Department of Electrical & Computer Engineering

UNIVERSITY OF WATERLOO

Methods and Tools for Software Engineering
ECE-650
PROJECT REPORT

Prof. Alireza Sharifi

NAME: SHARIQ KHAN
STUDENT ID: 20755086

NAME: AHMAD NAYYAR HASSAN
STUDENT ID: 20742695

SUBMISSION DATE: 9-DECEMBER-2018

INTRODUCTION

VERTEX COVER:

Vertex Cover is NP-Complete problem. In this problem we have an input of undirected graph {G=(V,E)}. Vertex Cover finds the vertices who are adjacent to all the edges, i.e every edge (x,y) either x or y is in the vertex cover list. The goal is to find the minimum sized vertex cover.
For this purpose, we have implemented the following 3 algorithms and have shown their efficiency for different value of vertices {5,10,15,17,20}.

ALGORITHMS:

1)CNF-SAT-VC:

CNF-SAT uses reduction to make formula from an input graph in CNF form. CNF is a conjunction of disjunction of literals i.e the literals are OR with each other and the clauses are AND with each other. The goal of CNF-SAT is to find if the Formula gets Satisfied, for a particular assignment of literals. We use CNF-SAT because it is polynomial time.

2)APPROX-VC-1:

In this algorithm we pick the highest degree vertex i.e the vertex having the most incident edges and move it to a set of vertex cover let's say "VC". After that we remove the edges that are incident to that vertex and keep repeating until all the edges are removed.

3)APPROX-VC-2:

In this algorithm we move every edge into a set lets say "R". After that we pick a random edge (x,y) and add x and y into result set **R** . After that we start to remove edges from **R** which are incident on x or y and then we return the result. We repeat it until all the edges are covered.

EFFICENCY:

To calculate Efficiency, we have analysed our code for running time and approximate ratio.

1)Running time:

To calculate running time we have used "pthread_getcpuclockid" to get thread clock Id and then "clock_gettime" to get the clock time.
We have used 4 threads in our program, and 3 threads used the above 3 algorithms respectively and running time for each thread was calculated as shown in the Excel file

2)Approximate Ratio:

To calculate approximate ratio we have used CNF-SAT-VC as optimal minimum sized vertex cover, thus the approximate ratio is calculated as:

Approximate Ratio $=\frac{Size\ of\ computer\ Vertex\ cover(Approx-VC-1\ ,Approx-VC-2,CNF-SAT-VC)}{Size\ of\ optimal\ minium\ sized\ Vertex\ cover\ (CNF-SAT-VC)}$

EXCEL FILE FOR RUNNING TIME AND APPROXIMATE RATIO:

The running time and approximate ratio are calculated for 5,10,15,17 and 20 vertices.

Vertices		Running Time									Approximate Ratio								
	Serial number	CNF-SAT-VC	CNF-SAT-VC Mean	CNF-SAT-VC Standard Deviation	APPROX-VC-1	APPROX-VC-1Mean	APPROX-VC-1 Standard Deviation	APPROX-VC-2	APPROX-VC-2Mean	APPROX-VC-2 Standard Deviation	CNF-SAT-VC	CNF-SAT-VC Mean	CNF-SAT-VC Standard Deviation	APPROX-VC-1	APPROX-VC-1Mean	APPROX-VC-1 Standard Deviation	APPROX-VC-2	APPROX-VC-2Mean	APPROX-VC-2 Standard Deviation
	1	0.000798272			5.96E-05			4.69E-05			1.00			1.00			2.00		
	2	0.00141265			6.42E-05			4.62E-05			1.00			1.00			1.33		
5	3	0.00075884			6.09E-05			4.60E-05			1.00			1.00			1.33		
	4	0.00137546			0.000118745			9.63E-05			1.00			1.00			1.33		
	5	0.00112739	0.001209394	0.000309917	6.11E-05	6.79E-05	1.79E-05	6.08E-05	5.30E-05	1.59E-05	1.00	1	0	1.00	1.00E+00	0.00E+00	1.00	1.43E+00	3.16E-01
	6	0.00149041			6.39E-05			4.62E-05			1.00			1.00			1.33		
	7	0.00150112			6.29E-05			4.70E-05			1.00			1.00			1.33		
	8	0.0014254			6.37E-05			4.82E-05			1.00			1.00			1.33		
	9	0.000799638			6.20E-05			4.59E-05			1.00			1.00			2.00		
	10	0.00140476			6.24E-05			4.66E-05			1.00			1.00			1.33		

