Enumerating Isolated Cliques in Synthetic and Financial Networks

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

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

Abstract. We do computational studies concerning the enumeration of maximal 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 perform experiments with synthetic graphs (in the $G_{n,m,p}$ model) and financial networks, proposing the enumeration of isolated cliques as a useful instrument in analyzing financial networks.

1 Introduction

We study the generation 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 $G[V'']$ is also complete. Unfortunately, already finding one maximum-cardinality clique is a notoriously hard computational problem, being NP-hard [8] as well as W[1]-hard [7] and hard to approximate [9]. By way of contrast, finding cliques is very important in many practical applications. Recent papers describe applications in computational finance [3, 4] as well as computational biochemistry and genomics [5, 6].

Enumerating all maximal cliques needs exponential time. For instance, a recent paper by Tomita et al. [14] 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. Recently, Ito et al. [10] 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 search for all maximal cliques any more, faster enumeration algorithms are possible. Second, isolated cliques may be an intrinsically relevant concept, because these cliques can represent structures with

* Supported by the DFG, Emmy Noether research group PIAF, NI 369/4.

** Supported by a PhD fellowship of the Carl-Zeiss-Stiftung.

*** Supported by the DFG, projects ITKO, NI 369/5 and AREG, NI 369/9.

particularly interesting properties that are detected in this way. Ito et al. [10] stated the linear-time enumerability of isolated cliques (for constant "isolation factor" c) by claiming an algorithm running in $O(4^c \cdot c^5 \cdot m)$ time for an m-edge graph. Unfortunately, their algorithm is flawed. Hence, in our recent theoretical work [12], we presented a nontrivially repaired algorithm with the same running time. Moreover, we introduced two closely related isolation concepts called min-c-isolation and max-c-isolation, respectively.

Here, we present the following results. First, we give a theoretical improvement for the enumeration of Ito et al.'s isolated cliques, now achieving a running time of $O(2.89^c \cdot c^2 \cdot m)$. The main focus of our work, however, is on computational studies, applying the three isolation concepts to random feature graphs (in the $G_{n,m,p}$ model) and financial networks. The random graphs serve as benchmark instances for charting the tractability borderlines of our algorithms. We find for min- and max-isolation that the algorithms are output-sensitive, and hence lead to very fast clique enumeration for lower values of c. For isolation as introduced by Ito et al., however, this is not always the case, and sometimes even for intermediary values of c the enumeration becomes infeasible. Interestingly, we observe that the practical and theoretical bottlenecks of the algorithm differ. The financial networks serve as an example of how isolation can be used to find particularly interesting cliques. In our experiments, we analyze the so-called *clique performance*, which represents the profit/loss of the underlying financial instruments, and observe significant differences between the three isolation concepts.

2 Fundamentals & Algorithms & Implementation Issues

The fundamental strategy and several basic ideas go back to Ito et al. [10]; while their work contains serious flaws as spotted in [12], it initiated the study of isolation in context with the enumeration of maximal cliques. Besides sketching the fundamental algorithmic ideas, we additionally describe a new theoretical result leading to an improved running time.

Fundamentals. Ito et al. [10] 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 subset $S \subseteq V$ 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 $V \setminus S$. In follow up-work, we further introduced the concepts of min-c-isolation and max-c-isolation as follows [12]. 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 $V \setminus S$. Fig. 1 illustrates the three concepts.

For notational simplification we will mostly use the terms min-isolation, avg-isolation, and max-isolation. Note that by definition min-c-isolation is weaker than avg-c-isolation in the sense that every avg-c-isolated clique is also min-c-isolated but not vice versa. The enumeration of maximal min-c-isolated cliques

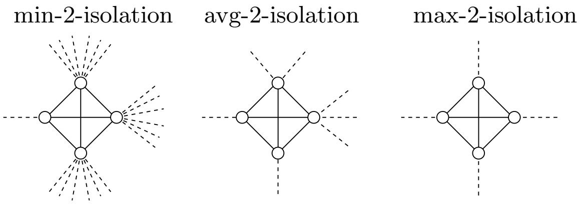


Fig. 1: Isolated 4-vertex cliques for $c = 2$ and the three isolation concepts. The dashed lines denote outgoing edges.

yields cliques that are at least as large as and often larger than avg-c-isolated cliques. By way of contrast, max-c-isolation is stronger than avg-c-isolation. 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. The theoretical study [12] of these three concepts led to the following theorem. Herein, n denotes the number of graph vertices and m the number of edges.

