ECE 650 Project
Analysis of Vertex Cover Algorithms

Meenaxi Chandra Prasad(20501898),Zeba Javedahamadkhan Vakil(20765060)
Department of Electrical and Computer Engineering,
University of Waterloo

1. Introduction

In graph theory, a vertex cover of a graph is a set of vertices such that each edge of the graph is incident to at least one vertex of the set. The problem of finding a minimum vertex cover is a classical optimization problem in computer science and is a typical example of an NP-hard optimization problem that has approximation algorithm (Wikipedia). SAT solver gives the minimum vertex cover set and the approximation algorithms gives the vertex cover set which is not minimum always. For SAT solver, it checks all possible combinations of vertices to find minimum vertex cover, which is not applicable for approximation algorithms. To analyze the three algorithms, the input samples are taken as $|V| \in [5,10,15,20]$. It is designed for running time and approximate ratio in CNF-SAT, Approx-1 and Approx-2.

Example:

Graph (V,E) as input:

V 5

E {<2,1>,<2,0>,<2,3>,<1,4>,<4,3>}

The output of the program:

CNF-SAT-VC: 2,4

APPROX-VC-1: 2,4

APPROX-VC-2: 0,2,3,4

2. Analysis

Algorithms are compared for two factors: Processing time and size of the vertex cover output. For this test setup, Minisat is copied from https://github.com/agurfinkel/minisat. Main executable file prj-ece650.cpp alongwith minisat and Cmake file is run on the Linux environment. For mathematical calculations and graph plotting Microsoft excel is used.

2.1 Processing time:

To measure the processing time, for each value of V, 10 input sets are executed and processing time is measured for 10 runs for each input set. The running time is calculated by pthread_getcpuclockid() function. Measuring time is copied in an excel file and average of all values of processing time and standard deviation are calculated for all V input sets (5, 10, 15, 20). These values are plotted in the graph. Processing time for Approx_VC1 and Approx_VC2 are represented in millisecond and time for CNF SAT solver is represented in seconds. As processing time of approximation algorithms is very less compared to CNF SAT solver processing time. Different graphs are plotted for approximation algorithms and CNF SAT solver processing time.

