Spatial-Aware Community Search Over Heterogeneous Information Networks

Yi Zhou(✉), Lihua Zhou, Jialong Wang, Lizhen Wang, and Bing Kong

School of Information Science and Engineering, Yunnan University, Kunming 650091, Yunnan, People's Republic of China
zhyi4635@163.com

Abstract. The prosperity of smart mobile devices and the popularity of social networks have led to the rapid growth of spatial social networks. Spatial-aware community search aims to look for a cohesive subgraph that contains a query vertex in spatial social networks, whose vertices are close structurally and spatially. However, existing studies only focus on homogeneous graphs, and ignore the heterogeneity of the networks, which results in the searched community is not refined enough to meet the specific applications of scenarios. In this paper, we propose a novel problem, named *spatial-aware community search over a heterogeneous information network* (SACS-HIN), which retrieves a refined community by capturing rich semantics in the network, taking into account spatial proximity and social relevance. To solve this problem, we develop three algorithms based on the structure-first strategy and distance-first strategy. Finally, extensive experiments are conducted on four datasets to evaluate both the effectiveness and efficiency of our proposed algorithms. The community size analysis and case study verify that the proposed algorithms can obtain a refined community that satisfies query conditions. The efficiency evaluation explores the effect of different parameters on the efficiency of the algorithms.

Keywords: Heterogeneous Information Network · Community Search · Spatial-Aware · Search Strategy

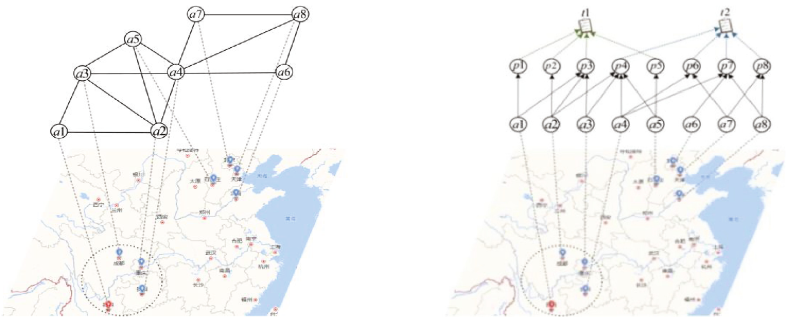
1 Introduction

Community search [1], as an important research problem in social network analysis, aims to search for a cohesive subgraph that contains a query vertex, while spatial-aware community search additionally considers the actual distance between vertices based on community search, requiring the target community not only needs to be structurally cohesive but also expects the spatial positions of the vertices to be close to each other. A large amount of location information contained in spatial social networks reveals people's various life patterns, laws, and preferences, which can be used to find a more refined community and to provide users with location-based activity planning and new applications [2], bringing convenience to life.

X. Meng et al. (Eds.): SpatialDI 2023, LNCS 13887, pp. 103–114, 2023.
https://doi.org/10.1007/978-3-031-32910-4_7

Spatial community search [3–5] has been studied by many researchers, but all these studies focused on homogeneous information networks with one type of vertices. Compared with homogeneous information networks, heterogeneous information networks (HINs) express richer semantic information and provide a more complete, natural, accurate, as well as detailed description of complex networks.

Figure 1(a) illustrates a homogeneous network consisting of authors with locations, extracted from DBLP, and Fig. 1(b) is a spatial HIN with three types of vertices, extracted from DBLP, namely, author (A), paper (P), and topic (T), which contains the semantic relationships among vertices of different types. For example, the authors a_1, a_2, and a_3 have written a paper p_3, which mentions the topic t_1.



