



GraphCombEx: a software tool for exploration of combinatorial optimisation properties of large graphs

David Chalupa^{1,2} · Ken A Hawick²

Published online: 12 May 2018
© Springer-Verlag GmbH Germany, part of Springer Nature 2018

Abstract

We present a prototype of a software tool for exploration of multiple combinatorial optimisation problems in large real-world and synthetic complex networks. Our tool, called GraphCombEx (an acronym of Graph Combinatorial Explorer), provides a unified framework for scalable computation of high-quality suboptimal solutions and bounds for a number of widely studied combinatorial optimisation problems in large graphs. The problems currently supported include: maximum clique, graph colouring, maximum independent set, minimum vertex clique covering, minimum dominating set, as well as the longest simple cycle problem. Suboptimal solutions and intervals for optimal objective values are estimated using scalable heuristics. GraphCombEx has previously or currently been tested in scenarios of exploring synthetic graph models, as well as real-world networks ranging from social network samples, biological networks, to very large networks from the SNAP network data repository. The tool has already been successfully used to support a number of recent studies and is particularly beneficial in exploring the combinatorial properties of previously unseen network data, before applying more sophisticated custom optimisation algorithms.

Keywords GraphCombEx · Graph combinatorial explorer · Complex networks · Large sparse graphs · Combinatorial optimisation problems · Research software

1 Introduction

The demand for research support software has been growing in the last years in both academic and industrial applications. As an increasing number of real-world problems are concerned with analysis of networked data, exploration of complex networks has become one of the crucial areas of focus, including social networks (Brandes et al. 2012; Pattillo et al. 2012), biological networks (Schreiber et al. 2009) or utility distribution networks (Hawick 2012a, b). This explo-

ration is usually concerned with both numerical and visual analysis of networked data.

From the other perspective, combinatorial optimisation problems and algorithms to solve these problems have been subject of empirical research for a long time. Nowadays, this holds mainly for the variety of well-known NP-hard combinatorial graph-theoretical optimisation problems (Karp 1972). Over the last decade, there have been a vast number of papers on algorithms for solving these problems, ranging from graph colouring (Brélaz 1979; Culberson and Luo 1995), through minimum dominating set (Chvátal 1979), to crossing minimisation problems in graph drawing (Branke et al. 1996; Garey and Johnson 1983; Rosete-Suárez et al. 1999). The workflow of experimental research on combinatorial optimisation usually follows a pattern: an experimental algorithm is proposed, independently implemented and tested on a standardised benchmark (Abualigah et al. 2017, 2018; Johnson and Trick 1996; Li et al. 2016). In some studies, a set of problem instances is generated for the experiments in controlled conditions. The common benchmark usually represents the unifying element.

Communicated by V. Loia.

✉ David Chalupa
dc@m-tech.aau.dk
Ken A Hawick
k.a.hawick@hull.ac.uk

¹ Department of Materials and Production, Faculty of Engineering and Science, Aalborg University, Fibigerstræde 16, 9220 Aalborg, Denmark

² School of Engineering and Computer Science, University of Hull, Hull HU6 7RX, United Kingdom

Even though a number of software tools have been proposed to explore networked data (Bastian et al. 2009; Csardi and Nepusz 2006; Czech et al. 2011; Ellson et al. 2002; Hawick 2010), this has had little impact on empirical research in combinatorial optimisation so far. In this paper, we present a prototype of a software tool that aims at bridging this gap, providing a unifying platform for exploration of combinatorial properties of large-scale synthetic and real-world complex networks.

The tool is called Graph Combinatorial Explorer (abbr. GraphCombEx). It is based on a unifying representation and schematic of a number of NP-hard graph-theoretical problems. Many of such problems (e.g. maximum clique or minimum dominating set) are modelled as 0–1 vertex subset selection problems, while other problems are represented as labelling (e.g. graph colouring or vertex clique covering). In GraphCombEx, we provide a unifying platform for exploration of such problems.

GraphCombEx has been designed and implemented in small steps over the course of several years. The traditional approach in experimental research on algorithms is to develop a number of isolated applications to support different problems and algorithms. The tool proposed in this paper is designed to address the gap between these isolated applications and provide a common underlying platform. As a consequence, constructive algorithms or iterative improvement heuristics, for example, maximum clique or maximum independent set can be implemented in one place, since they share common traits. This will likely support better software reuse, maintenance and exploration of the results.

Our tool has already been used to support several studies on combinatorial properties of complex networks, including detection of long cycles (Chalupa et al. 2017), k -reachability problems (Chalupa and Blum 2017) and exploration of protein–protein interaction networks (Chalupa 2016). GraphCombEx has been particularly useful in the practical scenario of exploring previously unseen problem instances with simple and scalable heuristics, before applying a more complex heuristic. The core functionality, therefore, includes a number of constructive heuristics that can be used to quickly find good suboptimal solutions and lower bounds for potentially very large networks. With the NP-hardness of many graph-theoretical problems in mind, the design of GraphCombEx fully acknowledges that scaling and generalisation to previously unseen networks are crucial. The aim is also to potentially support further problems, algorithms and metrics in the future.

GraphCombEx currently supports exploration of undirected graphs with up to 5 million vertices, which can easily be extended to larger graphs. By default, graphs are handled in a format similar to the DIMACS format for graph colouring (Johnson and Trick 1996). Several graph generators are supported in GraphCombEx, including complete trees, unit

disc graphs or scale-free and small-world networks. The tool currently supports suboptimal solution construction and lower bounds for maximum clique and graph colouring, maximum independent set and vertex clique covering, minimum dominating set and the longest simple cycle. A number of graph metrics and visualisation techniques are also supported.

