Proc. 13th COCOON 2007, LNCS 4598, pp. 140-150, 2007

Isolation Concepts for Enumerating Dense Subgraphs

Christian Komusiewicz*, Falk Hüffner**, Hannes Moser***, and Rolf Niedermeier

Institut für Informatik, Friedrich-Schiller-Universität Jena, Ernst-Abbe-Platz 2, D-07743 Jena, Germany.
{ckomus,hueffner,moser,niedermr}@minet.uni-jena.de

Abstract. In a graph $G = (V, E)$, a vertex subset $S \subseteq V$ of size k is called c-isolated if it has less than $c \cdot k$ outgoing edges. We repair a nontrivially flawed algorithm for enumerating all c-isolated cliques due to Ito et al. [European Symposium on Algorithms 2005] and obtain an algorithm running in $O(4^c \cdot c^4 \cdot |E|)$ time. We describe a speedup trick that also helps parallelizing the enumeration. Moreover, we introduce a more restricted and a more general isolation concept and show that both lead to faster enumeration algorithms. Finally, we extend our considerations to s-plexes (a relaxation of the clique notion), pointing out a W[1]-hardness result and providing a fixed-parameter algorithm for enumerating isolated s-plexes.

1 Introduction

Finding and enumerating cliques and clique-like structures in graphs has many applications ranging from technical networks [9] to social and biological networks [1–3]. Unfortunately, clique-related problems are known to be notoriously hard for exact algorithms, approximation algorithms, and fixed-parameter algorithms [7, 5]. Ito et al. [9] introduced an interesting way out of this quandary by restricting the search to *isolated* cliques. Herein, given a graph $G = (V, E)$, a vertex subset $S \subseteq V$ of size k is called *c-isolated* if it has less than $c \cdot k$ outgoing edges. As their main result, Ito et al. [9] claimed an $O(4^c \cdot c^5 \cdot |E|)$ time algorithm for enumerating all c-isolated cliques in a graph. In particular, this means linear time for constant c and fixed-parameter tractability with respect to the parameter c. Unfortunately, the algorithm proposed by Ito et al. [9] suffers from serious deficiencies[1].

* Partially supported by the Deutsche Forschungsgemeinschaft, project OPAL (optimal solutions for hard problems in computational biology), NI 369/2.

** Supported by the Deutsche Forschungsgemeinschaft, Emmy Noether research group PIAF (fixed-parameter algorithms), NI 369/4.

*** Supported by the Deutsche Forschungsgemeinschaft, project ITKO (iterative compression for solving hard network problems), NI 369/5.

[1] A later manuscript [8] does not fundamentally resolve the problem.

We start with describing the algorithm of Ito et al. [9] and show that it does not fulfill the claimed running time bound. Then, we present some new results that eventually help us to repair Ito et al.'s approach, ending up with an algorithm that enumerates all c-isolated cliques in $O(4^c \cdot c^4 \cdot |E|)$ time. We also observe a speedup trick which seems to have high practical potential and which allows to parallelize the so far purely sequential enumeration algorithm.

Next, inspired by Ito et al.'s isolation concept, we propose two further isolation definitions, a weaker (less demanding) and a stronger concept, both practically motivated. Somewhat surprisingly, we can show that *both* concepts lead to faster enumeration algorithms for isolated cliques, improving the exponential factor from 4^c to 2^c and 2.44^c, respectively.

Finally, we show how to adapt the isolation scenario to the concept of s-plexes, a relaxation of cliques occurring in social networks analysis [15, 1]. In a graph $G = (V, E)$, a vertex subset $S \subseteq V$ of size k is called an *s-plex* if the minimum degree in $G[S]$ is at least $k - s$. First, strengthening an NP-hardness result of Balasundaram et al. [1], we point out that the problem of finding s-plexes is W[1]-hard with respect to the parameter k; that is, the problem seems as (parameterized) intractable as Clique is. This motivates our final result, a fixed-parameter algorithm (the parameter is the isolation factor) for constant s that enumerates all of one type of maximal isolated s-plexes. As a side result, here we improve a time bound for a generalized vertex cover problem first studied by Nishimura et al. [12].

