This is the Appendix of the following paper.

Y. Wu, R. Jin, J. Li, and X. Zhang. Robust local community detection: on free rider effect and its elimination. *PVLDB*, 8(7):798-809, 2015.

11. APPENDIX

11.1 The Proofs in Section 3.3

The proof of Theorem 3 is as follows.

PROOF. Since f is supermodular, we have that $f(S)+f(S_l^*) \leq f(S \cap S_l^*)+f(S \cup S_l^*)$. Since $G[S_l^*]$ is the local optimal subgraph, we have that $f(S_l^*) \geq f(S \cap S_l^*)$. Combining these two inequalities, we have that $f(S) \leq f(S \cup S_l^*)$. □

The proof of Theorem 4 is as follows.

PROOF. Let $f'(S, \lambda) = g(S) - \lambda h(S)$, where λ is a real-valued constant. Let $\lambda = f(S)$. Proving that $f(S \cup S_l^*) \geq f(S)$ is equivalent to proving that $f'(S \cup S_l^*, \lambda) \geq 0$. Note that $g(S) - \lambda h(S) = 0$ since $\lambda = f(S)$. We have that

$$\begin{aligned} & f'(S \cup S_l^*, \lambda) = g(S \cup S_l^*) - \lambda h(S \cup S_l^*) \\ \geq\; & g(S) + g(S_l^*) - g(S \cap S_l^*) - \lambda(h(S) + h(S_l^*) - h(S \cap S_l^*)) \\ =\; & g(S_l^*) - g(S \cap S_l^*) - \lambda(h(S_l^*) - h(S \cap S_l^*)). \end{aligned}$$

Since g is monotonically increasing, $g(S_l^*) - g(S \cap S_l^*) \geq 0$. If $h(S_l^*) - h(S \cap S_l^*) \leq 0$, we have that $f'(S \cup S_l^*, \lambda) \geq 0$. Now suppose that $h(S_l^*) - h(S \cap S_l^*) > 0$. Since $G[S_l^*]$ is a local optimal subgraph, we have that $f(S_l^*) \geq f(S \cap S_l^*)$. Thus we can derive that

$$\frac{g(S_l^*) - g(S \cap S_l^*)}{h(S_l^*) - h(S \cap S_l^*)} \geq \frac{g(S_l^*)}{h(S_l^*)} = f(S_l^*) \geq \lambda.$$

So, we have that $f'(S \cup S_l^*, \lambda) \geq 0$, i.e., $f(S \cup S_l^*) \geq f(S)$. □

The proof of Lemma 6 is as follows.

PROOF. If we merge $G[S_l^*]$ into $G[S]$, its minimum degree will not decrease. □

The proof of Lemma 7 is as follows.

PROOF. Since $S \cap S_l^* = \emptyset$, we have that $e(S) + e(S_l^*) \leq e(S \cup S_l^*)$. Since $f(S_l^*) = e(S_l^*) - \alpha\binom{|S_l^*|}{2} \geq \alpha \cdot |S| \cdot |S_l^*| = \alpha[\binom{|S|+|S_l^*|}{2} - \binom{|S|}{2} - \binom{|S_l^*|}{2}]$, we have that $e(S_l^*) \geq \alpha\binom{|S|+|S_l^*|}{2} - \alpha\binom{|S|}{2}$. Combining these two inequalities, we have that $e(S) - \alpha\binom{|S|}{2} \leq e(S \cup S_l^*) - \alpha\binom{|S|+|S_l^*|}{2}$ thus $f(S) \leq f(S \cup S_l^*)$. □

The proof of Lemma 8 is as follows.

PROOF. Since $S_l^* \cap (S \cup \delta\overline{S})) = \emptyset$, $\delta(S \cup S_l^*) = \delta S \cup \delta S_l^*$ and $\delta S \cap \delta S_l^* = \emptyset$. Thus, we have that $e(\delta(S \cup S_l^*), S \cup S_l^*) = e(\delta S, S) + e(\delta S_l^*, S_l^*)$ and $e(\delta(S \cup S_l^*), V) = e(\delta S, V) + e(\delta S_l^*, V)$. So, $f(S \cup S_l^*) = \frac{e(\delta S, S) + e(\delta S_l^*, S_l^*)}{e(\delta S, V) + e(\delta S_l^*, V)} \geq f(S)$. □

The proof of Lemma 9 is as follows.

