Assignment 5 Report

Student ID: 20601564 **Student Name: Zhang Xiyue**

Introduction

1 input graph sample

The first sample $|V| \in [4,16]$ increment of 2, that is 4 6 ..16. It is designed for running time and approximate ratio in CNF-SAT, Approx-1 and Approx-2. It excludes $|V| \in [18,20]$ because there is an error called 'Cputime limit exceeded' on eceLinux5 when calculating data of $|V| \in [18,20]$.

The second sample $|V| \in [20,50]$ increment of 5, that is 20 25 30 ..50. It is designed for running time in Approx-1, Approx-2. It excludes CNF-SAT because CNF-SAT needs much time to solve vertex cover when $|V|>20$

2 data measuring method

The running time is calculated by the function of pthread_getcpuclockid(), while approximate ratio is calculated by return value from each thread. The data is stored in global variables, and there is a function in a5-ece650.c to calculate average and deviation. Then average and deviation are recorded into amp.dat and transfer into statistic chart by gnuplot.

Data analysis

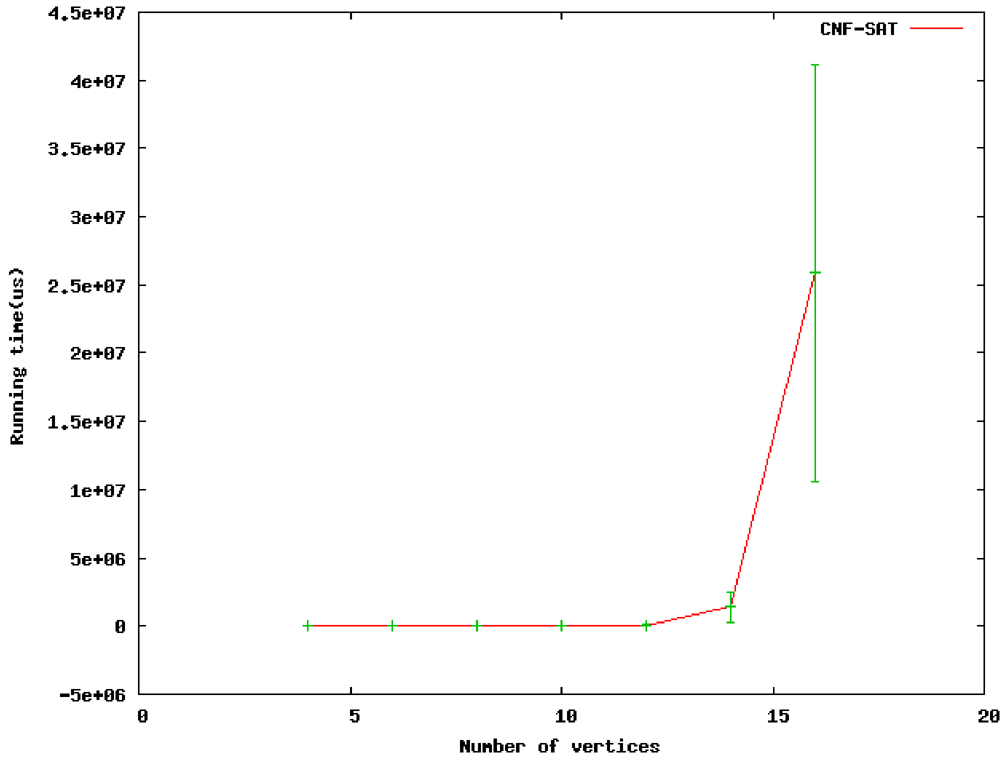

FIG. 1 Running Time For CNF-SAT

As the running time of CNF-SAT is far more than that of Approx-1 and Approx-2 before 16 vertexes, the graph is divided into FIG.1 graph (Running Time For CNF-SAT) and FIG.2 graph (Running Time For Approx-1 and Approx-2)

The FIG.1 graph shows the increasing trend of CNF-SAT. The line keeps stable before 12 vertexes and increases rapidly after 12 vertexes just like exponential growth.

The reason is that CNF-SAT needs far more time to solve the large-size vertex cover problem. According to the reduction algorithm in assignment 4, the propositional logic formula is comprised of clauses. The number of clauses is $k + n\binom{k}{2} + k\binom{n}{2} + |E|$. (the clause number grows in an exponential manner with vertex cover size) .It means that large-size vertex cover leads to more clauses in propositional logic formula. So the increasing speed of running time grows rapidly.

