International Journal of Engineering Research & Management Technology ISSN: 2348-4039
September-2015 Volume 2, Issue-5
Email: editor@ijermt.org www.ijermt

Algorithms on Distance Closed Dominating Sets

V. Sangeetha
Department of Mathematics
Christ University
Bangalore, India.

T. N. Janakiraman
Department of Mathematics
National Institute of Technology
Tiruchirappalli, India

ABSTRACT:
In a graph G = (V, E), a set $S \subset V(G)$ is a distance closed set of G if for each vertex $u \in S$ and for each $w \in V - S$, there exists at least one vertex $v \in S$ such that $d_{<S>}(u, v) = d_G(u, w)$. Also, S is a dominating set of G, if each vertex in V – S is adjacent to at least one vertex of S. Combining the above concepts, a distance closed dominating set of a graph G is defined as follows: A subset $S \subseteq V(G)$ is said to be a ***D**istance **C**losed **D**ominating (D.C.D) set* if S is both distance closed and dominating set. The cardinality of a minimum distance closed dominating set of G is called the distance closed domination number of G and is denoted by $\gamma_{dcl}(G)$. In this paper, algorithms to find the minimum distance closed dominating sets of some special classes of graphs are discussed.

KEYWORDS:
Distance, eccentricity, radius, diameter, degree, paths, cycles, self-centred graphs, trees, distance closed set, distance closed domination number.

INTRODUCTION:
For a graph, let V(G) and E(G) denotes its vertex and edge set respectively. The *degree* of a vertex v in a graph G is the number of edges incident with v and is denoted by $\deg_G(v)$. The length of the shortest path between any two vertices u and v of a connected graph G is called the *distance between u and v* and it is denoted by $d_G(u, v)$. For a connected graph G, the *eccentricity* $e_G(v) = \max\{d_G(u, v) : \forall\, u \in V(G)\}$. If there is no confusion, we simply use the notion d(v), d(u, v) and e(v) to denote degree, distance and eccentricity of v respectively for the connected graph G. The minimum and maximum eccentricities are called the *radius* and *diameter* of G, denoted by r(G) and d(G) respectively. If these two are equal in a graph, that graph is called *self-centered* graph with radius r and is called an *r self-centered* graph. Such graphs are 2-connected graphs. If v is a vertex with e(v) = r(G), then v is called a central vertex of G and if e(v) = d(G), then v is called a peripheral vertex of G. Also a vertex u is said to be an eccentric vertex of v in a graph G, if d(u, v) = e(v) in that graph. In general, u is called an *eccentric vertex*, if it is an eccentric vertex of some vertex. For $v \in V(G)$, the *neighbourhood* $N_G(v)$ of v is the set of all vertices adjacent to v in G. The set $N_G[v] = N_G(v) \cup \{v\}$ is called the *closed neighbourhood* of v. Also, $N_i(v) = \{u \in V(G) / d(u, v) = i\}$ is called the i^{th} *neighbourhood* of v. A set S of vertices in a graph is said to be independent if no two vertices in S are adjacent. An edge e = (u, v) is a *dominating edge* in a graph G if every vertex of G is adjacent to at least one of u and v. One important aspect of the concept of distance and eccentricity is the existence of polynomial time algorithm to analyze them. The concept of distance and related properties are studied in [1] and [2].
The concept of domination in graphs was introduced by Ore [8] in 1962. It is originated from the chess game theory which paved the way to the development of the study of various domination parameters and then relation to various other graph parameters. A set $D \subseteq V(G)$ is called a *dominating set* of G if every vertex in V(G) – D is adjacent to some vertex in D and the *domination number* $\gamma(G)$ is the minimum cardinality of a dominating set. Different types of dominating sets have been studied by imposing conditions on the dominating sets. The list of survey of domination theory papers are in [3], [4] and [5].

