Isolation Concepts for Clique Enumeration: Comparison and Computational Experiments [⋆]

Falk Hüffner [a,1] Christian Komusiewicz [b,2] Hannes Moser [b,3]
Rolf Niedermeier [b]

[a] *School of Computer Science, Tel Aviv University,*
69978 Tel Aviv, Israel

[b] *Institut für Informatik, Friedrich-Schiller-Universität Jena,*
Ernst-Abbe-Platz 2, D-07743 Jena, Germany

Abstract

We do computational studies concerning the enumeration of isolated cliques in graphs. Isolation, as recently introduced, measures the degree of connectedness of the cliques to the rest of the graph. Isolation helps both in getting faster algorithms than for the enumeration of maximal general cliques and in filtering out cliques with special semantics. We compare three isolation concepts and their combination with two enumeration modi for maximal cliques ("isolated maximal" vs "maximal isolated"). All studied concepts exhibit the fixed-parameter tractability of the enumeration task with respect to the parameter "degree of isolation". We provide a first systematic experimental study of the corresponding enumeration algorithms, using synthetic graphs (in the $G_{n,m,p}$ model), financial networks, and a music artist similarity network (proposing the enumeration of isolated cliques as a useful instrument in analyzing financial and social networks).

Key words: NP-hard problem, fixed-parameter tractability, exact algorithm, algorithm engineering, dense subgraph

1 Introduction

We study the enumeration of maximal cliques of an undirected graph $G = (V, E)$, that is, the enumeration of all vertex subsets $V' \subseteq V$ such that the induced subgraph $G[V']$ is complete and there is no $V'' \supsetneq V'$ such that the induced subgraph $G[V'']$ is also complete. Unfortunately, already finding one maximum-cardinality clique is a notoriously hard computational problem, being NP-hard [10] as well as W[1]-hard [8] and hard to approximate [11]. By way of contrast, finding cliques is very important in many practical applications and has been subject of the second DIMACS implementation challenge in 1996. Recent papers describe applications in computational finance [2, 3] and computational biochemistry and genomics [5, 7]. Moreover, clique finding also plays a role in classical computer science fields such as the analysis of web graphs for instance to identify web communities [9] or "link farms" (for the purpose of spam deletion and analysis) [20].

Enumerating all maximal cliques needs exponential time. For instance, a recent paper by Tomita et al. [21] proved a worst-case time complexity of $\Theta(3^{n/3})$ for an n-vertex graph, arguing for its optimality due to the fact that there are example graphs having $3^{n/3}$ maximal cliques [18]. Ito et al. [14] (also see the journal version [13]) proposed to restrict the search to certain types of cliques, that is, specifically *isolated cliques*. A clique V' of k vertices is called *c-isolated* in a graph G if there are less than $c \cdot k$ edges leaving the induced subgraph $G[V']$ in G. This concept is interesting for two reasons. First, since one does not

[⋆] A preliminary version of this paper appeared in the proceedings of the 2nd Annual International Conference on Combinatorial Optimization and Applications (COCOA '08), held in St. John's, Newfoundland, Canada, August 21–24 [12]. Note that in our conference version [12] we state that the work by Ito et al. [14], which is the conference version of Ito and Iwama [13], is flawed. A reason for this statement was a misunderstanding of Ito et al.'s [14] enumeration concept: they enumerate all isolated maximal cliques, and not all maximal isolated cliques, as we were assuming.

Email addresses: hueffner@tau.ac.il (Falk Hüffner), c.komus@uni-jena.de (Christian Komusiewicz), hannes.moser@uni-jena.de (Hannes Moser),
rolf.niedermeier@uni-jena.de (Rolf Niedermeier).

[1] Supported by the Deutsche Forschungsgemeinschaft, Emmy Noether research group PIAF (fixed-parameter algorithms), NI 369/4, and the Edmond J. Safra Foundation. Major parts of the work done while the author was with the Friedrich-Schiller-Universität Jena.

[2] Partially supported by the Deutsche Forschungsgemeinschaft, project OPAL (optimal solutions for hard problems in computational biology), NI 369/2, and a PhD fellowship of the Carl-Zeiss-Stiftung.

[3] Supported by the Deutsche Forschungsgemeinschaft, project ITKO (iterative compression for solving hard network problems), NI 369/5, and project AREG (algorithms for generating quasi-regular structures in graphs), NI 369/9.

search for all maximal cliques anymore, faster enumeration algorithms are possible. Second, isolated cliques may be an intrinsically relevant concept, because these cliques can represent structures with particularly interesting properties that are detected in this way.

In a companion paper [16], we extended Ito et al.'s [14, 13] studies by distinguishing three natural isolation concepts instead of only one. Moreover, whereas Ito et al. considered the task to enumerate all maximal cliques that are isolated ("isolated maximal"), in our follow-up work [16] we studied the task to enumerate cliques that are maximal additionally with respect to the property of being isolated ("maximal isolated"). For the first time, here we systematically compare all different combination possibilities, after all leading to five different fixed-parameter algorithms (see [19] for more on fixed-parameter algorithms) with respect to the parameter c for enumerating c-isolated cliques. A typical running time of these algorithms looks like $O(2^c \cdot c^5 \cdot |E|)$.

The main focus of our work is on computational experiments to explore the practical utility of the new clique enumeration algorithms based on isolation. Before doing that, we provide a systematic comparison of all five enumeration algorithms for isolated cliques. In particular, we can improve a previous running time of $O(4^c \cdot c^4 \cdot |E|)$ [13] to $O(2^c \cdot c^5 \cdot |E|)$ for one of the mentioned enumeration algorithms. Our experiments are based on synthetic ($G_{n,m,p}$ graphs) as well as real-world data (financial networks, music artist similarity network). With the help of the corresponding empirical investigations we can spot subtle but important (practical) differences between the various isolation concepts and enumeration tasks. The main conclusion substantiated by our findings is that the consideration of isolated cliques pays off because one may

- achieve faster algorithms for relevant special cases of clique enumeration in comparison with the famous Bron-Kerbosch algorithm and its variants [4, 15, 21, 6], and
- isolation concepts help filtering out semantically particularly interesting maximal cliques that may have remained undiscovered when enumerating all maximal cliques.

In conclusion, we believe that with our study (and the corresponding, freely available open source code) we contribute a practically useful tool for clique-based network analysis in all fields of applications.

Our work is organized as follows. In the next section, we overview and compare known theoretical results and present a small improvement concerning the running time of one of the studied fixed-parameter algorithms. Then, in Section 3, we discuss relevant issues concerning the implementation of the algorithms experimented with in Section 4. There, we first present results for synthetic data—emphasizing efficiency issues—and then we present results for real-world data—emphasizing semantic issues. In Section 5, we draw some conclusions for future work and briefly summarize our findings and their potential impact on clique-based network analysis.

2 Isolation Concepts and Maximality Definitions

In this section, we survey the different isolation concepts and the resulting enumeration algorithms. Furthermore, for two particular enumeration algorithms, we present an improved upper bound on the worst-case running time.

2.1 Comparison of the Isolation Concepts and Enumeration Tasks

Ito et al. [13, 14] introduced the concept of *c-isolation*—which, in the light of the following, is called *average-c*-isolation (*avg*-c-isolation for short) in this work—as follows: Let $G = (V, E)$ be an undirected graph and c be a positive integer. A vertex set $S \subseteq V$ of size k is called *avg-c-isolated* if it has less than $c \cdot k$ outgoing edges, where an outgoing edge is an edge between a vertex in S and a vertex in $V \setminus S$. Note that for reasons of simplicity we consider c to be a nonnegative integer. In follow up-work [16], we further introduced the concepts of min-c-isolation and max-c-isolation as follows. A vertex set $S \subseteq V$ is *min-c-isolated* if there is at least one vertex in S with less than c neighbors in $V \setminus S$. A vertex set $S \subseteq V$ is *max-c-isolated* if every vertex $v \in S$ has less than c neighbors in $V \setminus S$. Figure 1 illustrates the three concepts in case of S inducing a clique. Clearly, max-c-isolatedness implies avg-c-isolatedness implies min-c-isolatedness, but not vice versa. Max-c-isolation is useful when we want to exclude high-degree vertices from the enumerated sets. This can result in the enumeration of smaller cliques than in the other two cases. For notational simplification we will mostly use the terms min-isolation, avg-isolation, and max-isolation.

