# An Algorithm to Find Accurate Domination Number of a Graph

Jaishri B. Veeragoudar
*Department of Engineering Sciences and Humanities*
*KLE's College of Engineering and Technology, Belagavi, Karnataka, India*

Vijaylaxmi Patil
*Department of Masters of Computer Applications*
*KLS's Gogte Institute of Technology, Belagavi, Karnataka, India*

**Abstract- A dominating set D of a graph G=(V,E) is an accurate dominating set, if V – D has no dominating set of cardinality $|D|$. The accurate domination number $\gamma_a(G)$ is the minimum cardinality of an accurate dominating set. This concept was introduced by Kulli and Kattimani in [3]. In this paper, we present an algorithm to find domination number of any graph G without isolated vertices. We also check whether the domination number is the accurate domination number.**

**Keywords – Graph, Dominating set, Domination number, Perfect domination number**

## I. INTRODUCTION

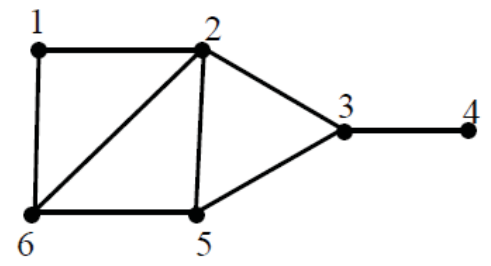
By a graph $G = (V, E)$, we mean a finite, undirected, graph without loops or multiple edges and no isolated vertices. Graphs are mathematical structures used to model pairwise relations between objects. A graph in this context is made up of vertices, nodes, or points which are connected by edges, arcs, or lines. We assume that |V |= n throughout this paper. Terms not defined are used in the sense of Harary [2] . A set D of vertices in a graph G is a dominating set of G if every vertex in V−D is adjacent to some vertex in D. The domination number $\gamma(G)$ of G is the minimum cardinality of a dominating set. A dominating set D of a graph G is an accurate dominating set, if V−D has no dominating set of cardinality |D|. The accurate domination number $\gamma_a(G)$ is the minimum cardinality of an accurate dominating set. This concept was introduced by Kulli and Kattimani in [3]. Investigation of some bounds for γ(G) and its relationship with other domination parameters has been done in [4] . Informally, an algorithm is any well-defined computational procedure that takes some value or set of values, as input and produces some value, or set of values, as output [1]. An algorithm is thus a sequence of computational steps that transform the input into the output. For the adjacency-matrix representation [1] of a graph G, we assume that the vertices are numbered 1, 2,...,|V |in some arbitrary manner. Then the adjacency-matrix representation of a graph G consists of a $|V| \times |V|$ matrix $A = (a_{ij})$ such that

$$(a_{ij}) = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & otherwise \end{cases}$$

For our convenience, we consider the following as the adjacency-matrix representation of a graph G:

$$(a_{ij}) = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & otherwise \\ 1 & \text{if } i = j \end{cases}$$

Following the definition of adjacency-matrix representation of a graph we write the adjacency matrix of a simple graph, see Figure 1.



(a)

$$\begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 2 & 1 & 1 & 1 & 0 & 1 & 1 \\ 3 & 0 & 1 & 1 & 1 & 1 & 0 \\ 4 & 0 & 0 & 1 & 1 & 0 & 0 \\ 5 & 0 & 1 & 1 & 0 & 1 & 1 \\ 6 & 1 & 1 & 0 & 0 & 1 & 1 \end{array}$$

(b)

Figure 1. (a) A graph G with 6 vertices (b) Adjacency-matrix of graph G

Following are the well-known algorithms for accepting and printing the adjacency-matrix representation of a graph G.

Algorithm A. To read the adjacency matrix of the given graph
Input : A graph G = (V,E) with V (G) = {1, 2, 3, ..., n}
Output : Adjacency matrix $A$ ■ $(a_{ij})$ in computer memory.

```
Step 1. For i = 1 to n
            For j = 1 to n
                Input $a_{ij}$
Step 2. Exit
```

Algorithm B. To print the adjacency matrix of the given graph
Input : Adjacency matrix $A$ ■ $(a_{ij})$
Output : Adjacency matrix $A$ ■ $(a_{ij})$ on screen

```
Step 1. For i= 1 to n
            For j = 1 to n
                Print $a_{ij}$
Step 2. Exit
```

In this paper, we develop algorithms using adjacency-matrix representation of the given graph. First we develop an algorithm to find whether the given graph G is of domination number 1 and check whether it is an accurate domination number also. Next we develop algorithm to find whether the given graph G is of domination number 2 and check whether it is an accurate domination number also. In this paper we have exclusively presented algorithms to check for domination number 1, 2, 3 and 4. The concept of these algorithms can be used to write algorithms for checking domination number with higher values. Then we have used these concepts for writing the complete algorithm for finding the domination number of a given graph and check whether it is an accurate domination number also. Our algorithm produces one dominating set of the given graph. If $\gamma(G)$ ■ $\gamma_a(G)$ then we have one more dominating set of the graph from the algorithm.