Preliminaries. We consider only undirected graphs $G = (V, E)$ with $n := |V|$, $m := |E|$, $V(G) := V$, and $E(G) := E$. Let $N(v) := \{u \in V \mid \{u, v\} \in E\}$ and $N[v] := N(v) \cup \{v\}$. For $v \in V$, let $\deg_G(v) := |N(v)|$. For $A, B \subseteq V$, $A \cap B = \emptyset$, let $E(A, B) := \{\{u, v\} \mid u \in A, v \in B\}$. For $V' \subseteq V$, let $G[V']$ be the subgraph of G induced by V' and $G \setminus V' := G[V \setminus V']$. For $v \in V$, let $G - v := G[V \setminus \{v\}]$. A set S with property P is called *maximal* if no proper superset of S has property P, and *maximum* if no other set with property P has higher cardinality.

Parameterized complexity [5, 11] is an approach to finding optimal solutions for NP-hard problems. The idea is to accept the seemingly inevitable combinatorial explosion, but to confine it to one aspect of the problem, the *parameter*. If for relevant inputs this parameter remains small, then even large problems can be solved efficiently. More precisely, a problem of size n is *fixed-parameter tractable* (FPT) with respect to a parameter k if there is an algorithm solving it in $f(k) \cdot n^{O(1)}$ time.

Due to the lack of space, most proofs will appear in the full version of this paper. Some material also appears in [10].

2 Enumerating Isolated Cliques

We begin with describing Ito et al.'s algorithm for enumerating maximal c-isolated cliques [9]. Given a graph $G = (V, E)$ and an isolation factor c, first the vertices are sorted by their degree such that $u < v \Rightarrow \deg(u) \leq \deg(v)$. The *index* of a vertex is its position in this sorted order. For a vertex set $C \subseteq V$, an *outgoing edge* is an edge $\{u, v\}$ with $u \in C$ and $v \notin C$, and for a vertex $v \in C$, its outgoing edges are the outgoing edges of C that are incident on v. Let $N_+[v] := \{u \in N[v] \mid u \geq v\}$ and $N_-(v) := \{u \in N(v) \mid u < v\}$.

In a c-isolated clique, the vertex with the lowest index is called the *pivot* of the clique. Clearly, a pivot has less than c outgoing edges. Since every c-isolated clique has a pivot, we can enumerate all maximal c-isolated cliques of a graph by enumerating all maximal c-isolated cliques with pivot v for each $v \in V$ and then removing those c-isolated cliques with pivot v that are a subset of a c-isolated clique with another pivot.

The enumeration of maximal c-isolated cliques with pivot v for each $v \in V$ is the central part of the algorithm. We call this the *pivot procedure*. It comprises three successive stages.

Trimming stage. In this stage, we build a candidate set C that is a superset of all c-isolated cliques with pivot v. The candidate set C is initialized with $N_+[v]$, and then vertices that obviously cannot be part of a c-isolated clique with pivot v are removed from C. We refer to Ito et al. [9] for details.

Enumeration stage. In this stage, all maximal c-isolated cliques with pivot v are enumerated. Let C be the candidate set after the trimming stage, which deleted d vertices from $N_+[v]$. In total, we can delete only less than c vertices from $N_+[v]$, since otherwise v obtains too many outgoing edges. Therefore, $\tilde{c} := c - d - 1$ is the number of vertices that we may still remove from C. We can enumerate cliques $C' \subseteq C$ of size *at least* $|C| - \tilde{c}$ by enumerating vertex covers of size *at most* $\tilde{c}$ in the complement graph $\overline{G[C]}$. Ito et al. propose to enumerate *all* vertex covers of size at most $\tilde{c}$ [9]. We point to problems with this approach in Sect. 2.1.

