Computing and Informatics, Vol. 34, 2015, 1397–1417

CONSTRUCTION OF NEAR-OPTIMAL VERTEX CLIQUE COVERING FOR REAL-WORLD NETWORKS

David Chalupa

Institute of Applied Informatics
Faculty of Informatics and Information Technologies
Slovak University of Technology
Ilkovičova 2
842 16 Bratislava, Slovakia
e-mail: chalupa@fiit.stuba.sk

Abstract. We propose a method based on combining a constructive and a bounding heuristic to solve the vertex clique covering problem (CCP), where the aim is to partition the vertices of a graph into the smallest number of classes, which induce cliques. Searching for the solution to CCP is highly motivated by analysis of social and other real-world networks, applications in graph mining, as well as by the fact that CCP is one of the classical NP-hard problems. Combining the construction and the bounding heuristic helped us not only to find high-quality clique coverings but also to determine that in the domain of real-world networks, many of the obtained solutions are optimal, while the rest of them are near-optimal. In addition, the method has a polynomial time complexity and shows much promise for its practical use. Experimental results are presented for a fairly representative benchmark of real-world data. Our test graphs include extracts of web-based social networks, including some very large ones, several well-known graphs from network science, as well as coappearance networks of literary works' characters from the DIMACS graph coloring benchmark. We also present results for synthetic pseudorandom graphs structured according to the Erdős-Rényi model and Leighton's model.

Keywords: Clique covering, independent set, community detection, stochastic algorithms, heuristics

Mathematics Subject Classification 2010: 90C59, 05B40, 05C69

1 INTRODUCTION

Currently, one of the most popular subfields of computer science is the *network analysis*. For the purpose of this paper, a network will be formalized simply as an *undirected graph* $G = [V, E]$. Networks with non-trivial structure are often referred to as *complex networks* [3, 29]. Complex networks can be found in a diverse spectrum of disciplines and include social networks, research citation networks, networks in bioinformatics or computer networks. Language networks are also of considerable interest [25]. With the rise of popularity of web-based social networks such as Facebook, Google+ or LinkedIn, this interdisciplinary topic has gained more attention also in public. Software tools are designed to support investigation and analysis of complex networks [10].

In addition to network analysis, *graph mining* is a very closely related field [1]. Although this field shares many similarities with social network analysis, probably the most significant difference is in the *size of the studied graphs* [7]. While social network analysis is focused on detailed study of small networks, in graph mining the attention is aimed to very large graphs, including computational efficiency of the methods, which are used. From this point of view, this paper could probably be assigned to graph mining, although it shares many features with social network analysis. However, first and foremost, the research presented here is based on the principles of heuristic computing since the methods we use are stochastic heuristic algorithms. We are dealing with a highly multidisciplinary topic.

Many complex networks are studied with regard to their community structure and clustering properties [17]. This basically means that there are subgraphs of these networks with relatively many edges, which are referred to as *communities* or *clusters*. Other studied aspects include the process of evolution and the degree distribution of these networks [30]. *Community detection* is sometimes also called graph clustering [31]. State-of-the-art indicates that graph clustering is a set of similar optimization problems rather than a single concept. Therefore, it is hard to formalize, what the objective of community detection is. In addition, some measures of quality in graph clustering are proven to lead to NP-complete problems [33].

One of the ways, how to formalize this concept, is to perceive community as a *clique*, i.e. a collection of vertices, where each pair is adjacent. On the surface, this might seem like a strong restriction, however, in a strict interpretation of the word, a community can be seen as a group, where all members know each other. The problem of decomposition of the vertices of a graph into groups, which induces as few cliques as possible, is called the (vertex) *clique covering problem* (CCP) [8]. CCP is interesting not only for its relation to community structure but also for the fact that it belongs to the NP-hard problems, since its decision variant is one of Karp's 21 classical NP-complete problems [19]. We note that a related but not equivalent problem is the edge clique covering problem, for which a similar and more theoretical study in the domain of complex networks is available [3]. Other graph covering problems are also studied, e.g. generalizations of the vertex cover problem [24].

In this paper, we propose a heuristic method for CCP based on combining a constructive and a bounding heuristic. The constructive heuristic is an *iterated greedy (IG) clique covering* algorithm. IG does not generally guarantee the quality of the obtained solution. It is closely related to evolutionary computation methodologies, which have a large spectrum of interesting applications in data analysis and prediction [15, 34]. Since CCP is NP-hard, even determining whether there is a solution with $k-1$ cliques if we know that there is one with k cliques, is also a computationally hard problem. Therefore, it is practically interesting to bound the clique covering number from below. Thus, we designed an approach to find a suitable lower bound. The theoretical background for the lower bound is developed more broadly, since two lower bounds can be combined. Consequently, we provide evidence that only one of these bounds seems to be interesting for real-world networks. This lower bound is based on the size of the maximum independent set for which a simple *randomized local search (RLS)* heuristic is proposed.

The experimental results are presented for a set networks which carry real-world information. Some of these graphs were introduced in our research and some of them are taken from other sources. The instances include extracts of web-based social networks with up to 2×10^4 vertices, several well-known graphs from network science and coappearance networks of classical literary works' characters from the DIMACS graph coloring benchmark. For these instances, in 13 out of 17 cases, optimality of the constructed clique covering was proven by the lower bound. Also, in the other 4 cases, the interval for the optimum was significantly reduced, i.e. near-optimal solutions were found. In addition, the running time of the proposed approach looks promising for its practical use. In contrast to this result, we determined the lower and upper bounds for the synthetic pseudorandom graphs following the Erdős-Rényi model and Leighton's model which exhibit much larger difference between the lower and upper bound. As a result of these observations, we conclude that the proposed method shows much promise in handling clique covering in complex networks and offers an interesting tool for solving this NP-hard problem in practical circumstances.

Regarding the experimental evaluation, we also note that the standard approach in the field of heuristics is that results of an algorithm are compared to other results from the literature. Generally, if the algorithm finds better suboptimal solutions, it is considered to offer an improvement in the state-of-the-art algorithms. However, the case of this paper is different, since we aim to provide a direct numerical proof of optimality. Therefore, in the cases, where we prove optimality, it is not relevant to compare quality.

Additionally, to the best of our knowledge, we do not know about a research on standardized heuristics for CCP. Combining other algorithms for the construction and the lower bound might be relevant, however, our approach provides solid results in practically interesting conditions. A further improvement seems to be possible, but mainly for synthetic data as we will give reasons in Section 4. The heuristic might also be compared to an exact algorithm, regarding its runtime, however, such an algorithm would have an exponential runtime, while our heuristic works in polynomial time. Thus, the empirical comparison to other methods is absent in

this paper, since it would probably not provide even slightly more valuable information.

The paper is structured as follows. In Section 2, we provide an overview of the problem and the related work. In Section 3, we specify the basic idea of our approach and describe both the IG heuristic for CCP which is used to construct the suboptimal clique covering, and the heuristic for maximum independent set which is used to find a lower bound. In Section 4, we evaluate the approach on the selected data sets, including real-world instances and pseudorandom instances, analyze the obtained results and provide a short discussion. Finally, in Section 5, we present conclusions and summary of this work.

2 BACKGROUND AND RELATED WORK

Let $G = [V, E]$ be an undirected graph. The density of G is defined as the ratio of the number of edges of the graph to the number of all pairs of vertices, i.e. $d(G) = 2|E|/(|V|(|V| - 1))$. If $d(G) = 1$, then G is called complete graph or clique (the term clique is more often used to refer to a subgraph of another graph). If $d(G) = 0$, then G is called independent set.