In addition to these three isolation concepts, we distinguish between two different enumeration tasks. In the first enumeration setting (as described by Ito and Iwama [13]), we want to output maximal cliques that are also isolated.

Definition 1 *Let G be a graph and $\mathcal{I}$ be an isolation concept. A vertex set S is called* $\mathcal{I}$-isolated maximal clique *if S is a maximal clique and $\mathcal{I}$-isolated.*

In contrast, we also proposed to enumerate cliques that are maximal with respect to the clique property *and* the isolation condition [16].

Definition 2 *Let G be a graph and $\mathcal{I}$ be an isolation concept. A vertex set $S \subseteq V$ is called* maximal $\mathcal{I}$-isolated clique *if S is an $\mathcal{I}$-isolated clique, and no*

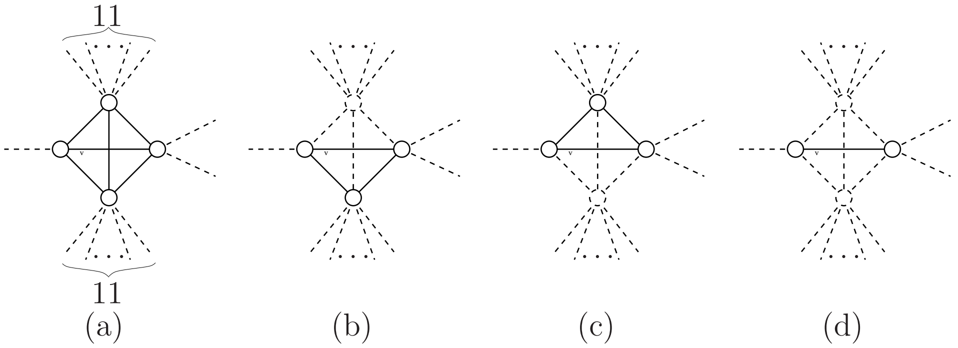


Fig. 1. Examples of isolated cliques. *Solid lines* are edges between members of a considered clique; *dashed lines* are outgoing edges. (a) A maximal 4-vertex clique, which is min-2-isolated, avg-7-isolated, and max-12-isolated. The clique is not avg-6-isolated; however, two subsets form a maximal avg-6-isolated clique ((b) and (c)). Moreover, it is not max-11-isolated, but one (unique) subset is (d).

superset $S' \supset S$ is an $\mathcal{I}$-isolated clique.

An example for the difference between these two definitions can be seen in Figure 1: the clique in Figure 1a is an avg-12-isolated maximal clique, since it is a maximal clique that is avg-12-isolated. The clique in Figure 1b is a maximal avg-6-isolated clique, but not avg-6-isolated maximal, since it is not a maximal clique. For all isolation conditions, the set of maximal isolated cliques always contains the set of isolated maximal cliques. Note that for min-isolation the two notions are identical: Since adding a vertex to a clique never results in a violation of min-isolation, every maximal min-isolated clique is also a min-isolated maximal clique. Altogether, we end up with five different enumeration tasks.

2.2 Enumeration Algorithms

In the following, we describe the algorithms for all five enumeration tasks. The overall structure of these algorithms is similar. First, we describe this structure, followed by the pseudo-codes of the algorithms, which are presented in tabular form in order to simplify comparisons between the algorithms.

We consider only undirected graphs $G = (V, E)$. For $v \in V, N(v) := \{u \in V \mid \{u, v\} \in E\}$ and $N[v] := N(v) \cup \{v\}$. Let c be the isolation factor, let n denote the number of vertices in G, and m denote the number of edges in G. 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. Let $N_+[v] := \{u \in N[v] \mid u > v\} \cup \{v\}$ and $N_-(v) := \{u \in N(v) \mid u < v\}$. In any isolated clique (according to our respective definition of isolation), the vertex with the lowest index is called the *pivot* of the clique [14]. Clearly, a pivot has less than c outgoing edges. Since every isolated clique has a pivot, we can enumerate all

```
Isolated Cliques($G = (V, E), c$)
Sort vertices in $G$ by degree
$\mathcal{C} \leftarrow \emptyset$
for each $v \in V$:                                  ▷ Pivot procedure
    $C \leftarrow$ Trimming($G, v, c$)                    ▷ Data reduction
    $\mathcal{C}' \leftarrow$ Enumeration($G, v, c, C$)         ▷ Enumerates isolated cliques
    $\mathcal{C} \leftarrow \mathcal{C} \cup$ Screening($G, v, c, \mathcal{C}', C$)   ▷ Removes non-maximal cliques
return $\mathcal{C}$
```

Fig. 2. The main procedure of the isolated clique enumeration algorithms.

isolated cliques of a graph by enumerating all isolated cliques with pivot v for all $v \in V$. The enumeration of maximal avg-isolated cliques with pivot v for some $v \in V$ is called the *pivot procedure*. It comprises three successive stages (see also Figure 2):

2.2.1 Trimming Stage.

In this stage we perform a data reduction. That is, for a pivot v we build a candidate vertex set $C \subseteq N[v]$ that contains all isolated cliques with pivot v by removing those vertices $u \in N(v)$ from C that cannot belong to an isolated clique with pivot v. Note that by the pivot definition we can always exclude the vertices from $N_-(v)$. Hence, we initially set $C = N_+[v]$. The actual data reduction rules differ depending on the corresponding isolation concept. However, for this stage it is irrelevant whether we want to enumerate isolated maximal or maximal isolated cliques. Therefore, there are three different algorithms (outlined in Figure 3). More details are described in the following paragraphs:

Min-Isolation. We remove vertices that have too few neighbors in the candidate set. Including these vertices would result in a clique in which v has at least c outgoing edges. Therefore, we check for each $u \in C$ whether the following condition—whose necessity can be easily seen—holds (see Figure 3).

Min-(1): u has at least $|C| - c$ adjacent vertices in C.

Avg-Isolation. We first remove vertices with very high degree, since including them results in a violation of the isolation condition. This exclusion enables us to achieve a linear worst-case running time. Furthermore, we remove vertices that have too few neighbors in the candidate set. The following conditions are checked, for details on the correctness and running time we refer to the work of Ito and Iwama [13].

```
Trimming($G, v, c$)
$C \leftarrow N_+[v]$
$c' \leftarrow c - |N[v] \setminus C| - 1$        ▷ Number of vertices that can still be deleted
while $c' \geq 0$:
  for each $u \in C$:                  ▷ Removal of high degree vertices
    Min-isolation:               Avg-isolation:                      Max-isolation:
                                 if $u$ violates Avg-(1):            if $u$ violates Max-(1):
                                     $C \leftarrow C \setminus \{v\}$                    $C \leftarrow C \setminus \{v\}$
                                     $c' \leftarrow c' - 1$                        $c' \leftarrow c' - 1$
  for each $u \in C$:
    Min-isolation:               Avg-isolation:                      Max-isolation:
    if $u$ violates Min-(1):     if $u$ violates Avg-(2),(3),(4):    if $u$ violates Max-(2):
      $C \leftarrow C \setminus \{v\}$              $C \leftarrow C \setminus \{v\}$                    $C \leftarrow C \setminus \{v\}$
      $c' \leftarrow c' - 1$                  $c' \leftarrow c' - 1$                        $c' \leftarrow c' - 1$
if $c' \geq 0$ then return $C$ else return $\emptyset$
```

Fig. 3. Pseudo-code of the trimming stage of the pivot procedure. The removal of high-degree vertices can only be performed for avg- and max-isolation. In order to achieve linear running-time for these two concepts, all high-degree vertices have to be removed before scanning the adjacency lists of the vertices.

Avg-(1): $\deg(u) < (c+1) \cdot |C| - 1$.
Avg-(2): u has fewer than $c \cdot |C|$ outgoing edges.
Avg-(3): u has at least $|C| - c$ adjacent vertices in C.
Avg-(4): C^u has less than $c \cdot (c+1) \cdot |C^u|$ outgoing edges, where $C^u := \{x \in C \mid x < u\} \cup \{u\}$.

