Computer Science Review 58 (2025) 100784

Contents lists available at ScienceDirect

Computer Science Review

journal homepage: www.elsevier.com/locate/cosrev

Review article

Maximal cliques summarization: Principles, problem classification, and algorithmic approaches

Marco D'Elia [a,*], Irene Finocchi [b], Maurizio Patrignani [a]

[a] Roma Tre University, Via della Vasca Navale 79, Rome, 00146, Italy
[b] Luiss Guido Carli, Viale Romania 32, Rome, 00197, Italy

ARTICLE INFO

Keywords:
Maximal cliques
Clique enumeration
Clique summarization

ABSTRACT

Several algorithms are available for computing all the maximal cliques of real-world graphs, both in centralized and distributed settings. However, in many application contexts, the sheer number of maximal cliques and their significant overlap call for strategies to reduce their quantity, maintaining only the most "meaningful" ones. In this survey we introduce a novel taxonomic framework that classifies summarization problems along two key dimensions: summarization principles and problem classes. Our framework provides a unified perspective on seemingly unrelated problems, organizing systematically the highly scattered literature on this topic, revealing underlying connections that were not previously well understood, and identifying several open problems in this field.

Contents

* Corresponding author.
E-mail addresses: marco.delia@uniroma3.it (M. D'Elia), finocchi@luiss.it (I. Finocchi), maurizio.patrignani@uniroma3.it (M. Patrignani).
URLs: https://compunet.ing.uniroma3.it/ (M. D'Elia), https://www.linkedin.com/in/irenefinocchi/ (I. Finocchi), https://compunet.ing.uniroma3.it (M. Patrignani).

https://doi.org/10.1016/j.cosrev.2025.100784
Received 28 January 2025; Received in revised form 26 May 2025; Accepted 14 June 2025
Available online 18 July 2025

1. Introduction

In the vast landscape of computer science, few challenges stand as prominently at the crossroads of theory and application as the enumeration of the communities in large graphs. The strictest definition of community is that of a maximal clique. A *clique* is a set of vertices with all the possible edges among them and it is *maximal* if no vertex of the graph can be added to the clique without violating this property. A (maximal) clique is a *maximum* clique if there is no clique in the graph with more vertices. For example, Fig. 1 shows a graph and its maximal cliques. The two cliques of size 5 in Fig. 1(b) are also maximum cliques.

Finding cliques and maximal cliques in graphs has applications in several domains, including web network analysis [1,2], community detection in real-world networks [3], economics and computational finance [4,5], biochemistry and genetics [6–8], microbiology [9], and telecommunications [10–12]. Computing a maximum clique is known to be NP-hard [13], W[1]-hard with respect to the size k of the clique [14,15] and hard to approximate within a factor of $n^{1-\epsilon}$ [16,17]. We remark that in the hardness proof of [13] all maximal cliques are also maximum cliques. Therefore, also computing maximal cliques is computationally hard. Despite this, both centralized [18–23] and distributed [24–27] algorithms are available to efficiently enumerate all maximal cliques of real-world graphs by leveraging their sparsity.

However, when dealing with large real-world graphs, the huge amount of such maximal cliques makes it unfeasible to use them in several application contexts. For instance, it is not unusual that a graph with less than 10^6 vertices has more than 10^7 maximal cliques [28]. Moreover, maximal clique have a significant amount of redundancy, as most of them overlap. The same issues occur with some popular relaxations of cliques such as k-cliques [29–31], k-plexes [32–35], γ-quasi cliques [36], k-diamonds [37], l-clubs [38], and s-defective cliques [39] (see [40] for a survey).

In this survey, we consider the following question: Given a graph G, can we produce a meaningful summary that represents all the maximal cliques of G in a way that their abundance and redundancy is reduced with respect to certain properties? This is a broad question, and there is no universal method for selecting a subset of relevant maximal cliques that, in some sense, represent all the excluded ones.

We aim at systematically defining various summarization principles, introducing a novel taxonomy, to produce summaries that exhibit different properties and that capture distinct perspectives on the maximal cliques of the graph. The concept of summarizing maximal cliques first appears in [28], where the focus is on finding a subset of maximal cliques that have a significant overlap with those that are excluded. As we show in Section 3, this is just one of several possible summarization principles that can be applied when dealing with maximal cliques. To the best of our knowledge, this is the first time that seemingly different research contributions related to maximal cliques summarization have been considered together as different viewpoints on the same goal, providing a classification of all such diverse approaches to maximal clique summarization.

This survey is devoted both to classifying existing problems in the literature within a comprehensive taxonomy and to introducing new research lines that naturally arise when summarization principles are combined with traditional problem targets. Indeed, we also classify summarization problems into four families, based on whether the summary is produced by considering Boolean properties or quality measures. This survey is primarily focused on the algorithmic aspects of the problems discussed, with a detailed exploration of methods and techniques presented in the literature.

The rest of the paper is organized as follows. In Section 2, we provide the needed definitions and preliminaries; in Section 3, we introduce a taxonomy of maximal clique summarization principles; in Section 4, we introduce a taxonomy of summarization problems and we review both those problems that have been studied in the literature and new problems that may naturally arise; in Section 5, we discuss problems that involve variations in the input graph or in the expected output; in Section 6, we outline promising research directions using the classification introduced in Section 4; in Section 7, we review the algorithmic contributions found in the literature addressing the problems listed in Section 4; finally, in Section 8, we conclude with a discussion of open problems and potential directions for future research.

2. Background on maximal clique enumeration

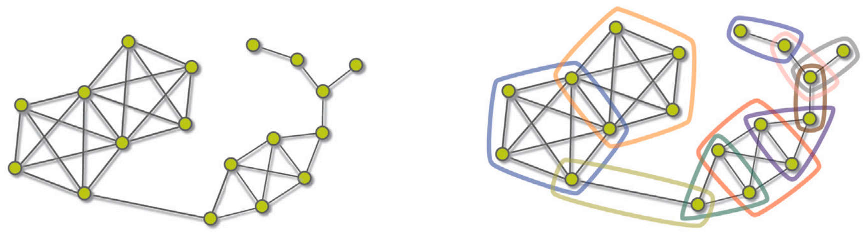
Let $G = (V, E)$ be a simple undirected graph. Denote by $n = |V|$ and by $m = |E|$ the number of vertices and edges of G, respectively. Denote by $\delta(v)$ the degree of a vertex v and by $N(v)$ the set of neighbors of a vertex v in G, i.e., $N(v) = \{u \mid (u, v) \in E\}$. Clearly, $|N(v)| = \delta(v)$. Let $X \subseteq V$ be any subset of vertices of G. We denote by $G[X]$ the induced subgraph of G whose vertex set is X and whose edge set consists of all the edges (u, v) such that both u and v belong to X. For simplicity, the vertex set and the edge set of $G[X]$ are denoted by $V[X]$ and $E[X]$, respectively.

A *clique* C is a complete subgraph of G. Given an integer k, the problem of finding a clique of G with at least k vertices is NP-hard [13], which implies that finding the maximum clique of G is also an intractable problem.

A clique C is called *maximal* if it is not included in any other clique C' of G. Let $\mathcal{M}(G)$ denote the set of all maximal cliques of G. It is known that there exist n-vertex graphs with $3^{n/3}$ maximal cliques [41], hence $|\mathcal{M}(G)|$ can be exponential in n. A (maximal) clique is a *maximum* clique if there is no clique in the graph with more vertices. Since in the construction of [41] all maximal cliques are also maximum cliques, the problem of enumerating all maximal cliques of a graph is intractable for two reasons: it is intractable to find the first maximal clique and is intractable because of the number of maximal cliques that have to be found. However, in several application contexts the graphs to be analyzed are sparse, and efficient algorithms are conceivable. Most known approaches to enumerate maximal cliques are based on the pioneering algorithm proposed by Bron and Kerbosch [18], which was later refined by Tomita, Tanaka, and Takahashi [22,42].

FPT enumerative approaches also exist, either in the degeneracy [20] or in the c-closedness [43,44] of the graph.[1]

[1] The degeneracy, a measure of sparseness, is the largest k such that the graph contains a k-core. A graph is c-closed if every pair of vertices that share at least c neighbors is adjacent.



(a)		(b)

Fig. 1. (a) An example graph. (b) The maximal cliques of the graph in (a).

Since in the following we will discuss variations of the approach in [22], we describe it in detail. The approach is based on a recursive backtracking procedure outlined in Algorithm 1. Three sets of vertices are maintained in the procedure: C, T, and D. Set C represents the current clique that is additively constructed; T is the set of candidate vertices that can potentially be added to C; and D is the set of vertices that should not be considered as they have already been included in a previous reported maximal clique from the current clique C or from a subset of it. Vertices in T and D are neighbors of all vertices in C. The main idea is to add a vertex from the set of candidate vertices in T to expand the current clique C, until it becomes maximal.

Algorithm 1 invokes Procedure ProcMCE(C, T, D) with $C = \emptyset$, $T = V$, and $D = \emptyset$ (Line 1). First, the procedure checks whether both T and D are empty. If so, C cannot be further expanded and is reported as a maximal clique (Line 4). Otherwise, in order to reduce the number of recursive calls, a pivot vertex v_p is chosen (Line 5) and the procedure is recursively called (Line 7) for each vertex $v \in T$ that is not a neighbor of v_p (this is because any maximal clique containing $C \cup \{w\}$, where w is a neighbor of v_p, can be reported either by recurring on $C \cup \{v_p\}$ or by recurring on $C \cup \{u\}$, where $u \in T$ is not a neighbor of v_p). The recursion is performed with $C = C \cup \{v\}$, $T = T \cap N(v)$, and $D = D \cap N(v)$ to ensure that every vertex in T and D is a neighbor of all vertices in C. When the recursive call returns, v is moved from T to D (Lines 8 and 9).

Algorithm 1: TomitaTanakaTakahashi

Input: Graph $G(V, E)$
Output: $\mathcal{M}(G)$
1 ProcMCE$(\emptyset, V, \emptyset)$
2 **Procedure** ProcMCE(C, T, D)
3 	**if** $T = \emptyset$ *and* $D = \emptyset$ **then**
4 		Report C as a maximal clique;
5 	choose a pivot vertex v_p in $T \cup D$ with highest $|N(v_p) \cap T|$;
6 	**foreach** $v \in T \setminus N(v_p)$ **do**
7 		ProcMCE$(C \cup \{v\}, T \cap N(v), D \cap N(v))$;
8 		$T \leftarrow T \setminus \{v\}$;
9 		$D \leftarrow D \cup \{v\}$;

The worst-case running time of Algorithm 1 is $\mathcal{O}(3^{n/3})$ [42], matching the number of maximal cliques in Moon–Moser graphs [41]. In [45] it is also proved that Algorithm 1 has $\Omega(3^{n/6})$ delay and that, even changing the pivoting strategy, a variant having polynomial delay cannot be designed unless P = NP. Notwithstanding these asymptotic complexity results, it has been experimentally observed that approaches based on Algorithm 1 and Bron–Kerbosch are fast in practice [20,42].

Consider as an example the graph depicted in Fig. 2(a). When Procedure ProcMCE is launched for the first time, $C = \emptyset$, $T = V$, and $D = \emptyset$ (all vertices of Fig. 2(b) are green). Since vertex v_3 satisfies the condition of Line 5, it is chosen as the pivot vertex v_p. Due to Line 6, the procedure will be recursively called for each $v \in T \setminus N(v_3)$, i.e., for $v = v_3$ only. Hence, the second recursive call of ProcMCE has $C = \{v_3\}$, $T = \{v_1, v_2, v_4\}$, and $D = \emptyset$ (see Fig. 2(c)). Assuming that vertex v_1 is now chosen as the pivot vertex v_p (notice that v_2 could be another possibility), the procedure will be recursively called for each $v \in T \setminus N(v_1)$, i.e., for $v = v_1$ and $v = v_4$. The third recursive call has $C = \{v_1, v_3\}$, $T = \{v_2\}$, and $D = \emptyset$ (see Fig. 2(d)). Vertex v_2 is then chosen as the pivot vertex and the procedure will be recursively called for $v = v_2$ only. Therefore, the fourth call of ProcMCE has $C = \{v_1, v_2, v_3\}$, $T = \emptyset$, and $D = \emptyset$ (see Fig. 2(e)). At this point, the conditions of Line 3 are met and the maximal clique $\{v_1, v_2, v_3\}$ is produced. Upon termination of the fourth call of ProcMCE, Lines 8 and 9 of the third recursive call move v_2 from T into D (see Fig. 2(f)). This terminates also the third call. Similarly, lines 8 and 9 of the second recursive call move v_1 from T to D (see Fig. 2(g)). Line 6 of the second call iterates with $v = v_4$ and recursively calls ProcMCE (Line 7) with $C = \{v_3, v_4\}$, $T = \emptyset$, and $D = \emptyset$ (see Fig. 2(h)). The conditions of Line 3 are met again and the maximal clique $\{v_3, v_4\}$ is produced. When the fifth call terminates, Lines 8 and 9 of the second call move v_4 from T to D. The second call also terminates and Lines 8 and 9 of the first call move v_3 from T to D. This terminates the first call, too. In the end, Algorithm 1 produced the maximal cliques $\{v_1, v_2, v_3\}$ and $\{v_3, v_4\}$.

Fig. 3 shows all the cliques of the graph depicted in Fig. 2(a) and the directed acyclic graph (DAG) of their inclusion relationships. The source of the DAG corresponds to the empty set while the maximal cliques are its two sinks. The nodes of the DAG with a solid border are those on which Algorithm 1 has been launched, exploring only a restricted portion of the solution space. The numbers close to these nodes correspond to the five recursive calls.

3. Summarization principles

Several principles can be adopted to pinpoint the relevant information provided by the maximal cliques of a graph, without necessarily relying on their exhaustive enumeration. The choice of the simplification principles depends on the application context at hand. In this section we discuss the main summarization approaches adopted so far in the literature, in the context of a new taxonomical framework (see Fig. 4).

Let $S \subseteq \mathcal{M}(G)$ be any subset of maximal cliques of an undirected graph G. We refer to S as a *summary* of $\mathcal{M}(G)$. In particular, if $|S| = s$, we call S an *s-summary*. Summarization principles can be classified as acquired or hereditary:

- A summarization principle P is *acquired* if for any summary $S \subseteq \mathcal{M}(G)$ that satisfies P, it holds that every S' such that $S \subseteq S' \subseteq \mathcal{M}(G)$ also satisfies P. For an acquired principle P it makes sense to search for a summary of reduced size, since the very $\mathcal{M}(G)$ would be a summary satisfying P.
- A summarization principle P is *hereditary* if for any summary $S \subseteq \mathcal{M}(G)$ that satisfies P, it holds that every $S'' \subseteq S$ also satisfies P. Observe that a hereditary principle P is always satisfied by the empty summary. Hence, such a principle is usually used in combination with other (non-hereditary) principles.

Acquired and hereditary principles can be further distinguished. Each of the remaining subsections focuses on one of these detailed principles, which in turn corresponds to one of the leaves of the taxonomy tree of Fig. 4.

