Questions

a. For a given random graph with n nodes and edges, what proportion of the minimum vertex cover is expected from mvc_1, mvc_2, and mvc_3? You may answer the question using a Table.
Part of my second experiment creates 100 random undirected graphs with 10 nodes and 10 edges. mvc_1 has perfect accuracy for these graphs, returning the exact same number of total nodes across all the graphs as the MVC function.

mvc _2 has a proportion of 1.25 of the MVC function, meaning mvc_2 typically includes 25% more nodes than the minimum number needed to span all edges

Finally, mvc_3 has a very similar proportion as mvc_2 at 1.2663. Which means that mvc_3 includes 26.6% more nodes than the minimum number needed to span all edges.

b. Is there a relationship between how good we would expect an approximation to be and the number of edges in a graph? In general, does the approximation get better/worse as the number of edges increases/decreases?

Yes there is a relationship between how good approximations are and the number of edges ina graph, generally as the number of edges increases, every vertex cover function becomes more accurate, except mvc_1 which is already practically perfect. For instance with mvc_2 has a proportion of 1.26 for 5 edges, and a proportion of 1.108 for 20 edges, and mvc_3 has a proportion of 1.4747… for 1 edge and a proportion of 1.128 for 20 edges.

c. Is there a relationship between how good we would expect an approximation to be and the number of nodes in a graph? In general, does the approximation get better/worse as the number of nodes increases/decreases? To answer this question, you may have to run an experiment where you vary the number of nodes instead of edges. Include the code of this experiment in your submission.

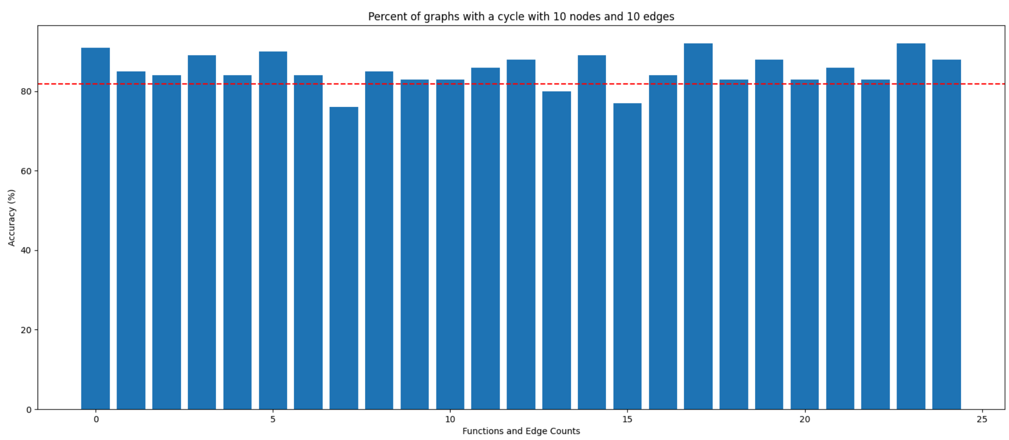
Yes, there is a relationship between number of nodes in a graph and how good the approximations are. In general the approximations get worse as the number of nodes increases. In my experiment 3, for mvc_2 5 nodes and 15 edges results in a proportion of 1.139, while 20 nodes and 15 edges results in a proportion of MVC of 1.4. A similar trend exists for mvc_3 but the gap between 5 nodes and 20 nodes is even more extreme.

d. Remember to include details of Part 5.
Part 5–or experiment 1–involved me creating 100 graphs, each with 10 nodes and 10 edges. I ran this experiment 25 times to ensure the accuracy of the test.