The *(vertex) clique covering problem (CCP)* is defined as the problem of searching for a partitioning of V into classes $V_1, V_2, \ldots, V_k$ such that:

- each vertex is in exactly one class, i.e. $\cup_{i=1}^{k} V_i = V$ and $\forall i, j = 1, 2, \ldots, k$ such that $i \neq j$ it holds that $V_i \cap V_j = \emptyset$ and
- each class induces a clique, i.e. $\forall i = 1, 2, \ldots, k$, $d(G(V_i)) = 1$.

Searching for the minimum k, for which this is possible, is an NP-hard problem [19]. This minimum k will be referred to as the *clique covering number* and denoted by $\vartheta(G)$. We also note that CCP is in a complementary relationship with the graph coloring, thus, clique covering of G represents a graph coloring of $\overline{G}$ and vice versa. However, the practical requirements on heuristics for these problems are different, especially due to sparseness and structure of graphs [8]. The minimum number of colors needed to color a graph is called *chromatic number* and is denoted by $\chi(G)$.

Similarly, the problems of searching for the maximum independent set and maximum clique are also NP-hard [19]. The size of maximum independent set will be denoted by $\alpha(G)$ and the size of maximum clique will be denoted by $\omega(G)$.

At this point, let us briefly review the heuristics, which are currently available for estimation of $\vartheta(G)$ and $\alpha(G)$. For clique covering, classical graph coloring algorithms, such as the Brélaz's heuristic [5], are relatively successful and scalable. The greedy coloring can be extended to iterated greedy heuristics [9]. These ideas were successfully adapted to CCP, showing that iterated greedy (IG) clique covering is propitious for its scalability and solid quality of results [8]. In this paper, we also apply IG to construct the clique covering. Other related algorithms include especially

a very large spectrum of k-fixed local search and evolutionary algorithms, where the number of colors is set to a constant as a part of the problem instance [14].

For the maximum indepedent set, the most popular heuristics include the classical GRASP heuristic [12] or local search heuristics based on swaps of single vertices with pairs or triplets [2].

Last but not least, there is a large spectrum of application areas, for which the construction of near-optimal clique covering can be helpful. These areas include clique detection in the social network analysis [29] or protein interaction networks in bioinformatics [16]. In operations research, the most interesting applications are tied to the closely related graph coloring problem and include frequency assignment in mobile radio networks [32]. Another popular and important field of interest is found in the detection of clusters on the World Wide Web [13, 23].

3 SPECIFICATION OF OUR APPROACH

The basic idea of our heuristic is very simple, but we will shortly demonstrate that it works well in practically interesting circumstances. We begin with a more general idea and then simplify it so that it will better fit the practical purposes of our approach. In Lemma 1, we have first proved bounds on the clique covering number $\vartheta(G)$, in a similar way to the bounds which are used in graph coloring.

Lemma 1. Let G be an undirected graph with minimum degree $\delta_{min}(G)$, clique covering number $\vartheta(G)$, maximum independent set size $\alpha(G)$ and maximum clique size $\omega(G)$. Then, $\vartheta(G)$ is bounded in the following way:

$$\max\left\{\alpha(G), \frac{|V|}{\omega(G)}\right\} \leq \vartheta(G) \leq |V| - \delta_{min}(G). \tag{1}$$

Proof. We first prove that $\max\left\{\alpha(G), \frac{|V|}{\omega(G)}\right\} \leq \vartheta(G)$. The property that $\alpha(G) \leq \vartheta(G)$ is implied by the fact that each vertex of an independent set needs its own clique to be covered. The second inequality that $\frac{|V|}{\omega(G)} \leq \vartheta(G)$ can be proven by an equivalent bound in graph coloring, since $\vartheta(G) = \chi(\overline{G})$. In graph coloring, it is well-known that $\frac{|V|}{\alpha(G)} \leq \chi(G)$ [4, 21], which yields $\frac{|V|}{\omega(\overline{G})} \leq \vartheta(\overline{G})$. By substituting $\overline{G}$ with G, we obtain the desired bound.

The upper bound can be obtained similarly by considering the performance of greedy graph coloring algorithm [35]. It is known that greedy graph coloring uses at most $\Delta(G)+1$ colors, where $\Delta(G)$ is the maximum degree of a vertex in G [11, 35]. Therefore, $\chi(G) \leq \Delta(G)+1$, which implies that $\vartheta(\overline{G}) \leq (|V| - 1 - \delta_{min}(\overline{G})) + 1 = |V| - \delta_{min}(\overline{G})$. By substituting $\overline{G}$ with G, we prove the upper bound. □

This result has to be slightly refined for practical use, since determining $\alpha(G)$ and $\omega(G)$ are both NP-hard problems. The upper bound of $|V| - \delta_{min}(G)$ can be

useful for graphs with high value of $\delta_{min}(G)$. However, $\delta_{min}(G)$ can generally be very low for real-world networks. Thus, we can simply substitute the upper bound by result of a constructive algorithm.

For the lower bound, we have to consider that in practice, $\alpha(G)$ and $\omega(G)$ should be suitably bounded. The value $\alpha(G)$ requires a lower bound and $\omega(G)$ requires an upper bound so that the estimate will not exceed $\max\left\{\alpha(G), \frac{|V|}{\omega(G)}\right\}$. While a lower bound $\alpha_L(G) \leq \alpha(G)$ can be estimated constructively, the upper bound $\omega_U(G) \geq \omega(G)$ requires a non-constructive approach. Some methods are known for computing such a bound [6], however, study of their suitability in this context is outside of the aim of this paper. Nevertheless, we will see that for many real-world networks, $\alpha_L(G)$ is a tighter bound than $|V|/\omega_U(G)$. Thus, we simplify Lemma 1 to the following formula:

$$\alpha_L(G) \leq \vartheta(G) \leq \vartheta_U(G), \tag{2}$$

where $\vartheta_U(G)$ is the number of cliques used to cover G by a constructive algorithm. Thus, the basic concept of our heuristic is as follows. First, we find a clique covering using the iterated greedy (IG) heuristic for CCP, which is further specified in Section 3.1. The number of cliques, which were used by IG, will be used also as the upper bound $\vartheta_U(G)$. In the next step, we find a lower bound $\vartheta_L(G) = \alpha_L(G)$ for $\vartheta(G)$ using a randomized local search (RLS) algorithm for maximum independent set, which will be introduced in Section 3.2. The output of the method is the interval $[\vartheta_L(G), \vartheta_U(G)]$ for the value of $\vartheta(G)$ and a suboptimal (or possibly optimal) clique covering with $\vartheta_U(G)$ cliques.

3.1 Constructing the Suboptimal Clique Covering Using Iterated Greedy

In Algorithm 1, we give a short pseudocode of IG algorithm for CCP. Since this algorithm is not new, we only shortly summarize, how it works. For more detailed information on this method, the reader may refer to [8].