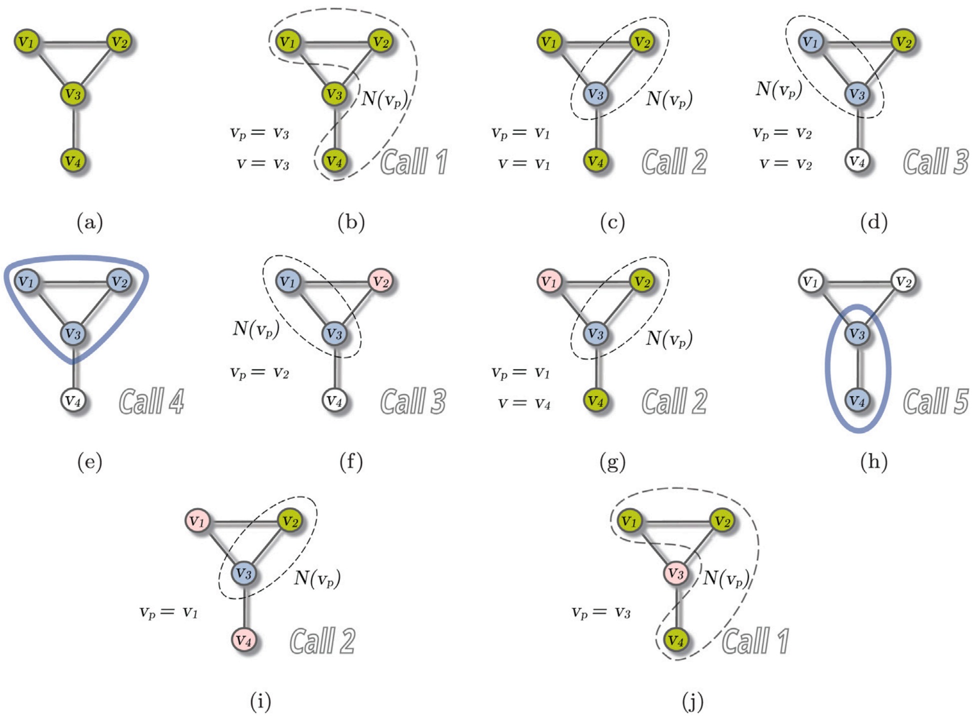


Fig. 2. A running example of Algorithm 1 on a small graph. Green, blue, and red vertices belong to T, C, and D, respectively.

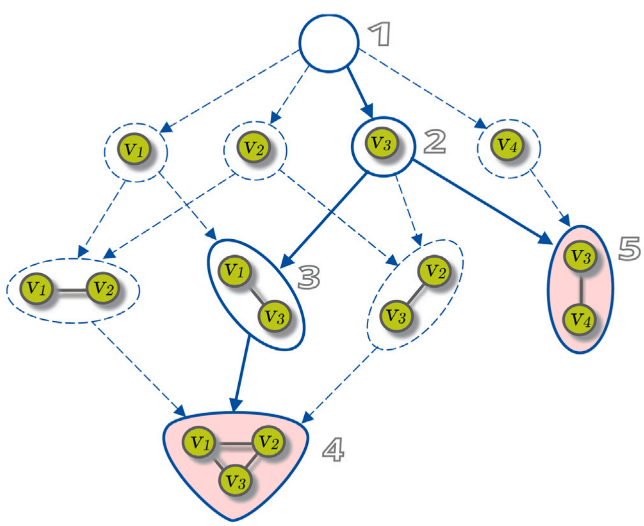


Fig. 3. The inclusion relationship of the cliques of the small graph depicted in Fig. 2. The maximal cliques are the sink of the DAG. The nodes of the DAG with solid border have been explored by Algorithm 1.

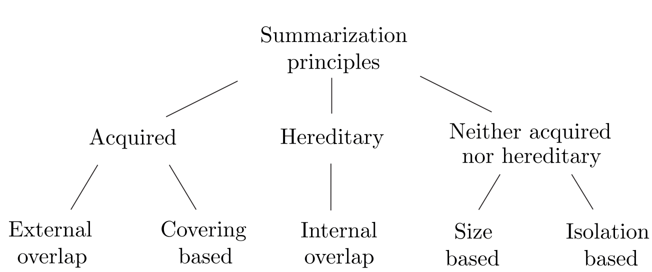


Fig. 4. A taxonomy of maximal cliques summarization principles.

3.1. Size-based summarization

Size-based criteria may be helpful in applications where large maximal cliques are more relevant than smaller ones. For instance, one could list all maximal cliques with a given size k (referred to as maximal k-cliques), or only those whose size is at least k. Another size-based strategy could be to find the largest (i.e., maximum) clique or to list all the maximum cliques. From the examples provided above it is apparent that this principle is neither acquired nor hereditary.

When a size-based criterion is adopted, the degree of relevance of a maximal clique depends exclusively on its own properties (i.e., the number of its vertices), and not on its relationship with other cliques. This is different from other summarization principles discussed in the following.

Practical applications of size-based summarization approaches include filtering the graph by removing noisy or peripheral regions and presenting only its denser portion to the user. In other cases, the goal is to produce a succinct core that highlights the most connected nodes, implicitly providing a clear description of their connectivity patterns.

Size-based approaches might be not always satisfactory, especially in applications where the summary should convey information about the entire graph. For instance, when listing maximal cliques of size at least k, some areas of the graph may not be properly summarized. For example, Fig. 5(b) shows an example of a size-based summary S when $k = 5$: in this summary all the rightmost portion of the graph is neglected. The summary S in Fig. 5(a) is instead obtained when $k = 4$. Notice that the summary for the larger value of k (Fig. 5(b)) is a subset of the summary for the smaller value (Fig. 5(a)). Also, it is often the case that large cliques are strongly overlapped, yielding a representation that may be redundant for the denser portions of the graph and incomplete for the sparser portions of it.

3.2. Overlap-based summarization

The enumeration of maximal cliques often produces cliques that are strongly overlapped. In order to grasp the extent of the overlap phenomenon, consider the following example. Let C be a maximal

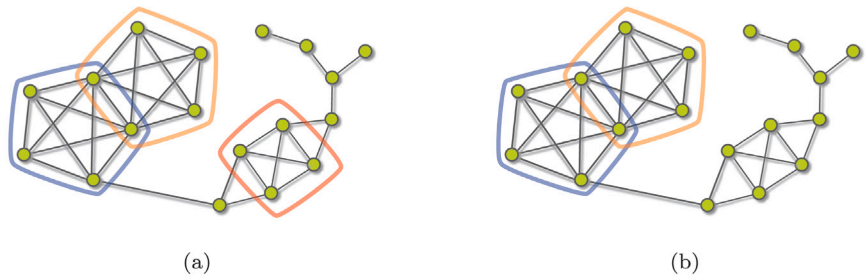


Fig. 5. (a) The maximal cliques in a size-based summary with $k = 4$. (b) The maximal cliques in a size-based summary with $k = 5$..

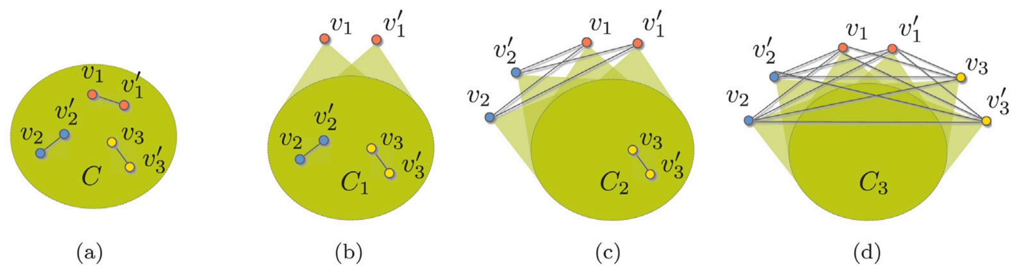


Fig. 6. (a) A clique C with three independent edges highlighted. (b), (c), (d) The effect of the removal of the edges.

clique of size n as shown in Fig. 6(a). Let $(v_1, v_1'), (v_2, v_2'), (v_3, v_3'), \ldots$ be a sequence of mutually non adjacent edges of C and let $C_1 = C \setminus \{v_1, v_1'\}$, $C_2 = C \setminus \{v_1, v_1', v_2, v_2'\}$, $C_3 = C \setminus \{v_1, v_1', v_2, v_2', v_3, v_3'\}$, …. By removing edge (v_1, v_1') from C, we obtain two maximal cliques of size $n-1$, namely, $C_1 \cup \{v_1\}$ and $C_1 \cup \{v_1'\}$ (see Fig. 6(b)). By removing the second edge (v_2, v_2'), we obtain four maximal cliques of size $n-2$, namely, $C_2 \cup \{v_1, v_2\}$, $C_2 \cup \{v_1, v_2'\}$, $C_2 \cup \{v_1', v_2\}$, and $C_2 \cup \{v_1', v_2'\}$ (see Fig. 6(c)). When we remove the ith edge, we obtain 2^i maximal cliques of size $n - i$ that share clique C_i of size $n - 2i$ (see Fig. 6(d) for $i = 3$). In conclusion, the elision of edges from the original maximal clique produces an exponential number of maximal cliques of linearly smaller size. All of them overlap in a portion that is also of linear size.

Since it is common in different domains that some edges are missing, due to elision, noise or uncertainty in the available data, the presence of a high number of strongly overlapped maximal cliques is very common. Therefore, in several applications a single maximal clique may account for all the other maximal cliques that are strongly overlapped with it. In these cases, a possible strategy to summarize the maximal cliques is that of avoiding overlapped ones. This could lead to two different objectives:

External-overlap summarization requires that the maximal cliques in $\mathcal{M}(G)$ not contained in summary S have a strong overlap with at least one of those contained in S;

Internal-overlap summarization requires that the maximal cliques contained in S have a small overlap among themselves.

Both these objectives motivate the quest for a precise formal definition of when a maximal clique C is considered 'represented' or 'visible' by another maximal clique C'. The *local visibility* [28] of a maximal clique C with respect to another maximal clique C' is defined as $\mathcal{V}_{C'}(C) = \frac{|C \cap C'|}{|C|}$. Notice that local visibility is neither transitive nor symmetric. In the following sections we describe how local visibility can be used both for external-overlap and internal-overlap summarization.

3.2.1. External-overlap summarization

In this kind of summarization the aim is ensuring that the maximal cliques not contained in S are somehow represented by those in S. For example, one could exclude from S those maximal cliques that are strongly overlapped with some maximal clique of S or search for a summary S that maximizes a measure of the overlap with the maximal cliques that are not in S. Observe that external-overlap summarization is an acquired principle.

Following the notation used in [28], the *visibility* of a maximal clique $C \in \mathcal{M}(G)$ with respect to S, denoted as $\mathcal{V}_S(C)$, is defined as:

$$\mathcal{V}_S(C) = max_{C' \in S}\ \mathcal{V}_{C'}(C) = max_{C' \in S} \frac{|C \cap C'|}{|C|} \qquad (1)$$

A summary S of a graph G is *τ-visible*, with $0 \le \tau \le 1$, if the visibility of each maximal clique $C \in \mathcal{M}(G)$ with respect to S is at least τ, i.e., $\mathcal{V}_S(C) \ge \tau$. For example, Figs. 7(a) and 7(b) show two possible τ-visible summaries when $\tau = 0.4$. If $\tau = 0$, any $S \subseteq \mathcal{M}(G)$ is τ-visible. If $\tau = 1$, the summary S necessarily coincides with $\mathcal{M}(G)$. Notice that a τ-visible summary S may in general not be a covering of V, i.e., $\exists v \in V \mid v \notin C, \forall C \in S$. The concept of τ-visibility can be used to define either Boolean properties for a summary S (asking, for example, that S is τ-visible) or an external-overlap-based quality measure for S (considering, for example, the maximum τ for which S is τ-visible).

External-overlap summarization is especially helpful in those applications that would need to review the whole set of maximal cliques, but that, for efficiency reasons, restrict their analysis to a subset of them, each representative of a group of "similar" ones.

3.2.2. Internal-overlap summarization

This summarization principle aims at ensuring that the maximal cliques in S have a small overlap and is especially recommended in those applications that need to cluster strongly connected vertices into groups that are "unrelated" among themselves. The overlap among the cliques in S can be measured in terms of the local visibility between each (ordered) pair of them.

We say that the *local independence* $\mathcal{I}(C, C')$ of two cliques C and C' is the maximum local visibility of C with respect to C' and vice-versa. More formally, $\mathcal{I}(C, C') = max\{\mathcal{V}_C(C'), \mathcal{V}_{C'}(C)\} = max\{\frac{|C \cap C'|}{|C|}, \frac{|C \cap C'|}{|C'|}\} = \frac{|C \cap C'|}{min(|C|, |C'|)}$. A summary S is *σ-independent*, with $0 \le \sigma \le 1$, if for each pair C and C' of maximal cliques in S we have $\mathcal{I}(C, C') \le \sigma$. For example, Fig. 8(a) shows a σ-independent summary when $\sigma = 0$. If $\sigma = 1$, any $S \subseteq \mathcal{M}(G)$ is σ independent. If $\sigma = 0$, a σ-independent summary S is composed by non-mutual intersecting maximal cliques.

When $\sigma = 0$, a stronger concept of independency can be enforced. Namely, one could ask that two maximal cliques in S are not connected by many edges. The *external edge set* of a maximal clique $C \in \mathcal{M}(G)$,

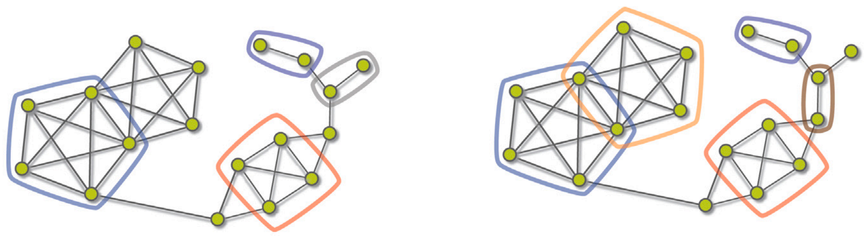


Fig. 7. (a) A τ-visible summary S with $\tau = 0.4$. Each excluded maximal clique has at least 40% of its vertices included into a maximal clique in S. (b) Another τ-visible summary S with $\tau = 0.4$.

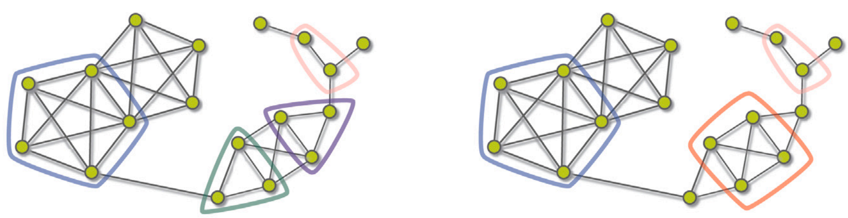


Fig. 8. Examples of internal overlap summarization. (a) A σ-independent summary S with $\sigma = 0$. (b) A ρ-edge-independent summary S with $\rho = 0$.

denoted as $\mathcal{E}(C)$, is defined as $\mathcal{E}(C) = \{(u, v) \mid u \in C \text{ XOR } v \in C\}$. Let C and C' be two maximal cliques of size c and c', respectively, both belonging to a 0-independent summary S. Observe that any vertex $v \in C$ is adjacent to at most $c' - 1$ vertices of C' (otherwise, C' would not be maximal). Therefore, the maximum number of edges connecting C with C' is $e_{\text{MAX}}(C, C') = min\{c'(c-1), c(c'-1)\} = c \cdot c' - max\{c, c'\}$. We define the *local edge independence* $\mathcal{E}(C, C')$ of two maximal cliques C and C' as

$$\mathcal{E}(C, C') = \frac{|\mathcal{E}(C) \cap \mathcal{E}(C')|}{e_{\text{MAX}}(C, C')} \tag{2}$$

A summary S is *ρ-edge-independent*, with $0 \le \rho \le 1$, if S is 0-independent and if $\mathcal{E}(C, C') \le \rho$ for every pair of maximal cliques C and C' in S. For example, Fig. 8(b) shows a ρ-edge-independent summary when $\rho = 0$. If $\rho = 0$, a ρ-edge-independent summary S consists of unconnected maximal cliques. Conversely, if $\rho = 1$, any $S \subseteq \mathcal{M}(G)$ is ρ-edge-independent.

Observe that internal-overlap summarization is a hereditary principle: a summary S remains σ-independent or ρ-edge-independent even when removing a maximal clique from it.

3.3. Covering-based summarization

Intuitively, when searching for a reduced size summary, maximizing the "coverage" of the graph vertices, i.e., the number of vertices that are contained into at least one maximal clique of the summary, may be a way to reduce the redundancy of the summarization. This principle is sometimes called in the literature "diversification" [46], since it is assumed that the maximal cliques in the summary have a small overlap. More formally, given a set of maximal cliques $S \subseteq \mathcal{M}(G)$ of a graph G, the *coverage* of S, denoted as $c(S)$, is the set of vertices of G covered by the maximal cliques in S, i.e., $c(S) = \bigcup_{C \in S} C$. We say that S is a *c-covering* of G, with $0 \le c \le 1$, if $\frac{|c(S)|}{|V(G)|} \ge c$. For example, Fig. 9(a) shows a c-covering summary when $c = 1$.

