Community Search: A Survey of the State-of-the-Art from Algorithms to Learning, Complex Graphs, and Interaction

Longxu Sun, Xin Huang
Hong Kong Baptist University
Hong Kong, China
{cslxsun, xinhuang}@comp.hkbu.edu.hk

Abstract

Community search is a query processing task of finding query-dependent communities in graphs, which has particularly useful applications in social circle discovery, personalized recommendation, and fraud detection. Over the past years, extensive studies have been devoted to this challenging problem due to diverse application scenarios and the complexity of communities in nature. Most classical community search works follow *an algorithmic framework* by designing a dense subgraph-based model and developing offline indices to accelerate online community search. However, many recent works have studied novel problems of community search over complex graphs and extended new directions from this algorithmic framework, which unfortunately lacks a timely review. This survey reviews the state-of-the-art community search works and summarizes them in terms of three aspects as follows. First, we introduce the latest community search works, in terms of methodologies from *querying algorithms* to *machine learning models*. Second, we give a review in terms of graph types, from structural and attributed graphs to complex graphs. We focus on the studies of community search over two kinds of complex graphs: *heterogeneous information networks* and *multilayer graphs*. Third, we present a newly exciting direction of *interactive community search*, which needs human-in-loop interactions and allows users to give feedback for advising community refinement. We finally share future directions and promising opportunities in this research area of community search.

1 Introduction

The topology structure of graphs widely exists in numerous real-world scenarios, such as social networks, citation networks, e-commerce platforms, traffic transportation systems, and biomedical networks [3, 52, 59, 60, 63]. Due to various attributes associated with nodes and edges, one simple graph cannot accurately depict real relationships. Therefore, many kinds of graphs are used to comprehensively model these scenarios, including attributed graphs, heterogeneous information networks (HINs), temporal graphs, multilayer graphs, bipartite graphs, dynamic graphs, and so on [15, 19, 22, 31, 33, 43, 58, 70, 81]. Communities that are formed by a group of individual nodes, naturally exist in graphs, such as family groups, biological functional organs, event groups in social networks, and so on. The discovery of community is very useful for understanding network dynamics and user behaviors. These communities are characterized by diverse attributes and different sizes of participants, which presents challenges in developing effective algorithms and models to discover them.

The task of *community search* focuses on identifying query-oriented cohesive communities in graphs, which is determined by user-provided queries [20, 30]. Unlike *community detection*, which seeks to uncover all potential communities within a graph, community search has demonstrated fast response efficiency and powerful capabilities in various personalized search tasks [13, 55, 63]. The applications of community search include social circle discovery in social networks, personalized recommendations in e-commerce platforms, the formation of interdisciplinary expert groups in collaborative networks, drug discovery in biological networks, and fraud detection across different financial systems [13, 59, 60, 80]. These applications highlight the utility and impact of community search in enhancing user experience across multiple domains.

A wide range of community search studies [19, 28, 29] have been investigated in the literature. Fang et al. [20] provides a systematic classification of existing community search studies based on various dense subgraphs, including the k-core, k-truss, k-clique, quasi-clique, k-plex, k-edge-connectivity components, and other metrics-based densest subgraphs. Besides structural community search over simple graphs, another major contribution of this survey summarizes various works of attributed community search for different attribute types, e.g., keywords, locations, temporal, weighted influences, and profiles. However, most discussed works [20][30] had been published on or before 2020. Based on our investigation of recent studies, community search have been studied from *static and querying algorithms* to *interactive and learning-based approaches* [2, 10, 12, 25, 36, 37, 42, 43, 45, 47, 53, 69, 70, 74, 82, 83, 88]. Even more, the problems of community search have been studied from *simple and attributed networks* to *more complex graphs (e.g. HINs and multilayer graphs)* [1, 7–9, 14, 17, 26, 39, 41, 65, 73, 79, 84–87]. Thus, it is timely and important to have a state-of-the-art survey on existing community search works. In this survey, we make contributions to discuss and categorize these state-of-the-art community search works, in terms of three aspects, including community search methodologies, complex networks, and user interaction.

First, in terms of community search methodologies, we review advanced works using querying algorithms and learning-based approaches. These works apply classical querying algorithms, e.g., exact algorithms, heuristic solutions with pruning strategies, and indexing-based methods, to address attribute community search. Furthermore, we summarize recent learning-based approaches, e.g., graph neural networks and transformers, to significantly improve the quality of community search and introduce fast solutions.

Second, in terms of complex networks, we summarize recent community search works over two complex networks of HINs and multilayer graphs. Specifically, HINs consist of multiple typed nodes and edges with a given network schema. Thus, community search over HINs usually needs an input of meta-path query. On the contrary, multilayer networks consist of multiple graph layers, where each layer has one graph with internal edges and two graphs at different layers have cross-layer edges. Community search over multilayer graphs usually needs no input meta-path but focuses on the interested querying nodes.

Third, in terms of user interaction, we introduce a new direction of interactive community search, which allows users to give feedback for improving community search answers. Last but not least, we also point out interesting research directions and promising opportunities leveraging recent large language models (LLMs), including the retrieval augmented generation (RAG) improvement using community search, LLM-enhanced community search solutions, and developing a unified dense subgraph model for various community search tasks.

2 Community Search on Attributed Graphs: Algorithms and Learning

We start with a widely used graph model of attributed graphs. Then, we introduce the state-of-the-art studies of community search over attributed graphs, in terms of *algorithmic approaches* and *learning models*.

Attributed graphs naturally exist in various real-world scenarios. For instance, individual nodes are often associated with multiple attributes of hobby tags in social networks; In academic collaboration networks, author nodes tend to specialize in a few attributes of research topics; In protein-protein interaction networks, each protein node is associated with several distinct functions. Therefore, a keyword-based attributed graph consists of vertices, edges, and a set of keywords, denoted as $G(V, E, A)$, where V is the set of vertices, $E \subset V \times V$ is the set of edges, and $A = \{a_1, a_2, ..., a_n\}$ is a set of keywords associated with vertices. The objective of community search over attributed graphs (ACS) is to find query-dependent communities with cohesive structure and homogeneous attributes. As summarized in the previous survey [20], there were two representative community search works on keyword-attributed graphs, which adopt k-core model [19] and k-truss model [28] to measure a structural density of communities. However, both methods [19, 28] use algorithmic approaches to solve community search problems. After an investigation of recent studies on attributed community search, we summarize new research directions, in terms of three aspects: community model improvement [6, 90], new problems with complex queries [44, 48, 78], and new solutions of machine learning models [18, 24, 34, 38, 61, 72, 78]. In the following, we introduce a few representative works of querying algorithms and machine learning based attributed community search.

2.1 Algorithmic Approaches

We discuss two representative ACS studies based on the types of keyword/numeric attributes and semantic-based attributes, respectively.

2.1.1 Keyword and numeric attributed community search

Different from previous ACS studies [19, 28] that only consider keyword attributes, recent works [48, 56, 57] consider the combination of *keyword attributes* and *numeric attributes* in attributed community search tasks.

Liu et al. [48] introduce a new combined attribute score to measure an attribute similarity of two vertices by employing a distance metric. Specifically, it applies Euclidean distance to compute the numeric location distance, while applying Jaccard distance to eliminate the similarity of textural keywords. Moreover, it employs a unified function to combine multiple kinds of distance functions, which provides better scalability to support multiple types of attributes, including both textual attributes (keywords) and numeric attributes (numbers). Based on this unified attribute score, Liu et al. propose the problem of <u>v</u>ertex-centric <u>a</u>ttributed <u>c</u>ommunity search problem (VAC-Problem) as follows.

