Mining k-Reachable Sets in Real-world Networks Using Domination in Shortcut Graphs

David Chalupa[a], Christian Blum[b]

[a] *Computational Science Research Group (CSRG)*
School of Engineering and Computer Science
University of Hull
Cottingham Road
Hull, HU6 7RX, United Kingdom
Email: d.chalupa@hull.ac.uk
[b] *Artificial Intelligence Research Institute (IIIA-CSIC)*
Campus of the UAB
08193 Bellaterra, Spain
Email: christian.blum@iiia.csic.es

Abstract

We propose a technique for mining minimum sets with bounded reachability in real-world networks i.e. the smallest vertex sets such that any other vertex is at distance at most k from at least one vertex of this set. Our technique uses a simple but efficient mechanism. We first introduce new edges to shorten the paths in our network to obtain a shortcut graph. As the next step, we search for the minimum dominating set in the shortcut graph. For this purpose, a variety of algorithmic approaches is applied and computationally studied. To the best of our knowledge, this approach and the impact of shortcut graph structure on the problem have not been systematically explored yet. Experimental results are presented for local samples of two social networks, as well as large network science graphs, including several research collaboration networks. Different profiles of k-reachable sets were found for different networks. We find that k-reachable set sizes sharply decline with increasing k and decreasing graph diameters in the context of shortcut graphs obtained from social networks, as well as the largest connected components of research collaboration networks. However, there are also real-world networks, which seem to have slightly different profiles.

Keywords:
k-reachable sets, minimum dominating set, real-world networks, shortcut graphs, graph mining

1. Introduction

Complex networks are networks with non-trivial structure, which occur in many real-world domains, including social networks [1], research collaboration networks [2], protein-protein interaction networks [3] or the World Wide Web [4]. The well known theory of the "six degrees of separation" is perhaps the most widely known example of a reference to the *small-world properties* of real-world complex networks [5]. This popular theory states that the shortest path between two arbitrary vertices of a large social network consists of at most six edge traversals on average.

Dominating sets represent a highly relevant research topic in the context of complex networks. A dominating set is a subset $S \subseteq V$ of vertices of an undirected graph $G = (V, E)$ such that $\forall v \in V$ it holds that $v \in S$ or $\exists w \in V$ with $\{v, w\} \in E$ and $w \in S$, i.e. each vertex is in the dominating set or has a neighbour in the dominating set. The cardinality of a minimum dominating set is often called *domination number* and is denoted by γ. The minimum dominating set problem (MDS) is one of the classical NP-hard combinatorial optimisation problems [6], with a close relation to distances in real-world networks and their small-world properties.

Distance-based perspective on dominating sets. Let $G = (V, E)$ be an undirected graph and let $d_G(v, w)$ be the shortest path length between two vertices $v, w \in V$ in G. Then, a set $S \subseteq V$ is a dominating set if $\forall v \in V\ \exists w \in S\ (d_G(v, w) \leq 1)$.

This distance-based perspective on MDS leads to a natural generalisation of the problem. From this point of view, a dominating set is a set of vertices such that each vertex is at distance at most 1 to some vertex of the dominating set. Therefore, it represents a set of "central" vertices of the graph. Suppose that one substitutes the upper bound for the distance by an integer parameter $k \geq 0$. For $k = 0$, we have that $S = V$. For $k = 1$, we obtain the original problem of searching for MDS. However, one may consider $k \geq 2$, for which smaller sets of "central" vertices in a larger bounded distance are identified.

There are numerous application domains for this generalised perspective on MDS. Some of them include *identification of hubs* [7], fast *network distance estimation* [8], or exploration of *stability and vulnerability* of real-world networks [9] such as *water distribution networks* [10] and *electrical power networks* [11]. Other related topics are represented by the modelling of *attacks and visualisation in cyber security* [12], modelling of *rumour propagation* [13], *detection of missing edges* in social networks [14] or the use of *clustering metrics in graph-based modelling* [15].

To formalise this generalised perspective on MDS, we define the term shortcut graph, which models the problem by adding shortcuts between more distant vertices. To the best of our knowledge, the idea of shortcut graphs and k-reachability has not been explored before, even though it seems very intuitive on the surface.

Shortcut graphs. Let $G = (V, E)$ be an undirected graph. The shortcut graph G_k of order $k \geq 0$ is a graph $G_k = (V_k, E_k)$ such that $V_k = V$ and $E_k = \{\{v, w\} : \ d_G(v, w) \leq k, v \neq w\}$.

For $k = 0$, we obtain that G_k is a set of isolated vertices and for $k = 1$, we have $G_k = G$. Note that for $k \geq 2$, one obtains a densified graph with all triplets of vertices with a path of length 2 extended with a direct shortcut to a triangle. The ratio of the number of triangles and the number of connected triplets is the *clustering coefficient* metric [16], which is usually used to measure the cohesion in complex networks. Therefore, one can naturally expect the shortcut graphs to have a more pronounced clustered structure. An analogical process of shortcut insertion is applied for higher values of k, leading to non-increasing sizes of small dominating sets for growing values of k.

Shortcut graphs and minimum dominating sets are closely related to the concepts of clusters [17], communities [18], and their detection [19, 20, 21]. Dominating sets represent groups of "central" vertices, which decompose the network into clusters such that each vertex is in a distance to a dominating set vertex, which is upper bounded by 1. The use of shortcut graphs generalises this distance to any fixed value $k \geq 0$. We will refer to a dominating set in a shortcut graph G_k as a *k-reachable set*. The concept of k-reachable sets is closely related to the hierarchy of communities [22]. Since different values of k lead to decompositions with different maximum distances to k-reachable set vertices, the resulting clusters have increasingly coarse granularities with

increasing k. A similar concept is represented by k-medoid clustering, in which the aim is to find a fixed set of k vertices such that distances to medoids in the clusters are minimised [23, 24].

It is worth noting that the traditional perspective on MDS concerns several other real-world applications. Additional constraints are sometimes added to model a specific real-world situation. The most widely known applications include *routing in wireless ad-hoc networks* [25, 26, 27] using connected dominating sets [28], *multi-document summarisation* [29] or *positive influence dominating sets in social networks* [30, 31].

Contributions. In this paper, we explore the k-reachability in real-world networks using shortcut graphs and several approaches to solve the minimum dominating set problem.

From the *network mining perspective*, we explore the non-increasing profile of k-reachable sets with growing k. Critical values of k are identified for several networks of small to medium sizes. For such values of k, relatively small k-reachable sets with a single-digit size can be found.

The networks studied in this paper include data from two different social network services, as well as instances from Newman's network data repository. This data set covers networks from a wide range of application areas, including research collaboration networks, a snapshot of the Internet or a power grid.

From the *algorithmic perspective*, we discover that the integer linear programming (ILP) formulation of the problem can be solved relatively efficiently by ILP solvers for graphs with thousands of vertices. We use the open-source ILP solver CBC from the COIN-OR package. It is well-known that ILP branch-and-cut-solvers such as CBC or CPLEX are quite efficient in solving MDS for sparse graphs up to a relatively high number of vertices [32]. However, memory demands of ILP solvers seem to grow quickly for large and dense shortcut graphs. Therefore, experimental results are also presented for a greedy approximation algorithm, the ant-based algorithm ACO-LS-S, as well as the order-based randomised local search algorithm RLS_o. These results indicate that while RLS_o provides results of good quality for most of the graphs, ACO-LS-S seems to be useful for shortcut graphs of large social networks. For some of the instances, RLS_o provided the best result, because of excessive memory demands of the ILP solver.

Our results indicate similar profiles of dominating set sizes for networks from the two social network services mentioned above, with a sharp decline

of the dominating set size for growing k, and with a single dominating vertex for $k = 5$. A very similar result is obtained for the Internet snapshot, with the minimum value being reached for $k = 6$. For large research collaboration networks, the patterns of decrease in dominating set size are more moderate, with the minimum values reached for $8 \leq k \leq 11$. The decline pattern becomes sharper when only the largest connected component of these networks is taken into account. However, this has no impact on the critical values identified. For a power grid network, we obtain that its structure is much more resilient from the perspective of k-reachability, with a gradual decrease in dominating set size, and a single dominating vertex obtained for $k = 23$.

The paper is structured as follows. In Section 2, we review the algorithms used to search for small dominating sets, with a particular focus on methods suitable for shortcut graphs obtained from real-world networks. In Section 3, we explore the concept of shortcut graphs and the impact of shortcuts on network structure. In Section 4, we present the experimental results for different real-world networks and values of k. The impact of increasing k on the size of dominating sets in the real-world networks is also studied. Finally, Section 5 presents a discussion and conclusions of this research.

2. Searching for Small Dominating Sets in Shortcut Graphs

MDS is an NP-hard problem [6]. The most efficient exact algorithm requires $\mathcal{O}(1.5137^n)$ time to solve MDS [33], which is intractable for large graphs. However, ILP formulations of the MDS problem and its variants seem to lead to surprisingly well scalable exact approaches to solve these problems [32].

For very large scale instances, approximation algorithms or heuristics represent practical solving techniques. Generally, the best approximation ratio achievable in polynomial time is $\mathcal{O}(\log \Delta)$, where Δ is the maximum degree of a vertex in the graph. Sublogarithmic approximation seems to remain hard [34].

Additionally, the MDS problem remains NP-hard for restricted graph classes. NP-hardness of the MDS problem, or hardness of its approximation, have been proven for *unit disk graphs*, which represent a model of wireless networks [35], *grids* [36], *bounded degree graphs* [37], and *power law graphs* [38], which occur in a wide range of real-world applications [4].

As we have outlined above, the MDS problem is relatively straightforward to formulate as an ILP. For the purpose of this paper, we use the ILP model

outlined below. The ILP models derived from the considered graphs will then be solved using the CBC branch-and-cut ILP solver. This solver is available as a part of COIN-OR, which is a popular open-source mixed ILP solving software package [39, 40].

In the following, we formulate the MDS problem as an ILP problem, and shortly describe the three main heuristic approaches that are employed in this paper to find small dominating sets in shortcut graphs.

