International Journal of Hybrid Information Technology
Vol.7, No.5 (2014), pp.183-190
http://dx.doi.org/10.14257/ijhit.2014.7.5.17

Degree Contribution Algorithm for Approximation of MVC

Imran khan*[1] and Hasham khan[2]

[1]Department of Computer and Software Technology, University of Swat, KPK, Pakistan
Zong Head Office, Islamabad Pakistan
imran@uswat.edu.pk, hasham_ibms@yahoo.com

Abstract

Approximation methods are best way to deal NP optimization problems and MVC is one of these. In this research paper we have presented a new extra fast approximation algorithm for solving MVC generally in all graphs. The proposed algorithm is named degree contribution algorithm (DCA), a new data structure proposed and employed in this algorithm, name degree contribution. It is first time in literature that such a sophisticated data structure for graphs is proposed which take account of whole graph for each node contribution value. All decisions regarding vertices are made on the basis of the proposed data structure. Effectiveness of DCA is shown by applying it to best available benchmarks and after large number of experiments worst approximation ratio recorded was 1.041 and an average approximation ratio was 1.005. These results show that algorithm can perform well in solving graphs faster as compared to other algorithms present.

Keywords: *MVC (Minimum vertex cover), MIS (Maximum independent sets), DCA (Degree Contribution Algorithm), VSA (Vertex Support Algorithm), DC (Degree Contribution), MWVC (Minimal Weighted Vertex Cover), MWIS (Maximal Weighted Independent Set)*

1. Introduction

A Graph G (V, E) is a combination of vertices and edges. Vertices or nodes are connected through edges. Many real life problems can be modeled using graphs and after modeling, these problems are manipulated by several techniques to optimize the specific objective of the area of the application. Application areas of MVC include wireless communications, civil, electrical engineering, multiple alignment of biochemistry, Bioinformatics *etc*. [1]. A problem with graph theory is that many problems are intractable *i.e.*, these cannot be solved in polynomial time and majority of the graph related problems belong to a class called NP-Complete. As it is widely believed that NP-Complete problems cannot be solved optimally in polynomial time, various alternative approaches have been considered by the researchers to solve these problems. These techniques are either based on some complex heuristics or approximation of the optimal actual solution. Heuristic solutions have no guarantee of producing a quality solution in reasonable amount of time in many cases. On the other hand, approximation techniques always produce an approximate solution in polynomial time. It is pertinent to mention that the quality of a solution depends on the approximation ratio. Approximation ratio is defined as the ratio of approximate solution to the actual optimal solution, $\rho_i = A(i)/OPT(i) \geq 1$, where 'i' is an instance of the problem, 'A' is the approximate solution and OPT is the optimal solution. ρ_i is the approximation ratio for a particular problem instance 'i' and $\rho_n = \mathrm{Max}_i \rho_i$ for all n, *i.e.*, ρ_n is the maximum value of all ρ_is When $\rho_i=1$ then the approximate solution is actual optimal solution, but this is not the case always because these problems are intractable and according to Garey and Jhonson a problem is

ISSN: 1738-9968 IJHIT

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 deviates from 1 the poorer is the solution.

Vertex cover is one of the graph related problem where the objective is to extract a set of vertices of a graph that covers all the edges of the graph. Minimal vertex cover is similar but here another objective is to optimize the solution such that the total vertices in the vertex cover set remain as minimal as possible.

In 1972, Richard Karp showed that finding the solution of minimal vertex cover in a graph is an NP-complete problem [3]. Thus, it is obvious that we can't get optimal solution to MVC till it is proved that P=NP. Due to the existence of wide range of real life problems that can be formulated as MVC, various approximation and heuristics techniques have been developed and deployed by researchers. 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].

MVC cannot be approximated within a factor of 1.36, unless P=NP [7]. Numerous techniques and approximation algorithms have been presented in literature like Greedy approach, list left, list right, vertex support algorithm etc., but all of these have limitations in one way or another. Some are reliable but complex. Some are simple but underperform when we take computational complexity into account. Some are simple and fast but not reliable *i.e.*, approximation solutions are poor. Some have a factor of 2-approximation while some are Δ-approximation where Δ is variable. Generally, 2-approximation algorithms are considered acceptable.