Additionally, the *edge-coverage* of S, denoted as $ec(S)$, is the set of edges of G covered by the maximal cliques in S, i.e., $ec(S) = \bigcup_{C \in S} E(C)$. We say that S is a *c-edge-covering* of G, with $0 \le c \le 1$, if $\frac{|ec(S)|}{|E(G)|} \ge c$. The fact that covering-based summarization is an acquired principle justifies the search for a reduced size summary.

Covering-based summarization can be used to produce summaries that represent a large portion of the vertices or edges, ensuring that a wide and meaningful area of the graph is captured. Vertex-covering produces a summary where a large portion of the vertices is represented, while edges could be significantly filtered out. Edge-covering coincides with distributing a large portion of the edges into cliques, actually producing a decomposition of the graph.

3.4. Isolation-based summarization

For some application contexts, maximal cliques that are poorly connected to the rest of the graph are more relevant than maximal cliques that are somehow intertwined with it. These applications search for bunches of vertices that are "independent", that is, strongly connected among themselves and weakly connected to the rest of the graph. This kind of summaries are used, for example, to coarsen real-world networks, such as the web graph, at various levels of abstraction while preserving their scale-free properties [47–49]. The concept of clique independency is captured by the notion of ℓ-isolation [50–52]. Given a graph $G = (V, E)$ a set of vertices $I \subseteq V$ of size k is *avg-ℓ-isolated* (or *ℓ-isolated* for simplicity) if the number $d_{out}(I)$ of edges that have one endpoint in I and one endpoint in $V \setminus I$ is less than $\ell \cdot k$. A summary S containing all the ℓ-isolated maximal cliques when $\ell = 1$ is shown in Fig. 9(b). A set of vertices $I \subseteq V$ is *max-ℓ-isolated* if each vertex $v \in I$ has less than ℓ neighbors outside I. A set of vertices $I \subseteq V$ is *min-ℓ-isolated* if at least one vertex $v \in I$ has less than ℓ neighbors outside I. Clearly, max-ℓ-isolatedness implies avg-ℓ-isolatedness, which in turn implies min-ℓ-isolatedness, but not vice versa. Number ℓ is sometimes called the *isolation factor*.

This summarization principle asks, for each concept of isolation above and for some value of ℓ, to focus on maximal cliques that are also ℓ-isolated (*ℓ-isolated maximal cliques*). In [51,52] a second target is also considered: computing ℓ-isolated cliques that are also "maximal", in the sense that they are not contained in other ℓ-isolated cliques (*maximal ℓ-isolated cliques*). In the remainder we only focus on ℓ-isolated maximal cliques, since the second target is out of the scope of this survey: an ℓ-isolated clique that is not contained in any other ℓ-isolated clique may not be maximal with respect to the graph. It can be shown, however, that maximal ℓ-isolated cliques are a superset of ℓ-isolated maximal cliques [50]. If follows that one could obtain all ℓ-isolated maximal cliques by filtering from maximal ℓ-isolated cliques those that are not maximal with respect to the graph.

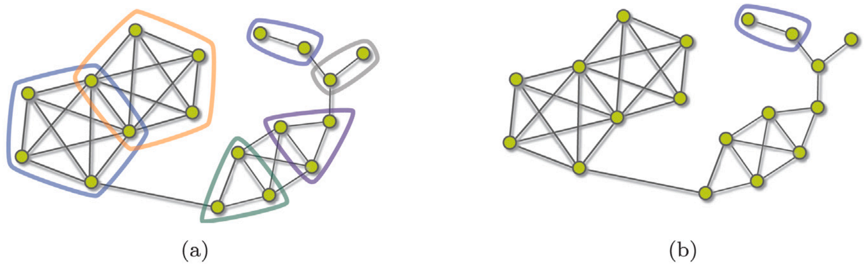


Fig. 9. (a) A c-covering summary S with $c = 1$: each vertex of the graph is contained into at least one maximal clique in S. (b) A summary S with all the ℓ-isolated maximal cliques when $\ell = 1$: the chosen maximal cliques have less than k outgoing edges.

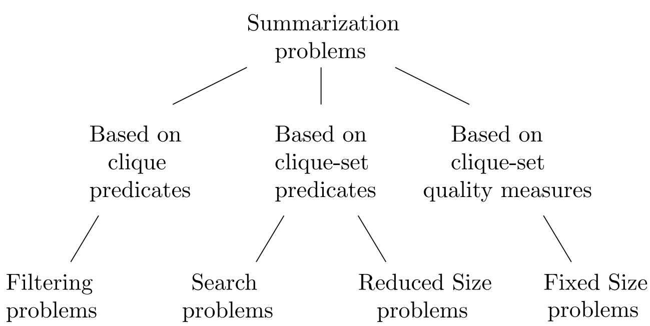


Fig. 10. A taxonomy of maximal cliques summarization problems.

Observe that this summarization principle is neither acquired nor hereditary.

4. Problem classification

For each summarization principle several different problems can be considered. We introduce a classification of such problems into four classes based on the type of property that the desired summary should satisfy (see Fig. 10). When this property is expressed with respect to each single maximal clique, we have a Filtering problem (see Section 4.1). When, instead, the property is expressed with respect to a whole summary S, then we have either a Search problem (see Section 4.2) or an optimization problem where the summary should have a small size (see Section 4.3). Finally, when the summary size is fixed, we might be interested in finding the best summary with respect to a given quality measure (see Section 4.4).

An overview of the problems discussed or proposed in this section is provided in Tables 1–3, where the problems are grouped according to the four classes introduced in this section. For each problem, the summarization principles are listed, along with whether the problem is open or the references to the literature where it has been studied.

4.1. Filtering

In this kind of problems, the summarization principles are used to construct a Boolean property that may or may not be satisfied by any maximal clique in $\mathcal{M}(G)$. The target is that of efficiently producing the summary $S \subseteq \mathcal{M}(G)$ composed by those maximal cliques that satisfy the given property. Observe that S, which could be empty, is unique.

Examples of Filtering problems include the following.

Problem 1 (*Maximal Cliques by Size*)**.** Given a graph G and a positive integer s, find all maximal cliques of size at least s.

Problem 1 is addressed in [53] and in Section 7.1.

Problem 2 (*ℓ-Isolated Maximal Cliques*)**.** Given a graph G and a positive integer ℓ, find all maximal cliques that are ℓ-isolated with respect to an isolation concept (max-, avg-, min-ℓ-isolation).

Table 1
An overview of Filtering and Search problems discussed or proposed in Section 4.

Filtering	Problem 1	MAXIMAL CLIQUES BY SIZE *Summarization principles: size* [53,54]
	Problem 2	ℓ-ISOLATED MAXIMAL CLIQUES *Summarization principles: isolation* [50–52]
	Problem 3	ℓ-ISOLATED MAXIMAL CLIQUES BY SIZE *Summarization principles: size, isolation* Open
Search	Problem 4	FIND ANY τ-VISIBLE σ-INDEPENDENT MAXIMAL CLIQUE SUMMARY *Summarization principles: external overlap, internal overlap* Open
	Problem 5	FIND ANY c-COVERING σ-INDEPENDENT MAXIMAL CLIQUE SUMMARY *Summarization principles: internal overlap, covering* Open in general. $c = 1$, $\sigma = 0$, non-maximal: [55–57]
	Problem 6	FIND ANY c-EDGE-COVERING σ-INDEPENDENT MAXIMAL CLIQUE SUMMARY *Summarization principles: internal overlap, covering* Open in general. $c = 1$, $\sigma = 0$: [58,59] (non-maximal: [60–62])
	Problem 7	c-COVERING ρ-EDGE-INDEPENDENT MAXIMAL CLIQUE SUMMARY *Summarization principles: internal overlap, covering* Open in general. $\rho = 0$, non-maximal: [63–67]
	Problem 8	c-EDGE-COVERING ρ-EDGE-INDEPENDENT MAXIMAL CLIQUE SUMMARY *Summarization principles: internal overlap, covering* Open in general. $\rho = 0$, non-maximal: [68–73]

Problem 2 is addressed in [50–52], which are surveyed in Section 7.4.

The Boolean property used for filtering can be obtained by combining different summarization principles. One example is the following.

Problem 3 (*ℓ-Isolated Maximal Cliques by Size*)**.** Given a graph G and two positive integers ℓ and s, find all maximal cliques that are ℓ-isolated with respect to an isolation concept and that have size at least s.

To the best of our knowledge, Problem 3 has never been addressed.

4.2. Search

In this kind of problems, the summarization principles are used to construct a Boolean property that may or may not be satisfied by a subset S of $\mathcal{M}(G)$. The aim is that of producing any summary $S \subseteq \mathcal{M}(G)$ that satisfies the property, provided it exists. Usually, the Boolean property combines at least two summarization principles, where the first is hereditary and the second is acquired. Examples include the following.

Table 2
An overview of Reduced Size problems discussed or proposed in Section 4.

Reduced Size	Problem 9	τ-VISIBLE MAXIMAL CLIQUE SMALL SUMMARY *Summarization principles: external overlap* [28,75,76]
	Problem 10	c-COVERING MAXIMAL CLIQUE SMALL SUMMARY *Summarization principles: covering* Open
	Problem 11	c-EDGE-COVERING MAXIMAL CLIQUE SMALL SUMMARY *Summarization principles: covering* Open in general. $c = 1$, non-maximal: [60,61,77]
	Problem 12	τ-VISIBLE c-COVERING MAXIMAL CLIQUE SMALL SUMMARY *Summarization principles: external overlap, covering* Open
	Problem 13	τ-VISIBLE σ-INDEPENDENT MAXIMAL CLIQUE SMALL SUMMARY *Summarization principles: external overlap, internal overlap* Open
	Problem 14	c-COVERING σ-INDEPENDENT MAXIMAL CLIQUE SMALL SUMMARY *Summarization principles: internal overlap, covering* Open in general. $c = 1$, $\sigma = 0$, non-maximal: [55–57]

Problem 4 (*Find Any τ-Visible σ-Independent Maximal Clique Summary*). Given a graph G and two values τ and σ in $[0, 1]$, find a τ-visible summary $S \subseteq \mathcal{M}(G)$ that is also σ-independent.

Problem 5 (*Find Any c-Covering σ-Independent Maximal Clique Summary*). Given a graph G and two values c and σ in $[0, 1]$, find a c-covering summary $S \subseteq \mathcal{M}(G)$ that is also σ-independent.

When $c = 1$ and $\sigma = 0$, Problem 5 corresponds to partitioning the vertices of G into non-overlapped maximal cliques. When cliques are non-necessarily maximal it has been addressed in [55–57].

Problem 6 (*Find Any c-Edge-Covering σ-Independent Maximal Clique Summary*). Given a graph G and two values c and σ in $[0, 1]$, find a c-edge-covering summary $S \subseteq \mathcal{M}(G)$ that is also σ-independent.

When $c = 1$ and $\sigma = 0$, Problem 6 corresponds to partitioning the edges of G into non-overlapped maximal cliques and has been studied in [58,59]. When cliques are non-necessarily maximal it has been addressed in [60–62] and surveyed in [74].

Problem 7 (*c-Covering ρ-Edge-Independent Maximal Clique Summary*). Given a graph G and two values c and ρ in $[0, 1]$, find a c-covering summary $S \subseteq \mathcal{M}(G)$ that is also ρ-edge-independent.

When $\rho = 0$, Problem 7 has been studied in [63–67] in a variant where the goal is to maximize c for cliques that are non-necessarily maximal. This variant is known with the name CLUSTER VERTEX DELETION or, equivalently, INDEPENDENT UNION OF CLIQUES.

Problem 8 (*c-Edge-Covering ρ-Edge-Independent Maximal Clique Summary*). Given a graph G and two values c and ρ in $[0, 1]$, find a c-edge-covering summary $S \subseteq \mathcal{M}(G)$ that is also ρ-edge-independent.

When $\rho = 0$, Problem 8 has been studied in [68–73] in a variant where the goal is to maximize c for cliques that are non-necessarily maximal. This variant is known with the name CLUSTER DELETION.

4.3. Reduced size

In this kind of problems, the summarization principles are used to construct a Boolean property that may or may not be satisfied by any subset S of $\mathcal{M}(G)$. The target is that of efficiently producing a summary $S \subseteq \mathcal{M}(G)$ that satisfies the property and has small size. Usually, the Boolean property refers to an acquired summarization principle, i.e., if S satisfies the property, any superset $S' \supseteq S$, also satisfies it. For example τ-visibility for a specific τ is one such properties. Another example is c-covering for a specific c. We provide hereunder some examples of Reduced Size problems.

Problem 9 (*τ-Visible Maximal Clique Small Summary*). Given a graph G and a value τ, with $0 \le \tau \le 1$, find a τ-visible summary $S \subseteq \mathcal{M}(G)$ of small size.

This problem coincides with the enumeration of maximal cliques when $\tau = 1$. Problem 9 is addressed in [28,75,76], which are surveyed in Section 7.2.

Problem 10 (*c-Covering Maximal Clique Small Summary*). Given a graph G and a value c, with $0 \le c \le 1$, find a c-covering summary $S \subseteq \mathcal{M}(G)$ of small size.

Problem 11 (*c-Edge-Covering Maximal Clique Small Summary*). Given a graph G and a value c, with $0 \le c \le 1$, find a c-edge-covering summary $S \subseteq \mathcal{M}(G)$ of small size.

When $c = 1$, Problem 11 has been studied for (non-necessarily maximal) cliques in [60,61,77]. The survey [74] refers to this variant.

The considered Boolean property could be obtained by combining different summarization principles. One example is the following.

Problem 12 (*τ-Visible c-Covering Maximal Clique Small Summary*). Given a graph G and two values τ and c in $[0, 1]$, find a τ-visible summary $S \subseteq \mathcal{M}(G)$ of small size that is also c-covering.

Also observe that any problem formulated as "find any" can be translated into a Reduced Size problem, where the target is that of producing a small-size summary $S \subseteq \mathcal{M}(G)$ among those that satisfy the property.

For example compare the following with Problem 4.

Problem 13 (*τ-Visible σ-Independent Maximal Clique Small Summary*). Given a graph G and two values τ and σ in $[0, 1]$, find a τ-visible summary $S \subseteq \mathcal{M}(G)$ of small size that is also σ-independent.

Compare the following with Problem 5.

Problem 14 (*c-Covering σ-Independent Maximal Clique Small Summary*). Given a graph G and two values c and σ in $[0, 1]$, find a c-covering summary $S \subseteq \mathcal{M}(G)$ of small size that is also σ-independent.

When $c = 1$ and $\sigma = 0$, Problem 14 corresponds to partitioning the vertex-set of a graph in such a way that each block of the partition corresponds to a maximal clique. The variant of this problem when cliques are non-necessarily maximal has been studied in [55–57].

Regarding the objective of reducing the size of S, a common practice in the literature is that of finding summaries whose cardinality is smaller than the cardinality of the smallest summaries computed by previous algorithms on the same datasets. In absolute terms "small size" could be interpreted as "minimum size" yielding more challenging versions of the above problems. A relaxed version of searching for minimum size summaries could be that of searching for summaries that are minimal, that is, they do not satisfy the property if any maximal clique is removed.

Regarding the efficiency, the proposed algorithms are sometimes compared among themselves, with Algorithm 1 as a baseline.

4.4. Fixed size

In this kind of problems the size k of the maximal clique summary is fixed and the summarization principles are used to obtain a quality measure for any subset S of $\mathcal{M}(G)$ such that $|S| = k$. The target is to find a set $S \subseteq \mathcal{M}(G)$ of size k that maximizes the quality measure. Examples include the following.

Table 3
An overview of Fixed Size problems discussed or proposed in Section 4.