(a) Homogeneous network (b) Heterogeneous network

Fig. 1. The network instance of DBLP

Spatial community search over HINs can have many applications in real life. For example, suppose authors $a_1 a_1$ want to organize a local workshop, we use the model $k-$core with $k = 2$ to ensure participants are required to be acquainted with at least two people in the community, the three authors $\{a_1, a_2, a_3\}$ is a good result since their research expertise is relevant and their locations are close, the relationship between them is defined by the meta-path $a_1 p t_1 p a$. However, in Fig. 1(a), the result set will be $\{a_1, a_2, a_3, a_4\}$, although author a_4 isn't the researcher of topic t_1 as other authors. From the above example, we can see that the community of $\{a_1, a_2, a_3\}$ is more refined than $\{a_1, a_2, a_3, a_4\}$ and will better meet the requirements of users.

In this paper, we study the problem of spatial-aware community search over heterogeneous information network, aims to search for a community that is both structurally cohesive and spatially adjacent in networks that contains multi-type objects and complex relationships, which can provide a feasible solution for offline social activities such as academic seminars and team collaboration. To achieve this purpose, we present a structure-first strategy and a distance-first strategy, and develop an algorithm named ReaFirst based on the structure-first strategy, and two algorithm called DistFirst and FastDist based on the distance-first search strategy, and in FastDist, a labeled batch search in introduced to optimize the search efficiency further.

The main contributions of this paper are as follows: (1) A spatial-aware community search problem SACS-HIN in heterogeneous information networks is proposed, and a formal definition is given for the proposed problem. (2) For the SACS-HIN problem, two search strategies, structure-first and distance-first, were designed respectively. In each strategy, the search was from the two stages of structurally cohesive and spatial proximity, and we propose the ReaFirst algorithm, the DistFirst algorithm, and the Fast-Dist algorithm. (3) We conduct extensive experiments on four datasets to evaluate the effectiveness and efficiency of our algorithms.

The remained paper is organized as follows: Sect. 2 introduces the related work; Sect. 3 elaborates some basic concepts and gives a formal problem definition; Sect. 4 presents three algorithms proposed based on two strategies; Sect. 5 describes the experimental preparation and settings, and reports the experimental results; Sect. 6 summarizes the paper.

2 Related Work

Existing community search methods focus on different types of networks and find communities in different ways. Fang et al. [6] gave a comprehensive overview of relevant community search work in homogeneous information networks, including spatial-aware community search [4], and temporal network community search [7] and influential community search [8]. We mainly introduce the research work related to this paper from two aspects: spatial-aware community search over homogeneous information networks and community search over HINs.

A large number of spatial-aware community search studies exist in homogeneous information networks. David et al. [9] showed that users' social network relationships are influenced by spatial location information through numerous experiments. Sozio [10] who proposed a community search problem around the scenario of how to plan a successful cocktail party. Zhu et al. [11] considered rectangular spatial window, relaxed nearest neighbor, and strict nearest neighbor constraint to solve the problem of GSGQ. Wang et al. [12] used a radius-bounded circle to limit the user's position in the spatial social network. Fang et al. [4] searched for a spatial-aware community (SAC) in a large spatial network, determined a minimum covering circle (MCC) through three points on the boundary. Although these researches achieved many results, whether it is based on a rectangle, based on a bounded circle, or based on a minimum coverage circle, which are essentially spatial community search based on the homogeneous information network.

HINs can fit better with real-life complex networks, so relevant research in recent years has begun to extend to HINs [13]. Wang et al. [14] investigated an online technique that can match user pairs in different social networks in a short time. Fang et al. [15] used meta-paths to define a series of relationships between different types of vertices in HINs, proposed the concept of $(k, \mathcal{P})$-core, and designed the FastBCore algorithm with good search performance. Qiao et al. [16] conducted research on keyword-centric community search in large HINs. Jiang et al. [17] performed an effective community search in a star-schema HIN, solved the limitation of requiring users to specify meta-paths and relational constraints for users unfamiliar with HINs. These community search methods all capture the rich semantics embedded among multiple types of objects in HINs, but they all do not take into account the actual spatial distance of the objects and still have a large gap with the real world.

Synthesizing the above analysis, most of the methods either ignore the heterogeneity of the network or the spatial distance between community members, and such community search results have limitations for many scenarios. To this end, we proposed a study of spatial-aware community search in HINs to mine the cohesive structural groups and spatial proximity groups in user groups and location circles, respectively, which can obtain the best community that meets the query requirements.