VAC-Problem [48]. Given an attributed graph $G = (V, E, A)$, a set of querying vertices $Q \subseteq V$, and a parameter k, the problem of vertex-centric attributed community search aims to find the community $H \subseteq G$ satisfying the following properties: (1) query participation, i.e., $Q \subseteq V(H)$; (2) structure cohesiveness, such that H is a connected k-truss; (3) attribute cohesiveness, requiring that H has the smallest attribute score; (4) the maximality, requiring that there does not exist another subgraph $H' \supset H$ satisfying the above three properties.

Here, the k-truss [28] is a dense subgraph pattern H, requiring that every edge (u, v) in $E(H)$ is contained in at least $k - 2$ triangles in H. The unified attribute score is denoted as $Ascore(u, v) = \alpha \cdot \frac{Sdist(u,v)}{Sdist_{max}} + (1 - \alpha) \cdot \frac{Tdist(u,v)}{Tdist_{max}}$, where $Sdist(\cdot)$ and $Tdist(\cdot)$ denote the spatial distance and textual distance, respectively. Note that $Sdist_{max}$ and $Tdist_{max}$ are the maximal limits used for normalization.

The VAC-Problem is proven as NP-hard, reduced from the maximum clique problem. To solve the VAC-problem, a DFS-manner exact algorithm and a BFS-based exact algorithm are proposed. Due to the NP-hardness, several heuristic rules are proposed for vertex addition and deletion separately to improve efficiency performance. These heuristic strategies can be summarized into three aspects: pruning strategy, early termination strategy, and ordered search strategy. The pruning strategy pre-removes some useless edges and vertices based on the property of the k-truss model to reduce the search space. The early termination strategy stops the algorithm early by precomputing the low bounds of subgraphs' attribute scores. The ordered search strategy precomputes the vertex attribute score to query vertices and

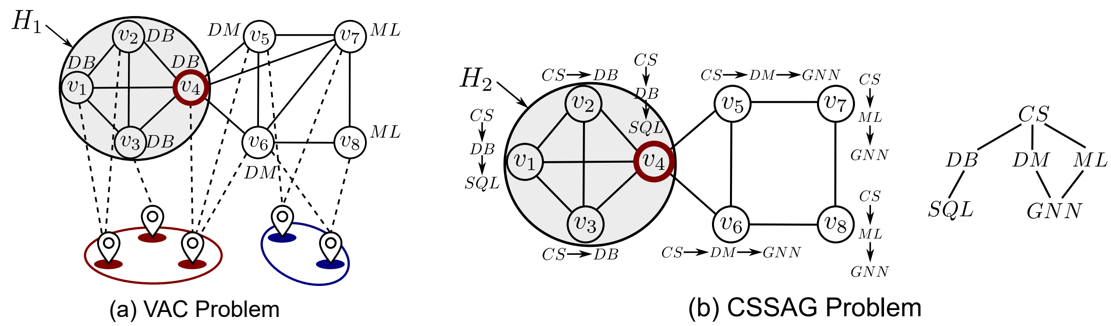

Figure 1: Typical examples of two attributed community search problems.

arranges the order when adding/deleting vertices from the graph. To further improve the efficiency and scalability of proposed algorithms, an approximate algorithm is proposed by achieving a 2-approximation guarantee to exact algorithms.

2.1.2 Semantic-based attributed community search

Besides the simple types of keywords and numerical values, attributes can have semantic relationships. Lin et al. [44] studied a problem of community search in semantic-based attributed graphs (CSSAG-Problem), which makes use of semantic information among attributes and gives an interpretability of community answers. In semantic-based attributed graphs, two keyword attributes may have a relationship in a directed acyclic semantic graph. This kind of graph is also meaningful in real life, e.g., identifying the synonyms and distinguishing polysynonyms. In the semantic attribute graph, attributes of vertex v is not a set of independent attributes, but a small directed acyclic semantic graph, denoted as $SG(v) = (V_{SG(v)}, E_{SG(v)})$, where $V_{SG(v)}$ is the semantic keyword set of the vertex v, $E_{SG(v)}$ is the semantic relations between theses keywords. In the semantic-based attributed graph H, it proposes the maximal common subgraph $MSG(H)$ to denote attribute cohesiveness, by satisfying the following properties: (1) $\forall v \in V(H), MSG(H) \subseteq SG(v)$; (2) there exists no other $MSG'(H)$ such that $MSG(H) \subseteq MSG'(H)$; (3) there exists at least one edge in $MSG(H)$. To find a subgraph H with the maximal common sub-semantic graph, the CSSAG-Problem is formulated as follows.

CSSAG-Problem [44]. Given a semantic-based attribute graph $G(V, E, A)$, a positive integer k, and a query vertex $q \in V$, the CSSAG-Problem is to find a maximal set of communities H satisfying the following properties: (1) H is a connected subgraph and $q \in V(H)$; (2) the degree of v in $V(H)$ is no less than k; (3) there exists no other $MSG'(H)$ such that $MSG(H) \subseteq MSG'(H)$, i.e., the $MSG(H)$ is maximal.

As shown in Figure 1(a), given an attributed graph with both keyword and location attributes, a query vertex v_4, and parameter $k = 4$, the VAC problem requires the community follows three constraints: H_1 is a 4-truss; vertices in H_1 has similar keywords; the vertices in H_1 is close to each other. Thus, the VAC problem aims to find the community H_1 that has similar keywords "DB" and is in the same region. Figure 1(b) provides an example of a semantic attributed graph, where each vertex corresponds to a small semantic subgraph. Given the query v_4 and $k = 3$, the CSSAG problem finds the community H_2, where each vertex in H_2 has a degree no less than 3, and the common sub-semantic graph $MSG(H) = \{CS, DB\}$ is maximal.

In summary, besides the above two ACS works, there are also other ACS studies, including anchored vertex set exploration in attributed graphs [6], k-truss-based cohesive attributed community search with maximal common attributes [90], and a unified hierarchy index structure for attributed community search [77].

2.2 Learning Approaches

Different from algorithmic approaches that bulid up the indexes in advance and then support the online community querying, there are a dozen of recent studies [4, 11, 18, 24, 34, 38, 61, 71, 72, 78] that develop machine learning models to learn the properties among trained communities and queries for tackling community search problems. In the following, we introduce two kinds of learning approaches based on *graph convolutional neural networks* [24, 34] and *transformer* [71, 76], respectively.

GNN-based Approach. Gao et al. [24] propose the ICS-GNN framework by applying GNN-based approaches to solve interactive community search problems for the first time. It first encodes the attributes to vertex features, then trains the GNN model by labeling a set of positive vertices and negative vertices separately for each query process. Following their work, several studies focus on applying the popular learning techniques to solve community search problems in attributed graphs. Jiang et al. [34] introduce the AQD-GNN model to solve community search problems on keyword-attributed graphs. Different from [24] that supports vertex-only queries, AQD-GNN can handle attribute-vertex combined queries in keyword-attributed graphs in three steps. First, it constructs a vertex-attribute bipartite graph to model the relationship between vertices and keyword attributes. Second, it applies the GNN model to bipartite graphs in order to learn both vertex feature similarity and topological similarity. In the training process, besides the labeling of positive/negative vertices, it further selects a set of query-community pairs as the training data to enhance the learning model. Finally, it constructs a fusion graph by capturing vertices with high attribute and topology similarity to the query vertices and applying BFS search to ensure the connectivity of answer community.