PROOF. Since $\phi(S \cup S_l^*) \leq \phi(V)/2$, subgraphs $G[S]$, $G[S_l^*]$ and $G[S \cup S_l^*]$ all have smaller volumes than the complements. In this case, external conductance degenerates to $f(S) = \frac{e(S,\overline{S})}{\phi(S)} = \frac{e(S,\overline{S})}{2e(S)+e(S,\overline{S})}$. Minimizing $f(S)$ is equivalent to maximizing $f'(S) = \frac{e(S)}{e(S,\overline{S})}$, which is subgraph modularity. So we have that $f'(S) \leq f'(S \cup S_l^*)$ and $f(S) \geq f(S \cup S_l^*)$. □

11.2 The Proof of Theorem 6

PROOF. Let $G[T]$ denote one connected component of the densest subgraph $G[S]$. We have that $\rho(T) \leq \rho(S)$. We also have that $\rho(T) \geq \rho(S)$, since otherwise $\rho(S \setminus T) > \rho(S)$. Thus $G[T]$ has equal density with $G[S]$. Suppose that $G[T]$ does not contain any query node.

First, let $u \in T$ be the node with the smallest value $\frac{\pi(u)}{w(u)}$, i.e., $\forall v \in N_u \cap T$, $\frac{\pi(v)}{w(v)} \geq \frac{\pi(u)}{w(u)}$. Since $w(v) \geq w(u,v)$, we have that $\forall v \in N_u \cap T, \pi(v) \geq \frac{w(u,v)}{w(u)} \cdot \pi(u)$. There exists a node $y \in N_u \cap \overline{T}$ such that $\pi(y) < \frac{w(u,y)}{w(u)} \cdot \pi(u)$. Otherwise, node u violates Lemma 10.

Next, we will prove that $\rho(T \cup \{y\}) > \rho(T)$. We have that $\frac{w(u)}{\pi(u)} \geq \frac{e(T)}{\pi(T)}$ because otherwise $\rho(T \setminus \{u\}) > \rho(T)$. Since $\pi(y) < \frac{w(u,y)}{w(u)} \cdot \pi(u)$, we have that $\frac{w(u,y)}{\pi(y)} > \frac{w(u)}{\pi(u)} \geq \frac{e(T)}{\pi(T)}$. So, we have that $\rho(T \cup \{y\}) \geq \frac{e(T)+w(u,y)}{\pi(T)+\pi(y)} > \rho(T)$.

In conclusion, if $G[T]$ does not contain any query node, there must exist a node in $\delta\overline{T}$, whose addition increases the density. This contradicts the assumption that $G[T]$ has the largest density. This completes the proof. □

11.3 The Proofs of Theorems 7 and 8

Before we prove Theorems 7 and 8, we first define a new proximity measure, PHP$''$, as follows.

$$r(u) = \begin{cases} c\sum_{v \in N_u} \frac{w(u,v)}{w_{\max}} r(v) + 1, & \text{if } u = q; \\ c\sum_{v \in N_u} \frac{w(u,v)}{w_{\max}} r(v), & \text{if } u \neq q. \end{cases}$$

Let PHP(u) and PHP$''(u)$ be the PHP and PHP$''$ proximity values of node u with regard to the query node q. PHP and PHP$''$ have the following relationship.

