In SimRank, the random walk process involves two random surfers. In PHP, the random walk process only involves one random surfer. There is no direct relationship between SimRank and PHP. Thus, the FLoS algorithm is not applicable to SimRank at this time. We will investigate how to apply FLoS to SimRank in our future work.