Transformer-based Approach. Most of the GNN-based community search approaches require labeling vertices during the training process, i.e., supervised learning and semi-supervised learning. Wang et al. [71] first propose an unsupervised transformer-based approach called transformer-based community search framework that uses zero label (TransZero), which does not required labeled vertices. TransZero contains two phases: pre-training phase and online search phase. In the pre-training phase, it proposes a community search graph transformer model to learn both vertex-level and subgraph-level representations. After that, it designs two kinds of self-supervised loss functions, including personalization loss and link loss, to avoid labeling vertices, and then conducts loss computation. In the online search phase, it first computes the community score for each vertex based on the pre-trained transformer model, and then designs a modularity-based function called community identification with expected score gain solver (IESG) to identify the community, where the IESG problem is proved to be NP-hard. Another transformer-based approach called CSFormer contains three phases [76]. First, it proposes a l-hop neighborhood community vector based on the n-th order h-index community model, which can be used to extract cohesiveness features. Based on the cohesiveness features extractor, it then proposes an attention-based transformer model to predict the vertex coreness. Finally, it conducts the online k-core community search based on predicted vertex coreness.

Besides the above studies, Li et al. [38] firstly introduce to use of contrastive learning approaches to train the community using semi-supervised learning. Wang et al. [72] extend the attributed community search works based on GNN models to billion scale graphs. Fang et al. [18] propose an inductive learning approach to help improve the scalability of learning models, which provides a fairer way to train the datasets.

3 Community Search on Complex Graphs

The investigation of community search on complex graphs has gained significant attention in recent years [5, 15, 22, 81]. Unlike simple graphs with categorized attributes or features [19, 28], complex graphs

Table 1: A summary of existing community search problems in HINs and multilayer graphs.

Community Model	Graph Type	Input Queries	Dense Subgraph Pattern	Community Homogeneity
CSH [22]	Heterogeneous Information Networks	Vertices; Meta-path	$(k, \mathcal{P})$-core	Single Vertex Type
CSSH [32]		Vertices	$(k, \mathcal{P})$-core	Single Vertex Type
ICSH [89]		Meta-path	$(k, \mathcal{P})$-core	Multiple Vertex Types
SACH [51]		Vertices; Meta-path	$(k, \mathcal{P})$-core	Single Vertex Type
SSH [49]		Vertices; Meta-path; Triplet Constraints	$(k, \mathcal{P})$-core	Multiple Vertex Types
MRCS [31]		Vertices; Triplet Constraints	k-core	Multiple Vertex Types
CASH [40]		Vertices	Learning-based Component	Single Vertex Type
CS-DAHIN [62]		Vertices; Meta-path	Learning-based Component	Single Vertex Type
LCD [55]	Multilayer Graphs	Vertices	Random Walk-based Component	Multiple Vertex Types
FTCS [5]		Vertices	k-truss	Multiple Vertex Types
CTCS [50]		Vertices	k-truss	Multiple Vertex Types
MCS [67]		Vertices	(k, d)-core	Multiple Vertex Types
SynCS [54]		Vertices	k-SynCore	Multiple Vertex Types
CSML [4]		Vertices	Learning-based Component	Multiple Vertex Types

encompass multiple types of attributes and exhibit diverse structures, e.g., heterogeneous information networks (HINs) [21, 22, 31, 32, 40, 49, 51, 62, 68, 89], multilayer graphs [4, 5, 50, 54, 55, 67], and dynamic graphs [53, 62, 69, 70, 81]. Table 1 summarizes the existing community search problems in heterogeneous information networks and multilayer graphs for a comparison, in terms of the query pattern, dense subgraph pattern, community type, and problem hardness. In the following section, we primarily focus on community search in two kinds of complex graphs: *heterogeneous information networks* and *multilayer graphs*.

3.1 Community Search over Heterogeneous Information Networks

Heterogeneous information networks (HINs) can model complex entities and relations, where nodes and edges have multiple types in the graph. For instance, authors write academic papers that are published in various venues. Thus, in this academic network, nodes can be categorized into distinct types such as authors, papers, and venues, while the edges represent different relationships including authorship and publication. HINs are composed of multiple types of nodes and edges, defined as $G(V, E, \mathcal{A}, \mathcal{R})$, where V is the set of vertices, E is the set of edges, $\mathcal{A} = \{A_1, A_2, ..., A_n\}$ is the set of node types, and $\mathcal{R} = \{R_1, R_2, ..., R_m\}$ is the set of edge types. The type mappings $\psi : V \rightarrow \mathcal{A}$ and $\phi : E \rightarrow \mathcal{R}$, such that $\psi(v) = A_1$, $\phi(e) = R_1$. Community search over heterogeneous information networks (HIN-CS) attracted significant attention in recent years [22, 31, 32, 40, 49, 51, 62, 89]. Various methodologies have been proposed to identify communities in HINs. In the following, we categorize HIN-CS studies in terms of two directions: meta-path-driven HIN-CS [22] and constraints-based HIN-CS [31].

3.1.1 Meta-path-driven HIN-CS

Fang et al. [22] propose a meta-path-based $(k, \mathcal{P})$-core model and study the community search problem in heterogeneous graphs (CSH-Problem) based on $(k, \mathcal{P})$-core. Given a heterogeneous graph G, a symmetric meta-path $\mathcal{P}$, and an integer k, a $(k, \mathcal{P})$-core is defined as the largest subgraph $H_{k,\mathcal{P}}$ of G such that every node v in $H_{k,\mathcal{P}}$ are $\mathcal{P}$-connected with at least k $\mathcal{P}$-neighbors of the same type, i.e. $\forall v \in V_{k,\mathcal{P}}$, $\alpha(v, H_{k,\mathcal{P}}) \geq k$. The $\mathcal{P}$-neighbors of v represent nodes connected to v via an instance of $\mathcal{P}$ (e.g., "$Author \rightarrow Paper \rightarrow Venue \rightarrow Paper \rightarrow Author$" in an academic HIN) in H. The $\mathcal{P}$-neighbors have the same type as node v. The definition of $\mathcal{P}$-degree is the number of $\mathcal{P}$-neighbors of v in H, denoted as $\alpha(v, H)$.

The problem of $(k, \mathcal{P})$-core-based community search [22]. Given a heterogeneous graph G, a query vertex q, a symmetric meta-path $\mathcal{P}$, and an integer k, the $(k, \mathcal{P})$-core-based community search problem is to output a community H, such that H contains q; H is a $(k, \mathcal{P})$-core. This problem extends the minimum degree measure k-core to a meta-path-based $(k, \mathcal{P})$-core. The target community H has two main characteristics. (1) For each node v of a community H, there are at least k other nodes, which can be connected to v via instances of a particular meta-path $\mathcal{P}$, within H. (2) H contains a group of nodes with the same type of query q.

Basic $(k, \mathcal{P})$-core algorithm. A straightforward solution of $(k, \mathcal{P})$-core-based community search problem is to transform the heterogeneous graph to a homogeneous graph via meta-path $\mathcal{P}$, and perform core decomposition to find the largest k-core subgraph. There are three main steps of basic algorithm. First, it collects all vertices with target node type to set S. Second, it enumerates all meta-path instances in $\mathcal{P}$ starting with v for each vertex $v \in S$, and adds edges between v and each of its $\mathcal{P}$-neighbors. Finally, it finds a connected k-core containing q. However, there are two weaknesses of this straightforward solution: (1) leads to some vertices weakly engaged; (2) causes high degrees and high clustering coefficients.