LEMMA 13. $\text{PHP}(u) = \frac{\text{PHP}''(u)}{\text{PHP}''(q)}$

PROOF. We can see that PHP and PHP$''$ have the same recursive equation for any node $u \neq q$. Thus we have that $\frac{\text{PHP}(u)}{\text{PHP}''(u)} = \frac{\text{PHP}(q)}{\text{PHP}''(q)}$. This completes the proof. □

Based on this, we can derive the PHP proximity matrix. Let $\mathbf{W}$ be the adjacency matrix with $\mathbf{W}_{u,v} = w(u,v)$, $\mathbf{P} = \mathbf{W}/w_{\max}$ be the transition probability matrix, $\mathbf{r}$ be the PHP$''$ proximity vector, and $\mathbf{1}_q$ be a vector with only one non-zero element $\mathbf{1}_q(q) = 1$. The recursive equation of PHP$''$ can be expressed as $\mathbf{r} = c\mathbf{P}\mathbf{r} + \mathbf{1}_q$. Thus the PHP$''$ proximity vector is $\mathbf{r} = (\mathbf{I} - c\mathbf{P})^{-1}\mathbf{1}_q$. So, the PHP proximity vector is $\mathbf{r} = \frac{1}{\text{PHP}''(q)}(\mathbf{I} - c\mathbf{P})^{-1}\mathbf{1}_q$ from Lemma 13. Let $\mathbf{\Lambda}$ be a diagonal matrix with $\mathbf{\Lambda}_{u,u} = (\text{PHP}_u''(u))^{-1}$, where PHP$_u''(u)$ denotes the PHP$''$ proximity value of node u when the query is u. Then, the PHP proximity matrix is

$$\mathbf{R} = (\mathbf{I} - c\mathbf{P})^{-1}\mathbf{\Lambda},$$

and the qth column of $\mathbf{R}$ represents the PHP proximity vector when the query is node q.

Suppose that the query node q belongs to a local community $G[S]$. Recall that the conductance of $G[S]$ is defined as

$$\frac{e(S, \overline{S})}{\min\{\phi(S), \phi(\overline{S})\}}.$$

When the volume $\phi(S) < \phi(V)/2$, the conductance value is equal to $e(S, \overline{S})/\phi(S)$.

Let $\mathbf{d}$ be a vector with $\mathbf{d}(u)=w(u)$ for any node $u\in V$. Let $\mathbf{d}_S$ be a vector with $\mathbf{d}_S(u)=w(u)$ if $u\in S$, and $\mathbf{d}_S(u)=0$ otherwise. Let $\mathbf{1}$ be a vector whose elements all are 1. Let $\mathbf{1}_S$ be a vector with $\mathbf{1}_S(u)=1$ if $u\in S$, and $\mathbf{1}_S(u)=0$ otherwise. Let $\mathbf{1}_S^{\mathrm{T}}$ be the transpose of $\mathbf{1}_S$. Let $\mathbf{R}$ be the proximity matrix with $\mathbf{R}_{u,v}$ denoting the PHP proximity value of node u when the query is node v.

LEMMA 14. *Suppose that $\phi(S)<\phi(V)/2$. We have that $\mathbf{1}_S^{\mathrm{T}}\mathbf{P}^k\mathbf{d}_S\le k\cdot e(S,\overline{S})$, for any integer k.*

PROOF. Since $\mathbf{1}_S^{\mathrm{T}}\mathbf{P}\mathbf{d}_S=\mathbf{1}_S^{\mathrm{T}}\mathbf{W}\mathbf{d}_S/w_{\max}\le\mathbf{1}_S^{\mathrm{T}}\mathbf{W}\mathbf{1}_S=e(S,\overline{S})$, this lemma holds when $k=1$.

Suppose that it holds for $k=i$. By induction, it suffices to show it holds for $k=i+1$. Let $\mathbf{x}=\mathbf{P}^i\mathbf{d}_S$, and $\mathbf{x}_S(u)=\mathbf{x}(u)$ if $u\in S$ and $\mathbf{x}_S(u)=0$ otherwise. We have that

$$\mathbf{1}_S^{\mathrm{T}}\mathbf{P}^{i+1}\mathbf{d}_S=\mathbf{1}_S^{\mathrm{T}}\mathbf{P}\mathbf{x}=\mathbf{1}_S^{\mathrm{T}}\mathbf{P}(\mathbf{x}_S+\mathbf{x}_{\overline{S}})\le\mathbf{1}_S^{\mathrm{T}}\mathbf{P}\mathbf{x}_S+\mathbf{1}_S^{\mathrm{T}}\mathbf{x}.$$