Fixed Size	Problem 15	TOP-k MAXIMAL CLIQUES BY SIZE *Summarization principles: size* Open
	Problem 16	SET OF k MAXIMAL CLIQUES MAXIMIZING COVERAGE *Summarization principles: covering* [46,78–80]
	Problem 17	SET OF k DISJOINT MAXIMAL CLIQUES MAXIMIZING COVERAGE *Summarization principles: internal overlap, covering* Open in general. $\sigma = 0$, non-maximal: [81,82]
	Problem 18	SET OF k EDGE DISJOINT MAXIMAL CLIQUES MAXIMIZING COVERAGE *Summarization principles: internal overlap, covering* Open
	Problem 19	SET OF k MAXIMAL CLIQUES MINIMIZING ISOLATION *Summarization principles: isolation* Open

Problem 15 (*Top-k Maximal Cliques by Size*)**.** Given a graph G and a positive integer k, find a summary S of size k such that $\sum_{C\in S}|C|$ is maximized.

Problem 16 (*Set of k Maximal Cliques Maximizing Coverage*)**.** Given a graph G and a positive integer k, find a summary S of size k such that the coverage size $|c(S)| = |\bigcup_{C\in S} C|$ is maximized.

For $k = 1$, Problems 15 and 16 coincide with the problem of searching for the maximum clique of G.

Problem 16, with the name TOP-k DIVERSIFIED MAXIMAL CLIQUES, has been studied in [46,78,79] which are surveyed in Section 7.3. Additionally, it has been further addressed in [80,83–85], considering a generalization in which vertices are weighted (see Problem 21).

Problem 17 (*Set of k Disjoint Maximal Cliques Maximizing Coverage*)**.** Given a graph G, a positive integer k, and a value $\sigma \in [0,1]$, find a σ-independent summary S of size k such that the coverage size $|c(S)| = |\bigcup_{C\in S} C|$ is maximized.

For $k = 1$, Problem 17 coincides with the problem of searching for the maximum clique of G. When $\sigma = 0$, Problem 17 has been studied in [81,82] for non-necessarily maximal cliques with the name DISJOINT UNION OF CLIQUES.

Problem 18 (*Set of k Edge Disjoint Maximal Cliques Maximizing Coverage*)**.** Given a graph G, a positive integer k, and a value $\rho \in [0,1]$, find a ρ-edge-independent summary S of size k such that the size of the coverage $|c(S)| = |\bigcup_{C\in S} C|$ of S is maximized.

For $k = 1$, Problem 18 coincides with the problem of searching for the maximum clique of G. Observe that any solution to Problems 17 and 18 is also a solution to Problem 16, while the converse does not necessarily hold.

Problem 19 (*Set of k Maximal Cliques Minimizing Isolation*)**.** Given a graph G and a positive integer k, find a summary S of size k composed by ℓ-isolated maximal cliques such that the isolation factor ℓ with respect to an isolation concept (max-, avg-, min-ℓ-isolation) is minimized.

5. Variants of summarization principles and problems

In addition to the classification of problems discussed in the previous section, several variants of summarization principles and problems can arise, depending for instance on the setting of the input graph or on the constraints that the desired summary must satisfy. This section explores some of these possible variants.

Table 4
An overview of the problems discussed or proposed in Section 5.

Filtering	Problem 20	MAXIMAL CLIQUES BY WEIGHT *Summarization principles: (weighted) size* `Variant: Weighted graphs` Open
	Problem 23	α-MAXIMAL CLIQUES BY SIZE ON UNCERTAIN GRAPHS *Summarization principles: size* `Variant: Uncertain graphs` [86–88]
Fixed Size	Problem 21	SET OF k MAXIMAL CLIQUES MAXIMIZING WEIGHTED COVERAGE *Summarization principles: covering* `Variant: Weighted graphs` [80,83–85]
Reduced Size	Problem 25	EXPECTED τ-VISIBLE MAXIMAL CLIQUE SMALL SUMMARY *Summarization principles: external overlap* `Variant: Probabilistic summary` [28,75]
Filtering + Fixed Size	Problem 22	TOP-k MAXIMAL CLIQUES ON UNCERTAIN GRAPHS *Summarization principles: size* `Variant: Uncertain graphs` [89]
	Problem 24	TOP-k α-MAXIMAL CLIQUES BY SIZE ON UNCERTAIN GRAPHS *Summarization principles: size* `Variant: Uncertain graphs` [90]

Section 5.1 addresses the scenario where the input graph has weights assigned to vertices and edges. This yields a natural extension to problems grounded in the covering-based summarization principle, where the weight of the coverage is considered instead of its cardinality. Section 5.2 focuses on the case where the input graph has probabilities assigned to both vertices and edges. This yields a natural extension to problems grounded in the size-based summarization principle, where the probability of existence of a maximal clique is also taken into account when computing the summary. Section 5.3 examines probabilistic relaxations of the problems, i.e., variants where the summary must satisfy the desired property in an expected way.

An overview of the problems discussed in this section is provided in Table 4, where they are grouped according to the four classes introduced in Section 4. For each problem, we report the summarization principles, the specific variant considered, and the references where it has been studied.

5.1. Weighted graphs

In a *vertex-weighted graph* $G = (V, E, w)$, each vertex of V is associated with a weight $w : V \to \mathbb{R}^+$. The meaningfulness of a clique in these application scenarios is given by the sum of the weights of its vertices rather than by their number. Therefore, a *maximal clique* is a clique whose weight is maximal. Fortunately, since the weight of each vertex is a positive number, a clique of maximal weight is also a maximal clique in terms of the number of its vertices and vice versa. However, the summarization principles listed in Section 3 are affected by this change in perspective. For example, size-based summarization asks to focus on maximal cliques of large weight rather than large number of vertices. Consider the following reformulation of Problem 1.

Problem 20 (*Maximal Cliques by Weight*)**.** Given a vertex-weighted graph G and a positive integer w, find all maximal cliques of weight at least w.

Given a summary $S \subseteq \mathcal{M}(G)$, the *weighted coverage* of S, denoted as $w(S)$, is defined as $w(S) = \sum_{v\in\bigcup_{C\in S} C} w(v)$ and is considered in the following problem.

Problem 21 (*Set of k Maximal Cliques Maximizing Weighted Coverage*)**.** Given a vertex-weighted graph G and a positive integer k, find a summary S of size k, such that the weighted coverage $w(S)$ of S is maximized.

Problem 21 is studied in [80,83–85] under the name TOP k-DIVERSIFIED WEIGHTED MAXIMAL CLIQUES. Notice that the case of unweighted graphs can be viewed as a special case when each vertex has weight 1. For example, Problem 21 generalizes Problem 16, and an algorithm for the former can be applied also to the latter. Instead, when the parameter $k = 1$, Problem 16 coincides with the maximum weight clique problem (see [91] and the references therein).

A similar extension can be considered by allowing edge-weighted graphs, although literature on this subject is available only for a few special cases [92–94].

5.2. Handling uncertainty

An *uncertain graph* [89] is a graph $G = (V, E, \varphi, \psi)$, where $\varphi : V \to [0, 1]$ is a function assigning existence probability values to the vertices in V, and $\psi : E \to [0, 1]$ is a function assigning existence probability values to the edges in E upon the condition that both the endpoints of each edge exist. Therefore, a traditional graph, also referred to as an *exact graph*, can be considered as a special case of an uncertain graph, with all vertices and edges having an existence probability of 1. Usually, it is assumed that both the existence probabilities of vertices and the conditional existence probabilities of edges are mutually independent.

An exact graph $G' = (V', E')$ is *implicated* by an uncertain graph $G = (V, E, \varphi, \psi)$, denoted by $G \Rightarrow G'$, if and only if $V' \subseteq V$ and $E' \subseteq E \cap (V' \times V')$. The *probability of G implicating G'* is

$$Pr(G \Rightarrow G') = \prod_{v \in V'} \varphi(v) \cdot \prod_{v \in V \setminus V'} (1 - \varphi(v)) \cdot \prod_{(u,v) \in E'} \psi((u, v) \mid u, v) \cdot \prod_{(u,v) \in (E \cap (V' \times V')) \setminus E'} (1 - \psi((u, v) \mid u, v)) \quad (3)$$

Let Ω denote the set of exact graphs implicated by an uncertain graph G. Eq. (3) defines a probability distribution over all graphs in Ω. The probability $\pi_{max}(C)$ of a vertex set $C \subseteq V$ being a maximal clique across all implicated graphs G' in Ω is

$$\pi_{max}(C) = \sum_{G' \in \Omega \text{ and } G' \text{ contains } C \text{ as a maximal clique}} Pr(G \Rightarrow G') \quad (4)$$

This probability is referred to as the *probability of the maximal clique* C, i.e., the probability that C is a maximal clique in any implication of G. Moreover, given a collection S of vertex sets, the *sum of clique probabilities* of S, is defined as $\Sigma(S) = \sum_{C \in S} \pi_{max}(C)$. Building on the aforementioned definitions, the following problem has been discussed in the literature.

Problem 22 (*Top-k Maximal Cliques on Uncertain Graphs*)**.** Given an uncertain graph G and two positive integers k and s, find a collection S of k vertex sets, such that each vertex set in S has size at least s and $\Sigma(S)$ is maximized.

When $k = |\mathcal{M}(G)|$, $s = 1$, and G is an exact graph, this problem coincides with maximal clique enumeration. Problem 22 has been addressed in [89].

A restricted version of this model is when the uncertain graph has existence probabilities defined only for its edges (the existence probabilities of the vertices are always 1). This is discussed in [86–88,90]. In this case Eq. (3) is simplified as follows

$$Pr(G \Rightarrow G') = \prod_{(u,v) \in E'} \psi((u, v)) \cdot \prod_{(u,v) \in E \setminus E'} (1 - \psi((u, v))) \quad (5)$$

Given an uncertain graph G where the existence probabilities of the vertices are equal to 1 and a set of vertices C, the *clique probability* of C, denoted by $\pi(C)$, is defined as the probability that in an exact graph implicated by G, C is a clique:

$$\pi(C) = \sum_{G' \in \Omega \text{ and } G' \text{contains } C \text{ as a clique}} Pr(G \Rightarrow G') \quad (6)$$

which coincides with:

$$\pi(C) = \prod_{u,v \in C} \psi((u, v)) \quad (7)$$

Given a value $\alpha \in [0, 1]$, a vertex set C is an *α-clique* if $\pi(C) \geq \alpha$. Moreover, C is an *α-maximal clique* if it is an α-clique and there does not exist a vertex set $C' \supset C$ such that C' is an α-clique.

We mention the following problems from the literature.

Problem 23 (*α-Maximal Cliques by Size on Uncertain Graphs*)**.** Given an uncertain graph G, a positive integer k and a value $\alpha \in [0, 1]$, find all α-maximal cliques of size at least k.

Problem 23 is addressed in [86–88] where it is referred to as (k, α)-MAXIMAL CLIQUES ON UNCERTAIN GRAPHS.

Problem 24 (*Top-k α-Maximal Cliques by Size on Uncertain Graphs*)**.** Given an uncertain graph G, a positive integer k and a value $\alpha \in [0, 1]$, find a set S of k α-maximal cliques such that $\sum_{C \in S} |C|$ is maximized.

Problem 24 is addressed in [90] where it is referred to as (k, α)-MAXIMAL CLIQUES ON UNCERTAIN GRAPHS.

Notice the differences between Problem 22 and Problem 24. In the former, vertex sets with size at least s are ranked according to their probability, whereas in the latter, vertex sets corresponding to α-maximal cliques are ranked according to their size.

5.3. Probabilistic relaxations

Given that computing an exact solution for most of the problems formulated above is challenging, we can relax the requirements by allowing the desired summary properties to be met by the algorithm's output with a certain probability. Till now, up to our knowledge, only the case of Problem 9, in which the external-overlap summarization principle is used, has been addressed in the literature.

Let $\mathcal{A}$ be a randomized algorithm that inserts a maximal clique $C \in \mathcal{M}(G)$ into its output summary S with probability $Pr[C \in S]$. The *expected visibility* of a maximal clique $C \in \mathcal{M}(G)$ with respect to S, denoted as $\mathbb{E}[\mathcal{V}_S(C)]$, is defined as:

$$\mathbb{E}[\mathcal{V}_S(C)] = 1 \cdot Pr[C \in S] + \mathcal{V}_S(C) \cdot Pr[C \notin S] \quad (8)$$

Observe that, the output S of $\mathcal{A}$ is not known in advance and, hence, $\mathcal{V}_S(C)$ can be exactly determined only *a posteriori*. However, its value could be estimated and lower-bounded to guarantee that $\mathbb{E}[\mathcal{V}_S(C)]$ is also lower-bounded. A summary S of a graph G is *expected τ-visible*, with $0 \leq \tau \leq 1$, if the expected visibility of each maximal clique $C \in \mathcal{M}(G)$ with respect to S is at least τ, i.e. $\mathbb{E}[\mathcal{V}_S(C)] \geq \tau$.

Compare the following with Problem 9.

Problem 25 (*Expected τ-Visible Maximal Clique Small Summary*)**.** Given a graph G and a value τ, with $0 \leq \tau \leq 1$, find an expected τ-visible summary $S \subseteq \mathcal{M}(G)$ of small size.

As for Problem 9, Problem 25 coincides with the enumeration of maximal cliques when $\tau = 1$. Problem 25 is addressed in [28,75], which are surveyed in Section 7.2.

5.4. Parametrized approaches

Parametrized complexity [14,15] offers a framework for addressing NP-hard problems by identifying a *parameter* k that captures some structural aspect of the problem. A problem is said to be *fixed-parameter tractable* (FPT) with respect to k if it can be solved in time $f(k) \cdot n^{\mathcal{O}(1)}$, where f is a computable function independent of the input size n. This allows for efficient algorithms when k is small, even for large instances. A common way to show that a problem is unlikely to be FPT with respect to a given parameter k is to give a parametrized reduction from a known W[1]-hard problem [14,15], thereby establishing W[1]-hardness. A *parametrized reduction* reduces an instance (X, k) of a problem P that is W[1]-hard with respect to the parameter k to instance (X', k') of a problem P' in $f(k) \cdot |X|^{\mathcal{O}(1)}$ time such that k' only depends on k and such that (X, k) is a positive instance of P if and only if (X', k') is a positive instance of P'.

As mentioned in Section 1, k-CLIQUE and, hence, Problem 1, is known to be W[1]-hard with respect to the size k of the clique [14,15]. To the best of our knowledge, among the summarization problems addressed in this survey, an FPT approach has been proposed only for Problem 2 [50–52] using the isolation factor ℓ as parameter and leading to FPT algorithms to enumerate all ℓ-isolated maximal cliques (see Section 7.4).

While any structural graph parameter could, in principle, be a candidate for FPT analysis, it is particularly interesting to consider parameters that are explicitly present in the problem definition itself. Unfortunately, most parameters that appear in the problems listed in Section 4, such as τ, σ, ρ, and c, are real numbers in $[0, 1]$ and, hence, cannot be directly used as parameters for an FPT reduction or algorithmic approach. To deal with this issue, a strategy could be that of addressing the equivalent problems that consider absolute values instead of percentages. For instance, in Problem 9 instead of considering the local visibility of C with respect to C' as $\mathcal{V}_{C'}(C) = \frac{|C \cap C'|}{|C|}$ one could consider the measure $|C \cap C'|$. This would yield a new visibility definition and a new formulation for Problem 9 using an integer parameter that could be exploited for an FPT analysis.

6. Research targets for summarization problems

In Sections 4 and 5 we introduced and classified several problems falling in the broad research area of maximal clique summarization. The main research target is that of devising algorithms to find solutions to such problems. However, specific problems could be addressed also from different perspectives, either to better understand the underlying complexity of the questions or to propose solutions tailored to particular application contexts. Such alternative research targets include devising exact or enumerative algorithms, investigating hardness or W[1]-hardness, conceiving FPT approaches, applying probabilistic techniques, generalizing to weighted graphs or focusing on particular graph families, extending the results to graph with uncertainty, proposing approximated solutions. In this section we highlight some promising research directions based on the classification introduced in Section 4.

Filtering problems. Since Filtering problems ask to evaluate a Boolean property independently for each maximal clique, these problems are well-suited for enumeration algorithms, while probabilistic approaches do not appear to be particularly meaningful.

