Algorithm Analysis

su wang ID:20625858

Abstract*-Three algorithms to solve vertexes cove problem. The first one is called CNF-SAT-VC which is based on the SAT algorithm. The second one is called APPROX-VC-1 which uses vertex from the rank of degree to filter the covered edges. The last one is called APPROX-VC-2 which uses the two vertexes of every edge to filter the covered edges .The three algorithms have different efficiency for vertex cover problem. CNF-SAT-VC can output the minimum-sized vertex cover, APPROX-VC1 is a little worse than CNF-SAT-VC, APPROX-VC2 is the worst one. But with the |V| increasing, the consuming time for CNF-SAT-VC has an apparent increasing trend and very big. The time for APPROX-VC1 increase not so quickly and APPROX-VC2 increases very little and mostly keeps the same time.*

I. 2-18 VERTEX COVER RUNNING TIME

With the output of three algorithms for the graph of 2-18 vertexes with step 2, figure 1,2,3,4 is their consuming time.

Figure 1 is CNF-SAT-CV time. During input vertexes of 2-14, the time is very little (<1s, most in μs), the deviation increases very small and is very little, while 14-18 input, the time abrupt rises hugely (5s< time <800s), the deviation increases very greatly and is very big. These results show that CNF-SAT-CV could have a very high efficiency for the less vertexes input (<14), but with the vertexes increasing, the time could be very big. The reason is that with vertexes increasing, the vertexes and the edges increase so the SAT algorithm's clauses will increase hugely which would costs a lot of time, the graphs become much more complex and the same |V| could has much more kinds of graphs lead to the deviation become much bigger.

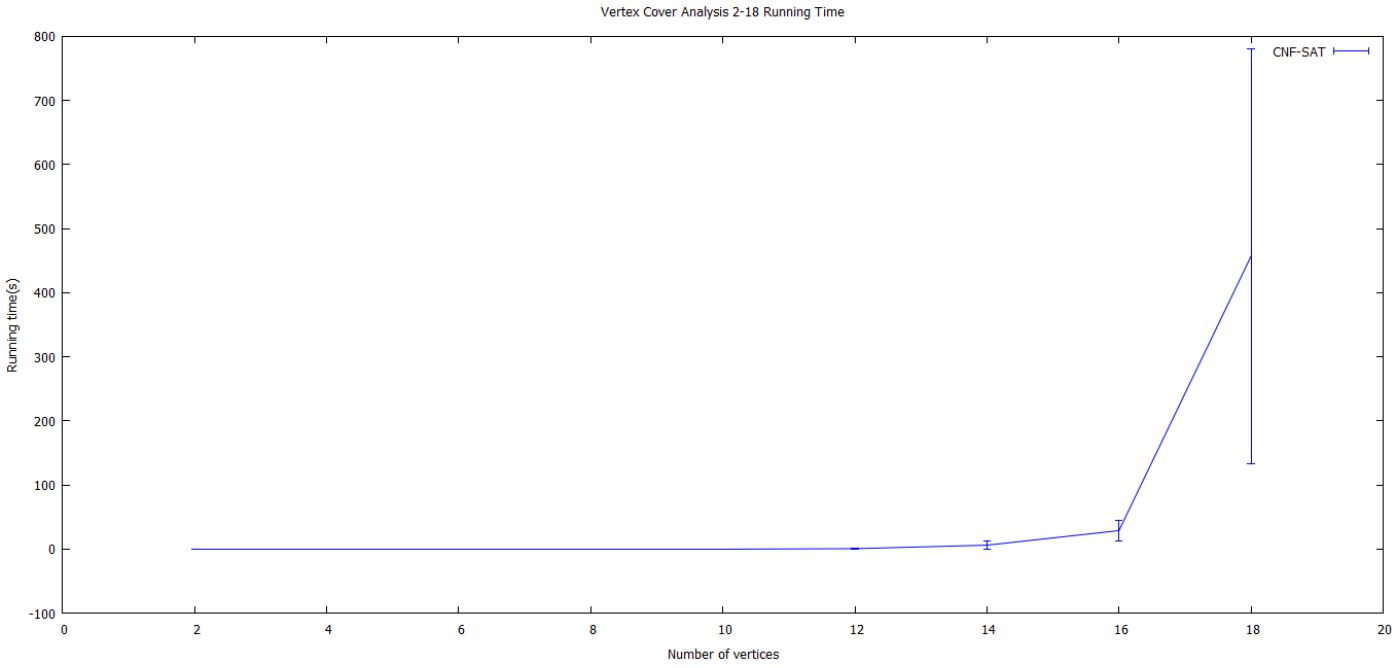

Figure 1. CNF-SAT-CV time

Figure 2 is APPROX-CV1 time. With the vertexes increasing, the time increases and the value is very little (μs level) but the ratio of increment increases and the deviation also increase. The reason is that with the number of vertexes increasing, the vertexes and the edges increase so APPROX-CV1 algorithm needs more and more time to filter the vertexes; the graphs become much more complex and the same |V| could has much more kinds of graphs lead to the deviation become much bigger.