PRIOR RESULTS:

The concept of distance closed set is defined and studied in the doctoral thesis of Janakiraman [6] and the concept of distance closed sets in graph theory is due to the related concept of ideals in ring theory in algebra. The ideals in a ring are defined with respect to the multiplicative closure property with the elements of that ring. Similarly, the distance closed dominating set is defined with respect to the distance closed property and the dominating set of the graph. Thus, the distance closed dominating set of a graph G is defined as follows:

A subset $S \subseteq V(G)$ is said to be a **D**istance **C**losed **D**ominating (D.C.D) set if

1. S is distance closed and
2. S is a dominating set.

The cardinality of a minimum distance closed dominating set of G is called the distance closed domination number of G and is denoted by $\gamma_{dcl}(G)$.

Clearly from the definition, $1 \leq \gamma_{dcl} \leq p$ and the graph with $\gamma_{dcl} = p$ is called a 0-distance closed dominating graph. The definition and the extensive study of the above said distance closed dominating sets in graphs are studied in [7]. The following are some important results proved in [8] has used here.

Proposition 3.2.1: If T is a tree with number of vertices $p \geq 2$, then $\gamma_{dcl}(T) = p - k + 2$, where k is the number of pendant vertices in T.

Proposition 3.2.2: If G is a 2 self-centered graph with a dominating edge, then $\gamma_{dcl}(G) = 4$.

Theorem 3.2.2: Let G be a graph of order p. Then $\gamma_{dcl}(G) = 2$ if and only if G has at least two vertices of degree $p - 1$.

Theorem 3.2.3: Let G be a graph of order p. Then $\gamma_{dcl}(G) = 3$ if and only if G has exactly one vertex of degree $p - 1$.

DOMINATION IN NETWORKS:

At present, domination is considered to be one of the fundamental concepts in graph theory and its various applications to ad-hoc networks, biological networks, distributed computing, social networks and web graphs partly explain the increased interest. Such applications usually aim at selecting a subset of nodes that will provide some definite services such that every node in the network is 'close' to some node in the subset. The following examples show when the concept of domination can be applied to real-life problems. In this paper, algorithms to find D.C.D sets of some special classes of graphs are studied. Also, a general algorithm to find a D.C.D set of any connected graph G with a given radius r and diameter d is given.

In general, the distance closed property in a graph can be checked in polynomial time. However, finding the connected domination number of a graph G is NP-complete. Hence, finding a minimum distance closed dominating set in a general graph is NP-complete and so attempts are made to develop polynomial time algorithms for finding D.C.D sets in some special classes of graphs. In all the algorithms, the distance matrix of G is computed as a pre-processing step, whose time complexity is $O(p^3)$. Further it is assumed that any graph G given as input is in the form of adjacency matrix and it is also assumed that the degrees of all vertices are part of the input. Hence, the complexity of each algorithm discussed in this chapter is given excluding the pre-processing time.

ALGORITHM TO FIND A MINIMUM D.C.D SET OF A GRAPH G WITH RADIUS 1

Input : Graph G with radius 1

Output : D.C.D set of G

Pseudocode:

Step 1:

For every vertex $v \in V(G)$, find $e(v) = \max\{d(v, u) \mid u \in V(G)\}$;

Step 2:

Set $C(G) = \{v \in V(G) \mid e(v) = 1\}$;

Step 3:

If (|C(G)| == 1) **then**

D = {*u*, *v*, *w*}, where *v* ∈ C(G) and *u*, *w* are any two non-adjacent vertices of V(G) – *v*;

Else D = {*u*, *v*}, where *u*, *v* ∈ C(G);

Step 4:

Output D; // *D is a D.C.D set of G*

Step 5:

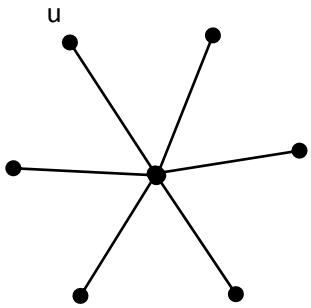
Exit

VALIDITY OF THE ALGORITHM:

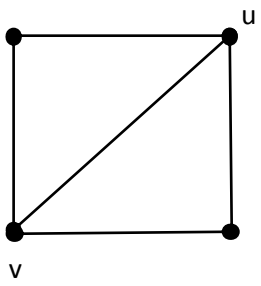
Validity of the above algorithm directly follows from Theorems 3.2.2 and 3.2.3.

