Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 4, April 2015, pg.472 – 475

RESEARCH ARTICLE

Several Algorithms to Solve Vertex Cover Problem

Soumya Godi[1]
[1]Department of Computer Science and Engineering, NHCE, Bangalore, India
[1] soumya.godi@gmail.com

Mrs.Tinu[2]
[2] Department of Computer Science and Engineering, NHCE, Bangalore, India
[2] tinuns@gmail.com

Abstract— The vertex cover problem belongs to Graph Theory concepts, is the class of NP-complete graph theoretical problems. Vertex cover problem has a numerous real life applications such as communications, bioinformatics, engineering, statics, and mathematics to study and process biological data. NP complete problems are unlikely to find a polynomial-time algorithm for solving vertex-cover problem accurately because they give only exponential time complexity. Vertex-cover problem is involves in two phase transitions i.e., Coverable and Uncoverable. This research work examines that several algorithms are being used to find minimum vertex cover set for different graphs. The comparison of approximation algorithm, greedy algorithm and alom's algorithm with respect to time complexity of algorithms and also with output solutions. By this comparison we realize that alom's gives the better solution than the other two mentioned algorithms.

Keywords— minimum vertex cover, biological data, approximation, clever-greedy, Alom's algorithm

I. INTRODUCTION

A vertex cover of an undirected graph G= (V, E) where V is a set of vertices and E is a set of edges. A vertex cover of an undirected graph is a subset V' ⊆ V such that if (u, v) is an edge of G, Then either u ∈ V' or v ∈ V' or both. Vertex cover problem is a one of the NP-Complete [2] problem and also the part of the Karp's 21 problems [3]. As it is a NP-Complete problem is unlikely to find solution for a problem. But we do have a hope to get an optimal solution for a problem. Vertex cover problem is having 2-versions.1.decision version 2.optimization version. Decision version which gives the prespecified size of vertex cover set of vertices. Optimization version gives the minimum vertex cover set .The problem in both NP-hard and NP problem is said to be NP-Complete problem. Vertex cover problem is to find minimum number of vertices from a graph and those minimum vertices should cover all the edges in a graph.

II. BACKGROUND RELATED

Among the several NP-complete classes of problems it is seen that vertex cover problem has many useful applications are in real world and a reduction in its exponential time complexity in finding optimal solution is very essential. There are several approximate algorithms to the solution of vertex cover problem. The real world problems in which the solution of vertex-cover problem can be applied are Wireless sensor network, airline communication network, terrorist communication network [5], network traffic measurement [6] and routing in a delay tolerant network.

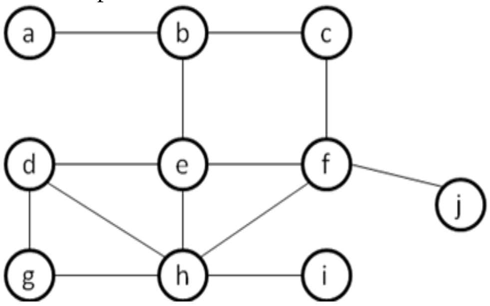
III. PROPOSED EVALUATION

A. Approximation Algorithm

There are 3-approaches to getting around NP-completeness.1. If actual inputs are small, an algorithm with exponential running time may be perfectly satisfactory. 2. We may be able to isolate important special cases that are solvable in polynomial time. 3. It may still possible to find near optimal solutions in polynomial time. An **approximation algorithm** [4] is the one where it returns the approx solution to the given problem, which is near to optimal solution but not exactly.

1. C ← Ø
2. E′ ← E [G]
3. While E′ ≠ Ø
4. Do let (u, v) be an arbitrary edge of E′
5. C ← C U {u, v}
6. Remove every edge in E′ incident on u or v
7. Return C

For a detailed explanation consider a Graph G as shown below



Graph G

According to algorithm from a graph G, we need to choose random vertex say, 'e'. Then the incident edges [(e, b), (e, f), (e, h), (e, d)] to vertex 'e' are being removed and keep the vertex 'e' to the vertex cover set C= {e}. So like this next vertex from a graph is being selected until it covers all the edges in the graph. Next we have chosen vertex 'h' then the incident edges (h, e), (h, f), (h, i), (h, g), (h, d) are removed and vertex added to vertex cover set C= {e, h}. Next vertex 'c', vertex 'a', vertex 'f', vertex 'd' are chosen to cover all the edges from a graph. So finally we are getting vertex cover set as follows C= {e, h, c, a, f, d}.

B. Greedy Algorithm

This is a simple method used to solve optimization problems. The problems that are solved using the greedy method include finding the best order to execute a certain set of jobs on a computer, finding the shortest path in a graph, etc.

Greedy algorithm moves slowly at each step. Rather than getting globally optimal solution to the problem, its focus is to find best solution. Local optimal choice is taken with the goal of achieving global optimal solution. Greedy algorithm never corrects its opted choice for the vertex-cover problem.

Input: Labeled Graph L (G) = G (L (V), E)

Output: S of L (G) where S is vertex cover set of L (G).

1. S←0
2. While (E ≠ ¢)
3. Select any edge e ϵ E and choose an end-point v of e
4. S←SU {v}
5. E' ←E //{e ϵ E: v ϵ e}
6. Return S

To find out the vertex cover by this algorithm first we need to keep vertex cover set as empty. We need to select one edge from a graph randomly. Then the incident vertices are been observed to check which vertex is having highest degree. Vertex with high degree will be added to vertex cover set and all the incident edges to that vertex are removed. This procedure is to be repeated until all the edges are to be covered.