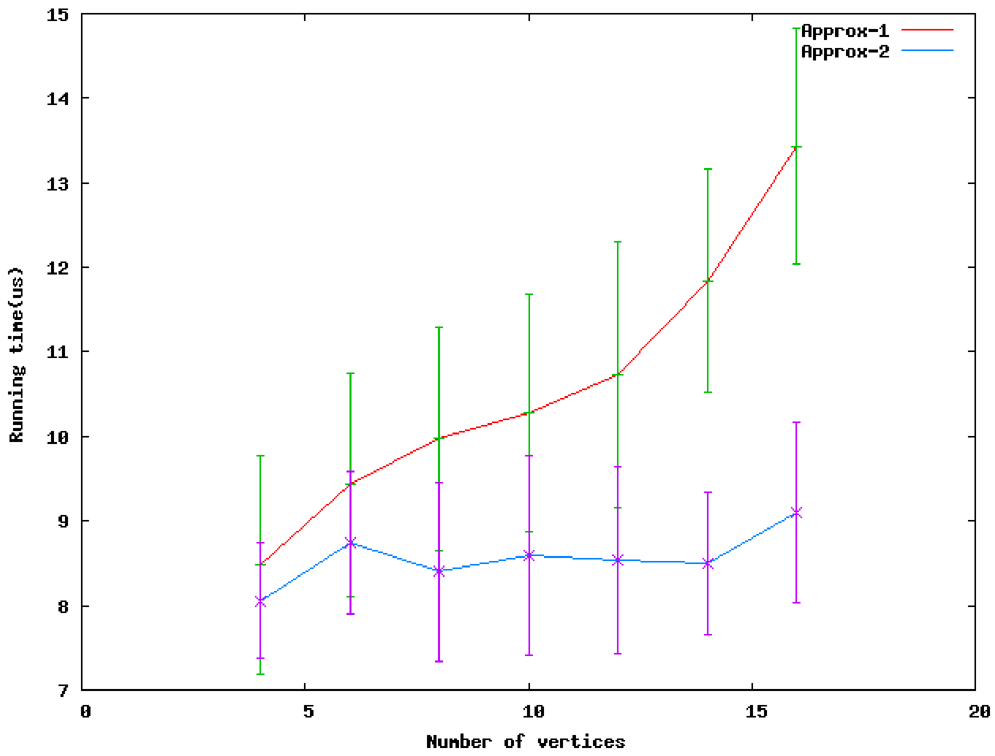

FIG. 2 Running Time For Approx-1 and Approx-2(from 4 to 16 vertexes)

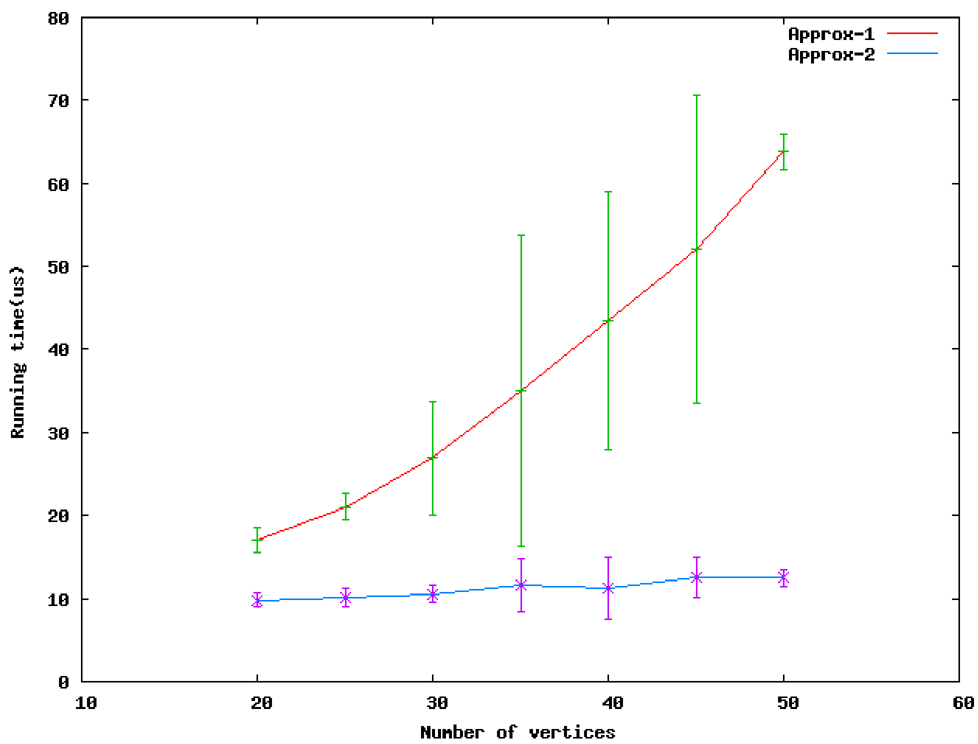

FIG. 3 Running Time For Approx-1 and Approx-2 (from 20 to 50 vertexes)