Since $\mathbf{d}_S\le\mathbf{d}$, $\mathbf{P}\mathbf{d}_S\le\mathbf{P}\mathbf{d}=\mathbf{W}\mathbf{d}/w_{\max}\le\mathbf{W}\mathbf{1}=\mathbf{d}$. Thus, we have that $\mathbf{P}^i\mathbf{d}_S\le\mathbf{d}$. So, $\mathbf{x}_S\le\mathbf{d}_S$ because $\mathbf{d}_S$ equals $\mathbf{d}$ restricted on S. Therefore, $\mathbf{1}_S^{\mathrm{T}}\mathbf{P}\mathbf{x}_S\le\mathbf{1}_S^{\mathrm{T}}\mathbf{P}\mathbf{d}_S\le e(S,\overline{S})$. Thus, $\mathbf{1}_S^{\mathrm{T}}\mathbf{P}^{i+1}\mathbf{d}_S\le e(S,\overline{S})+i\cdot e(S,\overline{S})=(i+1)e(S,\overline{S})$. □

Next we show the proof of Theorem 7.

PROOF. If we randomly pick a node from $G[S]$ with a probability proportional to its degree and use this node as the query node, the expected value of $r(\overline{S})$ can be expressed as $\frac{1}{\phi(S)}\mathbf{1}_S^{\mathrm{T}}\mathbf{R}\mathbf{d}_S$. Thus, it is equivalient to prove that $\mathbf{1}_S^{\mathrm{T}}\mathbf{R}\mathbf{d}_S\le\frac{c}{(1-c)^2}e(S,\overline{S})$.

For any node u, since $\mathrm{PHP}''_u(u)\ge 1$, $\mathbf{\Lambda_{u,u}}\le 1$. Thus,

$$\begin{aligned}\mathbf{1}_S^{\mathrm{T}}\mathbf{R}\mathbf{d}_S&=\mathbf{1}_S^{\mathrm{T}}(\mathbf{I}-c\mathbf{P})^{-1}\mathbf{\Lambda}\mathbf{d}_S\le\mathbf{1}_S^{\mathrm{T}}(\mathbf{I}-c\mathbf{P})^{-1}\mathbf{d}_S\\&=\mathbf{1}_S^{\mathrm{T}}\textstyle\sum_{k=0}^{\infty}(c\mathbf{P})^k\mathbf{d}_S=\sum_{k=0}^{\infty}c^k\mathbf{1}_S^{\mathrm{T}}\mathbf{P}^k\mathbf{d}_S\\&\le\textstyle\sum_{k=1}^{\infty}c^k k\cdot e(S,\overline{S})=\frac{c}{(1-c)^2}e(S,\overline{S}),\end{aligned}$$

where the fifth step is based on Lemma 14. □

The proof of Theorem 8 is as follows.

PROOF. First, we prove that the query biased densest subgraph does not contain any node in L. For any node $v\in L$, $w'(v)=\frac{w(v)}{\pi(v)}<\frac{e(S)}{|S|\cdot\pi(v)}<\frac{e(S)}{\pi(S)}=\rho(S)$. The query biased densest subgraph must have a density larger than $\rho(S)$. We can prove that any node in the densest subgraph must have query biased degree greater than $\rho(S)$. Thus, any node $u\in L$ does not belong to the densest subgraph.