For better understanding of this algorithm we have considered a graph G. According to the algorithm we need to select any edge randomly from a graph. So here we have chosen edge (d, e) then we need to check out which vertex is having high degree among selected edge. Here 'h' is having high degree than the 'e'. So delete the all other edges from a vertex 'e' except the selected edge. Later on 'e' should be added to vertex cover set C= {e}. Next step, consider one more edge randomly from graph. Here we have selected edge (h, i), vertex 'h' is having high degree than the vertex ‘i’. So remove all other incident edges to vertex ‘h’, add this vertex ‘h’ to vertex cover set C= {e, h}. Next edge (a, b) is chosen. Vertex ‘b’ is having high degree than the vertex ‘a’. So remove the edges incident to vertex ‘b’ and add vertex to vertex cover set C= {e, h, b}. This procedure has to be continued until algorithm covers all the edges in the graph. Further consider one more edge (c, f) .So vertex ‘f’ is having high degree than the ‘c’. So remove incident edges to it. Then add vertex ‘f’ to vertex cover set C= {e, h, b, f}. Then for next step consider one more edge (d, g) from a graph both are having same degree so consider any one of them and add to vertex cover set by deleting the incident edges to either of the vertex chosen. Here we are considering the vertex‘d’. So finally vertex cover set for greedy algorithm is C= {e, h, b, f, d}. By comparing the vertex cover set from the approximation algorithm and greedy algorithm we are getting different number of vertices in the set.

C. Alom’s Algorithm:

An improved vertex cover algorithm has been proposed by Monjurul Alom. Alom’s algorithm gives better optimal solutions other than the Approximation and greedy algorithms in this thesis.

Input: Labeled Graph L (G) = G (L (V), E)

Output: S of L (G) where S is vertex cover set of L (G).

1. ALOM_ VERTEX_COVER (V, E)
2. S ←0
3. E'←E [G]
4. While (E ≠ ¢) {
5. V' ←select vertex with highest degree;
6. If (vertex with highest degree are two or more) then
7. V' ←Select vertex with at least one edge not covered by others that have highest number of edges.
8. S ←S U V'
9. Remove all edges incident to V';
10. Count degree of each vertex of modified graph}
11. Return S

Alom’s algorithm chooses the vertex with the highest degree of incident edges and removes the all the incident edges to that vertex. If more than one vertex is having same degree then algorithm selects the vertex with at least one edge not covered by others which are having same degree. These procedures to be continued until all the edges are to be covered.

Let’s consider graph ‘G’. Initially the vertex cover set is empty. Select the vertex from a graph which is having high degree. Here vertex ‘h’ is having high degree among all other vertices. So add this vertex to vertex cover set by removing all incident edges to that vertex, C= {h}. Next consider a vertex which is having updated high degree after the first iteration step. Here vertex ‘f’, ‘e’ and ‘b’ are having the same high degree, in this case we need to select the vertex in such a way that vertex should not be covered at least one edge among other vertices are having high degree. So in our example edge (e,b) and (e,f) can be covered by the vertices which are having the same degree as vertex ‘e’, here no need of selecting vertex ‘e’. So select vertex ‘b’, remove incident vertices to it. Add vertex ‘b’ to vertex cover set C= {h, b}. Next update degree of each vertex, then select vertex ‘f’ which is having high degree add that to vertex cover set after removing all incident edges to it. So C= {h, b, f}. Then finally to cover all edges we need to select one more vertex from graph let’s say ‘d’ is having high degree , add ‘d’ to cover set by removing incident edge to it. Finally the vertex cover set for greedy algorithm is C= {h, b, f, d}.

Table I: comparative analysis of algorithms to find minimum vertex cover problem

Algorithm Name	**Complexity**	**Remarks**
Approximation Algorithm	O(V+E)	1. Gives the nearer optimal solution, but it gives every time different solutions when it run to more than once on same graph. 2. Solution yields twice the optimal solution; hence it is called as polynomial time 2-approximation algorithm.

Greedy algorithm	O(V+E)	1. Fails to give an optimal solution in some cases.
Alom's algorithm	O(V logV+E)	1. Gives better optimal solution. 2. Sometimes it doesn't give optimal solution for larger graph.

IV. CONCLUSION

The problem of finding minimal number of vertex that can cover all edges of the graph is known as vertex cover problem. So this gives the cost effective nature to the graph. After observing the all the three above mentioned algorithms we can come to a conclusion that Alom's algorithm performs better in terms of finding vertex cover set among other two approximations and greedy algorithm. Compare the vertex cover sets after each algorithm we can find that difference so easily. This comparison of vertex cover problem among different graph studied [7]. Proposed Alom's algorithm gives the better optimal solution for a problem.

REFERENCES

[1] "Vertex Cover Problem" [online]. Available: http://en.wikipedia.org/wiki/Vertex_cover.

[2] "Some simplified NP-Complete graph problems" Available:http://www.sciencedirect.com/science/article/pii/0304397576900591.

[3] "Karp's 21 NP-complete problems" [online]. Available:http://en.wikipedia.org/wiki/Karp%27s_21_NP-complete_problems.

[4] "Analysis of various algorithms to solve vertex cover problem", Sangeeta Bansal, Ajay Rana. IJITEE ISSN: 2278-3075, Volume-3, Issue-12, May 2014.

[5] "Network optimization algorithm" [Online]. Available:http://support.sas.comidocumentation/cdllen/ornoaugl66084/HTML/defaultlviewer.htm#ornoaug_ optnet_ examples01.htm; accessed Jan 2014.

[6] Zeng, Yongguo, Dezheng Wang, Wei Liu, and Ao Xiong. "An approximation algorithm for weak vertex cover problem in IP network traffic measurement." In Network Infrastructure and Digital Content, 2009. IC- NIDC 2009. IEEE International Conference on, pp. 182-186. IEEE, 2009.

[7] "The Vertex Cover Algorithm" [Online] Available: http://www.dharwadker. orglvertex_cover/.