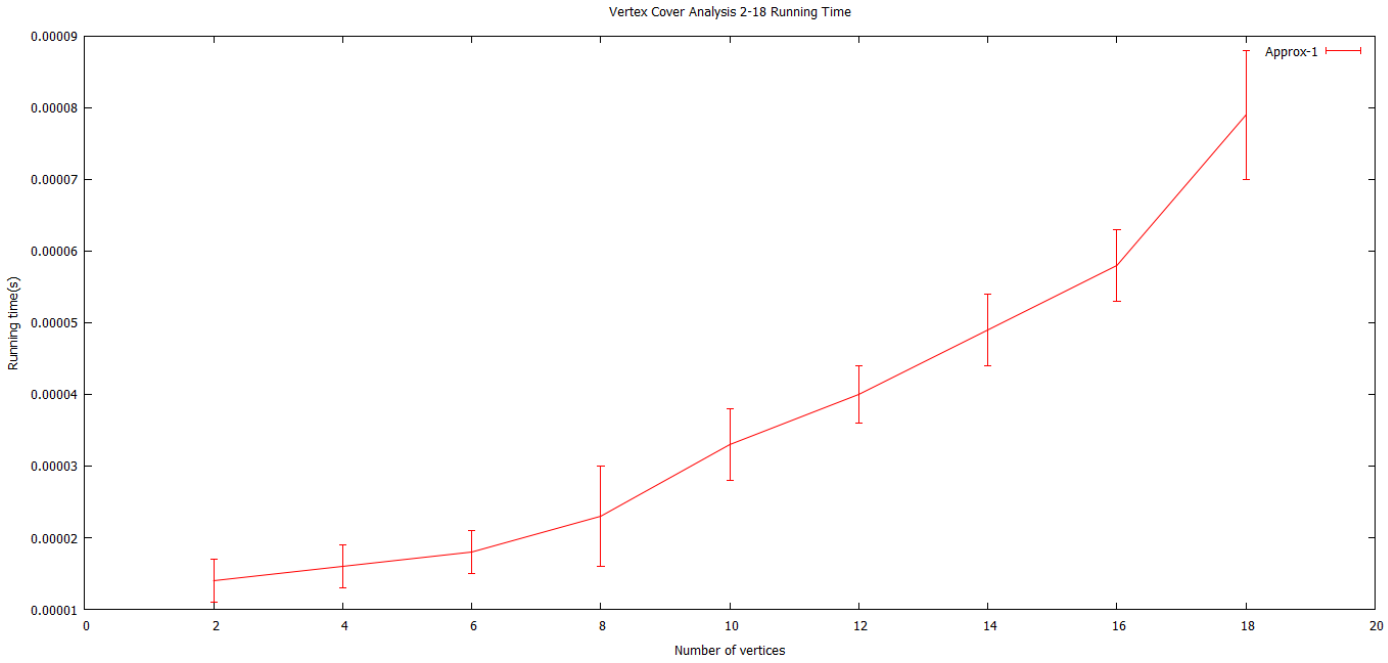

Figure 2. APPROX-CV1 time

Figure 3 is APPROX-CV2 time. With the vertexes increasing, the time increases and the value is very much less (10^{-1}μs level) but the ratio of increment increases and the deviation has no law. The reason is that with the number of vertexes increasing, the vertexes and the edges increase so APPROX-CV2 algorithm needs more and more time to filter the vertexes of edges. The graphs become much more complex and the same |V| could has much more kinds of graphs should have leaded to the deviation become much bigger but the result is not this law, the reason is that APPROX-CV2 algorithm every time choose the first one of the edges to filter edges then the next one … and the vertexes in this rank of edges will have different degree of each edge .

