

# Optimisation Assignment

Matt Ingram

February 21, 2019

## 1 Installation

A few small prerequisites need to be installed before running my program. These are: Python 3, GNU Multiple Precision Arithmetic Library (GMP), and the SoPlex solver. I shall include the GMP and SoPlex archived files (for installation on a Unix-based system) in my final submission. Alternatively, follow the blue links to go to their respective download pages. To install these dependencies, please follow the instructions below:

- **GNU Multiple Precision Arithmetic Library**

Please change directory to where the *gmp-6.1.2.tar.lz* file is located and run the following commands:

```
1  lzip -d gmp-6.1.2.tar.lz
2  tar -xf gmp-6.1.2.tar
3  cd gmp-6.1.2
4  ./configure
5  make
6  make check
7  make install
```

- **SoPlex Solver**

Please change directory to where the *soplex-4.0.1.tgz* file is located and run the following commands:

```
1  tar -xf soplex-4.0.1.tgz
2  cd soplex-4.0.1
3  mkdir build
4  cd build
5  cmake <path/to/SoPlex> -DGMP=true
6  make
7  make test
8  make install
```

- **Pyomo**

The easiest way to install Pyomo is to use Python's built-in package manager, *pip*. Once *pip* has been installed, please run the following command:

```
1  pip install pyomo
```

Should you run into any issues installing any of these dependencies, please consult the relevant INSTALL files and documentation.

## 2 Usage

### 2.1 Directory Structure

Once the dependencies have been installed, the directory structure should look something like Figure 1 (varying slightly if you installed the dependencies in a different location):

**Note:** The program will **not** work unless the four folders (*graphs*, *models*, *settings*, *solutions*) are present. This is what is contained in each folder:

- **/graphs:** This folder contains the input graph files. Graph files should be saved with a *.txt* extension, and follow the following format:

```
[V] [E]
[v11] [v12]
[v21] [v22]
...
[vE1] [vE2]
```

| Name             | Date Modified        | Size    | Kind              |
|------------------|----------------------|---------|-------------------|
| gmp-6.1.2        | 16 Dec 2016 at 15:45 | --      | Folder            |
| graphs           | Yesterday at 18:31   | --      | Folder            |
| models           | Yesterday at 18:33   | --      | Folder            |
| settings         | 18 Feb 2019 at 18:31 | --      | Folder            |
| solutions        | Yesterday at 18:33   | --      | Folder            |
| soplex-4.0.1     | Today at 11:34       | --      | Folder            |
| soplex-4.0.1.tgz | Today at 11:31       | 959 KB  | gzip co...archive |
| Assignment.py    | Today at 09:28       | 16 KB   | Python Source     |
| gmp-6.1.2.tar    | Today at 11:23       | 16.3 MB | tar archive       |

Figure 1: Directory Structure

where  $\mathbf{V}$  is the number of vertices,  $\mathbf{E}$  is the number of edges, and each  $[\mathbf{vi1}] [\mathbf{vi2}]$  pair (for  $i \in \{1, 2, \dots, \mathbf{E}\}$ ) represents the edge between vertex  $[\mathbf{vi1}]$  and  $[\mathbf{vi2}]$ . For example, the complete four-vertex graph would be represented as follows:

```

4 6
0 1
0 2
0 3
1 2
1 3
2 3

```

**Note:** Vertex numbering *must* start at 0, not 1, for the program to work.

- **/models:** This folder contains the *.mps* linear program modelling files generated by my program, which are then solved by the SoPlex solver.
- **/settings:** This folder contains the settings files for the SoPlex solver.
- **/solutions:** This folder contains the detailed solutions outputted by the SoPlex solver. Each time the program is executed, it will output two files - one for the Fractional Clique Cover Number, and one for the Shannon Entropy.

## 2.2 Running

To run the program, please run the following command (from the base directory):

```
python Assignment.py graphs/[graphName.txt]
```

The program will output the Fractional Clique Cover Number, the Shannon Entropy, and the solutions to both of the linear programs concisely, and in rational format. As mentioned above, more detailed output for both linear programs is saved in the /solutions folder.

## 3 Linear Program Reformulation

### 3.1 Fractional Clique Cover Number

- In the specification, variables are introduced for all the subsets  $S$  of  $V$ , the set of all vertices. However, as the only non-zero variables are those corresponding to the cliques of  $V$ , I introduced variables only for the cliques. To find the cliques of  $V$  efficiently, I modified the well-known *Bron Kerbosch* algorithm, so that it recursively finds *all* the cliques of  $V$ , not only the maximal ones.
- Whilst searching for how to further simplify the linear program, I found an article by Peter Cameron[1] which states that "It can be shown that the same minimum value [the Fractional Clique Cover Number] is obtained if we restrict to regular fractional clique covers.". A regular fractional clique cover is one

where, for every vertex, the sum of the values given to the cliques that contain this vertex **equals** one, rather than being bounded above by 1. The proof of this claim can be found in the paper referenced in the article[2], and so I implemented this simplification in my program.

