Implementation and Comparison of Vertex Cover Problem using Various Techniques

Reshu Tyagi

Muskaan Batra

ABSTRACT
The problem of finding Minimum Vertex Cover for graph belongs to the class of NP Complete and plays a key role in Computer Science Theory. The problems which belong to NP Complete set are not solvable in polynomial time in any known way. Since finding Minimum Vertex Cover (MVC) for a graph belongs to NP Complete class; so we are dubious to solve it in any polynomial time algorithm. Such problems are solved by algorithms which promise to give near optimum solution.

In this paper we have analyzed and scrutinized such algorithms like greedy algorithm, approximation algorithm, simple genetic algorithm (GA), primal-dual based algorithm (PDB), Alom's algorithm etc. on random directed and undirected graphs and found that all the algorithms give near optimum solution with a negligible performance difference. It was also observed that out of all the above said algorithms Alom's Algorithm is more effective in finding MVC for undirected graphs and for weighted graphs, superior performance is attained by primal-dual based approach.

Further the algorithm was implemented using JAVA and output demonstrates the various possible combinations of Minimum Vertex Cover.

Keywords
Vertex Cover, Approximation, Branch and Bound, Greedy, Alom's, Primal Dual, Genetic.

1. INTRODUCTION
The VC interrogation is a NP-complete interrogation [1]. A interrogation is NP-complete if we cannot find its polynomial-time principle procedure for resolving it to the accurate point. However, this does not mean this is all. We have 2 approaches for resolving NP-complete interrogations .Firstly if the real inputs are small, the principle procedure with no. raised to an exponent running time could be perfectly satisfactory. In another approach, it could be easy to find in time a "near optimum resolutions". In general, the near optimality or Approx is generally good enough. A principle procedure giving "near-optimum" resolution is known as Approx principle procedure.

In the subject of computer education, the "**VC interrogation"** or "**node cover interrogation"** is one of the 21 Karp NP-complete interrogations.

MIN VC problem (MVCP)[2] has attracted many researchers and practitioners due to two reasons. First one is for its NP-completeness and secondly, it can be helpful for resolving many tough real-life difficulties that can be formulated as instances of this problem but we have only few researches existing that analyze the performance and output of metamorphosis methods and principle procedures.

Now let us elaborate what exactly does this MIN vertex cover problem(MVCP)[2] means.

Explanation: Given a one dimension representation G= (V, E) where V and E are respectively vertex and edges. A vertex cover (VC) of a provided undirected one dimension representation is a subset V'⊆V such that if (v, u) is an edge of G(the one dimension representation provided), then either v belongs to V' or u belongs to V' or both.

So, the cumulative size of the VC is the cumulative no. of vertices present. The VC problem (VCP) is "To find MIN sized VC in an undirected one dimension representation". Such type of VC is known as an OVC (optimum VC). Few books describe an efficient Approx principle procedure with O(E) time for VC problem(VCP) .

We have 2 versions of the MIN VC (MVC): First is the decision version and second is optimization. In the decision version, we have to verify if there exists a vertex cover (VC) of a provided size for a one dimension representation. On the other side, in the optimization version of interrogation, we find a VC of MIN size.

MVC is amongst the Karp's21 [3] diverse combinational and one dimension representation theory interrogations, which are NP-complete. MVC is an important case of the "groups cover interrogations" which take inputs as an arbitrary gathering of sub covers S = (S_1, S_2, .., S_n) of the universal groups U, and the purpose is to find the smallest subset of sub covers from S whose union is U. The MVC interrogation is also related to other complex one dimension representation interrogations and so it attracts the researchers towards this field of designing, most effective and Approx principle procedures. The physicists recently were driven towards the study of NP-complete interrogations like VC. The reason for this is, when studied on suitable random ensembles, the interrogation shows phase transitions in the solvability which often coincides with the peaks in the typical computation complexity or changes of the typical complexity from no. raised to an exponent to polynomial**.**

2. PROBLEM STATEMENT
We are given a random undirected Graph G=(V,E) and we have to find a minimum set of vertices V' such that all the edges are incident to at least one vertex form the vertex set V'.

The **minimum vertex cover problem** is the optimization problem of finding a smallest vertex cover in a given graph.