ILP model of the MDS problem. The standard ILP model for the MDS problem makes use of a binary variable $x_i \in \{0, 1\}$ for each vertex $v_i \in V$. If, after solving the problem with an ILP solver, $x_i = 1$ this means that v_i forms part of generated optimal solution. Otherwise, when $x_i = 0$, v_i does not form part of that solution. Then, the MDS can be formulated in terms of the following ILP model:

$$\min \sum_{i=1}^{n} x_i \tag{1}$$

$$\text{s.t.} \quad x_i + \sum_{j;\{v_i,v_j\}\in E} x_j \geq 1 \quad \forall i = 1, 2, ..., n. \tag{2}$$

In the section on the experimental results, it will be seen that this formulation leads to a surprisingly well scalable approach to solve the problem. However, its scalability decreases not only with increasing n but also with increasing k i.e. the density of the graph also plays a role in the efficiency of the ILP approach.

Greedy approximation algorithm for MDS. This algorithm is derived from the approximation algorithm for set cover and achieves an approximation ratio of $H(\Delta) = \sum_{i=1}^{\Delta} 1/i = \mathcal{O}(\log \Delta)$ [41].

It starts with an empty dominating set $S = \emptyset$. At each step, given the current partial dominating set S, a vertex $v \in V$ is labelled non-dominated if $v \notin S$ and for all $v' \in V$ (with $v \neq v'$) it holds that $\{v, v'\} \in E \Rightarrow v' \notin S$ i.e. vertex v is not in the dominating set and has no neighbour in the dominating set. Let $w(v, S)$ denote the number of non-dominated vertices in $\{v\} \cup \{v' \in V : \{v, v'\} \in E\}$ i.e. among the neighbours of v and v itself. At the current construction step, the greedy approximation algorithm takes the vertex with the highest value of $w(v, S)$ and puts it into S. The algorithm stops when S is a dominating set.

This approach has the advantage of being fast and producing the logarithmic approximation, which seems to be the best approximation obtainable for general graphs. Several variants of this algorithm have been experimentally studied [42]. On the other hand, previous studies have shown that dominating sets empirically obtained by this algorithm tend to be larger than results produced by hybrid and randomised search algorithms [43, 44].

Hybrid algorithms for MDS. Hybrid algorithms for MDS include both techniques based on genetic algorithms and ant colony optimisation. Hybrid genetic algorithms combine crossover operators with local search subroutines [44, 45]. Very successful approaches are *hybrid ant colony optimisation algorithms*, which combine an ant-based construction heuristic with local search subroutines. In this paper, we will make use of an existing ant colony optimisation algorithm—known as ACO-LS—that has been applied to the classical MDS problem [44]. There is also a variant of this algorithm using preprocessing, applied to the minimum weight dominating set problem [46].

ACO-LS is a hybrid ant-based heuristic, which works as follows. The construction graph for ACO-LS is a complete graph, in which vertices correspond to vertices of the original graph. Pheromone is placed on vertices of the construction graph, determining the probability of occurrence of the vertex in the dominating set. In the beginning, an initial amount of pheromone is placed on each vertex. A dominating set is then constructed by adding vertices one by one. The probability for each vertex to be added is proportional to the amount of pheromone on it. The construction terminates once the constructed set is a dominating set.

At each iteration of ACO-LS, n_s dominating sets are constructed. The smallest of these dominating sets is taken and improved using local search. This local search subroutine iteratively excludes redundant vertices from the dominating set. A vertex v is redundant in dominating set S iff $S \backslash \{v\}$ is also a dominating set. Two strategies in local search are used probabilistically. The first strategy is chosen with probability p_r and excludes redundant vertices in a random order, while the second strategy eliminates redundant vertices in increasing order of their degree. The improved dominating set is then used to reinforce the pheromone on vertices, which are in the dominating set. The new pheromone value for a vertex in this dominating set is $\rho\tau + \frac{1}{10+f-F}$, where τ is the original pheromone value, ρ is the pheromone evaporation rate, f is the cardinality of the best dominating set in the current iteration of ACO-LS, and F is the cardinality of the smallest dominating set found so

far. For the other vertices, the new pheromone value will simply be $\rho\tau$ [44].

The literature also offers a modified version of ACO-LS, known as ACO-LS-S [43]. ACO-LS-S differs in two aspects from ACO-LS. Firstly, the vertices from the solution obtained by the greedy approximation algorithm obtain a higher initial pheromone value. Secondly, ACO-LS-S operates on the original graph instead of a complete construction graph. Therefore, once a vertex is added to the dominating set, the construction can only proceed with its neighbours. This helps to provide an approach, which is more scalable to large graphs [43]. For the purpose of this paper, we use ACO-LS-S, since it is much more scalable to the sizes of graphs we explore.

Order-based algorithm for MDS. The order-based randomised local search algorithm (RLS_o) for MDS is a technique, which also combines a greedy construction procedure with a local search mechanism. However, RLS_o uses the local search to optimise the input to the greedy construction procedure. In the following, we review the description of RLS_o from our previous work [43].

In each iteration, RLS_o uses a greedy algorithm that maps a permutation of vertices π to a dominating set S. The mapping algorithm starts with an empty dominating set S. This is followed by an iterative procedure. In each iteration, the current vertex v is taken from a fixed permutation of vertices π. This is followed by a check whether v is non-dominated or some neighbour of it is non-dominated. If yes, v is added to S and v is set as a dominated vertex. After the addition of v to S, all neighbours of v are set to dominated, too.

RLS_o starts by using the greedy approximation algorithm to construct the initial dominating set S. The vertices of S are then used to construct an initial permutation π. This is accomplished by placing the vertices from S at the first positions of π, while the remaining vertices from $V \backslash S$ after placed after then vertices from S, in a random order. This is followed by the local search. In each step of the local search, RLS_o performs the *jump* operator on a vertex in π chosen uniformly at random.

Operator $jump(i, P)$ takes the vertex at position i in permutation π and puts it to the first position in the permutation. The other vertices are then shifted to the right i.e. vertices formerly in positions $1, 2, ..., i-1$ are moved to positions $2, 3, ..., i$. The resulting permutation π' is returned.

A new dominating set S' is then constructed using the modified permutation π'. It is then checked whether S' is a dominating set with at most

as many vertices as S. If this is true, π' and S' are accepted as the new permutation π and the new dominating set S.

Lower bounds. There are several different lower bounds for MDS. Trivial lower bounds simply use the number of connected components c, the maximum degree Δ, and the approximation properties of the greedy algorithm.

Let γ be the domination number of a graph G i.e. the size of a minimum dominating set of G. Note that each connected component of G has to be dominated by at least one vertex i.e. $c \leq \gamma$. Another lower bound is implied by the fact that each vertex can dominate at most $\Delta + 1$ other vertices. Therefore, $n/(\Delta + 1) \leq \gamma$.

Another lower bound is implied by the logarithmic approximation guarantee of the greedy algorithm. The approximation is at most $H(\Delta) \leq \ln(\Delta)+1$ times larger than a minimum dominating set. Let γ_{gm} be the maximum size of a dominating set obtained in repeated runs of the greedy approximation algorithm. Then, $\gamma_{gm}/(\ln(\Delta) + 1) \leq \gamma$.

The last lower bound follows from the ILP formulation of MDS. By dropping the integrality constraints of the binary variables, one obtains the continuous LP relaxation of the problem, whose solution results in a very good lower bound.

3. Shortcut Graphs and k-Reachability in Real-world Networks

The methods introduced in the previous section can be used to construct optimal or near-optimal solutions to MDS and provide bounds for the optimum. In our previous study, the abilities of the greedy approximation algorithm, ACO-LS-S, and RLS_o to produce small dominating sets for real-world networks were explored [43].

In this paper, we are facing a slightly different point of view on MDS than in the previous works. Even though the same algorithms are available to explore dominating sets in shortcut graphs, it is the actual structure of the shortcut graphs, which influences the properties of MDS. As we have indicated above, the choice of the right algorithms to solve MDS depends on the graph structure. In the following, we will face the problem of solving the MDS problem in gradually densified shortcut graphs with up to tens of thousands of vertices.

Shortcut graphs naturally shorten the paths between vertices and can shorten the diameter of the network. The impact of such a shortening is illustrated in Figure 1, in which we depict the largest connected component of

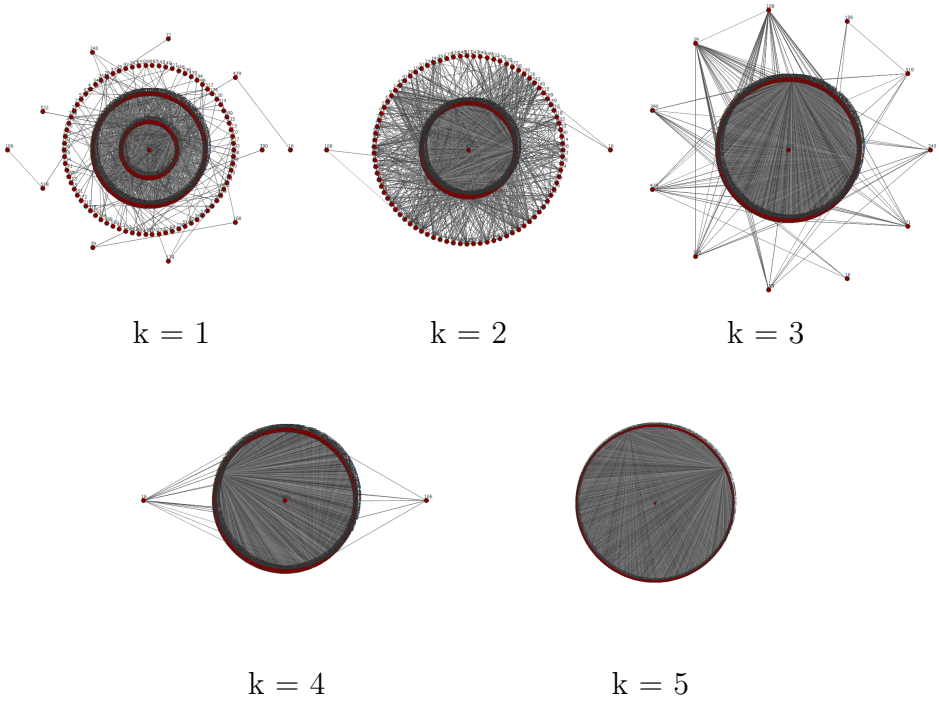

Figure 1: Illustration of shortcut graphs for the largest connected component of network *homer*.

a coapperance network for Illiad and Odyssey by Homer from the DIMACS graphs [47] and its corresponding shortcut graphs. Network *homer* was chosen for this illustration, since it combines most of the properties we aim to explore, including typical distance shortening in shortcut graphs, as well as a degree distribution, which seems to be well approximable by the power law.