### 3.2 Shannon Entropy

For the Shannon Entropy linear program, variables are required for all of the subsets of the set of vertices  $V$ . However, many of the constraints 'overlap' and so are redundant.

- The first constraint that I simplified was the:

$$x_T - x_S \geq 0 \quad \forall S \subseteq T \subseteq V$$

constraint. For every subset  $T$  of  $V$ , we do not need to create a constraint for every subset  $S$  of  $T$ ; it is sufficient to only create constraints for subsets  $S$  that are of size one less than  $T$ . This is because if  $R \subseteq S \subseteq T$ , then  $x_R \leq x_S$  and  $x_S \leq x_T \implies x_R \leq x_T$ . This is more easily shown diagrammatically:

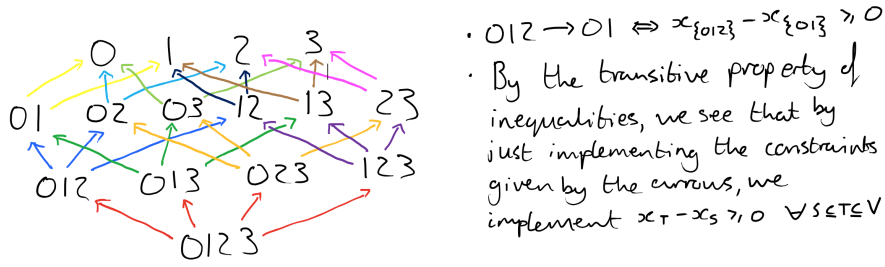


Figure 2: Constraint Reformulation

For a graph of  $N$  vertices, the total number of constraints without the reformulation would be  $\sum_{n=2}^N \binom{N}{n} (2^n - 1)$  (where constraints involving the empty set have been removed as the variable corresponding to the empty set is 0), whereas the the number of constraints with this reformulation is  $\sum_{n=2}^N \binom{N}{n} n$ . For the  $N = 8$  case, this is a reduction of  $6297 - 1016 = 5281$  constraints.

- To increase the efficiency of the constraint generation, I calculated the above constraint and the  $x_{N(v) \cup \{v\}} - x_{N(v)} = 0$  constraint in one traversal of the subsets pairs, as the neighbourhood constraint also involves two sets differing in size by only one variable. I don't think it is possible to further simplify the neighbourhood constraint, and as there is only one constraint per vertex, the number of these constraints is negligible with respect to the total number of constraints.
- The next constraint that I simplified significantly was the:

$$x_S + x_T - x_{S \cup T} - x_{S \cap T} \geq 0 \quad \forall S, T \subseteq V$$

constraint. Instead of creating a constraint for every pair of vertices  $S$  and  $T$ , of which there would be  $2^N \times 2^N = 2^{2N}$ , constraints are only required for pairs of vertices where  $S \not\subseteq T$  and  $T \not\subseteq S$ . If  $S \subseteq T$ , then:

$$x_S + x_T - x_{S \cup T} - x_{S \cap T} = x_S + x_T - x_T - x_S = 0$$

and so the constraint becomes trivial. The same argument holds for  $T \subseteq S$  by simply switching the variable labels.

- The final constraint I attempted to simplify was the:

$$x_{\{v\}} \leq 1 \quad \forall v \in V$$

constraint. After lots of experimentation, I believe that the inequality in this constraint can be replaced with an equality. This intuition is backed up by the proof that this modification holds in the Fractional Clique Cover Number linear program. Furthermore, when testing my program with the constraint as an inequality, SoPlex reported that there were a small number of redundant constraints, which I believe to relate to this constraint. However, as I could not come up with a formal proof to back up this intuition, I decided to leave the constraint as an inequality in my final implementation.