Examples of the previous use of GraphCombEx in initial instance exploration and visualisation of networks are summarised in the use case section of this paper. The tool has been tested with synthetic data generated within it, social network samples, protein–protein interaction networks from the UCLA database of interacting proteins (Salwinski et al. 2004; Xenarios et al. 2000, 2001, 2002), as well as a few large graphs from the SNAP network data repository (Leskovec and Krevl 2015).

The rest of the paper is structured as follows. In Sect. 2, we review the background and related work from relevant perspectives. In Sect. 3, we present the overall design elements of GraphCombEx. In Sect. 4, the summary of evaluation of GraphCombEx in several previous projects is presented, along with a discussion. Last but not least, Sect. 5 summarises this work and hints the future research directions.

2 Background and related work

At the start of this section, it is first worth noting the reasons for designing a new software platform for these types of problems. In fact, GraphCombEx draws inspiration from a number of research projects described below. However, combinatorial optimisation problems in complex networks tend to have several very specific properties that need to be addressed in design of a software to solve them. These can range from special data structures to complex parameterisations. Each problem and each algorithm tend to be quite specific, and a good trade-off between reuse and flexibility is, therefore, vital.

If exploring a previously unseen network, the optimum for many combinatorial properties may often be unknown. Furthermore, it is often unknown even how hard it will be to find the optimum. For some instances, it is possible that a simple heuristic will be sufficient to find a proven optimum. Optimality of the solution can sometimes be proven by simple automated bounds.

However, it is also possible that a highly advanced algorithm is required. In such a case, the requirement for a sophisticated algorithm should first be confirmed by the use of a simpler heuristic. In addition, some applications require processing of potentially huge graphs, particularly in the area of social networks. For small but structurally intricate networks, one can use high-performance algorithms to discover

previously unknown problem solutions. These are frequently obtained by employing advanced techniques to escape local optima.

For very large networks, it may be non-trivial to discover any reasonable solution. This has an impact on graph representation, selection of the right scalable heuristics, as well as the choice of the right data structures that allow the heuristics to run quickly. These observations have a significant impact on the overall design of a tool that should be useful for practice. Such specific demands are perhaps also the reason why combinatorial optimisation problems in complex networks are rarely explored by out-of-the-box solutions.

However, such solutions are quite prevalent in the areas of data mining and visualisation. Several software packages for complex network exploration and visualisation have been proposed and developed over the last years. We will now provide a brief overview of some of these solutions.

The *igraph* library is one of the most popular open-source software packages for network exploration (Csardi and Nepusz 2006). It is available in several variants, including interfaces for C, Python and R. The package provides a general-purpose platform for handling of both undirected and directed graphs. Solving routines for several classical graph-theoretical and graph mining problems are supported, e.g. maximum flow, minimum spanning tree or cluster detection.

Another general-purpose solution is the SNAP network analysis and graph mining library (Leskovec and Sosič 2016). This software package is intended as a high-performance large-scale complex network exploration tool, with support for handling potentially huge graphs. This platform is focused primarily on very large-scale network analytics, using a variety of metrics and simulations on the networks.

Graph drawing and visualisation are another significant aspect of complex network exploration software (Sugiyama 2002; Tamassia 2013). One of the most successful open-source software platforms focused mainly on graph visualisation is Gephi (Bastian et al. 2009). The features supported by Gephi include network exploration, interactive manipulation, 3D rendering, as well as of image exporting.

Another popular software package for graph visualisation is GraphViz (Ellson et al. 2002). This package is designed as a collection of customisable graph drawing tools, including batch layout and graph editors.

GraViz prototype (Hawick 2010) is aimed at interactive visualisation and exploration of graphs. This tool is particularly well suited for geographical data, for which it is easy to characterise their vertices by coordinates. As an example case study, this tool has been used to explore the structure and properties of power networks (Hawick 2012a). Another study was aimed at exploration of water distribution networks (Hawick 2012b).

Another specific property of NP-hard combinatorial optimisation problems in networks is that these are often formulated as integer linear programs. Out-of-the-box mixed integer linear programming (MILP) software tools are increasingly used in solving these problems in practical applications (Bonami et al. 2008; Linderoth and Lodi 2011). However, open-source MILP software and network exploration software still seem to be perceived in relative isolation, which is another interesting area of challenge.

3 GraphCombEx: graph combinatorial explorer and its design

The general idea behind GraphCombEx is quite simple and emerged from a number of similar problem solving projects. Usually, if several combinatorial optimisation problems and algorithms are explored, a suite of isolated ad hoc tools are developed, especially if new algorithms are designed. In this paper, a single unified solution for several problems and algorithms is attempted as an alternative.

In principle, GraphCombEx is designed as a platform for analysis and experimentation with static large sparse graphs. Specific focus is put on large-scale complex networks, as these have been gaining increasing attention in the last years. With particular interest, GraphCombEx addresses NP-hard combinatorial optimisation problems. Its design, therefore, reflects the need to incorporate several efficient and well-scalable heuristics to compute lower and upper bounds and high-quality suboptimal solutions to these problems for potentially very large complex networks.

GraphCombEx currently supports undirected graphs with up to 5 million vertices. This bound was only chosen for better predictability in memory management, computational complexity and testability, even though this bound can in theory be extended by changing a single constant in the source code. The graphs are loaded and saved into an extension of the COL format known in DIMACS graph colouring benchmarks (Johnson and Trick 1996). This format has been chosen to achieve simplicity. An example of a simple social network in this format is given in Listing 1. The tree-based visualisation and the adjacency matrix visualisation, generated by GraphCombEx for this particular network, are given in Fig. 1.

It is worth noting that a simple tool is also available with GraphCombEx, to allow conversion of the popular GML format, used in network science, to the extended COL format.