Edge/Vertex-disjoint $(k, \mathcal{P})$-core [22]. To overcome the weakness, two definitions of edge-disjoint path and vertex-disjoint path are proposed to formulate useful subgraphs of edge/vertex-disjoint $(k, \mathcal{P})$-cores. Edge-disjoint paths set $\psi[v]$ is a set of edge-disjoint paths if for any two path instances p_1, $p_2 \in \psi[v]$, their i-th $(1 \leq v \leq l)$ edges are different and $(l+1)$-th nodes are different. Similarly, the vertex-disjoint paths set $\psi[v]$ is a set of vertex-disjoint paths if for any two path instances p_1, $p_2 \in \psi[v]$, their i-th $(2 \leq v \leq l+1)$ nodes are different. Thus, given a heterogeneous graph G, a symmetric meta-path $\mathcal{P}$, and an integer k, an edge/vertex-disjoint $(k, \mathcal{P})$-core is the largest subgraph $E_{k,\mathcal{P}}$ / $V_{k,\mathcal{P}}$ of G such that every node v in $E_{k,\mathcal{P}}$ /$V_{k,\mathcal{P}}$ has at least k $\mathcal{P}$-neighbors with the same type and are $\mathcal{P}$-connected, i.e. $\forall v \in E_{k,\mathcal{P}}$ $(V_{k,\mathcal{P}})$, $\beta(v, E_{k,\mathcal{P}}) \geq k$ / $\gamma(v, V_{k,\mathcal{P}}) \geq k$.

A lazy peeling-based algorithm and its improvement. Based on the edge-disjoint $(k, \mathcal{P})$-core, a greedy algorithm is proposed to improve the efficiency of the basic algorithm. The lazy peeling process contains three parts: (1) it first collects a set S of all vertices with target type; (2) then it uses a greedy strategy to compute the edge-disjoint $(k, \mathcal{P})$-core, $\beta(v, E_{k,\mathcal{P}})$, with an approximation ratio of $\frac{1}{l}$; (3) it performs core decomposition by removing nodes with $\beta(v, E_{k,\mathcal{P}}) < k$ one by one. To further improve efficiency, a batch peeling-based algorithm is proposed. In summary, the $(k, \mathcal{P})$-core-based community search problem aims to find a single type of community close to the query node q by using semantic information offered by HINs.

3.1.2 Constraint-based HIN-CS

In the following, we introduce representative HIN-CS works, driven by query constraints instead of meta-paths.

Minimum relational community search [31]. Jian et al. [31] propose a problem called Minimum Relational Community Search problem (MRCS-Problem) by setting relational constraints. Given a heterogeneous graph G, a concerned label set L_S, a constraint set S, and a query node q, the MRCS-Problem is to find a community H in G, such that H contains q; H is a relational community r-com; and $|H|$ is minimized.

This problem allows users to design the paired relational constraints personally. Formally, the relational constraint is defined as a triplet $s =< l_1, l_2, k >$, where $l_1, l_2 \in L_G$ and $k \geq 1$. Note that each node with label l_1 must connect with at least k neighbors with label l_2, where l_1 and l_2 can be the same or different. Users can also define a constraint that focuses on not only node types but also edge types, such that, $s =< l_1, l_2, l_3, k >$, where node with label l_1 must connected with at least k neighbors with label l_2 and linked with edge labeled as l_3. The constraint set S is a set of relational constraints $S = \{s_1, s_2, ...s_t\}$. The set of concerned labels, L_S, is defined as $L_S = \{l| < l, l', k >\in S\}$.

MRCS exact algorithm and greedy algorithm. The key point to solve the MRCS-Problem is how to compute the qualified connected components in graph G. A message-passing algorithm [31] is developed to iteratively remove disqualified nodes. On the basis of this message passing algorithm, four extension algorithms are proposed, including an exact algorithm, a greedy algorithm, a local search approach, and a round index-based algorithm. The basic exact solution contains three main steps. It first computes all connected qualified node components using the message passing algorithm. Then, it finds the largest connected subgraph containing the query node. After that, it removes a group of nodes recursively until there is no node that can be removed. For different removing groups, it outputs the result with the minimum number of nodes. However, this method can be costly since it tries all possible groups of nodes. By improving the efficiency, a greedy algorithm has been proposed. It changes from trying all possible ways to get a globally minimized answer by removing the largest removable group in each iteration.

In summary, the MRCS-Problem is to find a query-dependent small-sized community that satisfies user-given relational constraints. The constraints can restrict both inter-layer and intra-layer relationships. However, the answer quality is highly related to the given constraints. However, it may be hard for users to design a set of suitable constraints.

3.1.3 Discussions

Both meta-path-based community search [22] and constraint-based community search [31] focus on heterogeneous information networks and adopt k-core dense subgraph models, but the community answers can be quite different. The meta-path-based $(k, \mathcal{P})$-core requires every vertex v in H has at least k numbers of neighbors u where u and v is connected via a meta-path $\mathcal{P}$. Users are required to provide a meta-path $\mathcal{P}$, and the community is required to be a $(k, \mathcal{P})$-core containing multiple of meta-paths. On the other hand, constraint-based relational community search [31] asks users to provide multiple pairwise constraints between pairs of vertices. For example, they can set up different k-core conditions for different pairs of vertices, i.e., set 3-core between "paper" and "venue", and set 2-core between "author" and "paper". It is more flexible to organize the relationship between different pairs of entities, but it needs more input parameters.

In addition to the above two HIN-CS works [22][31], there are other studies of community search over heterogeneous information networks. As shown in Table 1, several studies [32, 49, 51, 89] also employ meta-path-based approaches and $(k, \mathcal{P})$-core model, which facilitates the identification of communities that contain specified meta-paths critical to the analysis. Jiang et al. [32] propose the concept of non-nested meta-path core to identify communities in star-schema HINs and do not require a meta-path as input. [49] also supports user-defined constraints to help make the search more personalized. The works [40, 62] utilize learning-based approaches to develop a unified representation of HINs, enabling the discovery of flexible communities.

3.2 Community Search over Multilayer Graphs

Heterogeneous information networks mainly focus on the relationship between different types of entities, for example," $author \xrightarrow{write} paper \xrightarrow{published} venue$", which ignores the relationships between the same entities. Different from HINs, multilayer graph models the vertices with the same type in the same layer, and constructs multilayer graphs by entity types. Given a multilayer graph $MG(V_M, E_M, \mathcal{L})$, where $V_M = \bigcup_{i=1}^{l} V_i$ represents vertices in all l layers, $E_M = \bigcup_{i=1}^{l}\{E_i \cup \bigcup_{i<j\leq l} E_{ij}\}$ represents all inter-layer edges and cross-layer edges, and a function of layer identification $\mathcal{L}(.) \in \mathcal{Z}^+$. Thus, the academic graph

can be divided into a three layer graph containing the author layer, the paper layer, and the venue layer. Edges in the author layer represent the coauthorship between authors, edges in paper layer represent citations, and edges between author and paper represent authorships. In the literature, community search over multilayer graphs (MLCS) has attracted increasing attention [4, 5, 23, 46, 50, 54, 55, 67, 75], due to its significant impact in various applications. In this section, we mainly discuss two representative works of random walk-based community search [55] and (k, d)-core-based cross-layer community search [67] in multilayer graphs.