|  |  |   |
|--|--|---|
| <pre> ----- Fractional Clique Cover Number ----- Solutions: x01: (0, 1, 2, 3, 4, 5, 6, 7) = 1 (All other variables are 0) Objective Value: 1 ----- Shannon Entropy ----- Solutions: x21: (0, 1) = 1 x31: (1, 1) = 1 x41: (1, 2) = 1 x51: (1, 3) = 1 x61: (1, 4) = 1 x71: (1, 5) = 1 x81: (1, 6) = 1 x91: (1, 7) = 1 x101: (0, 1, 2) = 2 x111: (0, 2, 3) = 2 x121: (0, 3, 4) = 2 x131: (0, 4, 5) = 2 x141: (0, 5, 6) = 2 x151: (0, 6, 7) = 2 x161: (1, 2, 3) = 2 x171: (1, 3, 4) = 2 x181: (1, 4, 5) = 2 x191: (1, 5, 6) = 2 x201: (1, 6, 7) = 2 x211: (0, 1) = 2 x221: (0, 2) = 2 x231: (0, 3) = 2 x241: (0, 4) = 2 x251: (0, 5) = 2 x261: (0, 6) = 2 x271: (0, 7) = 2 x281: (1, 2, 3) = 2 x291: (1, 3, 4) = 2 x301: (1, 4, 5) = 2 x311: (1, 5, 6) = 2 x321: (1, 6, 7) = 2 x331: (0, 1, 2) = 2 x341: (0, 2, 3) = 2 x351: (0, 3, 4) = 2 x361: (0, 4, 5) = 2 x371: (0, 5, 6) = 2 x381: (0, 6, 7) = 2 x391: (1, 2, 3) = 2 x401: (1, 3, 4) = 2 x411: (1, 4, 5) = 2 x421: (1, 5, 6) = 2 x431: (1, 6, 7) = 2 x441: (0, 1, 2, 3) = 4 x451: (0, 2, 3, 4) = 4 x461: (0, 3, 4, 5) = 4 x471: (0, 4, 5, 6) = 4 x481: (0, 5, 6, 7) = 4 x491: (0, 1, 2, 3, 4) = 5 x501: (0, 2, 3, 4, 5) = 5 x511: (0, 3, 4, 5, 6) = 5 x521: (0, 4, 5, 6, 7) = 5 x531: (0, 1, 2, 3, 4, 5) = 5 x541: (0, 2, 3, 4, 5, 6) = 5 x551: (0, 3, 4, 5, 6, 7) = 5 x561: (0, 4, 5, 6, 7) = 5 x571: (0, 1, 2, 3, 4, 5, 6) = 6 x581: (0, 2, 3, 4, 5, 6, 7) = 6 x591: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x601: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x611: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x621: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x631: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x641: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x651: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x661: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x671: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x681: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x691: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x701: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x711: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x721: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x731: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x741: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x751: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x761: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x771: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x781: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x791: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x801: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x811: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x821: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x831: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x841: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x851: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x861: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x871: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x881: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x891: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x901: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x911: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x921: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x931: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x941: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x951: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x961: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x971: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x981: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x991: (0, 1, 2, 3, 4, 5, 6, 7) = 7 x1001: (0, 1, 2, 3, 4, 5, 6, 7) = 7 </pre> | <pre> ----- Fractional Clique Cover Number ----- Solutions: x101: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x111: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x121: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x131: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x141: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x151: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x161: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x171: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x181: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x191: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x201: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x211: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x221: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x231: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x241: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x251: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x261: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x271: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x281: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x291: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x301: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x311: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x321: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x331: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x341: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x351: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x361: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x371: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x381: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x391: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x401: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x411: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x421: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x431: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x441: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x451: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x461: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x471: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x481: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x491: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x501: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x511: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x521: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x531: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x541: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x551: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x561: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x571: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x581: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x591: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x601: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x611: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x621: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x631: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x641: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x651: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x661: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x671: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x681: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x691: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x701: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x711: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x721: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x731: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x741: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x751: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x761: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x771: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x781: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x791: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x801: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x811: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x821: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x831: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x841: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x851: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x861: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x871: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x881: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x891: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x901: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x911: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x921: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x931: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x941: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x951: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x961: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x971: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x981: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x991: (0, 1, 2, 3, 4, 5, 6, 7) = 5 x1001: (0, 1, 2, 3, 4, 5, 6, 7) = 5 </pre> | <pre> ----- Fractional Clique Cover Number ----- Solutions: x31: (0, 1, 5) = 8/3 x32: (0, 1, 6) = 8/3 x33: (0, 2, 3) = 8/3 x34: (0, 2, 4) = 8/3 x35: (0, 2, 5) = 8/3 x36: (0, 2, 6) = 8/3 x37: (0, 2, 7) = 8/3 x38: (0, 3, 4) = 8/3 x39: (0, 3, 5) = 8/3 x40: (0, 3, 6) = 8/3 x41: (0, 3, 7) = 8/3 x42: (0, 4, 5) = 8/3 x43: (0, 4, 6) = 8/3 x44: (0, 4, 7) = 8/3 x45: (0, 5, 6) = 8/3 x46: (0, 5, 7) = 8/3 x47: (0, 6, 7) = 8/3 x48: (1, 2, 3) = 8/3 x49: (1, 2, 4) = 8/3 x50: (1, 2, 5) = 8/3 x51: (1, 2, 6) = 8/3 x52: (1, 2, 7) = 8/3 x53: (1, 3, 4) = 8/3 x54: (1, 3, 5) = 8/3 x55: (1, 3, 6) = 8/3 x56: (1, 3, 7) = 8/3 x57: (1, 4, 5) = 8/3 x58: (1, 4, 6) = 8/3 x59: (1, 4, 7) = 8/3 x60: (1, 5, 6) = 8/3 x61: (1, 5, 7) = 8/3 x62: (1, 6, 7) = 8/3 x63: (2, 3, 4) = 8/3 x64: (2, 3, 5) = 8/3 x65: (2, 3, 6) = 8/3 x66: (2, 3, 7) = 8/3 x67: (2, 4, 5) = 8/3 x68: (2, 4, 6) = 8/3 x69: (2, 4, 7) = 8/3 x70: (2, 5, 6) = 8/3 x71: (2, 5, 7) = 8/3 x72: (2, 6, 7) = 8/3 x73: (3, 4, 5) = 8/3 x74: (3, 4, 6) = 8/3 x75: (3, 4, 7) = 8/3 x76: (3, 5, 6) = 8/3 x77: (3, 5, 7) = 8/3 x78: (3, 6, 7) = 8/3 x79: (4, 5, 6) = 8/3 x80: (4, 5, 7) = 8/3 x81: (4, 6, 7) = 8/3 x82: (5, 6, 7) = 8/3 x83: (0, 1, 2, 3, 4, 5, 6, 7) = 10/3 x84: (0, 1, 2, 3, 4, 5, 6, 7) = 10/3 x85: (0, 1, 2, 3, 4, 5, 6, 7) = 10/3 x86: (0, 1, 2, 3, 4, 5, 6, 7) = 10/3 x87: (0, 1, 2, 3, 4, 5, 6, 7) = 10/3 x88: (0, 1, 2, 3, 4, 5, 6, 7) = 10/3 x89: (0, 1, 2, 3, 4, 5, 6, 7) = 10/3 x90: (0, 1, 2, 3, 4, 5, 6, 7) = 10/3 x91: (0, 1, 2, 3, 4, 5, 6, 7) = 10/3 x92: (0, 1, 2, 3, 4, 5, 6, 7) = 10/3 x93: (0, 1, 2, 3, 4, 5, 6, 7) = 10/3 x94: (0, 1, 2, 3, 4, 5, 6, 7) = 10/3 x95: (0, 1, 2, 3, 4, 5, 6, 7) = 10/3 x96: (0, 1, 2, 3, 4, 5, 6, 7) = 10/3 x97: (0, 1, 2, 3, 4, 5, 6, 7) = 10/3 x98: (0, 1, 2, 3, 4, 5, 6, 7) = 10/3 x99: (0, 1, 2, 3, 4, 5, 6, 7) = 10/3 x100: (0, 1, 2, 3, 4, 5, 6, 7) = 10/3 </pre> |
|--|--|---|