Theorem 1 ([12]). *Maximal avg-c-isolated cliques can be enumerated in $O(4^c \cdot c^3 \cdot m)$ time, maximal min-c-isolated cliques in $O(2^c \cdot c \cdot m + n \cdot m)$ time, and maximal max-c-isolated cliques in $O(2.44^c \cdot m)$ time.*

Ito et al. [10] only considered avg-c-isolation and claimed a running time of $O(4^c \cdot c^5 \cdot m)$ for the enumeration of maximal avg-c-isolated cliques as their main result. Besides sketching the algorithms behind Theorem 1, we also prove a new result, improving the time bound for avg-c-isolated cliques to $O(2.89^c \cdot c^2 \cdot m)$.

Algorithms. In the following, we focus on describing the algorithm for avg-isolation. The corresponding algorithms for min-isolation and max-isolation have the same basic structure [12]; however, they differ in technical details we cannot go into here. 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. Let $N_+[v] := \{u \in N[v] \mid u > v\} \cup \{v\}$ and $N_-(v) := \{u \in N(v) \mid u < v\}$. In an avg-isolated clique, the vertex with the lowest index is called the *pivot* of the clique [10]. Clearly, a pivot has less than c outgoing edges. Since every avg-isolated clique has a pivot, we can enumerate all maximal avg-isolated cliques of a graph by enumerating all maximal avg-isolated cliques with pivot v for each $v \in V$ and then removing those avg-isolated cliques with pivot v that are a subset of an avg-isolated clique with another pivot.

The enumeration of maximal avg-isolated cliques *with pivot v for a $v \in V$* is called the *pivot procedure.* It comprises three successive stages:

Trimming stage. This stage builds in polynomial time a candidate set C that is a superset of all avg-isolated cliques with pivot v. The set C is initialized with $N_+[v]$, and then vertices that obviously cannot be part of an avg-isolated clique with pivot v are removed from C. In particular, we remove vertices from c that have too many neighbors outside of C or too few neighbors in C.

Enumeration stage. This stage enumerates cliques with pivot v. Let C be the candidate set after the trimming stage, and $|N[v] \setminus C| = d$. In total, we can delete at most $c-1$ vertices from $N[v]$, since otherwise v obtains too many outgoing edges. Therefore, $\tilde{c} := c - 1 - d$ 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[1] of size *at most* $\tilde{c}$ in the complement graph $\overline{G[C]}$: First, we enumerate all *minimal* vertex covers and thus obtain maximal cliques in the candidate set C. Then, to also capture avg-isolated cliques that are subsets of non-avg-isolated cliques enumerated this way, for each of these cliques, we enumerate all maximal subsets that fulfill the isolation condition. It is possible to show [12, Lemma 1] that given a non-avg-isolated clique C', we may only remove vertices from the set of vertices with the c highest indices in order to obtain a maximal avg-isolated clique that is a subset of C'. This is done in a brute-force way by enumerating subsets of the set of vertices that may be deleted, and then checking for each such subset whether removing this subset yields an avg-isolated clique. This stage has running time $O(2^c \cdot c^6 \cdot m)$ [12].
Screening stage. In the screening stage, all cliques that are either not avg-isolated or that are avg-isolated but not maximal are removed. First, avg-isolation is checked. Next, 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 running time of this stage is $O(4^c \cdot c^3 \cdot m)$.

Min-isolation and max-isolation lead to conceptually simpler pivot procedures. The new theoretical contribution provided in this paper when compared to our previous theoretical work [12] is to show an improvement of the screening stage in the case of the avg-isolation concept.

Suppose that an enumerated avg-isolated clique C with pivot v is not maximal. Then there must be a nonempty vertex set S such that $C \cup S$ is an avg-isolated clique. Obviously, $S \subseteq N[v] \setminus C$. Also, S must be a clique and all vertices in S have to be adjacent to all vertices in C. Let $D \subseteq N[v] \setminus C$ such that D contains exactly the vertices that are adjacent to all vertices in C. To test the maximality of C, we first enumerate all maximal cliques $D' \subseteq D$. Then, for each such clique D', the set $C \cup D'$ is a clique. If $C \cup D'$ is also avg-isolated, then C is clearly not maximal and thus removed from the output. If $C \cup D'$ is not avg-isolated, however, then we have to check whether there is an avg-isolated subset of $C \cup D'$ that is also a superset of C. This can be done by removing the vertices of highest degree from D' until either $C \cup D'$ becomes avg-isolated or D' is empty. In the first case, C is not a maximal avg-isolated clique and is thus removed from the output. In the second case, C is a maximal avg-isolated clique in $C \cup D'$. If this can be shown for all maximal cliques $D' \subseteq D$, then C is a

