Algorithms for finding all maximal bicliques of a graph

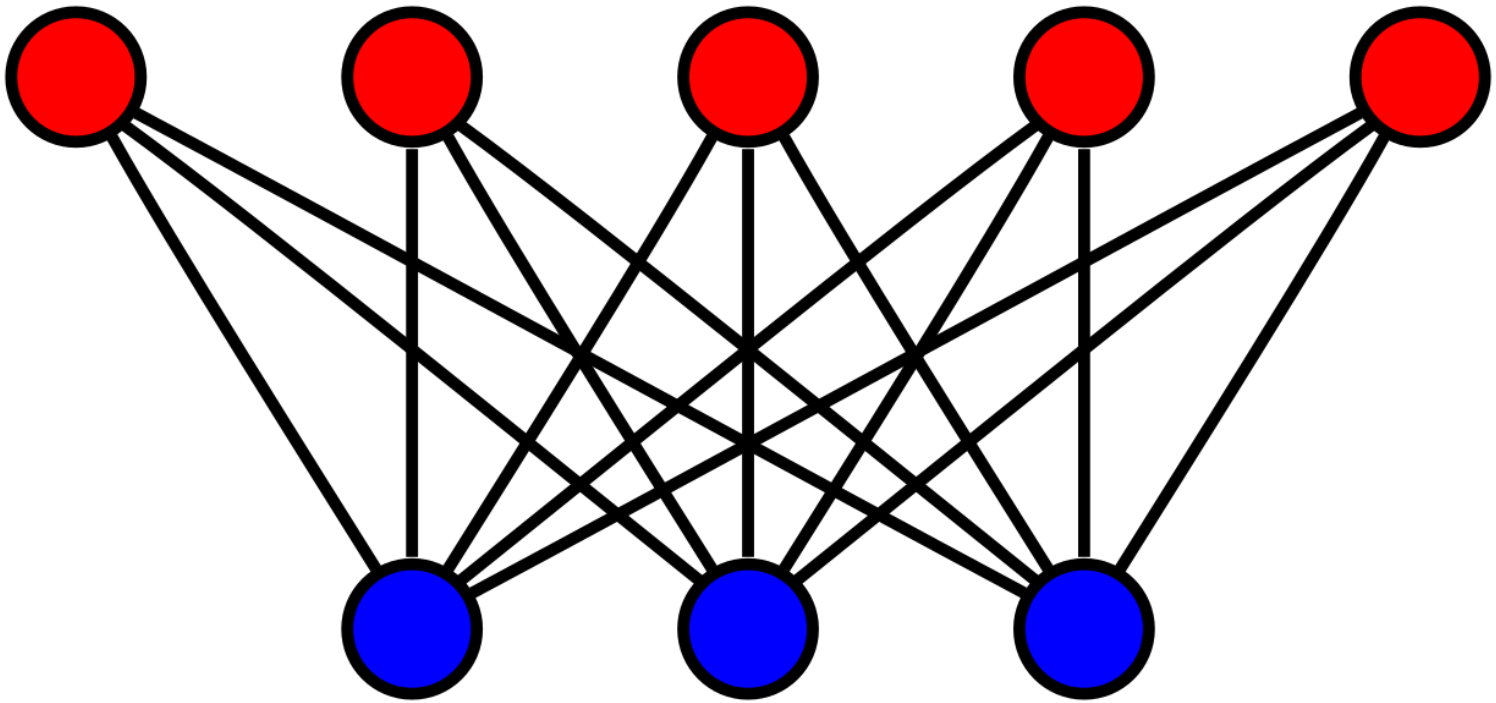
Mykyta Ielanskyi
Roland Koppenberger
Hadi Sanaei

June 27, 2019

Overview

Basic Concept

So, there are graphs and they have bicliques...