(a) Complete Eight Vertex Graph Solution

(b) Graph 1[3] Solution

Figure 3: Example Solutions

# 4 Examples

In Figure 3, the outputs for the complete eight vertex graph and Graph 1 are shown. As expected, the Fractional Clique Cover Number of the complete graph is 1, the Shannon Entropy of the complete graph is  $8 - 1 = 7$ , the Fractional Clique Cover Number of Graph 1 is  $10/3$ , and finally the Shannon Entropy of Graph 1 is  $11/3$ . Please note that multiple other example graphs are included in the /graphs folder, including all 6 graphs from the paper[3].

# 5 Additional Information

- If you would like the program to print out the linear program (the objective function and the constraints) before it is solved, please uncomment out line 435 of *Assignment.py*.
- I inverted all of the greater than constraints so that the model only consists of equality and less than constraints, in order to simplify the implementation.
- The vertex numbering in the input files must begin at vertex 0, and continue sequentially.
- In my initial implementation, I used the linear program solver included in the SciPy package, and then used the Fraction module to convert the floating point output to a rational format. However, this seemed unsatisfactory and susceptible to floating-point arithmetic nuances, and so I reimplemented my code using Pyomo and SoPlex. My program first generates all the variables and constraints, which are then passed to Pyomo. Pyomo builds the LP model and outputs it in the standard .mps format, and then my program adds a new process and executes the SoPlex program, which solves the model. My program then reads the exact solution outputted by SoPlex, parses the rational numbers as strings, creates exact fractions from these strings using the Fraction module, and then finds the exact optimal value using the exact optimal solution and the objective function.

# References

- [1] Peter Cameron: *Guessing Numbers of Graphs*, <https://cameroncounts.wordpress.com/2016/03/20/guessing-numbers-of-graphs/>
- [2] Peter Cameron, Anh N. Dang, Søren Riis: *Guessing Games on Triangle-free Graphs*, The Electronic Journal of Combinatorics
- [3] Maximilien Gadouleau: *On the possible values of the entropy of undirected graphs*, Journal of Graph Theory