Since $c<(N_{\max})^{-1}$, the query biased densest subgraph is connected and containing the query node based on Theorem 6. So, if a node y in S_l^* belongs to the densest subgraph, there must exist a path in the densest subgraph linking y to one query node. From the assumption, this path contains at least one node $v\in L$. However, we already prove that the nodes in L do not belong to the optimal subgraph. Thus, the local optimal subgraph $G[S_l^*]$ cannot exist in the densest subgraph. □

11.4 The Proof of Theorem 9

PROOF. The QDC problem can be reduced from the set cover problem [16]. Let $X=\{X_1,\cdots,X_l\}$ be a family of sets with $Y=\{y_1,\cdots,y_k\}=\bigcup_{i=1}^{l}X_i$ being the elements. The set cover problem consists of finding a minimum subset $X_{\mathrm{opt}}\subseteq X$, such that each element y_j is contained in at least one set in X_{opt}.

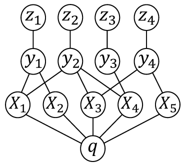


Figure 14: Graph construction from an instance of the set cover problem

The graph is constructed as follows. For each element $y_j\in Y$, we create two nodes y_j and z_j. Let node set $Z=\{z_1,\cdots,z_k\}$. Let node set $V=\{q\}\cup X\cup Y\cup Z$. Node q is connected to any node $X_i\in X$. Node $y_j\in Y$ is connected to node X_i if $y_j\in X_i$ in the set cover problem. Node $z_j\in Z$ is connected to node $y_j\in Y$. The edge (y_j,z_j) has weight $2+\frac{1}{k}+\epsilon$, where $0<\epsilon<\frac{1}{k^2}$ is a real number. The edges incident to X_i have equal weight $\frac{1}{|X_i|+1}$. Let the set of query nodes $Q=\{q\}\cup Q'$, where $Q'\subseteq Y\cup Z$. If Q' is empty, there is one query node q. The node weight $\pi(u)=1,\forall u\in V$.

Figure 14 gives an example of the graph constructed from an instance of the set cover problem with $X_1=\{y_1,y_2\}$, $X_2=\{y_1\},X_3=\{y_2,y_4\},X_4=\{y_2,y_3\}$, and $X_5=\{y_4\}$.

Consider a subgraph $G[S]$ which contains Q ($Q\subseteq S$) and is connected. We must have that $S=\{q\}\cup X'\cup Y'\cup Z'$, where $X'\subseteq X$, $Y'\subseteq Y$, $Z'\subseteq Z$, and $|Y'|\ge|Z'|$.

Suppose that at least one node z_j does not exist in S, i.e., $|Z'|\le k-1$. Let $S'=\{q\}\cup X'\cup Y'$. We have that $e(S')\le|X'|$ since the node degree $w(X_i)=1$. Consider the density $\rho(S)=\frac{\sum_{z_j\in Z'}(2+\frac{1}{k}+\epsilon)+e(S')}{|X'|+|Y'|+|Z'|+1}\le\frac{2|Z'|+|X'|+(\frac{1}{k}+\epsilon)|Z'|}{2|Z'|+|X'|+1}$. We have that $(\frac{1}{k}+\epsilon)|Z'|<(\frac{1}{k}+\frac{1}{k^2})(k-1)<1$. Thus $\rho(S)<1$.

Suppose that all the nodes Z exist in S. The density $\rho(S)=\frac{\sum_{z_j\in Z}(2+\frac{1}{k}+\epsilon)+|X'|}{2k+|X'|+1}>\frac{(2+\frac{1}{k})k+|X'|}{2k+|X'|+1}=1$. So, the subgraph containing Z always has larger density than that containing a proper subset of Z. Thus we only consider the subgraph containing Z in the following.

Next, let $X'\subseteq X$ be a feasible solution to the set cover problem. Let $S=\{q\}\cup X'\cup Y\cup Z$. Then $G[S]$ is connected and $\rho(S)=1+\frac{\sum_{z_j\in Z}(2+\frac{1}{k}+\epsilon)-2k-1}{2k+|X'|+1}$. We have that $\sum_{z_j\in Z}(2+\frac{1}{k}+\epsilon)-2k-1>(2+\frac{1}{k})k-2k-1=0$. So $\rho(S)$ is monotonically decreasing with regard to $|X'|$. Since $|X_{\mathrm{opt}}|\le|X'|$, the subgraph induced from $S=\{q\}\cup X_{\mathrm{opt}}\cup Y\cup Z$ contains the query nodes, has the largest density, and is connected.

