DCC 2017
Snowbird, Utah

Symmetry compressible graphs

Uroš Čibej and Jurij Mihelič
University of Ljubljana, Slovenia

Univerza v Ljubljani Fakulteta za računalništvo in informatiko

Summary

In this paper we investigate the possibilities to construct a compression method using graph symmetries. The main results of this work are as follows:

1. Definition of Symmetry-compressible graphs, i.e. graphs that can be compressed using automorphisms.
2. Near Symmerty-Compressible graphs, a class of graphs that can be efficiently transformed into a SCG.
3. Definition of two algorithms that can compress graphs using the above two concepts.
4. Empirical evaluation of the algorithms on real networks shows promising results, opening many possibilities for compression of other types of data.

Challenges

1. Symmetries indentify redundancies in graphs. In practice it is easy to identify them, but they have not been used for data compression.
   How do we define and efficiently exploit them?
2. Real graphs do not exhibit many global symmetries. But "almost" symmetries exist. Can we define what "almost" means? Is this a useful concept for practical applications?

Main concepts

- G The graph is a set of pairs, i.e., edges.
- π The symmetry is given by a permutation of the graph vertices. The size of of the symmetry is the number of pairs needed to represent it.
- G^π Residual graph: the graph that is expanded with a symmetry into the original graph
- $H \oplus G$ Symmetric difference of a graph - models the transformation of G into (more symmetric) graph H

Definitions

Symmetry-compressible (SC) graphs are

$G \in SC$ if $\exists \pi : |G^\pi| + |\pi| < |G|$

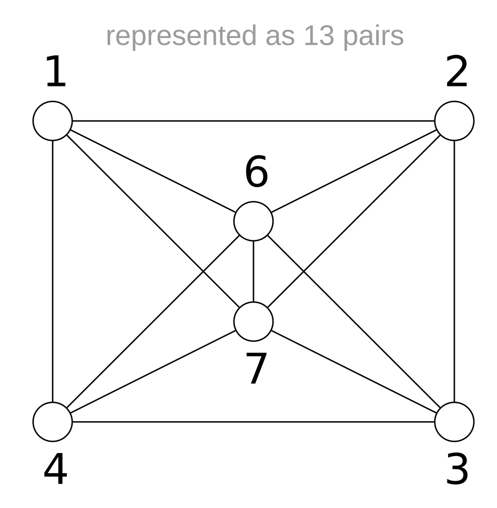
Near Symmetry-Compressible (NSC) graphs are

$G \in NSC$ if $\exists H, \pi : |H^\pi| + |\pi| + |H \oplus G| < |G|$

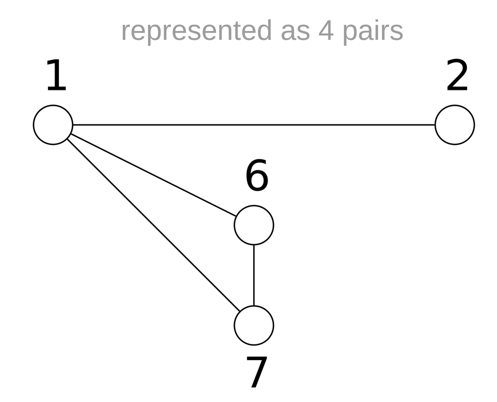
Examples

Two examples are shown below. The first example is a symmetry compressible graph (left), on the right is the residual graph and the corresponding symmetry (shown in cycle notation). The second example is a near-symmetry compressible graph, which has no symmetries, but when adding the edge (3,6) it becomes symmetry-compressible, compensating for the cost of an edge addition.

SC graph



$|G| = 13$

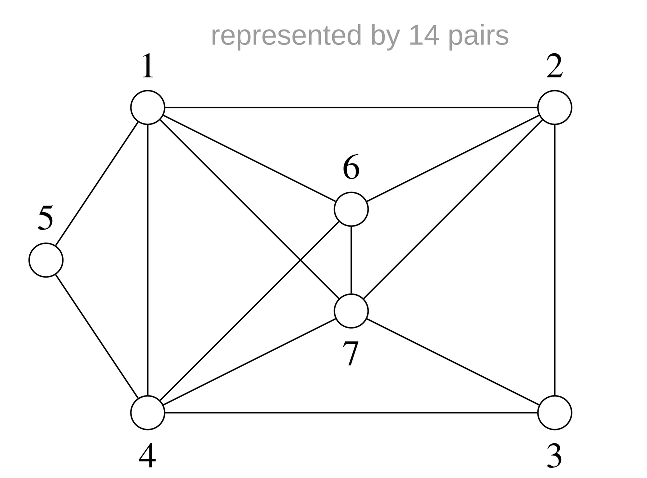


$\pi = (1234)$

represented as 3 pairs, i.e. (1,2),(2,3), and (3,4)