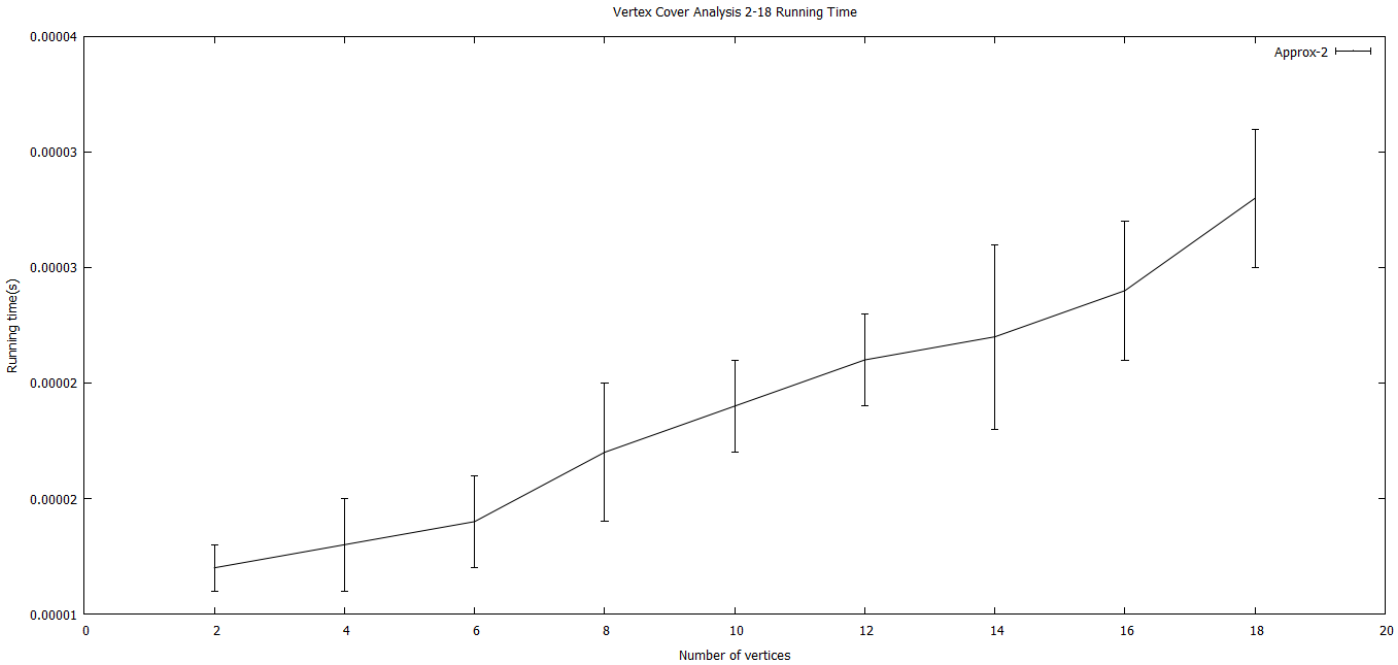

Figure 3. APPROX-CV2 time

Figure 4 is CNF-SAT-CV, APPROX-CV1, APPROX-CV2 time. Figure 4 shows that the three algorithms' time is very little for 2-14 vertexes, but the CNF-SAT-CV spends much greater time and the other two increases very little.