Screening stage. In the screening stage, all cliques that are either not c-isolated or that are c-isolated but not maximal are removed. First the c-isolation is checked. Then those cliques that pass the test for isolation are compared pairwise, and we only keep maximal cliques. Finally, we check each clique that is left for pivot v against each clique obtained during calls to pivot(u) with $u \in N_-(v)$, since these are the only cliques that can be superset of a clique obtained for pivot v. The claimed overall running time, in the exponential part dominated by this last step, is then $O(4^c \cdot c^5 \cdot |E|)$ [9].

2.1 Problems with the Algorithm

The crucial part of the algorithm is the enumeration stage, in which the algorithm enumerates *all* vertex covers of size less than $\tilde{c}$. The authors argued that for a graph of size $|C|$, this can be done in $O(1.39^{\tilde{c}} \cdot \tilde{c}^2 + |C|^2)$ time. In contrast, Fernau [6] showed that the vertex covers of size *exactly* k in a graph can be enumerated in time $O(2^k k^2 + kn)$ if and only if k is the size of a minimum vertex cover of the graph and that otherwise no algorithm of running time $f(k) \cdot n^{O(1)}$

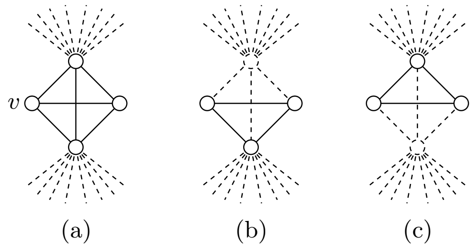

Fig. 1. Example for the enumeration stage with pivot v. *Solid lines* are edges between members of the clique; *dashed lines* are outgoing edges.

that enumerates all of these vertex covers exists, simply because there are too many. But in the course of the pivot procedure it may happen that we have to do just that: enumerate all vertex covers of size $\tilde{c}$ or less, where $\tilde{c}$ is not the size of a minimum vertex cover of $\overline{G[C]}$. Since this cannot be done in time $f(c) \cdot n^{O(1)}$, the algorithm does not yield fixed-parameter tractability with respect to the parameter c.

Figure 1 (a) illustrates such a situation. Consider the case $c = 4$ with v as pivot. No trimming takes place. This means that at the beginning of the enumeration stage, we may still remove up to $\tilde{c} = c - 1 = 3$ vertices from C to obtain a c-isolated clique. Since $C = N_+[v]$ forms a clique, the graph $\overline{G[C]}$ has only one minimum vertex cover, namely the empty set. This means that all subsets of $C \setminus \{v\}$ (v as pivot must not be eliminated from C) of size 3 or less are vertex covers that we would have to enumerate. Clearly, the number of such vertex covers is not only dependent on the size of the covers, but also on the size of $N_+[v]$. In our example, there are 8 such covers, and it is easy to see that we can increase the number of vertex covers simply by increasing the size of $N_+[v]$.

In contrast, enumeration of minimal vertex covers was shown to be *inclusion-minimally fixed parameter enumerable* [4]; in particular, all minimal solutions of size at most c can be enumerated in $O(2^c c^2 + m)$ time. So running time is not a problem here; however, we miss some c-isolated cliques when only considering minimal vertex covers. This is because we cannot simply discard a maximal clique that violates the isolation condition; it might have some subsets that *are* c-isolated. As an example, in Fig. 1 (a), the clique has 4 vertices and 16 outgoing edges and is thus not 4-isolated. However, two subsets ((b) and (c)) are cliques with 3 vertices and 11 outgoing edges, and thus are 4-isolated.

2.2 Repairing the Enumeration Stage

To cope with the problems described in Sect. 2.1, we propose a two-step approach for enumerating all maximal c-isolated cliques. First, we enumerate all *minimal* vertex covers and thus obtain maximal cliques in the candidate set C. Then, to also capture c-isolated cliques that are subsets of non-c-isolated cliques enumerated this way, for each of these cliques, we enumerate all maximal subsets that fulfill the isolation condition. The problem Isolated Clique Subset

procedure isolated-subset$(C, c, x_{\min})$

Input: A clique $C = \{v_1, v_2, \ldots, v_k\}$ with vertices sorted by degree, an isolation factor c and a minimum number $x_{\min}$ of outgoing edges from each vertex.