In all of these drawings, the vertex with maximum degree is put in the centre and other vertices are grouped into layers, based on the shortest path length between the central vertex and vertices in the particular layer. With growing k, one can see the shortened paths in the shortcut graphs in the largest connected component by gradual reduction of the number of layers in the drawings.

This is essentially the phenomenon we aim to explore in this paper. Shortcut graphs of order k represent instances, which model the bounded "degree of separation" as a neighbourhood of our vertex explicitly. In the context of social networks, this leads to a model, in which we are able to explore the

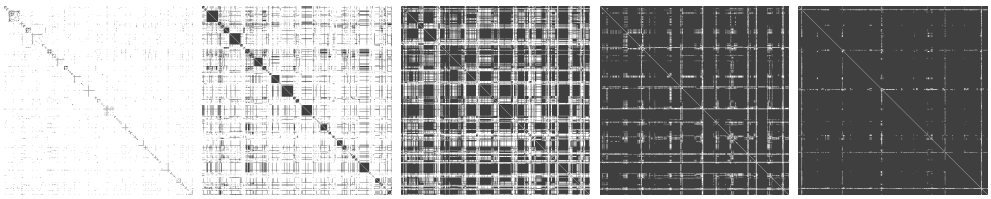

Figure 2: Visualisations of adjacency matrices for shortcut graphs constructed for the largest connected component of network *homer*.

contacts of our contacts directly and explore the gradual "collapse" of the network as k increases.

Figure 2 illustrates the densification in adjacency matrices of shortcut graphs for the largest connected component of network *homer*. Cells in Figure 2 represent adjacencies of corresponding vertices. The vertices are reordered according to MDS of the network, grouping the vertices into clusters containing a vertex from MDS and its neighbours. A sharp densification effect for increasing k can be observed, while the underlying pattern seems to be preserved. It is worth noting that for $k = 1$, the adjacency matrix is very sparse, leading to noticeable adjacencies mostly near the diagonal. The dominating set vertices can be noticed as intersections of the short lines around the diagonal. For $k = 2$, we obtain a sharp densification mainly within the clusters. An interesting transition seems to occur for $k = 3$, for which many inter-cluster edges seem to be introduced. These patterns are somewhat similar to the patterns observed for other real-world network models, such as the Kronecker graphs [48].

In Figure 3, the degree distributions of shortcut graphs for the largest connected component of *homer* are depicted. These plots illustrate the strong impact of shortcuts on the structural properties of our graphs. As we have indicated above, the structural and quantitative properties have a direct impact on the hardness and approximability of MDS [35, 36, 37, 38], as well as on the suitable choice of algorithms to solve MDS in practice [42, 43, 44].

The degree distribution of *homer* can be approximated relatively well by the power law in form $P(k) \approx k^{-1.498}$. This is illustrated in Figure 3 in the first histogram for $k = 1$. The sizes of the bins in the histograms are 3 for $k = 1$ and 10 for the rest of the plots. The linear model in logarithmic scale is fitted for data with $k \leq 40$, with the long-tail cut off, and is represented by the

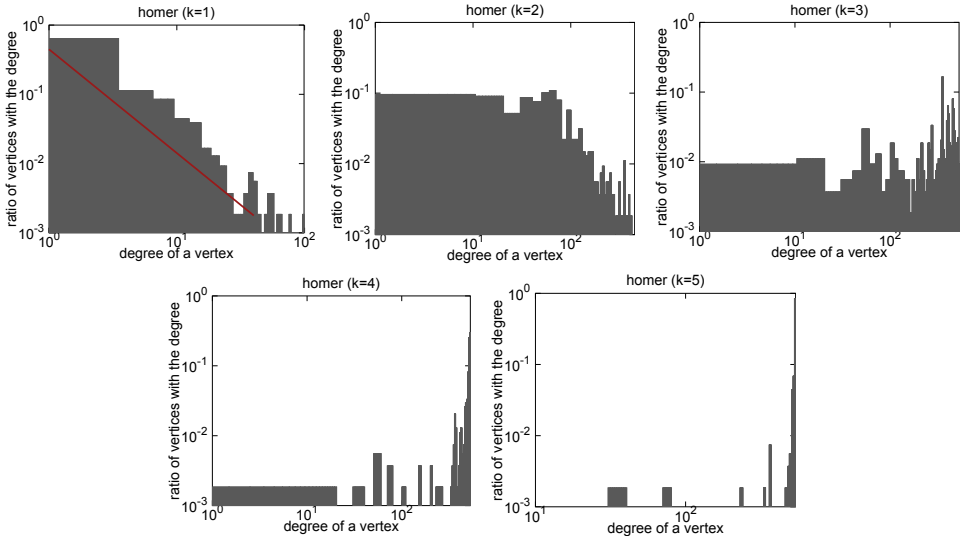

Figure 3: Plots of degree distributions for shortcut graphs constructed for the largest connected component of network *homer*.

bold red line. This produced a model with R^2 value 0.9502, which indicates a fairly representative power-law model of the network. Figure 3 also illustrates that shortcuts completely change the statistical properties of the graph. A pattern of a gradual shift to the long-tail end of the distribution can be observed, which is a natural consequence of the densification. However, it will be interesting to see how well the heuristics for MDS perform on these non-standard topologies, and how good lower bounds can be obtained for these shortcut graphs.

A similar process of shortcut insertion is illustrated for network *dolphins* in Figure 4. This network represents a social network of bottlenose dolphins [49]. It is of interest to us, because—among the smaller graphs—its "collapse" is relatively slow as k increases. Similar phenomena will be observed also for larger graphs in the next section.

4. Experimental Results

In this section, we present the experimental results obtained for MDS in shortcut graphs constructed for different real-world networks. Let us first describe the methodology of our experiments, the parameterisation of the

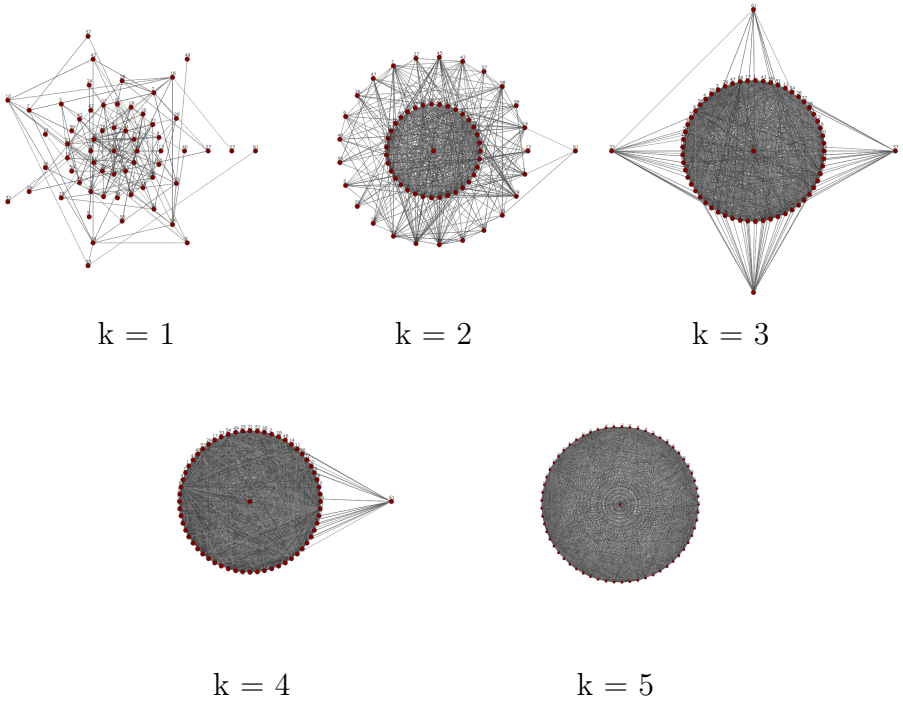

Figure 4: Illustration of shortcut graphs for *dolphins*.

algorithms, as well as the studied metrics.

Methodology of experimental evaluation. We have explored the small dominating sets in shortcut graphs for several real-world networks. As there are multiple approaches available for the problem, we have used the ILP formulation solved by the CBC branch-and-cut solver from the COIN-OR package [39, 40], the greedy approximation algorithm, denoted by GREEDY, as well as the ant-based algorithm ACO-LS-S, and the order-based randomised local search algorithm RLS_o. There are two main aspects we explore for the problem and the algorithms to solve it.

1. *Network aspect.* This aspect is closely related to the properties of MDS in the context of complex network analysis. The aim is to determine how sharply the dominating set cardinality decreases with increasing k. In other words, we explore the cardinality of "critical set" of vertices to identify possible vulnerabilities and understand the stability and security in the networks.

2. *Algorithmic aspect.* The aim is to determine, which of the approaches are suitable for MDS in shortcut graphs. This is also related to the degree distribution and overall structure of shortcut graphs. We explore the impact of increasing k and the change in degree distribution on the algorithmic properties of MDS and the suitable algorithm choice.

We provide the results for increasing values of k, until the dominating set size reaches the number of connected components of the original graph. We will call a value of k *critical*, if it is the lowest value of k such that there is a dominating set of a single-digit size for such a value. Additionally, we will call a value of k *marginal*, if it is the lowest value of k such that the size of the dominating set for this value is equal to the number of connected components. The critical value of k will represent a distance such that it is possible to reach any vertex from a set of less than ten vertices within k edge traversals. In other words, it will represent a value, for which there is a very small set of "central" k-reachable vertices. The marginal value of k will then represent a distance such that it is possible to reach any vertex from a single vertex within k edge traversals.

For networks with multiple connected components, we considered the entire network, as well as the largest connected component only. For each shortcut graph, we will also provide several metrics [50], to better relate the performance of the algorithms and properties of the k-reachability problem to the network structure, size, and density.