On each graph I ran the function has_cycle to determine if there existed a pattern on the graph somewhere, if this has_cycle returned true then

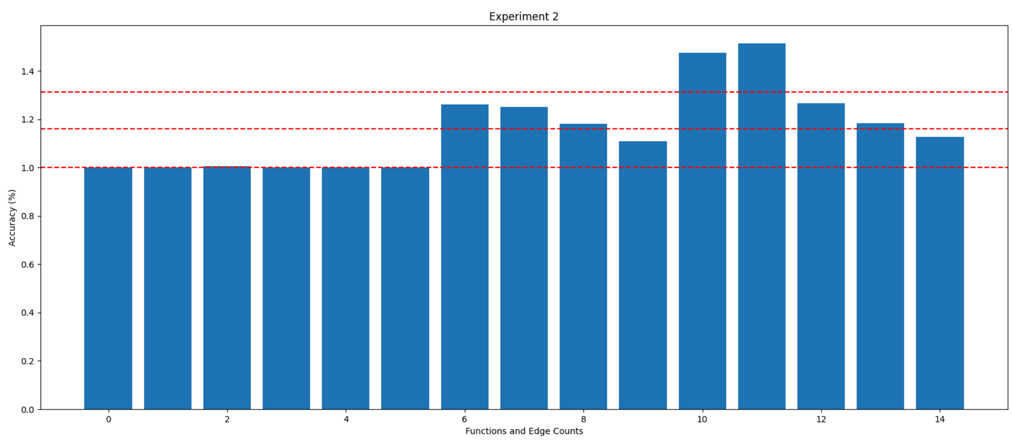
Appendix

Graph 1 (Experiment 1/Part 5):



Each line represents 1 iteration of the experiment, and as mentioned in question d there were 25 experiments run. The y-axis tracks the percent of graphs with a cycle, and the x-axis tracks the iteration number of the experiment. The total percent of graphs of all experiments with a cycle comes out to ~81.8% of graphs.

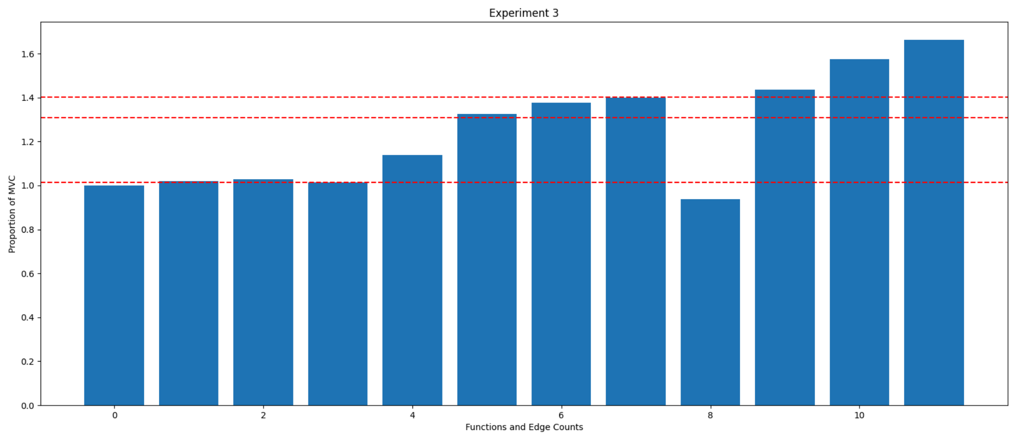
Graph 2 (Experiment 2/Part 7):



Above is my graph for experiment 2. Each of the bars represents an experiment. The first 5 are mvc run with mvc_1, the next 5 are from mvc_2, and finally the last 5 are the results of mc_3. The experiments are ordered in ascending order of lowest to highest number of edges. So the first bar is mvc_1 with 1 edge, then mvc_1 with 5 edges, and so on. The Y-axis is also the proportion, so what proportion of the Minimum vertex cover each experiment produces

The dotted red lines detail the proportion of the minimum vertex cover for each mvc function by averaging all their experiments together. The lowest is mvc_1 at essentially 1.0, the mean line is for mvc_2 at approximately 1.16, and the mean line of mvc_3 is approximately 1.313.

Graph 3 (Experiment 3/Part 7 continued):



Above is my graph for experiment 3. Each of the graphs represents an individual experiment like for experiment 2. Again the first 4 are mvc_1, the next 4 are mvc_2, and the last 4 are mvc_3. This time instead of varying edges, we're varying nodes, the first being 5 nodes, then 10 nodes, then 15, and 20. The y-axis is still the proportion of minimum vertex cover of each experiment.

The dotted red lines again detail the proportion of minimum vertex cover for each mvc function. The first mean line is mvc_1 at 1.015, the second mean line is mvc_2 at 1.3094, and finally the third mean line is mvc_3 at 1.402.