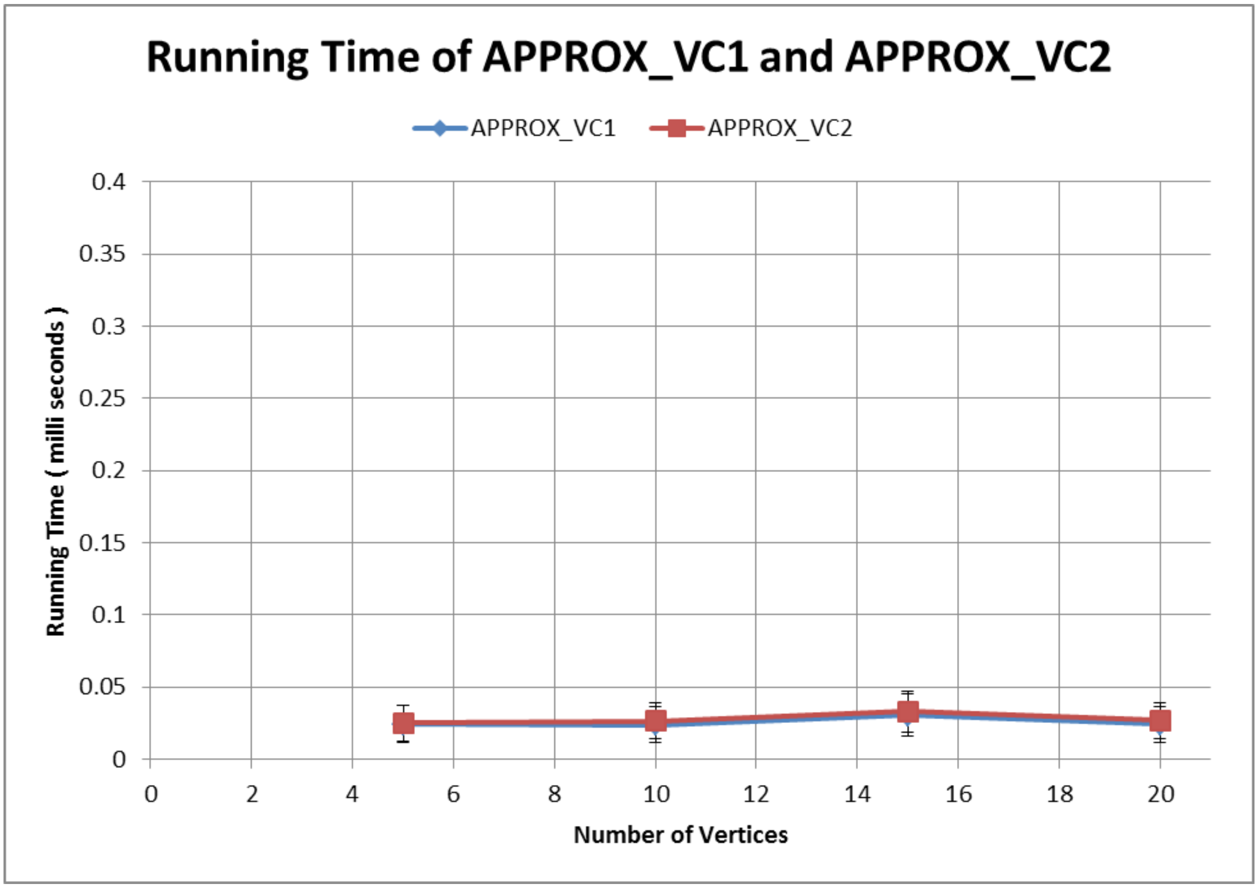

Figure 1: Running time of Approximation VC1 algorithm and VC2 algorithm (time in micro second was very low of these algorithms that’s why converted to millisecond for representation on the graph. There is very less difference in the processing time of both algorithms: Processing time for VC algorithm 1: 0.000001367 and for VC algorithm 2: 0.000001424. This difference is so less that it’s not clearly distinguishable on the plot. Still we can see two different colors on the plot for both algorithms).

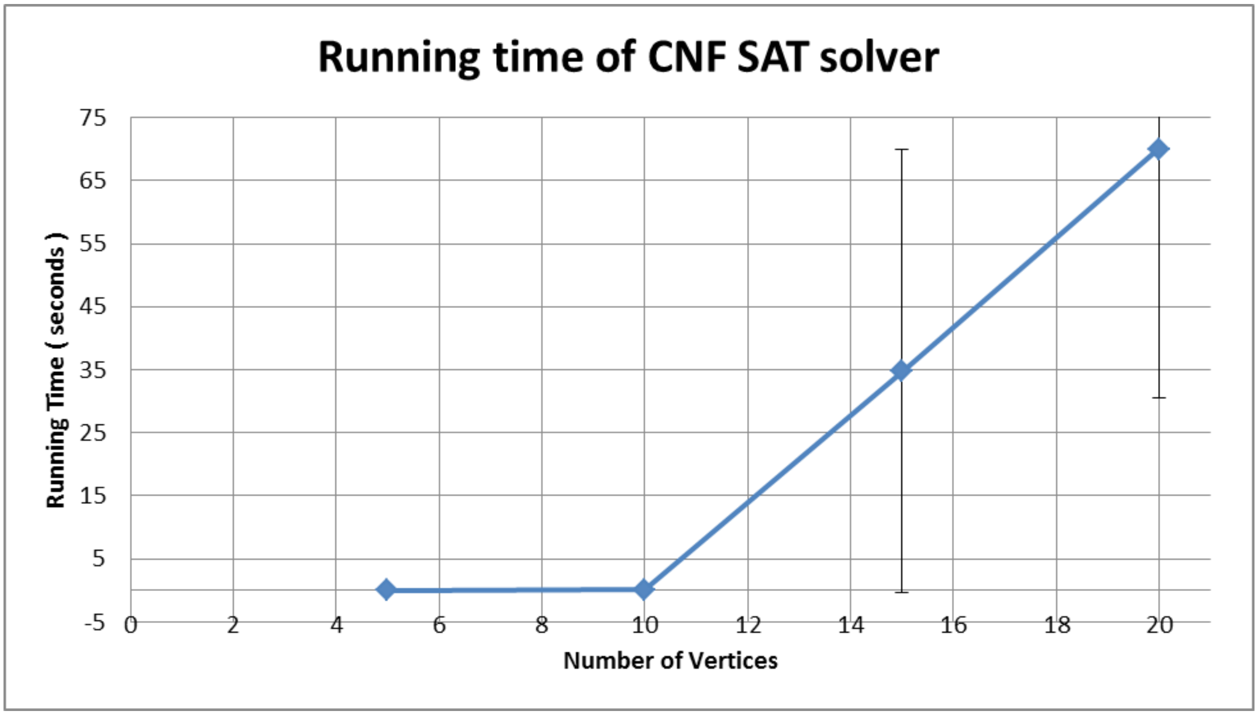

Figure 2: Running time of CNF SAT solver. (Separate graph is plotted due to big difference in the processing time of CNF SAT solver and other algorithms.)

As from figure 2, we can see that CNF SAT solver at solver performs very fast for less number of vertices(5..10). When number of vertices increase, it takes more time to find vertex cover. We can see an exponential growth between vertex 14 and 15. It seems that this growth will continue when we run SAT solver for graphs with larger number of vertices. This is because as number of vertices(v) increases and it will increase the number of clauses k * v * (v – 1)/2 where k is the size of vertex cover. And also with increased edge numbers it has to satisfy more clauses. In case of small number of vertices it takes less number of clauses. So it takes less time as it has to satisfy less number of combinations of variables. SAT Solver is shwoing exponential growth.