So, X_{opt} solves the set cover problem, and the subgraph induced from $S=\{q\}\cup X_{\mathrm{opt}}\cup Y\cup Z$ solves the QDC problem.

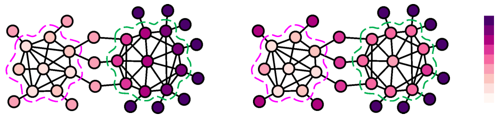
Continue with our example in Figure 14. $X_{\mathrm{opt}}=\{X_1,X_3,X_4\}$ solves the set cover problem. The subgraph induced from $S=\{q\}\cup X_{\mathrm{opt}}\cup Y\cup Z$ solves the QDC problem. □

11.5 Why is PHP better than RWR and PHP′?

Even though the PHP, RWR, and PHP′ are all random walk based proximity measures, there are some subtle differences. Here we discuss why PHP has better performance than RWR and PHP′.

11.5.1 PHP and RWR

Random walk with restart (also known as personalized PageRank) [35] is a widely used proximity measure. RWR can be described as follows. From a node u, the random walker can walk to its neighbors with probabilities proportional to the edge weights. In each step, it has a probability



(a) PHP (b) RWR

Figure 15: Comparison between PHP and RWR (The query node is the central node in the community on the left)

of c to return to the query node q, where c is a constant. The proximity of node u w.r.t. q is defined as the stationary probability that the random walker will finally stay at u. RWR can be defined recursively as

$$r(u) = \begin{cases} (1-c)\sum_{v\in N_u} \frac{w(v,u)}{w(v)} r(v) + c, & \text{if } u = q; \\ (1-c)\sum_{v\in N_u} \frac{w(v,u)}{w(v)} r(v), & \text{if } u \neq q, \end{cases}$$

where c $(0 < c < 1)$ is the constant restart probability.

The key reason why PHP performs better than RWR is that PHP does not have local maximum while RWR does. Thus RWR may favor the high degree nodes in irrelevant subgraphs.

Definition 4. [No Local Maximum] *A proximity measure has no local maximum if for any node $u \in V \setminus Q$, there exists a neighbor node v of u (i.e., $v \in N_u$), such that $r(v) > r(u)$.*

Lemma 15. *PHP has no local maximum.*

Proof. Suppose that node u is a local maximum. Thus, $r(u) = c\sum_{v\in N_u} \frac{w(u,v)}{w_{\max}} r(v) \leq c\sum_{v\in N_u} \frac{w(u,v)}{w_{\max}} r(u) = \frac{c\cdot w(u)}{w_{\max}} r(u) < r(u)$. We get a contradiction that $r(u) < r(u)$. □

No local maximum property in Definition 4 says that for any non-query node u, there exists a neighbor node v of u, i.e., $v \in N_u$, such that node v has larger proximity value than node u. If one proximity measure has no local maximum, the maximum proximity value in the boundary of a set of nodes S containing all the query nodes is the upper bound of the proximity values of nodes in $\overline{S}$. This is shown in the following Theorem.

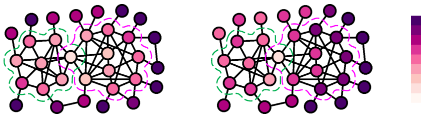
Theorem 10. *Let S be a node set containing all the query nodes, and u be the node with the largest proximity in δS. If a proximity has no local maximum, we have that $r(u) > r(v)$ $(\forall v \in \overline{S})$.*

Proof. Suppose otherwise. We have that $\exists v \in \overline{S}$, such that $r(u) \leq r(v)$. Now suppose that node y is the node with the largest proximity in $\overline{S}$. We have that $\forall z \in \overline{S} \cup \delta S$, $r(y) \geq r(z)$. The neighbors of node y must exist in $\overline{S} \cup \delta S$, i.e., $N_y \subseteq \overline{S} \cup \delta S$. Therefore, we have $r(y) \geq r(z)$ $(\forall z \in N_y)$, which means node y is a local maximum. This contradicts the assumption. □

Based on Theorem 10, for a given local community $G[S]$, any node outside $G[S]$ has smaller proximity value than the maximum proximity value in the boundary δS. This shows that the nodes outside $G[S]$ generally have small proximity values than the nodes in $G[S]$.