INSTANCE: Graph G

OUTPUT: Smallest number k such that G has a vertex cover of size k.

If the problem is stated as a decision problem, it is called the **vertex cover problem**:

INSTANCE: Graph G and positive integer k.

QUESTION: Does G have a vertex cover of size at most k?

The vertex cover problem is an NP-complete problem: it was one of Karp's 21 NP-complete problems. It is often used in computational complexity theory as a starting point for NP-hardness proofs.

3. RELATED WORK

Various researchers have proposed various methods to solve NP Complete Vertex Cover Problem. Delbot and C. Laforest[5] had proposed an effective algorithm in which they scanned all vertices of a graph from left to right depending upon one condition , which was"u is added to vertex cover if and only if it has at least one neighbor not in the cover".This algorithm is called as LIST LEFT [5]which is given as follows:

Labeled graph L(G) = (L(V), E)

```
1.  A ← Ø
2.  For each vertex, v є L(V) do
3.  If v has at least one right neighbor, then
4.  4.A ←A U {v}
5.  Return A
```

Another technique to deal with this problem is SORTED-LL[7], which is also denoted by SLL, is an extension of LISTLEFT and has been described in the article of D. Avis and T. Imamura. It solves the problem as"if there is at least one v є N(u)with lower degree, select u; otherwise, if u has only neighbor with higher degree, u is not selected."

Eric Angel , Romain Campigotto and Christian Laforest [6] suggested the algorithm ANTI SORTED-LL, works like SORTED-LL on sorted lists, but it reads the lists from the end to the beginning. Algo is as follows:

Labeled graph L(G) = (L(V), E)

```
1. A ← Ø
2. For each vertex, v є L(V) do
3. If u has at least one right neighbor with a larger degree or a
left neighbor with the same degree, then
4.A ←A U {v}
5. Return A
```

Apart from this various other authors and researchers have used and proposed various methods to deal with MVCP.

4. VARIANTS OF VERTEX COVER

Now we should talk about variants of the VCP -Capacitated VC[8], Connected VC[8] and MAX Partial VC[8] have been extensively researched in terms of polynomial-time Approx. With the help of contrast, their parameterized order has not been resolved , so it is open ended. We can close this gap by showing that, with the size of the VC as parameter, Capacitated VC and Connected VC both are fixed-parameter tractable while the MAX Partial VC is W[1]-complete. The results provide several related interrogations. Although the considered variants of VC seem very same as in terms of constant-factor Approx, they display a wide range of characteristics when investigating their parameter complexities.

5. ALGORITHMS FOR VERTEX COVER PROBLEM

We have two classes of principle procedures: incomplete and complete ones. Complete principle procedure ensures best or true resolution. Hence the full resolution space needs to be searched in principle. Incomplete principle procedures, do not ensures that the true resolution or the global best is what we got. This part of paper provides MAX already researched principle procedures to solve VC interrogation. That are (i) BB principle procedure approach[4] , (ii) Approx principle procedure[4], (iii) Greedy principle procedure approach[4][9] (iv) Genetic principle procedure approach[4] (v) PDA(Primal-Dual)[4] and (v) Alom's principle procedure[4].

5.1 Approx principle procedure

A lot of interrogations are important to analyze and optimize as getting an optimum resolution is intractable. If the Provided interrogation is NP-complete, it is very hard to get a polynomial-time principle procedure for resolving it accurately, but still, we cannot lose hope. In general, near optimality is mostly better. A principle procedure that results in near optimal resolutions is known as an **Approximation (Approx) principle procedure[4][14].**

Existing approximation algorithm of vertex cover problem

```
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
```

5.1.1 Complexity Analysis of the approximate vertex cover algorithm

Since the loop in algorithm 3, on lines (3-6) repeatedly picks an edge (u, v) from E′ adds its endpoints u and v to C, and deletes all edges in E′ that are covered by either u or v. The running time of this algorithm is O (E).