2.2 Approximation ratio:

For the approximate ratio is calculated by comparing the output of Approx_VC1 and Approx_VC2 with the output of SAT solver. (ApproximationRatio = size of vertex cover by approximation alogrithm/ size of vertex cover by CNF SAT solver).
SAT solver is used for comparison as SAT solver gives the optimal solution. Ratio of the output of approximation algorithms and SAT solver output is taken.

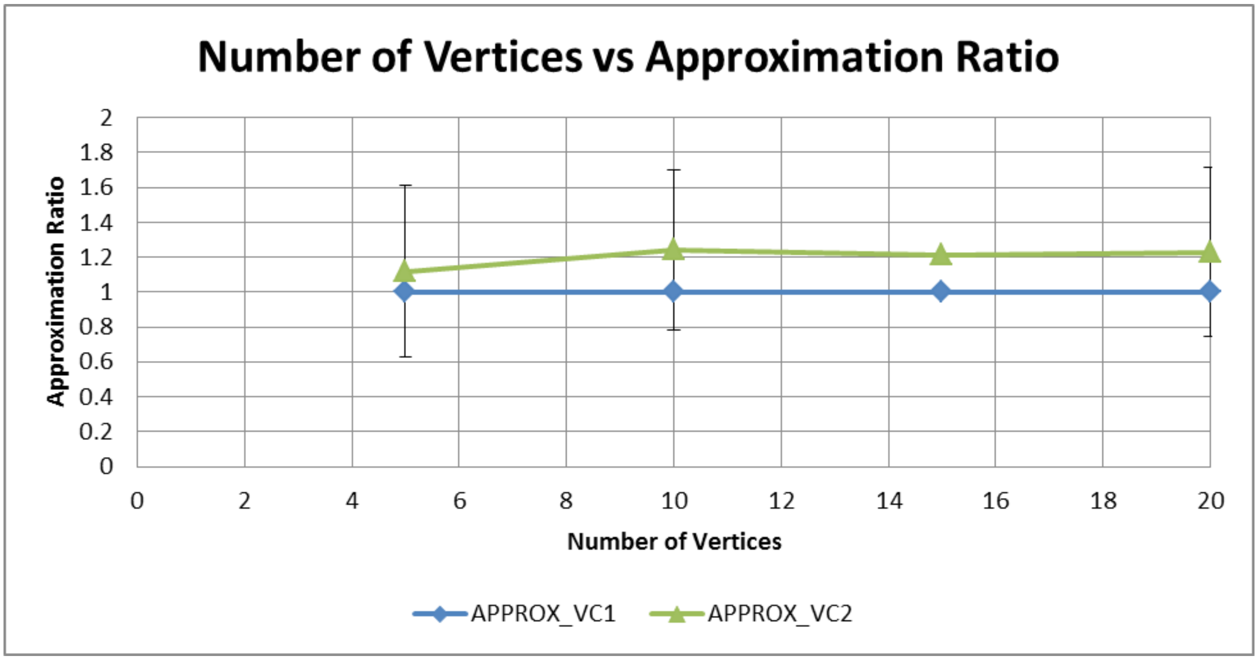

Figure 3: Approximation ratio for the two algorithms compared to CNF SAT solver.

As we can see in the figure 3, Algorithm VC1 gives same output as SAT solver that's why the approximation ratio is 1 and for VC2 algorithm approximation ratio is slightly greater than 1. Most of the approximation ratio for Approx-1 is very close to 1, which means that vertex cover of Approx-1 is nearly optimal (minimum-sized) vertex cover. For Approx-1, it picks vertex of highest degree and excludes more edges than Approx-2 in first step. For Approx-2, it picks an edge and at least 2 vertexes in first step, while Approx-1 only picks

1 vertex. So Approx-1 collects smaller vertexes to cover whole edges and closer to optimal (minimum-sized) vertex cover than Approx-2.

3. Conclusion

According to the run time and approximate ratio, three algorithms analyzed have good performance depending on the graph inputs. So they can be effective to solve specified vertex cover problem. It is hard to say which algorithm is best as it depends on the need, like number of inputs, how optimal is the output needed and processing time. In terms of run time, Approx-1 and Approx-2 are better. Approx-2 is better way to get vertex cover but not the minimum vertex cover.

In terms of approximation ratio, CNF-SAT definitely get minimum vertex cover, but it need more time than Approx-1 and Approx-2. So it is a better way to solve small scale vertex cover problem. In general, Approx-1 is a better way to solve large scale vertex cover problem, the running time is less than that of CNF-SAT, and it can sometime get minimum vertex cover result.