Output: The set of maximal c-isolated cliques $\mathcal{C}$ in C.

1: **foreach** $v \in C$: $x(v) := \deg(v) - |C| - 1 - x_{\min}$
2: $\hat{c} := c - x_{\min}$
3: $e(C) := (\sum_{v \in C} x(v)) - \hat{c} \cdot |C| + 1$
4: $\mathcal{D} := \{\emptyset\}$, $\mathcal{C} := \emptyset$
5: **repeat** $\hat{c}$ **times**
6:   **foreach** $D \in \mathcal{D}$
7:   **if** $C \setminus D$ is a c-isolated clique **then** $\mathcal{C} := \mathcal{C} \cup \{C \setminus D\}$
8:   **else**
9:   **if** $D = \emptyset$ **then** $i := k + 1$ **else** $i := \min_{v_l \in D}\{l\}$
10: $\mathcal{D} := \mathcal{D} \cup \{D \cup \{v_j\} \mid k - \hat{c} < j < i\}$
11: $\mathcal{D} := \mathcal{D} \setminus \{D\}$
12: **return** $\mathcal{C}$

Fig. 2. Algorithm for enumerating maximal c-isolated subsets of a clique C

of finding these c-isolated subsets is then: given a graph $G = (V, E)$ and a clique $C \subseteq V$, find a set $C' \subseteq C$ that forms a c-isolated vertex set, that is, a set with less than $c \cdot |C'|$ outgoing edges. The difficulty is in doing this fast enough, in particular with the running time depending only polynomially on $|C|$. For this (Theorem 1), the key is the following lemma, which reduces the choices of which vertices to omit from C.

Lemma 1. *Given a clique C with $|C| = k$, every maximal c-isolated subset of C is a superset of C^{k-c+1}, where C^{k-c+1} is the set of the $k - c + 1$ vertices with lowest index in C.*

According to Lemma 1, we may only remove vertices from the $c - 1$ vertices in $C \setminus C^{k-c+1}$ to obtain maximal c-isolated subsets of C. Hence, there are 2^{c-1} subsets of $C \setminus C^{k-c+1}$, and we enumerate maximal c-isolated subsets of C by generating the subsets of $C \setminus C^{k-c+1}$ in order of increasing cardinality and testing for each generated set whether its removal from C yields a maximal c-isolated subset. In this way, we can avoid examining supersets of removal sets for which a c-isolated clique was already output, since they would yield non-maximal cliques. The algorithm is shown in Fig. 2. Note that in lines 1–2 we compute an equivalent instance of Isolated Clique Subset with isolation factor $\hat{c}$ by decreasing the number of outgoing edges of each vertex by $x_{\min}$, where $x_{\min}$ is the minimum number of outgoing edges from each vertex in C. The computation of line 3 derives the number of outgoing edges above the threshold allowed by the isolation condition. With this, the condition in line 7 can be tested in constant time. This is needed to obtain the running time as claimed by the following theorem.

Theorem 1. *Given an instance of* Isolated Clique Subset *with at least* $x_{\min}$ *outgoing edges from each vertex, all of the at most* $O(2^{c-x_{\min}})$ *maximal solutions can be enumerated in* $O(2^{c-x_{\min}} + |C|)$ *time.*

We now describe how to use Theorem 1 to obtain a correct pivot procedure. Our modified pivot procedure differs from the original procedure only in the enumeration stage and in the screening stage. The enumeration stage is divided into two steps: the enumeration of maximal cliques and the enumeration of maximal subsets that fulfill the isolation condition for each of those cliques. Using Theorem 1, we can upper-bound the running time of the enumeration stage.