3 Basic Concepts and Problem Definition

In this paper, we introduce some concepts of HINs and community search and then give a specific definition for the problem of spatial-aware community search over the heterogeneous information network.

3.1 Basic Concepts

Definition 1: Heterogeneous Information Networks (HIN) [18]. An HIN is a directed graph $G = (V, E)$ composing of a vertex set V and an edge set E, the vertex type mapping function is $\varphi : V \to \mathcal{A}$ and the edge type mapping function is $\phi : E \to \mathcal{R}$. Where each vertex $v \in V$ belongs to a vertex type $\varphi(v) \in \mathcal{A}$, and each edge $e \in E$ belongs to an edge type (also known as a relation) $\phi(e) \in \mathcal{R}$. A network is called an HIN if $|\mathcal{A}| + |\mathcal{R}| > 1$.

Definition 2: Network Schema [18]. Given a HIN $G = (V, E)$ with vertex-type mapping $\varphi : V \to \mathcal{A}$ and edge-type mapping $\phi : E \to \mathcal{R}$, its schema T_G is a directed graph defined over vertex types $\mathcal{A}$ and edge types (relations) $\mathcal{R}$, namely $T_G = (\mathcal{A}, \mathcal{R})$.

Definition 3: Meta-path [18]. A meta-path $\mathcal{P}$ is a path defined on a heterogeneous information network schema $T_G = (\mathcal{A}, \mathcal{R})$ and represented as $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} A_{l+1}$, where $L = |\mathcal{P}|$ denotes the length of meta-path $\mathcal{P}$,$\mathcal{A}_i \in \mathcal{A}(1 \le i \le L+1)$,$\mathcal{R}_j \in \mathcal{R}(1 \le j \le L)$.

Definition 4: k-core [19]. Given an integer $k(k > 0)$, the $k-$core of G is the largest subgraph of G, in which the degree of each vertex v is no less than k.

3.2 Problem Statement

Given a HIN $G = (V, E)$, a non-negative integer k, a query vertex q and a maximum user spatial distance constraint d, the SACS-HIN returns a community $C \subseteq V$ satisfying the following constraints: (1) Connectivity: C is connected and needs to contain q; (2) Structure cohesiveness: The minimum degree of nodes in C is no less than k; (3) Spatial proximity: $\forall v \in C, d(v, q) \le d$; Note that a vertex $v \in V$ of the user type in the network has a location $(v.x, v.y)$, where $v.x$ and $v.y$ represent its position along the x-axis and y-axis in two-dimensional space.

4 SACS-HIN Search Algorithms

According to the definition of the SACS-HIN problem, two effective algorithms (ReaFirst, DistFirst) and one efficient algorithm (FastDist) are designed in this section based on the structure-first strategy and the distance-first strategy, respectively.

4.1 The ReaFirst Algorithm

Based on the structure-first search strategy, this section proposes the ReaFirst algorithm as shown in Algorithm 1. The main idea of this algorithm is: we start with a query vertex, search for a maximum subgraph that satisfies $(k, \mathcal{P})-$core as a candidate subgraph. Then search the subgraph with the spatial location in the range of d from the candidate subgraph. Finally, the set of vertices will be returned as the resultant community.

Algorithm 1: ReaFirst

Input: G , q , $\mathcal{P}$, k , d ;
Output: C ;
Initialize $C \leftarrow \varnothing$, $S \leftarrow \{q\}$;
add all the vertices with the same type as q to S ;
foreach vertex $v \in S$ **do**
  delete the vertices in S whose $\deg(v) < k$;
  if $d(v,q) > d$ **then** remove v from S ;
  foreach $u \in S$ **do**
    if u is the $\mathcal{P}$*-neighbor* of v **then** remove v from $\mathcal{P}$*-neighbor*$[u]$;
    update $\deg(v)$;
$C \leftarrow$ remove all the vertices that are not $\mathcal{P}$*-connected* with q from S ;
return C ;

