ECE650 - METHODS AND TOOLS FOR SOFTWARE ENGINEERING

UNIVERSITY OF WATERLOO

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Project: Analysis of Minimum Vertex Cover Algorithms

Authors:
Bowen Deng (ID: 20986098)

Date: April 10, 2022

1 Introduction

Minimum vertex cover problem is a classic optimal problem. It is aiming to look for the least number of vertices that edges of incident on those vertices cover the whole graph. The application of the vertex cover problem includes documents summarization [1], computational biology [2], wireless sensor networks [3] and so on. In this project, minimum vertex cover problem will be attempted to minimize the number of cameras but still monitor the traffic situation efficiently.

An undirected graph $G = \{E, V\}$ contains two features, E is all edges of G and V is all vertices of G. vertex cover is the subset of vertices $S \subseteq V$ which covers the whole graph G. In the mathematical way, for every $e \in E$ and e is composed by $(u, v) \in V$, at least one of $u \in S$ and $v \in S$ is true. The minimum vertex cover is trying to find a S with the minimum number of vertices.

In this project, three algorithms are discussed to solve the minimum vertex cover problem. The first kind of algorithms (CNF-SAT-VC) is to get the encoding of conjunctive normal form (CNF) clauses to convert the minimum vertex cover problem to satisfiability (SAT) problem. The open-source SAT solver Minisat is used to implement this algorithm. The second kind of algorithms is approximation algorithm, which tries to approximate to the optimal solution. Two algorithms(APPROX-VC-1 and APPROX-VC-2).

2 Algorithms

2.1 CNF-SAT-VC

CNF-SAT-VC algorithm is aiming to present a polynomial-time reduction from vertex cover (VC) problem to conjunctive normal form satifiability (CNF-SAT) problem. Considering we have a graph G and vertex cover set K. The CNF of reduction contains clauses meeting the following conditions:
1. No vertex can appear twice in K.
2. At least one vertex in G will appear in K.
3. No more than one vertex in G appears at the same position in VC.
4. Every edge of G is incident to as least one vertex in VC.

2.2 APPROX-VC-1

The first approximation algorithm is APPROX-VC-1. The pseudo code of APPROX-VC-1 is shown as Algorithm 1.

Algorithm 1 APPROX-VC-1

```
Input: vertices of the graph V, edges of the graph E
Output: vertex cover set res_VC
while len(E) ≠ 0 do
    for v in V do
        for e in E do
            if v in e then
                degree of the vertex + 1
            end if
        end for
    end for
    get the vertex with highest degree V_h
    add V_h to the output res_VC
    for e in E do
        if V_h in e then
            remove e from E
        end if
    end for
end while
return res_VC
```

2.3 APPROX-VC-2

The first approximation algorithm is APPROX-VC-2. The pseudo code of APPROX-VC-1 is shown as Algorithm 2.

Algorithm 2 APPROX-VC-2

```
Input: vertices of the graph V, edges of the graph E
Output: vertex cover set res_VC
while len(E) ≠ 0 do
    Pick an e in E
    get two vertex of e called v_e1 and v_e2
    add v_e1 and v_e2to the output res_VC
    for e in E do
        if v_e1 or v_e2 in e then
            remove e from E
        end if
    end for
```

end whilereturn res_{VC}

3 Analysis

3.1 Evaluation

Run time and approximation ratio are used to evaluate the performance of the algorithms.

1) Runtime
Run time is defined as the time required for an algorithm to obtain the vertex cover.

2) Approximation ratio
In this project, approximation ratio is defined as the ratio of the size of the computed vertex cover to the size of an optimal vertex cover.