Max-Isolation. As for avg-isolation, we remove vertices that have a very high degree compared to v. However, because of the stronger isolation condition, the degree of the vertices is even more restricted compared to avg-isolation. For details on the correctness of the following conditions we refer to [16].

Max-(1): $\deg(u) < |C| + c - 1$.
Max-(2): u has at least $|C| - c$ adjacent vertices in C.

For all three isolation concepts, we return the candidate set C if we have removed at most $c-1$ vertices during the trimming stage. Otherwise, there is no isolated clique with pivot c and we thus return the empty set.

2.2.2 Enumeration Stage.

This stage enumerates the isolated cliques with pivot v. The pseudo-code of this stage is shown in Figure 4. Clearly, the candidate set C is a superset of

```
Enumeration($G, v, c, C$)
$G' \leftarrow \overline{G[C]}$
$c' \leftarrow c - |N[v] \setminus C| - 1$          ▷ Number of vertices that can still be deleted
$\mathcal{S} \leftarrow$ MinimalVertexCovers($G', c'$)
for each $S \in \mathcal{S}$:
  Isolated Maximal:
    if $C \setminus S$ is isolated:
      $\mathcal{C} \leftarrow \mathcal{C} \cup \{C \setminus S\}$          ▷ $C \setminus S$ is maximal clique in $G[C]$
  Maximal Isolated:
    Avg-isolation:
    $\mathcal{C} \leftarrow \mathcal{C} \cup$ IsoSubsets($G, C \setminus S, c$)
    Max-isolation:
    $C' \leftarrow C \setminus S$
      while $C'$ is not max-$c$-isolated
        $u \leftarrow$ max-degree vertex of $C$
        $C' \leftarrow C' \setminus \{u\}$
        if $|C \setminus C'| \leq c'$ : $\mathcal{C} \leftarrow \mathcal{C} \cup \{C'\}$
return $\mathcal{C}$
```

Fig. 4. Pseudo-code of the enumeration stage of the pivot procedure. Input of the enumeration stage is the pivot v, the isolation factor c and the candidate set C.

all isolated cliques with pivot v. We enumerate the cliques by enumerating minimal vertex covers in the complement graph $\overline{G[C]}$. This is done because there are efficient algorithms for the enumeration of small minimal vertex covers. From each minimal vertex cover we obtain a clique that is maximal in $G[C]$. By definition, the pivot has less than c outgoing edges from any isolated clique. Therefore, the size of the enumerated minimal vertex covers can be at most $c' := c-1-|N[v]\setminus C|$, where $N[v]\setminus C$ is the set of vertices that were already removed during the trimming stage. If an enumerated clique is isolated, then we add it to the set of cliques that is checked in the screening stage. If it is not isolated, the actual algorithm depends on the isolation concept and enumeration task:

Isolated maximal cliques. We can discard non-isolated cliques, since we are only interested in maximal cliques. Hence, no subsets of the enumerated cliques have to be considered.

Maximal isolated cliques. We have to consider subsets of the enumerated non-isolated cliques. For max-isolation the situation is easy—we have to remove any vertex that has at least c outgoing edges from C. Either we end up with an isolated clique, or we have removed too many vertices from C and can thus discard C altogether. For avg-isolation, however, it is not clear, which vertex has to be removed, but it is possible to show that only subsets of the set of vertices with the c highest degrees in C [16] may be removed. This is

```
SCREENING($G, v, c, \mathcal{C}', C$)
for each $C' \in \mathcal{C}'$:
  Isolated Maximal:
    Min-isolation:
    for each $u \in N_-(v)$ :
      if $C' \subseteq N(u)$ : $\mathcal{C}' \leftarrow \mathcal{C}' \setminus \{C'\}$
    Max/Avg-isolation:
    for each $u \in N[v] \setminus C'$ :
      if $C' \subseteq N(u)$ : $\mathcal{C}' \leftarrow \mathcal{C}' \setminus \{C'\}$
  Maximal Isolated:
  for each $u \in N_-(v)$ :
    if $C' \subseteq N(u)$ : $\mathcal{C}' \leftarrow \mathcal{C}' \setminus \{C'\}$
    else:
    Avg-isolation:
    $D \leftarrow (C \setminus C') \cap N_\cap(C')$
    $\mathcal{D} \leftarrow$ MAXIMALCLIQUES($G[D]$)
      for each $D \in \mathcal{D}$ :
        if $D \cup C'$ has isolated subset:
          $\mathcal{C}' \leftarrow \mathcal{C}' \setminus \{C'\}$
    Max-isolation:
    $D \leftarrow (C \setminus C') \cap N_\cap(C')$
    for each $i \leq c - 1$:
      $D_i \leftarrow \{w \in D \mid \deg(w) \leq |C'| + c + i\}$
      $D' \leftarrow$ MAXIMUMCLIQUE($D_i$)
        if $|D'| \geq i + 1$ :
          $\mathcal{C}' \leftarrow \mathcal{C}' \setminus \{C'\}$
return $\mathcal{C}'$
```

Fig. 5. Pseudo-code of the screening stage of the pivot procedure. Input of the screening stage is the pivot v, the isolation factor c, the set $\mathcal{C}'$ of cliques enumerated in the enumeration stage, and the candidate set C.

performed in the ISOSUBSETS procedure, the details of this procedure can be found in [16].

2.2.3 Screening Stage.

In this stage we remove non-maximal cliques from the set of cliques that were enumerated during the enumeration stage. Basically, we check whether there are vertices in the common neighborhood $N_\cap(C') := (\bigcap_{u \in C'} N(u)) \setminus C'$ of an enumerated clique C' such that adding these vertices yields an isolated clique. The actual maximality tests depend on the isolation concept and enumeration task. The algorithm of the screening stage is shown in Figure 5.

Isolated maximal. We have to check whether an enumerated clique C' is a maximal clique in G. For min-isolation, we only need to consider vertices in $N_-(v)$ since the enumerated cliques are maximal cliques in $G[N_+[v]]$. For each $u \in N_-(v)$ we thus check whether $C' \subseteq N(u)$, and if so, then we discard the clique C'. For max-isolation and avg-isolation, we have to consider all vertices in $N[v] \setminus C'$ because we have removed high-degree vertices prior to the vertex cover enumeration. However, one of these vertices might be adjacent to all vertices in C'. Therefore, we check for each of the at most c deleted vertices from $N[v] \setminus C$ whether it is adjacent to all vertices in C'.[4]

Maximal Isolated. Here, the maximality check is more complicated. First, we check whether there is a vertex u that was removed during the trimming stage such that there is a superset of C' that is an isolated clique and contains u. Clearly, we cannot include any vertex $u \in N_+[v] \setminus C$, where C is the candidate set after the trimming stage: these vertices have been removed since their inclusion in a clique with pivot v always leads to a violation of the isolation condition. Therefore, we only need to consider vertices in $N_-(v)$ for this maximality test.

For maximal isolated cliques we need to perform a second maximality test. This is necessary, because in the enumeration stage it can happen that we enumerate two isolated cliques C' and C'' such that $C' \subsetneq C''$ (because we remove vertices from maximal cliques in order to establish the isolation condition). The corresponding maximality tests differ for avg-isolation and max-isolation. In the following, we briefly describe the idea behind these tests.

*Avg-isolation.*We are looking for subsets D of $C \setminus C'$ such that adding D to C' yields an avg-isolated clique. This subset must be a clique and its vertices must be adjacent to all vertices of C'. We can find such a set by enumerating all maximal cliques in $(C \setminus C') \cap N(C')$, and checking for each enumerated clique D whether $D \cup C'$ has an isolated subset that is superset of C'. This can be performed by removing the vertex of highest degree from $D \cup C'$ as long as the isolation condition is violated.

