International Journal of Advanced Science and Technology
Vol.67 (2014), pp.71-78
http://dx.doi.org/10.14257/ijast.2014.67.07

AVSA, Modified Vertex Support Algorithm for Approximation of MVC

Imran khan, Israr Ahmad and Muzammil Khan

*Department of Computer and Software Technology,
University of Swat, KPK, Pakistan*

imran@uswat.edu.pk, israrahmad009@gmail.com, muzammilkhan86@uswat.edu.pk

Abstract

Minimum vertex cover is very important among the NP-optimization problems and got the attention of the researchers in the past decade. Approximation techniques are used to solve the NP problems to get either optimal or near optimal solutions in polynomial time. In this paper, a modified vertex support algorithm is proposed that make use of same data structure as that of VSA but with different vertex selection criteria. Proposed algorithm is called advance vertex support algorithm (AVSA), it takes into account all the neighbors of a vertex twice while selection as surrounding vertices are very important for the decision. AVSA is analyzed experimentally against all benchmarks and compared with some better algorithms already present. The experimental results show that the worst approximation ratio was 1.032 with a ratio of average approximation 1.006. A comparative analysis with other algorithms indicates that AVSA can provide the best solution among them the minimum time requirements. Simplicity in design can give advantages to implement in various fields of application with minimum overhead.

Keywords: *MDG (Maximum degree greedy), MVC (Minimum vertex cover), MIS (Maximum independent set), VSA (Vertex support algorithm), MWVC (Minimum weighted vertex cover), MWIS (Maximum weighted independent set)*

1. Introduction

Graphs are combination of nodes and edges where nodes are connected through edges. Many real life problems can be modeled using graphs and for modeling these problems on graphs Graph theory is used. After modeling with graphs these are then manipulated by several techniques to optimize specific objective according to area of application. Application areas of graph theory include Logistic networks, Social science, Electrical engineering and chemistry, *etc*. [1]. Problem with graph theory is that many problems are intractable (that cannot be solved in polynomial time), and so majority of graph related problems belong to a class named NP-Complete. It is widely believed that we can't solve NP-Complete problems optimally in polynomial time so there are various other techniques developed by researchers to handle this. These techniques are either heuristics or approximation of the optimal solution. Heuristic solutions have no guarantee of producing quality solution in reasonable time. On the other hand approximation techniques always produce approximation solution in polynomial time, but the quality of solution depends on the approximation ratio. Approximation ratio is the ratio of approximate solution to the optimal solution, i-e ρ i = A (i)/OPT (i) $\geq$ 1. When ρi =1 then the approximate solution is actual optimal solution, but this is not the case always because these problems are intractable and according to Gray & Johnson a problem as

ISSN: 2005-4238 IJAST

intractable if it is so hard that no polynomial algorithm can possibly solve it [2]. The value of ρi determines the quality of solution, the more it diverts from 1 the more solution degrades. Basically there are two different versions of graph related problems one is decision version and other one is optimization. In decision version a try is made to answer a question, like is there exist such a solution that the whole graph is covered with ‘k’ nodes, *etc.* In this version just a check is made to answer this, while in optimization version a search is made for the optimal solution as much possible. Here we will deal with optimization version to find out the most optimal solution possible for a problem.

Vertex cover is one of the graph related problem where the objective is to extract a set of vertices in a specific graph which cover all the edges of the graph. Minimal vertex cover is just the same but here another objective is to optimize the solution such that the total vertices in the vertex cover set remains as minimum as possible, Let a Graph G (V, E) where ‘V’ represent vertices and ‘E’ represent edges then minimal vertex cover vertices are such that Vc ⊆ V which cover all the edges in the graph. In 1972, Richard Karp showed that finding the solution of minimal vertex cover in a graph is an NP-complete problem [3]. Vertex cover remains NP-complete even in cubic graphs [4] and even in planar graphs of degree at most 3 [5]. Li *et al.* argued that current heuristic algorithms of MVC only consider vertex features in isolation in order to decide whether a vertex is in or not in the solution set [6]. Independent set problem also belongs to NP complete and these both problems can be easily solved with just one solution. Independent set is a set of nodes which share no each with each other and in maximum independent set search is made to maximize total number of independent nodes as much as possible. Practically the nodes other than MVC are MIS nodes and vice versa.