3.2 Runtime Analysis

As shown in Figure 1 and Figure 2, I generate graphs for $|V| \in [5, 50]$, in increments of 5 and plot the related runtime of three algorithms. The x coordinate represents the number of vertices and the y coordinate represents runtime (ms). the point in the figure stands for mean time, and the error bar stands for the standard deviation of runtime. In Figure 1, runtime of CNF-SAT-VC can be seen from 5 vertices to 15 vertices, in increments of 5. The reason why there is only runtime of 5, 10, 15 vertices is that when the number of vertices bigger than 15, the algorithm spends a lot of time calculating the output. And the timeout method in the project will stop the thread to print a timeout result so that there will be no actual runtime with those numbers of vertices. It can be seen that as the number of vertices increases, the mean runtime and standard deviation also increase rapidly, which finally climb to around 426 ms and 221.

In Figure 2, runtime of APPROX-VC-1 and APPROX-VC-2 can be seen from 5 vertices to 50 vertices, in increments of 5. In general, runtime of APPROX-VC-1 is far greater than that of APPROX-VC-2. And it can be seen that as the number of vertices increases, the mean runtime and standard deviation also increase rapidly, which finally climb to around 1297 μs and 609.81. But for APPROX-VS-2, as the number of vertices increases, the mean runtime increases slightly, which finally climb to around 102.8 μs. And the standard deviation keep stable from 30 to 35.