*Max-isolation.*We are looking for subsets D of $C \setminus C'$ such that adding D to C' yields a max-isolated clique. This subset must be a clique and its vertices must be adjacent to all vertices of C'. Suppose that the maximum degree of any vertex in D is $|C'| + c + i$, with $i \leq c - 1$ (otherwise the isolation condition would be violated). Then, we know that D must have size at least $i + 1$. Otherwise, we would also violate the isolation condition. Hence, we check—for each possible i—whether the subset of $(C \setminus C') \cap N(C')$ that contains the vertices with degree at most $|C'| + c + i$ contains a clique of size at least $i + 1$.

Finally, we output all cliques that have passed the respective maximality tests. For min-isolation and the enumeration of maximal avg-isolated and maximal max-isolated cliques the presented algorithms have been previously described [16]. For the enumeration of isolated *maximal* cliques, the described algorithms are very similar to the algorithm of Ito and Iwama [13]. The only

[4] Note that this test is not included in the original algorithm by Ito et al. [14]. However, it is necessary for the stated reasons. In Theorem 1 we will show that this test can also be performed in linear overall running time.

	Isolated Maximal	Maximal Isolated
Min-Isolation	$O(2^c \cdot c \cdot n + m \cdot n)$ [16]	
Avg-Isolation	$O(2^c \cdot c^5 \cdot m)$	$O(2.89^c \cdot c^2 \cdot m)$ [16]
Max-Isolation	$O(2^c \cdot c^5 \cdot m)$	$O(2.44^c \cdot c^2 \cdot m)$ [16]

Table 1
Running times of the enumeration algorithms for the three isolation concepts and the two different enumeration tasks.

difference lies in the maximality test of our screening stage. It is required to exclude all non-maximal cliques, and—compared to the test of Ito and Iwama—it even helps in improving the worst case running time from $O(4^c \cdot c^4 \cdot m)$ to $O(2^c \cdot c^5 \cdot m)$.

Theorem 1 *All avg-c-isolated (max-c-isolated) maximal cliques can be enumerated in $O(2^c \cdot c^5 \cdot m)$ time.*

PROOF. Ito and Iwama showed that the overall running time of the trimming and enumeration stages of all calls to the pivot procedure is $O(2^c \cdot c^3 \cdot m)$ [13]. Furthermore, they proved that the sum of edges that enter the enumeration stage of the algorithm is $O(c^3 \cdot m)$ [13, Lemma 3.13]. We will use this to upper-bound the running time of our screening stage. For each enumerated clique C, we have to test whether there is a vertex $c \in N[v] \setminus C$ that is adjacent to all vertices of C. This can be done by scanning the adjacency lists of the vertices in C. For avg-isolation and max-isolation, we can furthermore show that at most $2^c \cdot c$ cliques are enumerated during one execution of the enumeration stage [16]. Therefore, the overall running time for this maximality test is

$$\sum_{v \in V} O(2^c \cdot c \cdot c) \sum_{u \in C} \deg(u) = O(2^c \cdot c^2 \cdot c^3 \cdot m). \qquad \square$$

Table 1 gives an overview of the theoretical worst-case running times of all enumeration algorithms. For min-isolation we could not achieve a linear running time for fixed c since we cannot exclude high-degree vertices during the trimming stage. For avg-isolation and max-isolation we have linear running times for fixed c and both enumeration tasks. However, for the enumeration of maximal isolated cliques, we have an inferior worst-case running time (with respect to the isolation factor c) due to the fact that we have to perform a more involved maximality test during the screening stage.

3 Implementation Issues

We briefly describe some notable differences between the theoretical algorithms from Section 2.2 and their actual implementations.[5] For avg-isolation and max-isolation, they apply only to the (more complex) algorithms for maximal isolated cliques in contrast to isolated maximal cliques.

Min-isolation. In the trimming stage, we remove vertices that have lower index than the pivot (this differs from the description in [16]). This does not help in achieving a better worst-case running time, but it speeds up the trimming stage and prevents the algorithm from needlessly entering the enumeration stage for vertices with at least c neighbors of lower index. In many instances this provided a speed-up of factor 3 or more.

Avg-isolation. Since our experiments showed that the enumeration of avg-isolated subsets of non-avg-isolated cliques was a bottleneck, we introduced an additional test: We check whether we can obtain an avg-isolated set by gradually removing the vertices of highest degree. If this is not the case, then no subset of the clique is avg-isolated. Thus, we can avoid unnecessarily enumerating subsets of non-avg-isolated cliques. Furthermore, we perform this test also before entering the enumeration stage, and only enter it when the enumerated cliques have a chance of being c-isolated. Both tests provided a speed-up of approximately two orders of magnitude in our experiments.

Max-isolation. The worst-case running time of $O(2.44^c \cdot c^2 \cdot m)$ can be shown using a maximum clique algorithm in the screening stage (for details see [16]). Running time analysis showed that, unexpectedly, in practice the screening stage was not the bottleneck of the enumeration algorithm. Therefore, in our implementation we instead enumerate all cliques in the set of deleted vertices to check whether an enumerated clique is maximal. This was sufficiently fast, while keeping the implementation simpler.

As maximal clique enumeration algorithm (required for the screening stage of avg-isolation and max-isolation), we used an improved variant of the standard Bron-Kerbosch algorithm [4, 15, 6]. This algorithm was not a bottleneck, in particular because of its good output-sensitivity (that is, it runs quickly if there are only few maximal cliques); it has also recently been shown to have optimal worst-case performance [21, 6]. We also use this algorithm as a comparison point for the running times of our clique enumeration algorithms.

[5] The program is written in Objective Caml and consists of about 1600 lines of code. It is free software and available from `http://theinf1.informatik.uni-jena.de/c-isol/`

4 Experimental Results

Our investigations concentrate on random feature graphs that were created according to the $G_{n,m,p}$ model and on financial networks. All experiments were run on an AMD Athlon 64 3700+ machine with 2.2 GHz, 1 M L2 cache, and 3 GB main memory running under the Debian GNU/Linux 4.0 operating system with the Objective Caml 3.09.2 compiler. Note that for some instances the enumeration did not terminate because the program exceeded the memory limit of 3 GB or the corresponding run timed out (after half an hour). This causes some missing data points in the diagrams. In the diagrams, we use "min-isolation" to denote the enumeration of "maximal min-isolated cliques", "max-isolation" stands for "maximal max-isolated cliques", "avg-isolation" for "maximal avg-isolated cliques", "maxm-isolation" for "max-isolated maximal cliques", "avgm-isolation" for "avg-isolated maximal cliques", and finally "bk" stands for the enumeration of all maximal cliques using the Bron-Kerbosch algorithm.

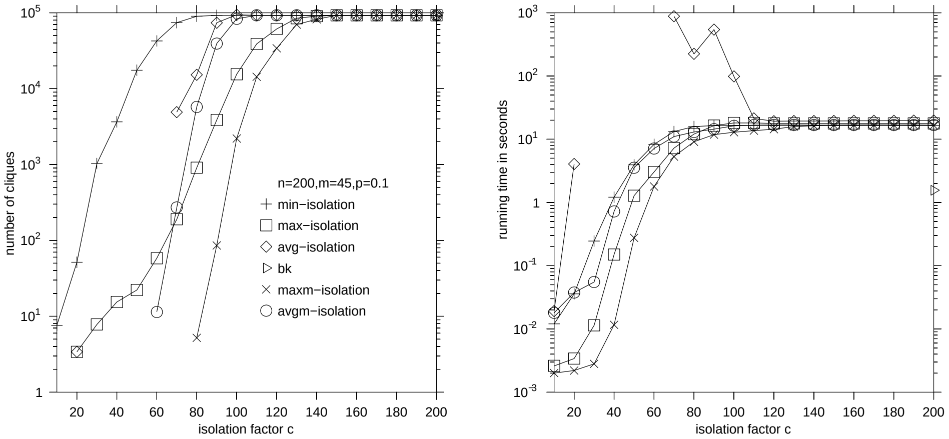
4.1 Synthetic Data

We generated random graphs using the $G_{n,m,p}$ model (see Behrisch and Taraz [1] and references therein). The underlying model is that cliques are defined by *features*. More precisely, each of n vertices draws each of m features with probability p, and two vertices are connected by an edge iff they have at least one feature in common (note that here m does not denote the number of edges as elsewhere). These graphs contain very many maximal cliques, and are tough inputs for clique enumeration.