Listing 1. Representation of a simple artificial social network graph in the extended DIMACS graph format.

```

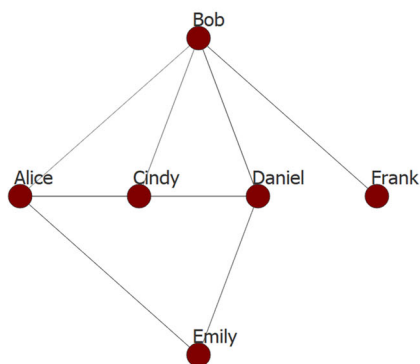
c 1 Alice
c 2 Bob
c 3 Cindy
c 4 Daniel
c 5 Emily
c 6 Frank
p edge 6 8
e 1 2
e 1 3
e 1 5
e 2 3
e 2 4
e 2 6
e 3 4
e 4 5

```

GraphCombEx supports generating multiple types of graphs. These currently include:

- complete n -ary trees;
- unit disc graphs;
- scale-free networks generated by the Barabási–Albert preferential attachment model (Albert and Barabási 2002; Barabási and Albert 1999);
- regular grids with probabilistic rewiring;
- small-world networks generated by the Watts–Strogatz model (Watts and Strogatz 1998).

Several simple graph manipulation and preprocessing routines are also included:



- generation of complementary graphs;
- pruning of leaves (iterative, i.e. the resulting graph will contain only vertices with degree at least 2);
- isolating the largest connected component;
- generation of shortcut graphs used in solving k -reachability problems (Chalupa and Blum 2017).

GraphCombEx computes a number of metrics for the graph explored. Those, which are currently supported, include:

- the numbers of vertices, edges, connected components, density, minimum, maximum, average and standard deviation of degrees;
- the number of triangles and the mean clustering coefficient of a vertex;
- graph girth (length of the shortest cycle) and maximum diameter of an isolated connected component.

Some of these are computed upon loading or generating the graph, as long as the complexity of the corresponding algorithms is at most $\mathcal{O}(m)$, where m is the number of edges. The numbers of vertices, edges, minimum, maximum and average degree, standard deviation of the degree are computed directly when graph is loaded or generated. Upon request, the number of triangles, the mean clustering coefficient and the diameter can be computed. The number of triangles is computed using scanning triplets of adjacent vertices and may potentially take a long time for very large graphs. The same holds also for the mean clustering coefficient of a vertex. The maximum diameter of a connected component is computed in $\mathcal{O}(nm)$ time, where n is the number of vertices.

Within GraphCombEx, the graph is represented as adjacency lists with sorted adjacencies, to allow for binary search to be used for efficient checking of adjacencies. An alternative extension would include a use of a hash table for direct

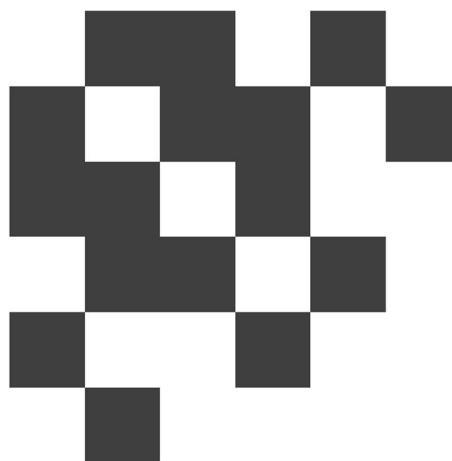


Fig. 1 A tree-based drawing and an adjacency matrix visualisation of the artificial social network defined in Listing 1, as generated by GraphCombEx

adjacency checking. This would speed up some computationally intensive routines but is not supported in the pilot version, as lower memory demands were preferred.

The core functionality of GraphCombEx comprises several heuristics and approximation algorithms, with a specific focus on constructive heuristics. These can be used to compute bounds and suboptimal solutions to a number of NP-hard graph-theoretical problems for very large graphs. Suboptima can then be improved by stochastic optimisation techniques, which are also partly supported within GraphCombEx. The problems that are currently supported within the prototype include:

- maximum clique and graph colouring / chromatic number problems (representing reciprocal lower and upper bounds);
- maximum independent set and vertex clique covering problems (representing reciprocal lower and upper bounds);
- minimum dominating set problem (both a suboptimum and simple lower bounds);
- the longest cycle problem (both a suboptimum and simple upper bounds).

These problems have been chosen for their high relevance for practice, as well as for their relatively diverse representations and properties. Maximum clique, maximum independent set and minimum dominating set problems represent 0–1 constrained substructure detection problems. Graph colouring and vertex clique covering are representatives of vertex labelling problems. Last but not least, the longest cycle problem is a problem of identification of specific walks on network. Extensions to other 0–1 optimisation problems (e.g. vertex cover), walk problems (e.g. longest path) or labelling problems (e.g. various community detection problems) should, therefore, be possible in the chosen framework.

The core of GraphCombEx is represented by techniques for computing bounds and high-quality suboptimal solutions to combinatorial problems in large sparse graphs. The heuristics currently supported include both fast constructive algorithms, as well as a few more computationally intensive iterative improvement algorithms. The constructive algorithms include the following heuristics and approximation algorithms:

- For maximum clique, GraphCombEx uses a greedy heuristic with binary heap, which orders the vertices by degree from largest to smallest, similar to the greedy heuristic for maximum independent set (Halldórsson and Radhakrishnan 1997).
- For the chromatic number problem, Brélez’s heuristic DSATUR is employed (Brélez 1979). Its implementation with binary heap is used, to allow for quick computing of

an upper bound of the chromatic number (Morgenstern 1991; Turner 1988).

- Maximum independent set is estimated by the greedy approximation heuristic, which orders the vertices by degree from smallest to largest. In our implementation, we use a binary heap. This heuristic guarantees $\mathcal{O}(\Delta)$ -approximation, where Δ is the maximum degree of a vertex in the graph (Halldórsson and Radhakrishnan 1997).
- Minimum dominating set is approximated by a classical greedy algorithm, originally proposed for the set covering problem. This algorithm guarantees an $\mathcal{O}(\log(\Delta))$ -approximation for an arbitrary graph (Chvátal 1979).
- The longest cycle problem is solved by a heuristic based on depth-first search or its improvement with local search (Chalupa et al. 2017).