	The IG Algorithm for CCP
	Input: graph $G = [V, E]$
	Output: clique covering S of G
1	$P = random\_permutation(1, 2, \ldots, \lvert V\rvert)$
2	while *stopping_criterion* is not met
3	$[V_1, V_2, \ldots, V_k] = greedy\_clique\_covering(G, P)$
4	if $\vartheta^*(G)$ is known and $k = \vartheta^*(G)$
5	return $S = \{V_1, V_2, \ldots, V_k\}$
6	$P = [V_1, V_2, \ldots, V_k]$
7	$P = random\_permutation(V_1, V_2, \ldots, V_k)$
8	return $S = \{V_1, V_2, \ldots, V_k\}$

Algorithm 1: The IG Algorithm for CCP [8]

In IG, greedy clique covering (GCC) is used as a subroutine in the step 3, which transforms a permutation of vertices to a clique covering. GCC works in the way that it takes vertices in the order determined by permutation P and labels them so that the equally labeled vertices form the cliques. During the construction, GCC always puts the current vertex into the clique with the lowest index (label), for which it is adjacent to all of the clique's vertices. If there is no such label, a new one is used. The complexity of GCC is $\mathcal{O}(|E|)$ [8]. After the clique covering is constructed, the vertices from the same cliques, as identified by GCC, are put together to blocks in permutation in the step 6. In the step 7, the blocks are rearranged in a random order and GCC is used again. This is repeated, until a stopping criterion is met. We note that although IG reminds one of random optimization, the number of cliques is guaranteed to be non-increasing because of the fact that the blocks of the solution are preserved. Therefore, this process can possibly decrease the number of cliques, which are used, and behaves rather like a local search algorithm.

3.2 Estimating the Lower Bound Using Randomized Local Search for Maximum Independent Set

For the lower bound, we use a similar heuristic approach to estimate the size of the largest independent set. The pseudocode is given in Algorithm 2. In this heuristic, we also have a greedy algorithm, which maps a permutation of vertices to an independent set in the following way. Suppose that we have a Boolean function $a : V \to \{0, 1\}$, such that $a(v) = 1$ if and only if v can be added to the constructed independent set, i.e. it is not adjacent to any of its vertices. Otherwise, $a(v) = 0$.

We begin with an empty independent set and we put $a(v) = 1$ for each $v \in V$. Then, we process the vertices in order, which is given by the input permutation. If $a(v) = 1$, we add v to the independent set. Consequently, we set $a(w) = 0$ for all w such that $\{v, w\} \in E$. We proceed with the next vertex, until all vertices are processed this way. This greedy procedure is performed in the step 3.

To improve the initial permutation and the independent set, we use a simple *jump* operator on a uniformly randomly chosen vertex, putting it to the first position in the permutation. This is done in the steps 6-7. The other vertices are then shifted right. The resulting permutation is used again to obtain an independent set. If the new independent set is at least as large as the current one, we accept the new permutation. We continue, until no improvement is obtained in a high number of iterations. The resulting algorithm is called permutation-based randomized local search with 1-neighborhood (RLS_p^1) and is well-studied in evolutionary computation theory [27]. We will give more detailed remarks on the stopping criteria both for IG and RLS_p^1 in the section on the experimental results.

4 EXPERIMENTAL RESULTS

In this section, we present the experimental evaluation of our approach. Firstly, we introduce the instances, in which we evaluated our heuristic. Secondly, we provide

	RLS_p^1 Algorithm for the Maximum Independent Set Size
	Input: graph $G = [V, E]$
	Output: the size $\alpha(G)$ of the maximum independent set
1	$P = \mathit{random\_permutation}(1, 2, \ldots, \lvert V\rvert)$, $P^* = P$, $k^* = 1$
2	while *stopping_criterion* is not met
3	$k = \lvert \mathit{greedy\_independent\_set}(G, P)\rvert$
4	if $k \geq k^*$
5	$k^* = k$, $P^* = P$
6	$j = \mathit{uniformly\_random}(2, \lvert V\rvert)$
7	$P = \mathit{jump}(j, 1, P^*)$
8	return $\alpha(G) = k^*$

Algorithm 2: RLS_p^1 Algorithm for the Maximum Independent Set Size

detailed computational results of the approach on real-world networks. Last but not least, we compare the results obtained on real-world networks to those obtained on synthetic pseudorandom graphs. Probably the most important part is the discussion on the relation between the structure of the graph and the ability of the heuristic to obtain the optimal result and prove its optimality, or at least, to provide a narrow interval for $\vartheta(G)$.

4.1 Description of Test Instances

We divided the test instances into two large groups: complex networks, which will be studied more carefully, and synthetic pseudorandom graphs, which are studied to confront the results on real-world networks[1].

The complex networks are further divided into three categories: extracts of web-based social networks, several instances used in network science and finally, coappearance networks of characters from several works of classical literature, which are a part of DIMACS graph coloring benchmark [18].

The extracts of web-based social networks were obtained using a web crawler based on breadth-first search in the case of Social network I. These instances have from 500 up to 20 000 vertices. The extract of Social network II is an immediate neighborhood of a single user.

The instances from network science were taken from various sources. Network *adjnoun* denotes word adjacencies of nouns and adjectives in the novel David Copperfield [28]. Network *netscience* is a collaboration network for both theoretical and experimental studies in the field of network science [28]. Instance *zachary* is a social network of friendships in a karate club at a university [36]. Network *football* describes football games in a season of an American college football league [17]. Network *lesmis* is a coappearance network for the characters of Les Miserables [20].

[1] All these instances are publicly available or a link to their direct source is provided at: `http://www.fiit.stuba.sk/~chalupa/benchmarks/ccp`. All of the instances are available in COL or GML format. In the provided files, the social networks are anonymized.

Finally, network *as − 22july06* is a relatively large Internet snapshot on the level of autonomous systems, made by M. Newman[2].

In addition to network *lesmis*, the DIMACS graphs of coappearances also come from the Knuth's Stanford GraphBase [20]. These include coappearance networks for classical literary works' characters, including Anna Karenina, David Copperfield, Huckleberry Finn, Iliad and Odyssey and Jean Valjean.

The synthetic pseudorandom graphs include uniform random graphs, generated according to the Erdős-Rényi model, which have from 1 000 up to 20 000 vertices and density 0.1 or 0.01 (in the case of the largest graph). These graphs are generated in the way that we begin with a set of isolated vertices and put an edge between each pair indepedently with probability p, which is equal to the desired density of the graph. The Leighton graphs, on the other hand, contain embedded cliques of predefined sizes, which are randomly connected and aim to model typical large scheduling problems [22]. Leighton graphs are also a standard part of the DIMACS graph coloring benchmark [18].

4.2 Computational Results of the Approach

At this point, we present the computational results of our approach on the 17 complex network instances, and shortly discuss the optimality and efficiency issue. All experiments were conducted on a standard machine with Intel Core i5 CPU @ 3.10 GHz and 4 GB RAM. All experiments were confirmed in 30 independent runs.

In both the IG heuristic for the construction and the RLS_p^1 heuristic for the lower bound, we used stopping criteria based on the number of iterations without improvement.

In the IG for the construction of the clique covering, the previous empirical and analytical evidence suggests that the successful moves of the heuristic are likely to occur when a particular block comes first in the permutation [8]. The probability of putting a particular block first in the random move is $1/k$, where k is the current number of cliques in the solution. With this in mind, we consider the probability that in ck moves, for some constant $c > 1$, this particular block was not chosen to be first. This probability is $(1 - 1/k)^{ck} \approx e^{-c}$. For $c = 5$, this probability is only approximately 0.7%. Therefore, we stop whenever $5k$ iterations without improvement occur.

