

Using the program

Dependencies

- networkx
- pulp
- fractions

Each of the above packages are used in my implementation. ‘networkx’ is used for graph representation, ‘pulp’ is used as our linear program solver and ‘fractions’ is used to ensure each output is rational.

In order to install, we simply do the following:

- pip install networkx
- pip install pulp
- fractions is included in the python standard library

Alternatively, you could manually download them from github or elsewhere.

Input

For a given graph G with n vertices, we require a simple method of inputting the graph into our tool. We represent the graph in a text file. The text file is represented in the following format:

vertex/line 1: $(1, 2) (1, 3) \dots (1, n-2) (1, n-1) (1, n)$

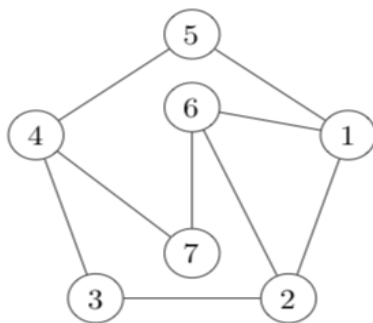
vertex/line 2: $(2, 3) (2, 4) \dots (2, n-1) (2, n)$

vertex/line i: $(i, i+1) (i, i+2) \dots (i, n)$

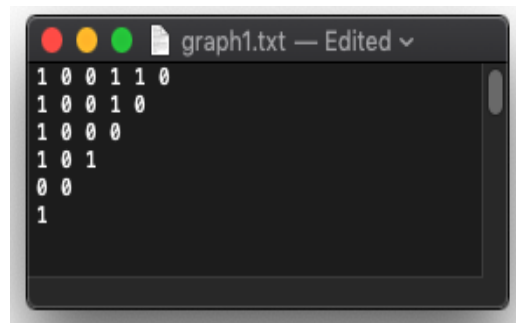
vertex/line n-2: $(n-2, n-1) (n-2, n)$

vertex/line n-1: $(n-1, n)$

For each (i, j) we either have $(i, j) = 0$ if there does not exist an edge between vertex i and vertex j , and $(i, j) = 1$ if there is an edge between vertex i and vertex j . Below, in Figure 1 we see an example graph and its corresponding representation in a text file. Note we don’t label lines ‘**vertex/line i**’ in the actual text file, this is simply in the above for clarity.



The graph G_1 .



The input representation of G_1 .

Figure 1: A given graph G_1 and its corresponding input representation in a text file.

```

Harrys-MBP:code harryrichards$ python optimisation.py
input graph file: 'graph1.txt'

fractional clique cover number value: 10/3
shannon entropy value: 11/3

fractional clique cover number solution saved to graph1_xs_solution.txt
shannon entropy solution saved to graph1_xv_solution.txt

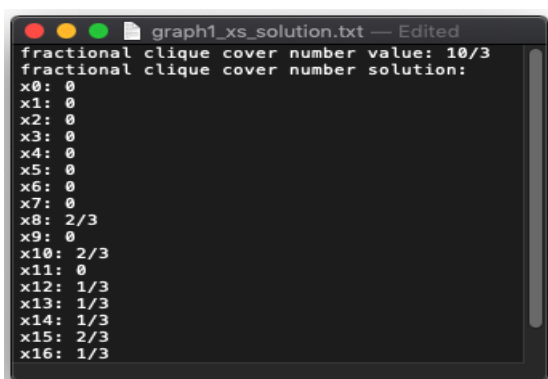
```

Figure 2: The input procedure for our program, in terminal.

- In order to run the program, we begin by opening the command line (in this case terminal) and navigating to the directory of our Python file.
- Once we're in the same directory as our python file, input 'python optimisation.py' to the run the file.
- At this point we're prompted to provide the input graph file that we'll be running the program on. Since our the input file for our graph is called 'graph1.txt' and it's in the same directory as our program, I can simply input 'graph1.txt'. If it was in another directory I would need to provide that directory.
- An example of me running the program on 'graph1.txt' is shown in Figure 2.