For Problem 1 (MAXIMAL CLIQUES BY SIZE), related to the size of maximal cliques, known results on W[1]-hardness [14,15] imply that designing FPT algorithms with respect to the size of the cliques is likely to be difficult. Instead, the research could focus on developing efficient enumerative algorithms, either with output-sensitive running time or with bounded-delay guarantees. When $s = \omega(G)$, randomized approaches for Problem 1 (MAXIMAL CLIQUES BY SIZE) have been presented in [95]. An alternative definition of clique size could be adopted for weighted input graphs, leading to a weighted variant of the problem (see Problem 20).

Regarding Problem 2 (ℓ-ISOLATED MAXIMAL CLIQUES), which focuses on the isolation of maximal cliques within the summary, positive results already exist in terms of FPT algorithms [50–52].

Search and reduced size problems. Search and Reduced Size problems are both defined through a Boolean property that has to be satisfied by the summary as a whole. Therefore, they often share common research goals. For several of these problems, the literature provides NP-hardness results, at least for specific values of the parameter. For example: Problem 9 (τ-VISIBLE MAXIMAL CLIQUE SMALL SUMMARY) coincides with the enumeration of the maximal cliques when $\tau = 1$, which implies the complexity bounds of the maximum clique problem discussed in Section 1. When $c = 1$, Problem 10 (c-COVERING MAXIMAL CLIQUE SMALL SUMMARY) is usually called CLIQUE COVER. Such a problem can have four different formulations, depending on whether the cliques are requested to be maximal and whether they can possibly intersect. The NP-hardness proof originally given by Karp [13, Problem 13] applies to non-maximal cliques, either disjoint (also called PARTITION INTO CLIQUES [96, Problem GT15]) or possibly intersecting. The problem of finding a small set of intersecting covering cliques can be easily reduced to that of finding covering intersecting maximal cliques and vice versa. Therefore, the latter problem is also NP-hard. To the best of our knowledge the complexity of finding a small set of covering maximal disjoint cliques is unknown. Observe that this problem coincides with Problem 14 (c-COVERING σ-INDEPENDENT MAXIMAL CLIQUE SMALL SUMMARY) when $c = 1$ and $\sigma = 0$. The existence version corresponds, instead, to Problem 5 (FIND ANY c-COVERING σ-INDEPENDENT MAXIMAL CLIQUE SUMMARY). Problem 6 (FIND ANY c-EDGE-COVERING σ-INDEPENDENT MAXIMAL CLIQUE SUMMARY) when $c = 1$ and $\sigma = 0$ is proven to be NP-hard in [59].

For Reduced Size problems, where the target is finding a "small" summary satisfying a given property, it make sense considering approximations of the minimum size summary. In this respect, very few results are known. For example, it can be proved that Problem 10 (c-COVERING MAXIMAL CLIQUE SMALL SUMMARY) for $c = 1$ does not admit, unless P = NP, a polynomial time approximation algorithm for any $\epsilon > 0$ that, on n-vertex graphs, achieves an approximation ratio better than $n^{1-\epsilon}$ on the number of maximal cliques covering the vertices of G. Indeed, the polynomial reduction from GRAPH COLORING to CLIQUE COVER provided by [13] maps the number of colors of the coloring to the number of (maximal) cliques covering the vertices of G. Therefore, the same hardness of approximation results for the former problem proved in [17] applies to the latter. Algorithmic FPT approaches could be devised using as FPT parameters the specific values in the problems definition (e.g., τ, σ, ρ, and c), as described in Section 5.4.

Probabilistic approaches could also prove useful in this setting, since the Boolean property is evaluated over the entire summary and the output property might be satisfied with a certain probability. Only Problem 25 (EXPECTED τ-VISIBLE MAXIMAL CLIQUE SMALL SUMMARY) has been addressed using a probabilistic algorithm [28,75].

Fixed size problems. All Fixed Size problems depend on a parameter k and, when $k = 1$, coincide with the search for one of the maximal cliques that maximize the objective function. Very often, these search problems are known to be NP-hard. For example, as noted in [46], Problem 15 (TOP-k MAXIMAL CLIQUES BY SIZE) and Problem 16 (SET OF k MAXIMAL CLIQUES MAXIMIZING COVERAGE) when $k = 1$ coincide with MAXIMUM CLIQUE. The NP-hardness of Problem 16 for an arbitrary k has been claimed in [78,79]. We remark that MAXIMUM CLIQUE admits an FPT algorithm in the degeneracy of the graph [97].

For Fixed Size problems, approximation approaches seem extremely challenging: in many cases it is not difficult to construct instances such that any k-summary that does not fully satisfy the problem condition is arbitrarily far from the optimum. For the special case of MAXIMUM CLIQUE, which, as said, is hard to approximate within a factor of $n^{1-\epsilon}$ [16,17], the best known approximation factor is $\mathcal{O}(n(\log\log n)^2/(\log n)^3)$ [98]. We refer to [99,100] for surveys on FPT approaches, W[1]-hardness, and approximation of MAXIMUM CLIQUE and its relaxations.

7. A review of summarization algorithms

In addition to outlining key problems and summarization principles, Tables 1–3 provide references to previous studies that addressed these problems, also highlighting which of them are still open. In this section we provide an in-depth review of algorithmic advancements, using the summarization principles introduced in Section 3 as a first level of classification.

7.1. Size-based summarization

Detecting a maximum clique. When $k = 1$, both Problem 15 (TOP-k MAXIMAL CLIQUES BY SIZE) and Problem 16 (SET OF k MAXIMAL CLIQUES MAXIMIZING COVERAGE) coincide with the problem of detecting a maximum clique. This is one of the most studied NP-hard problems. Several surveys on the maximum clique problem are available in the literature, including [101–103]. In [103], which is, to the best of our knowledge, the most recent, exact and heuristic approaches are considered in detail, and inapproximability results and parametric complexity completeness are briefly discussed. Benchmarks and experimental comparisons as well as generalizations to vertex-weighted and edge-weighted graphs are also discussed.

Enumerating maximum cliques. Problem 1 (MAXIMAL CLIQUES BY SIZE) is equivalent to the problem of enumerating all maximum cliques when only cliques of size $\omega(G)$ should be included in the summary, where $\omega(G)$ is the *clique number* of G, i.e., the number of vertices in any maximum clique. This problem has been addressed in [104] and in [95], providing exact and probabilistic solutions, respectively.

Enumerating large maximal cliques. In its general formulation, Problem 1 is addressed in [53,54], which propose an algorithmic approach that, before launching the enumeration of the large maximal cliques, performs two filtering steps to the input graph.

Since the summary S must contain only maximal cliques of size at least s, a vertex $v \in V$ cannot belong in any maximal clique of S unless its degree is at least $s-1$. The first filtering step recursively removes vertices based on this observation, actually computing the s-core of the input graph [105], an operation that can be efficiently performed [106].

An edge $e = (u, v)$ can participate to a maximal clique of size at least s only if both u and v have at least $s-1$ neighbors in common, i.e., $|N(u) \cap N(v)| \geq s-1$. Moreover, any vertex v in a maximal clique of size at least s must have at least $s-1$ neighbors, such that for each vertex u of them, it holds that $|N(v) \cap N(u)| \geq s-1$. Considering these two observations, the graph is filtered again by first removing edges and then removing vertices that cannot belong to a maximal clique of size at least s.

The enumeration algorithm explores the recursion tree of Algorithm 1 by means of a breadth-first search instead of a depth-first-search. Furthermore, when a vertex v is not added to the current solution, a *tabu list* is created with all the vertices in T that are not adjacent to v. The tabu lists are used to stop the exploration of a branch of the recursion tree when all the maximal cliques generated by that branch are guaranteed to have size less than s.

7.2. Overlap-based summarization

Problem 9 (τ-VISIBLE MAXIMAL CLIQUE SMALL SUMMARY) and its probabilistic relaxation Problem 25 (EXPECTED τ-VISIBLE MAXIMAL CLIQUE SMALL SUMMARY) are addressed in [28,75,76,107], where τ-visible summaries are computed by a modified Algorithm 1 approach discussed in Section 2. In particular, while traversing the Algorithm 1 recursive call tree, these approaches perform two kinds of operations:

RETENTION: when a maximal clique C is found a decision is taken about whether adding C to S or not;

PRUNING: when the current recursive call corresponds to a non-maximal clique a decision is taken about whether pruning the current recursion subtree, so reducing the considered solution space.

It can be observed that Algorithm 1 discussed in Section 2 yields an enumeration in which similar cliques tend to be near in the output sequence. Based on this observation, both the be above introduced operations can be accomplished by considering the last maximal clique C' added to S as a representative for all the cliques in S. In this case, RETENTION does not necessarily filter out all maximal cliques that have a visibility greater than τ with respect to maximal cliques already added to S, i.e., the produced summary is not guaranteed to be minimal. When performing PRUNING, instead, considering C' only does not affect the produced summary but only the computation time.

Pruning by lower bounding visibility. The criterion proposed in [28] to perform PRUNING is that of forecasting whether the subtree $\mathcal{T}$ of the complete search tree of Algorithm 1 rooted at the current node c and corresponding to clique C may contain a maximal clique that has low visibility with respect to C'. In particular, let $\underline{r}$ be a lower-bound on the visibility of the maximal cliques contained in $\mathcal{T}$ with respect to C', the current branch of the computation is pruned if $\underline{r} \geq \tau$. In order to compute $\underline{r}$, one needs to estimate the maximum size of a maximal clique X corresponding to a leaf x of $\mathcal{T}$, which is the size of C plus the depth d of $\mathcal{T}$, i.e., $|X| = |C| + d$. Second, one needs to estimate the size of the intersection between X and C', which may be given by $|C \cap C'|$ plus the number of vertices in $X \setminus C$ that are not in C'. Let d_y be the number of vertices of X that are not in C'. Then the size of the intersection $|X \cap C'| = |C \cap C'| + (d - d_y)$. Therefore, the lower-bound $\underline{r}$ on the visibility of X with respect to C' can be formulated as follows.

$$\underline{r} = \min_{1 \leq d \leq \overline{d}} \frac{|C \cap C'| + \max\{d - \overline{d_y}, 0\}}{|C| + d} \tag{9}$$

where $\overline{d}$ is an upper-bound on the depth d of $\mathcal{T}$; $\overline{d_y}$ is an upper-bound on d_y; and the max function is needed since $d - \overline{d_y}$ could be negative. It remains to discuss how to estimate $\overline{d}$ and $\overline{d_y}$. Recall that the set of candidate vertices that can possibly be added to C is denoted by T. Observe that an upper bound for d coincides with an upper bound for the maximum clique in $G[T]$. Hence, a trivial upper bound could be to consider the size of the candidate set T. In [28] three heuristics are proposed to upper bound d: (i) the maximum degree $\overline{d}_\Delta$ of any vertex in $G[T]$, (ii) the maximum value $\overline{d}_h$ of h such that there are h vertices with at least degree $h-1$ in $G[T]$, and (iii) the maximum core number $\overline{d}_{core}$ of $G[T]$. A fourth upper bound for d is provided in [107]: (iv) $\overline{d}_{truss}$ that is the maximum truss number of $G[T]$. Observe that:

$$\overline{d}_\Delta \geq \overline{d}_h \geq \overline{d}_{core} \geq \overline{d}_{truss} \tag{10}$$

The first inequality is trivial, the second one holds because $\overline{d}_{core}$ requires the h vertices to be connected and the third holds since a k-truss must be a $(k-1)$-core. Computational complexities for the four approaches above are, respectively:

$$\mathcal{O}(1) \leq \mathcal{O}(|V[T]|) \leq \mathcal{O}(|E[T]|) \leq \mathcal{O}(|E[T]|^{1.5}) \tag{11}$$

$\overline{d_y}$ can be estimated with $|T \setminus C'|$, or with the number of vertices in $T \setminus C'$ having degree at least d, or simply as the value of d.

Avoiding costly pruning operations. Even if the approach described above improves the running time with respect to the classic maximal clique enumeration algorithm, it still suffers from two main drawbacks: (a) it requires a new costly estimation of $\overline{d}$ independently at each recursion and (b) it tries to prune several times subtrees that are not prunable because of some common leaves. Observe that drawback (b) is also costly, since each pruning attempt needs an estimation of $\overline{r}$ which in turn requires an estimation of $\overline{d}$. To address these limitations, in [76] three novel keeping strategies are proposed to determine when a subtree is not prunable with the goal to avoid unnecessary bound estimation. In practice, these strategies consists of upper bounding strategies of visibility in contrast with the original lower bound approach exploited during the pruning attempt. Moreover, instead to calculate $\overline{d}$ from zero, an update approach is proposed to adjust the $\overline{d}$ from the parent recursion denoted by $\overline{d}_0$. More formally, each time before entering a new recursion with the current partial clique C, if the following inequality holds:

$$\frac{|C \cap C'|}{|C| + \overline{d}_0 - 1} \geq \tau \tag{12}$$

the current subtree can be pruned. Notice that this pruning strategy requires only $\mathcal{O}(1)$ extra time complexity.

We now discuss the keeping strategies. All of them are evaluated at each recursive step. If one of the following inequalities holds then the current subtree will be entirely explored without checking the pruning conditions. The inequalities guarantee that any maximal clique X generated from C cannot be witnessed by the last reported maximal clique C' in S. The first strategy is by *size filtering*

$$\frac{|C'|}{|C| + 1} < \tau \tag{13}$$

with a $\mathcal{O}(1)$ time complexity. The second one is by *bound reusing*

$$\frac{|C \cap C'| + \overline{d}_0 - 1}{|C| + \overline{d}_0 - 1} < \tau \tag{14}$$

with a $\mathcal{O}(1)$ time complexity. The last one is by *candidate tracing*

$$\frac{|C \cap C'| + |T \cap C'|}{|C| + 1} < \tau \tag{15}$$

with a time complexity of $\mathcal{O}(D/|X|)$ if degeneracy ordering is performed [20], where D is the degeneracy of G.

Putting it all together, the original approach can be refined processing sequentially the pruning and keeping strategies discussed before. In this way, only when all of them fail, the costly inequality $\underline{r} \geq \tau$ is evaluated.

A global retention approach. Instead of considering the visibility with respect to the last summarized clique, a different approach to remove redundancy might exploit a global view of summary S. In more details, when evaluating a maximal clique C, we might compare C against all the cliques in S. This is however a costly operation that requires an efficient implementation. A possible strategy is presented in [28]: clique C and any other clique of S are further compared only if the intersection of their vertex id ranges is nonempty.

7.2.1. Probabilistic method

In the randomized approach introduced in [28], Retention is performed with probability $1 - \sqrt[\overline{l}]{s(\underline{r})}$, where $\overline{l}$ is the denominator of Eq. (9), i.e., $\overline{l} = \overline{d} + |C|$, while Pruning is performed with probability $\sqrt[|C|]{s(r)}$, where $s(r)$ is a sampling probability function

$$s(r) = \frac{(1-r)(2-\tau)}{(2-r-\tau)} \tag{16}$$

that corresponds to a monotonically decreasing function with the property to have value 1 when $r = 0$ and value 0 when $r = 1$. Therefore, the larger is the estimated value of $\underline{r}$, the higher is the chance to effectively prune $\mathcal{T}$. Moreover, the larger is $\overline{d}$ (and thus $\overline{l}$) the larger is the number of recursion calls and the smaller is the probability of pruning $\mathcal{T}$.

In [75], a new sampling probability function is proposed to overcome two main issues on Eq. (16). The first is that when $r \in [\tau, 1]$, we always obtain $s(r) > 0$. This means that there is a positive probability to add the maximal clique into the summary, even if it is τ-visible with respect to S. The second issue is that when $r = 0$, we have $s(r) = 1$. In this situation, even if the newly generated maximal clique has vertices that are not covered in S, it could be covered by some future maximal cliques. So, it makes sense to have a non-null probability to sample the clique, especially considering that the aim is to compute an expected τ-visible summary. The sampling probability function proposed by [75] is

$$s_{opt}(r) = \begin{cases} \frac{\tau - r}{1 - r}, & \text{if } r \in [0, \tau) \\ 0, & \text{if } r \in [\tau, 1] \end{cases} \tag{17}$$

It can be proved [75] that the adoption of function $s_{opt}(r)$ guarantees to compute an expected τ-visible summary and that $s_{opt}(r) \leq s(r)$, $\forall r \in [0, 1]$, resulting in a more effective pruning strategy.