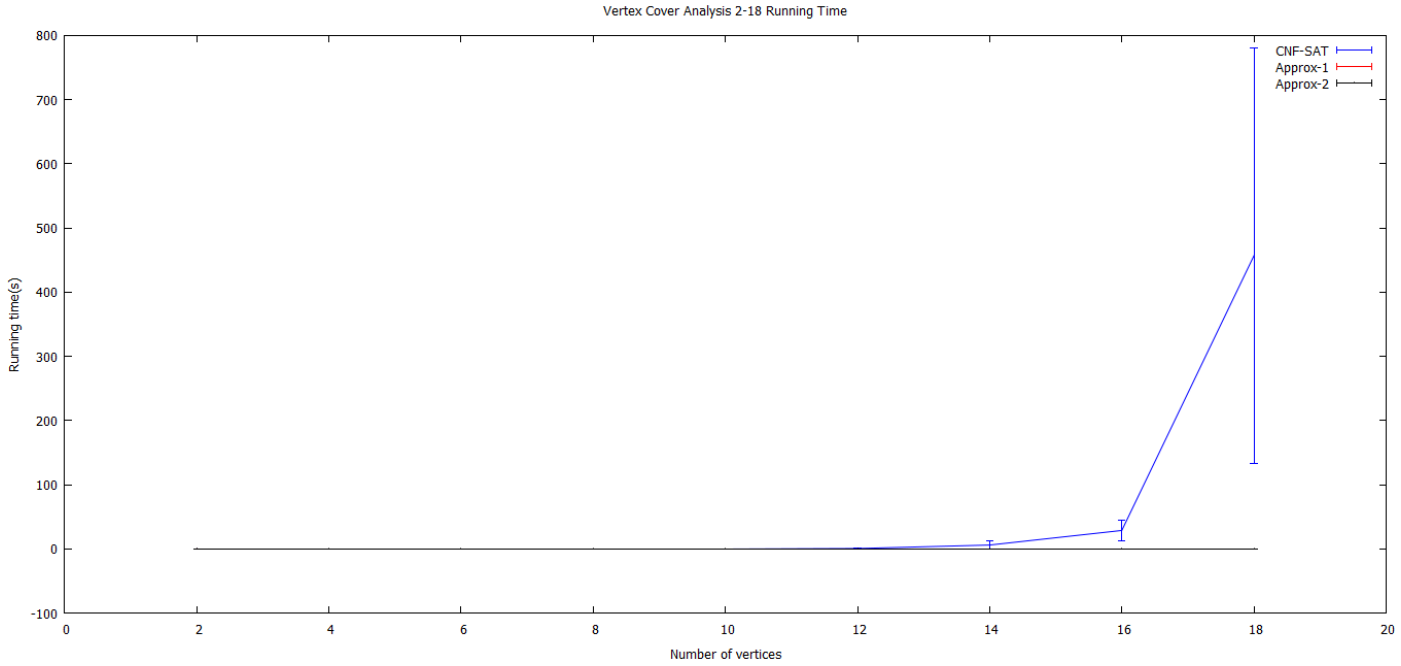

Figure 4. CNF-SAT-CV, APPROX-CV1, APPROX-CV2 time

II. 2-18 VERTEX COVER APPROXIMATION RATIO

Because CNF-SAT-CV always output the minimum vertexes, the output of CNF-SAT-CV is used as the criterion. Using ratio = N1/N2 to measure the APPROX-CV1 and APPROX-CV2 (N1: output of CNF-SAT-CV or APPROX-CV1 or APPROX-CV2, N2: output of CNF-SAT-CV) ,

Figure 5 is CNF-SAT-CV, APPROX-CV1 ratio. With the vertexes increasing, APPROX-CV1's ratio is very small at beginning, then increases, finally decreases and the deviation has the same law. In all, the value of ratio is small (<1.15). The reason is that at beginning, the number of vertexes and the edges are very less, so APPROX-CV1 could have the same ratio, then with the number of vertexes increasing, the vertexes and the edges increase, the graphs become much more complex and the same |V| could has much more kinds of graphs and vertexes have different degree which leads to APPROX-CV1 ratio increases. But because the graph are connected graphs, so the vertex could link much more edges which leads APPROX-CV1 ratio become more and more near CNF-SAT-CV.