Output

As we saw in Figure 2 once we run the program and provide it with an input graph, it outputs the fractional cover number value and shannon entropy value to the user - in **rational format**. However, since the solution may contain a large number of variables, rather than output each variable value we save the solution to a file. The aforementioned solution files are simply text files that contain the value of the objective function which is either the **fractional clique cover number** or the **shannon entropy**, followed by each variable in the linear program and the variable's value. The two text files produced for our earlier graph G_1 (displayed in Figure 1) are displayed in Figure 3. The text files are produced and saved in the same directory as our python file. Both the objective function values and each variable value are stored in rational format making the text file easily readable.

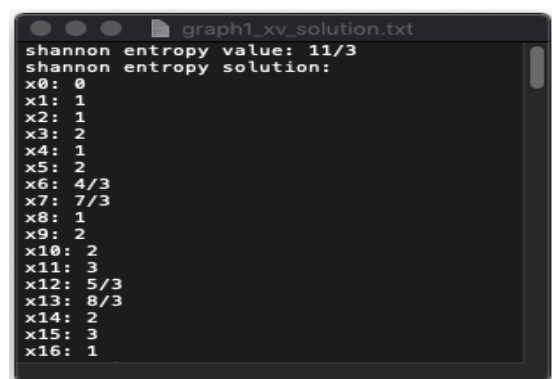


```

graph1_xs_solution.txt — Edited
fractional clique cover number value: 10/3
fractional clique cover number solution:
x0: 0
x1: 0
x2: 0
x3: 0
x4: 0
x5: 0
x6: 0
x7: 0
x8: 2/3
x9: 0
x10: 2/3
x11: 0
x12: 1/3
x13: 1/3
x14: 1/3
x15: 2/3
x16: 1/3

```

The saved solution for the fractional clique cover number for graph G_1 and each x_S value.



```

graph1_xv_solution.txt
shannon entropy value: 11/3
shannon entropy solution:
x0: 0
x1: 1
x2: 1
x3: 2
x4: 1
x5: 2
x6: 4/3
x7: 7/3
x8: 1
x9: 2
x10: 2
x11: 3
x12: 5/3
x13: 8/3
x14: 2
x15: 3
x16: 1

```

The saved solution for the shannon entropy for graph G_1 and each x_S value.

Figure 3: The above is the standard format of any of the solution files produced by the program, in any case each of the variable values or objective function values will be given in rational format.

Linear Program modifications

Here we use the same definitions for G , V , S and $\mathcal{K}(G)$ as stated in the assignment brief.

fractional clique cover number $\pi^*(G)$

$$\begin{aligned} & \text{minimise} && \sum_{S \subseteq V} x_S \\ & \text{subject to} && \sum_{S: v \in S} x_S \geq 1, \quad \forall v \in V \\ & && x_S = 0 \quad \forall S \notin \mathcal{K}(G) \\ & && x_S \geq 0 \quad \forall S \subseteq V \end{aligned}$$

Since the objective of the above is to minimise a sum, and we have the constraint

$$x_S = 0 \quad \forall S \notin \mathcal{K}(G)$$

we need only consider the values x_S such that $S \in \mathcal{K}(G)$, as any x_S that are not in the set of cliques will have a value of 0 and thus they will not contribute to the sum. As a result of this, the **fractional clique cover number** linear program becomes:

$$\begin{aligned} & \text{minimise} && \sum_{S \in \mathcal{K}(G)} x_S \\ & \text{subject to} && \sum_{S: v \in S \in \mathcal{K}(G)} x_S \geq 1, \quad \forall v \in V \\ & && x_S \geq 0 \quad \forall S \subseteq V \end{aligned}$$

and due to the reduction in number of variables and removal of redundant constraints, we are able to obtain a solution to the linear program far more quickly.

shannon entropy $\eta(G)$

$$\begin{aligned} & \text{maximise} && x_V \\ & \text{subject to} && x_\emptyset = 0 \\ & && x_{\{v\}} \leq 1 && \forall v \in V \\ & && x_{N(v) \cup \{v\}} - x_{N(v)} = 0 && \forall v \in V \\ & && x_T - x_S \geq 0 && \forall S \subseteq T \subseteq V \\ & && x_S + x_T - x_{S \cup T} - x_{S \cap T} \geq 0 && \forall S, T \subseteq V \\ & && x_S \geq 0 && \forall S \subseteq V \end{aligned}$$