COMPLEXITY OF THE ALGORITHM:

Since the distance matrix of G is given in t pre-processing step, the eccentricity of every vertex *v* in G in step 1 is computed in $O(p^2)$ units of time. In step 2, selecting the central vertices among the p vertices can be done in O(p) time. Also, the **if** loop in step 3 takes O(p) units of time. Thus the overall complexity of the algorithm is $O(p^2)$



(G_1)



(G_2)

Here, both the graphs G_1 and G_2 are of radius 1 and diameter 2. The graph G_1 has C(G) = {v} and the set {u, v, w} forms a D.C.D set for G_1. Also the graph G_2 has C(G) = {u, v} and the set {u, v} forms a D.C.D set for G_2.

ALGORITHM TO FIND A D.C.D SET OF A 2 SELF-CENTERED GRAPH:

Input : 2 self-centered graph G

Output : D.C.D set of G

Pseudocode:

Step 1:

Select a vertex *v* in G with minimum degree;

Step 2:

For i=1 to 2, do

{

Find $N_i(v) = \{u \in V(G) \mid d(v, u) = i\}$;

}

Step 3:

Set $R = \{u \in N_1(v) \mid N(u) \cap N_2(v) \neq \Phi\}$; *// R is the set of all vertices in $N_1(v)$ having*

// neighbours in $N_2(v)$

Step 4:
Choose u in R with maximum degree and w in $N(u) \cap N_2(v)$ with maximum degree;

Step 5:
Set $E(u) = \{u^* \in V(G) \mid d(u, u^*) = 2\}$; *// E(u) is the set of all eccentric*
// vertices of u

Step 6:
Set $D = \{v, u, w, u^*\}$, where $u^* \in E(u)$;

Step 7:
If (N[D] == V(G)) R**eturn** D as a D.C.D set of G;

Step 8:
Set $S = N(D) \cap N_2(v)$; Select a vertex z in S having maximum degree;

Step 9:
Set $D = D \cup \{z\}$ and **goto** Step 7;

Step 10:
Exit

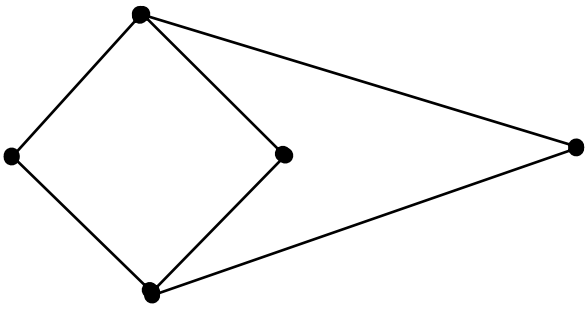
VALIDITY OF THE ALGORITHM:

For any 2 self-centered graph G, $\gamma_{dcl}(G) = 4$ if G has a dominating edge (proposition 3.2.2). Otherwise, $\gamma_{dcl}(G) \geq 4$ and the extra added vertices are only for the sake of domination. In step 6, the set D gives the distance closed set of G. If D dominates G, then D becomes the D.C.D set of G. Since the vertex v dominates all the vertices in $N_1(v)$, the maximum degree vertex is chosen from $N(D) \cap N_2(v)$ and added to D until D dominates G. Therefore, the final set D becomes a D.C.D set of G.

COMPLEXITY OF THE ALGORITHM:

1. Since the adjacency matrix of G is given as a part of input, selecting a vertex with minimum degree in G takes O(p) units of time. The **for** loop in step 2 gives the neighborhood sets of v and it needs O(p) time.
2. Each of the steps 3 to 5, for finding the other elements of D, takes O(p) units of time and forming the set D from the known values need constant time.
3. Testing whether D is a dominating set or not is done in steps 7 to 9 and it needs [O(p) + O(p) + constant)] = O(p) units of time.

Therefore, the above algorithm runs in O(p) units of time.



