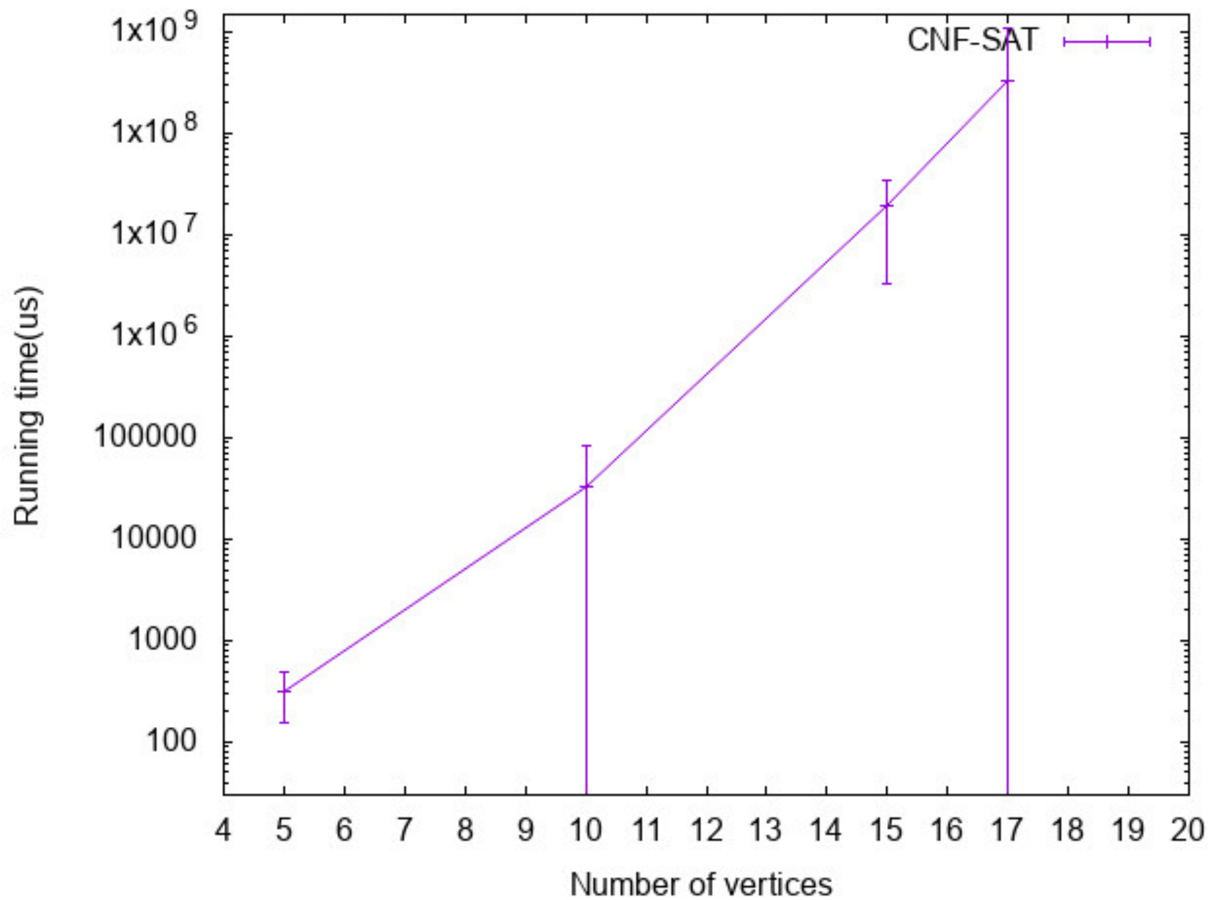
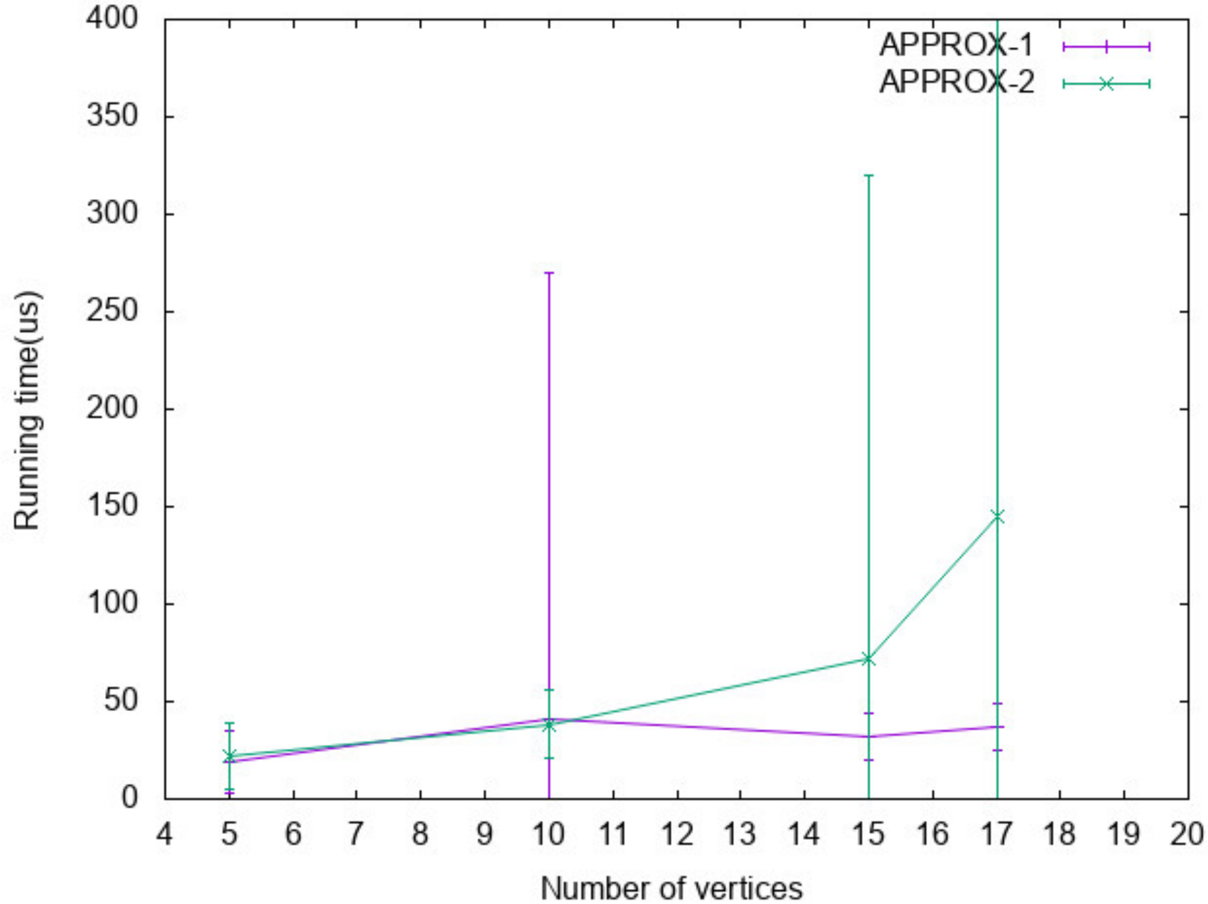