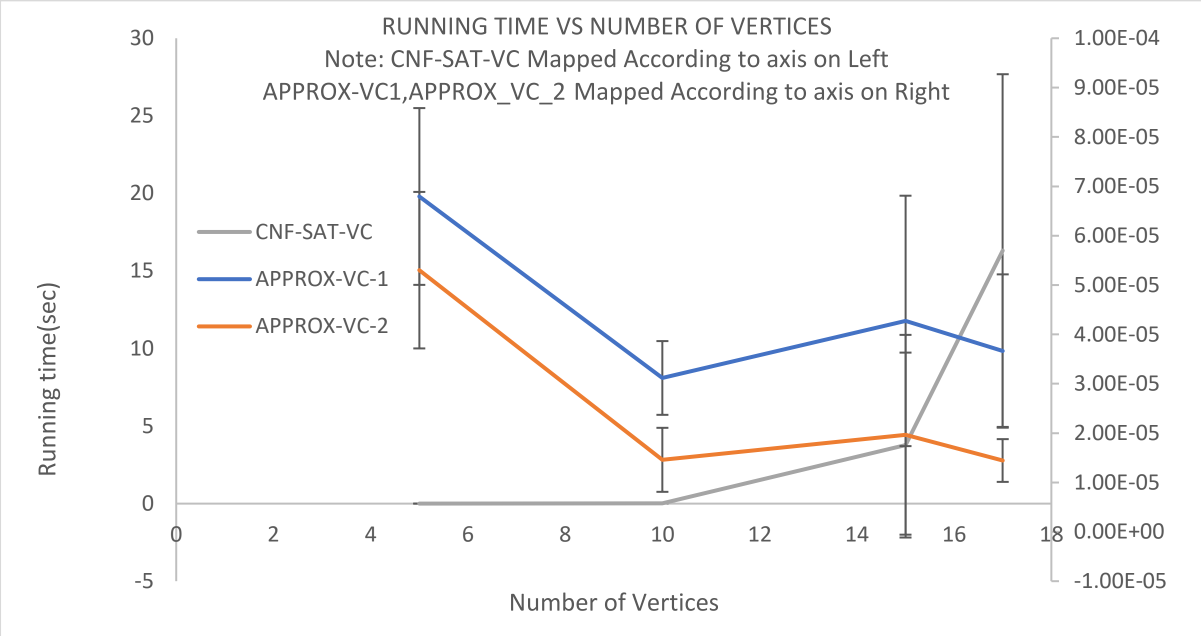
Vertices		Running Time									Approximate Ratio								
	Serial number	CNF-SAT-VC	CNF-SAT-VC Mean	CNF-SAT-VC Standard Deviation	APPROX-VC-1	APPROX-VC-1Mean	APPROX-VC-1 Standard Deviation	APPROX-VC-2	APPROX-VC-2Mean	APPROX-VC-2 Standard Deviation	CNF-SAT-VC	CNF-SAT-VC Mean	CNF-SAT-VC Standard Deviation	APPROX-VC-1	APPROX-VC-1Mean	APPROX-VC-1 Standard Deviation	APPROX-VC-2	APPROX-VC-2Mean	APPROX-VC-2 Standard Deviation
	1	0.0122464			3.17E-05			1.07E-05			1			1.20			1.60		
	2	0.00492217			1.57E-05			2.30E-05			1			1.00			1.50		
10	3	0.009035			4.73E-05			1.12E-05			1			1.00			2.00		
	4	0.00498606			3.11E-05			2.50E-05			1			1.00			2.00		
	5	0.0118113	0.008801521	0.00240585	3.09E-05	3.12E-05	7.47E-06	1.06E-05	1.46E-05	6.49E-06	1	1	0	1.00	1.04E+00	8.43E-02	1.60	1.59E+00	2.69E-01
	6	0.00857056			3.10E-05			1.05E-05			1			1.00			1.20		
	7	0.00890483			3.07E-05			2.38E-05			1			1.00			1.60		
	8	0.00999368			3.10E-05			1.03E-05			1			1.00			1.60		
	9	0.00886884			3.14E-05			1.07E-05			1			1.20			1.60		
	10	0.00867637			3.09E-05			1.01E-05			1			1.00			1.20		