In contrast, RWR has local maxima. Thus the high degree nodes in the irrelevant subgraph may have large RWR proximity values. This is an undesirable property of RWR.

Figure 15(a) shows the node weight distribution of PHP with decay factor $c = 0.9$, where the query node is the central node in the left community. PHP has no local maximum as shown in Lemma 15. Figure 15(b) shows the node weight distribution of RWR with restart probability $c = 0.1$. We can see that RWR has local maxima, i.e., the central node in the right community has larger RWR proximity value thus smaller node weight than all its neighbors. In RWR, the nodes with large degree in the irrelevant subgraph will have large proximity values, thus low node weights. Therefore, if the nodes with large degree also compose a dense subgraph, it is likely that this irrelevant subgraph has large query biased density and becomes the free rider subgraph.



(a) PHP (b) PHP′

Figure 16: Comparision between PHP and PHP′ (The query node is the overlapping node of the two communities)

11.5.2 PHP and PHP′

The degree normalized penalized hitting probability (PHP′) penalizes the random walk for each additional step. PHP′ can be defined recursively as

$$r(u) = \begin{cases} 1, & \text{if } u = q; \\ c\sum_{v\in N_u} \frac{w(u,v)}{w(u)} r(v), & \text{if } u \neq q, \end{cases}$$

where c $(0 < c < 1)$ is the decay factor in the random walk process.

In PHP′, the transition probability is normalized by the node degree. Therefore, the large degree node will have small transition probabilities to its neighbors, thus the probability of hitting the query node will be small. When we use PHP′ to weight the nodes, a dense subgraph will get small query biased density if it contains many large degree nodes. On the other hand, in PHP, the transition probability is normalized by the maximum degree, which is a constant value. So, the hitting probability of the large degree node will not be degraded. When we use PHP to weight the nodes, a dense subgraph still has high query biased density even it contains large degree nodes. This is the key reason why PHP performs better than PHP′.

Let us use the example in Figure 16 to further explain why PHP performs better than PHP′. The example graph in Figure 16(a) contains two communities, which have one overlapping node. We use this overlapping node as the query node. Figures 16(a) and 16(b) show the node weight distributions when using PHP and PHP′ respectively. The decay factor $c = 0.9$. Intuitively, the nodes in the right community should have larger proximity values than the nodes in the left community, since the nodes in the right community are more densely connected to the query node than the nodes in the left community are. However, from Figure 16(b), we can see that using PHP′, the nodes in the left community have smaller node weights (larger proximity values) than the nodes in the right community. The reason is that the nodes in the right community have large degrees, which reduce their hitting probabilities. The query biased densities of the left and right communities are 2.04 and 1.66 respectively when using PHP′. This means that the left community has larger query biased density than the right one does. In Figure 16(a), we can see that the nodes in the right community have smaller node weights (larger proximity values) than the nodes in the left community. The query biased densities of the left and right communities are 0.41 and 0.45 respectively when using PHP to weight the nodes.

11.6 Why does QDC perform better than LS?

Here we provide more detailed explanations on why our QDC method outperforms the LS method [25]. In LS, the objective is to minimize the conductance $e(S,\overline{S})/(\phi(S)\cdot\phi(\overline{S}))$.

The following lemma says that if (1) any pair of nodes u and v, where u is in the target local community and v is in the irrelevant local optimal subgraph, are at least two hops away from each other, (2) the local optimal subgraph has smaller conductance than the target local community, and (3) the volume of the whole graph is large enough, the goodness metric $f(S) = e(S,\overline{S})/(\phi(S)\cdot\phi(\overline{S}))$ causes the local free rider effect.