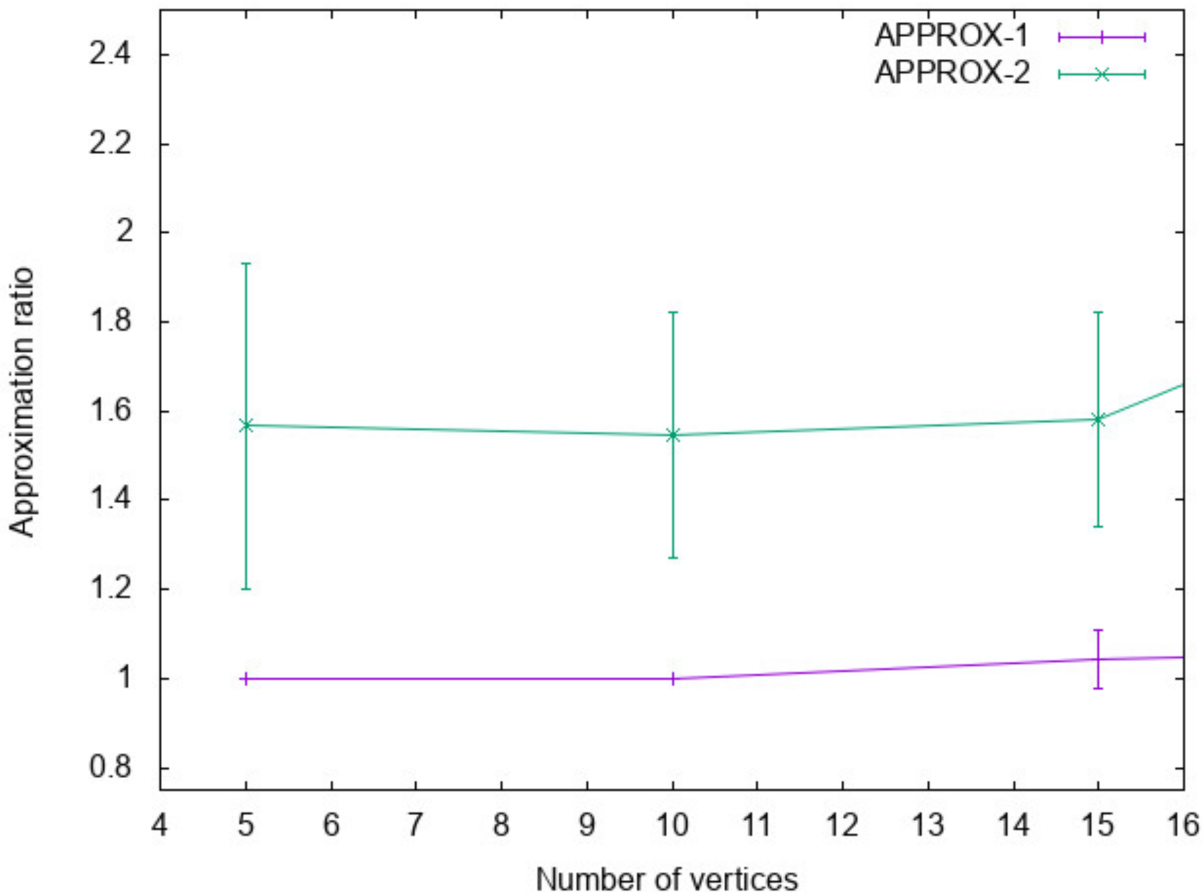