In the lower bound computation, based on RLS_p^1 for the maximum independent set, the stopping criterion is similar. We consider the probability that a particular vertex will not be put first in the permutation in $c|V|$ steps, where $c > 1$ is again a constant. This probability is $(1-1/|V|)^{c|V|} \approx e^{-c}$, thus, we stop the process when $c|V|$ iterations without improvement occur.

We note that nonlinearities in the waiting time for improvement can occur both in IG and RLS_p^1, especially in very hard instances. However, since their source

[2] This instance was not previously published in a research paper. The Internet snapshot is published on this site: `http://www-personal.umich.edu/~mejn/netdata/`.

is closely related to specific properties of the instances and we want to keep the heuristic simple, we rather use the linear stopping criteria.

Source of G	File Name	ϑ^*	Succ.	Iter.	CPU
Web-based social network extracts [8]					
Social network I.	*soc*500	$\vartheta \leq 377$	30/30	1 888	< 1 s
$\|V\| = 500, \|E\| = 924$		$\vartheta \geq 377$	30/30	3 764	< 1 s
Social network I.	*soc*1000	$\vartheta \leq 759$	30/30	3 801	1 s
$\|V\| = 1\,000, \|E\| = 1\,876$		$\vartheta \geq 759$	30/30	7 960	< 1 s
Social network I.	*soc*2000	$\vartheta \leq 1\,471$	30/30	7 372	4 s
$\|V\| = 2\,000, \|E\| = 4\,124$		$\vartheta \geq 1\,470$	30/30	17 430	< 1 s
Social network I.	*soc*10000	$\vartheta \leq 6\,618$	30/30	33 276	89 s
$\|V\| = 10\,000, \|E\| = 28\,675$		$\vartheta \geq 6\,618$	17/30	124 120	31 s
Social network I.	*soc*20000	$\vartheta \leq 12\,764$	30/30	64 651	366 s
$\|V\| = 20\,000, \|E\| = 63\,245$		$\vartheta \geq 12\,764$	25/30	274 529	147 s
Social network II.	*soc*52	$\vartheta \leq 15$	30/30	78	< 1 s
$\|V\| = 52, \|E\| = 822$		$\vartheta \geq 15$	30/30	508	< 1 s
Network science instances					
Adjective-noun adjacencies [28]	*adjnoun*	$\vartheta \leq 55$	30/30	364	< 1 s
$\|V\| = 112, \|E\| = 425$		$\vartheta \geq 53$	30/30	1 145	< 1 s
Network science collaborations [28]	*netscience*	$\vartheta \leq 630$	30/30	3 453	1 s
$\|V\| = 1\,589, \|E\| = 2\,742$		$\vartheta \geq 630$	30/30	11 874	< 1 s
Les Miserables network [20]	*lesmis*	$\vartheta \leq 35$	30/30	176	< 1 s
$\|V\| = 77, \|E\| = 254$		$\vartheta \geq 35$	30/30	546	< 1 s
Zachary Karate Club [36]	*zachary*	$\vartheta \leq 20$	30/30	101	< 1 s
$\|V\| = 34, \|E\| = 78$		$\vartheta \geq 20$	30/30	232	< 1 s
American College Football [17]	*football*	$\vartheta \leq 22$	22/30	118	< 1 s
$\|V\| = 115, \|E\| = 616$		$\vartheta \geq 21$	30/30	1 215	< 1 s
Snapshot of the Internet	$as - 22july06$	$\vartheta \leq 19\,661$	30/30	98 312	556 s
$\|V\| = 22\,963, \|E\| = 48\,436$		$\vartheta \geq 19\,660$	26/30	192 136	128 s
Characters' coappearance networks from DIMACS coloring instances [18]					
Anna Karenina	*anna*	$\vartheta \leq 80$	30/30	402	< 1 s
$\|V\| = 138, \|E\| = 986$		$\vartheta \geq 80$	30/30	1 022	< 1 s
David Copperfield	*david*	$\vartheta \geq 36$	30/30	182	< 1 s
$\|V\| = 87, \|E\| = 812$		$\vartheta \leq 36$	30/30	715	< 1 s
Huckleberry Finn	*huck*	$\vartheta \leq 27$	30/30	136	< 1 s
$\|V\| = 74, \|E\| = 602$		$\vartheta \geq 27$	30/30	516	< 1 s
Iliad and Odyssey	*homer*	$\vartheta \leq 341$	30/30	1 711	< 1 s
$\|V\| = 561, \|E\| = 3\,258$		$\vartheta \geq 341$	30/30	4 219	< 1 s
Jean Valjean	*jean*	$\vartheta \leq 38$	30/30	192	< 1 s
$\|V\| = 80, \|E\| = 508$		$\vartheta \geq 38$	30/30	574	< 1 s

Table 1. Detailed computational results of our approach on 17 complex network instances

In Table 1 we present the results of our approach on the 17 selected complex network instances. The first two columns contain the source of the graph, its size

and the file name. The next columns contain the obtained lower/upper bounds, the success rate, the average number of iterations of IG or RLS_p^1 and the average CPU time.

On Social network I, in all cases but *soc*2000, the heuristic was able to find the optimal solution to the problem and numerically prove it. The really good news here is that we do not encounter a change in the optimality issue as $|V|$ grows. It is intriguing that *soc*2000 turned out to be the hardest instance (we note that the smaller graphs here are not necessarily subgraphs of the larger ones, so this might be possible). The values 1 470 and 1 471 are both achieved with high success rate, which indicates that this is not due to the randomness of the algorithms and is most probably related to structural properties of the instance. Additionally, the growth of the CPU time also indicates that the approach is practically interesting for its scalability. On Social network II, the result is practically comparable to the results on Social network I.

The results on the network science data and the DIMACS graphs also illustrate the qualities of the algorithm. Although most of these graphs are generally smaller than the social network instances, we use them to test our approach also on data, which is used in other literature. For all instances, our approach obtains a near-optimal solution very quickly. The only exception is the large Internet snapshot *as − 22july06*. For *netscience*, *lesmis* and *zachary*, we proved the optimality, while for the other three instances, we obtained a very narrow interval and a very solid suboptimal solution. We note that *football* and *adjnoun* seem to be intriguing, regarding their degree distribution.

Nevertheless, even in the 4 cases, where we were not able to prove the optimality, the interval for the optimum turned out to be quite narrow. In addition, the heuristic with the stopping criteria, which we suggested above, has a relatively good computational complexity. Formally, the best obtainable bound is $\mathcal{O}(|V|^2|E|)$, but from the number of iterations we can see that it approximately behaves rather like $c|V||E|$, where c is some small constant. We also point out that although both the greedy algorithms for CCP and the maximum independent set have $\mathcal{O}(|E|)$ complexity, the complexity of the greedy algorithm for CCP is influenced by a higher constant factor. This is why the estimation of the upper bound takes more time, although the number of iterations is smaller than in the estimation of the lower bound. Furthermore, the space complexity of the approach is $\mathcal{O}(|V|)$. These properties of the IG and RLS_p^1 heuristics make our approach very interesting for a good tradeoff between quality of the results and scalability for very large graphs.

4.3 Summary, Interpretation and Conceptual Comparison of the Results

Table 2 shows a summary of the previously presented results with addition of the value $\frac{|V|}{\vartheta_U(G)}$, which is the average size of a clique in the obtained solution. In Social network I, this value indicates that there are many vertices, which were isolated, however, there actually are also relatively large cliques. Interestingly, the average

size of the clique seems to grow. This suggests that, possibly, when a vertex comes to a social network, it not only tends to connect to vertices with a higher degree but also might have a tendency to join larger cliques. For the other instances we have various average clique sizes ranging from 1.17 to 5.23.