Lemma 2. *Given a graph* $G = (V, E)$*, a vertex* $v \in V$*, a set* $C \subseteq N_+[v]$*, and an isolation factor* c*, there are at most* $2^{c-1} \cdot c$ *maximal* c*-isolated cliques with pivot* v*, and they can be enumerated in* $O(2^c \cdot c^2 \cdot m(C))$ *time, where* $m(C)$ *is the number of edges in* $G[C]$*.*

In the screening stage, we filter non-maximal cliques by $O(4^c \cdot c^3)$ pairwise comparisons. Since the cliques obtained in the enumeration stage have size at most $\deg_G(v)$, these comparisons can be performed in $O(4^c \cdot c^3 \cdot \deg_G(v))$ time. With the running times of the stages of the pivot procedure for pivot v we can upper-bound the running time of the whole algorithm:

Theorem 2. *All maximal* c*-isolated cliques of a graph can be enumerated in* $O(4^c \cdot c^3 \cdot m)$ *time.*

2.3 Improved Screening of Cliques

In addition to fixing Ito et al.'s algorithm [9], we present an improved screening stage. While we can improve the asymptotic running time derived in Theorem 2 only slightly, the improvement facilitates parallelization of the enumeration algorithm and allows an exponential speedup for a variant of isolated clique enumeration to be presented in Sect. 3.2. More precisely, instead of a brute-force all-pairwise comparison, we achieve a simple and efficient test for checking whether an enumerated clique is subset of a clique with a different pivot.

Lemma 3. *A* c*-isolated clique* C *with pivot* v *is subset of a* c*-isolated clique* C' *with pivot* $u \neq v$ *iff* $u \in N_-(v)$ *and* $N(u) \supseteq C$*.*

Proof. We prove both directions separately. If $C' \supsetneq C$ is a clique with pivot u, then u must be adjacent to all vertices in C, in particular $u \in N_+[v]$. Since u is the pivot of C', it has lower index than v and thus $u \in N_-(v)$.

If there is a vertex $u \in N_-(v)$ that is adjacent to all vertices in C, then $C \cup \{u\}$ is a clique and a superset of C. It is furthermore c-isolated, since with u we have added a vertex with less than c outgoing edges (because $u < v$). Also, u is its pivot, again because $u < v$. □

According to Lemma 3, we can replace the pairwise comparisons between cliques enumerated in previous calls of the pivot procedure and those of the current call for pivot v with a simple test that looks for vertices in $N_-(v)$ that are adjacent to all vertices of an enumerated clique. This test takes $O(c \cdot |C|)$ time. Since the enumerations of cliques for different pivots now run completely independent from each other, we can parallelize our algorithm by executing the pivot procedures for different pivot vertices on up to n different processors. Unfortunately, the asymptotic running time derived in Theorem 2 remains largely unchanged, since there are still $O(4^c c^2)$ pairwise comparisons between cliques for a single pivot; however, we save a factor of c and there is also a conceivable speedup in practice since we significantly reduce the number of brute-force set comparisons.

3 Alternative Isolation Concepts

Since isolation is not merely a means of developing efficient algorithms for the enumeration of cliques but also a trait in its own right, it makes sense to consider varying degrees of isolation. For instance, this is useful for the enumeration of isolated dense subgraphs for the identification of communities, which play a strong role in the analysis of biological and social networks [13].

In this context, the definition of c-isolation is not particularly tailored to these applications and we propose two alternative isolation concepts. One of them, min-c-isolation, is a weaker notion than c-isolation and the other, max-c-isolation, is a stronger notion than c-isolation. For both isolation concepts, we achieve a considerable speedup in the exponential part of the running time.

3.1 Minimum Isolation

Min-c-isolation is a weaker concept of isolation than the previously defined c-isolation, since we only demand that a set contains at least one vertex with less than c outgoing edges.

Definition 1. *Given a graph $G = (V, E)$ and a vertex set $S \subseteq V$ of size k, S is* min-c-isolated *when there is at least one vertex in S with less than c neighbors in $V \setminus S$.*

Obviously, every c-isolated set is also min-c-isolated. The enumeration of maximal min-c-isolated cliques consequently yields sets that are at least as large and often larger than c-isolated cliques.