## II. PROPOSED ALGORITHMS

Algorithm 1.1. To check whether domination number of the graph is1, if so get the dominating vertex set and check whether 1 is an accurate domination number.
Input : Adjacency matrix $A$ ■ $(a_{ij})$ .
Output : Whether the graph has domination number 1 if so whether it is accurate domination number also.

```
Step 1. For  i = 1 to n
          count = 0 ;
         For j = 1 to n
          count = count + $a_{ij}$
Step 2. If (count = n)
          Print 'Domination number = 1 and the dominating vertex is i'
```

```
Step 3. For r = i + 1 to n
            count = 0 ;
            For j = 1 to n
                count = count +$u_{rj}$
Step 4.If ( count= n)
            Print 'Another dominating vertex is r '
            Print 'In this case $\gamma(G) = \gamma_a(G)$ '
            Exit;
End of Step 3.
Step 5. Print 'No other vertex is a dominating vertex'
        Print 'In this case $\gamma(G) = \gamma_a(G)$ '
        Exit
End of Step 2.
End of Step 1.
```

Proof : Step 1 starts our procedure for the first row i = 1. We initialize count = 0 and add the entries in the first row of all columns j to count. In Step 2 we see if count = n. If count is n then dominating vertex is i = 1. If count ≠ n then Step 2 ends and the next value of i is taken from Step 1 and continued. If count = n for some row i then step 2 gives the domination number as 1 and dominating vertex is i. If there exists no such row that gives count = n then End of Step 1 is reached and the algorithm terminates. Once we have i as the dominating vertex from Step 2, Step 3 looks for another row r starting from r = i + 1 and add all the entries of the row r to count as in Step 1. In Step 4 if count = n then it gives r as the dominating vertex hence $\gamma(G) \neq \gamma_a(G)$ and we exit. If no such row r exist then we end Step 3 and Step 5 gives the conclusion that no other vertex than i is the dominating vertex, hence accurate domination number of the graph is also 1.

Algorithm 1.2. To check whether domination number of the graph is 2, if so get the dominating vertex set and check whether 2 is an accurate domination number.
Input : Adjacency matrix $A = (a_{ij})$.
Output : Whether the graph has domination number 2 if so get the dominating set and concluding whether it is accurate domination number also.

```
Step 1. For i = 1 to n
Step 2. For k = i + 1 to n
            count= 0
             For j = 1 to n
                count= count+ ($a_{ij}$ = 1 or $a_{kj}$= 1)
Step 3. If (count = n)
            Print 'Domination number = 2 and the dominating vertices are i and k '
Step 4. For  r1 = i + 1 to n
            If ( r1 ≠ i and r1 ≠k )
Step 5. For r2 = r1 + 1 to n
Step 6. If ( r2 ≠ i and r2 ≠ k )
            count = 0
            For  j = 1 to n
                count = count+ ($a_{r1j}$ = 1 or $a_{r2j}$ = 1)
Step 7.If (count = n)
            Print 'Another set of two dominating vertices is r1 and r2'
            Print 'In this case $\gamma(G) \neq \gamma_a(G)$'
            Exit
End of Step 6.
End of Step 5.
End of Step 4.
Step 8.Print 'No other set of two vertices is a dominating vertex'
            Print 'In this case $\gamma(G) = \gamma_a(G)$'
            Exit.
End of Step 3.
End of Step 2.
End of Step 1.
```