Source of G	File Name	$\vartheta_L(G)$	$\vartheta_U(G)$	$\frac{\lvert V\rvert}{\vartheta_U(G)}$
Web-based social network extracts [8]				
Social network I.	*soc*500	**377**	**377**	1.33
Social network I.	*soc*1000	**759**	**759**	1.32
Social network I.	*soc*2000	1 470	1 471	1.36
Social network I.	*soc*10000	**6 618**	**6 618**	1.51
Social network I.	*soc*20000	**12 764**	**12 764**	1.57
Social network II.	*soc*52	**15**	**15**	3.47
Network science instances				
Adjective-noun adjacencies [28]	*adjnoun*	53	55	2.04
Network science collaborations [28]	*netscience*	**690**	**690**	2.30
Les Miserables network [20]	*lesmis*	**35**	**35**	2.20
Zachary Karate Club [36]	*zachary*	**20**	**20**	1.70
American College Football [17]	*football*	21	22	5.23
Snapshot of the Internet	*as* − *22july06*	19 660	19 661	1.17
Characters' coappearance networks from DIMACS coloring instances [18]				
Anna Karenina	*anna*	**80**	**80**	1.73
David Copperfield	*david*	**36**	**36**	2.42
Huckleberry Finn	*huck*	**27**	**27**	2.74
Iliad and Odyssey	*homer*	**341**	**341**	1.65
Jean Valjean	*jean*	**38**	**38**	2.11

Table 2. Summary of the upper and lower bounds for ϑ obtained by our approach on complex network instances

Table 3 shows results of these experiments for the set of Leighton graphs. First and foremost, we can see the larger difference in the bounds. Except for *le*450_25*a*, no solution was proven to be optimal. The size of the interval $[\vartheta_L(G), \vartheta_U(G)]$ seems to vary according to the parameters of the Leighton graph. In addition, the average clique sizes are also much larger than in the real-world networks, ranging from 4.95 to 8.82. We note that most of these values are for suboptimal solutions, further improvement would make the cliques even larger.

For Erdős-Rényi graphs, we have a different situation regarding the lower bound. Both real-world networks and Leighton graphs are prone to contain large independent sets. In fact, we obtained that $\max\{\alpha(G), |V|/\omega(G)\} = \alpha(G)$ for all of the previous instances. In Erdős-Rényi uniform random graphs, both their maximum independent sets and maximum cliques are of size $\Theta(\log |V|)$ almost surely [26], i.e. for large values of $|V|$ it holds that $\max\{\alpha(G), |V|/\omega(G)\} = |V|/\omega(G)$. There is a closed formula based on this property, which can be applied to find a lower bound, since the chromatic number of an Erdős-Rényi graph satisifes $\chi(G) \geq$

Source of G	File Name	$\vartheta_L(G)$	$\vartheta_U(G)$	$\frac{\lvert V\rvert}{\vartheta_U(G)}$
Leighton graphs from DIMACS coloring instances [18]				
Leighton graph (15-colorable)	*le*450_15*a*	75	80	5.63
Leighton graph (15-colorable)	*le*450_15*b*	78	82	5.49
Leighton graph (15-colorable)	*le*450_15*c*	41	58	7.76
Leighton graph (15-colorable)	*le*450_15*d*	41	58	7.76
Leighton graph (25-colorable)	*le*450_25*a*	**91**	**91**	4.95
Leighton graph (25-colorable)	*le*450_25*b*	78	80	5.63
Leighton graph (25-colorable)	*le*450_25*c*	47	54	8.33
Leighton graph (25-colorable)	*le*450_25*d*	43	51	8.82

Table 3. Summary of the upper and lower bounds for ϑ obtained by our approach on synthetic graphs following the Leighton's model

$|V| \,/\, \lceil 2\log_d |V| - \log_d \log_d |V| + 2\log_d(e/2) + 1\rceil$ almost surely, where $d = 1/(1-p)$ for probability of edge generation p [4]. It is straightforward to adapt this to $\vartheta(G)$. By putting $d = 1/p$, we will instead calculate a lower bound for $\chi(\overline{G})$, which is obviously equivalent to $\vartheta(G)$. Hence, for Erdős-Rényi graphs, we obtain the following lower bound:

$$\vartheta(G) \geq |V| / \lceil 2\log_{1/p} |V| - \log_{1/p} \log_{1/p} |V| + 2\log_{1/p}(e/2) + 1\rceil. \tag{3}$$

The obtained values are presented in Table 4, along with the results of the IG algorithm for the construction. IG was also used with a different stopping criterion, which was simply that the search was stopped after 10^4 iterations. This led to better results than the stopping criterion, which we used for other graphs. Despite this fact, the interval $[\vartheta_L(G), \vartheta_U(G)]$ here turned out to be quite large. This indicates that solving CCP in real-world networks is a much easier problem than solving it in uniform random graphs.

Source of G	File Name	$\vartheta_L(G)$	$\vartheta_U(G)$	$\frac{\lvert V\rvert}{\vartheta_U(G)}$
Erdős-Rényi uniform random graphs				
Uniform random graph	*unif*1000_0.1	147	243	4.12
Uniform random graph	*unif*5000_0.1	617	1 066	4.69
Uniform random graph	*unif*10000_0.1	1 154	2 025	4.94
Uniform random graph	*unif*20000_0.01	3 796	6 387	3.13

Table 4. Summary of the upper and lower bounds for ϑ obtained by our approach on synthetic graphs following the Erdős-Rényi model

The previous results encourage us to look at the degree distributions of the studied graphs and at the distributions of sizes of the obtained cliques, whether there is a correlation between quality of the solution and statistical properties of the graph. In Figures 1 and 2, we plot the degree distributions and the distributions of the clique sizes in the obtained solutions in log log scale. The horizontal axis contains the