The iterative improvement algorithms are used as longer-running procedures that improve an initially generated solution. These are, therefore, executed in a separate thread, and their intermediate results are updated within the user interface over the course of the computation:

- For the chromatic number and maximum clique problems, GraphCombEx uses the iterated greedy (IG) graph colouring algorithm (Culberson 1992; Culberson and Luo 1995), enhanced by an order-based randomised local search (RLS) for maximum clique, similar to RLS used for maximum independent set.
- For combined maximum independent set and minimum vertex clique covering, GraphCombEx employs the IG clique covering algorithm, combined with RLS for maximum independent set (Chalupa 2015).

For graphs with bounded numbers of vertices and edges, the GraphCombEx prototype also supports several different forms of their visualisation:

- centrality-based (also sometimes referred to as radial);
- grid-based;
- tree-based (simple hierarchical visualisation);
- circular.

Additionally, GraphCombEx supports visualisation of the currently longest cycle found, if the heuristic is used. The vertices of this cycle can also be rearranged as an “outer circle”, in a modification of the centrality-based visualisation. This is referred to as a cycle-based visualisation and was used in the study on the longest cycle problem (Chalupa et al. 2017).

In the centrality-based (radial) visualisation, the vertex with maximum degree is put in the middle and other vertices are placed in levels, based on the distance to the centre,

similarly to the GraViz prototype (Hawick 2010). Radial drawings are quite popular in graph visualisation in general (Bachmaier 2007; Brandes and Pich 2011). In the grid-based visualisation, the vertices are distributed simply in a regular grid. The third visualisation is the tree-based visualisation, in which the vertex with maximum degree is placed at the top and the other vertices are arranged hierarchically, based on the distance to this vertex. This was chosen as a rather simple implementation of one of the hierarchical drawings methods (Binucci et al. 2012; Holten 2006). The last drawing method supported is the circular drawing method (Baur and Brandes 2004).

Our prototype also visualises a bitmap, which represents the adjacency matrix of the graph. Visualisation of the shape of degree distribution is used without restrictions on the size of the graph. Export features for graph visualisation, as well as adjacency matrix visualisation and degree distribution in CSV format, are provided.

GraphCombEx was implemented in C++ using the Qt framework and is available under the GNU General Public License v3. It can, therefore, be used on several operating systems, including Windows, as well as Unix-like systems. General information on GraphCombEx is freely available online,¹ along with the source code repository on GitHub.²

4 Use cases for GraphCombEx and discussion

Evaluating software tools and prototypes with a suitable level of rigour is a complex task, especially when the software is designed to explore complex systems such as the networks studied here. We first illustrate a typical scenario of the use of GraphCombEx. Next, we will summarise a number of studies, in which GraphCombEx has already been successfully used to support the research projects. We hope that GraphCombEx will further be extended by more algorithms, problems to explore and used in further projects. This followed by a discussion to provide further context, as well as the way forward in extensions and applications of this prototype.

Figure 2 shows the main graphical user interface, illustrating the main features for four complex networks. Networks explored in (a)–(b) are a regular grid and a scale-free network that were generated, while (c)–(d) represent large-scale social networks from the SNAP network data repository (Leskovec and Krevl 2015). These illustrate the functionality in an overview, including the metrics provided, as well as the problems and analytics supported.

If GraphCombEx is built into a 64-bit binary, it can be used to explore relatively large graphs. Networks (c)–(d) within

Fig. 2 are large snapshots of social networks Pokec (Takac and Zabolovsky 2012) and LiveJournal (Backstrom et al. 2006; Leskovec et al. 2009). The latter consists of more than 4.8 million vertices, and a 64-bit version of GraphCombEx is required to process a network of this size.

One could argue that potential applications of GraphCombEx are numerous, which stems from its relatively general design. The combinatorial properties supported and heuristics provided indeed represent a very general view and are specifically beneficial for initial exploration of previously unseen networked data. However, specialised computational problems related to those supported by GraphCombEx range from scheduling and routing problems in operations research, through data analytics for social media (Brandes et al. 2012; Pattillo et al. 2012), to interaction exploration in bioinformatics in support of drug discovery (Csermely et al. 2013). This opens a number of pathways for applications of the tool.

A typical use case for GraphCombEx. We will illustrate the power of GraphCombEx using the simple small network presented in Listing 1. In principle, this use case can also be applied to larger networks, even though the tractability of more computationally intensive routines may depend on the size of the graph. After opening the graph file using *File > Open*, one will see the centrality-based visualisation by default. By using *File > Visualization type*, the graph visualisation mode can be changed. By using the options within the *Compute* menu, one can compute the on-demand characteristics of the network. For the network from Listing 1, the values presented in Table 1 will be obtained. GraphCombEx finds the two triangles in the networks and computes the mean clustering coefficient. Maximum clique size is 3, and the graph is 3-colourable. Maximum independent set and clique covering of size 3 are also computed. Dominating set of size 2 is also found. From the drawings, one can see that this dominating set is {*Bob*, *Emily*}. The longest cycle of size 5 is also found. All these values are proven optima for the network. If only bounds were computed, GraphCombEx would present intervals for each optimal value. An illustration of the 3-colouring and the longest cycle on 5 vertices in different graph layouts supported by GraphCombEx is given in Fig. 3.

Support for exploration of very large sparse graphs. GraphCombEx has been employed to support several previous studies on combinatorial optimisation in large sparse graphs. It has been partially used to support a previous study on detection of long cycles in real-world complex networks (Chalupa et al. 2017). This included social network samples, as well as protein–protein interactions. In addition, it supports generation of shortcut graphs used in solving the k -reachability problem, in which one aims to find the smallest set such that each vertex is in distance at most k to at least one ver-

¹ <http://davidchalupa.github.io/research/software/graphcombex.html>.