Proof. This module works when the domination number of the graph is not 1. We look for the possibility of domination number 2. Step 1 starts the procedure by considering the first row entries i = 1. In Step 2 we consider next row k = i + 1. We initialize count = 0. We consider the entries of row i and row k in the column j = 1. If any of the entries is 1 we add 1 to count. If for any of the column j , this is not true, i.e both the entries are zero then End of Step 2 is reached and we start the procedure again by taking next i from Step 1. In Step 2 if count = n, then we conclude that the vertices i = 1 and k = i+1 = 2 are dominating vertices. If this is not the case then Step 2 looks for next value of k such that for all the columns j, at least one of the rows entries is 1. In Step3 if count = n some values of i and k then domination number of the graph is 2 and dominating vertices are i and k. If no such k exists then we reach End of Step 3. Once we have the dominating vertices as i and k, Step 4 looks for possibility of another set of rows r1 = i+ 1 such that r1 ≠ k . If this is not true then we reach End of Step 4 and take the next value for i. Once we have such r1 we look for r2 in Step 5 where r2 is a row near to r1 such that r2 ≠ i, k, r1. Once we have such rows r1 and r2 Step 6 finds the count as in Step 2. Step 7 checks whether count = n. If count = n then it prints r1 and r2 as another set of dominating vertices and hence in this case $\gamma(G) \neq \gamma_a(G)$. We exit from the algorithm. If no such pair of rows is found then we reach Step 8 saying no other vertex set of two vertices is the dominating set and hence in this case $\gamma(G) = \gamma_a(G)$ and we exit from the algorithm. Similarly, if once we know the domination number of the graph is not 2 we look for the possibility of domination number 3 for the graph and so on. In this paper, we are also presenting algorithms, Algorithm 1.3 and Algorithm 1.4, for domination numbers 3 and 4.

Algorithm 1.3. To check whether domination number of the graph is 3, if so get the dominating vertex set and check whether 3 is an accurate domination number.

Input : Adjacency matrix $A = (a_{ij})$ .

Output : Whether the graph has domination number 3 if so get the dominating set and find whether it is accurate domination number also.

```
Step 1. For row1 = 1 to n − 1
Step 2. For row2 = row1 + 1 to  < n
Step 3. For row3 = row2 + 1 to  < n
                count= 0
                For col = 1 to n
                        count= count + ($a_{row1col}$ = 1 or $a_{row2col}$ = 1 or $a_{row3col}$ = 1)
Step 4. If ( count = n)
                Print 'Domination number is 3 and the dominating vertices are row1, row2 and row3 '
Step 5. For r1 = row1 + 1 to <  n − 1
                If ( r1 ≠ row1 and r1 ≠ row2 and r1 ≠ row3 )
Step 6. For r2 = r1 + 1 to <  n
                If ( r2 ≠ row1 and r2 ≠ row2 and r2 ≠ row3 )
Step 7. For r3 = r2 + 1 to n
Step 8. If ( r3 ≠ row1 and r3 ≠ row2 and r3 ≠ row3 )
                count= 0
                For col = 1 to n
                        count = count + ($a_{r1col}$ = 1 or $a_{r2col}$ = 1 or $a_{r3col}$ = 1 )
Step 9. If ( count= n)
                Print 'Another set of three dominating vertices is r1, r2 and r3'
                Print 'In this case $\gamma(G) \neq \gamma_a(G)$'
                Exit
End of Step 8.
End of Step 7.
End of Step 6.
End of Step 5.
Step 10. Print 'No other vertex set of 3 vertices is a dominating vertex'
                Print 'In this case $\gamma(G) = \gamma_a(G)$ '
                Exit
End of Step 4.
End of Step 3.
End of Step 2.
End of Step 1.
```

Algorithm 1. 4. To check whether domination number of the graph is 4, if so get the dominating vertex set and check whether 4 is an accurate domination number)

```
Step 1. For row1 = 1 to < n − 2
Step 2. For row2 = row1 + 1 to < n − 1
Step 3. For row3 = row2 + 1 to < n
Step 4. For row4 = row3 + 1 to n
                count = 0
                For col = 1 to n
                count =  count + ( $a_{row1col}$ = 1 or $a_{row2col}$ = 1 or $a_{row3col}$ = 1 or $a_{row4col}$ = 1 )
Step 5. If ( count= n)
                Print 'Domination number  =  4 and the dominating vertices are row1 ,row2 , row3 and row4 '
Step 6. For  r1 = row1 + 1 to < n − 2
                If ( r1 ≠ row2 and r1 ≠ row3 and r1 ≠ row4 )
Step 7. For r2 = r1 + 1 to < n − 1
                If ( r2 ≠ row2 and r2 ≠ row3 and r2 ≠ row4 )
Step 8. For r3 = r2 + 1 to < n
                If (r3 ≠ row2 and r3 ≠ row3 and r3 ≠ row4 )
Step 9. For r4 = r3 + 1 to n
                count = 0
                For col = 1 to n
                count = count+( $a_{r1col}$ = 1 or $a_{r2col}$ = 1 or $a_{r3col}$ = 1 or $a_{r4col}$ = 1 )
Step 10. If ( count= n)
                Print 'Another set of four dominating vertices is r1, r2 , r3 and r4 '
                Print 'In this case $\gamma(G) \neq \gamma_a(G)$ '
                Exit.
End of Step 9.
End of Step 8.
End of Step 7.
End of Step 6.
Step 11. Print 'No other vertex set of 4 vertices is a dominating vertex'
                Print 'In this case $\gamma(G) = \gamma_a(G)$'
                Exit
End of Step 5.
End of Step 4.
End of Step 3.
End of Step 2.
End of Step 1.
```