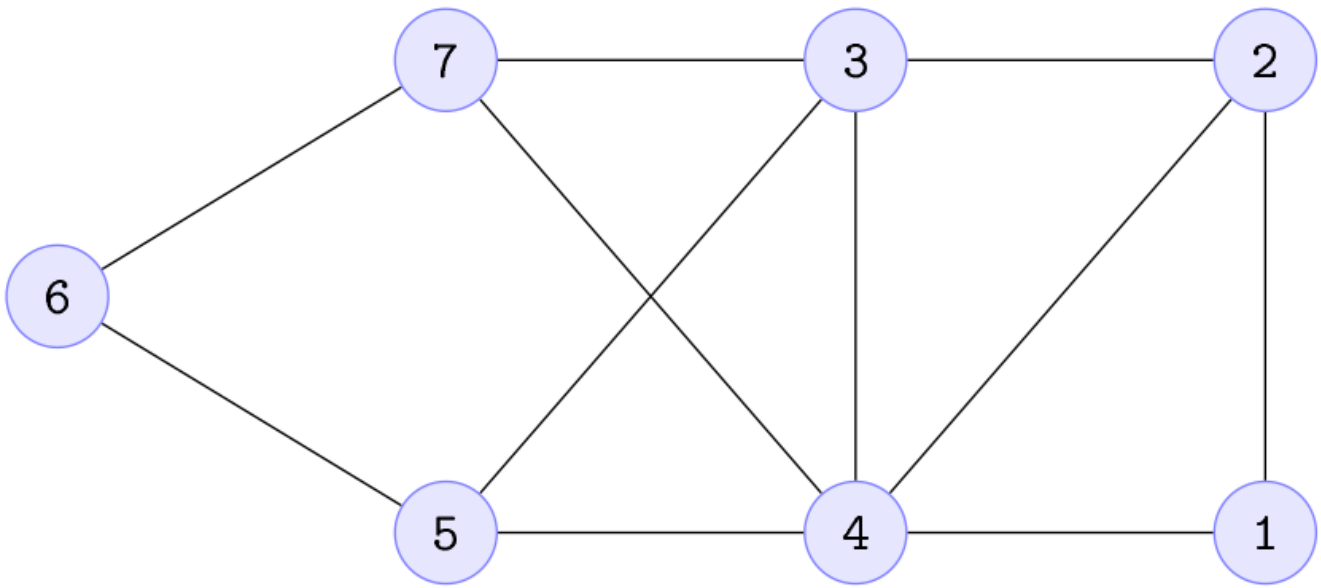


Bicliques

- Maximum Biclique Problem
- Has many applications, including clustering and m

Example

Compute all maximal bicliques of the following graph:



Algorithm LEX

- The most basic algorithm
- Relies on two biclique operations:
  - Absorption
  - Consensus Adjunction
- Main idea:
  - a seed list of bicliques is created
  - absorption applied for each biclique against every other biclique
  - consensus adjunction applied to create new bicliques
- non polynomial complexity in *both* graph size *and* the number of maximum bicliques!

Algorithm LEX

Let $B_1 = (X_1, Y_1)$ and $B_2 = (X_2, Y_2)$ be two bicliques of G. We say that B_1 *absorbs* or *contains* B_2 if $X_2 \subseteq X_1$ and $Y_2 \subseteq Y_1$, or if $X_2 \subseteq Y_1$ and $Y_2 \subseteq X_1$.

If $Y_1 \cap Y_2 \neq \emptyset$, we call $(X_1 \cup X_2, Y_1 \cap Y_2)$ one of the *consensuses* of B_1 and B_2. Similarly, each of those pairs of subsets $(X_1 \cap X_2, Y_1 \cup Y_2)$, $(Y_1 \cup X_2, X_1 \cap Y_2)$, $(X_1 \cup Y_2, Y_1 \cap X_2)$ which define bicliques (i.e., which involve two non-empty subsets) are *consensuses* of B_1 and B_2. In this way, a pair of bicliques may have 0, 1, 2, 3 or 4 consensuses.