3.2.1 Random-walk based MLCS

Random-walk-based multilayer community search [55]. Luo et al. [55] propose a problem called local community detection in multiple networks, which aims to find a community in each layer of the heterogeneous graph. Given a heterogeneous graph $G = \{G^1, G^2, ...G^n\}$, a set of query nodes Q, the community search problem in heterogeneous graphs is to detect all relevant communities in each layer of G.

RWM-based method. A model of random walk in multiple networks (RWM) [55] is proposed in the following procedures: (1) send a random walker to each layer of the graph; (2) detect query-relevant subnetwork relations by calculate the cosine similarity between two layer; (3) the walkers reinforce each other by dynamically modifying their transition matrices; (4) sort nodes according to the converged score vector; (5) for each compute the conductance of the subgraph introduced by the top l ranked nodes (L is the number of non-zero elements in $x_i^{(T)}$.) ; (6) the node set with the smallest conductance can be considered as the detected community. However, a straightforward solution of performing RWM suffers from the limitation of imbalanced visiting probability vectors. To tackle it, an early stopping strategy splits the RWM model into two parts, before the error bound T, it updates both the transition matrices and the visiting vectors; after T, it keeps the transition matrices static and only updates the visiting probability vectors. Another is a partially updating strategy which splits the visiting probability vector into two parts and keeps the unrelated part static.

In summary, this work of community search over multilayer graphs [55] proposes a RWM model motivated by the random walk with restart strategy, trying to find out communities with a small conductance at each layer.

3.2.2 Cross-layer based MLCS

Sun et al. [67] introduce a (k, d)-core model to identify communities characterized by dense inter-layer and intra-layer connections, as well as robust connectivity among all pairs of sub-communities across different layers. It presents three key contributions: (1) the introduction of a fully-connected and a path-connected community model; (2) the design of a compact (k, d)-core index; and (3) the development of several efficient algorithms for community querying.

Fully-connected multilayer community [67]. Before introducing the fully connected community, let's first introduce the cross-layer (k, d)-core, which is defined across two layers. Let $H(H_i, H_j, E_{ij}^H) \subseteq MG$ denote a two-layer subgraph in the multilayer graph MG, and assume that k and d are two user-specified parameters. Given a multilayer subgraph $H \subseteq MG$, we say that H is a full-layer connected multilayer community *if and only if* for every pair of layers $i, j \in \mathcal{L}(H)$, there exists a strong cross-layer connectivity between G_i and G_j, such that $\forall i, j \in \mathcal{L}(H)$, $H_i \xleftrightarrow{H} H_j$.

Cross-layer community search [67]. Given a multilayer graph $MG(\ V_M, E_M, \mathcal{L})$, a set of query vertices $Q \subseteq V_M$, two positive integer parameters k, d, the problem of cross-layer community search in multilayer graphs (MCS-Problem) is to find a connected community $H \subseteq MG$ satisfying the following four constraints: query-dependent personalization such that $Q \subseteq V(H)$; Core-dense internal layers such that $\forall i \in \mathcal{L}(H)$, H_i is a connected k-core; *fully-connected cross-layers:* $\forall i, j \in \mathcal{L}(H)$, two layers H_i and H_j are connected via a (k, d)-core in H; *cross-layer maximization:* $|\mathcal{L}(H)|$ is maximized. To address this problem, a fast search algorithm is developed using a binary search strategy of potential community's layer number. The (k, d)-core index-based algorithms have demonstrated efficient performance on a large-scale multilayer graph dataset, FriendFeed (with 510,338 nodes, 20,204,534 edges, and 3 layers), using a short response time of around ten seconds. In contrast, RWM [55] and FirmTruss have failed to generate community results on FriendFeed.

Discussion of multilayer community search. Besides the above two studies [55][67], there are several other multilayer graph analytics studies on community search in multilayer graphs. Several studies [23, 46, 75] work on core decomposition of multilayer graphs, which help keep the records of dense multilayer subgraphs for efficient community search. Another line of research investigates k-truss-based community models and querying approaches in multilayer graphs [5, 50]. The k-core-based community search in multilayer graphs [54, 67] explores a new formulation of cross-layer communities and fast index construction. Furthermore, Behrouz et al. [4] leverage graph neural network (GNN) techniques to discover communities in multilayer graphs.

4 Interactive Community Search

Communities in graphs exhibit a wide range of different topological structures and attributes, which makes it challenging to identify exact communities relying solely on *a single and static community model*. To fit with different structures of communities and enhance the accuracy of query answers, existing community search models usually need a set of parameters to adjust the density, community size, diameter, target attributes, or even meta-paths. However, it is difficult for users to adjust parameters to navigate the correct search process, especially when users are not familiar with datasets and queries. To tackle these challenges, a few studies [24][66] are devoted to the problem of interactive community search (ICS), which allows users to refine community search answers by simply adding/deleting nodes into/from the community answer within a given small rounds of interactions. In this section, we first introduce the graph neural network-based interactive community search (ICS-GNN) [24]. We then present a flexible framework of interactive community search that allows for fitting with the interaction schema for several existing community search models [66].

4.1 Interactive Community Search via Graph Neural Network

Gao et al. [24] study the problem of interactive community search using GNN models as follows.

The k-sized maximum-GNN-score based community search [24]. The problem of k-sized Maximum-GNN-scores community search aims to find a k-sized connected community with the maximum GNN scores. The authors first abstract the GNN model as $P = GNN(A, F, W)$, where A is the adjacency matrix, F is the feature matrix, and W is the matrix of learnable parameters. Here, P is a mapping of probability, such that $P[i]$ represents the probability that the i-th vertex belongs to the target community. Given an enriched graph with GNN scores $G = (V, E, F, P)$, query vertex q and community size k, where P is the community membership possibility, the kMG problem is to find the k-sized community H with the maximum GNN scores, satisfies three constraints: (1) the query vertex $q \in V(H)$ and H is connected; (2) $|V(H)| = k$; and (3) the sum of GNN scores $\sum_{u \in V(H)} P[u]$ is maximum. This problem is shown to be NP-hard by reducing the Knapsack problem. The solution of GNN-based approach first builds a suitable-sized candidate subgraph; Then, it trains the GNN model on the candidate subgraph guided by the query vertex and labeled positive/negative vertices. After that, it infers the probability scores for all relevant vertices. Finally, it outputs the k-sized community with the maximum GNN scores.

Interactive community search via GNN. An interactive community search framework [24] is proposed to allow users to interactively provide feedback during the community search process so that the answer can be refined iteratively. The interactive process is conducted as follows: in each iteration, users can label a new set of positive vertices and negative vertices, which are included in the training set. The candidate subgraph for searching can also be updated based on user-provided labeled vertices. After the retraining and inferring, the GNN score for each vertex in the candidate subgraph is updated and produced as a refined k-sized community with the maximum GNN scores. If users are still unsatisfied with the answer, they can further label new vertices and retrain the model. To enhance the efficiency of the proposed ICS-GNN framework, the authors also develop a ranking-based loss function to simplify the labeling tasks by allowing users to compare and rank the importance of two vertices, instead of labeling them positive or negative. They also propose a greedy method based on a global view to identify the community to tackle the scenario where query vertices are at the boundary of the community.