This is a self centered graph of diameter 2 with minimum D.C.D set {v, u, w, u^*}, where $u^* \in E(u)$ and also v and w are eccentric nodes of each other

ALGORITHM TO FIND A MINIMUM D.C.D SET OF A TREE:

Input : Tree T with $p \geq 2$

Output : D.C.D set of T

Pseudocode:

Step 1:

For every vertex $u \in V(T)$, find $e(u) = \max\{d(u, v) \mid v \in V(T)\}$;

Step 2:

Set $d = \max\{e(u) \mid u \in V(T)\}$;

Step 3:

Select a vertex v with $e(v) = d$, where degree of v is maximum among such vertices;

Step 4:

Set $T^1 = BFS(v)$; *// T^1 is the spanning tree rooted at v resulting from BFS*

Step 5:

In T^1, find a path P of length d, rooted at v, where P is $\{v = x_1, x_2 \ldots x_{d+1}\}$;

Step 6:

Set $D = \{x_1, x_2 \ldots x_{d+1}\}$, $S = V - D$;

Step 7:

For each $w \in S$ do

{

If $(\deg(w) \geq 2)$ **then**

$D = D \cup \{w\}$;

}

Step 8:

Output D; *// D is the minimum D.C.D set of T*

Step 9:

Exit

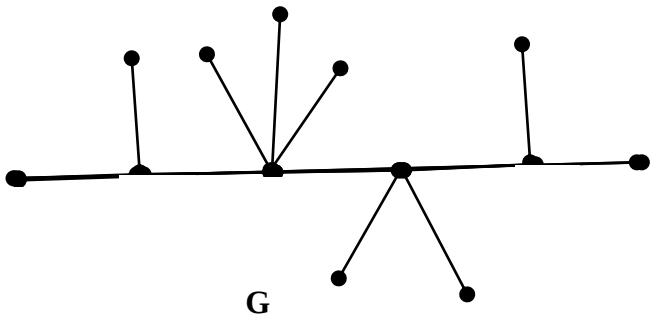
VALIDITY OF THE ALGORITHM:

If T is a tree with number of vertices $p \geq 2$, then $\gamma_{dcl}(T) = p - k + 2$, where k is the number of pendant vertices in T (from proposition 3.2.1). Thus, if D is a D.C.D set of T, then D must contain exactly two pendant vertices of T and all those vertices with degree greater than or equal to 2. Clearly the path P given in step 5 contains exactly two pendant vertices of T and therefore the set D. Also in step 7, we are adding those vertices of V – D with degree greater than or equal to 2 to D. Hence, the above algorithm gives the exact bound for $\gamma_{dcl}(T)$.

COMPLEXITY OF THE ALGORITHM:

1. Steps 1-3: The eccentricity of every vertex v of T is computed in $O(p^2)$ time. Finding the maximum eccentricity d is done in O(p) units of time and also selecting a maximum degree vertex v among all such vertices with eccentricity equal to d takes O(p) time.
2. Steps 4-7: Finding a path P of length d rooted at v using BFS algorithm takes O(p) units of time and assigning those vertices of P in D takes a constant time. Also, the **for** loop in step 7 needs O(p) units of time.

Thus the total complexity of the algorithm is $O(p^2)$.



G

This is a tree with radius 3 and diameter 5. Here, v is a vertex with eccentricity e(v)=5=d(G). Now find a path of length d=5 rooted at v by using BFS. Then the set $\{x_1, x_2, x_3, x_4, x_5, x_6\}$ forms a minimum D.C.D set for G.

ALGORITHM TO FIND A D.C.D SET OF A CONNECTED GRAPH G WITH RADIUS r AND DIAMETER d = 2r (OR) d = 2r – 1

Input : Connected graph G with radius r and diameter d = 2r (or) d = 2r – 1
Output : D.C.D set of G

Pseudocode:
Step 1:
For every vertex $u \in V(G)$, find $e(u)= \max\{d(u, v) \mid v \in V(G)\}$;
Step 2:
Set $d = \max\{e(u) : u \in V(G)\}$;