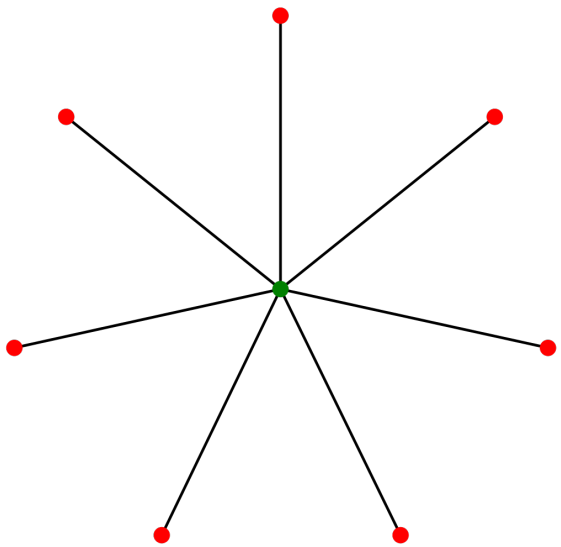
Algorithm MBEA

(not implemented yet)

Algorithm MICA

Modular input consensus algorithm (MICA)

- **Modular consensus**: sequence of operations performed on each pair of bicliques in the repeat loop.
- **Input**: starting with a special input set of maximal bicliques that covers the stars of the graph.



Algorithm MICA

The algorithm has two steps (n is the number of vertices):

- **Start** with a list C_0 of at most n maximal bicliques that covers the stars of the graph. Let $C := C_0$.
- **Repeat**: For every pair of distinct bicliques $B_1 \in C_0$ and $B_2 \in C$, if B_1 and B_2 have a consensus B_3 which is not absorbed by any member of C, then extend B_3 to a maximal biclique B_4, and add B_4 to C.

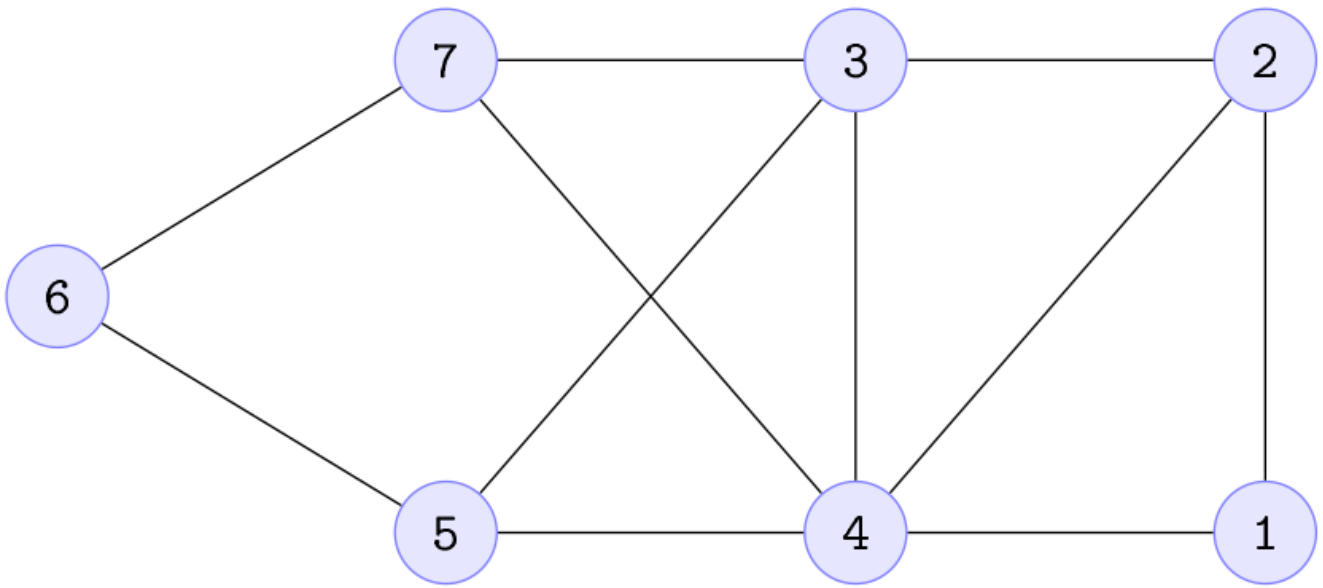
Algorithm MICA

Some subtantial improvements for implementation:

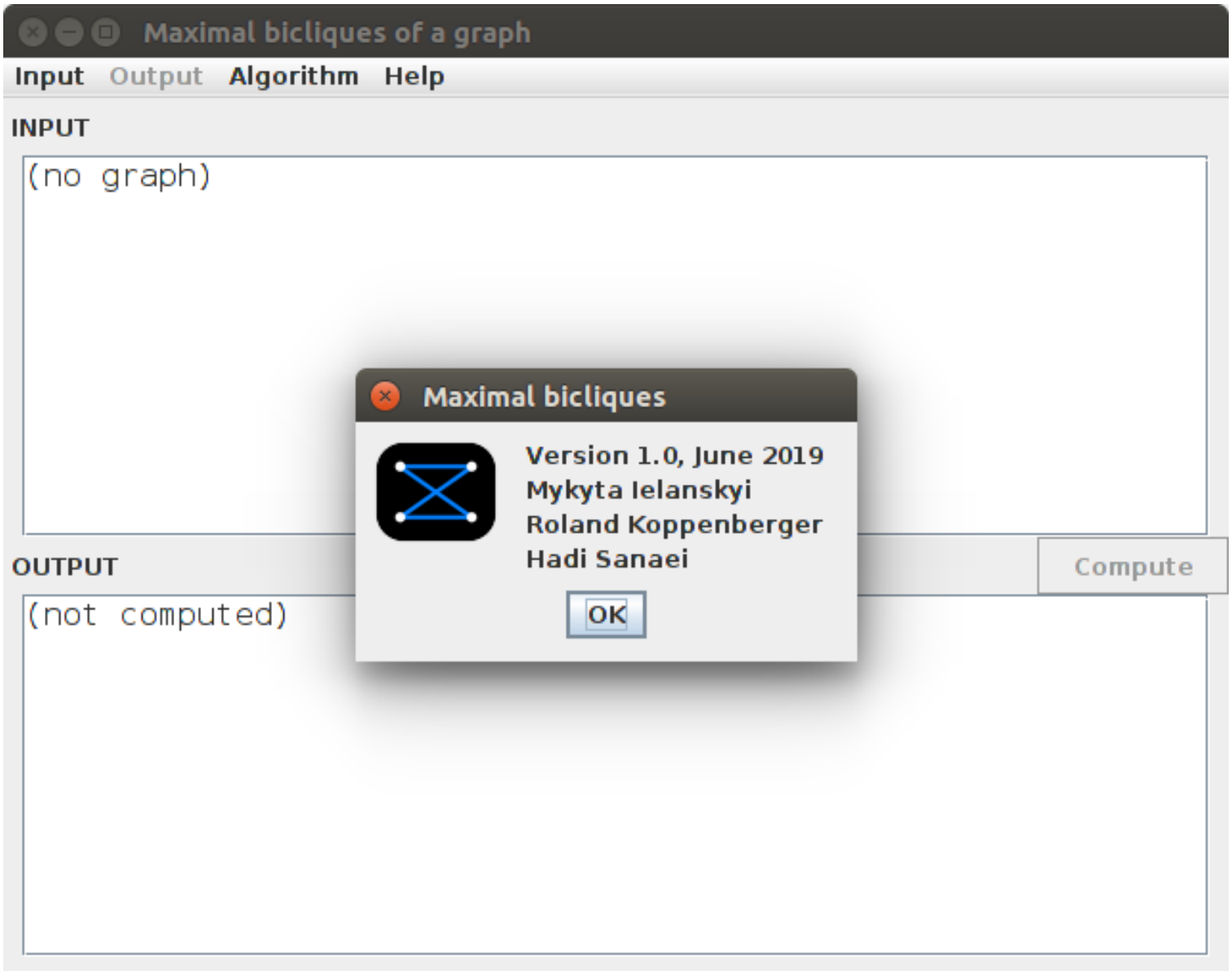
- Many consensus adjunction operations produce the same output.
- $X \subset \mathit{Vertices}$: $\Gamma(X) :=$ set of vertices adjacent to each vertex in X. For all $X \subset \mathit{Vertices}$: $(\Gamma^2(X), \Gamma(X))$ is a maximal biclique.
- Proceed in stages, consider only newly added bicliques in each stage.
- Time complexity: incrementally polynomial, i.e. time for computation of next new maximal biclique is bounded polynomially in number of vertices and number of maximal bicliques computed so far.