Note that our diagrams contain the data corresponding to all isolation concepts and enumeration tasks. However, to better distinguish between the task of enumerating all *maximal isolated* cliques and the task of enumerating all *isolated maximal* cliques, we first report our findings corresponding to maximal isolated cliques, then report our findings concerning isolated maximal cliques, and finally we compare the respective results.

4.1.1 Maximal Isolated Cliques

Our main finding is that enumerating min- and max-isolated cliques is feasible over a by far wider parameter range than enumerating general maximal cliques or avg-isolated cliques, and that the isolation concepts can help keeping the number of enumerated *isolated* cliques in check even in graphs that contain excessively many *maximal* cliques. Furthermore, we observe a difference in output-sensitivity. Whereas min-isolation seems to be output-sensitive



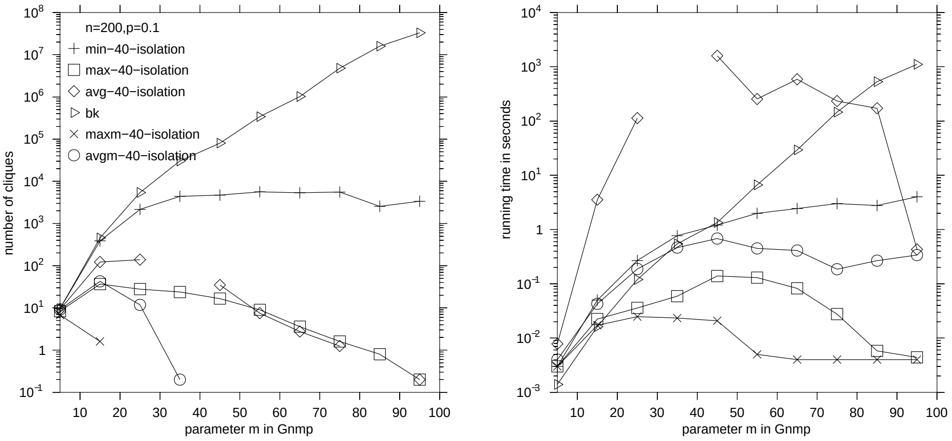
(a) (b)

Fig. 6. $G_{n,m,p}$ model with $n = 200$, $m = 45$, and $p = 0.1$. The average running time for Bron-Kerbosch is 5.06 seconds.

in general and max-isolation in most instances, avg-isolation had high running times sometimes even for relatively few enumerated cliques. Starting from a base setting with $c = 40$, $n = 200$, $m = 45$, and $p = 0.1$, we examined the effect of varying parameters. For each parameter setting we created five random instances and measured the average running time as well as the average number of enumerated cliques.

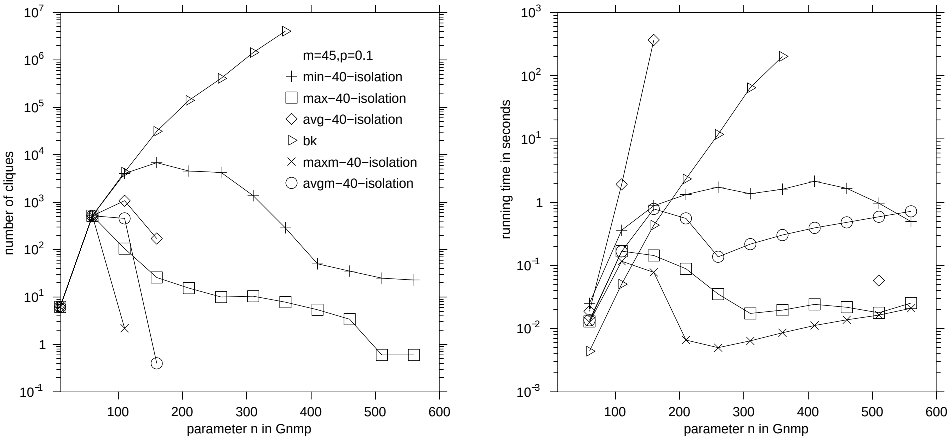
Figure 6a shows the number of cliques output for varying c. The average number of maximal cliques is about 93000. Starting from $c \approx 80$, all maximal cliques are enumerated using min-isolation. For avg- and max-isolation all maximal cliques are found with $c \approx 150$. In Figure 6b, we see that the running time of the min- and max-isolation concepts closely follows the number of cliques output, that is, the algorithms are output-sensitive. This can not be observed for avg-isolation because of its running time peaks for intermediary values of c. Notably, for all three isolation concepts almost all time is spent in the enumeration stage. Therefore, the increased running time and lack of output-sensitivity for avg-isolation stems from the enumeration of isolated subsets of non-avg-isolated cliques, since this is where the enumeration stages differ. Furthermore, this means that in practice the screening stage, which dominates the overall worst-case running time, is not the bottleneck of the algorithm. Compared to the Bron-Kerbosch algorithm, which enumerates the *whole* set of maximal cliques, all three algorithms are about ten times slower, but min- and max-isolation are significantly faster when the output is restricted by a small c (see Figure 6b).

We next examine variation of the feature number m (Figure 7). More features lead to an exponential growth of the number of maximal cliques (Figure 7a). This growth only wears off when the graph becomes very dense ($m = 85$, about 57 % of all possible edges present). In contrast, the number of min-40-



(a) (b)

Fig. 7. $G_{n,m,p}$ model with $c = 40$, $n = 200$, and $p = 0.1$. The missing point for avg-isolation is due to the memory limit of the test runs (3 GB).

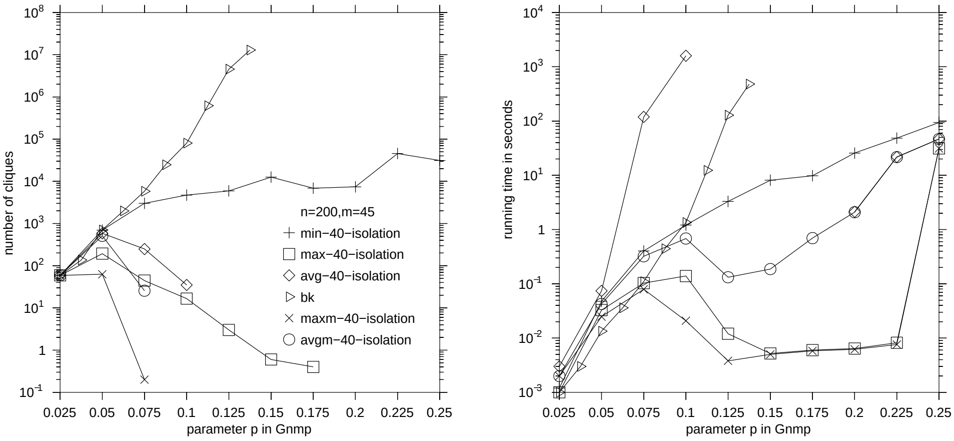


(a) (b)

Fig. 8. $G_{n,m,p}$ model with $c = 40$, $m = 45$, and $p = 0.1$.

isolated cliques reaches a plateau, and for the more stringent criteria, we even notice a drop-off already for $m \geq 30$. While for the Bron-Kerbosch algorithm and min-isolation, we have running times mostly following the number of generated cliques, for max- and avg-isolation, we have a maximum for $m = 35$ and $m = 45$, respectively. Again, almost all time is spent in the enumeration stage.

Next, we consider varying n (Figure 8). Here, enumerating avg-isolated cliques becomes infeasible for $n \approx 150$, and the Bron-Kerbosch algorithm for $n \approx 350$. In contrast, running times for min- and max-isolation stay within a few seconds. For max-isolation with high n, we get very few isolated cliques. This is because, e.g., for $n = 500$, the expected size of a feature clique is 50, and thus a vertex with a feature that is not part of a clique already produces an



(a) (b)

Fig. 9. $G_{n,m,p}$ model with $c = 40$, $n = 200$, and $m = 45$.

expected number of 49 outgoing edges.