We have solved the corresponding ILP problem using CBC with a 10 hour time limit. GREEDY was applied 1000 times to each (shortcut) graph. The result reported for GREEDY is the size of the best one among the 1000 generated solutions. The other algorithms were applied 10 times to each (shortcut) graph. In preliminary experiments, ACO-LS in its original form [44] had a tendency to perform very slowly because of the quadratic complexity of the solution construction process. This is because a complete construction graph is used in this algorithm. This led to results of low quality and we decided to make use of ACO-LS-S, instead. To achieve scalability of the approach to large graphs, only $n_s = 1$ dominating set was constructed per iteration and the probability for the random choice of a vertex in local search was $p_r = 1$. This choice is motived by the fact that a random choice is less time-consuming than choosing a vertex based on its degree. A custom priority queue data structure could potentially be used to provide an efficient implementation of a choice based on the degree. However, our preliminary

efforts indicated that this does not seem to lead to a significant improvement of the results. Finally, the pheromone evaporation rate was $\rho = 0.985$. For vertices used in the initial dominating set constructed by GREEDY, the initial pheromone value was set to 1000.0, while the initial pheromone value for the other vertices was 10.0.

Each run of ACO-LS-S and RLS_o was either stopped after 60 minutes or when no improved solution was found during the last 10^6 iterations. A constant number of iterations without improvement was chosen as a practically manageable stopping criterion. The experimental software was written in C++ using Qt, compiled with 64-bit Visual Studio 2013 compiler. A 64-bit platform was chosen because of the high memory demand of the approaches, resulting from some of the large networks with large k. As the ILP solver, we used CBC from COIN-OR 64-bit binary Windows package compiled with Intel 11.1 compiler. All experiments were carried out on a machine with Intel Core i7-5960X 3 GHz CPU with 64 GB RAM.

Results for social networks. The first set of results is presented for data obtained from social networks *Google+* and Pokec. The Google+ networks are constructed from the public information on circles, obtained by a web crawler using breadth-first search. Therefore, the social network data samples represent local subgraphs of different sizes. Pokec is a Slovak social network with publicly available information on contacts obtained by a similar web crawler. This social network has previously been studied in large scale [51] and its large snapshot forms part of the Stanford's SNAP network data set[1].

Table 1 summarises these results. The first column contains the name of the graph. The second column presents the value of k used. The next columns contain the metrics computed for the corresponding (shortcut) graphs. This includes the number of vertices n, the number of edges m, the average local clustering coefficient $\overline{C}$, the density δ, the number of connected components c, as well as the diameter d. It is worth noting that computing $\overline{C}$ is extremely time-consuming for large and dense graphs, which is why some of these values are omitted. This is followed by the domination number γ or its lower bound computed for the particular instance. If the proven optimum was reached by CBC, then an equal sign is used in this column. The lower bound sign $\leq$ means that only a lower bound has been found by CBC, while symbol "-" means that the memory limit has been exceeded an no solution has been

[1] `http://snap.stanford.edu/data/`

Table 1: Experimental results of the studied algorithms for samples from social networks *Google+* and Pokec.

graph	k	n	m	$\overline{C}$	δ	c	d	γ	CBC	GRE	ACO-LS-S	RLS$_o$
Samples from *Google+**												
gplus_500	1	500	1006	0.32	0.008	1	6	42	**42**	**42**	**42**	**42**
	2		6644	0.823	0.053		3	6	**6**	**6**	**6**	**6**
	3		21717	0.765	0.174		2	1	**1**	**1**	**1**	**1**
gplus_2000	1	2000	5343	0.25	0.003	1	7	170	**170**	175	**170**	**170**
	2		52257	0.713	0.026		4	15	**15**	16	**15**	**15**
	3		235357	0.647	0.118		3	2	**2**	**2**	**2**	**2**
	4		738687	0.730	0.37		2	1	**1**	**1**	**1**	**1**
gplus_10000	1	10000	33954	0.218	0.001	1	8	861	**861**	891	862	**861**
	2		588469	0.629	0.012		4	82	**82**	89	**82**	**82**
	3		4.3×10^6	0.585	0.087		3	9	**9**	11	**9**	**9**
	4		1.7×10^7	0.746	0.33		2	1	**1**	**1**	**1**	**1**
gplus_20000	1	20000	81352	0.192	<0.001	1	9	1716	**1716**	1799	1727	1717
	2		2×10^6	0.598	0.01		5	159	**159**	176	160	163
	3		1.8×10^7	0.572	0.089		3	19	**19**	23	**19**	**19**
	4		7.3×10^7	0.777	0.363		3	2	**2**	3	**2**	**2**
	5		1.4×10^8	0.874	0.708		2	1	**1**	**1**	**1**	**1**
gplus_50000	1	50000	2.3×10^5	0.176	<0.001	1	10	4568	**4568**	4815	4639	4586
	2		9.5×10^6	0.569	0.008		5	461	**461**	509	462	483
	3		5.5×10^7	0.552	0.044		3	43	**43**	52	**43**	49
	4		3×10^8		0.242		3	≥ 2	-	10	**6**	**6**
	5		8×10^8		0.639		2	1	-	**1**	**1**	**1**
Samples from *Pokec**												
pokec_500	1	500	993	0.41	0.008	1	4	16	**16**	**16**	**16**	**16**
	2		11608	0.912	0.093		2	1	**1**	**1**	**1**	**1**
pokec_2000	1	2000	5893	0.294	0.003	1	6	75	**75**	**75**	**75**	**75**
	2		68898	0.715	0.034		3	6	**6**	**6**	**6**	**6**
	3		289707	0.682	0.145		2	1	**1**	**1**	**1**	**1**
pokec_10000	1	10000	44745	0.209	0.00089	1	7	413	**413**	**413**	**413**	**413**
	2		683759	0.533	0.014		4	32	**32**	**32**	**32**	**32**
	3		5.3×10^6	0.552	0.106		3	2	**2**	**2**	**2**	**2**
	4		2×10^7	0.731	0.407		2	1	**1**	**1**	**1**	**1**
pokec_20000	1	20000	102826	0.195	<0.001	1	8	921	**921**	922	**921**	**921**
	2		1.7×10^6	0.51	0.009		4	74	**74**	76	**74**	**74**
	3		1.5×10^7	0.522	0.074		3	6	**6**	**6**	**6**	**6**
	4		6.3×10^7	0.672	0.315		2	1	**1**	**1**	**1**	**1**
pokec_50000	1	50000	281726	0.173	<0.001	1	9	2707	**2707**	2761	2771	2716
	2		5.3×10^6	0.450	0.004		5	221	**221**	242	222	234
	3		5.5×10^7	0.417	0.044		3	21	**21**	22	**21**	**21**
	4		3×10^8		0.242		3	2	**2**	**2**	**2**	**2**
	5		8×10^8		0.639		2	1	-	**1**	**1**	**1**

* All of these network samples are publicly available at:

`http://davidchalupa.github.io/research/data/social.html`.

found. The last four columns contain the sizes of the smallest dominating sets constructed by CBC, GREEDY, ACO-LS-S, and RLS_o for each instance.

For most of the instances, CBC was able to find the optimum. It is worth noting that the computational and the memory demands seem to grow quickly with both growing number of vertices, and growing k. This is the reason why CBC was not able to cope with the memory demands and was not able to produce any result for some of the largest instances. While CBC finds the optimum in seconds for the smallest graphs, it took more than 2 hours to find the optimum for *gplus*_50000, $k = 3$, and more than 7 hours for *pokec*_50000, $k = 3$. For many of the large and dense shortcut graphs, CBC requires tens of gigabytes of RAM. It also requires gigabytes of RAM to store the shortcut graphs themselves if heuristic methods are used. Therefore, 64-bit versions of the software need to be used to obtain these results.

The obtained results confirm that GREEDY finds slightly larger dominating sets than ACO-LS-S and RLS_o, especially for smaller values of k. For larger values of k, the differences between algorithms become less significant as the sizes of the dominating sets sharply decrease. Interestingly, ACO-LS-S seems to outperform RLS_o for some of the larger networks with 20000 and 50000 vertices. As the next results indicate, this phenomenon has not been detected for other instances and may be specific to the inherent structural properties of these online social networks. This happens for several shortcut graphs and one can observe that they were constructed for larger samples with mainly $k = 2$ and with $k = 3$ in one instance. The diameters are between 3 and 5, while the values of $\overline{C}$ range from 0.45 to 0.598 for these instances.

For almost all of the instances, the proven optimum was found by either CBC or some of the heuristics. The only exception is the instance (*gplus_50000*, $k = 4$), for which CBC was unsuccessful, presumably due to excessive memory demands. The trivial lower bound is still relatively distant from the dominating set of size 6 found by ACO-LS-S and RLS_o.

The critical values for samples with 10000 vertices are $k = 3$, while for 20000 and 50000 vertices, the critical value rises to $k = 4$. Marginal values of $k = 5$ have been identified for both of the largest samples with 50000 vertices.

Results for large research collaboration networks. These networks are taken from Newman's network data repository. Network *astro-ph* represents collaborations in astrophysics, while networks *cond-mat*, *cond-mat-2003*, and