Example

Again: *Compute all maximal bicliques of the following graph:*



GUI

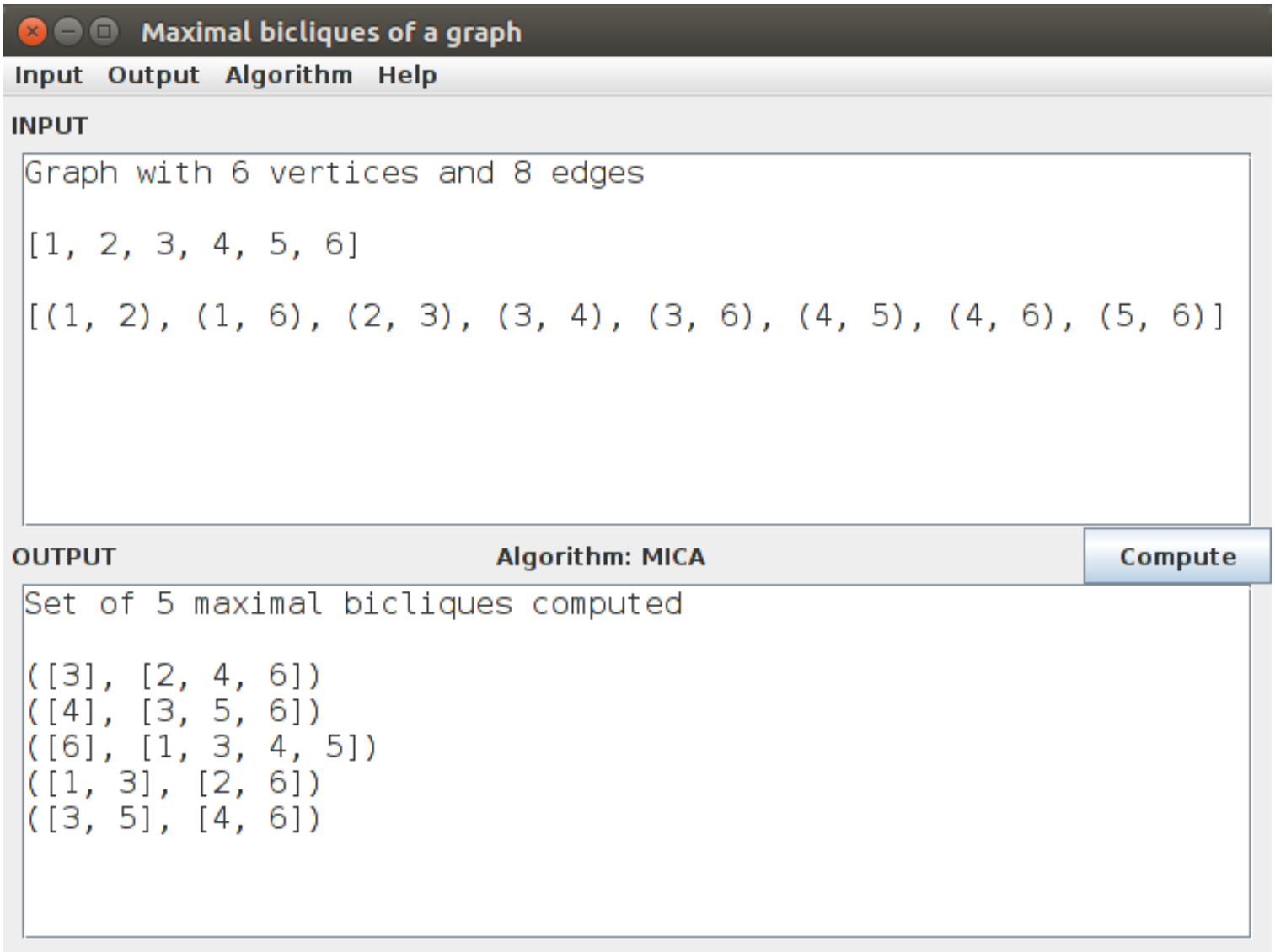

Maximal bicliques of a graph
Input
Output
Algorithm
Help
INPUT
(no graph)
OUTPUT