## III. COMPLETE ALGORITHM

In this section, we present the complete algorithm to find the domination number of a given graph G = (V,E) without isolated vertices and check whether the domination number is an accurate domination number. The algorithm gives the dominating set as well as the other dominating set also if $\gamma(G) \neq \gamma_a(G)$. We look at the following results before giving the algorithm.

Theorem 1. [6] If G is a graph with no isolated vertices, then $\gamma(G) \leq \frac{n}{2}$.

Theorem 2. [4] For any graph G, $\gamma(G) \leq \gamma_a(G)$.

```
Algorithm 1.5. Main algorithm
Step 1. Input the number of vertices n of the given graph G.
Step 2. Input the adjacency matrix of the graph.
          Execute Algorithm A.
Step 3. Print 'The adjacency matrix of the graph'
          Execute Algorithm B.
Step 4. For i = 1 to n/2
          Execute Algorithm 1. i
Step 5. Stop
```

Proof. Step 1 takes the value n where n = |V |. Step 2 accepts the adjacency matrix of the graph G. Step 3 makes the matrix visible. Step 4 begins with i = 1 and calls for Algorithm 1.1 for checking the possibility of domination number 1. If this is not the case then next value i = 2 is taken and Algorithm 1.2 is executed to look for the possibility of domination number 2. The process continues up to  i = n/2. Theorem 1 and Theorem 2 supports us in stating whether the domination number is an accurate domination number or no.

## IV. CONCLUSION

Our algorithm works efficiently for any connected graph G with small number of vertices however, with the increase in the size of the graph, the length of the program increases reducing its efficiency. The basic concept of domination number and dominating sets has immense applications in the field of Graph Theory and Computer Science. This will help to program many other domination parameters leading the researchers in the design and analysis of algorithms to solve many of the graph-theoretic problems. Recursive algorithms can be tried to increase its efficiency.

The application of this algorithm can be made in flooding packets in Computer Networks.  Flooding is a technique where every incoming packet is sent out on every outgoing line except the one it arrived on. The issue with this technique is that it generates vast number of duplicate packets.  There are various techniques of controlling the number of packets that the duplicated and avoid congestion of the network[5].  Using domination number of block transformation graph, the node or nodes can be determined those connect to every node in the network and use that node for flooding the packets to all the nodes in the network.  If a network graph has an accurate domination number one then that node can be chosen to flood packets in the network.  If a graph has a domination number one and not an accurate domination number then one of the two or more nodes can be used to flood the packets.  If a network has domination number as two or more then the packets need to flood through these nodes.  The algorithm takes a maximum of $O(n^3)$ time to determine domination number in worst case (graph has no accurate domination) and $O(n^2)$ in best case (graph has no accurate domination number is available)[7].

## REFERENCES

[1] T. H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, "Introduction to algorithms",  MIT Press and Mc Graw-Hill, Newyork, 2001.
[2] F. Harary, "Graph theory", Addison-Wesley, Reading, Mass, 1969.
[3] V. R. Kulli and M. B. Kattimani, "The accurate domination number of a graph", Technical Report 2000:01, Dept. Mathematics, Gulbarga University, Gulbarga, India, 2000.
[4] V. R. Kulli and M .B. Kattimani, "Accurate domination in Graphs, Advances in Domination Theory 1", Vishwa International Publications, 1-8, 2012.
[5] Levitin, Anany,Introduction to the Design and Analysis of Algorithms Anany Levitin - 2nd ed. - New Delhi Pearson 2011.
[6] O. Ore, Theory of graphs, Amer. Math. Soc. Colloq. Publ. 38. Providence, 1962.
[7] Tanenbaum, S, Andrew ,The  Network Layer, in Computer Networks Andrew. S. Tanenbaum. - 4th Ed - New Delhi PHI 2006, ch.5, sec.5.2.3, pp.368–370