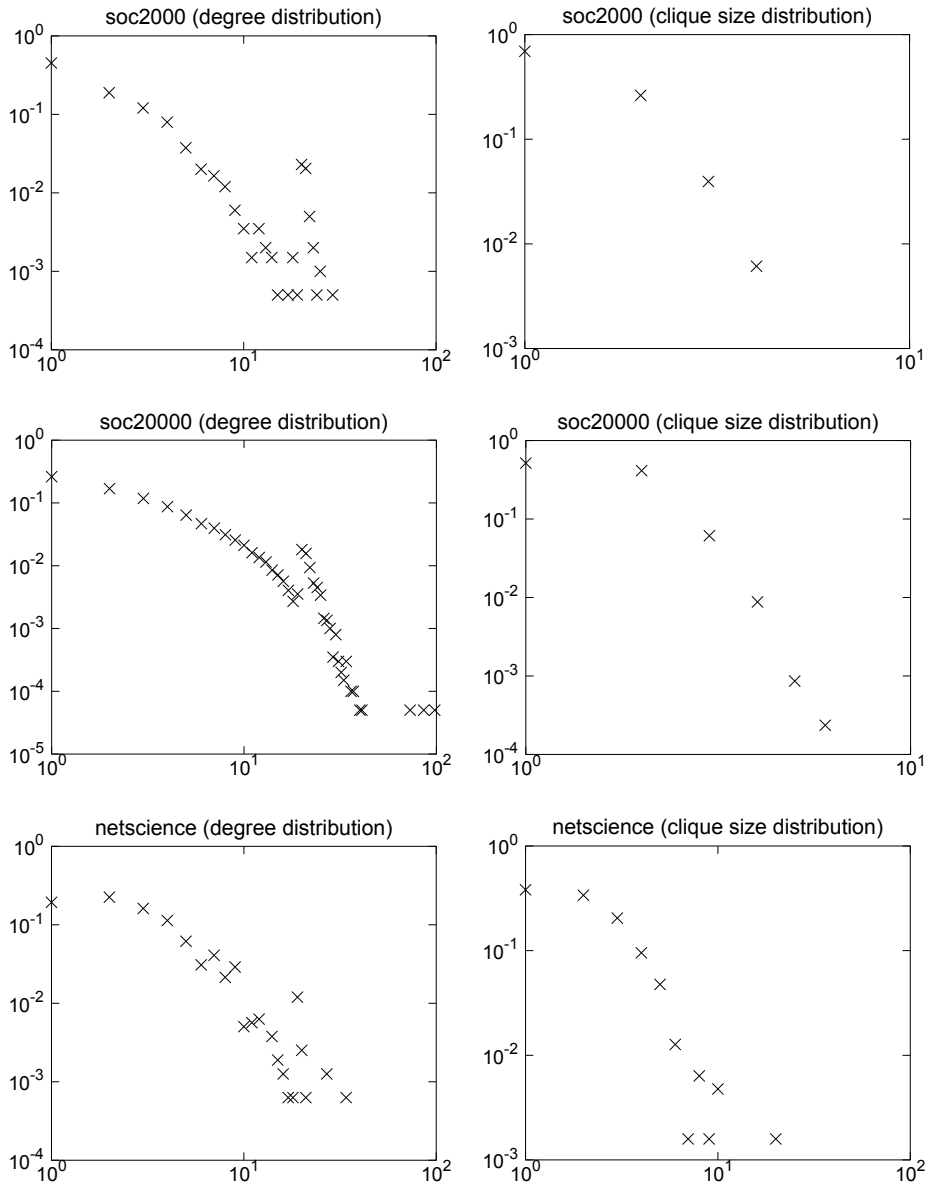

Figure 1. The visualization of degree and clique size distributions for chosen real-world network test instances and the obtained solutions in log log scale (part I)

degrees or clique sizes and the vertical axis contains the fractions of vertices or cliques with a particular degree or size. Social network extracts *soc*2000 and *soc*20000 and the network science instances *netscience*, *football*, $as-22july06$ and *homer* were chosen as probably the most representative instances of their types. For *soc*2000 and *soc*20000, we obtain relatively typical degree distributions for social networks, which seem to be well approximable by power law, which is typical for scale-free networks. Interestingly, the clique size distributions and the degree distributions seem to have a very similar profile. For *netscience*, we also obtain an analogous result, which supports a hypothesis that our approach works well with this type of degree distribution. However, it is sad that the distributions of the social network extracts seem also very similar showing no evidence why we were able to prove optimality of the result for *soc*20000 and not for *soc*2000.

For the Internet snapshot $as-22july06$ and the coappearance network *homer*, the distributions both for degrees and clique sizes are also very typical. On the other hand, the *football* instance, for which we were not able to prove optimality, has quite a peculiar degree distribution, which is reflected also in the clique size distribution. This distribution reminds once more the Erdős-Rényi graphs and Leighton graphs. We note that also the network *adjnoun*, for which we were also not able to prove the optimality, seems to have similar properties.

In Figure 3, we depict degree distributions for one Erdős-Rényi graph and two Leighton graphs, particularly $unif20000\_0.01$, $le450\_15c$ and $le450\_25b$. The Erdős-Rényi graphs are widely known to have a Poissonian degree distribution [4, 26], which is well illustrated by the figure. However, the distributions of the Leighton graphs are also closer to this pattern than to the patterns we discussed above. This supports our hypothesis that to some extent, the results of our approach are influenced by the degree distribution of the graph, which is naturally reflected in the distribution of the clique sizes in the obtained solutions.

Summarizing our investigation, the results suggest that the approach works very well for distributions, which are well approximable by the power law of scale-free networks. Even though, in some cases of such distributions, we were not able to prove optimality, still the obtained results were always very close to the optimum. This can be explained by the fact that networks with high variance in degree distribution tend to have high variance also in the distribution of the clique sizes for the obtained solutions. Such solutions seem to be obtainable by IG more easily than in the cases of synthetic graphs, where the clique sizes are distributed within a tight interval. In other words, the real-world graphs naturally tend to have an "asymmetric" structure, for which our approach seems to be suitable.

5 CONCLUSIONS AND DISCUSSION

We presented a heuristic approach to the (vertex) clique covering problem (CCP), i.e. covering the vertices of a graph with as few disjoint cliques as possible. In our approach we used an iterated greedy (IG) heuristic for the construction of a solution