(not computed)
Compute
Maximal bicliques
Version 1.0, June 2019
Mykyta Ielanskyi
Roland Koppenberger
Hadi Sanaei
OK

GUI

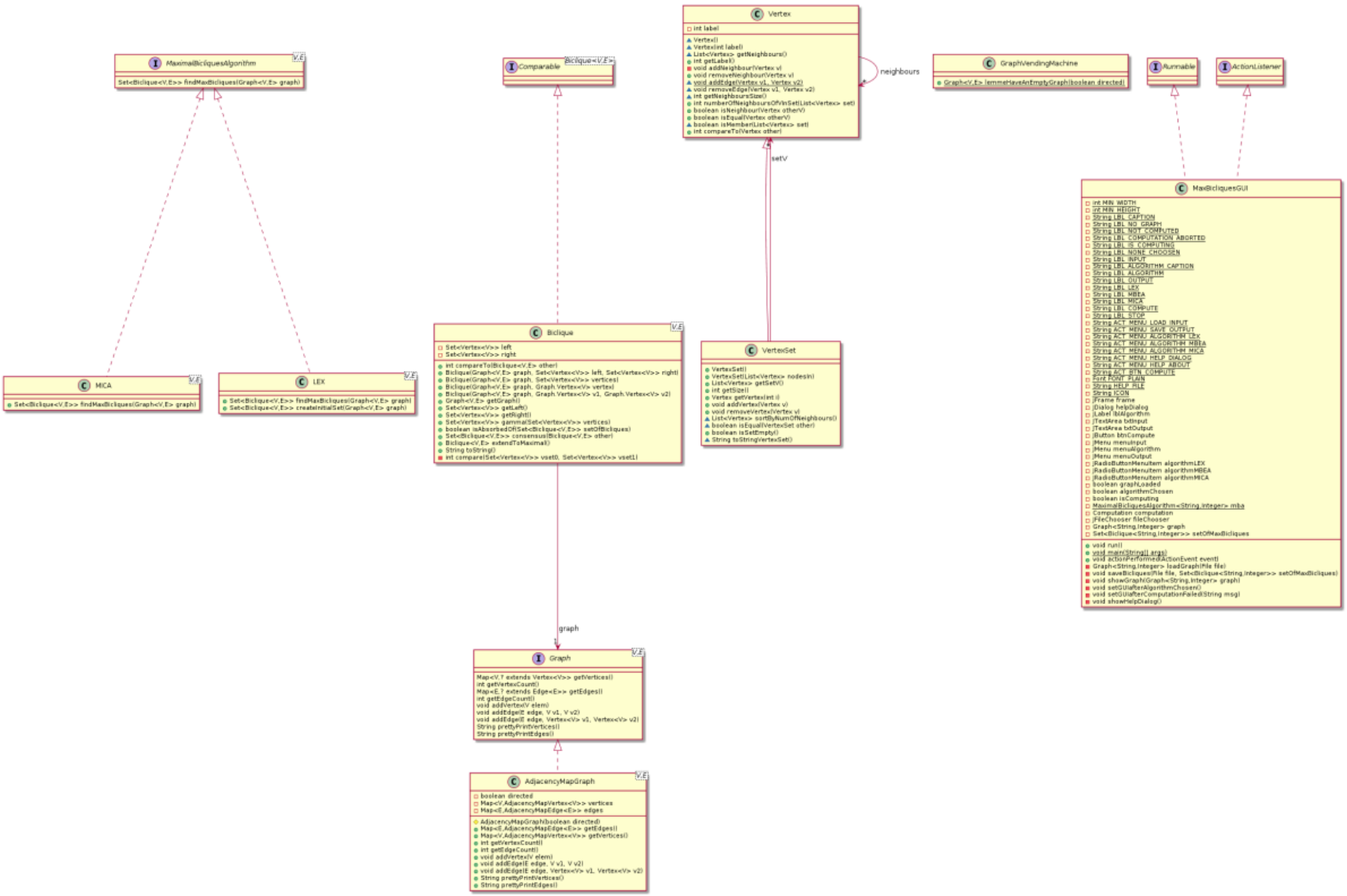
What can one do with the graphical user interface?