² <https://github.com/davidchalupa/GraphCombEx>.

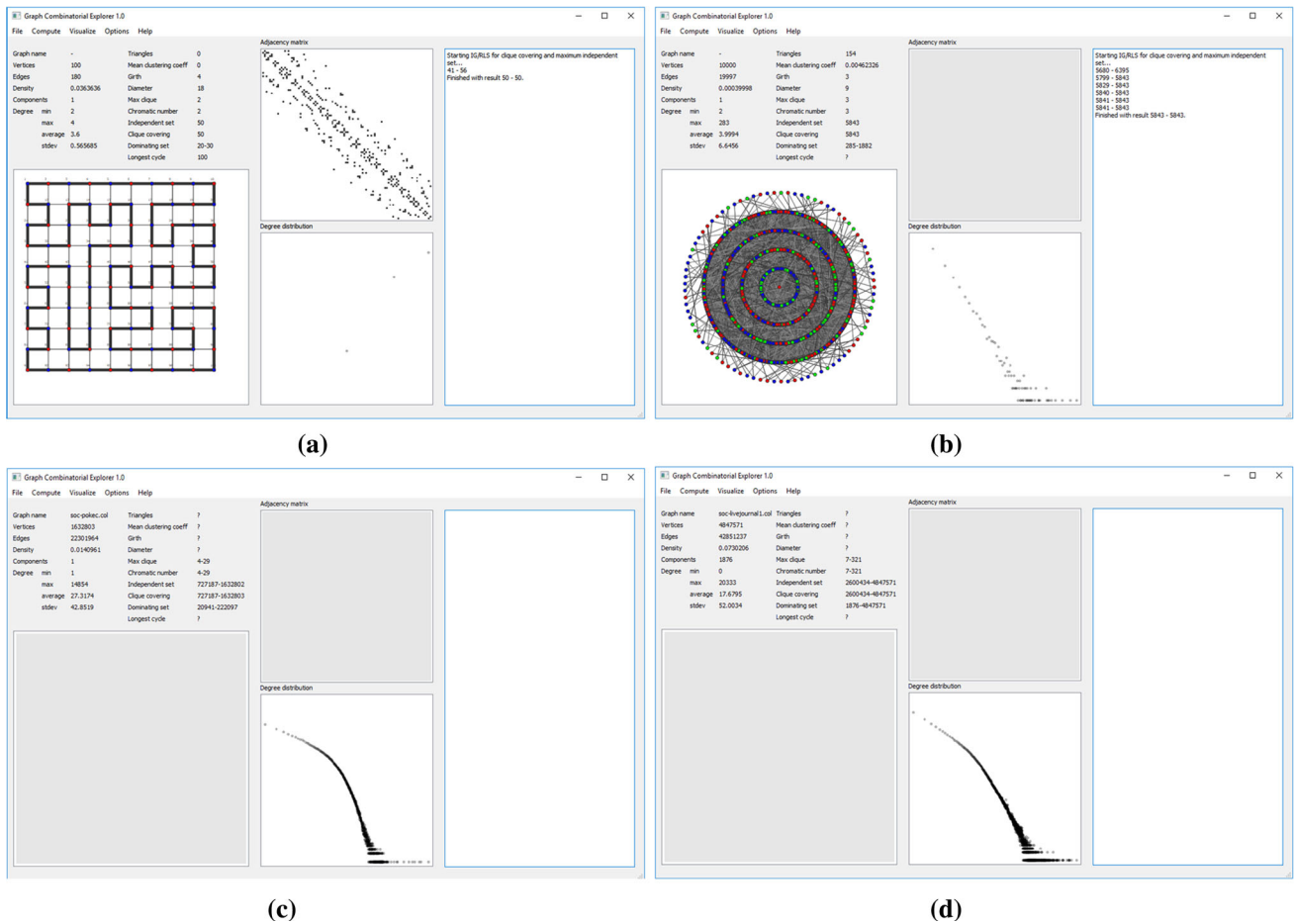


Fig. 2 Graphical user interface (GUI) of GraphCombEx prototype, exploring four different complex networks. Network (a) is a 10 × 10 regular grid, while network (b) is a scale-free network on 10, 000 vertices generated by Barabási–Albert model. Networks (c), (d) are large-scale social snapshots of networks Pokec and LiveJournal from the SNAP

network data repository. GUI shows a visualisation of the graph and its adjacency matrix, if the graph size is within a suitable range for the visualisation. Degree distribution shape is also depicted, and properties are computed for each graph, including ranges for the optima of the NP-hard problems studied

Table 1 The values of metrics and bounds computed by GraphCombEx for the simple social network from Listing 1

Metric/bound	Value
Triangles	2
Mean clustering coefficient	0.277778
Girth	3
Diameter	3
Maximum clique	3
Chromatic number	3
Independent set	3
Clique covering	3
Dominating set	2
Longest cycle	5

text of such a set (Chalupa and Blum 2017). The iterated greedy heuristic for vertex clique covering is also supported

in GraphCombEx (Chalupa 2014, 2015). A typical application area for these studies is analysis of large-scale social networks.

Analysis of scientific collaboration networks Collaboration networks are of a high interest within the scientific community (Newman 2001). Combinatorial properties can offer further insights into the phenomena of scientific collaboration patterns and their understanding. In particular, the *k*-reachability problem has already been studied for several scientific collaboration networks, with partial support of GraphCombEx. It has been revealed that for most collaboration networks studied, there is a single vertex that is roughly in a distance between 8 and 11 to other vertices of the collaboration networks (Chalupa and Blum 2017).