5.2 Greedy Approach to find VC

To find result of a most effective interrogation, we find the groups of participants having a result that optimizes (maximizes or minimizes) the worth of the objective function. The greedy principle procedure[9][4] executes step by step. Firstly the groups of selected participants is empty. Then at every step, we try to find and add the best out of rest participants in the groups, our selection must be guided by selection function. The selection function depends on the present interrogation. For an instance, the selection function in the case of MIN weight spanning tree selects an edge of MIN weight from the every rest of the edges, an object with MAX profit per unit weight out of the rest of objects is chosen for placing in the knapsack if knapsack interrogation is the case. If the large groups of selected participants is no longer correct, we discard the candidate we just added: removed candidate is never considered again. In the project, if the larger groups is still feasible, then the candidate we just added remains in the groups of selected participants from then and on. Every time we large the groups of selected participants, we find whether the groups now constitutes a resolution of the interrogation.

A generally accepted method to construct successively space of resolutions is greedy approach i.e based on the proved principle of selecting the (local) best choice at every stage of the principle procedure in order to get the global best of few objective function.

5.2.1 Greedy Principle procedures of VC interrogation

```
1. C ← Ø
2. while E ≠ Ø
3. Pick any edge e ∈E and choose an end-point v of e
4. C ← C U {v}
5. E ← E \ {e ∈ E : v ∈ e}
6. return C
```

5.2.2 Clever Greedy Principle procedure

```
1. C ← Ø
2. while E ≠ Ø
3. Pick a vertex v ∈V of maximum degree in the current
graph
4. C ← C U {v}
5. E ← E \ {e ∈ E : v ∈ e}
6. return C
```

5.3 BB Principle procedure

Branch and Bound (BB)[4] is a general principle procedure for getting optimum results for various problems and more importantly in discrete and combinatory . It contains systematic gathering of every candidate resolutions where covers of useless participants are discarded, with the help of lower and upper assumed bounds of the quantity being optimized. Branch-and-bound is a technique for exploring an implicit directed acyclic graph like the backtracking method. Optimal solutions to some problems like assignment of tasks to workers, etc. can be found using the technique of branch-and-bound. The branch-and-bound (BB) algorithm is a complete algorithm, meaning that it guarantees the exact solution even though the time complexity may increase exponentially with the graph size. Every vertex is set as free in the beginning of the principle procedure. The principle procedure forward by marking a random free vertices as covered if the vertex has free or not included neighbors. The size of the biggest found VC is n,. If the n covering marks are not in use, the principle procedure can move on with the tree searching, else it needs to backtrack. If the principle procedure come back to the node via backtracking, then we make the vertex non-covered and then the branch of the configuration tree is considered. If every neighbor of node are covered, then it is firstly made noticeable and treated as non-covered. We use a simple bound to search the configuration tree ; we should not mark any vertex as non-covered, although it is having non-covered neighbors.

The bound that we are with the help of in the below principle procedure applies the present vertex degree v(i), which signifies the no. of non-covered neighbors at a distinct stage of the calculation. By including a vertex in the sum we have reduced no. of not covered edges by v(i). If various vertices t_1, t_2, . . . , t_k are covered, the no. of non-covered edges is max decreased by $v(t_1) + v(t_2) + \ldots + v(t_k)$. Assume that tIn the project is a certain stage while we are backtracking the tree, we need to uncover D edges non-covered and still m vertices to cover. Then a lower bound S that is for the MIN no. of non-covered edges in the sub tree is Provided by $S= \text{Max}[0,D\text{-max } v(t_1)+v(t_2)+\ldots+v(t_k)]$.

The below representation shortens the principle procedure for recount every arrangements exhibiting a MIN no. of non-covered edges. Assume G= (V,E) is a one dimension representation, n is the no. of vertex to cover and j is the no. of edges to cover. In the starting n= x and j= |E|. The variable t is assigned with t = |E| and contains the MIN no. of non-covered edges found so far. The solution of t is passed with the help of call by reference. At the beginning every vertices v ∈ V are made noticable as free. These marks are assumed to be passed with the help of call by reference also. Also, it is presumed that tIn the project is a groups of (best) resolutions can be stored.