- Load a graph from a text file.
- Select an algorithm for computation.
- Compute the set of maximal bicliques.
- Save the result of the computation in a text file.

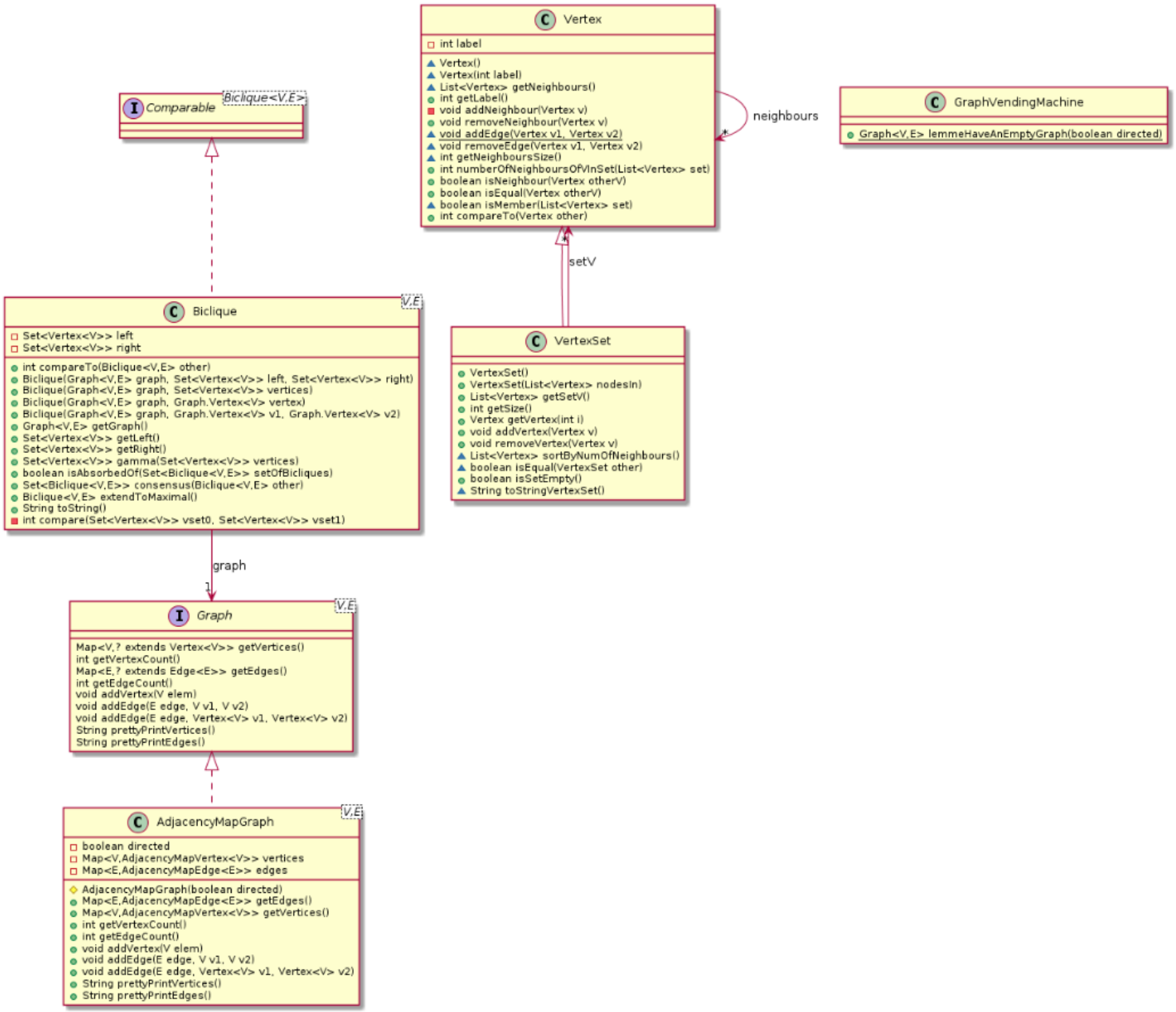
GUI


Maximal bicliques of a graph
Input Output Algorithm Help
INPUT
Graph with 6 vertices and 8 edges
[1, 2, 3, 4, 5, 6]
[(1, 2), (1, 6), (2, 3), (3, 4), (3, 6), (4, 5), (4, 6), (5, 6)]
