A Proofs

Proof for Theorem 1.

Proof (sketch) Removing u from A_{w_s} and D_{w_s} only affects the labeling of vertex u. Considering any vertex u_x ($u \neq u_x$) in G, there are two cases:

1) $\overline{uu_x}$ does not pass through w_s, meaning that $\overline{uu_x}$ must be covered by another 2-hop center. Removing u from A_{w_s} and D_{w_s} does not affect this path.

2) $\overline{uu_x}$ passes through w_s, which means that $\overline{uu_x} = \overline{uw_s} + \overline{w_s u_x}$. Since $\overline{uw_s}$ passes through w_b, w_b also covers the path $\overline{uu_x}$. Therefore, $\exists w_b, w_b \in L_{out}(u) \wedge w_b \in L_{in}(u_x) \wedge Dist_{sp}(u, u_x) = Dist_{sp}(u, w_b) + Dist_{sp}(w_b, u_x)$. It also means that removing u from A_{w_s} and D_{w_s} does not affect this path.

To summary, removing u from A_{w_s} and D_{w_s} does not affect the completeness of 2-hop labeling.

Proof for Theorem 3.

Proof (sketch) If $\exists j, u_i \notin Area(R_j)$, according to Definition 4, it means that there exists no vertex u_j having the same label of v_j, where $L_\infty(u_i, u_j) \leq \delta$. According to Theorem 2, $L_\infty(u_i, u_j)$ is a lower bound for $Dist_{sp}(u_i, u_j) \leq \delta$. Thus, there exists no vertex u_j labeled the same as v_j, where $Dist_{sp}(u_i, u_j) \leq \delta$. It also means that u_i cannot match v (in Q), since there exist no vertex that can match v_j, where v_j is v's neighbor. Therefore, u_i can be pruned safely.

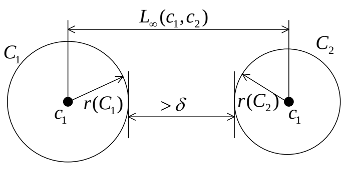

Fig. 29 Proof for Theorem 5

Proof for Theorem 5.

Proof (sketch) Given any two vertices u_1 and u_2 from C_1 and C_2, respectively, we prove that $L_\infty(u_1, u_2) > \delta$ (i.e., join result is empty) if $L_\infty(c_1, c_2) > r(C_1) + r(C_2) + \delta$. Figure 29 demonstrates the proof process.

Since L_∞ is a metric distance, it satisfies triangle inequality. $L_\infty(u_1, u_2) > L_\infty(c_1, u_2) - L_\infty(c_1, u_1)$ and $L_\infty(c_1, u_2) > L_\infty(c_1, c_2) - L_\infty(c_2, u_2)$. Therefore, we know $L_\infty(u_1, u_2) > r(C_1) + r(C_2) + \delta - L_\infty(c_1, u_1) - L_\infty(c_2, u_2)$. Obviously, $r(C_1) > L_\infty(c_1, u_1)$ and $r(C_2) > L_\infty(c_2, u_2)$. Consequently, $L_\infty(u_1, u_2) > \delta$. □

Proof for Theorem 6.

Proof Since L_∞ is a metric distance, it satisfies the triangle inequality. Given a point q in C_1, if q can be joined with p, then $L_\infty(pq) \leq \delta$.

1) According to the triangle inequality, we know $\delta \geq L_\infty(p, q) > L_\infty(p, c_1) - L_\infty(q, c_1)$. Thus, we can conclude that $L_\infty(p, c_1) - \delta \leq L_\infty(q, c_1)$.

2) Also, according to the triangle inequality, we know that $L_\infty(q, c_1) < L_\infty(c_1, p) + L_\infty(p, q) \leq L_\infty(c_1, p) + \delta$. Consequently, we know $L_\infty(q, c_1) < L_\infty(c_1, p) + \delta$.

3) Since q is a point in C_1, $0 \leq L_\infty(q, c_1) \leq r(C_1)$.

According to 1), 2) and 3), we have $Max(L_\infty(p, c_1) - \delta, 0) \leq L_\infty(q, c_1) \leq Min(L_\infty(p, c_1) + \delta, r(C_1))$.□

B Complexity Analysis of Offline Processing

B.1 Time Complexity

According to the framework in Section 3, the whole offline processing includes two steps: computing 2-hop labels and graph embedding. The time complexity of offline processing is $O(|E(G)|\ |V(G)| + |V(G)|^2\ log|V(G)|)$, computed as follows:

Computing 2-hop labels. Algorithm 1 computes 2-hop labels. Line 1 is Dijkstra's algorithm whose time complexity is $O(k|E(G)| + k|V(G)|\ log|V(G)|)$. We partition the graph into n blocks in $O(E(G))$ [19], where each block has $|S_i|$ vertices (Lines 2-4). Thus, there are $O(|S_i|^2)$ pairwise shortest paths in each block. We adopt the greedy method in [9] to compute local 2-hop labeling in $O(|S_i|^2)$ time (Line 6). Since $|S_i| < |V(G)|$ and n is a constant, the time complexity of Lines 5-6 is $O(|V(G)|^2)$. For each vertex $u \in W_b \cup W_s$, Dijkstra's algorithm is performed to compute skeleton 2-hop labeling. Therefore, Lines 7-14 need $O(|V(G)||E(G)| + |V(G)|^2\ log|V(G)|)$ time. Consequently, the total time complexity of computing 2-hop labels is $O(|V(G)||E(G)| + |V(G)|^2\ log|V(G)|)$.

Graph Embedding. There are ($\sum_{n=1,\ldots,\beta; m=1,\ldots,\kappa} |S_{n,m}|$) selected vertices, where $\sum_{n=1,\ldots,\beta; m=1,\ldots,\kappa} |S_{n,m}| = O(|V(G)|)$. For each selected vertex, we employ Dijkstra's algorithm. Thus, the time complexity of graph embedding is $O(|E(G)||V(G)| + |V(G)|^2\ log|V(G)|)$.

B.2 Space Complexity

The space cost of 2-hop labeling is $O(|V(G)||E(G)|^{\frac{1}{2}})$ [7]. According to graph embedding, each vertex is mapped into a point in $\Re^k$ space, where $k = O(log^2|V(G)|)$. Thus, the space cost of graph embedding is $O(|V(G)|\ log^2|V(G)|)$. Therefore, the total space cost of offline processing is $O(|V(G)||E(G)|^{\frac{1}{2}} + |V(G)|\ log^2|V(G)|)$.