Vertices		Running Time									Approximate Ratio								
	Serial number	CNF-SAT-VC	CNF-SAT-VC Mean	CNF-SAT-VC Standard Deviation	APPROX-VC-1	APPROX-VC-1Mean	APPROX-VC-1 Standard Deviation	APPROX-VC-2	APPROX-VC-2Mean	APPROX-VC-2 Standard Deviation	CNF-SAT-VC	CNF-SAT-VC Mean	CNF-SAT-VC Standard Deviation	APPROX-VC-1	APPROX-VC-1Mean	APPROX-VC-1 Standard Deviation	APPROX-VC-2	APPROX-VC-2Mean	APPROX-VC-2 Standard Deviation
	1	0.959585			2.09E-05			1.18E-05			1			1.14			1.71		
	2	0.957267			4.49E-05			2.50E-05			1			1.00			1.43		
15	3	12.2385			9.14E-05			7.60E-05			1			1.13			1.50		
	4	1.10224			8.64E-05			1.48E-05			1			1.14			1.71		
	5	17.3472	3.77421297	5.960552901	3.42E-05	4.27E-05	2.54E-05	1.15E-05	1.97E-05	2.02E-05	1	1	0	1.00	1.06E+00	7.52E-02	1.50	1.64E+00	1.78E-01
	6	2.4777			3.30E-05			1.15E-05			1			1.00			1.71		
	7	0.873945			3.25E-05			1.14E-05			1			1.00			1.71		
	8	1.06459			3.23E-05			1.18E-05			1			1.00			1.71		
	9	0.0902707			3.21E-05			1.15E-05			1			1.17			2.00		
	10	0.630832			1.95E-05			1.15E-05			1			1.00			1.43		