The time complexity of Algorithm 1 is $\mathcal{O}(n_1 \times d_{1,2} + n_1 \sum_{i=1}^{L} n_i \times d_{i,i+1})$, where n_i represents the number of vertices corresponding to the i-th vertex type of the meta-path $\mathcal{P}$, and $d_{i,i+1}$ denotes the maximum number of vertices corresponding to the $(i + 1)$-th vertex type of the meta-path $\mathcal{P}$, which are connected to the vertices corresponding to the i-th vertex type.

4.2 The DistFirst Algorithm

Algorithm 1 considers the structural cohesiveness of the community in the first stage, in which there exist a part of distant vertices. After removing these over-range distant vertices in the second stage, their degrees of these vertices will be affected and become no longer satisfied with k, so it is necessary to recalculate $k-$core. To optimize this step, this section considers dealing with the distant vertices first and proposes the DistFirst algorithm shown in Algorithm 2.

Algorithm 2: DistFirst
Input: G , q , $\mathcal{P}$, k , d ;
Output: C ;
1 Initialize $C \leftarrow \varnothing$, $S \leftarrow \{q\}$;
2 add all the vertices with the same type as q to S ;
3 **foreach** vertex $v \in S$ **do**
4   if $d(v,q) > d$ **then** remove v from S ;
5   **foreach** $u \in S$ **do**
6   **if** u is the $\mathcal{P}$-*neighbor* of v **then** remove v from $\mathcal{P}$-*neighbor*$[u]$;
7   update $\deg(v)$;
8   delete the vertices in S whose $\deg(v) < k$;
9 $C \leftarrow$ remove all the vertices that are not $\mathcal{P}$-*connected* with q from S ;
10 return C ;

First, initialize the target community C, and the vertex set S containing the query vertex (line 1). Then, in the first stage, all vertices with the target type are added to the set of vertices S. Vertices outside the specified range d are iteratively removed, and these vertices are removed from the set of their neighbors (lines 2–6); based on this, the second stage is performed, where the degree of each vertex is updated, and vertices not satisfying k are removed under the condition that the spatial proximity constraint is satisfied (lines 7–8), and the final S is the community searched by the algorithm (lines 9–10). The time cost of Algorithm 2 is $\mathcal{O}(n_1 \times d_{1,2} + n_1 \sum_{i=1}^{L} n_i \times d_{i,i+1})$.

4.3 The FastDist Algorithm

Inspired by Algorithm 2, we incorporate the labeled batch search strategy (BSL) based on the distance-first approach and propose a fast search algorithm FastDist as shown in Algorithm 3.

Algorithm 3: FastDist
Input: G , q , $\mathcal{P}$, k , d ;
Output: C ;
Initialize $C \leftarrow \varnothing$, $S \leftarrow \{q\}$;
for $i \leftarrow 1$ to L **do**;
  $Y \leftarrow \varnothing$;
  for each vertex $v \in S$ **do**
    for each neighbor u of v **do**
    if (v,u) matches with i-th edge of $\mathcal{P}$ **then**
      if (v,u) does not have a lable i **then**
        Y.add(v) and attach a lable i to (v,u) ;
$S \leftarrow Y$;
foreach vertex $v \in S$ **do**
  if $d(v,q) > d$ **then** remove v from S ;
    foreach $u \in S$ **do**
      if u is the $\mathcal{P}$-*neighbor* of v **then** remove v from $\mathcal{P}$-*neighbor*$[u]$;
      update $\deg(v)$;
      delete the vertices in S whose $\deg(v) < k$;
$C \leftarrow$ remove all the vertices that are not $\mathcal{P}$-*connected* with q from S ;
return C ;

Similarly, first, initialize the target community C, and the vertex set S containing the query vertices (line 1). Use the BSL strategy to search for all $\mathcal{P}-$neighbor vertices of the query vertex q and assign them to S (lines 2–9); in the first stage, vertices outside the specified range d are iteratively deleted, and these vertices are removed from their neighbor vertex sets (lines 10–13). Then, the second stage is performed by updating the degree of each vertex and removing the vertices that do not satisfy k under the condition that the spatial proximity constraint is satisfied (lines 14–15), and the remaining S is the community searched by the algorithm (lines 16–17). The total time cost of Algorithm 3 is $\mathcal{O}(d_{1,2} + \sum_{i=2}^{L} n_i \times d_{i,i+1})$.