Finally, we consider varying p (Figure 9). For very small p, the feature cliques are disjoint, and thus isolated. With growing p, we get overlap between the feature cliques, and thus an exponential growth with respect to the number of maximal cliques. Again, we observe a much reduced growth in the number of min-isolated cliques, and a drop-off in the number of avg- and max-isolated cliques, which can be explained by the increased interconnectedness of feature cliques, which makes isolation less likely. Again, min- and max-isolated cliques could be enumerated over a wider range of parameter values than avg-isolated and maximal cliques. In particular, the algorithms for enumerating min- and max-isolated cliques were output-sensitive while this was not the case for avg-isolation.

4.1.2 Isolated Maximal Cliques

Our main finding is that enumerating max- and avg-isolated maximal cliques is feasible over a wide parameter range. Since this enumeration type is more stringent, it generally also leads to less enumerated cliques. A clear difference to enumerating maximal isolated cliques is that actually there are very few maximal cliques that are also max-isolated or avg-isolated, respectively—for wide parameter ranges there are no cliques enumerated at all. In the following, we again study the effect of varying parameters, analogously to Section 4.1.1.

First, we consider the variation of c (Figure 6). An important observation is that there are no isolated maximal cliques for isolation factors below $c \approx 50$. Starting from $c \approx 110$, all maximal cliques are enumerated using avg-isolation. For max-isolation, this is the case above $c \approx 150$. Note that for low isolation factors there are less avg-isolated maximal cliques than maximal max-isolated cliques, but above $c \approx 70$ we have more avg-isolated maximal cliques than maximal max-isolated cliques. For both max- and avg-isolation, most of the time is spent in the enumeration stage. Notably, even for low isolation factors with (almost) no isolated maximal cliques, the enumeration already needs a considerable amount of time, and for isolation factors which yield more cliques, the running time is already almost as high as the running time needed for enumerating all maximal cliques. In this sense, enumerating max- and avg-isolated maximal cliques is not output-sensitive, but the running times are comparable with the running times of enumerating maximal min- and max-isolated cliques.

Next, we consider the variation of the feature number m (Figure 7). The plateau and the drop-off described in Section 4.1.1 can be observed already for very low values of m for avg-isolation, and for the even more restrictive max-isolation, we observe an immediate drop-off. For max- and avg-isolation, the running time is always below one second. However, for avg-isolation we do not have output sensitivity—the maximum running time is observed when no cliques are output. For avg-isolation and max-isolation the maximum running time is around $m = 45$ and $m = 25$, respectively.

When varying n (Figure 8), the number of avg-isolated and max-isolated cliques reaches a plateau for $n \approx 50$, but drops off quickly for increasing n. Above $n \approx 150$, there are no more max- and avg-isolated maximal cliques, for the same reason as for the other enumeration type. For both max- and avg-isolated, the running time stays below one second.

Finally, we vary p (Figure 9). Again, we observe reduced growth of the number of max-isolated and avg-isolated maximal cliques, followed by a clear drop-off. Above $p \approx 0.1$ there are no isolated maximal cliques left for both max- and avg-isolation. The reason is again the increased interconnectedness of the feature cliques. The running times for both max- and avg-isolation stay within a few seconds.

4.1.3 Comparison

In the following, we compare the theoretical running times (see Table 1) with the running times we observed in practice on the $G_{n,m,p}$ graphs. For this, we interpolate for each isolation concept and enumeration type the running time function based on the data used for the diagram in Figure 6b, compute the basis of the exponential function, and compare it with the theoretical basis. In Table 2 we present the corresponding results. All the enumeration algorithms seem to be faster in practice, and generally the faster concepts in theory are also faster in practice. An exception are the max-isolated maximal cliques, which perform similarly as the maximal max-isolated cliques in

<table>
<tr><th></th><th>Isolated Maximal</th><th>Maximal Isolated</th></tr>
<tr><td>Min-Isolation</td><td colspan="2">2^c / 1.16^c</td></tr>
<tr><td>Avg-Isolation</td><td>2^c / 1.23^c</td><td>2.89^c / 1.70^c</td></tr>
<tr><td>Max-Isolation</td><td>2^c / 1.28^c</td><td>2.44^c / 1.27^c</td></tr>
</table>

Table 2
Exponential part of the running times of the enumeration algorithms for the three isolation concepts and the two different enumeration types. The first entry corresponds to the theoretical worst-case result (as in Table 1), and the second entry to the hypothesized exponential growth of the running times in practice. These practical running times are determined by the maximum gradient of the corresponding running time function in logarithmic scale.

practice, but which should be faster when comparing the theoretical running times. Min-isolation shows the best exponential running time behavior in practice, although it is not the fastest concept when looking at the total running times. As expected, enumerating maximal avg-isolated cliques shows the worst running time behavior.

In our comparison, min-isolation turns out to be the concept with the lowest exponentially growing running time. For avg-isolation, comparing the concepts of “maximal isolated” and “isolated maximal”, the former is clearly the faster one in practice, while the latter is typically too slow for many applications. For max-isolation, there is no significant difference between “maximal isolated” and “isolated maximal”, thus in practice either of the two could be chosen.

4.2 Financial Networks

Many investigations concerning financial network analysis are based on market graphs (see, e.g., [17]). We generated market graphs from publicly available stock data.[6] A market graph is constructed as follows. Financial instruments (e.g., stocks or indices) are represented by vertices. For each pair of vertices u, v there is an edge connecting them if the corresponding correlation coefficient C_{uv} based on the price fluctuations of u and v in some prespecified time range exceeds some prespecified threshold θ, where $-1 \leq \theta \leq 1$. Informally speaking, two instruments u and v have a positive correlation coefficient C_{uv} if they show similar daily fluctuations in the prespecified time range, and they have a negative correlation coefficient if their daily fluctuations behave oppositional. Details about the construction of market graphs can be found, e.g., in [2].

Experimental Setup. We considered various market graphs based on the

[6] We used the data from `finance.yahoo.com`.

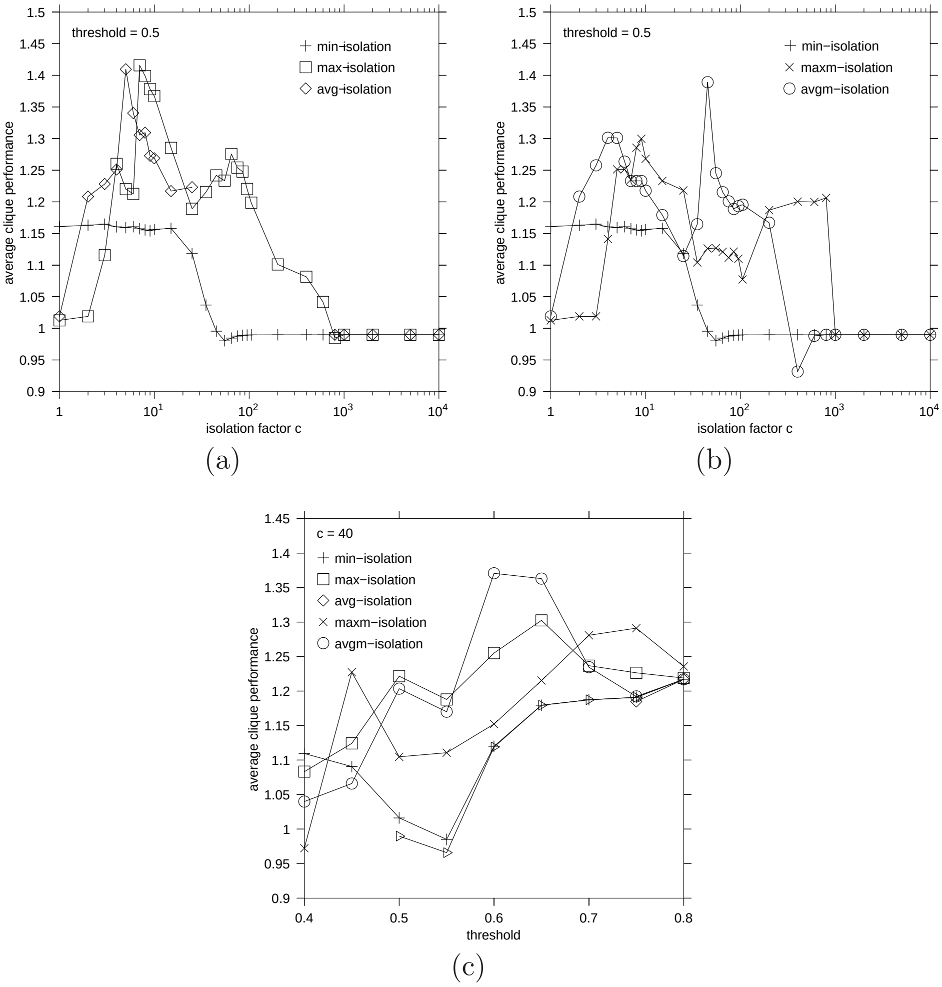
daily fluctuations of several thousands of financial instruments during 500 consecutive trading days. Basic properties of such graphs, like degree distribution, edge density, clustering coefficient, maximum clique size, and maximum independent set size have been analyzed by Boginski et al. [2, 3].