<u>Algorithm min-cover (G, k, uncov, opt)</u>

```
begin
if k = 0 then {leaf of tree reached?}
begin
if uncov < opt then {new minimum found?}
begin
opt := uncov;
clear set of stored configurations;
end;
store configuration;
end;
if bound condition is true then
return;
let i∈ V a vertex marked as free of maximal current degree;
mark i as covered;
k := k − 1;
adjust degrees of all neighbours j of i : d(j) :=    d(j) − 1;
min-cover(G, k, uncov − d(i), opt) {branch into ‗left‘ subtree};
mark i as uncovered;
k := k + 1;
(re)adjust degrees of all neighbours j of i: d(j):= d(j) + 1;
min-cover (G, k, uncov, opt) {branch into ‗right‘ subtree};
mark i as free;
end
```

In the actual implementation, the algorithm does not descend further into the tree as well, when no uncovered edges are left. In this case, the vertex covers of the corresponding sub tree consist of the vertices covered so far and all possible selections of *k* vertices among all uncovered vertices.

5.4 Genetic Principle procedure

Genetic principle procedure[4] is a most effective approach based on the natural metamorphosis. It maintains a aggregation of strings, known as gene carrier that encrypt candidate resolutions to an problem The principle procedure chooses few parent gene carrier from the aggregation groups as per the their fitness worth, which are analyzed with the help of fitness function. The most fit gene carrier have more chances of getting selected for genetic operations in future generation. Different types of genetic processors are applied to the selected parent gene carrier, obviously as per the possibility of processor, and future generation aggregation groups is produced. In every generation, a new groups of artificial creatures is created with the help of bits and pieces of the most fit gene carrier of the old aggregation.

Although GA may or may not give correct result, in most cases it produces better aggregation as compared to their parent aggregation As selected parents are the most fit among the whole aggregation groups, and the worse gene carrier die off in successive generations. This process is continued till some user defined ending criteria is satisfied.

GA provide variation processors inspired by natural metamorphosis and genetics. The fitness function has a significant role in GA. As it decides how good a gene carrier is. The Fitness function can be no. of vertices helpful for covering every edges in the one dimension representation.

$$M=\sum_{i=1}^{v} Vi \text{ where } V_i=1 \text{ if } V_i \in V_{cover} \text{ else } 0$$

In HGA, we next generation is produced from two parent gene carrier. So, in that way best 50% gene carrier will directly go in the future generation with the help of reproduction. Every gene carrier is helpful to form next generation with the help of heuristic vertex crossover processor (HVX). As we believe that every chromosome has few important genes, which may become useful to obtain global optimum resolution. Then mutation processor is applied to next generation. Mutation is helpful to expel local minima and it should be applied on every next generation.

Algorithm HVX

```
begin
V' = { }
Create tables VT and ET
VT = (F(v), N(v)), where F(v) is the frequency of the vertex v
in P1 and P2, N(v) is the degree of vertex v in G, for ∀ v ∈P1
and ∀ v ∈ P2
ET = E (x, y) for ∀ E∈ G
while ET <> { } do
Select v1 ∈ VT such that N(v1) > N(v) for ∀ v ∈VT. If more
than one vertex has same number of degree then select that
vertex, whose frequency (F(v1)) is high. If still more than one
vertex is candidate for selection then select any vertex
randomly. Say v1
ET = ET — {E(x, y) : x = v1 or y = v1)}
V' = V' — {v1}
end while
return V'
end
```

5.5 Primal-Dual Principle procedure

A primal-dual principle procedure[4] starts with a infeasible primal resolution and an feasible dual. All the way its execution this type of principle procedure improves the dual objective function i.e worth of the already dual resolution and it decreases the degree of infeasibility of the primal also at the similar time. The principle procedure ends as soon as the feasible primal resolution is there. The final dual resolution is helpful as a lower bound for the best resolution worth by means of weak duality.

1. Start with x = 0 (variables of primal LP) and y = 0 (variables of dual LP). The conditions that:

- y is feasible for Dual LP.
- Primal Complementary Slackness is satisfied.

Are invariants and hence, hold for the algorithm. But the condition that:

- Dual Complementary Slackness is satisfied.

Might not hold at the beginning of algorithm. x does not satisfy the primal LP as yet.

2. Raise some of the ye‘s, either simultaneously or one-by-one.
3. Whenever a dual constraint becomes tight, freeze values of corresponding y‘s and raise value of corresponding x.
4. Repeat from Step 2 until all the constraints become tight.