The algorithm for the enumeration of maximal min-c-isolated cliques is mainly a simplification of the algorithm from Sect. 2. However, we lose linear-time solvability in the case of constant isolation factors c—the running time then becomes $O(n \cdot m)$. We use the same pivot definition and enumerate cliques for each possible pivot; from our definition of min-c-isolation it follows directly that the pivot of a min-c-isolated clique must have less than c neighbors outside of the clique. Subsequently, we point out the differences in the three main stages of the pivot procedure.

In the trimming stage, we start with $C := N[v]$ as candidate set. After trimming, we can assume that every vertex u that was not removed has at least $|C|-c$ neighbors in C. In the enumeration stage, we simply enumerate minimal vertex covers in $\overline{G[C]}$ of size at most $\tilde{c}$, where $\tilde{c}$ is the number of vertices that can still be removed from the candidate set C. For each enumerated minimal vertex cover D, the set $C \setminus D$ is a maximal min-c-isolated clique. Hence, we need not test for maximality, but the enumerated cliques might contain a vertex with lower index than v, since we have not necessarily removed all vertices from $N_-(v)$. If a clique C' features a vertex with lower index than v, then C' is removed from the output. Compared to Theorem 2, the fact that we do not perform any maximality test results in an improved exponential part of the running time.

Theorem 3. *All maximal min-c-isolated cliques of a graph can be enumerated in $O(2^c \cdot c \cdot m + n \cdot m)$ time.*

3.2 Maximum Isolation

Compared to c-isolation, max-c-isolation is a stronger notion. This results in most cases in the enumeration of smaller cliques for equal values of c.

Definition 2. *Given a graph $G = (V, E)$ and a vertex set $S \subseteq V$ of size k, S is* max-c-isolated *if every vertex $v \in S$ has less than c neighbors in $V \setminus S$.*

This isolation concept is especially useful for graphs where the vertices have similar degrees. Consider for example a graph in which all vertices have the same degree. Here, the notions c-isolation and max-c-isolation become equivalent for cliques, but max-c-isolation allows a better worst-case running time.

We apply the algorithm scheme presented in Sect. 2, that is, for every vertex $v \in V$ we enumerate all maximal max-c-isolated cliques with pivot v.

Trimming Stage. We compute a candidate set $C \subseteq N_+[v]$ by removing every vertex from $N_+[v]$ that cannot be in a maximum max-c-isolated clique with pivot v.

Enumeration Stage. In this stage, we enumerate max-c-isolated cliques $C' \subseteq C$ with pivot v. As in Sect. 2, we first enumerate maximal cliques in C via enumeration of minimal vertex covers of size at most $\tilde{c}$ in $\overline{G[C]}$, where $\tilde{c}$ is the number of vertices that can still be removed from the candidate set C. The cliques thus obtained may violate the isolation condition, since they may contain vertices with too many outgoing edges. We can restore the isolation condition for each enumerated clique by simply removing these vertices. This is done until the resulting clique is either max-c-isolated or we have removed more than $\tilde{c}$ vertices. In the latter case we discard the clique. The remaining enumerated cliques are not necessarily maximal, and therefore non-maximal cliques must be removed from the output in the screening stage.

Screening Stage. There are two possibilities for an enumerated clique C to be non-maximal. First, it can be proper subset of another max-c-isolated clique with pivot v. Second, it can be proper subset of a max-c-isolated clique with pivot $u < v$. For the first possibility, we test whether there is a set of vertices $D \subseteq N_+[v] \setminus C$ such that $C \cup D$ is a max-c-isolated clique. Clearly, D has to form a clique and all its vertices have to be adjacent to all vertices in C. Furthermore, whenever D contains a vertex u with degree $|C| + c + x$, then $|D|$ must have size at least $x + 1$. Otherwise, $C \cup D$ is not max-c-isolated, because u has at least c outgoing edges from $C \cup D$. Hence, we test for all $0 \leq x < c - 1$ whether the set

$$D^x := \{w \in N_+[v] \setminus C \mid \; C \subseteq N(w) \land \deg(w) \leq |C| + c + x\}$$

contains a clique of size at least $x + 1$. If this is not the case for any x, then C is a maximal max-c-isolated clique for pivot v. Otherwise, C is removed from the output.