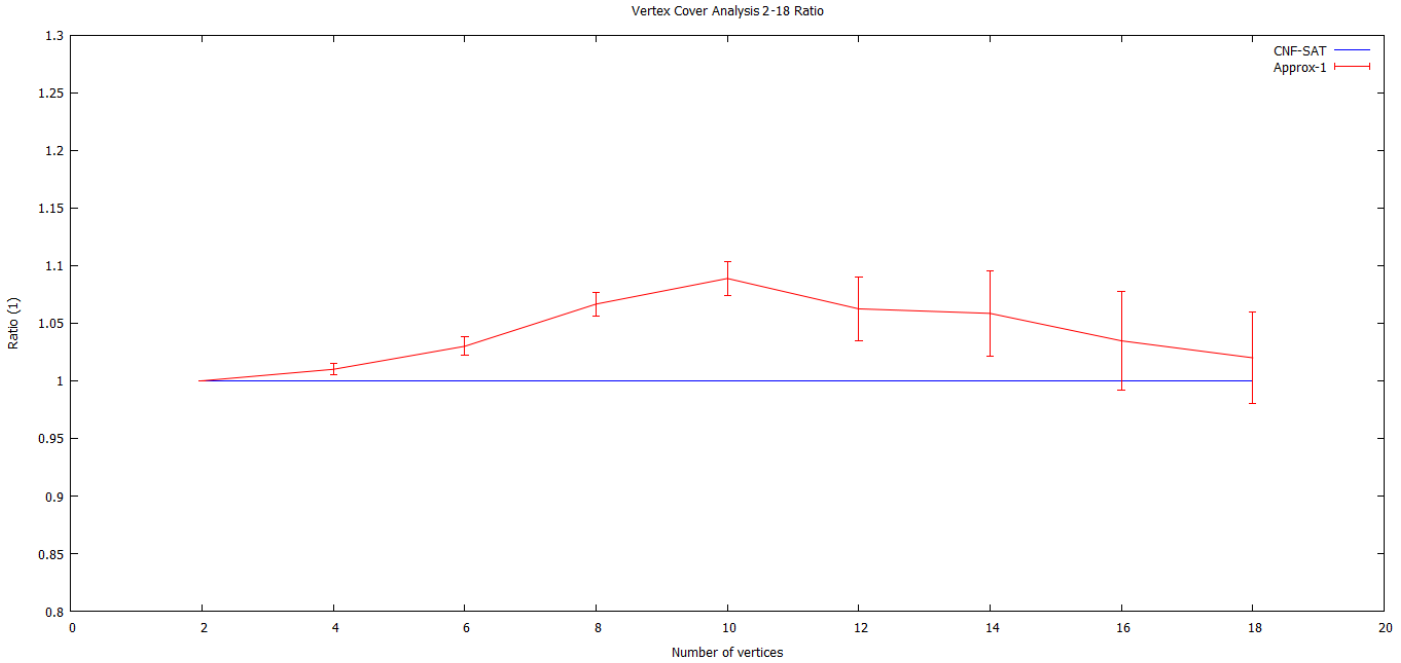

Figure 5. CNF-AST-CV, APPROX-CV1 ratio

Figure 6 is CNF-SAT-CV, APPROX-CV2 ratio. With the vertexes increasing, APPROX-CV2's ratio is big at beginning ,then decreases and the deviation has the same law. In all, the biggest value of ratio is less than about 2.3. The reason is that at beginning, the number of vertexes and the edges are very less and the APPROX-CV2 should output diploid number of vertexes, although the first vertex has covered all edges, so APPROX-CV2 could have big ratio and big deviation at beginning, then with the number of vertexes increasing, the vertexes and the edges increase, the graphs become much more complex and the same |V| could has much more kinds of graphs and vertexes have different degree which leads to APPROX-CV2 ratio increases. But because the graph are connected graphs, so the vertex could link much more edges which leads APPROX-CV2 ratio become more and more near CNF-SAT-CV. Because APPROX-CV2 algorithm every time choose the first one of the edges to filter edges then the next one … and the vertexes in this rank of edges will have different degree of edges, the deviation has the trend to decrease but not must according to the law that the former is less than the next one .

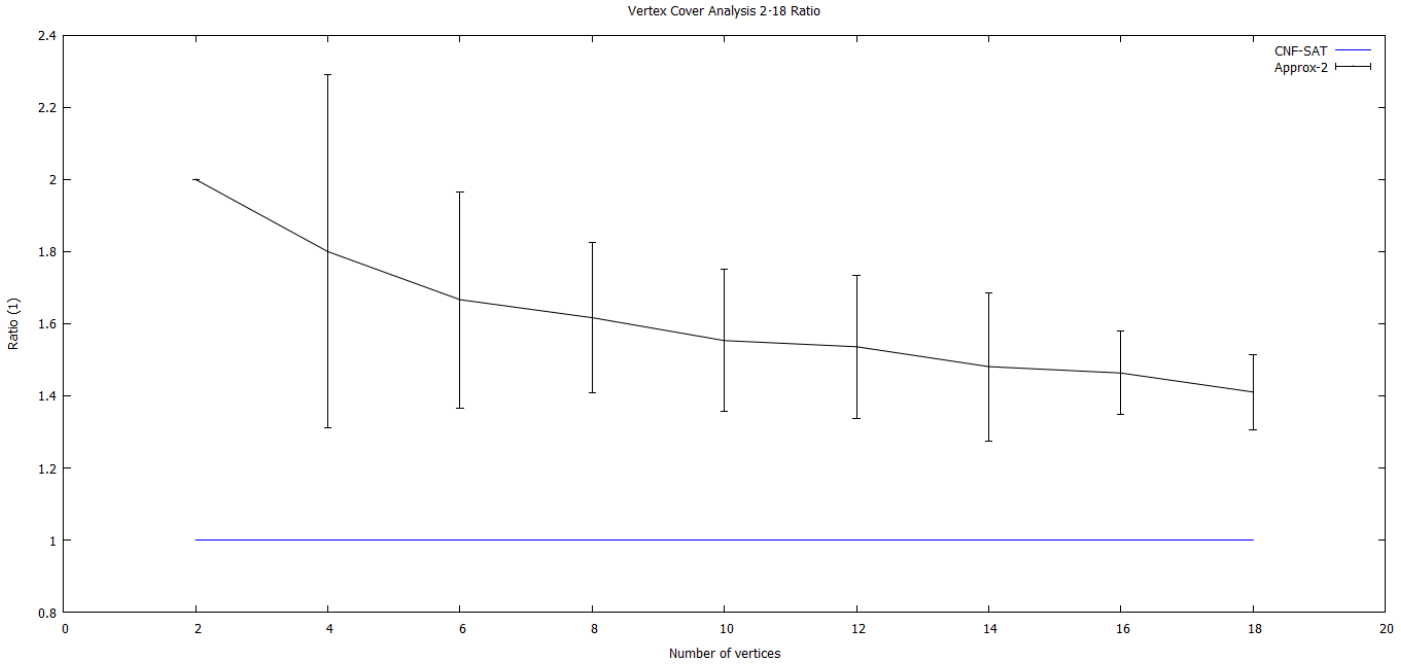

Figure 6. CNF-AST-CV, APPROX-CV2 ratio

Figure 7 is CNF-AST-CV, APPROX-CV1, APPROX-CV2 ratio. With the comparison, APPROX-CV1's ratio is much nearer CNF-SAT-CV's than APPROX-CV2.