Table 2: Experimental results of the studied algorithms for network science instances

graph	k	n	m	δ	c	d	γ	CBC	GRE	ACO-LS-S	RLS$_o$
Graphs from Newman's network data repository (research collaboration networks [2])											
astro-ph	1	16706	121251	0.001	1029	14	2930	**2930**	3012	3157	**2930**
	2		1.8×10^6	0.013		7	1464	**1464**	1530	1673	1471
	3		1.5×10^7	0.105		5	1163	**1163**	1187	1247	1171
	4		4.8×10^7	0.346		4	1074	**1074**	1083	1112	1079
	5		8.2×10^7	0.587		3	1044	**1044**	1048	1056	1045
	6		10^8	0.717		3	1035	**1035**	1037	1039	**1035**
	7		1.1×10^8	0.768		2	1031	**1031**	**1031**	1032	**1031**
	8		1.1×10^8	0.784		2	1029	**1029**	**1029**	**1029**	**1029**
cond-mat	1	16726	47594	<0.001	1188	18	3394	**3394**	3442	3590	**3394**
	2		322714	0.002		9	1886	**1886**	1965	2142	**1886**
	3		1.8×10^6	0.013		6	1477	**1477**	1520	1614	1480
	4		7.6×10^6	0.054		5	1304	**1304**	1332	1390	1308
	5		2.3×10^7	0.163		4	1236	**1236**	1246	1277	1238
	6		4.7×10^7	0.336		3	1209	**1209**	1213	1229	1210
	7		7×10^7	0.501		3	1196	**1196**	1198	1204	1197
	8		8.5×10^7	0.605		3	1190	**1190**	**1190**	1195	**1190**
	9		9.2×10^7	0.656		2	1189	**1189**	**1189**	**1189**	**1189**
	10		9.5×10^7	0.678		2	1188	**1188**	**1188**	**1188**	**1188**
cond-mat-2003	1	31163	120029	<0.001	1599	16	5379	**5379**	5493	5759	**5379**
	2		1.4×10^6	0.003		8	2631	**2631**	2747	3088	2639
	3		1.2×10^7	0.024		6	1939	**1939**	1999	2195	1967
	4		5.9×10^7	0.122		4	1716	**1716**	1742	1820	1735
	5		1.7×10^8	0.338		4	1640	**1640**	1650	1686	1648
	6		2.8×10^8	0.572		3	1613	**1613**	1616	1632	1616
	7		3.4×10^8	0.708		3	1604	**1604**	1605	1611	1605
	8		3.7×10^8	0.76		2	1600	**1600**	**1600**	1602	**1600**
	9		3.8×10^8	0.775		2	1599	**1599**	**1599**	1600	**1599**
cond-mat-2005	1	40421	175692	<0.001	1798	18	6508	**6508**	6637	7012	6509
	2		2.6×10^6	0.003		9	3013	**3013**	3157	3598	3046
	3		2.6×10^7	0.032		6	2170	**2170**	2235	2472	2210
	4		1.3×10^8	0.164		5	1922	**1922**	1945	2038	1943
	5		3.5×10^8	0.426		4	1840	**1840**	1848	1890	1847
	6		5.4×10^8	0.658		4	$\geq$ 1798	-	1817	1829	**1816**
	7		6.3×10^8	0.767		3	$\geq$ 1798	-	1804	1809	**1803**
	8		6.5×10^8	0.802		3	$\geq$ 1798	-	**1800**	1803	**1800**
	9		6.6×10^8	0.811		3	$\geq$ 1798	-	**1799**	**1799**	**1799**
	10		6.6×10^8	0.813		2	1798	-	**1798**	**1798**	**1798**
hep-th	1	8361	15751	<0.001	1332	19	2613	**2613**	2630	2697	**2613**
	2		84368	0.002		10	1768	**1768**	1803	1886	**1768**
	3		376431	0.011		7	1517	**1517**	1541	1598	**1517**
	4		1.3×10^6	0.038		5	1418	**1418**	1433	1463	1419
	5		3.5×10^6	0.101		4	1374	**1374**	1384	1396	1375
	6		6.9×10^6	0.199		4	1353	**1353**	1359	1364	1354
	7		1.1×10^7	0.304		3	1342	**1342**	1347	1348	**1342**
	8		1.4×10^7	0.387		3	1337	**1337**	1338	1339	**1337**
	9		1.5×10^7	0.439		3	1334	**1334**	**1334**	**1334**	**1334**
	10		1.6×10^7	0.467		2	1333	**1333**	**1333**	**1333**	**1333**
	11		1.7×10^7	0.479		2	1332	**1332**	**1332**	**1332**	**1332**

Table 3: Experimental results of the studied algorithms for network science instances with only the largest component taken into account

graph	k	n	m	$\overline{C}$	δ	c	d	γ	CBC	GRE	ACO-LS-S	RLS$_o$
Graphs from Newman's network data repository (research collaboration networks [2]), largest component only												
astro-ph	1	14845	119652	0.67	0.001	1	14	1892	**1892**	1970	2100	**1892**
	2		1.8×10^6	0.613	0.016		7	436	**436**	502	635	444
	3		1.5×10^7	0.66	0.133		5	135	**135**	159	214	142
	4		4.8×10^7	0.815	0.438		4	46	**46**	55	74	53
	5		8.2×10^7	0.905	0.743		3	16	**16**	20	25	17
	6		10^8	0.958	0.908		3	7	**7**	9	9	**7**
	7		1.1×10^8	0.985	0.972		2	3	**3**	**3**	**3**	**3**
	8		1.1×10^8	0.995	0.992		2	1	**1**	**1**	**1**	**1**
cond-mat	1	13861	44619	0.651	<0.001	1	18	2172	**2172**	2220	2347	**2172**
	2		318876	0.709	0.003		9	698	**698**	778	940	**698**
	3		1.8×10^6	0.574	0.018		6	290	**290**	333	421	291
	4		7.6×10^6	0.589	0.079		5	117	**117**	146	192	122
	5		2.2×10^7	0.686	0.238		4	49	**49**	59	80	52
	6		4.7×10^7	0.805	0.489		3	22	**22**	26	32	23
	7		7×10^7	0.888	0.729		3	19	**9**	11	14	10
	8		8.5×10^7	0.942	0.881		3	3	**3**	**3**	4	**3**
	9		9.2×10^7	0.974	0.955		2	2	**2**	**2**	**2**	**2**
	10		9.5×10^7	0.99	0.985		2	1	**1**	**1**	**1**	**1**
cond-mat-2003	1	27519	116181	0.655	<0.001	1	16	3742	**3742**	3853	4075	**3742**
	2		1.4×10^6	0.66	0.004		8	1032	**1032**	1151	1483	1045
	3		1.2×10^7	0.536	0.031		6	341	**341**	400	571	363
	4		5.9×10^7	0.646	0.156		4	118	**118**	144	205	135
	5		1.7×10^8	0.788	0.433		4	42	**42**	52	78	51
	6		2.8×10^8		0.733		3	15	**15**	18	26	18
	7		3.4×10^8		0.908		3	6	**6**	7	10	7
	8		3.7×10^8		0.975		2	2	**2**	**2**	**2**	**2**
	9		3.8×10^8		0.993		2	1	**1**	**1**	**1**	**1**
cond-mat-2005	1	36458	171735	0.657	<0.001	1	18	4678	**4678**	4807	5130	**4678**
	2		2.6×10^6	0.643	0.004		9	1214	**1214**	1359	1753	1245
	3		2.6×10^7	0.539	0.039		6	373	**373**	440	624	409
	4		1.3×10^8	0.687	0.202		5	125	**125**	148	204	146
	5		3.5×10^8		0.524		4	43	**43**	52	77	52
	6		5.4×10^8		0.809		4	≥ 2	-	20	26	**17**
	7		6.3×10^8		0.943		3	≥ 2	-	7	8	**6**
	8		6.5×10^8		0.985		3	≥ 2	-	**3**	**3**	**3**
	9		6.6×10^8		0.996		3	2	-	**2**	**2**	**2**
	10		6.6×10^8		0.999		2	1	-	**1**	**1**	**1**
hep-th	1	5835	13815	0.506	0.001	1	19	1241	**1241**	1260	1320	**1241**
	2		81697	0.705	0.005		10	433	**433**	466	542	**433**
	3		373571	0.614	0.022		7	186	**186**	211	256	**186**
	4		1.3×10^6	0.634	0.079		5	87	**87**	102	122	**87**
	5		3.5×10^6	0.7	0.207		4	43	**43**	53	58	44
	6		6.9×10^6	0.791	0.408		4	22	**22**	28	29	23
	7		1.1×10^7	0.859	0.623		3	11	**11**	16	15	**11**
	8		1.4×10^7	0.911	0.794		3	6	**6**	7	7	**6**
	9		1.5×10^7	0.949	0.901		3	3	**3**	**3**	**3**	**3**
	10		1.6×10^7	0.975	0.958		2	2	**2**	**2**	**2**	**2**
	11		1.7×10^7	0.989	0.984		2	1	**1**	**1**	**1**	**1**

cond-mat-2005 represent three different stages of the evolution of a network of condensed matter collaborations, and *hep-th* represents collaborations in high energy theory [2].

Results for these instances are presented in two stages. Table 2 summarises the results obtained for the entire collaboration networks with large numbers of connected components c, while Table 3 presents the results for the largest connected components only. For the collaboration networks, CBC also provides optimal results for most of the instances. RLS_o seems to provide—quite consistently—the best results among the heuristics. Its results seem to be closer to the optimum found by CBC for low and high values of k, with larger gaps being left between the optimum and the result of RLS_o for values of k in the middle. Surprisingly, ACO-LS-S seems to lag behind the results of both RLS_o and GREEDY, which is in sharp contrast to the results obtained for social networks. Several instances have been found with similar metrics as observed for the social network samples. However, these were obtained for higher values of k, as the diameters of these networks are shortened less rapidly.

The limitations of CBC start to be noticeable for graph *cond-mat-2005*. For $k \geq 6$, CBC was unsuccessful, similarly to the results obtained for some of the largest instances for social networks. For these instances, RLS_o provided the best result among the heuristics. However, no better than the trivial lower bound was found.

The marginal values obtained for collaboration networks are considerably higher than those for the social network samples. For network *astro-ph*, the marginal value was $k = 8$, while for *cond-mat*, *cond-mat-2003*, and *cond-mat-2005*, these were $k = 10$, $k = 9$, and $k = 10$, respectively. Since these networks represent the state of condensed matter collaborations in progressing years, this reveals that the evolution of the network over time does not necessarily have to lead to an increase of the marginal value. This can perhaps be explained by new collaborations being initiated by scientists over time, which are modelled by edge introduction rather than attachment of new vertices. For network *netscience*, we obtained a marginal value of $k = 9$, while for *hep-th*, it was $k = 11$.