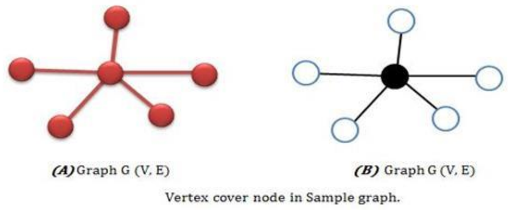

Figure 1. Sample graph representing vertex cover, (B) represent same graph as (A) but with center vertex cover node

Graph shown in figure1 demonstrates nodes which are in MVC and nodes which are in MIS, ‘A’ part of Figure 1 is a simple graph with nodes connected through edges while ‘B’ part represents processed form, node with black color represents MVC which cover all the edges of graph and this is optimal solution because it is impossible to solve is with less number of nodes than 1, nodes other than this are MIS nodes and it is clearly evident from Figure 1 no such relationship present among these.

This paper is organized as in Section 2 we have presented literature review, in Section 3 demonstrates proposed algorithm, Section 4 contains empirical results with detailed analysis and Section 5 constitutes conclusion and future work.

2. Literature review

Richard Karp showed that Minimum vertex cover is NP-Complete [3]. It is widely believed that finding optimal solution to these problems is impossible in polynomial time. Clarkson proposed a greedy approach for MVC which makes selection on the basis of maximum degree node; this approach was named MDG [7]. Run time complexity of this approach presented is in O (E^2) where 'E' is total number of edges in a graph. Its worst case approximation ratio is 'Δ' which is maximum degree in the graph. Delbot and Laforect experimentally analysed these approaches and among those MDG gives max of 33% error on ERDOS RENYI graphs, 9% on trees, 44% on BHOSLIB, 32% on regular graphs and 70% on average worst case graphs [8].

The key point in solving graphs for MVC is that it happens most of the time when we a select a node with maximum degree, it compels us to select extra nodes for covering all edges of graphs and affects the final outcome. Another greedy approach was presented by Chavatal which select a node with minimum degree [9]. This was originally presented for approximation of MIS but as MIS is also NP-Complete so MIS and MVC are reducible to each other, means we can solve these both problems on a single algorithm and practically it is simple because nodes other than MVC are MIS nodes [10]. Its run time complexity is in O (E^2) [9]. It is mentioned by Halldarson and Radhakrishnan that GIC can find optimal solution in trees and therefore in paths [11].

List left was devised an approach which works on sorting all nodes in a list and then processes it from left to right and showed that its worst case approximation ratio is and minimum approximation ratio of [12]. Experimental results presented by Delbot and Laforest shows that List left can't provide better or even same results compared to other algorithms implemented for analysis [8].

Delbot and Laforest Presented the same approach named List right with change in order of processing list, they process list from right to left [13]. Its maximum error percentage never exceeds 55 and provides better results than list left. Balaji et al devised a new approach with new data structure named support of a vertex [14]. All decisions regarding vertices are made on the basis of this value. Support of vertex that they proposed is the sum of degrees of all vertices adjacent to a vertex. They have tested their approach on large number of benchmarks and are optimal in most of the cases and its rum time complexity is O (EV^2). Li *et al.* employed greedy approach in a different way names share of a vertex, where share of vertex is the total number of vertices it shares6. MVC node selection is made on the basis of this value but this approach not seems to be efficient on large graphs because of their complex data structure and calculations. A new clever intelligent greedy approach is presented by Gajurel and Bielefeld named NOVAC-1[15]. This approach works on a clever concept raised from the keen observation and analysis of relationship among vertices. The vertices attached to minimum degree nodes are candidate of MVC with high probability and they deployed this concept. Result shows that it provide optimal results on 35% of benchmark graphs tested and approximation ratio never exceeds 1.077 with an average approximation ratio of 1.008 [15].