OUTPUT
Algorithm: MICA
Compute
Set of 5 maximal bicliques computed
([3], [2, 4, 6])
([4], [3, 5, 6])
([6], [1, 3, 4, 5])
([1, 3], [2, 6])
([3, 5], [4, 6])

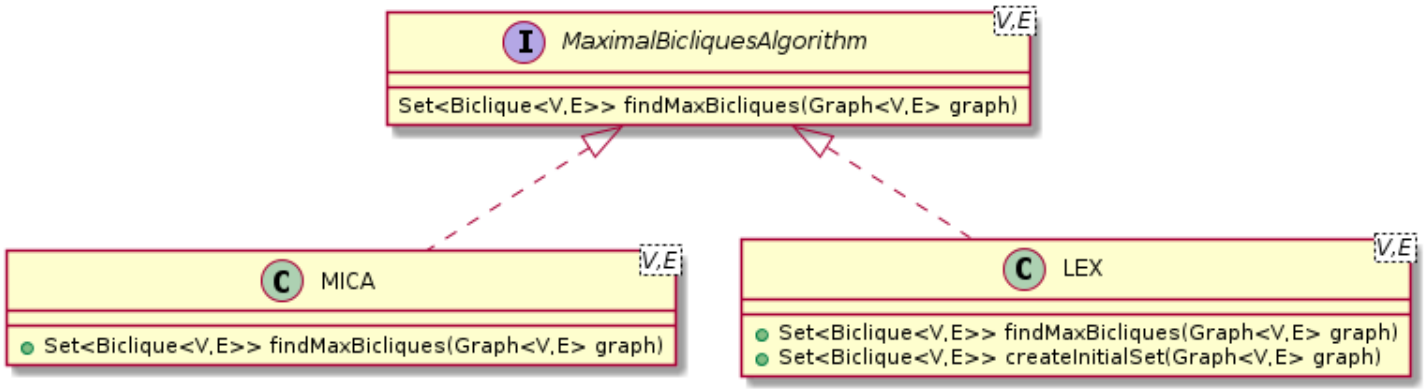
package bicliques



package bicliques.graphs



package bicliques.algorithms



package bicliques.ui

Practical demonstration

We conclude with some practical computations.