5 Experiments

5.1 Experiment Settings

Datasets. In this paper, experiments are conducted on four datasets of HINs, and the details of the datasets are shown in Table 1. Among them, DBLP [20] is a co-author network constructed by research papers published in the field of computer science. SDBLP contains real communities and is a small dataset extracted from DBLP for the case study in Sect. 5.2. The Foursquare [21], containing check-in records of U.S. cities, is extracted from a service website based on user's location information. IMDB [22] contains rating records of 1,648,434 movies.

In addition, since users in DBLP, Foursquare, and IMDB are without specific location information, this paper uses a random number generated within a specified interval as the spatial location information of each user.

Table 1. Statistical information about the dataset

Dataset	Vertices	Edges	Vertex Type	Edge Type	Meta-paths
SDBLP	37791	170794	4	3	12
DBLP	682819	1951209	4	3	12
Foursquare	43199	405476	5	4	20
IMDB	2467806	7597591	4	3	12

Evaluation Indicators and Parameters. The ranges of the parameters involved in the experiments and their default values are shown in Table 2. k denotes the minimum degree required to be satisfied by a vertex, d is the minimum distance between two objects, c is the maximum number of core, and n represents the percentage size of the subgraph drawn from the original dataset. When a parameter is changed, the rest of the parameters are set to their default values. When testing the scalability, the percentage size from 20% to 100% of the original dataset are randomly selected by varying n, with the default value fixed at 100%. Note that to avoid accidental extreme values affecting the accuracy of the experimental results, the data are the average results after 100 tests.

Table 2. Summary of Parameters

Parameter	Range	Default
c	4, 8, 12, 16, 20	4
k	4, 8, 12, 16, 20	4
d	50, 150, 250, 350, 450	350
n	20%, 40%, 60%, 80%, 100%	100%

5.2 Effectiveness Evaluation

In order to evaluate the effectiveness of the SACS-HIN problem, this section compares the proposed algorithm with the HomeBCore [15] algorithm that does not consider user's location information and then presents a case study on SDBLP.

Community Size Analysis. Figure 2 reports the experimental results compared with the HomeBCore algorithm that does not consider the user's geographic location information. Note that since the community results calculated by our proposed three algorithms are as same, in this paper, we only show one result computed by the ReaFirst algorithm to compare with HomeBCore. From the experimental results, we can see that under the same parameter settings, the community results obtained by considering user locations are all smaller in size than the community results that do not involve user locations. It's meaningful to explore the impact of the user's spatial location on the resultant community in community search, which will find scenario-appropriate and more accurate communities for the query.