Exploration of protein–protein interactions One application area of a high interest for GraphCombEx is the study

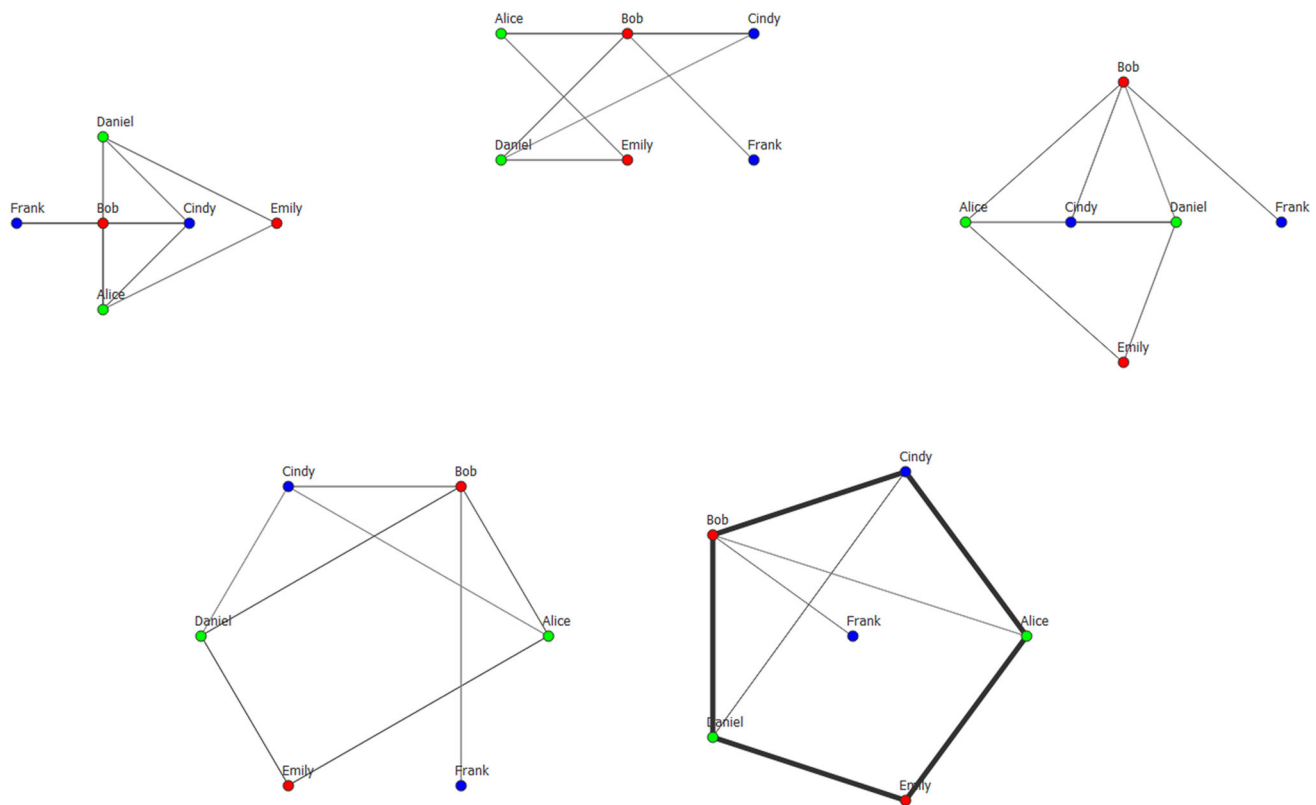


Fig. 3 An illustration of the visualisation types provided by GraphCombEx, applied to the simple social network from Listing 1. GraphCombEx supports different layouts, as well as highlighting of the colouring and the longest cycle identified

of protein–protein interaction networks. A general study of protein–protein interactions and combinatorial optimisation has been conducted with the use of GraphCombEx (Chalupa 2016). Further studies can be conducted, especially given the rich data currently available, e.g. at the UCLA database of interacting proteins (Salwinski et al. 2004; Xenarios et al. 2000, 2001, 2002). This data on protein–protein interactions have also been tested and used within the framework of GraphCombEx (Chalupa et al. 2017).

Modelling of randomised lattice-based systems and utility distribution networks One of the interesting applications of GraphCombEx, that are still left to be largely explored, is investigation of lattice-based network models. These potentially have structures similar to power networks (Hawick 2012a) or water distribution networks (Hawick 2012b). Effects of edge rewiring, insertion or deletion on long cycles, flows and robustness of these networks can be explored using efficient mechanisms, expanding on the existing studies (Chalupa and Blum 2017; Chalupa et al. 2017). Adding the spatial information for these networks is another interesting future direction.

This list is not necessarily exhaustive and highlights that the potential of GraphCombEx as a multi-perspective complex network analysis and visualisation tool. Graph-

CombEx has been demonstrated as useful in analysis of social network samples of different scales or analysis and visualisation of protein–protein interactions. Other applications can follow, including utility distribution networks, further exploration of biological networks such as gene expression data, or scheduling and transportation networks in operations research. In addition, there is further space for extending GraphCombEx with new combinatorial properties and new algorithms, including local search and evolutionary algorithms or swarm intelligence algorithms approaches.

Furthermore, these algorithms and applications can further be facilitated by integration with integer linear programming (ILP) problem solving tools. As the first example, GraphCombEx currently provides the functionality to export ILP-based representation of the minimum dominating set problem, along with its LP relaxation. MPS linear programming instance format is currently used for this export. MILP software can be used to solve these instances, e.g. the CBC branch-and-cut solver from the COIN-OR package (Bonami et al. 2008; Linderoth and Lodi 2011). In the broader context of research software, further applications in, e.g. cyber-physical systems, may also be of a high interest (Khosiawan et al. 2018).