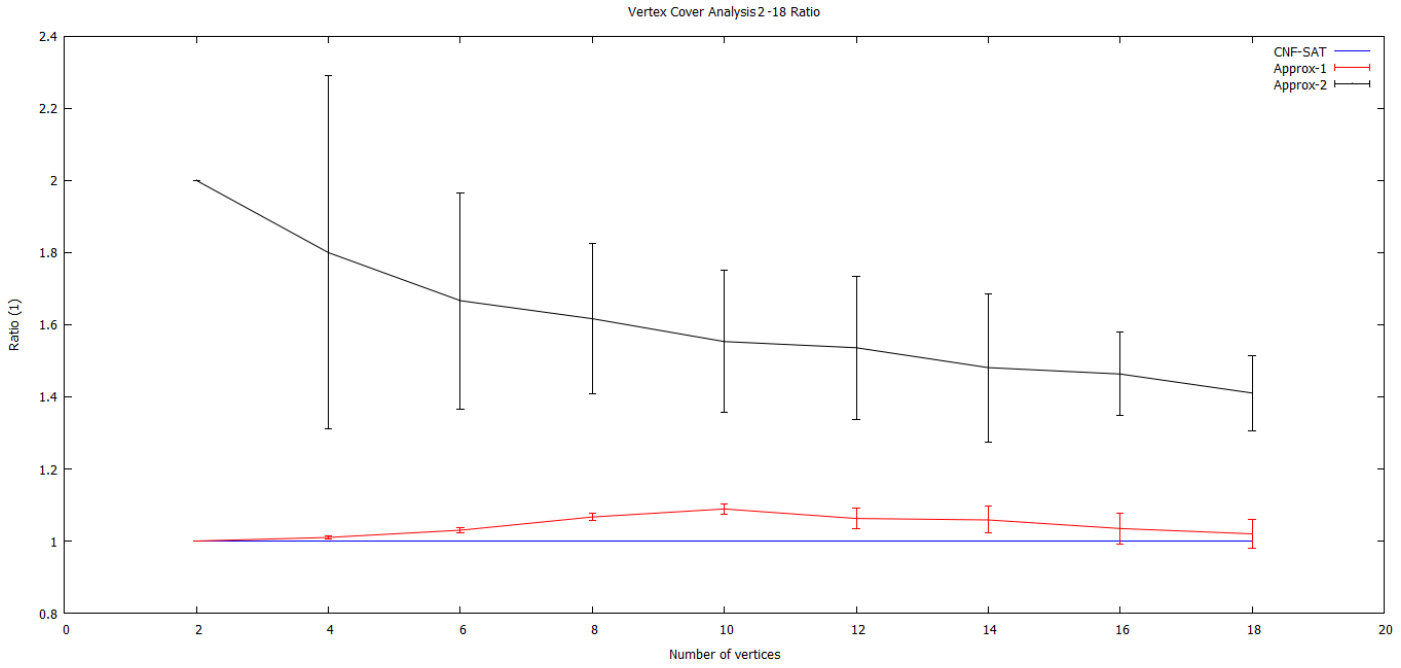

Figure 7. CNF-AST-CV, APPROX-CV1,APPROX-CV2 ratio

III. 5-50 VERTEX COVER RUNNING TIME

Figure 8 is APPROX-CV1 time. With the vertexes increasing, the time increases and the value is little (10^{-2}s level) but the ratio of increment increases and the deviation also increase. The reason is that with the number of vertexes increasing, the vertexes and the edges increase so APPROX-CV1 algorithm needs more and more time to filter the vertexes; the graphs become much more complex and the same |V| could has much more kinds of graphs lead to the deviation become much bigger.