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. Chavatal proposed a simplest approximation algorithm for MVC which select a vertex randomly to be in MVC set, all adjacent edges are deleted and the process continues till no edge remains [7]. This was not a good approach because selection of node for MVC needs quite intelligent guess not a random guess. Clarkson modified this random approach and random selection was changed with selection made on the basis of degree [8]. The vertex with maximum degree is selected for MVC which gives better results than random guess. This approach was named MDG [8]. Run time complexity of this approach presented is in O (E^2) where 'E' is total number of edges in a graph8. 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 [9].

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 [7]. 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) [7]. 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 [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 [9].

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 shares [6]. 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 1.00816.

3. Proposed Algorithm

The way of processing graphs to achieve the desire objective is one of the most important parameter in getting success. Keeping in front this point we have created a new structure named “Degree Contribution” for graphs processing parameters which is very easy to create, manipulate and take decisions. The DC is not limited jus to a single node but we take care of complete graph to calculate values for each node. DC for a node is the sum of degree of that node and total number of nodes with that degree in graph. Algorithm proposed in this paper makes use of this new data structure which helps in making intelligent decisions about the fate of vertices. The proposed Degree Contribution decides for each node that how much it can be beneficial if selected for the MVC. In the proposed algorithm decisions are made on the basis of this value.

3.1. Working and Pseudo Code

The working of algorithm is made as much simpler as it can be; first degree of each node is calculated followed by calculation of degree contribution value for each vertex. Vertices having higher degree contribution value have higher probability to be in MVC set. For this a search is made for finding out all vertices having higher degree contribution value and a set is formed. Now these are filtered nodes for MVC but for making quality results this set is further processed and a node having higher degree is selected for the MVC, all its adjacent edges are deleted and the process continues till no edge remains. Pseudo code of the proposed algorithm is given below.

```
Algorithm (Graph G)
{
        MVC [],
DC [].
        While (All edges not covered)
                {
                For each v ∈ V calculate degree of v.
                Calculate Degree contribution value for each of vertex related with whole
                graph.
                Find out all nodes with same maximum degree contribution value.
                        Select node with maximum degree contribution value.
                Delete all its edges
                }
```

End
}

At each iteration degrees for the remaining nodes are calculated and this process s followed by the calculation of DC for each node. The run time complexity of algorithm is EV where 'E' is total number of edges in the graph and 'V' is total number of nodes in that graph.

4. Empirical Results

In this section we outline the effectiveness of algorithm by applying it against best benchmarks which can confuse these techniques. Coding was done in Mat lab and all these experiments were carried out on Core i3 System running windows 8. After a large number of experiments we noted that worst case approximation ratio obtained is '1.041' and an average approximation ratio recorded is '1.005'. These ratios show that this algorithm can outperform various other approaches present in literature. Table 1 demonstrates the results obtained, first column constitutes benchmarks tested, second column contains total number of nodes in each corresponding benchmarks, third column shows optimal result of each benchmark, fourth column shows the results obtained through DCA and last column shows approximation ratio obtained from the results of DCA and optimal results.

Table 1. Benchmark Results of DCA

Benchmarks	Nodes	Optimal	DCA	App. Ratio
graph50_6	50	38	38	1.000
graph50_10	50	35	35	1.000
graph100_1	100	60	60	1.000
graph100_10	100	70	70	1.000
graph200_5	200	150	150	1.000
graph500_1	500	350	350	1.000
graph500_2	500	400	400	1.000
graph500_5	500	290	290	1.000
phat300_1	300	292	294	1.007
phat700_1	700	689	694	1.007
jhonson8_2_4	28	24	25	1.042
jhonson16_2_4	120	112	114	1.018
jhonson32_2_4	496	480	483	1.006
broc200_2	200	188	192	1.021
keller4	171	160	161	1.006
Hamming6_2	64	32	32	1.000
Hamming6_4	64	60	60	1.000
Hamming8_2	256	128	128	1.000
Hamming8_4	256	240	248	1.033
Hamming10_2	1024	512	512	1.000
cfat200_1	200	188	188	1.000
cfat200_2	200	176	178	1.011
cfat200_5	200	142	142	1.000
cfat500_1	500	486	486	1.000
cfat500_2	500	474	474	1.000
cfat500_5	500	436	436	1.000

Figure 1 shows that algorithm is optimal on most of the benchmarks. Those graphs where DGA fails to provide optimal results, the approximation ratio there never exceeds a lot.