3. Proposed algorithm

The Algorithm proposed in this paper is named Advanced Vertex Support Algorithm named after Vertex Support Algorithm because the data structure proposed in [14]. Support of a vertex is the sum of degree of all vertices adjacent to it. The algorithm idea was reflected from very important0 observation of graphs when they are processed for the objective of MVC. Actually those nodes which are attached to minimum degree nodes are the candidates of MVC with very high probability and this fact is exploited in AVSA to produce as much optimum solution as possible. The VSA algorithm presented in [14] often fails on those graphs where MDG fails and this is due to the similarities of selection, MDG makes selection on the basis of maximum degree node in a graph while VSA selects vertex with maximum support value. It is possible most of the time that vertex with maximum degree have maximum support value too.

It is most probable that vertices adjacent to minimum support value are candidate for MVC because it can help in keeping total number of vertices in MVC as much minimum as possible. This fact if employed can give better results, so we exploit this fact by presenting a new approach using support value which works on calculating degree of each vertex followed by calculation of support value for each vertex. Search is then made for minimum support value and all vertices with this minimum support value are added to 'min_support' set, then all neighbor nodes to min_support set are searched and among those searched nodes vertex with maximum support value is selected as candidate for MVC. The approach seems to be amazing to work for MVC because it is selecting minimum value again and again but interestingly this leads to optimal results most of the time.

The pseudo code for the proposed AVSA algorithm is given below.

Algorithm for Minimum Vertex Cover

```
AVSA (Graph G)
    {
MVC [],
D [],
S [],
min_support [],
While (Edges ≠ ∅)
    {
For each v   V calculate degree of v
    // calculate degree for each node in the graph.
For each v   V
Calculate support value for each v that is S (V) =$\sum_{u\in N}(v)d(u)$.
Find out all nodes with minimum support value and add it to 'min_support'
Find out neighbors to all nodes in 'min_support'.
Find node with maximum support value in neighbors of 'min_support'.
Add it to MVC, Drop all its edges.
    }
}
```

There is no extra complex computation involved and decisions are made straight forward. Run time complexity of the proposed algorithm is in O (EV log v).

3.1. Empirical results

This section constitutes detailed explanation of experimental results recorded when AVSA was applied to a large number of best available benchmarks. All implementation process was carried out in Mat lab on core 2 duo system running windows 7. Detailed comparative analysis of AVSA with MDG and NOVAC-I has been carried out to check the effectiveness relatively. Apart from all the benchmarks tested AVSA was able to provide quality results with minimum approximation ratio, worst case approximation ratio recorded was '1.032' and an average approximation ratio of '1.006'. Table 1 outlines results of various benchmarks which were used for experimental analysis. First column shows benchmarks, second column demonstrates total number of nodes, third column contains optimal results, fourth is for results obtained through AVSA, fifth is for MDG results, sixth is for NOVAC-1 results, seventh shows approximation ratio achieved through AVSA, eighth show approximation ratio of MDG and ninth last one shows approximation ratio of NOVAC-1.Approximation ratios from Table 1 advocate that AVSA is able to produce best results among those tested.

