ECE650 - METHODS AND TOOLS OF SOFTWARE ENGINEERING

UNIVERSITY OF WATERLOO

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

FALL 2019

Analysis of Three Algorithms for Solving MIN-VERTEX-COVER Problem

Submitted By:
Student 1 ID: 20820331 ; NAME- Chitraa Ramachandran
Email : chitraa.ramachandran@uwaterloo.ca
Student 2 ID: 20843127 ; NAME- Bhavani Bandhu
Email : bbandhu@uwaterloo.ca

Faculty of Engineering

Abstract

In this report, we are considering the Vertex Cover problem. The vertex cover of a graph G{V,E} is a set of vertices (C) such that each edge (e) of the graph is incident to at least one vertex of the set (C).It is a classical NP-complete VERTEX COVER problem. We are using three algorithms to analyse about the minimum vertex cover namely: CNF-SAT-VC, Approx-VC1 and Approx-VC2. The analysis is made by running these three algorithms on varying graphs generated by "graphgen" functionality for a given set of vertice number. In minimum at least 10 graphs are considered for a vertex number. The factors on which the analysis is done are: vertex cover obtained, running time and the mean approximation ratio.

1 Introduction

The minimum vertex cover problem is the optimization problem of finding a smallest vertex cover in a given graph. A vertex cover of an undirected graph is a subset of such that, it is a set of vertices where every edge has at least one endpoint in the vertex cover. Such a set is said to cover the edges of G. Eg. The set of all vertices is a vertex cover. So, the vertex cover of the smallest size is called the minimum vertex cover. We have used three different algorithms to find a solution to this problem and they are:

Algorithm 1 (CNF-SAT-VC): the approach uses a polynomial time reduction to CNF-SAT, and then use a SAT solver to get the final results.

Algorithm 2 (APPROX-VC-1): the approach is to pick a vertex of highest degree (most incident edges) and add it to "vertex cover" container then throw away all edges incident on that vertex and repeat those steps till no edges remain.

Algorithm 3 (APPROX-VC-2): pick an edge (u,v) and add both u and v to your vertex cover and throw away all edges attached

The algorithms are implemented in CPP programming language along with the usage of pthreads and cpu time clock to bring in multi-threading and to calculate the running time of each of the above algorithm for a given graph specification.

2 Program Design

We use a polynomial time reduction of vertex cover problem to CNF-SAT. Polynomial-time reduction is an algorithm that runs in time polynomial in its input. It takes as input G,k and produces a CNF formula (Conversion of vertex cover problem to CNF-SAT).CNF is Conjunctive Normal Form described as the conjunction of the disjunctions. Any propositional formula can be converted into an equivalent formula that is in CNF. The CNF-SAT is a Boolean satisfiablity problem which determines if there exists a interpretation that satisfies the Boolean formula.

So, in our program we make four threads, one for the user input/output which takes the input from the user and continues to be running while waiting for the input or until it sees end of file. The other threads are dedicated for each of the three algorithms that is for CNF-SAT-VC, APPROX-VC-1 and APPROX-VC-2. The user input from input/output thread is passed to these threads running the three different algorithms concurrently

The analysis is done by considering the vertex number between the range of 5-50 with a difference of 5.For each of the vertex number consider ten different edge structures are considered,thereby ten different graphs.Their corresponding vertex cover and the run time is considered whose mean value and the standard deviation values helps us to analyse about the efficiency and performance of the three algorithms.The higher value of vertices have higher run time for CNF-SAT-VC.

3 Data Analysis of the Three Algorithm

To compare the running time of the three algorithms we are considering two graphs.One to compare the running time of APPROX VC-1 & APPROX VC-2,whereas the other one to demonstrate the running time of CNF-SAT-VC algorithm.The graph separation between these algorithms are done as there is a huge difference in their running time.The standard deviation that occurs for these inputs are also plotted in the graph.The system on which these threads are run is the same so as to obtain the robustness of the data obtained.This difference is obtained because of the different approach that is followed by these algorithms.The time taken by the APPROX VC-1 is greater because of the greedy approach that it follows(APPROX VC 1 goes through the set of vertices and edges iteratively and will find the vertex with the higher incident degree to find the vertex cover).