[1] A vertex cover of a graph is a subset D of vertices such that each graph edge has at least one endpoint in D. See Abu-Khzam et al. [1] for algorithm engineering results in determining minimum-cardinality vertex covers.

maximal avg-isolated clique in G. With this maximality test, we can improve the asymptotic running time bound of the enumeration algorithm (cf. Theorem 1).

Theorem 2. *Maximal avg-c-isolated cliques of an m-edge graph can be enumerated in $O(2.89^c \cdot c^2 \cdot m)$ time.*

Proof. Since the trimming stage and enumeration stage of the algorithm have not changed, their running time amounts to $O(c^4 \cdot m + 2^c \cdot c^2 \cdot c^4 m) = O(2^c \cdot c^6 \cdot m)$ [12]. In the screening stage of the pivot procedure, we have to test each clique for maximality. At most $2^{c-1} \cdot c$ cliques are enumerated during the enumeration stage of the pivot procedure for a pivot v. For any enumerated avg-isolated clique C, we have to enumerate all maximal cliques in a subset of $N[v] \setminus C$. Since $|N[v] \setminus C| \leq c-1$, this can be done in $O(3^{c/3})$ time [14]. For each pair of an enumerated avg-isolated clique C and a maximal clique D', we decide whether a subset of $C \cup D'$ is avg-isolated by successively removing the vertices with highest degree from D'. Clearly, this can be done in $O(c)$ time. Overall, one execution of the screening stage thus has a worst-case running time of $O(2^c \cdot c) \cdot O(3^{c/3}) \cdot O(c) = O(2.89^c \cdot c^2)$. There are n runs of the screening stage, and together with the running times of the other stages, we achieve a total worst-case running time of $O(2^c \cdot c^6 \cdot m) + O(2.89^c \cdot c^2 \cdot n) = O(2.89^c \cdot c^2 \cdot m)$. □

Implementation Issues. We briefly describe some notable differences between the theoretical algorithms [12] and their actual implementations.[2]

Min-isolation. In the trimming stage, we remove vertices that have lower index than the pivot (this differs from the description in [12]). 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 $O(2.44^c \cdot c \cdot m)$ can be shown using a maximum clique algorithm in the screening stage (for details see [12]). 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

[2] 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/.

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 by Koch [11]. 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). We also use this algorithm as a comparison point for the running times of our clique enumeration algorithms.[3]

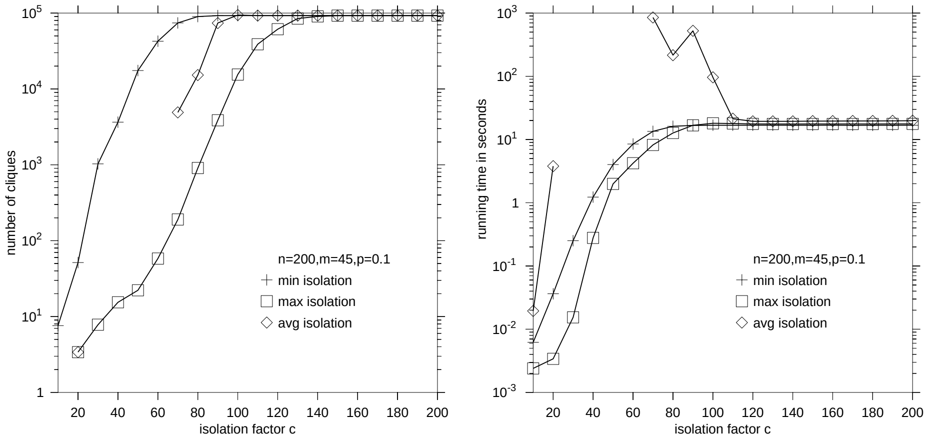
3 Experimental Results

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 of avg-isolated cliques 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 for avg-isolation in the diagrams.

Synthetic Data. We generated random graphs using the $G_{n,m,p}$ model (see Behrisch and Taraz [2] 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). Since every nonempty intersection of vertex sets corresponding to some features defines a maximal clique, these graphs contain very many maximal cliques, and are tough inputs for clique enumeration.