Results for power grid and Internet snapshot. For the power grid network *power*, and for the Internet snapshot *as-22july06*, slightly different patterns have been observed. These results are summarised in Table 4. The Internet snapshot is a connected graph with critical value of $k = 4$ and a marginal

Table 4: Experimental results of the studied algorithms for network science instances

graph	k	n	m	$\overline{C}$	δ	c	d	γ	CBC	GRE	ACO-LS-S	RLS$_o$
Graphs from Newman's network data repository (power grid and Internet snapshot)												
power [52]	1	4941	6594	0.08	<0.001	1	46	1481	**1481**	1546	1650	**1481**
	2		22629	0.653	0.002		23	658	**658**	717	806	**658**
	3		53125	0.659	0.004		16	345	**345**	393	451	**345**
	4		105233	0.686	0.009		12	207	**207**	239	276	**207**
	5		185992	0.698	0.015		10	131	**131**	151	181	**131**
	6		301550	0.716	0.025		8	83	**83**	100	118	**83**
	7		460075	0.726	0.038		7	57	**57**	73	82	**57**
	8		668664	0.736	0.055		6	37	**37**	48	54	**37**
	9		932433	0.742	0.076		6	25	**25**	35	38	**25**
	10		1.1×10^6	0.748	0.103		5	18	**18**	22	26	19
	11		1.6×10^6	0.752	0.134		5	11	**11**	16	17	**11**
	12		2.1×10^6	0.757	0.17		4	10	**10**	12	12	**10**
	13		2.6×10^6	0.761	0.211		4	7	**7**	11	9	**7**
	14		3.1×10^6	0.765	0.256		4	6	**6**	9	7	**6**
	15		3.7×10^6	0.77	0.306		4	5	**5**	7	**5**	**5**
	16		4.4×10^6	0.777	0.359		3	4	**4**	5	**4**	**4**
	17		5.1×10^6	0.785	0.415		3	3	**3**	6	**3**	**3**
	18		5.8×10^6	0.796	0.472		3	3	**3**	5	**3**	**3**
	19		6.5×10^6	0.807	0.529		3	3	**3**	4	**3**	**3**
	20		7.1×10^6	0.82	0.585		3	2	**2**	4	**2**	**2**
	21		7.8×10^6	0.834	0.639		3	2	**2**	3	**2**	**2**
	22		8.4×10^6	0.85	0.692		3	2	**2**	3	**2**	**2**
	23		9.1×10^6	0.866	0.742		2	1	**1**	**1**	**1**	**1**
*as-22july06**	1	22963	48436	0.23	<0.001	1	11	2026	**2026**	2028	**2026**	**2026**
	2		1.1×10^7	0.832	0.042		6	312	**312**	313	388	**312**
	3		9.6×10^7	0.84	0.363		4	47	**47**	51	67	**47**
	4		2.1×10^8		0.793		3	8	**8**	**8**	**8**	**8**
	5		2.5×10^8		0.963		3	2	**2**	**2**	**2**	**2**
	6		2.6×10^8		0.996		2	1	**1**	**1**	**1**	**1**

* Snapshot of the Internet has not been previously published in a research paper. It is published in Newman's network data repository:
`http://www-personal.umich.edu/~mejn/netdata/`.

value of $k = 6$. In the next paragraphs, it will be seen that this decline is somewhat similar to the pattern observed for social networks. However, it is worth mentioning that for *as-22july06*, RLS$_o$ outperformed ACO-LS-S, while for social networks, this was reversed.

Network *power* is clearly the most resilient in terms of k-reachability. Apparently, rather than being a small world network, this network is lattice-based, with a grid as the underlying structure. RLS$_o$ produced the best results for this network. ACO-LS-S and GREEDY performed less efficiently. The critical value for *power* we obtained is $k = 13$, while the marginal value is $k = 23$.

Results for easy problem instances. These instances include both instances from Newman's network data repository and DIMACS graphs. The exper-

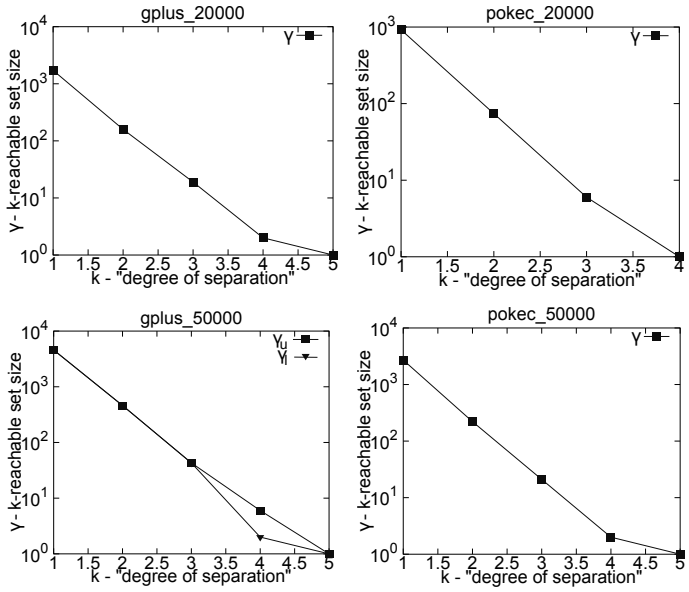

Figure 5: Visualisations of the generalised dominating set size, as a function of the "degree of separation" k in logarithmic scale for y axis. Visualisations are provided for networks *gplus_20000*, *pokec_20000*, *gplus_50000*, and *pokec_50000*. The presented values represent the best bounds we were able to find using any approach.

imental results are presented in Table 5. For networks with more than one connected component, the results obtained for the largest connected components are given in Table 6.

Network *adjnoun* is a network of adjective-noun adjacencies for David Copperfield [53]. Network *football* represents games in a season of an American college football league [1]. Instance *lesmis* represents a coappearance network of characters for Les Miserables [54]. Instance *netscience* represents network science collaborations [53]. Network *zachary* represents friendships in a karate club [55]. Instance *celegansneural* represents a neural network for C. Elegans [52]. Instance *dolphins* represents a social network of bottlenose dolphins [49] illustrated above. Finally, *polbooks* is a network of Krebs' political books, previously used in community detection literature [56].

DIMACS instances for graph colouring include coappearance networks of characters for some of the famous literary works. Coappearance network *anna* represents characters for Anna Karenina, *homer* represents Iliad and

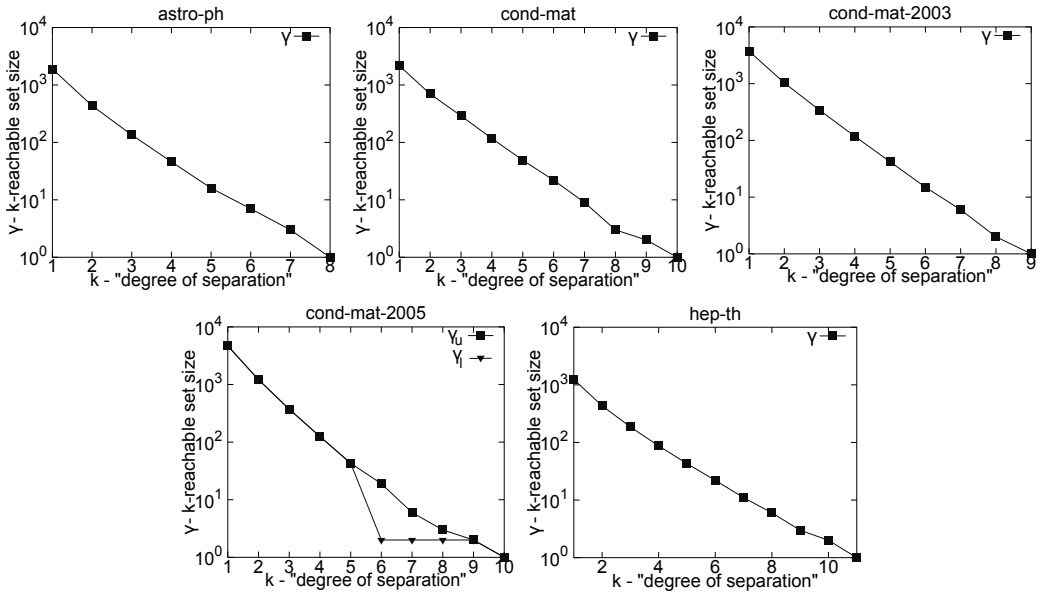

Figure 6: Visualisations of the k-reachable set profile γ, as a function of the "degree of separation" k. Visualisations are provided for networks *netscience*, *astro-ph*, *cond-mat*, *cond-mat-2003*, *cond-mat-2005*, and *hep-th*. The presented values represent the best bounds we were able to find using any approach.

Odyssey, *david* represents David Copperfield, and *huck* is a coappearance network for Huckleberry Finn.

An interesting observation is that RLS_o performs well for these networks, while ACO-LS-S and GREEDY overestimate the dominating set size for several instances. For most of the small networks, including *adjnoun*, *football*, *lesmis*, *zachary*, and *celegansneural*, the marginal value of k was obtained at $k = 3$. Networks *dolphins* and *polbooks* are slightly different, with a "plateau" of dominating set sizes for $3 \leq k \leq 4$ for *dolphins* and $2 \leq k \leq 3$ for *polbooks*. Due to this phenomenon, these two networks seem slightly more resilient, with marginal values $k = 5$ for *dolphins* and $k = 4$ for *polbooks*.

Impact of increased k on dominating set size. Apart from the computational results of the algorithms, we are also interested in the decline of upper and lower bounds for dominating set sizes as k grows.

Figure 5 represents the plots of such a decline for social network samples from *Google+* and *Pokec*. The x-axis in these plots represents the value of

k and the y-axis is logarithmic, representing the dominating set size γ. For three of the graphs, the optimal profile of γ was found, while for *gplus_50000*, upper bounds γ_u and the lower bounds γ_l are provided.

The plots suggest that in social networks, the decline is very sharp, which is supported by the small-world properties of these networks. For the scale of several thousands of vertices, small dominating sets for $k = 3$ can already guarantee a relatively wide reachability. For graphs with 50000 vertices, $k = 4$ was already a critical value, with the sampled k-reachable set sizes below the threshold of 10. All the obtained values are currently within $k \leq 6$, which seems to be in line with the "six degrees of separation theory". An interesting open problem seems to be an estimation of how this value grows for larger graph sizes. Given the potential sizes and densities of the shortcut graphs, this may be investigated by means of graph decomposition techniques, and possibly, elements of high performance computing. Another option could be the exploration of representative network sampling methods [57], and an extrapolation of the k-reachable set sizes, as well as the critical and marginal values obtained from these samples.

Figure 6 presents the k-reachable set profiles for scientific collaboration networks. These include astrophysics, condensed matter and high energy theory. For all of these networks, the k-reachable set sizes represent the data obtained for the largest connected components. Critical values for these research collaboration networks seem to be located at higher values of k, as the diameters of their shortcut graphs are slightly higher. The pattern of the decline is also slightly milder, even though the plots indicate that the decline may still be superpolynomially or even exponentially fast.