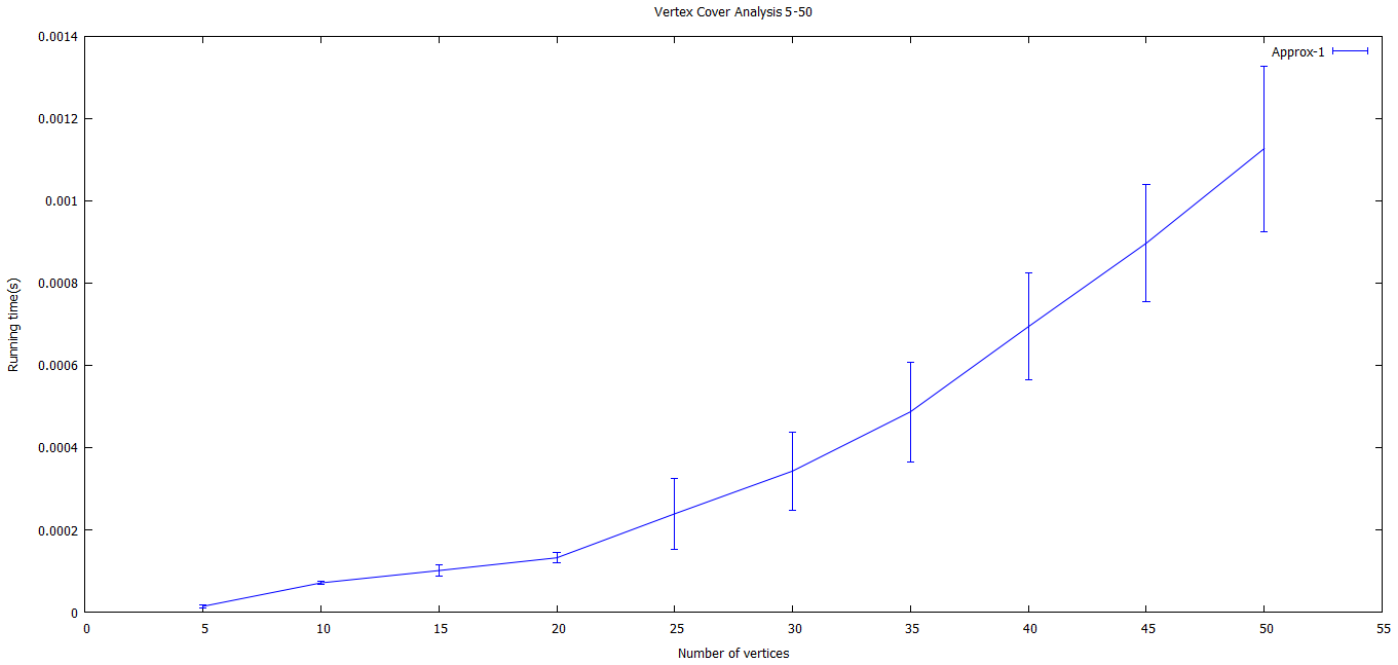

Figure 8. APPROX-CV1 time

Figure 9 is APPROX-CV2 time. With the vertexes increasing, the time increases and the value is very much less (10^{-1}μs level) but the ratio of increment increases and the deviation has the trend that the value is small at beginning then becomes big. The reason is that with the number of vertexes increasing, the vertexes and the edges increase so APPROX-CV2 algorithm needs more and more time to filter the vertexes of edges. The graphs become much more complex and the same |V| could has much more kinds of graphs should have leaded to the deviation become much bigger but the result is not so strictly, like the running time of APPROX-CV2 in 2-18.