4.2 A Flexible Interactive Community Search Framework

Although ICS-GNN [24] tackles the interactive community search using GNN models, it is challenging to extend it to other dense subgraphs based community models. Moreover, this method may be inefficient due to the need to compute GNN scores at each round of interaction, which requires rerunning learning models multiple times. To tackle these challenges, Sun et al. [66] propose a flexible framework to generalize existing community models and handle query-oriented interactive community search, called GICS. The techniques of GICS [66] can support different community models, e.g., attributed k-core-based community search [19], k-core-based weighted community search [64], and also the GNN-based community search [24].

A unified notation system to generalize existing community models. Sun et al. [66] first summarize the representative community search models and introduce an integrated notation system to define community models, denoted as $\mathbb{M}(\mathcal{G}, \mathcal{M}, \mathcal{O}, \mathcal{P})$, where $\mathcal{G}$ represents graph data, $\mathcal{M}$ defines various community metrics on the density, the graph size, or attributes, $\mathcal{O}$ contains a set of required operations, and $\mathcal{P}$ is a set of input parameters. Based on the notation system $\mathbb{M}$, it formulates the problem of interactive community search as follows. Given a graph $G = (V, E, A)$, a community model $\mathbb{M}_x(G, \mathcal{M}, \mathcal{O}, \mathcal{P}_0)$, query vertices $Q_0 \in \mathcal{P}_0$, the first iteration community search result H_0, and the number of maximum interactive round $I_{max} \in \mathbb{Z}^+$, the GICS problem aims to refine community H^{i-1} to H^i in the i-th iteration where $1 \leq i \leq I_{max}$, with a sequence of actions to add/remove vertices $Q^{\pm} \subseteq V$.

The GICS framework. The GICS framework for interactive community search [66] consists of three main steps: personalized adding/deleting recommendations, an auto-tuning parameter mechanism, and a fast refinement strategy. After the user chooses the specified community model $\mathbb{M}_x$, the system first implements the existing community search algorithm to get the initial community search result H_0. To enhance the efficiency of the interaction process, GICS system suggests a limited number of high-quality adding/deleting candidate vertices for users. Users can choose to add vertices to the community or remove vertices from the community by clicking the recommended vertices. This recommendation procedure first computes an integrated community relevance score for candidate vertices and then outputs a group of vertices with high community relevance and diverse location and attribute distribution. This paper also proposes an auto-tuning parameter mechanism to automatically refine the required parameters based on user adding/deletion actions, e.g., density metrics k for k-core and the size parameter k for k-sized community, which can significantly affect the community search answer. Finally, the GICS system develops a fast refinement strategy to help refine the choice of candidate subgraphs during the community search process, supporting partial refinement of communities.

5 Future Directions and Opportunities

In this section, we highlight a few future directions and promising opportunities for community search.

Leveraging community search for retrieval-augmented generation systems. Retrieval Augmented Generation (RAG) is a promising direction to enhance the performance of large language models (LLMs) by extracting relevant text chunks to a given query. Naturally, community search aims to find relevant communities containing query nodes. Thus, community search has potential applications for RAG systems with graph-structured documents. It is well-known that knowledge graphs are becoming increasingly important for organizing and querying vast amounts of structured information. The semantic information and complex relationships within knowledge graphs can help identify relevant subgraphs or communities for these knowledge-graph-based RAG systems. GraphRAG [16, 27] introduces a Leiden-based hierarchical community indexing solution, which leaves room for further exploration using advanced community search techniques.

LLM-enhanced community search over text-attributed graphs. Attributes are used to describe node properties, such as user interests, keywords, and check-in locations. Existing community search studies focus on attributed graphs with simple formats of category and numerical keywords. Nowadays, attribute information becomes richer and richer. Nodes and edges are associated with rich textual information, reflecting their diverse properties and complex semantics, denoted by text-attributed graphs [35]. An example of healthcare network consists of patients, doctors, diseases, and treatments, incorporating medical records and histories to improve patient care. Thus, it is important to further investigate community search over this complex and challenging textual graph, leveraging the strong analytics ability of LLMs on long texts [35]. Moreover, LLM-based systems allow a user-friendly interaction between users and systems. It suggests developing an interactive community search system to support queries issued by natural languages without a formal formulation, which allows multiple rounds of exploration in large-scale graphs. Such systems can further provide NLP-based interpretation to community search results, which advances the results of community search with an explainable insight.

Developing a unified dense subgraph model to handle various community search problems. Most existing community search models are usually proposed based on a particular subgraph pattern, e.g., k-core, k-truss, and quasi-clique. Thus, various community search algorithms and indexes are designed to support a particular subgraph maintenance, which have different search strategies and updating rules over dynamic graphs. Therefore, it is essentially important to develop a unified structure to generalize all existing and frequently used subgraphs, e.g., k-core and k-truss. Although GICS framework [66] gives a unified notation system for many dense subgraph-based community models, it still needs to implement specific updating algorithms to maintain these dense subgraphs. Moreover, the corresponding subgraph decomposition and index maintenance algorithms also need a unified solution over dynamic graphs, where node/edge insertions/deletions can occur anywhere in graphs.

References

[1] W. Ai, C. Xie, T. Meng, J. Du, and K. Li. A d-truss-equivalence based index for community search over large directed graphs. *TKDE*, 2024.

[2] A. Al-Baghdadi and X. Lian. Topic-based community search over spatial-social networks. *PVLDB*, 13(12):2104–2117, 2020.

[3] A. Arenas, L. Danon, A. Diaz-Guilera, P. M. Gleiser, and R. Guimera. Community analysis in social networks. *EPJB*, 38:373–380, 2004.

[4] A. Behrouz and F. Hashemi. Cs-mlgcn: Multiplex graph convolutional networks for community search in multiplex networks. In *CIKM*, pages 3828–3832, 2022.

[5] A. Behrouz, F. Hashemi, and L. V. Lakshmanan. Firmtruss community search in multilayer networks. *PVLDB*, 16(3):505–518, 2022.

[6] T. Cai, J. Li, N. A. H. Haldar, A. Mian, J. Yearwood, and T. Sellis. Anchored vertex exploration for community engagement in social networks. In *ICDE*, pages 409–420. IEEE, 2020.

[7] W. Cai, F. Zhu, Z. Liu, and M. Wu. Heterocs: A heterogeneous community search system with semantic explanation. In *SIGIR*, pages 3155–3159, 2023.

[8] G. Chen, F. Guo, Y. Wang, Y. Liu, P. Yu, H. Shen, and X. Cheng. Fcs-hgnn: Flexible multi-type community search in heterogeneous information networks. In *CIKM*, pages 207–217, 2024.

[9] J. Chen and J. Gao. Vics-gnn: A visual interactive system for community search via graph neural network. In *ICDE*, pages 3150–3153. IEEE, 2022.

[10] J. Chen, Y. Xia, and J. Gao. Communityaf: An example-based community search method via autoregressive flow. *PVLDB*, 16(10):2565–2577, 2023.

[11] J. Chen, Y. Xia, J. Gao, Z. Li, and H. Chen. Communitydf: A guided denoising diffusion approach for community search. In *ICDE*, pages 460–473. IEEE, 2025.

[12] L. Chen, C. Liu, K. Liao, J. Li, and R. Zhou. Contextual community search over large social networks. In *ICDE*, pages 88–99. IEEE, 2019.