Figure 7 represents the non-increasing profiles for dominating set sizes for the power grid and the Internet snapshot. The power grid network *power* is by far the most resilient network among all instances, which is most probably related to the fact that the underlying structure of it is a grid, rather than a scale-free structure. The obtained values indicate a pattern of a gradual slowdown, with non-uniform decline and flat regions obtained especially for higher values of k. Despite this fact, structural vulnerabilities of power grids have previously been identified [58]. Network *as-22july06* is a snapshot of the Internet on the level of autonomous systems, reconstructed from BGP tables. The network shows a profile, which is very similar to profiles observed for social networks, with a very sharp decline of the k-reachable set size and a critical value of $k = 4$.

Last but not least, a comparison of the k-reachable set profiles for net-

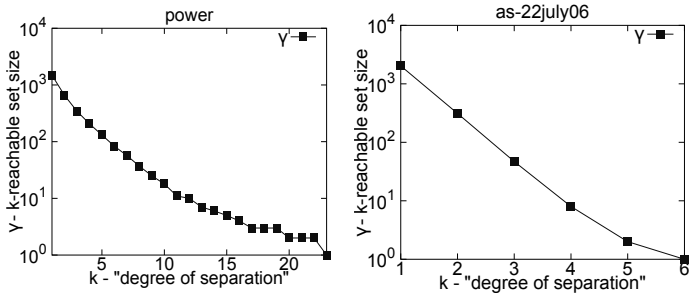

Figure 7: Visualisations of the k-reachable set profile γ, as a function of the "degree of separation" k. Visualisations are provided for networks *power* and *as-22july06*. These plots use logarithmic scale for y axis.

works of similar origin is given in Figure 8. These were obtained for the social networks, as well as for the research collaboration networks. For social networks, one can observe very similar decline patterns. An increase in the number of vertices seems to have an impact on the value observed for $k = 1$, with the rest of the values scaling comparably. For research collaboration networks, *astro-ph* has the steepest profile, while *hep-th* exhibits the most moderate decline of γ. The profiles for *cond-mat*, *cond-mat-2003* and *cond-mat-2005* are similar at first sight. However, while *cond-mat-2005* has the highest value of γ for $k = 1$, one can notice that this is not the case for $k = 7$. This suggests that the processes linked to the evolution of these networks may have a complex impact on k-reachability.

5. Discussion and Conclusions

We presented a technique for mining of k-reachable sets in real-world networks. A k-reachable set represents a set of vertices of a network such that all vertices can be reached within distance k from some vertex of the k-reachable set.

The concept of shortcut graphs has been introduced to model the k-reachability, and approaches to solve the minimum dominating set (MDS) problem were employed to find k-reachable sets for a diverse set of real-world networks. The data set for our experiments included samples from two social network services, as well as network science data, particularly research collaboration networks, a power grid and an Internet snapshot.

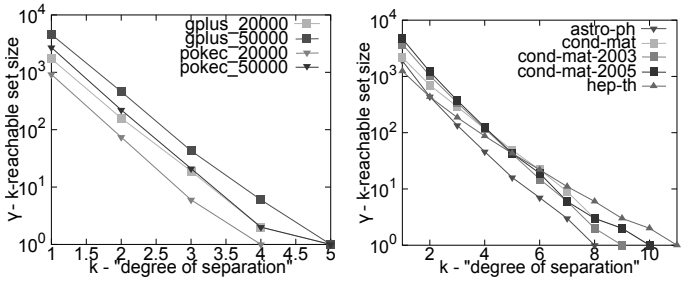

Figure 8: Comparative visualisations of the k-reachable set profile γ for the social networks and research collaboration networks.

For most instances, the approach based on an integer linear programming (ILP) formulation and the application of an open-source ILP branch-and-cut solver led to optimal results. However, the memory demands of the ILP-based approach tend to grow very quickly, making it unsuccessful for some of the largest graphs with high values of k. Therefore, we also explored the use of three heuristics, including a greedy approximation algorithm, an ant-based algorithm ACO-LS-S and an order-based algorithm RLS_o. Interestingly, RLS_o works well for most instances, while ACO-LS-S has shown stronger performance for shortcut graphs of large social networks.

Patterns of the decline of k-reachable set size with growing k have been investigated. Social networks exhibit a pattern of a very sharp decline, which is in line with their small-world structure. A similar pattern was observed for the Internet snapshot. A slightly slower decline is exhibited for research collaboration networks, even though this still seems likely to be superpolynomial. On the other hand, a very slow decline profile was observed for the power grid.

Acknowledgement. Christian Blum acknowledges support by grant TIN2012-37930-C02-02 of the Spanish Government.

References

[1] M. Girvan, M. E. J. Newman, Community structure in social and biological networks, Proceedings of the National Academy of Sciences 99 (12) (2002) 7821–7826.

[2] M. E. J. Newman, The structure of scientific collaboration networks, Proceedings of the National Academy of Sciences 98 (2) (2001) 404–409.

[3] C. Pizzuti, S. E. Rombo, Algorithms and tools for protein-protein interaction networks clustering, with a special focus on population-based stochastic methods, Bioinformatics 30 (10) (2014) 1343–1352.

[4] A. L. Barabási, R. Albert, Emergence of scaling in random networks, Science 286 (5439) (1999) 509–512.

[5] D. J. Watts, Small Worlds, Princeton University Press, Princeton, NJ, 1999.

[6] M. R. Gary, D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-completeness, W. H. Freeman and Co., New York, 1979.

[7] E. Ravasz, A. L. Barabási, Hierarchical organization in complex networks, Physical Review E 67 (2) (2003) 026112.

[8] M. Potamias, F. Bonchi, C. Castillo, A. Gionis, Fast shortest path distance estimation in large networks, in: Proceedings of the 18th ACM conference on Information and knowledge management, ACM, 2009, pp. 867–876.

[9] R. Albert, H. Jeong, A. L. Barabási, Error and attack tolerance of complex networks, Nature 406 (6794) (2000) 378–382.

[10] K. A. Hawick, Water distribution network robustness and fragmentation using graph metrics, in: Proceedings of the International Conference on Water Resource Management (AfricaWRM 2012), no. 762-037, IASTED, Gabarone, Botswana, 2012, pp. 304–310, CSTN-158.

[11] K. A. Hawick, Betweenness centrality metrics for assessing electrical power network robustness against fragmentation and node failure, in: Proceeding of the International Conference on Power and Energy Systems (EuroPES 2012), IASTED, Napoli, Italy, 2012, pp. 186–193, CSTN-119.

[12] M. Chu, K. Ingols, R. Lippmann, S. Webster, S. Boyer, Visualizing attack graphs, reachability, and trust relationships with NAVIGATOR, in: Proceedings of the Seventh International Symposium on Visualization for Cyber Security, ACM, 2010, pp. 22–33.

[13] B. Doerr, M. Fouz, T. Friedrich, Why rumors spread so quickly in social networks, Communications of the ACM 55 (6) (2012) 70–75.

[14] P. Luo, Y. Li, C. Wu, K. Chen, Detecting the missing links in social networks based on utility analysis, Journal of Computational Science 16 (2016) 51–58.

[15] G. Thareja, Z. Kronfol, K. Suhre, P. Kumar, A graph based method for depicting population characteristics using genome wide data, Journal of Computational Science 15 (2016) 11–17.

[16] B. Bollobás, O. M. Riordan, Mathematical results on scale-free random graphs, in: S. Bornholdt, H. G. Schuster (Eds.), Handbook of Graphs and Networks, Wiley, 2005, pp. 1–34.

[17] S. E. Schaeffer, Graph clustering, Computer Science Review 1 (1) (2007) 27–64.

[18] G. W. Flake, S. Lawrence, C. L. Giles, Efficient identification of web communities, in: R. Ramakrishnan, S. Stolfo, R. Bayardo, I. Parsa (Eds.), Proceedings of the 6th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '00, ACM, New York, NY, 2000, pp. 150–160.

[19] J. Leskovec, K. J. Lang, M. W. Mahoney, Empirical comparison of algorithms for network community detection, in: J. F. M. Rappa, P. Jones, S. Chakrabarti (Eds.), Proceedings of the 19th international conference on World wide web, WWW '10, ACM, New York, NY, 2010, pp. 631–640.

[20] J. Leskovec, K. J. Lang, A. Dasgupta, M. W. Mahoney, Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters, Internet Mathematics 6 (1) (2009) 29–123.

[21] K. Liu, J. Huang, H. Sun, M. Wan, Y. Qi, H. Li, Label propagation based evolutionary clustering for detecting overlapping and non-overlapping

communities in dynamic networks, Knowledge-Based Systems 89 (2015) 487–496.

[22] F. Chen, K. Li, Detecting hierarchical structure of community members in social networks, Knowledge-Based Systems 87 (2015) 3–15.

[23] L. Kaufman, P. J. Rousseeuw, Finding groups in data: an introduction to cluster analysis, Wiley, 1990.

[24] H. S. Park, C. H. Jun, A simple and fast algorithm for k-medoids clustering, Expert Systems with Applications 36 (2, Part 2) (2009) 3336–3341.

[25] F. Dai, J. Wu, An extended localized algorithm for connected dominating set formation in ad hoc wireless networks, IEEE Transactions on Parallel and Distributed Systems 15 (10) (2004) 908–920.

[26] F. Kuhn, G. Wattenhofer, Constant-time distributed dominating set approximation, Distributed Computing 17 (4) (2005) 303–310.

[27] J. Wu, H. Li, On calculating connected dominating set for efficient routing in ad hoc wireless networks, in: Proceedings of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, DIALM '99, ACM, New York, NY, USA, 1999, pp. 7–14.

[28] R. Jovanovic, M. Tuba, Ant colony optimization algorithm with pheromone correction strategy for the minimum connected dominating set problem, Computer Science and Information Systems 10 (1) (2013) 133–149.

[29] C. Shen, T. Li, Multi-document summarization via the minimum dominating set, in: Proceedings of the 23rd International Conference on Computational Linguistics, COLING '10, Association for Computational Linguistics, Stroudsburg, PA, USA, 2010, pp. 984–992.

[30] T. N. Dinh, Y. Shen, D. T. Nguyen, M. T. Thai, On the approximability of positive influence dominating set in social networks, Journal of Combinatorial Optimization 27 (3) (2014) 487–503.

[31] F. Wang, H. Du, E. Camacho, K. Xu, W. Lee, Y. Shi, S. Shan, On positive influence dominating sets in social networks, Theoretical Computer Science 412 (3) (2011) 265–269.