7.3. Covering-based summarization

Problem 16 (set of k maximal cliques maximizing coverage) has been addressed in [46,78,79] using very different approaches: a variation of Algorithm 1 for the maximal clique enumeration [46]; a configuration checking strategy [78]; and a strategy based on formal concept analysis [79].

7.3.1. Approaches based on maximal clique enumeration

The approach proposed in [46] to find a summary S is based on a simple change to the maximal clique enumeration algorithm described in Section 2 (Algorithm 1), where at Line 4 a maximal clique C is not necessarily reported. In order to describe this approach we introduce some definitions. Given a summary $S \subseteq \mathcal{M}(G)$ of maximal cliques of G, the *private-vertex-set* for each clique $C \in S$, denoted by $p(C, S)$, is the set of vertices in C that are not contained in any other maximal clique in S, i.e., $p(C, S) = C \setminus c(S \setminus \{C\})$. Intuitively, $p(C, S)$ is the set of vertices of G that would be removed from $c(S)$ if C was removed from S. Each vertex $v \in p(C, S)$ is called a *private vertex of C in S*. The *min-cover-clique of S*, denoted by $C_m(S)$, is the clique $C \in S$ with minimum private-vertex-set size $|p(C, S)|$, i.e., $C_m(S) = \arg\min_{C \in S}\{|p(C, S)|\}$. If more than one maximal clique of S has minimum $|p(C, S)|$ then one such clique is arbitrarily chosen to be the min-cover-clique of S. Intuitively, the min-cover-clique of S is the clique that reduces the coverage size $|c(S)|$ by the minimum possible when removed from S.

During the enumeration process, the first k maximal cliques are added to summary S and, upon discovering a new maximal clique C, the min-cover-clique $C_m(S)$ is replaced by C when this yields a summary S' with a sufficiently larger coverage, that is, when the private-vertex-set size of C in S' is such that $|p(C, S')| > |p(C_m(S), S)| + \alpha \cdot \frac{|c(S)|}{|S|}$, where α is a parameter $(0 \geq \alpha \geq 1)$. In [46] it is shown that, when $\alpha = 1$, the above approach computes a summary S, such that $|c(S)| \geq 0.25 \cdot |c(S^*)|$, where S^* is the optimal summary, i.e., the set of k maximal cliques that maximizes coverage.

Pursuing efficiency. A naïve implementation of the above described approach would be extremely inefficient, since each update of S to S' implies computing the min-cover-clique $C_m(S)$ of S, the size of the private-vertex-set of C in S', and the coverage size $|c(S)|$. Overall, this requires computing again the private-vertex-set of each maximal clique in S in the worst case. A data structure to efficiently address this problem is presented in [46]. This is based on the concept of 'reverse coverage' for vertices, i.e., the cliques of the summary S that a vertex belongs to. The reverse coverage is exploited to update a data structure that makes it possible to retrieve the aforementioned quantities (min-cover-clique, private-vertex-set size and coverage size) without the need of recomputing them from scratch.

It is proved in [46] that this implementation, while using only $\mathcal{O}(\sum_{C \in S} |C|)$ space, can produce a summary in the same time required for enumerating all the maximal cliques in G, in short, without requiring any extra asymptotic costs with respect to both time and space complexities.

On top of this approach, as shown in [46], local and global pruning strategies of the recursion tree can be efficiently implemented to reduce the search space. Global pruning prioritizes the expansion of the current clique toward vertices with a higher 'appetibility' score. This has the effect of allowing to prune those subtrees that correspond to low 'appetibility' score and that have no potential to be expanded to improve the quality of the current candidate. Local pruning is based on an upper-bound estimation of the potential of the maximal cliques contained into the subtree rooted at the current clique. Adding pruning does not increase the asymptotic running time but yields a substantial boost on the computation times in practice.

A further strategy used in [46] to improve efficiency is to start from a suitable set of initial candidate maximal cliques of S such that both global and local pruning conditions can be satisfied earlier.

7.3.2. Approaches based on configuration checking strategy

Another approach to solve Problem 16 is proposed in [78] and it is based on a variant of *Configuration Checking* (CC) [108] combined with an application of the *Best from Multiple Selection* (BMS) [109]. Configuration Checking is a simple strategy used to reduce the cycling phenomenon in a local search when solving combinatorial optimization problems. In this case, it is used to sample the solution space maintaining a tabu set to enforce the exploration of different portions of it. Since CC is not applicable to Problem 16, a variant of it named *Enhanced Configuration Checking* (ECC) is proposed in [78]. The idea of ECC is that a maximal clique can be added to S only if it does not contain tabu vertices.

1. When a maximal clique C is removed from S, every vertex $v \in p(C,S)$ is added to the tabu set and every tabu vertex u neighbor of v is removed from the tabu set.
2. When a maximal clique C is added to S, every tabu vertex neighbor of a vertex $v \in p(C,S)$ is removed from the tabu set.

A Boolean array $ConfChange$ can be used to implement the configuration checking strategy: each element of array $ConfChange$ refers to a vertex v and has value 0 if the vertex belongs to the tabu set and 1, otherwise. At the beginning each vertex $v \in V$ has $ConfChange[v] = 1$. Given a set of maximal cliques S, the *configuration* of a vertex $v \in V$ is the set $conf(v) = \{u | u \in N(v) \setminus c(S)\}$. Observe that $ConfChange[v] = 1$ either if v has never been considered so far or if a neighbor of v has been removed or added to $c(S)$, that is, if its configuration changed.

A clique C is first constructed starting from a random vertex v having $ConfChange[v] = 1$. A *candidate set* T is maintained, consisting of non-tabu neighbors of all vertices in C. Each vertex in T is associated with a benefit measure $\hat{b}[v] = |N(v) \cap T|$. While T is non-empty one vertex $v \in T$ is moved to C using the BMS strategy. If T is sufficiently small ($|T| < m$, where m is a suitable parameter), v is greedily chosen based on its benefit. Otherwise, v is selected among m random vertices of T maximizing the benefit. Observe that, the greater is m the higher is the probability of having a summary with high coverage, at the cost of a higher computation time. Using the above procedure the generated clique C may not be maximal in G as it could be contained into a larger clique involving tabu vertices. Therefore, in a post-processing phase, clique C is extended to a maximal one by iteratively adding tabu vertices adjacent to all vertices in C [110].

The general approach constructs several candidate summaries until a cut-off time is reached, choosing the one with highest coverage. The candidate summaries are computed using different values of m. Each candidate summary is built by adding k maximal cliques computed with the above described strategy, and then refining the solution for a bounded number of times by trying to replace the maximal clique C in S with smaller $|p(C,S)|$ with some other clique that improves $|c(S)|$.

7.3.3. Approaches based on formal concept analysis

A completely different approach to solve Problem 16 is proposed in [79] casting the problem into the framework of formal concept analysis (FCA), where the maximal cliques of a graph G correspond to equiconcepts in the formal concept lattice of G. The algorithm is based on a strategy that leverages FCA to find k-cliques originally proposed in [111].

In FCA, a triple $K = (O, A, I)$ is called a *formal context* if O and A are sets and $I \subseteq O \times A$. Elements of O are called *objects*, those of A are called *attributes*, whereas I is the *incidence* of the context K. For $X \subseteq O$, we define $X' = \{a \in A \mid \forall o \in X : (o,a) \in I\}$. Analogously, for $Y \subseteq A$, we define $Y' = \{o \in O \mid \forall a \in Y : (o,a) \in I\}$. A pair $H = (X,Y)$ is a *formal concept* of K if and only if $X \subseteq O$, $Y \subseteq A$, $X' = Y$, and $X = Y'$. Formal concepts of a given formal context are ordered by a partial ordering '$\leq$' defined as follows: $(X_1,Y_1) \leq (X_2,Y_2) \iff X_1 \subseteq X_2$ or $Y_2 \subseteq Y_1$. Let $C(K)$ denote the set of all formal concepts of K, then '$\leq$' is a partial relation of $C(K)$. A concept lattice $L = (C(K), \leq)$ can be obtained by all elements in $C(K)$ of a context K with the partial order $\leq$. An *equiconcept* is a special concept $H = (X,Y) \in C(K)$, such that $X = Y$. We denote by $EC(K)$ the set of all equiconcepts of $C(K)$. Finally, given three equiconcepts $H_1 = (X_1,Y_1)$, $H_2 = (X_2,Y_2)$ and $H_3 = (X_3,Y_3) \in EC(K)$, such that $H_1 \leq H_3$ and $H_2 \leq H_3$, then H_3 is denoted as the *father* concept of H_1 and H_2, and H_1 is denoted as the *brother* concept of H_2.

The *formal context K of a graph $G = (V,E)$* is constructed by choosing $O = A = V$ and, for every $(v_i,v_j) \in E$, both $(v_i,v_j) \in I$ and $(v_j,v_i) \in I$. Also, for every $v_i \in V$, $(v_i,v_i) \in I$.

It is proved in [79, Theorem 1] that $\mathcal{M}(G) = EC(K)$, i.e., the set $\mathcal{M}(G)$ of maximal cliques of G corresponds to the set $EC(K)$ of equiconcepts in the formal concept lattice $L = (C(K), \leq)$ obtained from the formal context K of G.

In order to obtain a summary S, $EC(K)$ is computed in a similar way as described in [111]. The procedure described in [79] adds k elements to S following an iterative approach where, at each step, the clique C corresponding to the equiconcept $H = (X,X)$ with the largest $|X|$ is added to S. After the addition, in order to pursue diversification, both H and all its brother equiconcepts are removed from $EC(K)$.

7.4. Isolation-based summarization

In this section, we review the results from the literature concerning Problem 2 (ℓ-ISOLATED MAXIMAL CLIQUES) that has been studied in [50–52]. We first discuss the approach proposed in [50] to obtain a summary S that contains all ℓ-isolated maximal cliques in a given graph $G = (V,E)$ with n vertices and m edges in $\mathcal{O}(\ell^4 \cdot 2^{2\ell} \cdot m)$ time. Then, we discuss the approach proposed in [51], which shows how to solve Problem 2 for the two other isolation concepts: when S must contain all max-ℓ-isolated maximal cliques or all min-ℓ-isolated maximal cliques.

In the following, we assume that the vertices of G are sorted by their degree such that $u < v \Rightarrow \delta(u) \leq \delta(v)$, meaning that we associate an *index* to each vertex. Let $N[v] = N(v) \cup \{v\}$. Let $N_+[v] = \{u \in N[v] \mid u > v\}$ and $N_-(v) = \{u \in N(v) \mid u < v\}$. Let $k = |N[v]|$. For any ℓ-isolated maximal clique C, there is a vertex in C, referred to as a *pivot*, that has less than ℓ edges outgoing from C. Observe that, an ℓ-isolated maximal clique may have more than one pivot. Using the vertex order, we can select the pivot to be the vertex in the maximal clique having the minimum index.

The general idea discussed in [50] to find the summary S is to scan each vertex v and enumerate all ℓ-isolated maximal cliques whose pivots are v, by removing at most $\ell - 1$ vertices from $N(v)$. For every vertex v, three main steps are performed, which are discussed in the following.

Trimming stage. First, if $|N_-(v)| > \ell - 1$, vertex v can safely be discarded as it cannot be a pivot of any ℓ-isolated maximal clique. Otherwise, a candidate set $C \subseteq N[v]$ is defined as $C = N[v] \setminus N_-(v) = \{v = u_1, \dots, u_k\}$. The idea is to remove from C all vertices that cannot belong to an ℓ-isolated maximal clique with v as the pivot. At the end, C contains all ℓ-isolated maximal cliques with v as the pivot. Additionally, let $C^{u_i} = \{w \in C \mid w < u_i\} \cup \{u_i\}$. The following conditions are then considered:

(a) $\delta(u_i) < (\ell + 1) \cdot |C| - 1$;
(b) u_i has less than $\ell \cdot |C|$ outgoing edges from C;

(c) u_i has at least $|C| - \ell$ neighbors in C;
(d) C^{u_i} has less than $\ell \cdot (\ell + 1) \cdot |C^{u_i}|$ outgoing edges.

For each vertex u_i, where $i = 1, \ldots, k$, condition (a) is tested and u_i is eventually removed from C. Next, conditions $(b) - (d)$ are checked for each vertex u_i, where $i = 2, \ldots, k$, and u_i is eventually removed from C. At any time, if the number of deleted vertices exceeds $\ell - 1$ (equivalently, if $|C| < k - \ell - 1$), then v cannot be a pivot and the entire procedure is terminated early.

Condition (a) ensures that high-degree vertices are early removed. Then, the other three conditions are evaluated only on small-degree vertices.

Enumerating stage. After the trimming stage, let k' denote the size of the candidate set C, with $k - \ell < k' \leq k$. At this point, C is a superset of all ℓ-isolated maximal cliques with pivot v. The goal is to find maximal cliques by removing at most $\ell' = \ell - 1 - (k - k')$ vertices from C. Observe that, if $C' \subseteq C$ is a clique, then $C \setminus C'$ forms a vertex cover in the complement graph $\overline{G[C]}$. Using this property, listing all maximal cliques in $G[C]$ that can be obtained by removing at most ℓ' vertices is equivalent to enumerating minimal vertex covers with at most ℓ' vertices in $\overline{G[C]}$. There exists an algorithm for this problem with a time complexity of $\mathcal{O}(2^{\ell'} \cdot \overline{m})$ [15], where $\overline{m}$ is the number of edges of $\overline{G[C]}$. We denote as Q_C the set of maximal cliques obtained at the end of this stage.

Screening stage. This final step aims to test whether elements in Q_C are indeed ℓ-isolated maximal cliques. Regarding isolation, each $Q \in Q_C$ can be efficiently tested to determine if it is ℓ-isolated; eventually Q is removed from Q_C. Regarding maximality, a simpler check is introduced in [51]. Since high-degree vertices are removed before the enumeration stage, all vertices in $N[v] \setminus Q$ must be considered, as one of these vertices could potentially be a common neighbor of all vertices in Q. Therefore, for each deleted vertex w from $N[v] \setminus C$, it is checked whether w is a neighbor of every vertex in Q. If so, Q is discarded. Otherwise, Q is added to S.

Other isolation concepts. We now discuss how the procedure described above can be adapted [51] to obtain a summary S that must contain either all min-ℓ-isolated maximal cliques or all max-ℓ-isolated maximal cliques. For both the isolation concepts, the enumeration stage remains unchanged.

When considering min-ℓ-isolated maximal cliques, recall that only maximal cliques with at least one vertex having less than ℓ neighbors outside the clique are relevant. During the trimming stage, vertices with few neighbors in the candidate set C must be removed because including them would result in a maximal clique where the pivot vertex v has at least ℓ outgoing edges. Only a single condition has to be checked for each vertex u_i:

(a') u_i has at least $|C| - \ell$ neighbors in C.

In the screening stage, only the vertices in $N_{-}(v)$ must be considered, as the enumerated cliques are maximal in $G[N_{+}[v]]$. For each $w \in N_{-}(v)$ and for each $Q \in Q_C$, it is tested whether $Q \subseteq N(w)$. If so, Q is discarded. Otherwise, Q is added to S.

When considering max-ℓ-isolated maximal cliques, recall that only maximal cliques in which every vertex has less than ℓ neighbors outside the clique are relevant. In the trimming stage, high-degree vertices are removed as in the case of ℓ-isolated maximal cliques, but since the max-isolation concept is stricter, the degree of the vertices is even more restricted compared to avg-isolation. Two conditions are checked for each vertex u_i:

(a") $\delta(u_i) < |C| + \ell - 1$;
(b") u_i has at least $|C| - \ell$ neighbors in C.

The screening stage remains unchanged with respect to avg-isolation.

8. Concluding remarks

The problem of summarizing maximal cliques in a graph can be approached in various ways, depending on the constraints imposed by specific application contexts. The sheer number of maximal cliques makes this task particularly challenging and computationally intensive. The literature on this topic is however highly scattered: the various contributions often lack cross-references, show limited awareness of the broader landscape, and thus miss opportunities to exploit potentially valuable techniques from related works.