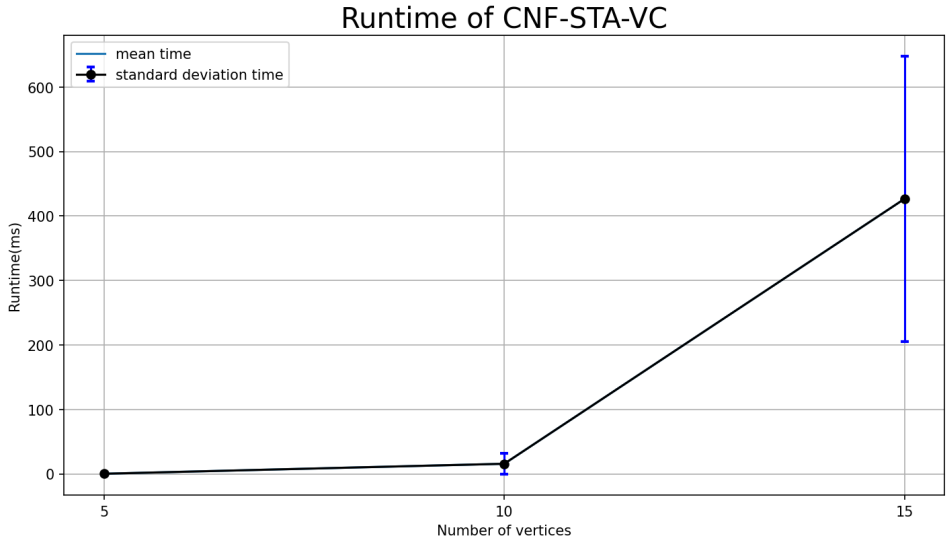

Figure 1: Runtime of CNF-SAT-VC

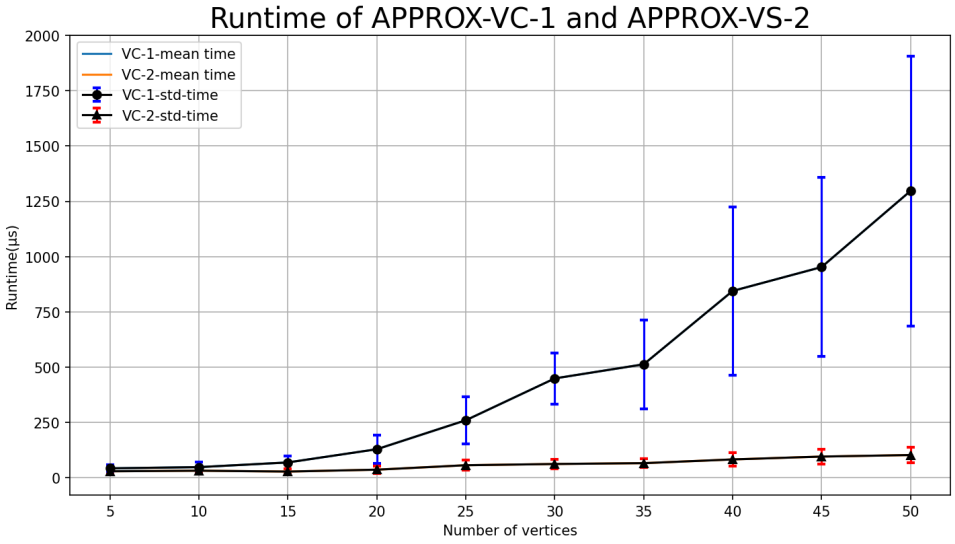

Figure 2: Runtime of APPROX-VC-1 and APPROX-VC-2

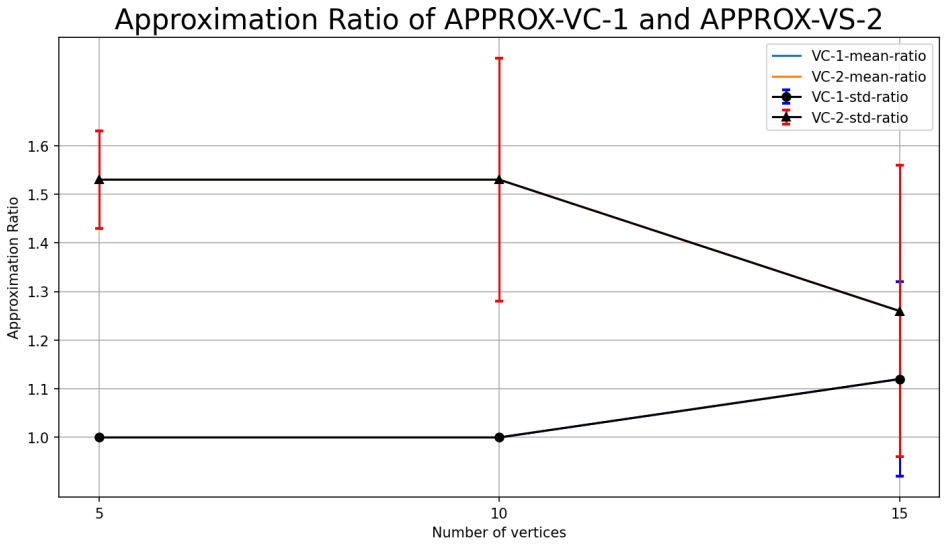

Figure 3: Approximation Ratio of APPROX-VC-1 and APPROX-VC-2

3.3 Approximation Ratio Analysis

In Figure 3, the approximation ratios are plot with 2 algorithms from 5 to 15, in increments of 5 because we can only get result of CNF-SAT-VC algorithm from 5 to 15. For APPROX-VC-1, the approximation ratio is close to 1 in general, which means APPROX-VC-1 successfully approximate to the optimal algorithm. And for APPROX-VC-2, The ratio is always bigger than 1.3, which means APPROX-VC-2 does not approximate the optimal algorithm as well as APPROX-VC-1.

4 Conclusion

For runtime, the CNF-SAT-VC algorithm is at the ms level, which is much larger than the APPROX-VC-1 algorithm and the APPROX-VC-2 algorithm and is relatively unstable. The runtime of APPROX-VC-2 algorithm is also significantly higher than that of APPROX-VC-1 algorithm. For the approximation ratio, APPROX-VC-1 has a lower approximation ratio than APPROX-VC-2, which means APPROX-VC-1 is closer to the optimal algorithm.

References

[1] A. John and M. Wilscy, "Vertex cover algorithm based multi-document summarization using information content of sentences," **Procedia Computer Science**, vol. 46, pp. 285–291, 12 2015.

[2] Y.-C. Chen, W.-C. Peng, and S.-Y. Lee, "Efficient algorithms for influence maximization in social networks," **Knowledge and Information Systems**, vol. 33, pp. 577–601, 12 2012.

[3] V. Kavalci, A. Ural, and O. Dagdeviren, "Distributed vertex cover algorithms for wireless sensor networks," **International journal of Computer Networks & Communications**, vol. 6, 02 2014.