LEMMA 16. *If $S_l^* \cap (S \cup \delta\overline{S}) = \emptyset$, $\frac{e(S,\overline{S})}{\phi(S)} > \frac{e(S_l^*,\overline{S_l^*})}{\phi(S_l^*)}$, and*

$$\phi(V) \geq \phi(S) + \frac{e(S,\overline{S})\phi(S_l^*)(\phi(S)+\phi(S_l^*))}{e(S,\overline{S})\phi(S_l^*) - e(S_l^*,\overline{S_l^*})\phi(S)},$$

the conductance defined in LS satisfies that $f(S) \geq f(S \cup S_l^*)$.

PROOF. Let $T = S_l^*$. Since $\frac{e(S,\overline{S})}{\phi(S)} > \frac{e(T,\overline{T})}{\phi(T)}$, we have that $e(S,\overline{S})\phi(T) - e(T,\overline{T})\phi(S) > 0$. Then, from the third condition in the lemma, we have that

$$\frac{e(S,\overline{S})}{\phi(S)\phi(\overline{S})} \geq \frac{e(S,\overline{S}) + e(T,\overline{T})}{(\phi(S)+\phi(T))(\phi(V)-\phi(S)-\phi(T))}.$$

Since $T \cap (S \cup \delta\overline{S}) = \emptyset$, we have that $\phi(S \cup T) = \phi(S) + \phi(T)$ and $e(S \cup T, \overline{S \cup T}) = e(S,\overline{S}) + e(T,\overline{T})$. Thus, we have that

$$\frac{e(S,\overline{S})}{\phi(S)\phi(\overline{S})} \geq \frac{e(S \cup T,\overline{S \cup T})}{\phi(S \cup T)\phi(\overline{S \cup T})}.$$

This completes the proof. □

The LS method has an additional constraint which requires the correlation between the solution vector and the input preference vector to be greater than a threshold. The intension of this constraint is to force the solution subgraph to be near the query node. However, this problem formulation still suffers from the local free rider effect.

Let's first consider the problem formulation [25]. Let $G[S]$ be the target local community that contains the set of query nodes. Let $G[S_l^*]$ be a local optimal subgraph. Suppose that $G[S]$ and $G[S_l^*]$ satisfy the conditions in Lemma 16. Then, $G[S \cup S_l^*]$ has smaller goodness value than $G[S]$. Let $\mathbf{x}_S$ and $\mathbf{x}_{S \cup S_l^*}$ denote the solution vectors of the local spectral optimization program corresponding to the solution subgraphs $G[S]$ and $G[S \cup S_l^*]$ respectively. Since both $G[S]$ and $G[S_l^*]$ are small subgraphs, $\mathbf{x}_S$ and $\mathbf{x}_{S \cup S_l^*}$ may both have high correlation with the preference vector. So, both $\mathbf{x}_S$ and $\mathbf{x}_{S \cup S_l^*}$ satisfy the constraint with a threshold. However, $G[S \cup S_l^*]$ has smaller goodness value than $G[S]$. Then, $G[S \cup S_l^*]$ is a better solution subgraph. Thus, the problem formulation still suffers from the local free rider effect.

To solve the problem, the algorithm developed in [25] contains two steps. First, it computes the vector $\mathbf{y}$ with regard to the set of query nodes. The vector $\mathbf{y}$ is similar to the RWR proximity vector with regard to the same set of query nodes. Second, for each $i = 1, \cdots, n$, it computes the conductance value of the set S_i composed of the nodes corresponding to the first i elements in $\mathbf{y}$ with largest values, and selects the set S_j with the minimum conductance value. We can see that this procedure is similar to the algorithm in the PRN method [2]. Thus, the LS and PRN methods have similar performance as shown in Table 5. They both suffer from the free rider effect.

In conclusion, the constraint with a threshold in the local spectral problem formulation cannot eliminate the free rider effect. The algorithm in [25] is similar to that of PRN and still suffers from the free rider effect.