[32] S. Bouamama, C. Blum, A hybrid algorithmic model for the minimum weight dominating set problem, Simulation Modelling Practice and Theory 64 (2016) 57–68.

[33] F. V. Fomin, F. Grandoni, D. Kratsch, A measure & conquer approach for the analysis of exact algorithms, Journal of the ACM 56 (5) (2009) 25:1–25:32.

[34] U. Feige, A threshold of ln n for approximating set cover, Journal of the ACM 45 (4) (1998) 634–652.

[35] S. Masuyama, T. Ibaraki, T. Hasegawa, The computational complexity of the m-center problems in the plane, Transactions of IECE Japan E64 (2) (1981) 57–64.

[36] B. N. Clark, C. J. Colbourn, D. S. Johnson, Unit disk graphs, Discrete Mathematics 86 (1-3) (1990) 165–177.

[37] M. Chlebík, J. Chlebíková, Approximation hardness of dominating set problems in bounded degree graphs, Information and Computation 206 (11) (2008) 1264–1275.

[38] M. Gast, M. Hauptmann, M. Karpinski, Inapproximability of dominating set on power law graphs, Theoretical Computer Science 562 (2015) 436–452.

[39] P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuéjols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, et al., An algorithmic framework for convex mixed integer nonlinear programs, Discrete Optimization 5 (2) (2008) 186–204.

[40] J. T. Linderoth, A. Lodi, MILP software, in: Wiley Encyclopedia of Operations Research and Management Science, John Wiley, 2010.

[41] V. Chvátal, A greedy heuristic for the set-covering problem, Mathematics of Operations Research 4 (3) (1979) 233–235.

[42] L. A. Sanchis, Experimental analysis of heuristic algorithms for the dominating set problem, Algorithmica 33 (1) (2002) 3–18.

[43] D. Chalupa, An Order-based Algorithm for Minimum Dominating Set with Application in Graph Mining, ArXiv e-prints, arXiv:1705.00318.

[44] A. Potluri, A. Singh, Two hybrid meta-heuristic approaches for minimum dominating set problem, in: Proceedings of the Second International Conference on Swarm, Evolutionary, and Memetic Computing - Volume Part II, SEMCCO'11, Springer, Berlin, Heidelberg, 2011, pp. 97–104.

[45] A. R. Hedar, R. Ismail, Hybrid genetic algorithm for minimum dominating set problem, in: D. Taniar, O. Gervasi, B. Murgante, E. Pardede, B. O. Apduhan (Eds.), Computational Science and Its Applications - ICCSA 2010, Vol. 6019 of Lecture Notes in Computer Science, Springer, 2010, pp. 457–467.

[46] A. Potluri, A. Singh, Hybrid metaheuristic algorithms for minimum weight dominating set, Applied Soft Computing 13 (1) (2013) 76–88.

[47] D. S. Johnson, M. Trick, Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge, American Mathematical Society, Providence, RI, 1996.

[48] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, Z. Ghahramani, Kronecker graphs: An approach to modeling networks, The Journal of Machine Learning Research 11 (2010) 985–1042.

[49] D. Lusseau, K. Schneider, O. J. Boisse, P. Haase, E. Slooten, S. M. Dawson, The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations, Behavioral Ecology and Sociobiology 54 (4) (2003) 396–405.

[50] J. Kleinberg, The small-world phenomenon: An algorithmic perspective, in: Proceedings of the thirty-second annual ACM symposium on Theory of computing, ACM, 2000, pp. 163–170.

[51] L. Takac, M. Zabovsky, Data analysis in public social networks, in: International Scientific Conference and International Workshop Present Day Trends of Innovations, 2012, pp. 1–6.

[52] D. J. Watts, S. H. Strogatz, Collective dynamics of 'small-world' networks, Nature 393 (1998) 440–442.

[53] M. E. J. Newman, Finding community structure in networks using the eigenvectors of matrices, Physical Review E 74 (036104) (2006) 036104–1–036104–19.

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

[55] W. W. Zachary, An information flow model for conflict and fission in small groups, Journal of Anthropological Research 33 (1977) 452–473.

[56] C. Pizzuti, A multiobjective genetic algorithm to find communities in complex networks, IEEE Transactions on Evolutionary Computation 16 (3) (2012) 418–430.

[57] J. Leskovec, C. Faloutsos, Sampling from large graphs, in: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2006, pp. 631–636.

[58] R. Albert, I. Albert, G. L. Nakarado, Structural vulnerability of the North American power grid, Physical review E 69 (2) (2004) 025103.

Table 5: Experimental results of the studied algorithms for easy problem instances.

graph	k	n	m	$\overline{C}$	δ	c	d	γ	CBC	GRE	ACO-LS-S	RLS$_o$
Graphs from Newman's network data repository												
adjnoun [53]	1	112	425	0.173	0.068	1	5	18	**18**	**18**	**18**	**18**
	2		3082	0.779	0.496		3	3	**3**	4	**3**	**3**
	3		5634	0.944	0.906		2	1	**1**	**1**	**1**	**1**
football [1]	1	115	613	0.403	0.094	1	4	12	**12**	13	13	**12**
	2		2919	0.56	0.445		2	3	**3**	4	**3**	**3**
	3		6247	0.957	0.953		2	1	**1**	**1**	**1**	**1**
lesmis [54]	1	77	254	0.573	0.087	1	5	10	**10**	**10**	**10**	**10**
	2		1249	0.864	0.426		3	2	**2**	**2**	**2**	**2**
	3		2500	0.923	0.854		2	1	**1**	**1**	**1**	**1**
netscience [53]	1	1589	2742	0.638	0.002	396	17	477	**477**	**477**	482	**477**
	2		6722	0.736	0.005		9	418	**418**	421	423	**418**
	3		13087	0.731	0.01		6	404	**404**	407	**404**	**404**
	4		22847	0.738	0.018		5	400	**400**	402	**400**	**400**
	5		34931	0.742	0.028		4	398	**398**	400	**398**	**398**
	6		47479	0.752	0.038		3	397	**397**	398	**397**	**397**
	7		58734	0.764	0.047		3	397	**397**	**397**	**397**	**397**
	8		66083	0.774	0.052		3	397	**397**	**397**	**397**	**397**
	9		70436	0.78	0.056		2	396	**396**	**396**	**396**	**396**
zachary [55]	1	34	78	0.571	0.139	1	5	4	**4**	**4**	**4**	**4**
	2		343	0.861	0.611		3	2	**2**	**2**	**2**	**2**
	3		480	0.922	0.856		2	1	**1**	**1**	**1**	**1**
celegansneural [52]	1	297	2148	0.292	0.049	1	5	16	**16**	17	**16**	**16**
	2		24122	0.766	0.549		3	3	**3**	**3**	**3**	**3**
	3		41637	0.965	0.947		2	1	**1**	**1**	**1**	**1**
dolphins [49]	1	62	159	0.259	0.08408	1	8	14	**14**	15	**14**	**14**
	2		607	0.729	0.321		4	4	**4**	5	**4**	**4**
	3		1107	0.842	0.585		3	2	**2**	3	**2**	**2**
	4		1459	0.897	0.772		2	2	**2**	**2**	**2**	**2**
	5		1717	0.944	0.908		2	1	**1**	**1**	**1**	**1**
*polbooks**	1	105	441	0.488	0.081	1	7	13	**13**	14	**13**	**13**
	2		2002	0.764	0.367		4	2	**2**	**2**	**2**	**2**
	3		3510	0.846	0.643		3	2	**2**	**2**	**2**	**2**
	4		4685	0.917	0.858		2	1	**1**	**1**	**1**	**1**
DIMACS graphs [47]												
anna	1	138	493	0.653	0.052	1	5	12	**12**	**12**	**12**	**12**
	2		5131	0.86	0.543		3	3	**3**	**3**	**3**	**3**
	3		9087	0.984	0.961		2	1	**1**	**1**	**1**	**1**
homer	1	561	1628	0.404	0.01	12	9	96	**96**	**96**	**96**	**96**
	2		21738	0.819	0.139		5	31	**31**	32	**31**	**31**
	3		91537	0.848	0.583		3	16	**16**	**16**	**16**	**16**
	4		133957	0.939	0.853		3	14	**14**	**14**	**14**	**14**
	5		143912	0.968	0.916		2	12	**12**	**12**	**12**	**12**
david	1	87	406	0.688	0.109	1	3	2	**2**	**2**	**2**	**2**
	2		3539	0.979	0.946		2	1	**1**	**1**	**1**	**1**
huck	1	74	301	0.774	0.111	3	4	9	**9**	**9**	**9**	**9**
	2		1737	0.896	0.643		2	3	**3**	**3**	**3**	**3**

Table 6: Experimental results of the studied algorithms for easy problem instances with only the largest component taken into account.

graph	k	n	m	$\overline{C}$	δ	c	d	γ	CBC	GRE	ACO-LS-S	RLS$_o$
Graphs from Newman's network data repository												
netscience [53]	1	379	914	0.741	0.013	1	17	55	**55**	**55**	56	**55**
	2		3830	0.831	0.053		9	21	**21**	23	22	**21**
	3		9523	0.784	0.133		6	7	**7**	10	8	**7**
	4		18892	0.792	0.264		5	5	**5**	7	**5**	**5**
	5		30667	0.805	0.428		4	3	**3**	5	**3**	**3**
	6		43019	0.84	0.601		3	2	**2**	3	**2**	**2**
	7		54228	0.888	0.757		3	2	**2**	**2**	**2**	**2**
	8		61577	0.928	0.86		3	2	**2**	**2**	**2**	**2**
	9		65930	0.953	0.92		2	1	**1**	**1**	**1**	**1**
DIMACS graphs [47]												
homer	1	542	1619	0.413	0.011	1	9	85	**85**	**85**	**85**	**85**
	2		21728	0.837	0.148		5	20	**20**	21	21	**20**
	3		91527	0.867	0.624		3	5	**5**	**5**	**5**	**5**
	4		133947	0.961	0.914		3	3	**3**	**3**	**3**	**3**
	5		143902	0.991	0.982		2	1	**1**	**1**	**1**	**1**
huck	1	69	297	0.786	0.127	1	4	7	**7**	**7**	**7**	**7**
	2		1733	0.917	0.739		2	1	**1**	**1**	**1**	**1**