For each value of vertex,It is always seen that the performance(Running time and standard deviation) of APPROX-VC-2 is better than APPROX -VC-1.The reason is that APPROX-VC-2 is probabilistic and randomly chooses a vertex and add into the vertex cover whose probability of achieving a optimal vertex cover is minimal.The only similarity between APPROX -VC-1 and APPROX -VC-2 is that there running time increases polynomially with increase in the number of vertices considered.

The CNF-SAT-VC uses SAT solver and the CNF-SAT is an NP-Hard problem.The CNF-SAT-VC running time is very high because of the time taken by the MINISAT to solve

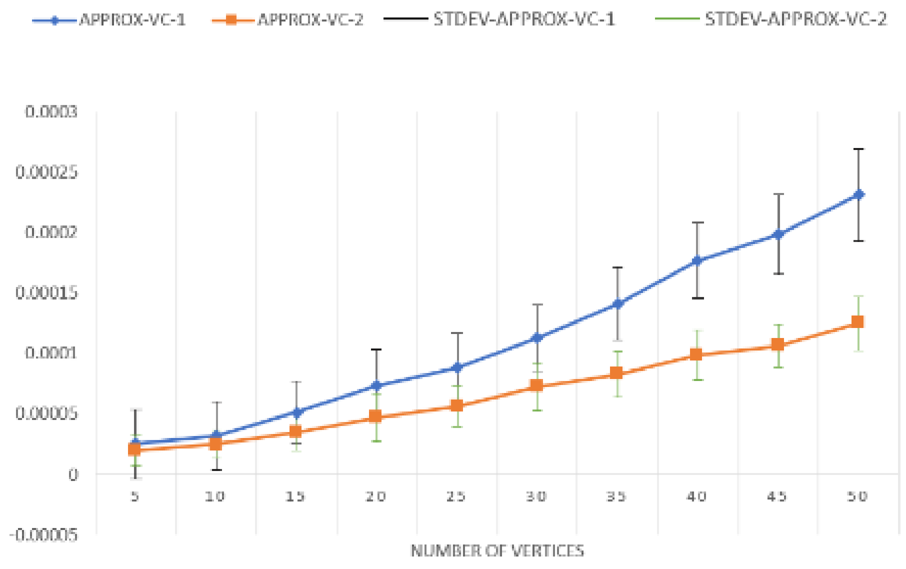

Figure 1: Mean run time plots(in microseconds) for APPROX-VC-1 and APPROX-VC-2.

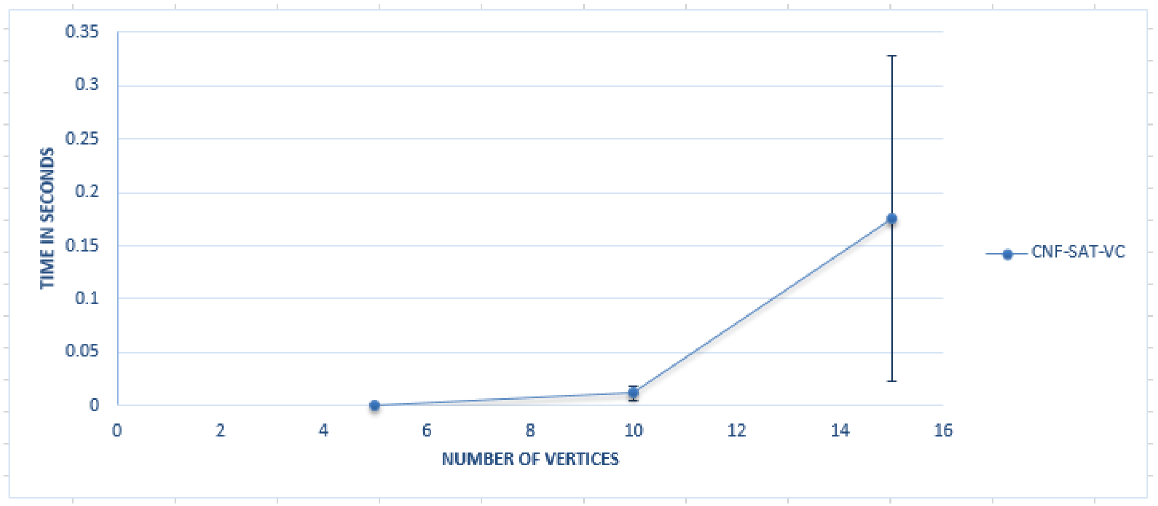

Figure 2: Mean run time plot for CNF-SAT-VC.