Table 1. Approximation results of AVSA, MDG and NOVAC-1 on various benchmarks

Benchmarks	Total Vertices	Optimal MVC	AVSA	MDG	NOVAC-1	Approximation Ratio		
						AVSA	MDG	NOVAC-1
cfat200_1	200	188	188	188	188	1.000	1.000	1.000
cfat200_2	200	176	176	176	176	1.000	1.000	1.000
cfat200_5	200	142	142	142	142	1.000	1.000	1.000
cfat500_1	500	486	486	486	486	1.000	1.000	1.000
cfat500_2	500	474	474	474	474	1.000	1.000	1.000
cfat500_5	500	436	436	436	436	1.000	1.000	1.000
mann_a27	378	252	254	261	253	1.008	1.036	1.004
graph50_6	50	38	38	38	38	1.000	1.000	1.000
graph100_1	100	60	60	60	60	1.000	1.000	1.000
graph100_10	100	70	70	70	70	1.000	1.000	1.000
graph200_5	200	150	150	150	150	1.000	1.000	1.000
graph500_1	500	350	350	350	350	1.000	1.000	1.000
graph500_2	500	400	400	400	400	1.000	1.000	1.000
graph500_5	500	290	290	290	290	1.000	1.000	1.000
Hamming6_2	64	32	32	32	32	1.000	1.000	1.000
Hamming6_4	64	60	60	60	60	1.000	1.000	1.000
Hamming8_2	256	128	128	128	128	1.000	1.000	1.000
Hamming8_4	256	240	244	248	240	1.017	1.033	1.000
Hamming10_2	1024	512	512	512	512	1.000	1.000	1.000
Hamming10_4	1024	984	1004	1015	1004	1.020	1.032	1.020
phat300_1	300	292	294	293	293	1.007	1.003	1.003
phat300_2	300	275	276	278	275	1.004	1.011	1.000
phat300_3	300	264	272	269	266	1.030	1.019	1.008
phat700_1	700	689	692	693	692	1.004	1.006	1.004
phat700_2	700	656	660	660	657	1.006	1.006	1.002
phat700_3	700	638	649	642	641	1.017	1.006	1.005
jhonson8_2_4	28	24	24	24	24	1.000	1.000	1.000
jhonson8_4_4	70	56	56	62	56	1.000	1.107	1.000
jhonson16_2_4	120	112	112	112	112	1.000	1.000	1.000
jhonson32_2_4	496	480	480	480	480	1.000	1.000	1.000
sanr200-0.7	200	182	185	184	185	1.016	1.011	1.016
sanr200-0.9	200	158	163	164	159	1.032	1.038	1.006
sanr400_0.5	400	387	389	392	388	1.005	1.013	1.003
sanr400_0.7	400	379	381	384	381	1.005	1.013	1.005

fbr_30_15_5	450	420	424	429	424	1.010	1.021	1.010
fbr_35_17_2	595	560	565	570	565	1.009	1.018	1.009
c 125	125	91	91	93	92	1.000	1.022	1.011
c 250	250	206	211	211	211	1.024	1.024	1.024
c500.9	500	≤443	451	453	449	1.018	1.023	1.014
broc200_2	200	188	191	192	190	1.016	1.021	1.011
broc200_4	200	183	188	188	192	1.027	1.027	1.049
dsjc500	500	487	488	491	488	1.002	1.008	1.002
keller4	171	160	160	164	164	1.000	1.025	1.025
keller5	776	749	752	764	761	1.004	1.020	1.016

Worst approximation ratio of AVSA is ‘1.032’ and that of MDG and NOVAC-1 is ‘1.107’ and ‘1.049’ respectively. These ratios advocate the effectiveness of AVSA in producing quality solutions as compared to others, not only the worst case ratio is better but average approximation ratio of AVSA is acceptable among others. Average approximation ratio of AVSA is ‘1.006’ while MDG and NOVAC-1 has ’1.012’ and ‘1.006’ respectively. Figure 2 demonstrates the approximation ratio comparisons of AVSA, MDG and NOVAC-1; straight line shows the optimal results while variation from straight curve demonstrates diversion from optimal solution.