Now let us consider the primal-dual algorithm for vertex cover.

Primal-Dual Algorithm for Vertex Cover

1. Start with x = 0 and y = 0.
2. Pick any edge e for which ye is not frozen yet.
3. Raise the value of ye until some vertex constraint v goes tight.
4. Freeze all ye‘s for edges incident on v. Raise xv to 1.
5. Repeat until all ye‘s are frozen.

5.6 Alom’s principle procedure for VC

Monjurul alom[4][14] discussed a new principle procedure for VC interrogation that provides the efficient approximate resolution that is better than existing approximate principle procedure, greedy approach and genetic principle procedure. This VC principle procedure selects the vertex which has MAX no. of edges striking to it. Every the edges are discarded striking to that vertex. If more than we vertex have same MAX no. of edges, this principle procedure select that vertex which have minimum we edge that is not covered by other vertices, which has MAX edge. This process is repeated until to cover every the vertices of the one dimension representation. This principle procedure takes same time as the existing approximate principle procedure takes but it provides the resolution that is always better than the approximate resolution.

```
1. OPTIMAL_VT_COVER (E, V) {// E is an edge and V is
   an vertex
2. V′ ←ϕ;
3. E′ ←E [G]
4. While (E′ ≠ ϕ) {
5. M ← Choose vertex which has maximum incident edge;
6. If (More than one vertex have maximum number of edges)
   then
7. M ← Choose that node which has at least one edge that is
   not covered by others which have maximum number of
   edges.
8. V′ ← V′ U M;
9. Remove the all incident edges at vertex M;
10. Count incident edge of new graph.}
11. Return V′ }
```

5.6.1 Complexity anlaysis of Alom’s principle procedure

As, the no. of iterations in the loop is at most E. So time complexity of this Principle procedure is O (E), where E denotes cumulative no. of edges.

6. RESULT AND ANALYSIS

This part provides the analysis of every provided principle procedures and complexity as shown below in the table. We show the behavior of every researched principle procedures such as greedy principle procedure, Approx principle procedure, genetic principle procedure, primal-dual principle procedure and alom’s principle procedure on the below one dimension representation.

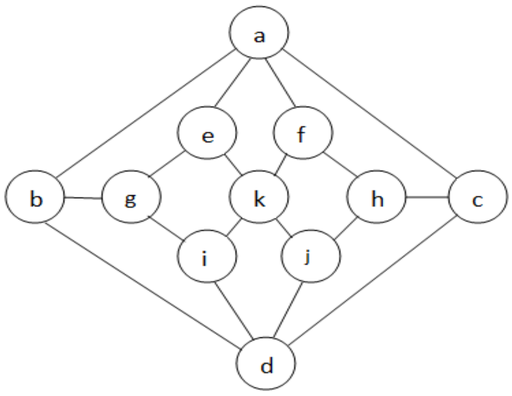


Fig.1: Graph G with 8 vertices

Table 1: Adjacency List of above graph

Vertex v	**No. of edges connected to v**	**Connected edges**			
a	4	b	c	e	f
b	3	a	d	g	
c	3	a	d	h	
d	4	b	c	i	j
e	3	a	g	k	
f	3	a	h	k	
g	3	b	e	i	
h	3	c	f	j	
i	3	d	g	k	
j	3	d	h	k	
k	4	e	f	i	j

6.1 Comparison analysis of presented Algorithms on above Graph.

Algorithm: Branch and bound

Size of Vertex Cover: 5

Solution Set: {a,d,g,h,k}

Complexity: grows exponentially fast with problem size for all values of c.

Remarks: If no vertex cover of the desired size is found, some covering marks have to be removed and be placed elsewhere, i.e. the algorithm has to backtrack.

Algorithm: Approximation

Size of Vertex Cover: 10, 6, 6

Solution Set: {a,b,c,d,e,g,h,i,j,k}{b,c,g,h,i,k} {b,c,e,f,i,j}

Complexity: O(V+E)

Remarks: This is a polynomial-time 2- approximation algorithm means that the solution returned by algorithm is at most twice the size of an optimal.