If we consider the case when $S \subseteq T$, then we have that $S \cup T = T$ and $S \cap T = S$. As a result $x_S + x_T - x_{S \cup T} - x_{S \cap T} = x_S + x_T - x_T - x_S = 0$, and applying the constraint

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

doesn't give us any new information as it essentially states $0 \geq 0$. Thus in the case that $S \subseteq T$ we need not apply $x_S + x_T - x_{S \cup T} - x_{S \cap T} \geq 0 \quad \forall S, T \subseteq V$ - as the constraint is redundant.

Similarly, when $S = T$, the constraint $x_T - x_S \geq 0 \quad \forall S \subseteq T \subseteq V$ simply gives us $x_T - x_S = x_T - x_T = 0 \geq 0$, again this constraint is redundant. We avoid this by altering the constraint to

$x_T - x_S \geq 0 \quad \forall S \subset T \subseteq V$ so that it's only applied when S is a proper subset of T .

As a result of both of the above alterations, the **shannon entropy** linear program becomes:

$$\begin{aligned}
 & \text{maximise} && x_V \\
 & \text{subject to} && x_\emptyset = 0 \\
 & && x_{\{v\}} \leq 1 && \forall v \in V \\
 & && x_{N(v) \cup \{v\}} - x_{N(v)} = 0 && \forall v \in V \\
 & && x_T - x_S \geq 0 && \forall S \subset T \subseteq V \\
 & && x_S + x_T - x_{S \cup T} - x_{S \cap T} \geq 0 && \forall S \not\subseteq T \subseteq V \\
 & && x_S \geq 0 && \forall S \subseteq V
 \end{aligned}$$

The above alterations to the two linear programs result in a reduction in the number of constraints for any graph, as it prevents us from producing (and applying) a number of redundant constraints. As a result of this, solutions to the two linear programs may be obtained far more quickly.

Solutions

In this section I will discuss some of the solutions obtained, and the speed at which they were attained. Due to the large number of variables involved in the solution, I will simply state the value of the objective function. If you wish to investigate the specific variable values - find them in solution text files and graph representations in the submission folder. The random graph is shown in Figure 4.

number of vertices	10	8	8	8	8	7	7
graph type	complete	complete	random	star	cycle	G_1	G_2
fractional cover clique number	1	1	3	7	4	$\frac{10}{3}$	$\frac{7}{2}$
shannon entropy	9	7	5	1	4	$\frac{11}{3}$	$\frac{7}{2}$
time taken (seconds)	100.124	3.258	4.202	3.423	3.932	0.784	0.693

In the table we see a number of shannon entropy and fractional cover clique numbers for various graphs - calculated using our program. The graphs G_1 and G_2 are the same G_1 and G_2 found in [1]. In each case we see that the $\eta(G) + \pi^*(G) \geq n$ is satisfied. The 'time taken' shown in the table is the combined time to obtain and save a solution for both linear programs. However, obtaining the fractional cover clique number is almost instantaneous and the majority of the time is spent obtaining the shannon entropy. In comparison to a naive implementation (which implements every constraint in the original linear program), the altered linear programs are consistently faster. When both the altered and naive linear program were performed on a graph with 8 vertices: the altered shannon entropy linear program was approximately 114.6% faster (i.e. it took less than half the time), whilst the altered fractional clique cover number linear program was approximately 205.26% faster (i.e. it took less than a third of the time).

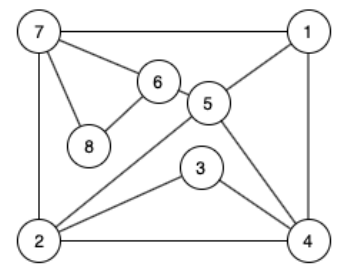


Figure 4: The random graph with 8 vertices and $p = \frac{1}{2}$.

References

- [1] Maximilien Gadouleau. On the possible values of the entropy of undirected graphs. *Journal of Graph Theory*, 82:302–311, 2018.