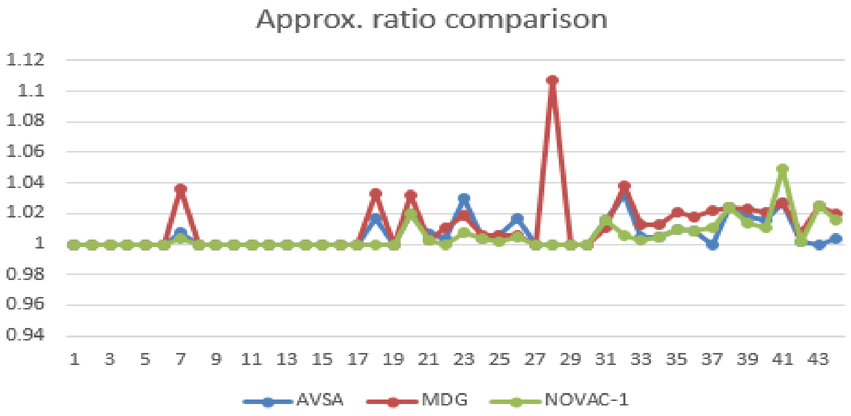

Figure 2. AVSA, NOVAC-1 and MDG approximation ratio comparison

4. Conclusion

A modified form of VSA is proposed which can take decision regarding vertices in very simple way; experimental results show that AVSA can produce optimal results with high probability. In case of failure in providing optimal solutions the approximation ratio never increases 1.032 which is very low as worst case. The importance of neighbors of a vertex is fully utilized here by making a two way decision process. In future we are planning to improve AVSA in order to provide as much optimal solutions as possible. We are also planning to extend our work to weighted graphs where MWVC is vertex cover of minimum weights.

Acknowledgements

I am thankful to Dr. Naveed Riaz Ansari who guide me how to do a research and become a good researcher, I am also grateful to Hasham khan who help in solving and understanding vertex cover using multiple approaches.

References

[1] J. Chen, X. Huang, I. Kanj and G. Xia, "Linear FPT reductions and computational lower bound", Proceedings of 36th ACM symposium on Theory of computing, Berlin Heidelberg, **(2004)**.

[2] M. Garey and D. Johnson, "Computers and intractability", Freeman, New York, **(1979).**

[3] R. Karp, "Reducibility among combinatorial problems", Plenum Press, New York, **(1972).**

[4] E. Demaine, "Sub exponential parameterized algorithms on bounded-genus graphs and H-minor-free graphs", Journal of the ACM (JACM), vol. 6, no. 52, **(2005).**

[5] I. Dinur and S. Safra, "On the hardness of approximating minimum vertex cover", Annals of Mathematics, vol. 162, **(2005).**

[6] S. Li, "An Approximation Algorithm for Minimum Vertex Cover on General Graphs", in Proceedings of 3rd International Symposium: Electronic Commerce and Security Workshops (ISECS '10), China, **(2010).**

[7] K. Clarkson, "A modification to the greedy algorithm for vertex cover problem", Information Processing Letters, vol. 16, **(1983).**

[8] F. Delbot and C. Laforest, "Analytical and experimental comparison of six algorithms for the vertex cover problem", Journal of Experimental Algorithmics, vol. 15, **(2010).**

[9] V. Chvatal, "A Greedy Heuristic for the Set-Covering Problem", Mathematics of Operations Research, vol. 4, **(1979).**

[10] T. H. Cormen, C. E Lieserson, R. L Rivest and C. Stein, "Introduction to Algorithms", 3rd Ed, MIT Press England, **(2009).**

[11] M. Halldorsson and J. Radhakrishnan, "Greed is good: Approximating independent sets in sparse and bounded-degree graphs", In Proceedings of 26th Annual ACM Symposium on Theory of Computing, New York: ACM, **(1994).**

[12] D. Avis and T. Imamura, "A List Heuristic for Vertex Cover", Operations research letters, vol. 35, **(2007).**

[13] F. Delbot and C. Laforest, "A better list heuristic for vertex covers, Information Processing Letters, vol. 107, **(2008).**

[14] S. Balaji, V. Swaminathan and K. Kannan, "Optimization of Un-weighted Minimum Vertex Cover", World Academy of Science, Engineering and technology, vol. 67, **(2010).**

[15] S. Gajurel and R. Bielefeld, "A Simple NOVCA: Near Optimal Vertex Cover Algorithm", Procedia Computer Science, vol. 9, **(2012).**

Authors

Imran Khan

I am lecturer in Computer science with Department of C and ST, University of Swat. I am working on this topic since last 3 years and have produced multiple research papers. This work is our collaboration with my friends.

Israr Ahmad

I am lecturer in Computer science with Department of C and ST, University of Swat. This is my first experience to work with NP-Complete problems and to develop a new algorithm. This will act as a milestone in my research work because future work is based on the above mentioned algorithm.

Muzammil Khan

I am lecturer in Computer science with Department of C and ST, University of Swat. I am working on this topic since last 6 months with my colleagues. We are working in group in the field of theoretical computer sciences to produce a new way to deal to NP problems. This work is our initial step and in future we are planning a general model for these problems.