5 Conclusions

We presented a prototype of Graph Combinatorial Explorer (abbr. GraphCombEx), which is an open-source software platform aimed at unified exploration of combinatorial properties of large networks. GraphCombEx takes the specific properties of the numerous NP-hard combinatorial optimisation problems in networks into account in its design. The tool currently supports graphs with up to 5 million vertices, even though this bound is extensible and has been chosen for practical reasons such as memory management.

GraphCombEx employs several heuristic procedures to find high-quality solutions to NP-hard graph problems. These problems currently include maximum clique, chromatic number, maximum independent set, minimum vertex clique covering, minimum dominating set, as well as the longest simple cycle problem. However, the tool is designed with extensibility by further problems and algorithms in mind, including iterative improvement algorithms. The core functionality consists mainly of constructive heuristics enhanced by efficient data structures that can be used to find good sub-optimal solutions and bounds for very large graphs.

Complex networks with no explicit spatial properties are particularly addressed in the design of the prototype of this software tool. GraphCombEx has been successfully used as a support tool in a number of studies, including exploration of combinatorial properties of protein–protein interactions (Chalupa 2016), detection of long simple cycles (Chalupa et al. 2017) or the k -reachability problem in small-world complex networks (Chalupa and Blum 2017), including social networks or scientific collaboration networks.

Possible future applications of GraphCombEx include analysis of massive public social networks (Takac and Zabovsky 2012), further exploration of protein–protein interaction networks (Hawick 2014), metabolic pathways (Becker and Rojas 2001) or utility distribution networks (Hawick 2012a, b).

Acknowledgements The core work presented in this paper was conducted while the first author was at the University of Hull.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

Abualigah LM, Khader AT, Hanandeh ES, Gandomi AH (2017) A novel hybridization strategy for krill herd algorithm applied to clustering techniques. *Appl Soft Comput* 60:423–435

- Abualigah LM, Khader AT, Hanandeh ES (2018) A hybrid strategy for krill herd algorithm with harmony search algorithm to improve the data clustering. *Intell Decis Technol* 12(1):3–14
- Albert R, Barabási AL (2002) Statistical mechanics of complex networks. *Revi Mod Phys* 74(1):47–97
- Bachmaier C (2007) A radial adaptation of the sugiyama framework for visualizing hierarchical information. *IEEE Trans Vis Comput Grap* 13(3):583–594
- Backstrom L, Huttenlocher D, Kleinberg J, Lan X (2006) Group formation in large social networks: membership, growth, and evolution. In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp 44–54
- Barabási AL, Albert R (1999) Emergence of scaling in random networks. *Science* 286(5439):509–512
- Bastiani M, Heymann S, Jacomy M et al (2009) Gephi: an open source software for exploring and manipulating networks. In: *Proceedings of the third international conference on weblogs and social media*, ICWSM vol 8, pp 361–362
- Baur M, Brandes U (2004) Crossing reduction in circular layouts. In: *International workshop on graph-theoretic concepts in computer science*, Springer, pp 332–343
- Becker MY, Rojas I (2001) A graph layout algorithm for drawing metabolic pathways. *Bioinformatics* 17(5):461–467
- Binucci C, Brandes U, Di Battista G, Didimo W, Gaertler M, Palladino P, Patrignani M, Symvonis A, Zweig K (2012) Drawing trees in a streaming model. *Inf Process Lett* 112(11):418–422
- Bonami P, Biegler LT, Conn AR, Cornuéjols G, Grossmann IE, Dand Laird J C, Lee Lodi A, Margot F, Sawaya N et al (2008) An algorithmic framework for convex mixed integer nonlinear programs. *Discret Optim* 5(2):186–204
- Brandes U, Pich C (2011) More flexible radial layout. *J Graph Algorithms Appl* 15(1):157–173
- Brandes U, Indlekofer N, Mader M (2012) Visualization methods for longitudinal social networks and stochastic actor-oriented modeling. *Soc Netw* 34(3):291–308
- Branke J, Bucher F, Schmeck H (1996) Using genetic algorithms for drawing undirected graphs. In: *The third nordic workshop on genetic algorithms and their applications*, pp 193–206
- Brélez D (1979) New methods to color vertices of a graph. *Commun ACM* 22(4):251–256
- Chalupa D (2014) Partitioning networks into cliques: a randomized heuristic approach. *Inf Sci Technol Bull ACM Slovak* 6(3):1–8
- Chalupa D (2015) Construction of near-optimal vertex clique covering for real-world networks. *Comput Inform* 34(6):1397–1417
- Chalupa D (2016) On combinatorial optimisation in analysis of protein–protein interaction and protein folding networks. In: Squillero G, Burelli P (eds) *Proceedings of the 19th european conference on applications of evolutionary computation, EvoApplications '16*, Springer, Lecture notes in computer science, vol 9597, pp 91–105
- Chalupa D, Blum C (2017) Mining k -reachable sets in real-world networks using domination in shortcut graphs. *J Comput Sci* 22:1–14
- Chalupa D, Balaghan P, Hawick KA, Gordon NA (2017) Computational methods for finding long simple cycles in complex networks. *Knowl Based Syst* 125:96–107
- Chvátal V (1979) A greedy heuristic for the set-covering problem. *Math Oper Res* 4(3):233–235
- Csardi G, Nepusz T (2006) The igraph software package for complex network research. *InterJ Complex Syst* 1695(5):1–9
- Csermely P, Korcsmáros T, Kiss HJM, London G, Nussinov R (2013) Structure and dynamics of molecular networks: a novel paradigm of drug discovery: a comprehensive review. *Pharmacol Ther* 138(3):333–408
- Culberson JC (1992) Iterated greedy graph coloring and the difficulty landscape. *Tech. Rep. TR92-07*, University of Alberta
- Culberson JC, Luo F (1995) Exploring the k -colorable landscape with iterated greedy. In: Johnson DS, Trick M (eds) *Cliques, color-*