It remains to check whether C is a proper subset of a clique with another pivot $u < v$. This can be tested in the manner described in Sect. 2.3. The running time of the pivot procedure is dominated by the first maximality test of the screening stage. For each of the $O(2^c)$ enumerated cliques, we have to solve MAXIMUM CLIQUE up to c times. Since $|D^x| < c$ for all $0 \leq x < c - 1$, this can be done in $O(1.22^c)$ time [14]. The overall running time of this test is then

$$O(2^c \cdot 1.22^c \cdot c) = O(2.44^c \cdot c).$$

The running time of the whole enumeration can be bounded in a similar way as in Sect. 2.2.

Theorem 4. *All maximal max-c-isolated cliques of a graph can be enumerated in $O(2.44^c \cdot c \cdot m)$ time.*

4 Enumerating Isolated s-Plexes

In many applications such as social network analysis, cliques have been criticized for their overly restrictive nature or modelling disadvantages. Hence, more relaxed concepts of dense subgraphs such as s-plexes [15, 1] are of interest.

An s-plex is a degree-based relaxation of the clique concept. In a graph $G = (V, E)$, a subset of vertices $S \subseteq V$ of size k is called an *s-plex* if the minimum degree in $G[S]$ is at least $k - s$. It has been shown that the problem of deciding whether a graph has an s-plex of size k is NP-complete [1]. We strengthen this by the corresponding parameterized hardness result. The parameter-preserving reduction from CLIQUE is given in the full version of this paper. It shows that MAXIMUM s-PLEX is W[1]-hard with respect to the combined parameter (s, k) and thus also if parameterized only by either one of s and k. Therefore, as for CLIQUE, we rather consider isolation as parameter in terms of studying fixed-parameter tractability.

We present an algorithm for the enumeration of maximal min-c-isolated s-plexes that runs in FPT time with respect to parameter c for any constant s. In this paper, we have chosen to consider only min-c-isolation, since the enumeration algorithm is easier to describe with this isolation concept. A min-c-isolated s-plex S contains at least one vertex that has less than c neighbors in $V \setminus S$. Compared to the enumeration of maximal min-c-isolated cliques, we face two obstacles when enumerating maximal min-c-isolated s-plexes. First, we cannot use the algorithm for the enumeration of minimal vertex covers, since an s-plex does not necessarily induce an independent set in the complement graph. Instead, since in an s-plex S of size k every vertex $v \in S$ is adjacent to at least $k - s$ vertices, the subgraph induced by S in the complement graph $\overline{G[S]}$ is a graph with maximum degree at most $s - 1$. Consider therefore the following generalization of a vertex cover:

Definition 3. *Given a graph G and a nonnegative integer d, we call a subset of vertices $S \subseteq V$ a* max-deg-d deletion set *if $G[V \setminus S]$ has maximum degree at most d.*

The idea is to enumerate maximal s-plexes in G by enumerating minimal max-deg-d deletion sets in $\bar{G}$. We present a fixed-parameter algorithm for the enumeration of minimal max-deg-d deletion sets that uses the size of the solution sets as parameter.

Finding a minimum max-deg-d deletion set was also considered by Nishimura et al. [12], who presented an $O((d+k)^{k+3} \cdot k + n(d+k))$ time algorithm for the decision version. We improve the exponential part of this running time while also covering the enumeration version. The idea is to pick a vertex v with more than d neighbors and then branch into $d+2$ cases corresponding to the deletion of v or the deletion of one of the $d+1$ first neighbors of v:

Lemma 4. *Given a graph G and an integer k, all minimal max-deg-d deletion sets of size at most k can be enumerated in $O((d+2)^k \cdot (k+d)^2 + m)$ time.*

The second obstacle lies in the fact that given a pivot vertex v, maximal min-c-isolated s-plexes with pivot v are not necessarily a subset of $N_+[v]$, since they can contain up to $s - 1$ vertices that are not adjacent to v. We deal with this by enumerating all maximal min-c-isolated s-plexes for a given pivot *set* instead of a single pivot. The *pivot set* of a min-c-isolated s-plex is defined as the set that contains the pivot vertex v of the s-plex and those vertices that belong to the s-plex but are not adjacent to v. The *pivot vertex* is defined as the vertex with lowest index among the vertices with less than c neighbors outside of the s-plex. There has to be at least one such vertex, since otherwise the condition of min-c-isolation would be violated, but it does not necessarily have to be the vertex with the lowest index of all vertices in the s-plex.