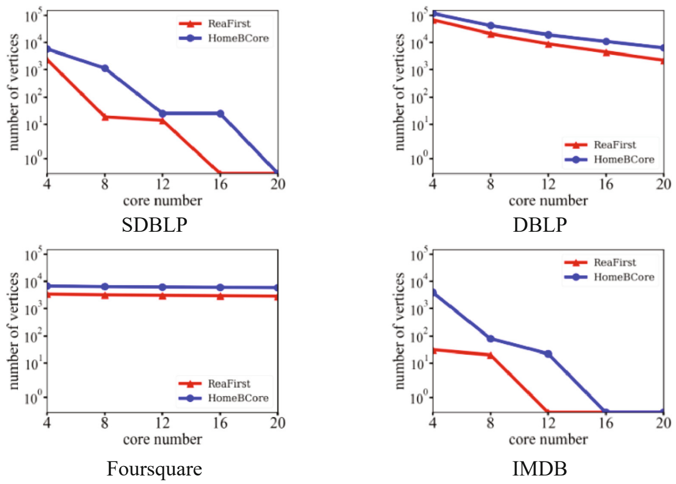
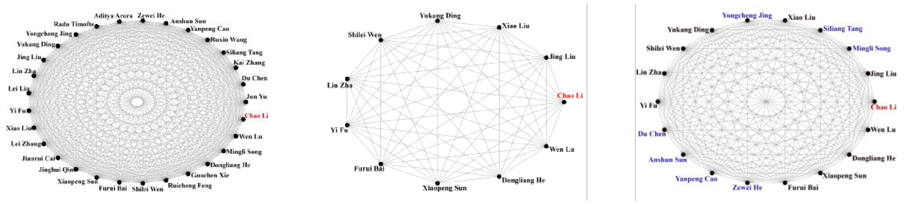


Fig. 2. Core size distributions

Case Study. Suppose a famous scholar plans to organize a local offline workshop. He hopes that the participants will have a certain social connection with him and be close to the area where he is located. Based on the description of the case, the search for eligible participants with Chao Li as the organizer is reported in Fig. 3 for the unspecified range (Fig. 3(a)), the specified range of 50 km (Fig. 3(b)), and 200 km (Fig. 3(c)) settings, respectively.



(a) without distance (b) d=50km (c) d=200km

Fig. 3. Case study on SDBLP with distance (q = ChaoLi, $k = 4$)

It is observed that without considering the spatial proximity of users, 29 scholars will be invited to the conference. When the distance range is specified as 50 km, 11 scholars are found by the organizer. When the distance range was expanded four times to 200 km, the number of eligible participants found increased to 18. As can be seen, considering the spatial proximity of users in a specific scenario is beneficial for identifying more accurate communities.

5.3 Efficiency Evaluation

The three algorithms proposed in this paper, ReaFirst, DistFirst, and FastDist, all involve parameters k, d, and n, therefore, this section conducts the following three sets of experiments on the four data sets mentioned above to test the effect of several parameters on the efficiency of the algorithms.

The Influence of Parameter k on the Efficiency of the Algorithm. The variation of the running time with k for the three algorithms on different datasets is reported in Fig. 4. It can be seen that the runtime of the ReaFirst algorithm is consistently higher than the other two algorithms on all datasets, which is due to a lot of time consumed in constructing an induced homogeneous graph for all vertices matching the target type.

The Influence of Parameter d on the Efficiency of the Algorithm. The variation of the running time with d for the three algorithms on different datasets is reported in Fig. 5. It can be seen from the results that the running time of the FastDist algorithm is always the least, the performance of the DistFirst algorithm is second, and the time required for the ReaFirst algorithm is the most. To a certain extent, the running time consumption of the algorithm mostly shows an increasing trend as the parameter d rising on different data sets. This is because under the relaxed distance constraint, the range of query object-based lookups is consequently expanded, and thus the required running time increases.