In this survey, we introduced a novel taxonomy for classifying summarization problems along two key dimensions: summarization principles and problem classes. Our dual-axis approach provides a unifying framework that encompasses seemingly disparate problems and reveals underlying connections that were not previously well understood. To the best of our knowledge, this is the first attempt at offering a comprehensive view and comparison of different maximal clique summarization problems.

Several versions of maximal clique summarization (e.g., Problems 6 and 11) have a rich and well-known literature, especially devoted to combinatorial results. The interested reader could find some references in [58,59,74]. In this survey, we focus on the algorithmic aspects of maximal clique summarization, emphasizing the methods and approaches in the literature. Most of the contributions date back to the last decade and many problems are still open or have not been addressed at all (see Tables 1–3). Investigating such open problems would be beneficial not only for advancing the theoretical understanding of maximal clique summarization, but also for leveraging maximal cliques in further application contexts.

Our contribution paves the road for several promising research directions.

- As highlighted in Tables 1–3, many problems that can be cast in our framework and have practical applications have not been addressed in the literature. A better understanding of such open problems would add a significant contribution to the literature on clique summarization.
- Efficiency has not been sufficiently addressed so far, especially for Search and Reduced Size problems. There is potential to introduce novel algorithms that perform better on graphs with huge size. Moreover, since producing minimal summaries in Reduced Size problems is often hard, novel algorithms could be compared to the state of the art in terms of the size of the produced summaries.
- To improve scalability on large graphs, other computational models could be considered, such as cache-efficient, external memory, parallel, or distributed algorithms.
- Investigating streaming version of the problems would be crucial to handle dynamic graphs, such as those found in social or communication networks, where the structure of the graph changes over time.
- Algorithms tailored for graphs with special properties (e.g., sparseness, scale-freeness, hierarchical structure, small-world property, high clustering coefficient) may find applications in practical scenarios.
- There is a general lack of tools for experimental analysis, open-source software, and shared benchmarks with known optimal or high-quality summaries.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Maurizio Patrignani reports financial support was provided by Italian Ministry of University and Research (MUR). Irene Finocchi reports financial support was provided by Italian Ministry of University and Research (MUR). If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was supported in part by MUR PRIN Projects EXPAND and NextGRAAL [grant numbers 2022TS4Y3N and 2022ME9Z78].

Data availability

No data was used for the research described in the article.

References

[1] G.W. Flake, S. Lawrence, C.L. Giles, Efficient identification of web communities, in: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2000, pp. 150–160, http://dx.doi.org/10.1145/347090.347121.

[2] H. Saito, M. Toyoda, M. Kitsuregawa, K. Aihara, A large-scale study of link spam detection by graph algorithms, in: Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web, 2007, pp. 45–48, http://dx.doi.org/10.1145/1244408.1244417.

[3] M. D'Elia, I. Finocchi, M. Patrignani, Clique-TF-IDF: A new partitioning framework based on dense substructures, in: R. Basili, D. Lembo, C. Limongelli, A. Orlandini (Eds.), AIxIA 2023, in: Lecture Notes in Computer Science, vol. 14318, Springer, 2023, pp. 396–410, http://dx.doi.org/10.1007/978-3-031-47546-7_27.

[4] V. Boginski, S. Butenko, P.M. Pardalos, Statistical analysis of financial networks, Comput. Statist. Data Anal. 48 (2) (2005) 431–443, http://dx.doi.org/10.1016/J.CSDA.2004.02.004.

[5] V. Boginski, S. Butenko, P.M. Pardalos, Mining market data: A network approach, Comput. Oper. Res. 33 (11) (2006) 3171–3184, http://dx.doi.org/10.1016/J.COR.2005.01.027.

[6] S. Butenko, W.E. Wilhelm, Clique-detection models in computational biochemistry and genomics, European J. Oper. Res. 173 (1) (2006) 1–17, http://dx.doi.org/10.1016/J.EJOR.2005.05.026.

[7] E.J. Chesler, L. Lu, S. Shou, Y. Qu, J. Gu, J. Wang, H.C. Hsu, J.D. Mountz, N.E. Baldwin, M.A. Langston, et al., Complex trait analysis of gene expression uncovers polygenic and pleiotropic networks that modulate nervous system function, Nature Genet. 37 (3) (2005) 233–242, http://dx.doi.org/10.1038/ng1518.

[8] A. Töpfer, T. Marschall, R.A. Bull, F. Luciani, A. Schönhuth, N. Beerenwinkel, Viral quasispecies assembly via maximal clique enumeration, PLoS Comput. Biol. 10 (3) (2014) e1003515, http://dx.doi.org/10.1371/JOURNAL.PCBI.1003515.

[9] A. Bhar, L.C. Gierse, A. Meene, H. Wang, C. Karte, T. Schwaiger, C. Schröder, T.C. Mettenleiter, T. Urich, K. Riedel, L. Kaderali, Application of a maximal-clique based community detection algorithm to gut microbiome data reveals driver microbes during influenza A virus infection, Front. Microbiol. 13 (2022) http://dx.doi.org/10.3389/fmicb.2022.979320.

[10] F. Chen, H. Zhai, Y. Fang, Available bandwidth in multirate and multihop wireless ad hoc networks, IEEE J. Sel. Areas Commun. 28 (3) (2010) 299–307, http://dx.doi.org/10.1109/JSAC.2010.100402.

[11] K. Jain, J. Padhye, V.N. Padmanabhan, L. Qiu, Impact of interference on multi-hop wireless network performance, Wirel. Networks 11 (4) (2005) 471–487, http://dx.doi.org/10.1007/S11276-005-1769-9.

[12] B. Balasundaram, S. Butenko, Graph domination, coloring and cliques in telecommunications, in: M.G.C. Resende, P.M. Pardalos (Eds.), Handbook of Optimization in Telecommunications, Springer, 2006, pp. 865–890, http://dx.doi.org/10.1007/978-0-387-30165-5_30.

[13] R.M. Karp, Reducibility among combinatorial problems, in: R.E. Miller, J.W. Thatcher (Eds.), Proc. Symp. Compl. Comp. Computat., Plenum Press, New York, 1972, pp. 85–103, http://dx.doi.org/10.1007/978-1-4684-2001-2_9.

[14] R.G. Downey, M.R. Fellows, Fixed-parameter tractability and completeness II: On completeness for W[1], Theoret. Comput. Sci. 141 (1&2) (1995) 109–131, http://dx.doi.org/10.1016/0304-3975(94)00097-3.

[15] R.G. Downey, M.R. Fellows, Parameterized complexity, Monographs in Computer Science, Springer, 1999, http://dx.doi.org/10.1007/978-1-4612-0515-9.

[16] J. Hastad, Clique is hard to approximate within $n^{1-\epsilon}$, Acta Math. 182 (1) (1999) 105–142, http://dx.doi.org/10.1007/BF02392825.

[17] D. Zuckerman, Linear degree extractors and the inapproximability of max clique and chromatic number, in: J.M. Kleinberg (Ed.), Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006, ACM, 2006, pp. 681–690, http://dx.doi.org/10.1145/1132516.1132612.

[18] C. Bron, J. Kerbosch, Finding all cliques of an undirected graph (Algorithm 457), Commun. ACM 16 (9) (1973) 575–576, http://dx.doi.org/10.1145/362342.362367.

[19] F. Cazals, C. Karande, A note on the problem of reporting maximal cliques, Theoret. Comput. Sci. 407 (1–3) (2008) 564–568, http://dx.doi.org/10.1016/j.tcs.2008.05.010.

[20] D. Eppstein, M. Löffler, D. Strash, Listing all maximal cliques in large sparse real-world graphs, ACM J. Exp. Algorithmics 18 (2013) http://dx.doi.org/10.1145/2543629.

[21] I. Koch, Enumerating all connected maximal common subgraphs in two graphs, Theoret. Comput. Sci. 250 (1–2) (2001) 1–30, http://dx.doi.org/10.1016/S0304-3975(00)00286-3.

[22] E. Tomita, A. Tanaka, H. Takahashi, The worst-case time complexity for generating all maximal cliques, in: K. Chwa, J.I. Munro (Eds.), in: COCOON 2004, vol. 3106, Springer, Germany, 2004, pp. 161–170, http://dx.doi.org/10.1007/978-3-540-27798-9_19.

[23] Y. Jin, B. Xiong, K. He, Y. Zhou, Y. Zhou, On fast enumeration of maximal cliques in large graphs, Expert Syst. Appl. 187 (2022) 115915, http://dx.doi.org/10.1016/J.ESWA.2021.115915.

[24] J. Cheng, Y. Ke, A.W.-C. Fu, J.X. Yu, L. Zhu, Finding maximal cliques in massive networks, ACM Trans. Database Syst. 36 (4) (2011) 21, http://dx.doi.org/10.1145/2043652.2043654.

[25] J. Cheng, L. Zhu, Y. Ke, S. Chu, Fast algorithms for maximal clique enumeration with limited memory, in: KDD, 2012, pp. 1240–1248, http://dx.doi.org/10.1145/2339530.2339724.

[26] A. Conte, R. De Virgilio, A. Maccioni, M. Patrignani, R. Torlone, Finding all maximal cliques in very large social networks, in: EDBT 2016, OpenProceedings.org, Konstanz, Germany, 2016, pp. 173–184, http://dx.doi.org/10.5441/002/edbt.2016.18.

[27] Y. Xu, J. Cheng, A.W. Fu, Distributed maximal clique computation and management, IEEE Trans. Serv. Comput. 9 (1) (2016) 110–122, http://dx.doi.org/10.1109/TSC.2015.2479225.

[28] J. Wang, J. Cheng, A.W. Fu, Redundancy-aware maximal cliques, in: I.S. Dhillon, Y. Koren, R. Ghani, T.E. Senator, P. Bradley, R. Parekh, J. He, R.L. Grossman, R. Uthurusamy (Eds.), The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013, ACM, 2013, pp. 122–130, http://dx.doi.org/10.1145/2487575.2487689.

[29] F.N. Afrati, D. Fotakis, J.D. Ullman, Enumerating subgraph instances using map-reduce, in: C.S. Jensen, C.M. Jermaine, X. Zhou (Eds.), 29th IEEE International Conference on Data Engineering, ICDE 2013, IEEE Computer Society, 2013, pp. 62–73, http://dx.doi.org/10.1109/ICDE.2013.6544814.

[30] I. Finocchi, M. Finocchi, E.G. Fusco, Clique counting in MapReduce: Algorithms and experiments, ACM J. Exp. Algorithmics 20 (2015) 1.7:1–1.7:20, http://dx.doi.org/10.1145/2794080.

[31] E. Coppa, I. Finocchi, R.L. Garcia, Counting cliques in parallel without a cluster: Engineering a fork/join algorithm for shared-memory platforms, Inf. Sci. 496 (2019) 553–571, http://dx.doi.org/10.1016/J.INS.2018.07.018.

[32] S.B. Seidman, B.L. Foster, A graph-theoretic generalization of the clique concept*, J. Math. Sociol. 6 (1) (1978) 139–154, http://dx.doi.org/10.1080/0022250X.1978.9989883.

[33] A. Conte, D. Firmani, C. Mordente, M. Patrignani, R. Torlone, Fast enumeration of large k-plexes, in: T. Eliassi-Rad, R. Kumar (Eds.), 23rd SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2017, ACM, 2017, pp. 115–124, http://dx.doi.org/10.1145/3097983.3098031.

[34] A. Conte, D. Firmani, M. Patrignani, R. Torlone, A meta algorithm for finding large k-plexes, Knowl. Inf. Syst. 63 (2021) 1745–1769, http://dx.doi.org/10.1007/s10115-021-01570-8.

[35] A. Conte, D. Firmani, M. Patrignani, R. Torlone, Shared-nothing distributed enumeration of 2-plexes, in: P. Cui, E. Rundensteiner, D. Carmel, Q. He, J. Yu (Eds.), 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, ACM, 2019, pp. 2469–2472, http://dx.doi.org/10.1145/3357384.3358083.

[36] J. Abello, M.G.C. Resende, S. Sudarsky, Massive quasi-clique detection, in: S. Rajsbaum (Ed.), LATIN 2002: Theoretical Informatics, 5th Latin American Symposium, Cancun, Mexico, April 3-6, 2002, Proceedings, in: Lecture Notes in Computer Science, vol. 2286, Springer, 2002, pp. 598–612, http://dx.doi.org/10.1007/3-540-45995-2_51.

[37] I. Finocchi, R.L. Garcia, B. Sinaimeri, From stars to diamonds: Counting and listing almost complete subgraphs in large networks, Comput. J. 67 (6) (2024) 2151–2161, http://dx.doi.org/10.1093/COMJNL/BXAD129.

[38] R.J. Mokken, et al., Cliques, clubs and clans, Qual. Quant. 13 (2) (1979) 161–173, http://dx.doi.org/10.1007/BF00139635.

[39] H. Yu, A. Paccanaro, V. Trifonov, M. Gerstein, Predicting interactions in protein networks by completing defective cliques, Bioinform. 22 (7) (2006) 823–829, http://dx.doi.org/10.1093/BIOINFORMATICS/BTL014.

[40] J. Pattillo, N. Youssef, S. Butenko, Clique relaxation models in social network analysis, in: M.T. Thai, P.M. Pardalos (Eds.), Handbook of Optimization in Complex Networks: Communication and Social Networks, Springer New York, New York, NY, 2012, pp. 143–162, http://dx.doi.org/10.1007/978-1-4614-0857-4_5.

[41] J. Moon, L. Moser, On cliques in graphs, Israel J. Math. 3 (1965) 23–28, http://dx.doi.org/10.1007/BF02760024.

[42] E. Tomita, A. Tanaka, H. Takahashi, The worst-case time complexity for generating all maximal cliques and computational experiments, Theoret. Comput. Sci. 363 (1) (2006) 28–42, http://dx.doi.org/10.1016/J.TCS.2006.06.015.

[43] J. Fox, T. Roughgarden, C. Seshadhri, F. Wei, N. Wein, Finding cliques in social networks: A new distribution-free model, in: I. Chatzigiannakis, C. Kaklamanis, D. Marx, D. Sannella (Eds.), 45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic, in: LIPIcs, vol. 107, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018, pp. 55:1–55:15, http://dx.doi.org/10.4230/LIPICS.ICALP.2018.55.

[44] J. Fox, T. Roughgarden, C. Seshadhri, F. Wei, N. Wein, Finding cliques in social networks: A new distribution-free model, SIAM J. Comput. 49 (2) (2020) 448–464, http://dx.doi.org/10.1137/18M1210459.

[45] A. Conte, E. Tomita, On the overall and delay complexity of the CLIQUES and Bron-Kerbosch algorithms, Theoret. Comput. Sci. 899 (C) (2022) 1–24, http://dx.doi.org/10.1016/j.tcs.2021.11.005.

[46] L. Yuan, L. Qin, X. Lin, L. Chang, W. Zhang, Diversified top-k clique search, VLDB J. 25 (2) (2016) 171–196, http://dx.doi.org/10.1007/S00778-015-0408-Z.

[47] Y. Uno, Y. Ota, A. Uemichi, Web structure mining by isolated cliques, IEICE Trans. Inf. Syst. 90-D (12) (2007) 1998–2006, http://dx.doi.org/10.1093/IETISY/E90-D.12.1998.

[48] T. Shigezumi, Y. Uno, O. Watanabe, A new model for a scale-free hierarchical structure of isolated cliques, J. Graph Algorithms Appl. 15 (5) (2011) 661–682, http://dx.doi.org/10.7155/JGAA.00243.

[49] Y. Uno, F. Oguri, Contracted webgraphs - Scale-freeness and structure mining, IEICE Trans. Commun. 96-B (11) (2013) 2766–2773, http://dx.doi.org/10.1587/TRANSCOM.E96.B.2766.

[50] H. Ito, K. Iwama, Enumeration of isolated cliques and pseudo-cliques, ACM Trans. Algorithms 5 (4) (2009) 40:1–40:21, http://dx.doi.org/10.1145/1597036.1597044.