The enumeration algorithm also consists of the three stages. In the trimming stage, we build a candidate set C by removing vertices from $N(v)$ that obviously cannot belong to a min-c-isolated s-plex with pivot v. For each possible pivot set P with pivot v, we independently enumerate the maximal min-c-isolated s-plexes. This is done in the enumeration stage by first building the complement graph $\overline{G[C \cup P]}$ and then enumerating minimal max-deg-$(s-1)$ deletion sets of size at most $c-1$ in $\overline{G[C \cup P]}$. In the screening stage, we first test whether any of the enumerated min-c-isolated s-plexes contains a vertex $u < v$, where u has less than c neighbors outside of the s-plex. Then this s-plex has pivot u and not v and is therefore removed from the output. Finally we perform a maximality test and remove non-maximal min-c-isolated s-plexes from the output.

Theorem 5. *All maximal min-c-isolated s-plexes of a graph can be enumerated in $O((s+1)^c \cdot (s+c) \cdot n^{s+1} + n \cdot m)$ time.*

Thus, for every fixed s, we obtain a fixed-parameter algorithm for enumerating all maximal min-c-isolated s-plexes with respect to the parameter c.

Acknowledgement. We thank H. Ito and K. Iwama (Kyoto) for making their manuscript [8] available to us and J. Guo (Jena) for the idea for Lemma 4.

References

1. B. Balasundaram, S. Butenko, I. V. Hicks, and S. Sachdeva. Clique relaxations in social network analysis: The maximum k-plex problem. Manuscript, 2006.
2. B. Balasundaram, S. Butenko, and S. Trukhanovzu. Novel approaches for analyzing biological networks. *Journal of Combinatorial Optimization*, 10(1):23–39, 2005.
3. S. Butenko and W. E. Wilhelm. Clique-detection models in computational biochemistry and genomics. *European Journal of Operational Research*, 173(1):1–17, 2006.
4. P. Damaschke. Parameterized enumeration, transversals, and imperfect phylogeny reconstruction. *Theoretical Computer Science*, 351(3):337–350, 2006.
5. R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Springer, 1999.
6. H. Fernau. On parameterized enumeration. In *Proc. 8th COCOON*, volume 2387 of *LNCS*, pages 564–573. Springer, 2002.
7. J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182(1):105–142, 1999.
8. H. Ito and K. Iwama. Enumeration of isolated cliques and pseudo-cliques. Manuscript, Aug. 2006.
9. H. Ito, K. Iwama, and T. Osumi. Linear-time enumeration of isolated cliques. In *Proc. 13th ESA*, volume 3669 of *LNCS*, pages 119–130. Springer, 2005.
10. C. Komusiewicz. Various isolation concepts for the enumeration of dense subgraphs. Diplomarbeit, Institut für Informatik, Friedrich-Schiller Universität Jena, 2007.
11. R. Niedermeier. *Invitation to Fixed-Parameter Algorithms.* Oxford University Press, 2006.
12. N. Nishimura, P. Ragde, and D. M. Thilikos. Fast fixed-parameter tractable algorithms for nontrivial generalizations of vertex cover. *Discrete Applied Mathematics*, 152(1–3):229–245, 2005.
13. G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.
14. J. M. Robson. Algorithms for maximum independent sets. *Journal of Algorithms*, 7(3):425–440, 1986.
15. S. B. Seidman and B. L. Foster. A graph-theoretic generalization of the clique concept. *Journal of Mathematical Sociology*, 6(1):139–154, 1978.