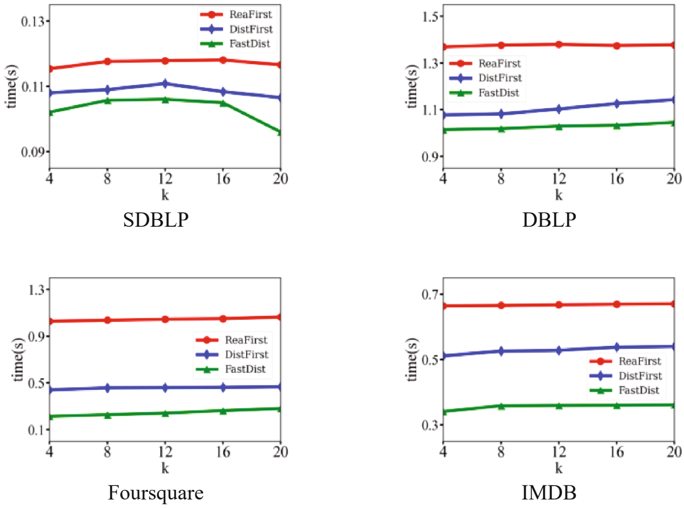


Fig. 4. The effect of varying k

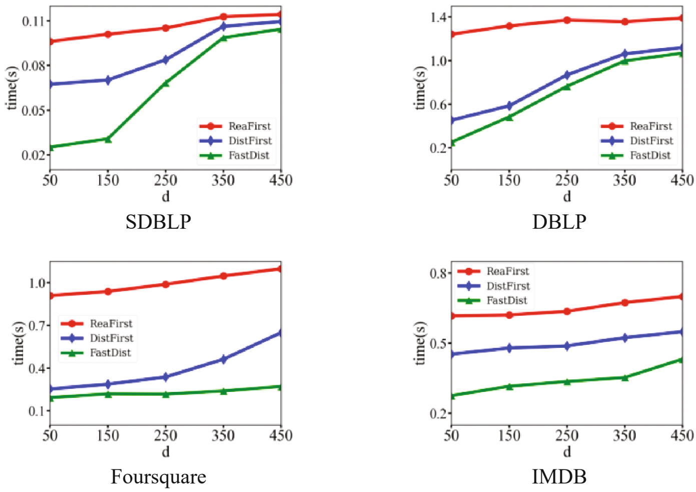


Fig. 5. The effect of varying d

The Influence of Parameter n on the Efficiency of the Algorithm. The variation of the running time with n for the three algorithms on different datasets is reported in Fig. 6. It can be seen that all algorithms scale well with the number of vertices. Among algorithms based on priority judgment distance, FastDist is always the best. In particular, the FastDist algorithm performs better when the dataset is larger. Even in the million-scale dataset IMDB, FastDist can complete in a very short time.

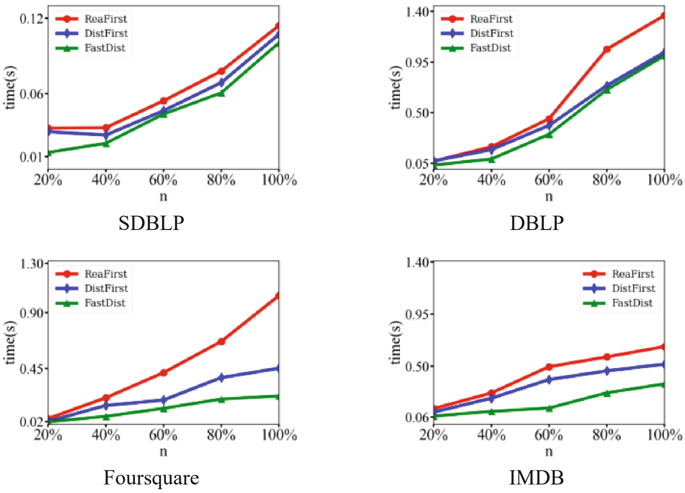


Fig. 6. The effect of varying *n*

6 Conclusion

In this paper, we study the problem of SACS-HIN and proposed two search strategies, structure-first and distance-first, which are designed for three algorithms. The experimental results on four HINs show that the proposed solutions are effective and efficient for searching communities on HINs, and a real application case is given. In the future, we will study how to conduct a more efficient heterogeneous information network community search while incorporating time.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (62062066, 61762090, 61966036), Yunnan Fundamental Research Projects (202201AS070015), the Scientific Research Fund Project of Yunnan Provincial Education Department (2023Y0249), and the Postgraduate Research and Innovation Foundation of Yunnan University (2021Y024).