the CNF-SAT polynomial reduction of the vertex cover.So,we are going to consider the run time of CNF-SAT-VC only till vertice number less than or equal to 15.Inspite of these issues,the minisat solver helps CNF-SAT-VC to provide the optimal vertex cover for the given set of graph.

Figure 3 plots the mean approximation ratio of APPROX-VC-1 and APPROX-VC-2 when compared to CNF-SAT-VC .The mean approximation ratio for the APPROX-VC-2 is al-

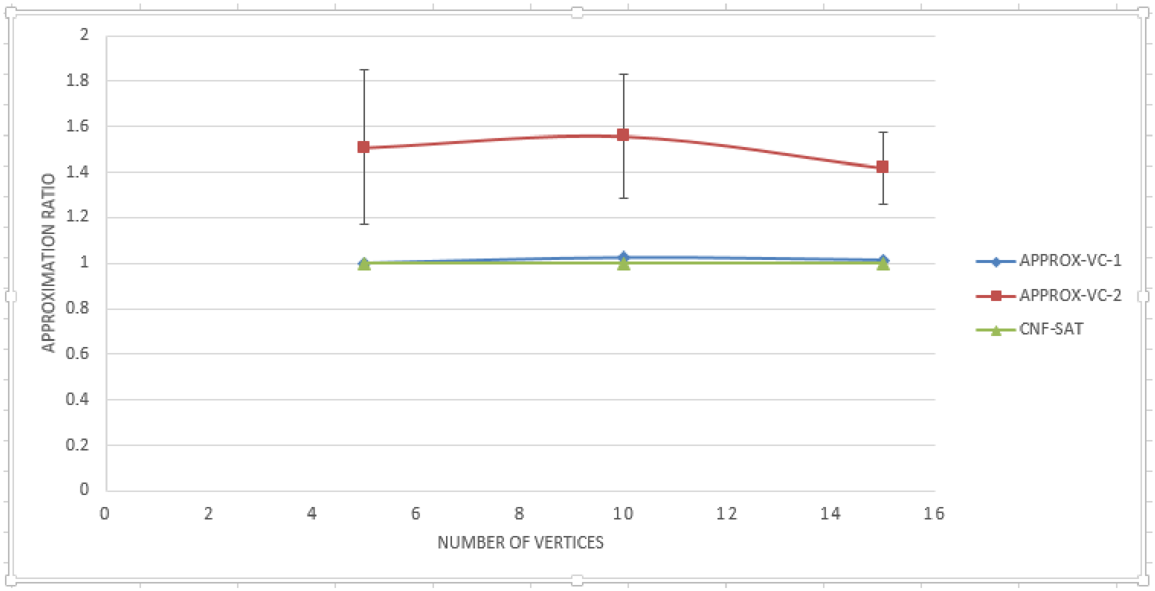

Figure 3: Mean Approximation Ratio.

ways higher when compared to the other two algorithms because of the random approach it follows.Whereas,APPROX-VC-1 has the mean approximation ratio closer to the results obtained by optimal CNF-SAT-VC.High standard deviations are found in APPROX-VC-2 .The minisat CNF-SAT-VC always outputs the optimal vertex cover but it becomes intractable for larger values of inputs.Next to this the algorithm that provides the near optimal result is APPROX-VC-1 as it follows heuristic approach to find the vertex cover.APPROX-VC-2 provides the results that are farthest to the optimal solution.

Conclusion

In conclusion when considering the performance of the three algorithms is determined in terms of both the runtime and the optimal vertex cover.APPROX-VC-1 outperforms both CNF-SAT-VC and APPROX-VC-2 as it returns a near optimal vertex cover and stands robust in the running time even for larger values of inputs in contrast to the exponential increase in the running time by CNF-SAT-VC for higher values of input.