[13] P. Chen and S. Redner. Community structure of the physical review citation network. *JOI*, 4(3):278–290, 2010.

[14] Y. Dai, M. Qiao, and R.-H. Li. On density-based local community search. *PACMMOD*, 2(2):1–25, 2024.

[15] Z. Dong, X. Huang, G. Yuan, H. Zhu, and H. Xiong. Butterfly-core community search over labeled graphs. *PVLDB*, 14(11):2006–2018, 2021.

[16] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, D. Metropolitansky, R. O. Ness, and J. Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

[17] S. Fang, K. Zhao, G. Li, and J. X. Yu. Community search: a meta-learning approach. In *ICDE*, pages 2358–2371. IEEE, 2023.

[18] S. Fang, K. Zhao, Y. Rong, Z. Li, and J. X. Yu. Inductive attributed community search: to learn communities across graphs. *PVLDB*, 17(10):2576–2589, 2024.

[19] Y. Fang, C. Cheng, S. Luo, and J. Hu. Effective community search for large attributed graphs. *PVLDB*, 9(12), 2016.

[20] Y. Fang, X. Huang, L. Qin, Y. Zhang, W. Zhang, R. Cheng, and X. Lin. A survey of community search over big graphs. *VLDBJ*, 29:353–392, 2020.

[21] Y. Fang, K. Wang, X. Lin, and W. Zhang. Cohesive subgraph search over big heterogeneous information networks: Applications, challenges, and solutions. In *SIGMOD*, pages 2829–2838, 2021.

[22] Y. Fang, Y. Yang, W. Zhang, X. Lin, and X. Cao. Effective and efficient community search over large heterogeneous information networks. *PVLDB*, 13(6):854–867, 2020.

[23] E. Galimberti, F. Bonchi, F. Gullo, and T. Lanciano. Core decomposition in multilayer networks: Theory, algorithms, and applications. *TKDD*, 14(1):1–40, 2020.

[24] J. Gao, J. Chen, Z. Li, and J. Zhang. Ics-gnn: lightweight interactive community search via graph neural network. *PVLDB*, 14(6):1006–1018, 2021.

[25] F. Guo, Y. Yuan, G. Wang, X. Zhao, and H. Sun. Multi-attributed community search in road-social networks. In *ICDE*, pages 109–120. IEEE, 2021.

[26] F. Hashemi, A. Behrouz, and M. Rezaei Hajidehi. Cs-tgn: Community search via temporal graph neural networks. In *WWW*, pages 1196–1203, 2023.

[27] H. Huang, Y. Huang, J. Yang, Z. Pan, Y. Chen, K. Ma, H. Chen, and J. Cheng. Retrieval-augmented generation with hierarchical knowledge. *arXiv preprint arXiv:2503.10150*, 2025.

[28] X. Huang and L. V. Lakshmanan. Attribute-driven community search. *PVLDB*, 10(9), 2017.

[29] X. Huang, L. V. Lakshmanan, and J. Xu. Community search over big graphs: Models, algorithms, and opportunities. In *ICDE*, pages 1451–1454. IEEE, 2017.

[30] X. Huang, L. V. Lakshmanan, and J. Xu. *Community search over big graphs*. Morgan & Claypool Publishers,

2019.
[31] X. Jian, Y. Wang, and L. Chen. Effective and efficient relational community detection and search in large dynamic heterogeneous information networks. *PVLDB*, 13(10):1723–1736, 2020.
[32] Y. Jiang, Y. Fang, C. Ma, X. Cao, and C. Li. Effective community search over large star-schema heterogeneous information networks. *PVLDB*, 15(11):2307–2320, 2022.
[33] Y. Jiang, X. Huang, and H. Cheng. I/o efficient k-truss community search in massive graphs. *VLDBJ*, 30(5):713–738, 2021.
[34] Y. Jiang, Y. Rong, H. Cheng, X. Huang, K. Zhao, and J. Huang. Query driven-graph neural networks for community search: From non-attributed, attributed, to interactive attributed. *PVLDB*, 2022.
[35] B. Jin, G. Liu, C. Han, M. Jiang, H. Ji, and J. Han. Large language models on graphs: A comprehensive survey. *TKDE*, 2024.
[36] J. Kim, T. Guo, K. Feng, G. Cong, A. Khan, and F. M. Choudhury. Densely connected user community and location cluster search in location-based social networks. In *SIGMOD*, pages 2199–2209, 2020.
[37] J. Kim, S. Luo, G. Cong, and W. Yu. Dmcs: Density modularity based community search. In *SIGMOD*, pages 889–903, 2022.
[38] L. Li, S. Luo, Y. Zhao, C. Shan, Z. Wang, and L. Qin. Coclep: Contrastive learning-based semi-supervised community search. In *ICDE*, pages 2483–2495. IEEE, 2023.
[39] L. Li, Y. Zhao, S. Luo, G. Wang, and Z. Wang. Efficient community search in edge-attributed graphs. *TKDE*, 35(10):10790–10806, 2023.
[40] Y. Li, X. Chen, Y. Zhao, W. Shan, Z. Wang, G. Yang, and G. Wang. Self-training gnn-based community search in large attributed heterogeneous information networks. In *ICDE*, pages 2765–2778. IEEE, 2024.
[41] Z. Li, P. Zou, X. Chen, S. Hu, P. Zhang, Y. Zhang, B. He, Y. Li, and X. Tang. From community search to community understanding: A multimodal community query engine. In *CIKM*, pages 4749–4753, 2021.
[42] X. Liao, Q. Liu, X. Huang, and J. Xu. Truss-based community search over streaming directed graphs. *PVLDB*, 17(8):1816–1829, 2024.
[43] L. Lin, P. Yuan, R.-H. Li, C. Zhu, H. Qin, H. Jin, and T. Jia. Qtcs: Efficient query-centered temporal community search. *PVLDB*, 17(6):1187–1199, 2024.
[44] P. Lin, S. Yu, X. Zhou, P. Peng, K. Li, and X. Liao. Community search over large semantic-based attribute graphs. *WWW*, 25(2):927–948, 2022.
[45] B. Liu, F. Zhang, W. Zhang, X. Lin, and Y. Zhang. Efficient community search with size constraint. In *ICDE*, pages 97–108. IEEE, 2021.
[46] D. Liu and Z. Zou. Gcore: Exploring cross-layer cohesiveness in multi-layer graphs. *PVLDB*, 16(11):3201–3213, 2023.
[47] Q. Liu, M. Zhao, X. Huang, J. Xu, and Y. Gao. Truss-based community search over large directed graphs. In *SIGMOD*, pages 2183–2197, 2020.
[48] Q. Liu, Y. Zhu, M. Zhao, X. Huang, J. Xu, and Y. Gao. Vac: vertex-centric attributed community search. In *ICDE*, pages 937–948. IEEE, 2020.
[49] X. Liu, L. Chen, C. Liu, and R. Zhou. Searching society over large heterogeneous information networks. In *ICDE*, pages 1402–1414. IEEE, 2025.
[50] X. Liu and Z. Zou. Common-truss-based community search on multilayer graphs. In *ADMA*, pages 277–291. Springer, 2023.
[51] Y. Liu, F. Guo, B. Xu, P. Bao, H. Shen, and X. Cheng. Sach: Significant-attributed community search in heterogeneous information networks. In *ICDE*, pages 3283–3296. IEEE, 2024.
[52] Y. Lu, L. Zhao, and B. Wang. From virtual community members to c2c e-commerce buyers: Trust in virtual communities and its effect on consumers' purchase intention. *ECRA*, 9(4):346–360, 2010.
[53] Z. Lu, Y. Zhu, M. Zhong, and J. X. Yu. On time-optimal (k, p)-core community search in dynamic graphs. In *ICDE*, pages 1396–1407. IEEE, 2022.
[54] C. Luo, Q. Liu, Y. Gao, and J. Xu. Synergetic community search over large multilayer graphs. *PVLDB*, 3(v6):v9, 2024.
[55] D. Luo, Y. Bian, Y. Yan, X. Liu, J. Huan, and X. Zhang. Local community detection in multiple networks. In *SIGKDD*, pages 266–274, 2020.
[56] J. Luo, X. Cao, X. Xie, and Q. Qu. Best co-located community search in attributed networks. In *CIKM*,