Algorithm: Greedy , Clever greedy

Size of Vertex Cover: 7,5

Solution Set: {a,b,c,g,h,i,k}, {a,d,g,h,k}

Complexity: O(V+E) , O (logV)

Remarks: Greedy algorithm is not a 2- Approximation. Clever greedy algorithm always gives solutions better than simple greedy.

Algorithm: Genetic
Size of Vertex Cover: 6

Solution Set: {a,c,d,g,h,k}

Complexity: Time complexity measured by the overall number of candidate solutions examined until the optimum is found. **Remarks**: GA fails to obtain consistent results for specific type of regular graphs. For large problems, the growth of the number of evaluations required by GA becomes faster.

Algorithm: Primal dual
Complexity: O(V log V+E)

Remarks: It reduces the degree of infeasibility of

the primal one at the same time. The algorithm terminates as soon as the primal solution is feasible.

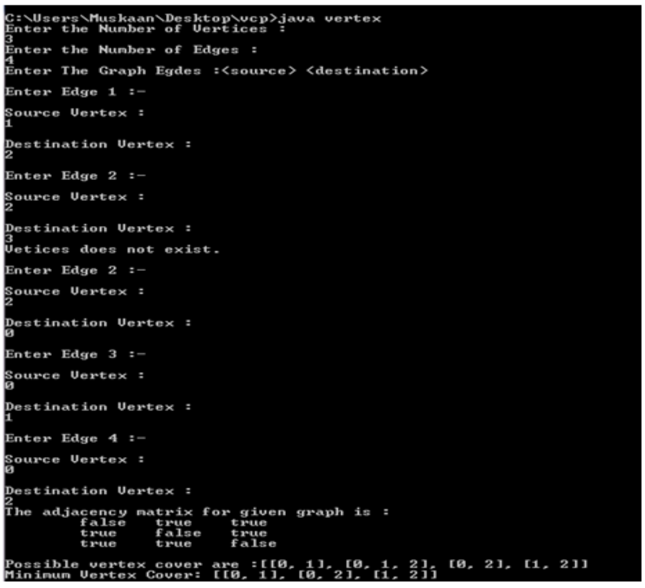
Algorithm: Alom‘s
Size of Vertex Cover: 5

Solution Set: {a,d,g,h,k}

Complexity: O(E)

Remarks: It gives always optimal solution to the given graph. Complexity is same as with approximation algorithm. For larger graphs, may be this algorithm lost to give an optimal solution.

We also implemented this algorithm in Java and below given is a snapshot of the output obtained .



```
C:\Users\Muskaan\Desktop\vcp>java vertex
Enter the Number of Vertices :
3
Enter the Number of Edges :
4
Enter The Graph Egdes :<source> <destination>

Enter Edge 1 :-

Source Vertex :
1

Destination Vertex :
2

Enter Edge 2 :-

Source Vertex :
2

Destination Vertex :
3
Vetices does not exist.

Enter Edge 2 :-

Source Vertex :
2

Destination Vertex :
0

Enter Edge 3 :-

Source Vertex :
0

Destination Vertex :
1

Enter Edge 4 :-

Source Vertex :
0

Destination Vertex :
2
The adjacency matrix for given graph is :
        false   true    true
        true    false   true
        true    true    false

Possible vertex cover are :[[0, 1], [0, 1, 2], [0, 2], [1, 2]]
Minimum Vertex Cover: [[0, 1], [0, 2], [1, 2]]
```

7. CONCLUSION

This research work analyzed performance of the different principle procedures on MIN VC for some randomly chosen graphs. Different procedures like BB, simple genetic principle procedure (GA), greedy principle procedure, alom’s principle procedure and primal dual principle procedure produce different results under different situations.

1. BB approach forwards further when we calculate a bound . If the bound analyzed predicts that any further solution found must necessarily not more efficient than the efficient resolution found so far, then we do not need to explore that part of the graph. The BB principle procedure makes choice in the project to put covering marks on graph representation. If no VC of the specified size is found, backtracking is done which is done in a symmetric way which allows investigation of the full configuration space.

2. The standard Approx principle procedure performs by choosing the random edge of the graph representation thus providing bigger resolution sometimes. This principle procedure is a 2-Approx polynomial time principle procedure means that the resolution provided by principle procedure is at most twice the size of an optimum VC, as size of the optimum VC is unknown.