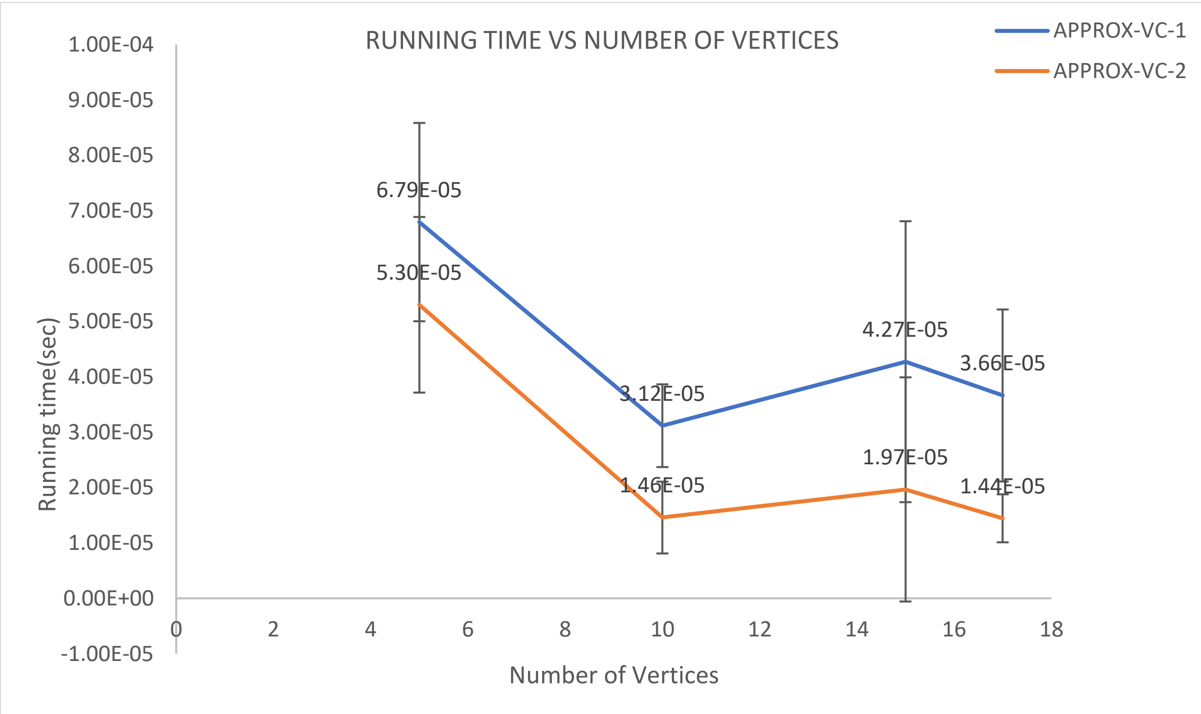
Vertices		Running Time									Approximate Ratio								
	Serial number	CNF-SAT-VC	CNF-SAT-VC Mean	CNF-SAT-VC Standard Deviation	APPROX-VC-1	APPROX-VC-1Mean	APPROX-VC-1 Standard Deviation	APPROX-VC-2	APPROX-VC-2Mean	APPROX-VC-2 Standard Deviation	CNF-SAT-VC	CNF-SAT-VC Mean	CNF-SAT-VC Standard Deviation	APPROX-VC-1	APPROX-VC-1Mean	APPROX-VC-1 Standard Deviation	APPROX-VC-2	APPROX-VC-2Mean	APPROX-VC-2 Standard Deviation
	1	12.3263			5.34E-05			1.22E-05			1			1.00			1.50		
	2	0.704484			2.27E-05			1.20E-05			1			1.00			1.71		
17	3	43.198			2.37E-05			1.24E-05			1			1.00			1.25		
	4	14.8344			2.43E-05			1.43E-05			1			1.13			2.00		
	5	26.4403	16.3137984	11.36860272	2.81E-05	3.66E-05	1.55E-05	2.60E-05	1.44E-05	4.33E-06	1	1	0	1.00	1.03E+00	5.27E-02	1.75	1.62E+00	2.10E-01
	6	12.5739			3.22E-05			1.63E-05			1			1.00			1.50		
	7	11.6053			5.15E-05			1.23E-05			1			1.00			1.50		
	8	17.4497			3.86E-05			1.50E-05			1			1.00			1.75		
	9	11.1242			2.49E-05			1.18E-05			1			1.13			1.50		
	10	12.8814			6.70E-05			1.22E-05			1			1.00			1.75		

Vertices		Running Time				Approximate Ratio		
	Serial number	CNF-SAT-VC	CNF-SAT-VC Mean	APPROX-VC-1	APPROX-VC-2	CNF-SAT-VC	APPROX-VC-1	APPROX-VC-2
20	1	1364.2	1364.2	4.40E-05	1.42E-05	1	1.11	1.56

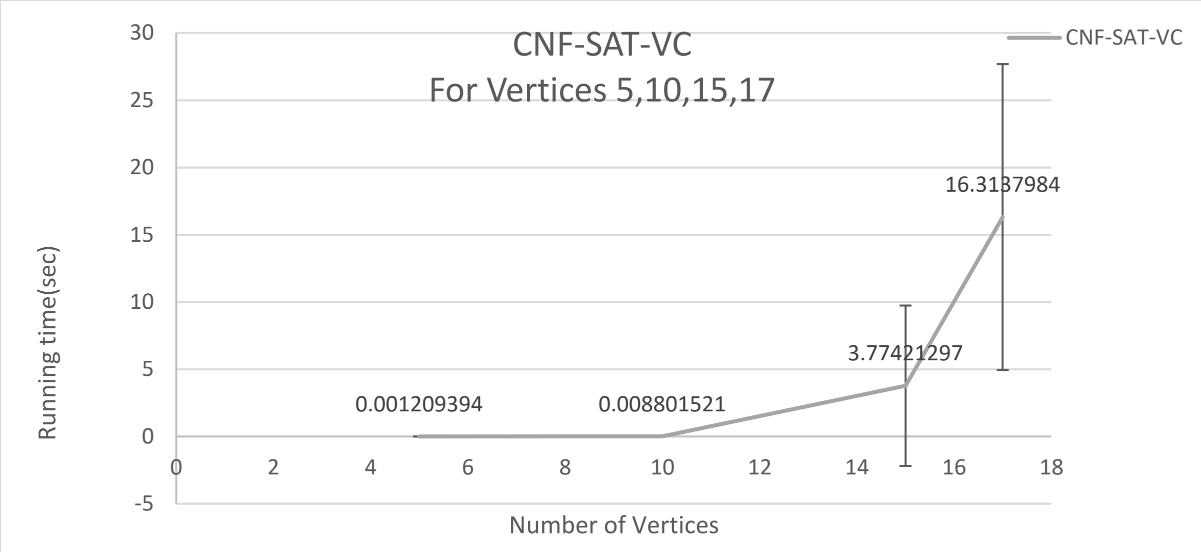
GRAPH PLOTS:



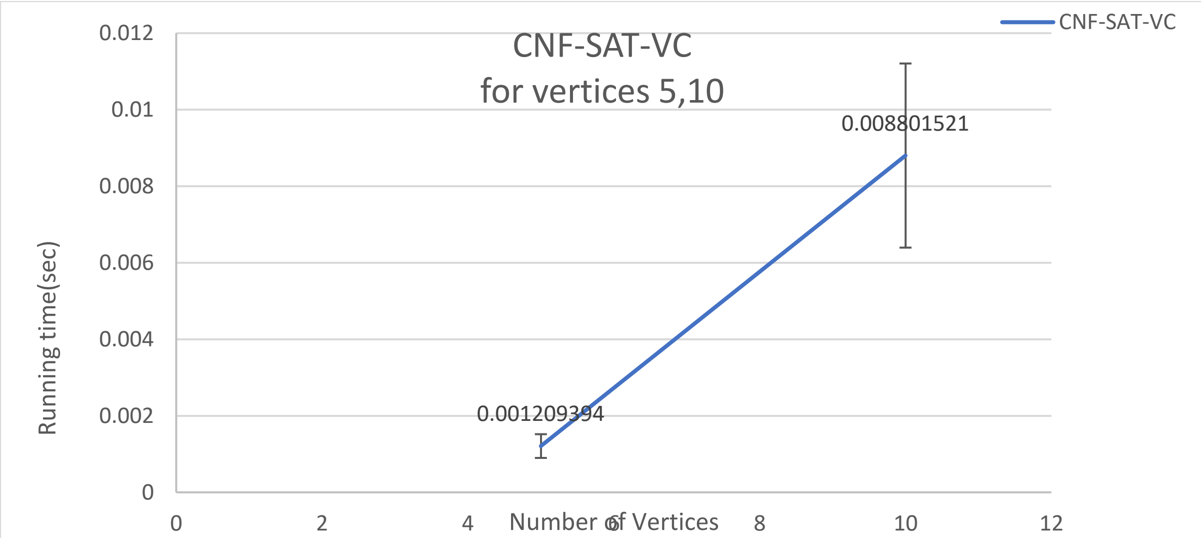
GRAPH1: RUNNING TIME VS NUMBER OF VERTICES FOR CNF-SAT-VC, APPROX-VC-1,APPROX-VC-2