The following diagrams rely on data from 2204 financial instruments beginning at 12/02/2003 over 500 consecutive trading days. However, the experiments were also executed on many other graphs (based on data from other start dates and other threshold values) for which the following observations also hold true (in the qualitative sense). We excluded trivial cliques from the output, that is, cliques containing only one or two vertices, but the results also hold if the output contains trivial cliques. Note that the graphs do not include financial instruments whose values get below one dollar in the considered time period, since such "penny stocks" often show strong daily fluctuations, which are additionally biased by the rounding of the available data. In the experiments with fixed threshold, the threshold is set to $\theta = 0.5$ as proposed by Boginski et al. [3] in order to ensure that only significantly correlated stocks are adjacent. Moreover, our experiments showed that for $\theta = 0.5$ there is a good balance between the number of isolated cliques in the graph and the edge density (for low threshold levels, the graph gets too dense to contain many isolated cliques, and for high threshold levels, the graph gets too sparse to contain interesting cliques of significant size). For threshold $\theta = 0.5$, the graph contains 2204 vertices and 64376 edges and approximately 70000 maximal cliques.

Boginski et al. [2, 3] suggested the use of clique analysis for classifying stocks, based on the property that cliques represent sets of "similar" financial instruments. However, they do not provide any method to find cliques of good quality. Therefore, we measured the average performance of the enumerated cliques. The *average price* of a financial instrument at some given trading day t is the mean price of the instrument at day t and the 10 trading days before and after t. Average prices are used to balance stronger daily fluctuations of financial instruments. The *performance* in the time interval $[t_1, t_2]$ ($t_1 < t_2$) of a financial instrument is the average price at day t_2 divided by the average price at day t_1. The performance of a clique is the mean performance of its vertices. The *average performance* of a set of cliques is the mean performance of the cliques. We always measure the performance in the time period the market graph is based on. We first study our enumeration concept of enumerating all maximal isolated cliques, then report our findings concerning isolated maximal cliques, and finally we compare the respective results.

4.2.1 Maximal Isolated Cliques

Basic Results. As for the $G_{n,m,p}$ graphs, we found enumerating min- and max-isolated cliques to be feasible over a wide range of parameters, while the



(a) (b)

(c)

Fig. 10. Average clique performance in a market graph based on 500 consecutive trading days beginning at 12/02/2003. Note that the performance of the NASDAQ in the considered time period is 1.01.

Bron-Kerbosch algorithm and the avg-isolation algorithm are sometimes too slow. For all three isolation concepts and for $c \leq 10$ the running time is around a second. For intermediate isolation factors we observe a peak in the running time of max- and avg-isolation.

The number of enumerated isolated cliques ranges from a few hundred for very low isolation factors up to all maximal cliques (≈ 70000) for high isolation factors, where there are generally much more min-isolated cliques than max- and avg-isolated cliques (up to one order of magnitude). For low isolation factors, max- and avg-isolated cliques have size at most 10, whereas there are already min-1-isolated cliques of size ≈ 50. For high isolation factors, the enumerated cliques have maximum size ≈ 80.

Clique Performance. We can observe (Figure 10a) that the performance of the enumerated min-, max-, and avg-isolated cliques is better for lower to intermediate isolation factors and generally exceeds the performance of all maximal cliques. For higher isolation factors, the min-isolated cliques show a performance which is similar to the average performance of all vertices in the graph. Most notably, max-isolated cliques have especially high performance for intermediate isolation levels; we can observe a peak of the performance for max-isolation around $c = 10$. Avg-isolation seems to perform similarly as max-isolation, but we usually observe running time or memory consumption problems for intermediate isolation levels. For very high isolation factors, all three isolation concepts generate all maximal cliques and therefore obviously yield the same average performance. In general, the described effects depend on the underlying graph and the performance of the overall market and are more or less pronounced. Note that in our example (Figure 10a), max- and avg-isolation perform worse than min-isolation for very low isolation factors, however, this was not the case in other graphs (based on other time periods). The average performance of all financial instruments in the considered time period is approximately 1.19. Surprisingly, the maximal cliques have an average performance of about 0.99. This is caused by financial instruments with a particularly bad performance that are included in many maximal cliques, but not in *isolated* cliques.

When varying the threshold value θ, Figure 10c shows that the enumerated cliques perform generally better for higher threshold levels. The performance of the min-isolated cliques is comparable to the performance of all maximal cliques for the chosen isolation $c = 40$, whereas max-isolation performs better in general. Note that this only holds true for low isolation factors $c \leq 100$, since for higher isolation factors the performance of all three isolation concepts gets closer to the performance of all maximal cliques.

4.2.2 Isolated Maximal Cliques

Basic Results. All max- and avg-isolated cliques can be enumerated over a wide range of parameters, where the running times are always below the running time for enumerating min-isolated cliques. The peak in the running time, as it was observed for enumerating all maximal max-isolated cliques, cannot be observed. Among all isolation concepts and enumeration tasks, enumerating max-isolated maximal cliques has the best running time but also the fewest cliques. In contrast to the $G_{n,m,p}$ graphs, the running time closely follows the number of enumerated isolated maximal cliques for both max- and avg-isolation, thus the algorithms are output-sensitive for the finance graph.

Clique Performance. We can observe in Figure 10b that the performance is also generally better for lower to intermediate isolation factors, and it also exceeds the performance of all maximal cliques. When varying the threshold value, Figure 10c shows that among all isolation concepts and enumeration types, the avg-isolated maximal cliques seem to perform best for intermediate threshold levels, and max-isolated maximal cliques perform slightly worse than maximal max-isolated cliques. Generally, the performance of the enumerated cliques increases with increasing threshold. Compared to all maximal cliques, all sorts of isolated cliques (with respect to all isolation concepts and enumeration types) perform better. For very low threshold levels the performance of the min-isolated cliques becomes better than the performance of other isolation concepts. Note that our algorithm could not enumerate all maximal cliques for low threshold levels in reasonable time, because the graph becomes too dense.

Summarizing, when using a low isolation factor, then min-isolation seems to be a good choice in order to get cliques with a good performance, whereas for intermediate isolation factors it seems that the other isolation concepts and enumeration types seem to be better. There appears to be no big difference between the two enumeration types—both yielding better performing cliques compared to all maximal cliques for intermediate isolation factors; however, maximal isolation yields significantly more cliques with good performance.

4.2.3 Possible Applications

We observed that isolated cliques have interesting properties compared to general maximal cliques. For example, looking more closely at the cliques responsible for the peaks of the performance for intermediate isolation levels (for max- and avg-isolation and both enumeration types), we observe that these cliques represent some niche in the market. For instance, in Figure 10a and Figure 10b the peak is caused by American raw material, oil, and energy stocks, and by related industries like transportation, pipeline construction, and refineries. This peak is less pronounced in graphs based on earlier time periods (that is, beginning before 12/02/2003) and becomes even more pronounced for graphs based on later time periods (that is, beginning after 12/02/2003). This indicates that isolation can be useful to detect market trends. Finally, isolated cliques performed better than general maximal cliques. Hence, we can employ isolation to filter out financial instruments with bad performance when enumerating cliques. This could provide a new alternative for investors to classify financial instruments (using clique analysis as proposed by Boginski et al. [2]). Here, a more thorough and detailed study is necessary, cooperating with financial experts.