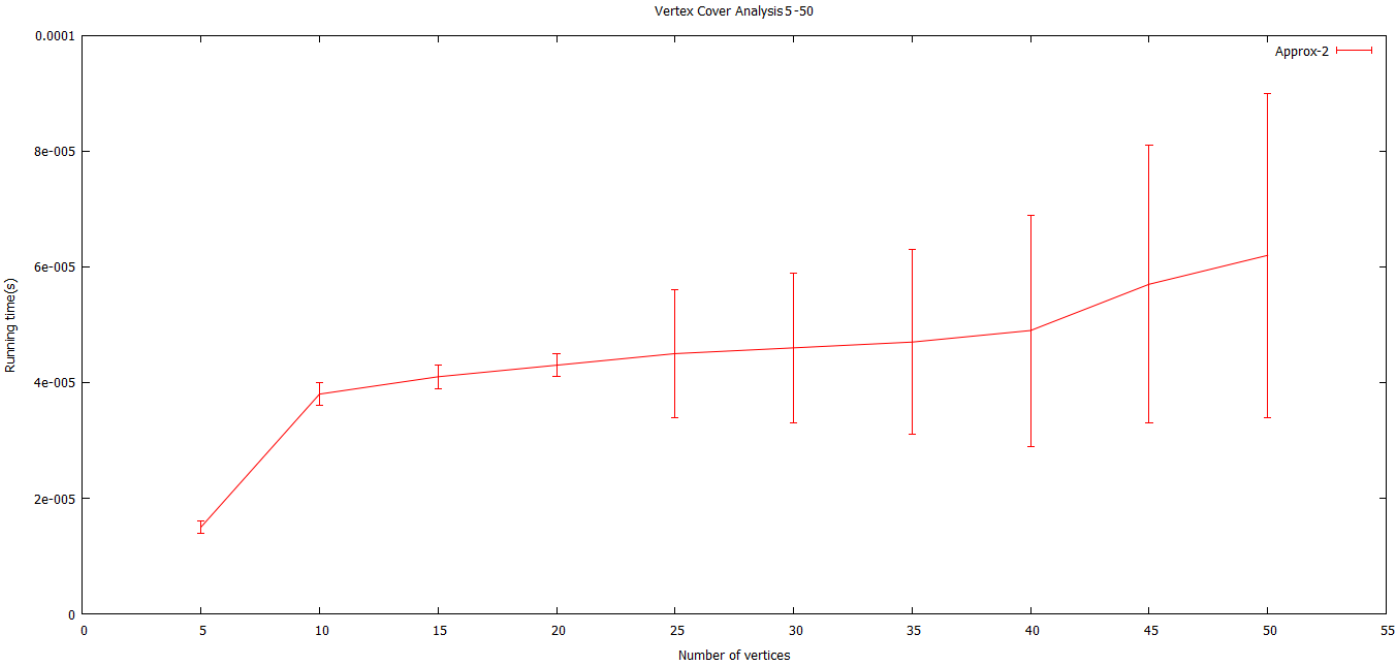

Figure 9. APPROX-CV2 time

Figure 10 is APPROX-CV1, APPROX-CV2 time. With the comparison, although the two algorithms' running time increases, APPROX-CV2's running time increases much smaller than APPROX-CV1's.

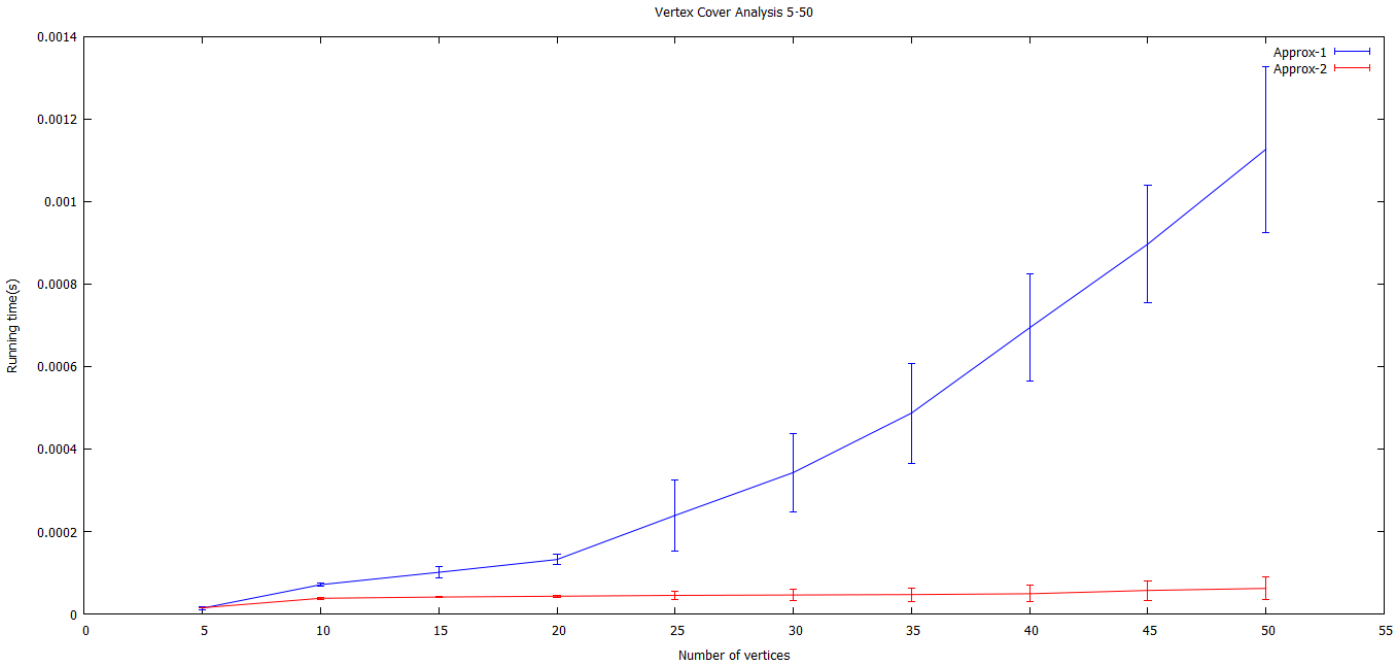

Figure 10. APPROX-CV1,APPROX-CV2 time

IV. CONCLUSION

With these analysis, the three algorithms have different advantages and disadvantages. From the approximation ratio, CNF-SAT-CV is the best, the next is APPROX-CV1, then APPROX-CV2. From the time consuming, the best is APPROX-CV2, the second is APPROX-CV1, then CNF-SAT-CV. Considering time and ratio, under 14 vertexes' input, CNF-SAT-CV is the best, then APPROX-CV1, the last one is APPROX-CV2; when vertexes is bigger than 14, the best one is APPROX-CV1, the next is APPROX-CV2, the last is CNF-SAT-CV.

So, choosing which algorithm should base on the number of vertexes for the input and the priority of goal (ratio or time or ratio and time)