[51] F. Hüffner, C. Komusiewicz, H. Moser, R. Niedermeier, Isolation concepts for clique enumeration: Comparison and computational experiments, Theoret. Comput. Sci. 410 (52) (2009) 5384–5397, http://dx.doi.org/10.1016/J.TCS.2009.05.008.

[52] C. Komusiewicz, F. Hüffner, H. Moser, R. Niedermeier, Isolation concepts for efficiently enumerating dense subgraphs, Theoret. Comput. Sci. 410 (38–40) (2009) 3640–3654, http://dx.doi.org/10.1016/J.TCS.2009.04.021.

[53] N. Modani, K. Dey, Large maximal cliques enumeration in sparse graphs, in: J.G. Shanahan, S. Amer-Yahia, I. Manolescu, Y. Zhang, D.A. Evans, A. Kolcz, K. Choi, A. Chowdhury (Eds.), CIKM 2008, ACM, 2008, pp. 1377–1378, http://dx.doi.org/10.1145/1458082.1458288.

[54] N. Modani, K. Dey, Large maximal cliques enumeration in large sparse graphs, in: S. Chawla, K. Karlapalem, V. Pudi (Eds.), Proceedings of the 15th International Conference on Management of Data, December 9-12, 2009, International School of Information Management, Mysore, India, Computer Society of India, 2009.

[55] J. Bhasker, T. Samad, The clique-partitioning problem, Comput. Math. Appl. 22 (6) (1991) 1–11, http://dx.doi.org/10.1016/0898-1221(91)90001-K.

[56] J.T. Kim, D.R. Shin, New efficient clique partitioning algorithms for register-transfer synthesis of data paths, Journal- Korean Phys. Soc. 40 (2002) 754–758.

[57] S. Sundar, A. Singh, Two grouping-based metaheuristics for clique partitioning problem, Appl. Intell. 47 (2) (2017) 430–442, http://dx.doi.org/10.1007/S10489-017-0904-5.

[58] N. Pullman, H. Shank, W. Wallis, Clique coverings of graphs V: maximal-clique partitions, Bull. Aust. Math. Soc. 25 (3) (1982) 337–356, http://dx.doi.org/10.1017/S0004972700005414.

[59] I. Holyer, The NP-completeness of some edge-partition problems, SIAM J. Comput. 10 (4) (1981) 713–717, http://dx.doi.org/10.1137/0210054.

[60] P. Erdöos, R. Faudree, E.T. Ordman, Clique partitions and clique coverings, Discrete Math. 72 (1–3) (1988) 93–101, http://dx.doi.org/10.1016/0012-365X(88)90197-5.

[61] N.J. Pullman, Clique covering of graphs IV. Algorithms, SIAM J. Comput. 13 (1) (1984) 57–75, http://dx.doi.org/10.1137/0213005.

[62] D.A. Gregory, S. McGuinness, W. Wallis, Clique partitions of the cocktail party graph, Discrete Math. 59 (3) (1986) 267–273, http://dx.doi.org/10.1016/0012-365X(86)90173-1.

[63] F. Hüffner, C. Komusiewicz, H. Moser, R. Niedermeier, Fixed-parameter algorithms for cluster vertex deletion, Theory Comput. Syst. 47 (1) (2010) 196–217, http://dx.doi.org/10.1007/S00224-008-9150-X.

[64] A. Boral, M. Cygan, T. Kociumaka, M. Pilipczuk, A fast branching algorithm for cluster vertex deletion, Theory Comput. Syst. 58 (2) (2016) 357–376, http://dx.doi.org/10.1007/S00224-015-9631-7.

[65] Z. Ertem, E. Lykhovyd, Y. Wang, S. Butenko, The maximum independent union of cliques problem: complexity and exact approaches, J. Global Optim. 76 (3) (2020) 545–562, http://dx.doi.org/10.1007/S10898-018-0694-2.

[66] K. Tian, M. Xiao, B. Yang, Parameterized algorithms for cluster vertex deletion on degree-4 graphs and general graphs, in: W. Wu, G. Tong (Eds.), Computing and Combinatorics - 29th International Conference, COCOON 2023, Hawaii, HI, USA, December 15-17, 2023, Proceedings, Part I, in: Lecture Notes in Computer Science, vol. 14422, Springer, 2023, pp. 182–194, http://dx.doi.org/10.1007/978-3-031-49190-0_13.

[67] M. Aprile, M. Drescher, S. Fiorini, T. Huynh, A tight approximation algorithm for the cluster vertex deletion problem, Math. Program. 197 (2) (2023) 1069–1091, http://dx.doi.org/10.1007/S10107-021-01744-W.

[68] A. Natanzon, R. Shamir, R. Sharan, Complexity classification of some edge modification problems, Discrete Appl. Math. 113 (1) (2001) 109–128, http://dx.doi.org/10.1016/S0166-218X(00)00391-7.

[69] R. Shamir, R. Sharan, D. Tsur, Cluster graph modification problems, Discrete Appl. Math. 144 (1–2) (2004) 173–182, http://dx.doi.org/10.1016/J.DAM.2004.01.007.

[70] J. Gramm, J. Guo, F. Hüffner, R. Niedermeier, Graph-modeled data clustering: Exact algorithms for clique generation, Theory Comput. Syst. 38 (4) (2005) 373–392, http://dx.doi.org/10.1007/S00224-004-1178-Y.

[71] S. Böcker, P. Damaschke, Even faster parameterized cluster deletion and cluster editing, Inform. Process. Lett. 111 (14) (2011) 717–721, http://dx.doi.org/10.1016/J.IPL.2011.05.003.

[72] G.F. Italiano, A.L. Konstantinidis, C. Papadopoulos, Structural parameterization of cluster deletion, in: C. Lin, B.M.T. Lin, G. Liotta (Eds.), WALCOM: Algorithms and Computation - 17th International Conference and Workshops, WALCOM 2023, Hsinchu, Taiwan, March 22-24, 2023, Proceedings, in: Lecture Notes in Computer Science, vol. 13973, Springer, 2023, pp. 371–383, http://dx.doi.org/10.1007/978-3-031-27051-2_31.

[73] A. Dessmark, J. Jansson, A. Lingas, E. Lundell, M. Persson, On the approximability of maximum and minimum edge clique partition problems, Internat. J. Found. Comput. Sci. 18 (2) (2007) 217–226, http://dx.doi.org/10.1142/S0129054107004656.

[74] N.J. Pullman, Clique coverings of graphs—a survey, in: Combinatorial Mathematics X: Proceedings of the Conference Held in Adelaide, Australia, August 23–27, 1982, Springer, 2006, pp. 72–85, http://dx.doi.org/10.1007/BFb0071509.

[75] X. Li, R. Zhou, Y. Dai, L. Chen, C. Liu, Q. He, Y. Yang, Mining maximal clique summary with effective sampling, in: J. Wang, K. Shim, X. Wu (Eds.), 2019 IEEE International Conference on Data Mining, ICDM 2019, Beijing, China, November 8-11, 2019, IEEE, 2019, pp. 1198–1203, http://dx.doi.org/10.1109/ICDM.2019.00147.

[76] X. Li, R. Zhou, L. Chen, Y. Zhang, C. Liu, Q. He, Y. Yang, Finding a summary for all maximal cliques, in: 37th IEEE International Conference on Data Engineering, ICDE 2021, Chania, Greece, April 19-22, 2021, IEEE, 2021, pp. 1344–1355, http://dx.doi.org/10.1109/ICDE51399.2021.00120.

[77] J. Orlin, Contentment in graph theory: covering graphs with cliques, in: Indagationes Mathematicae (Proceedings), vol. 80, (no. 5) Elsevier, 1977, pp. 406–424, http://dx.doi.org/10.1016/1385-7258(77)90055-5.

[78] J. Wu, C. Li, L. Jiang, J. Zhou, M. Yin, Local search for diversified Top-*k* clique search problem, Comput. Oper. Res. 116 (2020) 104867, http://dx.doi.org/10.1016/J.COR.2019.104867.

[79] F. Hao, Z. Pei, L.T. Yang, Diversified top-k maximal clique detection in Social Internet of Things, Future Gener. Comput. Syst. 107 (2020) 408–417, http://dx.doi.org/10.1016/J.FUTURE.2020.02.023.

[80] J. Xue, J. Zheng, K. He, C. Li, Y. Liu, Diverteam: An effective evolutionary algorithm for diversified top-k (weight) clique search problems, in: U. Endriss, F.S. Melo, K. Bach, A.J.B. Diz, J.M. Alonso-Moral, S. Barro, F. Heintz (Eds.), ECAI 2024 - 27th European Conference on Artificial Intelligence, Vol. 392, 19-24 October 2024, Santiago de Compostela, Spain, IOS Press, 2024, pp. 4108–4115, http://dx.doi.org/10.3233/FAIA240981.

[81] K. Jansen, P. Scheffler, G. Woeginger, The disjoint cliques problem, RAIRO-Oper. Res. 31 (1) (1997) 45–66, http://dx.doi.org/10.1051/ro/1997310100451.

[82] B.P.W. Ames, S.A. Vavasis, Convex optimization for the planted k-disjoint-clique problem, Math. Program. 143 (1–2) (2014) 299–337, http://dx.doi.org/10.1007/S10107-013-0733-1.

[83] J. Wu, C.M. Li, Y. Zhou, M. Yin, X. Xu, D. Niu, HEA-D: A hybrid evolutionary algorithm for diversified top-k weight clique search problem, in: L.D. Raedt (Ed.), Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022, ijcai.org, 2022, pp. 4821–4827, http://dx.doi.org/10.24963/IJCAI.2022/668.

[84] J. Wu, M. Yin, A restart local search for solving diversified top-k weight clique search problem, Mathematics 9 (21) (2021) 2674, http://dx.doi.org/10.3390/math9212674.

[85] J. Zhou, C. Li, Y. Zhou, M. Li, L. Liang, J. Wang, Solving diversified top-k weight clique search problem, Sci. China Inf. Sci. 64 (5) (2021) http://dx.doi.org/10.1007/S11432-020-3069-4.

[86] A.P. Mukherjee, P. Xu, S. Tirthapura, Enumeration of maximal cliques from an uncertain graph, IEEE Trans. Knowl. Data Eng. 29 (3) (2017) 543–555, http://dx.doi.org/10.1109/TKDE.2016.2527643.

[87] R. Li, Q. Dai, G. Wang, Z. Ming, L. Qin, J.X. Yu, Improved algorithms for maximal clique search in uncertain networks, in: 35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, April 8-11, 2019, IEEE, 2019, pp. 1178–1189, http://dx.doi.org/10.1109/ICDE.2019.00108.

[88] Q. Dai, R. Li, M. Liao, H. Chen, G. Wang, Fast maximal clique enumeration on uncertain graphs: A pivot-based approach, in: Z.G. Ives, A. Bonifati, A.E. Abbadi (Eds.), SIGMOD '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022, ACM, 2022, pp. 2034–2047, http://dx.doi.org/10.1145/3514221.3526143.

[89] Z. Zou, J. Li, H. Gao, S. Zhang, Finding top-k maximal cliques in an uncertain graph, in: 2010 IEEE 26th International Conference on Data Engineering, ICDE 2010, 2010, pp. 649–652, http://dx.doi.org/10.1109/ICDE.2010.5447891.

[90] J. Bai, J. Zhou, M. Du, Z. Chen, Index-based top kα-maximal-clique enumeration over uncertain graphs, J. Supercomput. 78 (17) (2022) 19372–19400, http://dx.doi.org/10.1007/S11227-022-04613-1.

[91] K. Rozman, A. Ghysels, D. Janežič, J. Konc, An exact algorithm to find a maximum weight clique in a weighted undirected graph, Sci. Rep. 14 (1) (2024) 9118, http://dx.doi.org/10.1038/s41598-024-59689-x.

[92] W.J. Pullan, Approximating the maximum vertex/edge weighted clique using local search, J. Heuristics 14 (2) (2008) 117–134, http://dx.doi.org/10.1007/S10732-007-9026-2.

[93] D. Yu, L. Zhang, Q. Luo, X. Cheng, Z. Cai, Maximal clique search in weighted graphs, IEEE Trans. Knowl. Data Eng. 35 (9) (2023) 9421–9432, http://dx.doi.org/10.1109/TKDE.2023.3239409.

[94] S. Shimizu, K. Yamaguchi, S. Masuda, A maximum edge-weight clique extraction algorithm based on branch-and-bound, Discret. Optim. 37 (2020) 100583, http://dx.doi.org/10.1016/J.DISOPT.2020.100583.

[95] C. Lu, J.X. Yu, H. Wei, Y. Zhang, Enumerating maximum cliques in massive graphs, IEEE Trans. Knowl. Data Eng. 34 (9) (2022) 4215–4230, http://dx.doi.org/10.1109/TKDE.2020.3036013.

[96] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman, 1979.

[97] A. Buchanan, J.L. Walteros, S. Butenko, P.M. Pardalos, Solving maximum clique in sparse graphs: an o(nm+n2)$^{d/4}$ algorithm for d-degenerate graphs, Optim. Lett. 8 (5) (2014) 1611–1617, http://dx.doi.org/10.1007/S11590-013-0698-2.

[98] U. Feige, Approximating maximum clique by removing subgraphs, SIAM J. Discret. Math. 18 (2) (2004) 219–225, http://dx.doi.org/10.1137/S089548010240415X.

[99] C. Komusiewicz, Multivariate algorithmics for finding cohesive subnetworks, Algorithms 9 (1) (2016) 21, http://dx.doi.org/10.3390/A9010021.

[100] B. Balasundaram, F.M. Pajouh, Graph theoretic clique relaxations and applications, Handb. Comb. Optim. (2013) 1559–1598, http://dx.doi.org/10.1007/978-1-4419-7997-1_9.

[101] P.M. Pardalos, J. Xue, The maximum clique problem, J. Global Optim. 4 (1994) 301–328, http://dx.doi.org/10.1007/BF01098364.

[102] I.M. Bomze, M. Budinich, P.M. Pardalos, M. Pelillo, The maximum clique problem, in: Handbook of Combinatorial Optimization: Supplement Volume A, Springer, 1999, pp. 1–74, http://dx.doi.org/10.1007/978-1-4757-3023-4_1.

[103] Q. Wu, J. Hao, A review on algorithms for maximum clique problems, European J. Oper. Res. 242 (3) (2015) 693–709, http://dx.doi.org/10.1016/J.EJOR.2014.09.064.

[104] J.D. Eblen, C.A. Phillips, G.L. Rogers, M.A. Langston, The maximum clique enumeration problem: algorithms, applications, and implementations, BMC Bioinform. 13 (S-10) (2012) S5, http://dx.doi.org/10.1186/1471-2105-13-S10-S5.

[105] S.B. Seidman, Network structure and minimum degree, Soc. Networks 5 (3) (1983) 269–287, http://dx.doi.org/10.1016/0378-8733(83)90028-X.

[106] V. Batagelj, M. Zaversnik, An O(m) algorithm for cores decomposition of networks, 2003, http://dx.doi.org/10.48550/arXiv.cs/0310049, CoRR cs.DS/0310049.

[107] X. Li, R. Zhou, L. Chen, C. Liu, Q. He, Y. Yang, Efficiently finding a maximal clique summary via effective sampling, 2020, CoRR abs/2009.10376 arXiv:2009.10376.

[108] S. Cai, K. Su, A. Sattar, Local search with edge weighting and configuration checking heuristics for minimum vertex cover, Artificial Intelligence 175 (9–10) (2011) 1672–1696, http://dx.doi.org/10.1016/J.ARTINT.2011.03.003.

[109] S. Cai, Balance between complexity and quality: Local search for minimum vertex cover in massive graphs, in: Q. Yang, M.J. Wooldridge (Eds.), Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI, AAAI Press, 2015, pp. 747–753.

[110] J. Wu, C. Li, L. Jiang, J. Zhou, M. Yin, Personal communication, 2024.

[111] F. Hao, G. Min, Z. Pei, D. Park, L.T. Yang, K-clique community detection in social networks based on formal concept analysis, IEEE Syst. J. 11 (1) (2017) 250–259, http://dx.doi.org/10.1109/JSYST.2015.2433294.