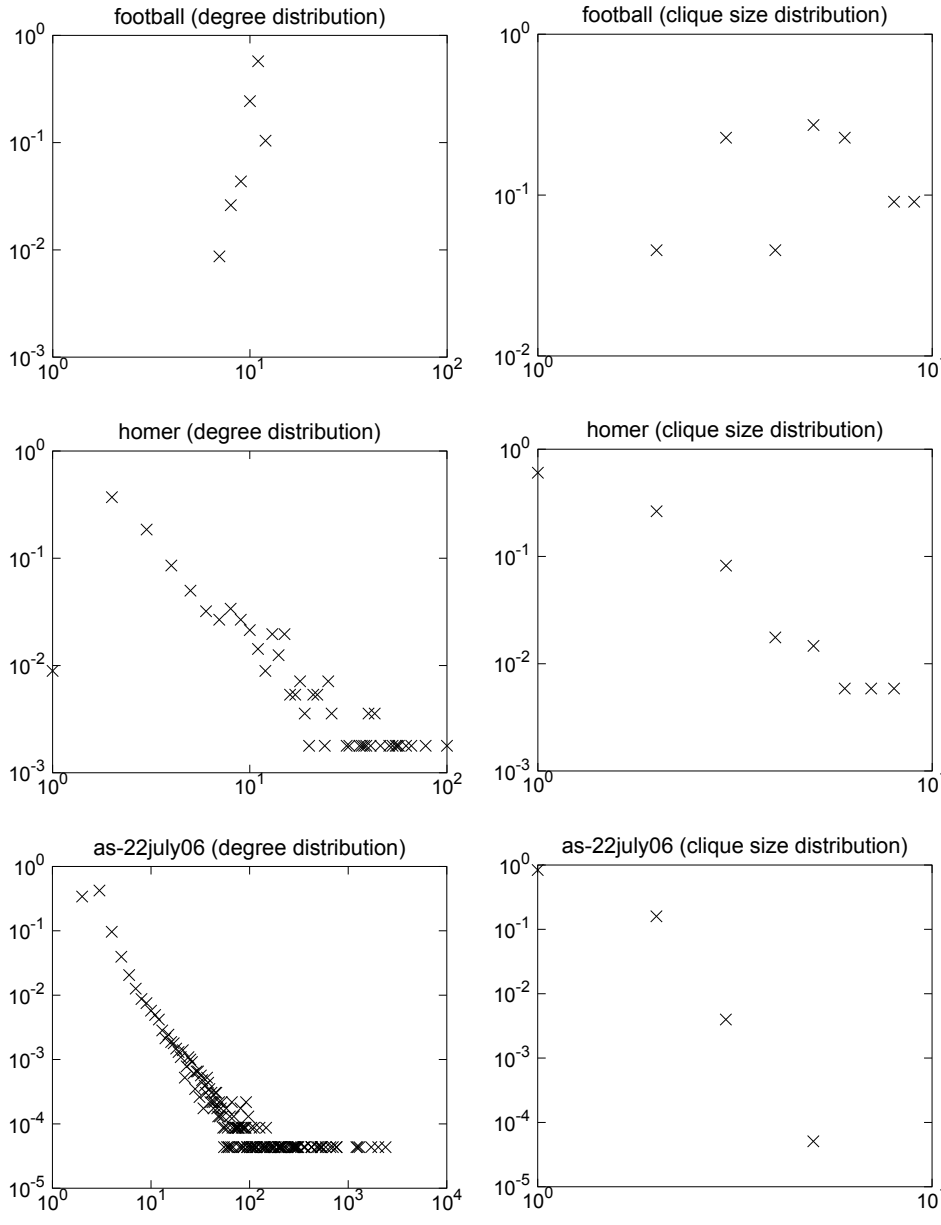

Figure 2. The visualization of degree and clique size distributions for chosen real-world network test instances and the obtained solutions in log log scale (part II)

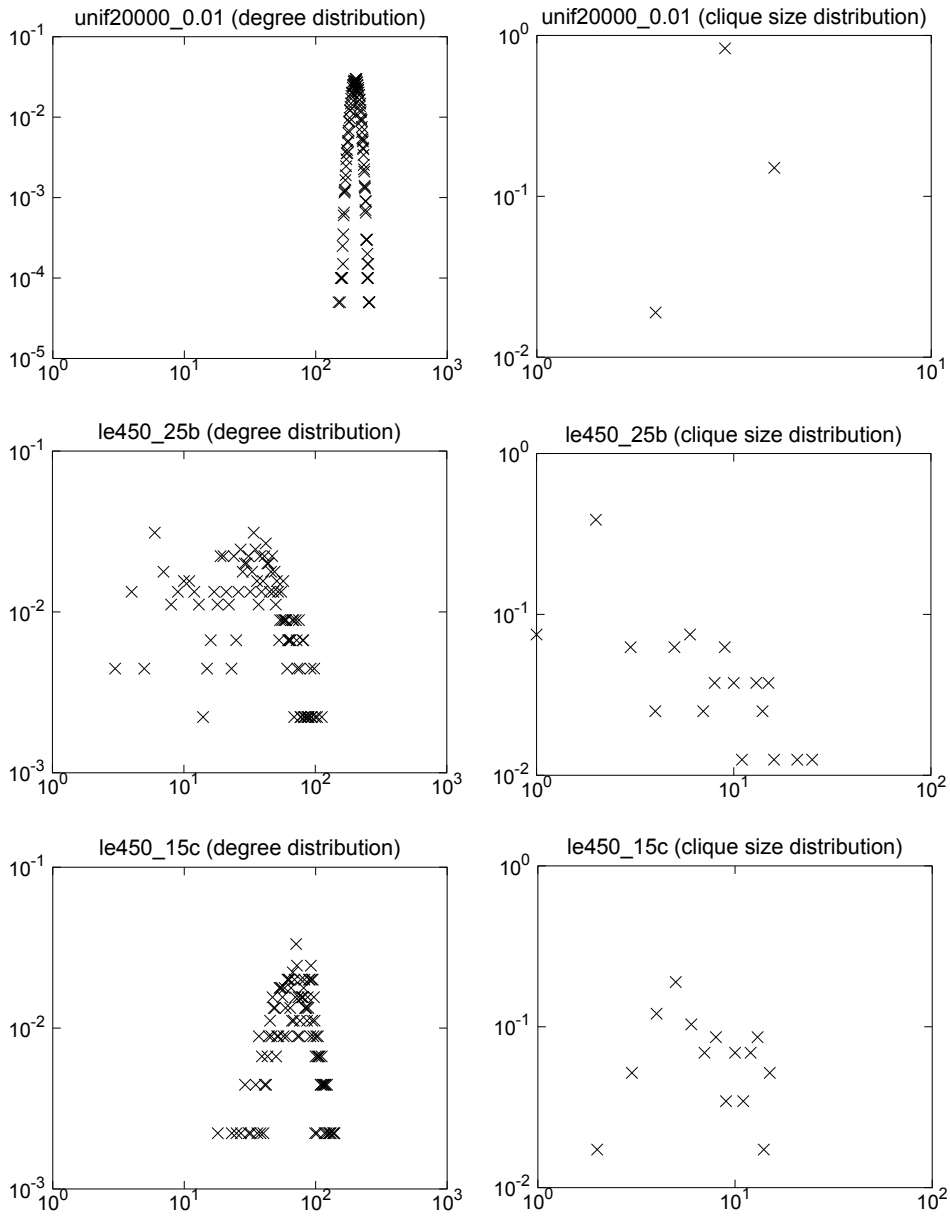

Figure 3. The visualization of degree and clique size distributions for chosen synthetic test instances in log log scale

to CCP and a permutation-based randomized local search algorithm for maximum independent set (RLS^1_p) to estimate a lower bound. The lower bound allowed us to determine the quality of the obtained solutions. Experimental verification was presented on a set of 17 complex networks, including web-based social network extracts and several instances, which are well-known in network science and discrete applied mathematics literature. In 13 out of the 17 networks we obtained an optimal clique covering, while in the other 4 instances we obtained a narrow interval for the optimum. To confront this result we also presented results on synthetic Erdős-Rényi graphs and Leighton graphs, where the algorithm was less successful. Analysis of the degree distributions of the studied networks and distributions of the clique sizes for the obtained solutions also suggests that our approach is well-suited for typical complex networks with high variance in degree distributions. This includes especially the distributions, which are well approximable by the power law of scale-free networks.

An interesting question, which remains open, is whether there is a more exhaustive algorithm, which could be used to further narrow the obtained interval for the optimum in cases, when our approach is not able to prove the optimality. Search algorithms, which use a fixed number of cliques, could possibly be used but our preliminary experiments have not shown much improvement so far. In addition, the observed efficiency could possibly be studied analytically. However, this would require investigation of the behavior on a selected model, e.g. on graphs for which the degree distribution follows the power law. Such results would not be entirely generalizable for real-world application, but could probably offer some insight into the behavior of heuristic algorithms on such graphs.

Acknowledgement

The author would like to thank Jiří Pospíchal and the anonymous referees for their very helpful comments on this work. This contribution was supported by Grant Agency VEGA SR under the Grant 1/0553/12.

REFERENCES

[1] Aggarwal, C. C.—Wang, H.: Managing and Mining Graph Data. Springer, Berlin/Heidelberg, 2010.

[2] Andrade, D. V.—Resende, M. G. C.—Werneck, R. F.: Fast Local Search for the Maximum Independent Set Problem. Journal of Heuristics, Vol. 18, 2012, No. 4, pp. 525–547.

[3] Behrisch, M.—Taraz, A.: Efficiently Covering Complex Networks with Cliques of Similar Vertices. Theoretical Computer Science, Vol. 355, 2006, No. 1, pp. 37–47.

[4] Bollobás, B.: The Chromatic Number of Random Graphs. Combinatorica, Vol. 8, 1988, No. 1, pp. 49–55.