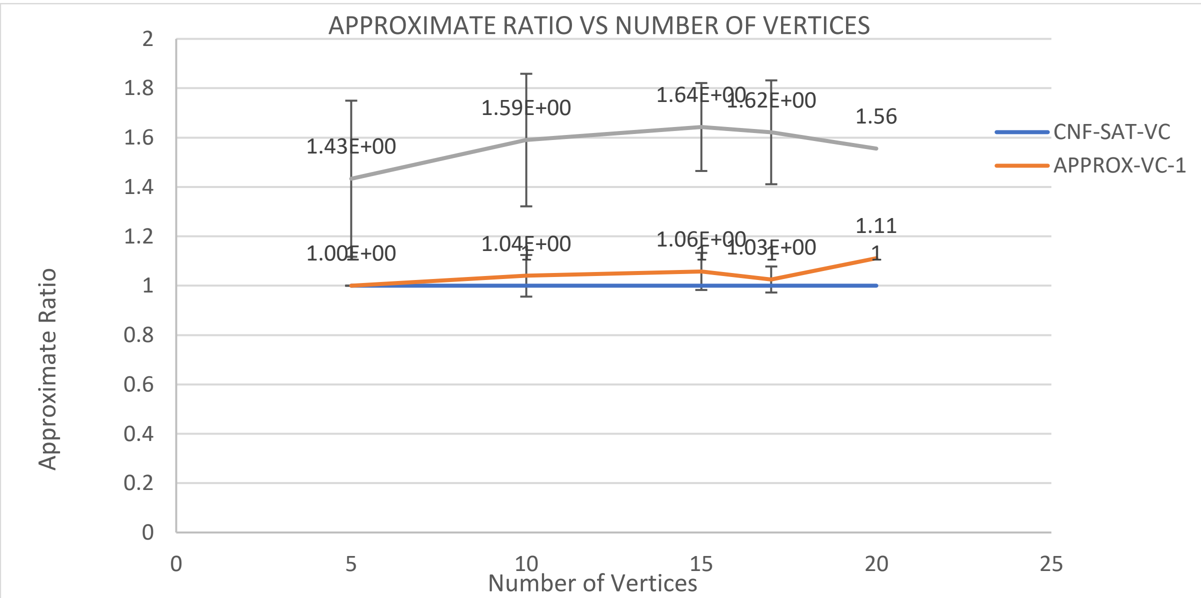
GRAPH2: RUNNING TIME VS NUMBER OF VERTICES FOR APPROX-VC-1,APPROX-VC-2



GRAPH3: RUNNING TIME VS NUMBER OF VERTICES FOR CNF-SAT-VC FOR VERTICES 5,10,15,17



GRAPH4: RUNNING TIME VS NUMBER OF VERTICES FOR CNF-SAT-VC FOR VERTICES 5,10



GRAPH5: APPROXIMATION RATIO VS NUMBER OF VERTICES FOR CNF-SAT-VC, APPROX-VC-1,APPROX-VC-2

ANALYSIS

For Running Time:

1) CNF-SAT-VC:

In the Graph 4 it can be seen that for smaller values of vertices(5,10) the running time is around exponential -3 secs and in Graph 3 it can be seen that the running time for vertices (10,15) shoots to exponential 0 secs and for 20 vertices the time taken was 1364.2 secs this trend is because of the following reason

Lower the number of vertices lower are the number of literals and thus lower the value of k thus lower number of clauses. Since our algorithm for CNF-SAT-VC is using linear search for “k” thus for smaller values of vertices it takes less time to find satisfiability because less literals and clauses and for larger values of vertices, larger the number of literals and clauses and also due to linear searching it is taking a lot of time.

2) APPROX-VC-1

Since this algorithm first looks for the vertices of highest degree and then finds the vertex cover list, also it visits each node in every iteration, that’s why this algorithm has a little bit higher time as compared to APPROX-VC-2. Although for greater number of nodes we would expect the running time to decrease as it is shown by the graph 2.

3) APPROX-VC-2

By graph2 we can see that the running time for APPROX-VC-2 is slightly less than APPROX-VC-1 because this algo doesn’t first find highest degree vertex. Also in each iteration it marks nodes as visited and doesn’t go back to that node again , so number nodes to visit in next iteration is less.

For Approximation Ratio:

1) CNF-SAT-VC:

Since CNF-SAT-VC guarantees to find the minimum number of vertex cover thus its approximation ratio is equal to 1 in all vertex cases

2) APPROX-VC-1:

By graph 5 we can see that the approximation ratio of this algo is much better than Approx-VC-2 because it finds the highest degree vertex.

3) APPROX-VC-2:

We can see that this algorithm has the worst approximation ratio among the other 2 because, it doesn’t find vertex with the highest degree instead it takes an arbitrary edge. i.e it takes two nodes and thus it has more elements in the vertex cover list as compared to other two.