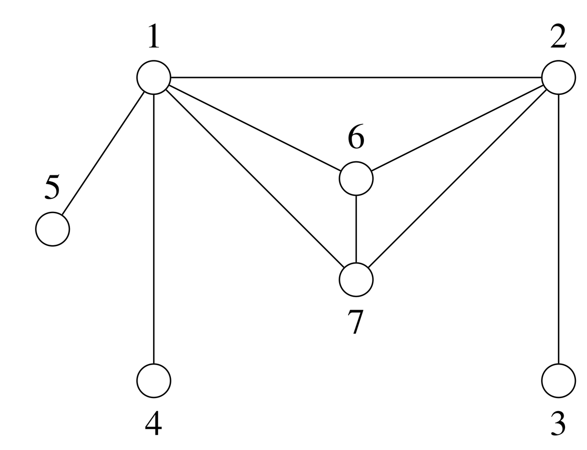
$$|G^\pi| + |\pi| = 4 + 3 = 7 < 13$$

NSC graph



$|G| = 14$

$H \oplus G = \{(3,6)\}$



$|H^\pi| = 9$

$\pi = (14)(23)$

$$|G \oplus H| + |\pi| + |H^\pi| = 1 + 2 + 9 = 12 < 14$$

uros.cibej@fri.uni-lj.si

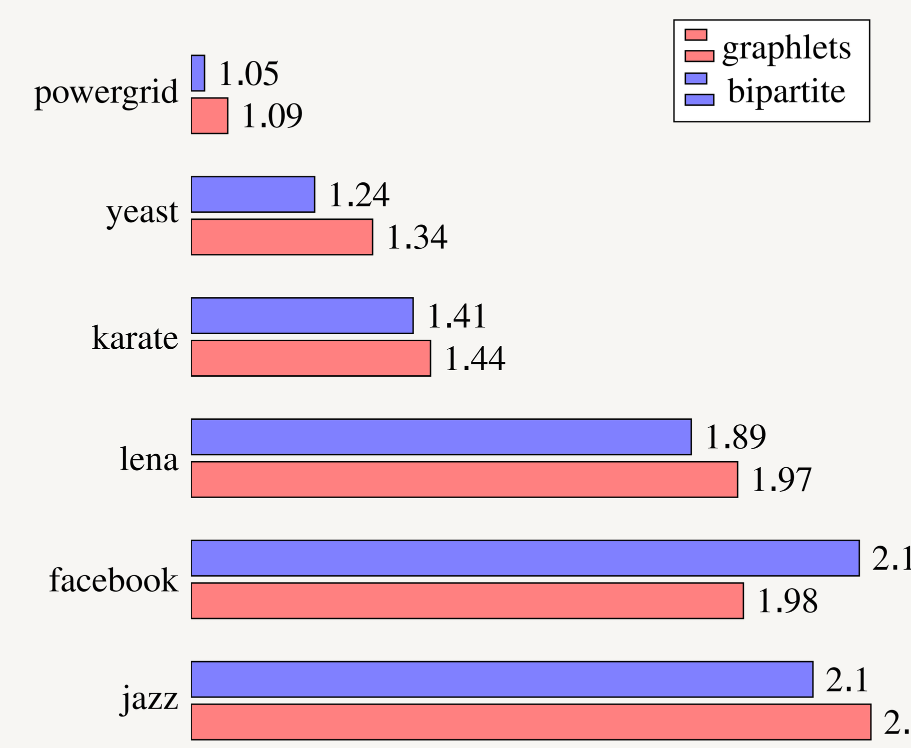
https://github.com/chibo17/dcc2017

Algorithms

1. The first algorithm searches for symmetry-compressible graphlets of sizes 4-9 (these graphlets have been found by exhaustive search of small graphs). When found, the graphlets are encoded with their symmetries, and removed from the graph. This process is repeated until no such graphlet is present in the graph
2. The second algorithm searches for dense bipartite subgraphs. By adding edges these subgraph made complete bipartite subgraphs and then encoded with a known symmetry (very efficient). These graphs are then removed from the graph and the process is repeated.

Experiments

We conducted the experiment on 6 different real graphs and below the compression ratios are shown for both algorithms. For more tree-like graphs, there is no compression, whereas for denser and larger graphs, the compression ratio is much better.



Future work

Similar techniques can be used for other types of data. The most promising are images, that can simply be transformed into graphs and many naturally occuring symmetries in images can be efficiently exploited. The algorithms presented here use only simple graphs, more advanced techniques should use more complex patterns.