- ing and satisfiability: second DIMACS implementation challenge. American Mathematical Society, Providence, pp 245–284
- Czech W, Dzwiniel W, Goryczka S, Arodz T, Dudek AZ (2011) Exploring complex networks with graph investigator research application. *Comput Inform* 30(2):381–410
- Ellson J, Gansner E, Koutsofios L, North SC, Woodhull G (2002) Graphviz - open source graph drawing tools. In: Mutzel P, Jünger M, Leipert S (eds) *Graph drawing*, vol 2265. Lecture notes in computer science. Springer, Berlin, pp 483–484
- Garey MR, Johnson DS (1983) Crossing number is np-complete. *SIAM J Algebr Discret Methods* 4(3):312–316
- Halldórsson MM, Radhakrishnan J (1997) Greed is good: approximating independent sets in sparse and bounded-degree graphs. *Algorithmica* 18(1):145–163
- Hawick KA (2010) Interactive graph algorithm visualization and the GraViz prototype. Tech. Rep. CSTN-061, Computer Science, Massey University, Albany, North Shore 102-904, Auckland, New Zealand
- Hawick KA (2012a) Betweenness centrality metrics for assessing electrical power network robustness against fragmentation and node failure. In: *Proceedings of the International Conference on power and energy systems (EuroPES 2012)*, Napoli, Italy., pp 186–193
- Hawick KA (2012b) Water distribution network robustness and fragmentation using graph metrics. In: *Proceedings of the international conference on water resource management (AfricaWRM 2012)*, Gabarone, Botswana, 762-037, pp 304–310
- Hawick KA (2014) Centrality metrics for comparing protein-protein interaction networks with synthesized nk systems. In: *Proceedings of the IASTED international conference on biomedical engineering*, Zurich, Switzerland, pp 1–8
- Holten D (2006) Hierarchical edge bundles: visualization of adjacency relations in hierarchical data. *IEEE Trans Vis Comput Graph* 12(5):741–748
- Johnson DS, Trick M (1996) Cliques, coloring, and satisfiability: second DIMACS implementation challenge. American Mathematical Society, Providence
- Karp RM (1972) Reducibility among combinatorial problems. In: Miller R, Thatcher J (eds) *Proceedings of a symposium on the complexity of computer computations*, Plenum Press, New York, NY, pp 85–103
- Khosiawan Y, Park Y, Moon I, Nilakantan JM, Nielsen I (2018) Task scheduling system for UAV operations in indoor environment. *Neural Comput Appl*. <https://doi.org/10.1007/s00521-018-3373-9>
- Leskovec J, Krevl A (2015) {SNAP Datasets}: {Stanford} large network dataset collection
- Leskovec J, Sosič R (2016) Snap: a general-purpose network analysis and graph-mining library. *ACM Trans Intell Syst Technol (TIST)* 8(1):1
- Leskovec J, Lang KJ, Dasgupta A, Mahoney MW (2009) Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Math* 6(1):29–123
- Li Z, Janardhanan MN, Tang Q, Nielsen P (2016) Co-evolutionary particle swarm optimization algorithm for two-sided robotic assembly line balancing problem. *Adv Mech Eng* 8(9):1687814016667907
- Linderoth JT, Lodi A (2011) MILP software. *Wiley encyclopedia of operations research and management science*, vol 5, pp 3239–3248. <https://doi.org/10.1002/9780470400531.eorms0524>.
- Morgenstern C (1991) Improved implementations of dynamic sequential coloring algorithms. Tech. Rep. CoSc-91-4, Texas Christian University, Department of Computer Science
- Newman MEJ (2001) The structure of scientific collaboration networks. *Proc Natl Acad Sci* 98(2):404–409
- Pattillo J, Youssef N, Butenko S (2012) Clique relaxation models in social network analysis. In: Thai MT, Pardalos PM (eds) *Handbook of optimization in complex networks*. Springer, Berlin, pp 143–162
- Rosete-Suárez A, Ochoa-Rodríguez A, Sebag M (1999) Automatic graph drawing and stochastic hill climbing. *Proceedings of the genetic and evolutionary computation conference 2*:1699–1706
- Salwinski L, Miller CS, Smith AJ, Pettit FK, Bowie JU, Eisenberg D (2004) The database of interacting proteins: 2004 update. *Nucleic Acids Res* 32(suppl 1):D449–D451
- Schreiber F, Dwyer T, Marriott K, Wybrow M (2009) A generic algorithm for layout of biological networks. *BMC Bioinform* 10(1):1
- Sugiyama K (2002) *Graph drawing and applications for software and knowledge engineers*, vol 11. World Scientific, Singapore
- Takac L, Zabolovsky M (2012) Data analysis in public social networks. In: *International scientific conference and international workshop present day trends of innovations*, pp 1–6
- Tamassia R (2013) *Handbook of graph drawing and visualization*. CRC Press, Boca Raton
- Turner JS (1988) Almost all k-colorable graphs are easy to color. *J Algorithms* 9(1):63–82
- Watts DJ, Strogatz SH (1998) Collective dynamics of "small-world" networks. *Nature* 393(6684):440–442
- Xenarios I, Rice DW, Salwinski L, Baron MK, Marcotte EM, Eisenberg D (2000) Dip: the database of interacting proteins. *Nucleic Acids Res* 28(1):289–291
- Xenarios I, Fernandez E, Salwinski L, Duan XJ, Thompson MJ, Marcotte EM, Eisenberg D (2001) Dip: the database of interacting proteins: 2001 update. *Nucleic Acids Res* 29(1):239–241
- Xenarios I, Salwinski L, Duan XJ, Higney P, Kim SM, Eisenberg D (2002) Dip, the database of interacting proteins: a research tool for studying cellular networks of protein interactions. *Nucleic Acids Res* 30(1):303–305

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.