pages 2453–2456, 2019.
[57] J. Luo, X. Cao, X. Xie, Q. Qu, Z. Xu, and C. S. Jensen. Efficient attribute-constrained co-located community search. In *ICDE*, pages 1201–1212. IEEE, 2020.
[58] W. Luo, X. Zhou, K. Li, Y. Gao, and K. Li. Efficient influential community search in large uncertain graphs. *TKDE*, 35(4):3779–3793, 2021.
[59] M. E. Newman. Scientific collaboration networks. i. network construction and fundamental results. *Physical review E*, 64(1):016131, 2001.
[60] D. Rowangould, A. Karner, and J. London. Identifying environmental justice communities for transportation analysis. *Transportation Research Part A: Policy and Practice*, 88:151–162, 2016.
[61] Q. Sima, J. Yu, X. Wang, W. Zhang, Y. Zhang, and X. Lin. Deep overlapping community search via subspace embedding. *SIGMOD*, 3(1):1–26, 2025.
[62] Y. Song, L. Zhou, P. Yang, J. Wang, and L. Wang. Cs-dahin: Community search over dynamic attribute heterogeneous network. *TKDE*, 2024.
[63] G. Su, J. H. Morris, B. Demchak, and G. D. Bader. Biological network exploration with cytoscape 3. *Current protocols in bioinformatics*, 47(1):8–13, 2014.
[64] L. Sun, X. Huang, R.-H. Li, B. Choi, and J. Xu. Index-based intimate-core community search in large weighted graphs. *TKDE*, 34(9):4313–4327, 2020.
[65] L. Sun, X. Huang, R.-H. Li, and J. Xu. Fast algorithms for intimate-core group search in weighted graphs. In *WISE*, pages 728–744. Springer, 2019.
[66] L. Sun, X. Huang, J. Wang, and J. Xu. A flexible framework for query-oriented interactive community search. *PVLDB*, 2025.
[67] L. Sun, X. Huang, Z. Wu, and J. Xu. Efficient cross-layer community search in large multilayer graphs. In *ICDE*, pages 2959–2971. IEEE, 2024.
[68] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu. Heterogeneous information networks: the past, the present, and the future. *PVLDB*, 15(12), 2022.
[69] Y. Tang, J. Li, N. A. H. Haldar, Z. Guan, J. Xu, and C. Liu. Reliable community search in dynamic networks. *PVLDB*, 15(11):2826–2838, 2022.
[70] A. Tian, A. Zhou, Y. Wang, and L. Chen. Maximal d-truss search in dynamic directed graphs. *PVLDB*, 16(9):2199–2211, 2023.
[71] J. Wang, K. Wang, X. Lin, W. Zhang, and Y. Zhang. Efficient unsupervised community search with pre-trained graph transformer. *PVLDB*, 17(9):2227–2240, 2024.
[72] J. Wang, K. Wang, X. Lin, W. Zhang, and Y. Zhang. Neural attributed community search at billion scale. *PACMMOD*, 1(4):1–25, 2024.
[73] K. Wang, S. Wang, X. Cao, and L. Qin. Efficient radius-bounded community search in geo-social networks. *TKDE*, 34(9):4186–4200, 2020.
[74] K. Wang, W. Zhang, X. Lin, Y. Zhang, L. Qin, and Y. Zhang. Efficient and effective community search on large-scale bipartite graphs. In *ICDE*, pages 85–96. IEEE, 2021.
[75] R.-A. Wang, D. Liu, and Z. Zou. Focuscore decomposition of multilayer graphs. In *ICDE*, pages 2792–2804. IEEE, 2024.
[76] Y. Wang, X. Gou, X. Xu, Y. Geng, X. Ke, T. Wu, Z. Yu, R. Chen, and X. Wu. Scalable community search over large-scale graphs based on graph transformer. In *SIGIR*, pages 1680–1690, 2024.
[77] Y. Wang, Z. Peng, X. Ke, X. Xu, T. Wu, and Y. Gao. Cohesiveness-aware hierarchical compressed index for community search on attributed graphs. *SIGMOD*, 3(1):1–27, 2025.
[78] Y. Wang, S. Ye, X. Xu, Y. Geng, Z. Zhao, X. Ke, and T. Wu. Scalable community search with accuracy guarantee on attributed graphs. In *ICDE*, pages 2737–2750. IEEE, 2024.
[79] X. Xie, S. Liu, J. Zhang, S. Han, W. Wang, and W. Yang. Efficient community search based on relaxed k-truss index. In *SIGIR*, pages 1691–1700, 2024.
[80] R. Xu and D. C. Wunsch. Clustering algorithms in biomedical research: a review. *RBME*, 3:120–154, 2010.
[81] T. Xu, Z. Lu, and Y. Zhu. Efficient triangle-connected truss community search in dynamic graphs. *PVLDB*, 16(3):519–531, 2022.
[82] R. Yang, X. Xiao, Z. Wei, S. S. Bhowmick, J. Zhao, and R.-H. Li. Efficient estimation of heat kernel pagerank for local clustering. In *SIGMOD*, pages 1339–1356, 2019.

[83] K. Yao and L. Chang. Efficient size-bounded community search over large networks. *PVLDB*, 14(8):1441–1453, 2021.

[84] J. Ye, Y. Zhu, and L. Chen. Top-r keyword-based community search in attributed graphs. In *ICDE*, pages 1652–1664. IEEE, 2023.

[85] F. Zhang, H. Guo, D. Ouyang, S. Yang, X. Lin, and Z. Tian. Size-constrained community search on large networks: An effective and efficient solution. *TKDE*, 36(1):356–371, 2023.

[86] Y. Zhang, L. Lin, P. Yuan, and H. Jin. Significant engagement community search on temporal networks. In *DASFAA*, pages 250–258. Springer, 2022.

[87] Y. Zhang, K. Wang, W. Zhang, X. Lin, and Y. Zhang. Pareto-optimal community search on large bipartite graphs. In *CIKM*, pages 2647–2656, 2021.

[88] Z. Zhang, X. Huang, J. Xu, B. Choi, and Z. Shang. Keyword-centric community search. In *ICDE*, pages 422–433. IEEE, 2019.

[89] Y. Zhou, Y. Fang, W. Luo, and Y. Ye. Influential community search over large heterogeneous information networks. *PVLDB*, 16(8):2047–2060, 2023.

[90] Y. Zhu, J. He, J. Ye, L. Qin, X. Huang, and J. X. Yu. When structure meets keywords: Cohesive attributed community search. In *CIKM*, pages 1913–1922, 2020.