The FIG. 2 and FIG. 3 compare the increasing trend of Approx-1 and Approx-2. The running time of Approx-1 is more than that of Approx-2. Approx-1 rises quickly and keeps increasing, while Approx-2 fluctuates between 8μs to 9μs before 16 vertexes and increases slightly after 20 vertexes.

Compare with Approx-2, Approx-1 takes more time on picking vertex of highest degree. The following steps like throwing away all edges attached and repeating till no edge remain have similar running time. So Approx-1 takes more time to get result.

Even though Approx-2 fluctuates before 16 vertexes, the running time of 16 vertexes is more than that of 4 vertexes, which means that Approx-2 could have increasing trend if the sample is large enough.

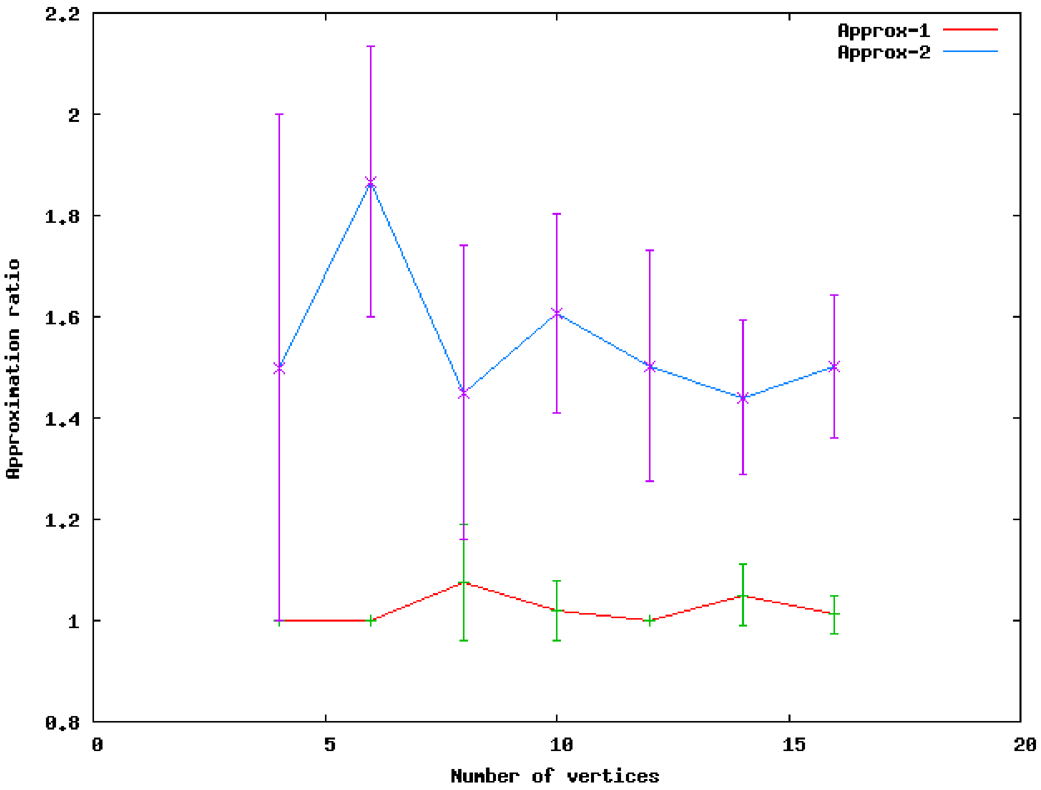

FIG. 4 Approximation Ratio For Approx-1 and Approx-2

Given the FIG. 4 graph above, approximation ratio of Approx-1 is smaller than that of Approx-2. It shows that the size of computed vertex cover for Approx-1 is smaller. 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.

Conclusion

According to the run time and approximate ratio, three algorithms can have good performance depend on which kind of graph is .So they can be effective to solve specified vertex cover problem.

In terms of run time, Approx-2 is better, but it fails to get minimum vertex cover. So it is a 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, for example the number of vertex in graph is under 16.

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.