3. The greedy principle procedure returns better resolution to the issue than Approx principle procedure. The principle procedure necessarily makes the best choice at that point of time. A good greedy principle procedure always run further by considering the vertex with the highest degree, then adding it to the cover groups, and discarding it from the graph, and process is again executed. But the greedy heuristic does not guarantees always an optimum resolution.

4. Genetic principle procedure is weaker than a local step of BB. For big interrogations, the increment of the no. of evaluations required by GA gets faster. The HVX required for MIN VC interrogation, performs very well and provides optimum resolution fast. As per the HVX and LOT (local most effective approach), we can get optimum resolution with few generation and aggregation size .

5. The primal dual principle procedure is only helpful for weighted one dimension representations and principle procedure is a two-Approx. This decreases the degree of infeasibility of the primal one at that time.

6. Under specific situations, Genetic and greedy principle procedures surpasses Branch and Bound, that is not an astonishing output as Branch and Bound is an efficient method that gives accurate global best result.

7. At last, the alom's principle procedure which is the best principle procedure for the VC interrogation as it returns optimum resolutions in almost cases. This principle procedure also executes by selecting a random edge when the condition coincide. So, for few bigger graphs Alom's principle procedure may not return the accurate optimum resolution . That is why the alom's principle procedure extension is provided so that it returns every possible resolutions i.e best MIN VCs . From every possible available solutions we can easily select the accurate optimum resolution which we want. Extended Aloms principle procedure is more complex as compared to others.

8. ACKNOWLEDGEMENT

We would like to thank all the authors and researchers who have done valuable research on the above discussed topic. We would also like to express our gratitude to Dr. Deepak Garg and Mr. K.V.R Kumar [4]. His research work on this topic was very valuable and we were able to write this paper. We humbly extend our thanks to all the people whose research co-operated us in writing this paper.

9. REFERENCES

[1] http://cse.unl.edu/~choueiry/Documents/intro_to_npc.pdf

[2] https://en.wikipedia.org/wiki/Vertex_cover

[3] Richard M. Karp "Reducibility among Combinatorial Problems" http://www.cs.berkeley.edu/~luca/cs172/karp.pdf

[4] http://www.gdeepak.com/thesisme/thesisChoosing%20the%20Efficient%20Algorithm%20for%20Vertex%20Cover%20problem.pdf

[5] Delbot and C. Laforest "A Better list heuristic for vertex Cover" Inf. Process. Lett. 107, 125–127 (2008)

[6] Eric Angel , Romain Campigotto and Christian Laforest "Algorithm for the vertex cover problem on large graphs." IBISC Research report (2010)

[7] Kartik Shah, Praveenkumar Reddy and R. Selvakumar Vertex Cover Problem- Revised Approximation Algorithm

[8] Jiong Guo, Rolf Niedermeier et al. "Parameterized Complexity of Vertex Cover Variants" Institut f¨ur Informatik, Friedrich-Schiller-Universit¨at Jena,

[9] Harsh Bhasin, Mohammed Amini "The Applicability of Genetic Algorithm to Vertex Cover" International Journal of Computer Applications(0975-8887) Volume 123 No. 17, August 2015

[10] Omar Kettani, Faycal Ramdani, Benaissa Tadili "A Heuristic Approach for the vertex Cover Problem" IJCA(0975-8887)Volume 82, No.4-2013

[11] Imran Khan, Sangeen Khan "Experimental Comparison of Five Approximation Algorithms for minimum Vertex Cover" International Journal of u- and e- Service, Science and Technology Vol. 7, No. 6 (2014), pp. 69-84

[12] Sushil Chandra Dimri, Kamlesh Chandra Purohit, Durgesh Pant "A greedy approach based Algorithm for the Vertex Cover Problem" International Journal of Scientific and Engineering Research Volume-4, Issue-3, March 2013

[13] Mohammed Eshtey et al. "NMVSA greedy solution for Vertex Cover Problem" *IJACSA*) International Journal of Advanced Computer Science and Applications, Vol. 7, No. 3, 2016

[14] Soumya Godi et al. "Several Algorithms to solve vertex Cover Problem" IJCMS, Vol.4, Issue 4, April 2015