This picture shows the running time of CNF-SAT, Y axis is the running time (us), X axis is the number of vertices, each value for V each value was generate at least 10 graphs, compute the time and approximation ratio for each such graph. The running time for at least 10 runs of each such graph was measured. Then, the mean (average) and standard deviation across those 100 runs for each value of V were computed. This approach is real time consuming, and as the number of V growing, the running time increased exponentially, this is because that this approach is a polynomial-time algorithm.



This picture shows the running time of APPROX-1 and APPROX-2, Y axis is the running time (us), X axis is the number of vertices, each value for V each value was generate at least 10 graphs, compute the time and approximation ratio for each such graph. The running time for at least 10 runs of each such graph was measured. Then, the mean (average) and standard deviation across those 100 runs for each value of V were computed. The APPROX-1 and APPROX-2 approach do not show any significant difference. Comparing with CNF-SAT approach, these two approaches are significantly cost less running time, this is because that this approach is a linear algorithm.

.



This picture shows the approximation ratio of APPROX-1 and APPROX-2 over the output of CNF-SAT approach. Y axis is the approximation ratio, X axis is the number of vertices, each value for V each value was generate at least 10 graphs, compute the time and approximation ratio for each such graph. The approximation ratio for at least 10 runs of each such graph was measured. Then, the mean (average) and standard deviation across those 100 runs for each value of V were computed. The approximation ratio was characterized as the ratio of the size of the computed vertex cover to the size of an optimal (minimum-sized) vertex cover. This chart demonstrates that APPROX-1 and APPROX-2 have similar accuracy with CNF-SAT approach. Additionally, APPROX-1 works better than APPROX-2.

In conclusion, APPROX-1 algorithm becomes more efficient than APPROX-2, based on the conception of efficient, which is characterized in one of two ways: (1) running time, and (2) approximation ratio. Nevertheless, the accurate of APPROX-1 algorithm increased with the increasing number of vertice. So, when V<15, it will be better to use CNF-SAT approach, whereas when V>15 APPROX-1 algorithm is a good option.