Our main finding is that enumerating min- and max-isolated cliques is feasible over a far wider 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 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. Fig. 2a shows the number of cliques output for varying c averaged over 5 instances. The average number of maximal cliques is 92611. 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 Fig. 2b, 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,

[3] Note that we could not perform comparisons with the claimed fastest general clique enumeration algorithm by Tomita et al. [14], since the code is unavailable.



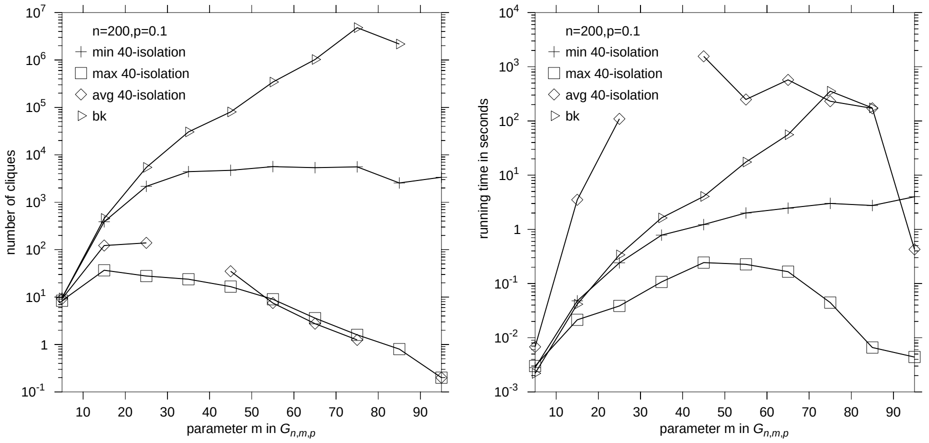
(a) (b)

Fig. 2: $G_{n,m,p}$ model with $n = 200$, $m = 45$, and $p = 0.1$. Average running time for Bron–Kerbosch is 5.06 seconds.

since 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 the only part 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, when enumerating the *whole* set of maximal cliques, all three algorithms are about 4 times slower, but min- and max-isolation are significantly faster when the output is restricted by a small c (see Fig. 2).

We next examine variation of m (Fig. 3). More features lead to an exponential growth of the number of maximal cliques (Fig. 3a). 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-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.

Similar observations were made for varying values of p and n. For both p and n, increasing the parameter value leads to an exponential growth in the number of maximal cliques of the graph. 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



(a)

(b)

Fig. 3: $G_{n,m,p}$ 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).

max-isolated cliques were output-sensitive while this was not the case for avg-isolation.

Financial Networks. Many works on financial network analysis are based on market graphs (see, e.g., [13]). We generated market graphs from publicly available stock data[4]. 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 [3].

Experimental Setup. We considered various market graphs based on the daily fluctuations of several thousand 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. [3, 4].

The following diagrams rely on data from 2204 financial instruments beginning at 2003-12-02 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

[4] We used the data from finance.yahoo.com.

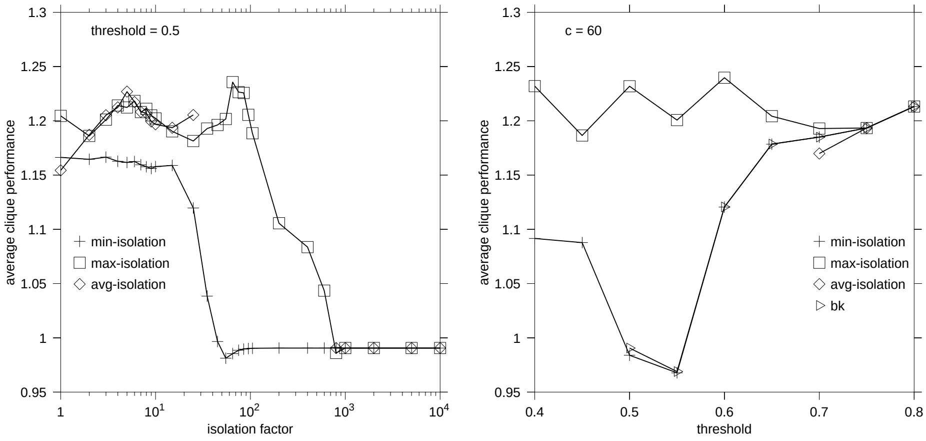
(in the qualitative sense). Note that the graphs do not include financial instruments whose values get below one dollar in the considered time period, since such "penny stocks" often show strong daily fluctuations, which are additionally biased by the rounding of the available data. In the experiments with fixed threshold, the threshold is set to $\theta = 0.5$ as proposed by Boginski et al. [4] 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 very low threshold levels, the graph gets too dense to contain many isolated cliques, and for very high threshold levels, the graph can get 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.