References

1. Huang, X., Lakshmanan, L., Xu, J.: Community Search over Big Graphs, vol. 14, pp. 1–206 (2019)
2. Ghosh, B., Ali, M.E., Choudhury, F.M., Apon, S.H., Sellis, T., Li, J.: The flexible socio spatial group queries. Proc. VLDB Endow. **12**, 99–111 (2018)
3. Al-Baghdadi, A., Lian, X.: Topic-based community search over spatial-social networks. Proc. VLDB Endow. **13**, 2104–2117 (2020)
4. Fang, Y., Cheng, R., Li, X., Luo, S., Hu, J.: Effective community search over large spatial graphs. Proc. VLDB Endow. **10**, 709–720 (2017)
5. Guo, F., Yuan, Y., Wang, G., Zhao, X., Sun, H.: Multi-attributed community search in road-social networks. In: 2021 IEEE 37th International Conference on Data Engineering (ICDE), pp. 109–120 (2021)

6. Fang, Y., et al.: A survey of community search over big graphs. VLDB J. **29**, 353–392 (2019)
7. Li, R.H., Su, J., Qin, L., Yu, J.X., Dai, Q.: Persistent community search in temporal networks, pp. 797–808 (2018)
8. Bi, F., Chang, L., Lin, X., Zhang, W.: An optimal and progressive approach to online search of top-k influential communities. In: Very Large Data Bases (2017)
9. Liben-Nowell, D., Novak, J., Kumar, R., Raghavan, P., Tomkins, A.: Geographic routing in social networks. Proc. Natl. Acad. Sci. U. S. A. **102** (2005)
10. Sozio, M., Gionis, A.: The community-search problem and how to plan a successful cocktail party. In: Knowledge Discovery and Data Mining (2010)
11. Zhu, Q., Hu, H., Xu, C., Xu, J., Lee, W.C.: Geo-social group queries with minimum acquaintance constraints (2017)
12. Kai, W., Xin, C., Lin, X., Zhang, W., Lu, Q.: Efficient computing of radius-bounded k-cores. In: 2018 IEEE 34th International Conference on Data Engineering (ICDE) (2018)
13. Shi, C., Li, Y., Zhang, J., Sun, Y., Yu, P.S.: A survey of heterogeneous information network analysis. IEEE Trans. Knowl. Data Eng. **29**, 17–37 (2017)
14. Wang, Z., Yuan, Y., Zhou, X., Qin, H.: Effective and efficient community search in directed graphs across heterogeneous social networks. In: Borovica-Gajic, R., Qi, J., Wang, W. (eds.) ADC 2020. LNCS, vol. 12008, pp. 161–172. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-39469-1_13
15. Fang, Y., Yang, Y., Zhang, W., Lin, X., Cao, X.: Effective and efficient community search over large heterogeneous information networks. Proc. VLDB Endow. **13**, 854–867 (2020)
16. Qiao, L., Zhang, Z., Yuan, Y., Chen, C., Wang, G.: Keyword-centric community search over large heterogeneous information networks. In: Jensen, C.S., et al. (eds.) DASFAA 2021. LNCS, vol. 12681, pp. 158–173. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-73194-6_12
17. Jiang, Y., Fang, Y., Ma, C., Cao, X., Li, C.: Effective community search over large star-schema heterogeneous information networks. Proc. VLDB Endow. **15**, 2307–2320 (2022)
18. Sun, Y., Han, J., Yan, X., Yu, P.S., Wu, T.: PathSim: meta path-based top-k similarity search in heterogeneous information networks **4**, 992–1003 (2011)
19. Seidman, S.B.: Network structure and minimum degree (1983)
20. Liu, W., Jiang, X., Pellegrini, M., Wang, X.: Discovering communities in complex networks by edge label propagation. Sci. Reports (2006)
21. Amelio, A., Pizzuti, C.: Overlapping community discovery methods: a survey. In: Gündüz-Öğüdücü, Ş, Etaner-Uyar, A.Ş (eds.) Social Networks: Analysis and Case Studies. LNSN, pp. 105–125. Springer, Vienna (2014). https://doi.org/10.1007/978-3-7091-1797-2_6
22. Bao, X., Wang, L.: A clique-based approach for co-location pattern mining. Inf. Sci. **490**, 244–264 (2019)