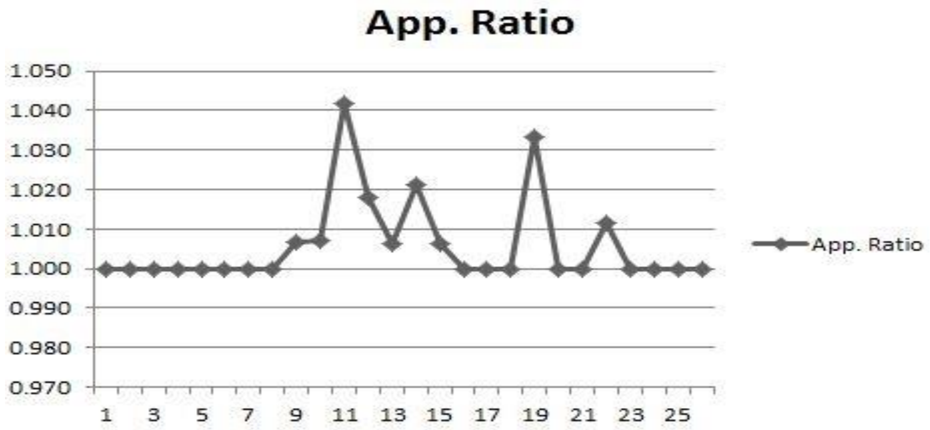

Figure 1. Approximation Results of Proposed Algorithm

The distinguishing feature of DCA is its faster response in solving graphs; even larger graphs can be solved in much lesser time. Figure 2 demonstrates this feature of algorithm and is captured by using date of Table 2 which outlines time taken in solving each benchmark. First column of Table 2 contains benchmarks, second column shows total number of nodes and third column constitutes time in seconds which was taken in solving the corresponding benchmark. Time taken May seems to be high but this is because implementation was carried out in Mat lab which performs slower as compared to C++ environment.

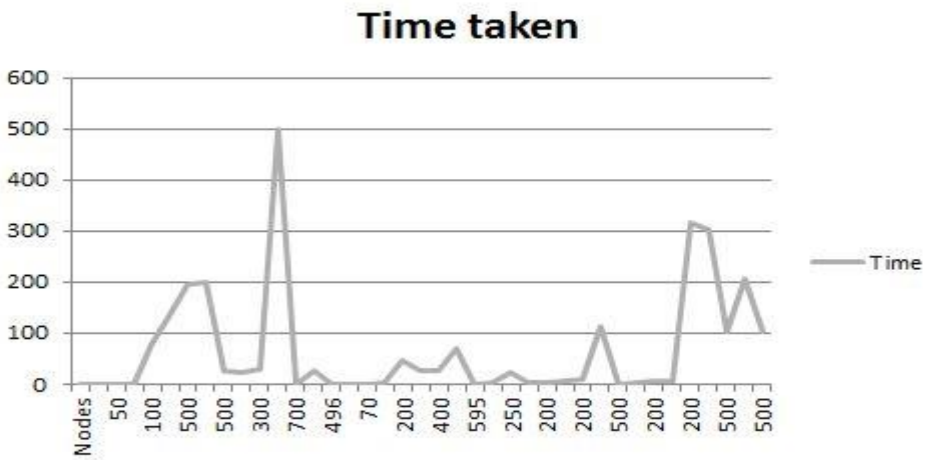

Figure 2. Time to Solve Graphs Demonstration

5. Conclusion and Future Work

Approximation methods are best way to deal with NP optimization problems and MVC is one of these problems. In this paper we have presented a new extra fast approximation algorithm for solving MVC generally in all graphs. The proposed algorithm is named degree contribution algorithm (DCA), a new data structure proposed and employed in this algorithm, name degree contribution. Degree Contribution helps in decisions for MVC. This is first time in literature that such a data structure for graph is proposed whose values are reflection from the whole graph. This is initial step and in future we are planning to optimize it as much as possible and also we will try to extend this work to MWVC and MWIS.