4.3 Music Artist Similarity Network

Last.fm is a music community website with over 20 million active users. Based on user statistics, Last.fm is able to calculate a similarity score for any two artists. A network is obtained by applying a threshold value for the similarity score. The resulting network with 6797 vertices and 108 314 edges is an interesting test case, since we would expect features such as musical genres, groups of artists, and geographical or temporal proximity to induce isolated cliques, and it is interesting to see whether our algorithms can retrieve them.

4.3.1 Maximal Isolated Cliques

We tried to find the isolation factor that would yield a manageable number of cliques: with min-1-isolation, we obtain 215 cliques, with avg-5-isolation 180 cliques, and with max-8-isolation 204 cliques.

All generated cliques look reasonable in the sense that not only do they constitute clearly similar artists, but usually it is also possible to label them by a combination of genre, time, and location. For example, the largest min-1-isolated clique contains 14 current European and Commonwealth Drum 'n' Bass artists. For min-isolation, not all cliques are this specific: for example, the clique containing Mick Jagger, Joe Cocker, Dire Straits, Eric Clapton, Bruce Springsteen, Aerosmith, Rod Stewart, Queen, U2, and The Rolling Stones could probably only be described as "mainstream rock". For max-isolation, the cliques tend to be smaller and more specific, usually homogeneous with respect to all of genre, place, and time, coinciding with the intuition that these cliques are "more isolated". For example, a clique contains 6 contemporary Polish reggae bands. Further, groups of closely connected artists appear such as Mike Patton together with his three projects Tomahawk, Mr. Bungle, and Fantômas, or John Zorn together with his bands Masada and Naked City.

4.3.2 Isolated Maximal Cliques

We found that there are only small differences in the results between maximal isolated cliques and isolated maximal cliques. This could be explained by the structure of the graph: due to the assignment of similarity scores, all vertices have similar degrees. Therefore, the situation that an isolated clique is extensible by a vertex that has high enough degree to destroy the isolation is less frequent.

5 Conclusion and Outlook

Summarizing, our experimental results strongly support the practical relevance of various isolation concepts [13, 16] for enumerating maximal cliques. Indeed, with the exception of maximal avg-isolated cliques, the enumeration of isolated cliques is fast and feasible compared to the enumeration of maximal cliques. Since the isolation concepts significantly restrict the number of maximal cliques to be enumerated, the corresponding algorithms for small degrees of isolation are significantly faster than algorithms enumerating all cliques (such as the famous Bron-Kerbosch algorithm [4]). When comparing "isolated maximal" with the "maximal isolated" enumeration task, it turns out that (not surprisingly) the first is faster than the latter both in theory and in practice. However, the enumeration algorithms for isolated maximal cliques are not output-sensitive, since many isolated cliques are discarded because they are not maximal cliques. Moreover, enumerating maximal isolated cliques is worthwhile, since more cliques are output. When comparing min-isolation versus avg-isolation versus max-isolation, the following general observations have been made. Recall that max-isolation implies avg-isolation implies min-isolation with respect to the enumerated cliques. Notably, avg-isolation makes the biggest problems in achieving output-sensitive algorithms. As a rule, the algorithms for min- and max-isolation turn out to be faster. However, for all isolation concepts problem instances of interesting size could be solved and, hence, it often will depend on the specific application behind which of the isolation concepts (and also which of the two enumeration tasks) is to be preferred from a semantic point of view. This is indicated by our studies with real-world networks from finance and music artist similarity. Min-isolation, being the weakest demand, can be used for the enumeration of some maximal cliques, in case there are too many of them.

As to future work, we see the following challenges. First, trying to further speedup the presented algorithms is a worthwhile task; the strongest demand here concerns the avg-isolation concept. Second, there is no need to restrict isolation concepts only to clique enumeration. In particular, incorporating other (somewhat more relaxed) concepts of dense subgraphs such as pseudo-cliques [13] and s-plexes [16] into further experimental investigations is clearly interesting. Third, based on the duality of the Clique problem with the Independent Set problem, it seems promising to explore whether extending the experiments to the complements of the input graphs provides further insights in certain cases (cf. [2, 3]). Note that complementation makes a dense graph sparse and vice versa. Finally, there is an almost unlimited potential for studying further real-worlds networks (e.g., biological ones) using our clique enumeration tools and giving plausible semantic interpretations for the respective meaning of isolation in the corresponding application context.

Acknowledgement. We thank our student assistants Robert Bredereck and Manuel Sorge for their excellent support in gathering experimental data and performing and analyzing experiments.

References

[1] M. Behrisch and A. Taraz. Efficiently covering complex networks with cliques of similar vertices. *Theoretical Computer Science*, 355(1):37–47, 2006.

[2] V. Boginski, S. Butenko, and P. M. Pardalos. Statistical analysis of financial networks. *Computational Statistics and Data Analysis*, 48(2): 431–443, 2005.

[3] V. Boginski, S. Butenko, and P. M. Pardalos. Mining market data: A network approach. *Computers and Operations Research*, 33(11):3171–3184, 2006.

[4] C. Bron and J. Kerbosch. Finding all cliques of an undirected graph (algorithm 457). *Communications of the ACM*, 16(9):575–576, 1973.

[5] S. Butenko and W. E. Wilhelm. Clique-detection models in computational biochemistry and genomics. *European Journal of Operational Research*, 173(1):1–17, 2006.

[6] F. Cazals and C. Karande. A note on the problem of reporting maximal cliques. *Theoretical Computer Science*, 407(1–3):564–568, 2008.

[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, D. W. Threadgill, K. F. Manly, and R. W. Williams. Complex trait analysis of gene expression uncovers polygenic and pleiotropic networks that modulate nervous system function. *Nature Genetics*, 37(3):233–242, 2005.

[8] R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Springer, 1999.

[9] G. W. Flake, S. Lawrence, and C. L. Giles. Efficient identification of web communities. In *Proceedings of the 6th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '00)*, pages 150–160. ACM Press, 2000.

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

[11] J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182(1):105–142, 1999.

[12] F. Hüffner, C. Komusiewicz, H. Moser, and R. Niedermeier. Enumerating isolated cliques in synthetic and financial networks. In *Proceedings of the 2nd Annual International Conference on Combinatorial Optimization and Applications (COCOA '08)*, volume 5165 of *LNCS*, pages 405–416. Springer, 2008.

[13] H. Ito and K. Iwama. Enumeration of isolated cliques and pseudo-cliques.

ACM Transactions on Algorithms, 2008. To appear.

[14] H. Ito, K. Iwama, and T. Osumi. Linear-time enumeration of isolated cliques. In *Proceedings of the 13th Annual European Symposium on Algorithms (ESA '05)*, volume 3669 of *LNCS*, pages 119–130. Springer, 2005.

[15] I. Koch. Enumerating all connected maximal common subgraphs in two graphs. *Theoretical Computer Science*, 250(1–2):1–30, 2001.

[16] C. Komusiewicz, F. Hüffner, H. Moser, and R. Niedermeier. Isolation concepts for enumerating dense subgraphs. In *Proceedings of the 13th International Computing and Combinatorics Conference (COCOON '07)*, volume 4598 of *LNCS*, pages 140–150. Springer, 2007. Long version to appear in *Theoretical Computer Science*.

[17] R. N. Mantegna and H. E. Stanley. *Introduction to Econophysics: Correlations and Complexity in Finance*. Cambridge University Press, 2000.

[18] J. W. Moon and L. Moser. On cliques in graphs. *Israel Journal of Mathematics*, 3(1):23–28, 1965.

[19] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms.* Oxford University Press, 2006.

[20] H. Saito, M. Toyoda, M. Kitsuregawa, and 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 (AIRWeb '07)*, volume 215 of *ACM International Conference Proceeding Series*, pages 45–48. ACM Press, 2007.

[21] E. Tomita, A. Tanaka, and H. Takahashi. The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science*, 363(1):28–42, 2006.