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 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. Surprisingly, we also find that enumerating all maximal cliques using the algorithm for min-∞-isolation is faster than Bron–Kerbosch by one order of magnitude.

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. Boginski et al. [3, 4] 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 can observe (Fig. 4a) that the performance of the enumerated min-, max-, and avg-isolated cliques is better for lower 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



(a) (b)

Fig. 4: Average clique performance in a market graph based on 500 consecutive trading days beginning at 2003-12-02. Note that the performance of the NASDAQ in the considered time period is 1.01.

especially high performance for intermediate isolation levels; we can observe a peak of the performance for max-isolation around $c = 100$. 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 for low isolation factors ($c \leq 20$) we could not observe a significant general difference of the performance of the three isolation concepts. In our example (Fig. 4a), max- and avg-isolation are slightly better for low isolation factors, but there are other graphs (based on other time periods), for which min-isolation performs better. Note that 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* maximal cliques.

When varying the threshold value, Fig. 4b shows that the performance of max-isolation is relatively independent of the threshold level, whereas min-isolated and all maximal cliques perform better for higher threshold levels. 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.

Possible Applications. We believe that especially max-isolated cliques have some interesting properties with respect to the average clique performance: First of all, the average performance of max-isolated cliques is relatively independent from the chosen threshold values. This is beneficial in practice, as finding a good threshold value is usually a relatively difficult task. Moreover, looking more closely at the cliques responsible for the peak of the performance for intermediate isolation levels, we observe that these cliques represent some niche in the market. For instance, in Fig. 4a 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 2003-12-02) and becomes even more pronounced for graphs based on later time periods (that is, beginning after 2003-12-02). This indicates that max-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. [3]). Here, a more thorough and detailed study is necessary, cooperating with financial experts.

4 Conclusion and Outlook

Our results indicate the relevance of the newly introduced isolation concepts [12] in comparison with the older avg-isolation [10]. For min- and max-isolation, the enumeration algorithms show output-sensitivity. Therefore, for both of these isolation concepts the restricted number of cliques output can make enumeration algorithms for isolated cliques much faster than the standard Bron–Kerbosch algorithm. However, for avg-isolation, further algorithmic improvements have to be made in order to obtain output-sensitivity. In particular, the enumeration of isolated subsets of non-avg-isolated cliques needs to be improved. For certain instances the c-isolation algorithms are faster than Bron–Kerbosch even for $c = \infty$, which results in the same output as the Bron–Kerbosch algorithm has. It would be interesting to see whether we could further optimize our implementations for this goal and for which kind of graphs we see a gain over Bron–Kerbosch. Our findings with financial networks support that isolation provides "interesting" cliques. In particular, max-isolated cliques perform better for intermediary isolation factors. This should be analyzed more thoroughly (with the help of financial experts) to better understand what distinguishes stocks in max-isolated cliques from those in general maximal cliques or min-isolated cliques, and hence what leads to the difference in clique performance. Furthermore, are there any application scenarios in which the relatively weak min-isolation concept is useful? For example, does the pivot element of a min-isolated clique which has the fewest (and thus less than c) neighbors outside of the clique somehow characterize the whole clique?

Acknowledgements. We thank our student assistants Robert Bredereck and Manuel Sorge for their excellent support in gathering experimental data.

References

[1] F. N. Abu-Khzam, R. L. Collins, M. R. Fellows, M. A. Langston, W. H. Suters, and C. T. Symons. Kernelization algorithms for the vertex cover problem: Theory and experiments. In *Proc. 6th ALENEX*, pages 62–69. SIAM, 2004.

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

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

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

[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] 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.

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

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

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

[10] 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.

[11] I. Koch. Enumerating all connected maximal common subgraphs in two graphs. *Theoret. Comput. Sci.*, 250(1-2):1–30, 2001.

[12] C. Komusiewicz, F. Hüffner, H. Moser, and R. Niedermeier. Isolation concepts for enumerating dense subgraphs. In *Proc. 13th COCOON*, volume 4598 of *LNCS*, pages 140–150. Springer, 2007.

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

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