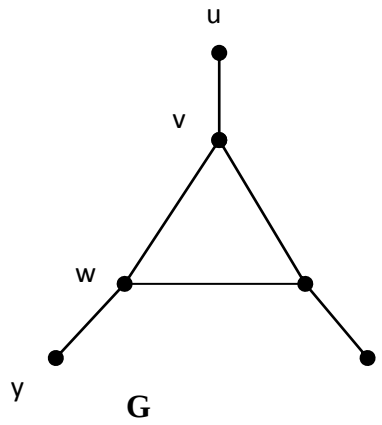
Step 3:
Select a vertex v with $e(v) = d$, where degree of v is maximum among such vertices;
Step 4:
Set $S = \{v\}$, $i = 1$;
Step 5:
If $(i \geq d)$ **then**
goto step 7;
Else
Set $T = N_i(v)$, $S^1 = \phi$, $k = 1$;
Step 6:
Choose a vertex $v_i^{(k)}$ from T with maximum degree;
$S^1 = S^1 \cup \{v_i^{(k)}\}$;
If $(N(S^1) == N_{i+1}(v))$ then
{
$S = S \cup S^1$;
$i = i + 1$;
goto step 5;
}
Else
{
$T = T - S^1$;
$k = k+1$;
goto step 6;
}
Step 7:
Set $D = S \cup \{w\}$, where $w \in N_d(v)$;
Step 8:
Output D; *// D is a D.C.D set of G*
Step 9:
Exit

VALIDITY OF THE ALGORITHM:
If G is a connected graph with radius r and diameter d = 2r (or) d = 2r – 1, then the vertices in the diametral path form a distance closed set of G. Therefore, a vertex with eccentricity d is taken for getting

a diametral path of G in step 3. In step 6, the set S covers all $N_i(v)$ (i =1 to d – 1). Then clearly D = S ∪ {*w*}, where $w \in N_d(v)$ forms a D.C.D set of G.

COMPLEXITY OF THE ALGORITHM:

Steps 1-3 need $O(p^2)$ time to find a maximum degree vertex *v* with e(*v*) = d. The **if** loop in step 5 is executed in O(p) units of time and all other steps for assigning the given values take constant time. Thus the complexity of the above algorithm is $O(p^2)$.



G

This is a graph with radius 2 and diameter 3. Here starting from the vertex u with e(u) = d=3, the set {u, v, w, y} forms a minimum D.C.D set for G.

GENERAL ALGORITHM TO FIND A D.C.D SET OF A GRAPH G WITH RADIUS r AND DIAMETER d

Input : Connected graph G with radius r and diameter d
Output : D.C.D set of G

Pseudocode:

Step 1:

Set $D_1 = \{v \in V(G) \mid v \text{ is a cut vertex of } G\}$;

Step 2:

For every $u \in V(G)$, find
$e(u) = \max\{d(u, v) \mid v \in V(G)\}$;
$E(u) = \{v \in V(G) \mid d(u, v) = e(u)\}$;
Set $D_2 = \{v \in V(G) : |E(v)| = 1\}$; *// D_2 is the set of all vertices in G having*
// unique eccentric vertices

Step 3:

Set $D = D_1 \cup D_2$;

Step 4:

If (D == V(G)) **then**
Return D; *// G is a 0- distance closed dominating graph.*

Step 5:

Set Q = {$v \in V(G)$ | e(*v*) = r and E(*v*) contains at least one vertex with eccentricity d};

Step 6:

From Q, choose a vertex *v* with maximum degree. Let v_r be the peripheral vertex in $N_r(v)$ and P the r-path $v = v_0, v_1, v_2 \ldots v_r$, where each $v_i \in N_i(v)$, i = 1 to r;
Set D = {*v*, v_1, v_2 … v_r}; *// Store the vertices of the path in D*

Step 7:

Set S = $\{u \in N_1(v) - \{v_1\} \mid u$ is not adjacent to $v_1\}$,
T = $\{u \in N(D) \mid u$ has exclusive domination vertices$\}$,
R = $(V - N[D]) \cup (S \cup T \cup \{v\})$;

Step 8:

For each $v_i \in D$, i = r to 1 **do**
{
Find v_i^* in R such that $v_i^* \in E(v_i)$;
If ($v_i^* \notin D$) **then**
{
Set P_i as the shortest path from v to v_i^* and store its vertices in D_i;
Set $D = D \cup D_i$;
}
}

Step 9:

If (N[D] == V(G)) **then**
Return D;

Step 10:

Set U = V – D;

Step 11:

For i = r-1 to 1 **do**
{

If ($N_{i+1}(v) \cap U \neq \Phi$) **then**
{
Set $S_i = \{u \in N_i(v) \cap U \mid u$ has exclusive domination vertices in $N_{i+1}(v) \cap U\}$;

Step 12:

If ($N(S_i) == N_{i+1}(v) \cap U$) **then**
$D = D \cup S_i$;
Else
{
Choose a maximum degree vertex u_i in $N_i(v) \cap (U - S)$;
Set $S = S \cup \{u_i\}$ and go to step 12;
}
}
Set $U = U - N_{i+1}(v)$, i = i-1;
}

Step 13:

Output D; *// D is the distance closed dominating set of G.*

Step 14:

Exit

VALIDITY OF THE ALGORITHM:

According to Theorem 1.4.2, steps 1-4 check whether the given graph G is a 0-distance closed dominating graph. If not, then any central vertex v with maximum degree in which E(v) has a peripheral vertex v_r is taken as a root vertex. In step 6, an r-path P connecting v to v_r is obtained and its vertices are stored in D. Also, their eccentric vertices (if not in D) are included in D. Hence, the final set D in step 8 gives the distance closed set for any graph G. In steps 9-12, the domination property is checked and the uncovered vertices are added to D to obtain a D.C.D set of G.

COMPLEXITY OF THE ALGORITHM:

1. Steps 1-4, which check whether G is a 0-distance closed dominating graph or not, takes $O(p^2)$ units of time. Forming a set D with the vertices in a path rooted at a central vertex given in step 6 takes O(p) units of time.
2. Step 8 checks the distance closed property in D and it takes $O(p^2)$ units of time. The **for** loop in steps 11-12 checks the domination property of D and it takes O(p) units of time.

Thus the complexity of the above algorithm is $O(p^2)$.

CONCLUSION:

In this paper, algorithms to find a distance closed dominating set of some special classes of graphs are proposed and their complexities are studied. Also, a general algorithm to find the distance closed dominating set of any graph G with a given radius r and diameter d is proposed. The validity and complexity of each of these algorithms are also given and they can be checked in polynomial time. Since the distance closed dominating set is distance preserving, in most of the cases, it is useful to find a sub-network in a given communication network, which is fault tolerant. It is not true that all graphs have at least one distance preserving subset which is a dominating set. Also, every graph has at least one distance closed dominating set even if it does not have a distance preserving dominating set. In those cases, we can cover at most all the vertices using this distance closed domination. Hence, this concept is very useful to analyze the worst case complexity in fault tolerance. Also, the above obtained results are used to analyze the behavior of different communication networks in different situations. In particular, in fault tolerance analysis of networks, parallel architecture designing and signal processing.

REFERENCES:

1. **Bondy, J.A. and U.S.R. Murty** (1976) Graph Theory with Applications. *American Elsevier, New york.*
2. **Buckley, F. and Harary –** Distance in graphs, Addison – Wesley, Redwood City, CA (1990).
3. **Cockayne, E.J. and S.T. Hedetniemi** (1977) Towards a theory of domination in Graphs. *Networks*, **7**, 247 – 261.
4. **Haynes, T.W., S.T. Hedetniemi and P.J. Peter Slater** (1998) Fundamentals of domination in graphs. *Marcel Dekker, New York.*
5. **Haynes, T.W., S.T. Hedetniemi and P.J. Slater** (1998) Domination in Graphs: Advanced Topics. *Marcel Dekker, New York.*
6. **Janakiraman, T.N.** (1991) On Some eccentricity properties of the graph. *Ph.D thesis, Madras University.*
7. **Janakiraman, T.N., P.J.A. Alphonse and V. Sangeetha** (2010) Distance closed domination in graph. *International Journal of Engineering Science Advanced Computing and Bio-Technology*, **1**, 109 – 117.
8. **Ore, O.** (1962) Theory of Graphs. *Amer. Soc. Colloq. Publ., 38,* Amer. Math. Soc., *Providence, RI.*