Table 2. Time Taken Results for DCA while Solving Benchmarks

Benchmarks	Nodes	Time
graph50_6	50	0.0468
graph50_10	50	0.0312
graph100_1	100	0.1872
graph100_10	100	0.2808
graph200_5	200	80.78
graph500_5	500	135.2373
graph500_2	500	194.736
graph500_1	500	200.002
phat300-2	300	25.59
phat300-1	300	21.8401
phat300-3	300	30.01
phat700-1	700	500.12
jhonson16-2-4	120	0.2496
jhonson32-2-4	496	26.5202
jhonson8-2-4	28	0.0313
jhonson8-4-4	70	0.0624
sanr200-0.9	200	0.7332
sanr200-0.7	200	1.8876
sanr400_0.5	400	45.4899
sanr400_0.7	400	27.7994
fbr_30_15_5	450	26.1302
fbr-35-17	595	70.5593
c 125	125	0.156
c 250	250	1.6536
c500.9	500	22.7605
broc200-2	200	2.9172
broc200-4	200	3.9312
gen200_p0.9_55	200	5.3352
Hamming-8-4	256	10.0777
dsjc500	500	112.6951
keller4	171	1.2324
cfat200-5	200	2.5584
cfat200-1	200	5.068
cfat200-2	200	4.5864
cfat500-1	500	316.5104
cfat500-2	500	303.0631
cfat500-10	500	103.9591
cfat500-5	500	204.5953
mann-a27	378	102.1

Acknowledgements

We are thankful to Mr.Muhammad Fayaz whose interesting nature and friendship helped us to devise an algorithm and work on vertex cover.

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, **(2004)**, Berlin Heidelberg
[2] Gary M. and Johnson D., Computers and intractability, New York: Freeman, **(1979)**
[3] Karp R., Reducibility among combinatorial problems, New York: Plenum Press, **(1972)**
[4] Demaine E. et al, Sub exponential parameterized algorithms on bounded-genus graphs and H Minor-free graphs, Journal of the ACM (JACM), **52(6)**, 866-893, **(2005)**
[5] Dinur. I and Safra S., on the hardness of approximating minimum vertex cover, Annals of Mathematics, **(162)**, 439-485, **(2005)**
[6] Li S., et al, An Approximation Algorithm for Minimum Vertex Cover on General Graphs, in Proc. 3rd International Symposium: Electronic Commerce and Security Workshops (ISECS ’10), China: Academy, pp. 249-252, **(2010)**
[7] V. Chvatal, A Greedy Heuristic for the Set-Covering Problem, Mathematics of Operations Research, 4, **(1979)**
[8] K. Clarkson, A modification to the greedy algorithm for vertex cover problem, Information Processing Letters, 16, **(1983)**
[9] F. Delbot and C. Laforest, Analytical and experimental comparison of six algorithms for the vertex cover problem, Journal of Experimental Algorithmics, 15, **(2010)**
[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, **(1994),** New York: ACM
[12] D. Avis and T. Imamura, A List Heuristic for Vertex Cover, Operations research letters, 35, **(2007)**
[13] F. Delbot and C. Laforest, A better list heuristic for vertex covers, Information Processing Letters, 107, **(2008)**
[14] S. Balaji, V. Swaminathan and K. Kannan, Optimization of Un-weighted Minimum Vertex Cover, World Academy of Science, Engineering and technology, 67, **(2010)**
[15] S. Gajurel and R. Bielefeld, A Simple NOVCA: Near Optimal Vertex Cover Algorithm, Procedia Computer Science, 9, **(2012)**

Authors

Imran Khan, his a 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.

Hasham Khan, his a IT Officer at Zong Head Office, Islamabad Pakistan and working with Imran khan since 2013 on NP complete problems.