[5] Brélaz, D.: New Methods to Color Vertices of a Graph. Communications of the ACM, Vol. 22, 1979, No. 4, pp. 251–256.

[6] Budinich, M.: Exact Bounds on the Order of the Maximum Clique of a Graph. Discrete Applied Mathematics, Vol. 127, 2003, No. 3, pp. 535–543.

[7] Chakrabarti, D.—Faloutsos, C.: Graph Mining: Laws, Generators, and Algorithms. ACM Computing Surveys, Vol. 38, 2006, No. 1, Art. No. 2.

[8] Chalupa, D.: On the Efficiency of an Order-Based Representation in the Clique Covering Problem. In: Moore, J., Soule, T. (Eds.): Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation (GECCO 2012), Philadelphia, PA, July 2012, pp. 353–360.

[9] Culberson, J. C.—Luo, F.: Exploring the k-Colorable Landscape with Iterated Greedy. In: Johnson, D. S., Trick, M. (Eds.): Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 26, 1995, pp. 245–284.

[10] Czech, W.—Dzwinel, W.—Goryczka, S.—Arodz, T.—Dudek, A. Z.: Exploring Complex Networks with Graph Investigator Research Application. Computing and Informatics, Vol. 30, 2011, No. 2, pp. 381–410.

[11] Drechsler, N.—Günther, W.—Drechsler, R.: Efficient Graph Coloring by Evolutionary Algorithms. In: Reusch, B. (Ed.): Computational Intelligence, Theory and Applications International Conference Proceedings, 6th Fuzzy Days, 1999, pp. 30–39.

[12] Feo, T. A.—Resende, M. G. C.—Smith, S. H.: A Greedy Randomized Adaptive Search Procedure for Maximum Independent Set. Operations Research, Vol. 42, 1994, No. 5, pp. 860–878.

[13] Flake, G. W.—Lawrence, S.—Giles, C. L.: Efficient Identification of Web Communities. In: Ramakrishnan, R., Stolfo, S., Bayardo, R., Parsa, I. (Eds.): Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2000), Boston, MA, August 2000, pp. 150–160.

[14] Galinier, P.—Hertz, A.: A Survey of Local Search Methods for Graph Coloring. Computers and Operations Research, Vol. 33, 2006, No. 9, pp. 2547–2562.

[15] García, S.—Quintana, D.—Galván, I. M.—Isasi, P.: Multiobjective Algorithms with Resampling for Portfolio Optimization. Computing and Informatics, Vol. 32, 2013, No. 4, pp. 777–796.

[16] Gao, L.—Sun, P.—Song, J.: Clustering Algorithms for Detecting Functional Modules in Protein Interaction Networks. Journal of Bioinformatics and Computational Biology, Vol. 7, 2009, No. 1, pp. 217–242.

[17] Girvan, M.—Newman, M. E. J.: Community Structure in Social and Biological Networks. Proceedings of the National Academy of Sciences, Vol. 99, 2002, No. 12, pp. 7821–7826.

[18] Johnson, D. S.—Trick, M.: Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 26, American Mathematical Society, 1996.

[19] Karp, R. M.: Reducibility Among Combinatorial Problems. In: Miller, R. E., Thatcher, J. W. (Eds.): Proceedings of a Symposium on Complexity of Computer Computations, 1972, pp. 85–103.

[20] Knuth, D. E.: The Stanford GraphBase: A Platform for Combinatorial Computing. Addison-Wesley, Reading, MA, 1993.

[21] Krivelevich, M.—Sudakov, B.: Coloring Random Graphs. Information Processing Letters, Vol. 67, 1998, No. 2, pp. 71–74.

[22] Leighton, F. T.: A Graph Coloring Algorithm for Large Scheduling Problems. Journal of Research of the National Bureau of Standards, Vol. 84, 1979, No. 6, pp. 489–503.

[23] Leskovec, J.—Lang, K. J.—Mahoney, M. W.: Empirical Comparison of Algorithms for Network Community Detection. In: Rappa, M., Jones, P., Freire, J., Chakrabarti, S. (Eds.): Proceedings of the 19th International Conference on World Wide Web, WWW 2010, pp. 631–640.

[24] Milanović, M.: Solving the Generalized Vertex Cover Problem by Genetic Algorithm. Computing and Informatics, Vol. 29, 2010, No. 6+, pp. 1251–1265.

[25] Náther, P.—Markošová, M.: Positional Word Web and Its Numerical and Analytical Studies. Computing and Informatics, Vol. 30, 2011, No. 6, pp. 1287–1302.

[26] Nehéz, M.—Olejár, D.—Demetrian, M.: A Detailed Study of the Dominating Cliques Phase Transition in Random Graphs. In: Agrawal, M., Cooper, S. B., Li, A. (Eds.): Theory and Applications of Models of Computation, 9th Annual Conference (TAMC 2012), LNCS, Vol. 7287, 2012, pp. 594–603.

[27] Neumann, F.—Witt, C.: Bioinspired Computation in Combinatorial Optimization: Algorithms and Their Computational Complexity. Natural Computing Series, Springer, 2010.

[28] Newman, M. E. J.: Finding Community Structure in Networks Using the Eigenvectors of Matrices. Physical Review E, Vol. 74, 2006, No. 3, Art. No. 036104.

[29] Pattillo, J.—Youssef, N.—Butenko, S.: Clique Relaxation Models in Social Network Analysis. In: Thai, M. T., Pardalos, P. M. (Eds.): Handbook of Optimization in Complex Networks, Springer Optimization and Its Applications, Vol. 58, 2012, pp. 143–162.

[30] Rudolf, B.—Markošová, M.—Čajági, M.—Tiňo, P.: Degree Distribution and Scaling in the Connecting-Nearest-Neighbors Model. Physical Review E, Vol. 85, 2012, No. 2, Art. No. 026112.

[31] Schaeffer, S. E.: Graph Clustering. Computer Science Review, Vol. 1, 2007, No. 1, pp. 27–64.

[32] Smith, D. H.—Hurley, S.—Thiel, S. U.: Improving Heuristics for the Frequency Assignment Problem. European Journal of Operational Research, Vol. 107, 1998, No. 1, pp. 76–86.

[33] Šíma, J.—Schaeffer, S. E.: On the NP-Completeness of Some Graph Cluster Measures. In: Wiedermann, J., Tel, G., Pokorný, J., Bieliková, M., Štuller, J. (Eds.): Proceedings of the 32nd International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2006), LNCS, Vol. 3831, 2006, pp. 530–537.

[34] Vivekanandan, P.—Rajalakshmi, M.—Nedunchezhian, R.: An Intelligent Genetic Algorithm for Mining Classification Rules in Large Datasets. Computing and Informatics, Vol. 32, 2013, No. 1, pp. 1–22.

[35] Welsh, D. J. A.—Powell, M. B.: An Upper Bound for the Chromatic Number of a Graph and Its Application to Timetabling Problems. The Computer Journal, Vol. 10, 1967, No. 1, pp. 85–86.

[36] Zachary, W. W.: An Information Flow Model for Conflict and Fission in Small Groups. Journal of Anthropological Research, Vol. 33, 1977, No. 4, pp. 452–473.

David Chalupa received his Master's degree in software engineering from Slovak University of Technology, Faculty of Informatics and Information Technologies in 2011. Currently, he is a Ph.D. candidate at the Institute of Applied Informatics at the same university. His research interests include heuristic algorithms, evolutionary computation and social networks.