Minimal Spanning Tree From a Minimum Dominating Set

M. YAMUNA
VIT University
School of advanced sciences
Vellore, Tamilnadu
INDIA
myamuna@vit.ac.in

K. KARTHIKA
VIT University
School of advanced sciences
Vellore, Tamilnadu
INDIA
karthika.k@vit.ac.in

Abstract: In this article, we provide a constructive procedure to generate a spanning tree for any graph from its dominating set, γ - set. We introduce a new kind of minimum dominating set and hence generate a minimum weighted spanning tree from a γ - set for G. We also provide a method for generating a minimum weighted spanning tree using adjacency matrix of G.

Key–Words: Spanning tree, Minimum dominating set, Minimum weighted spanning tree.

1 Introduction

Minimum spanning trees (MSTs) have long been of interest to mathematicians because of their many applications.

1. It has direct applications in the design of computer and communication networks, power and leased-line telephone networks, wiring connections, links in a transportation network, piping in a flow network, etc.

2. It offers a method of solution to other problems to which it applies less directly, such as network reliability, clustering and classification problems.

3. It often occurs as a subproblem in the solution of other problems. For example, minimum spanning tree (MST) algorithms are used in several exact and approximation algorithms for the traveling salesman problem, the multi-terminal flow problem, the matching problem and the capacitated MST problem. [1]

The problem originated in the 1920s when O. Boruvka identified and solved the problem during the electrification of Moravia. However, the language of graph theory is not used to describe the algorithm in his papers from 1926. Boruvka's algorithm picks the next edge by considering the cheapest edge leaving each component of the current forest [7].

In the 1950s, many people contributed to the MST problem. Among them were R. C. Prim and J. B. Kruskal, whose algorithms are very widely used today. Kruskal's algorithm maintain an acyclic spanning subgraph H, enlarging it by edges with low weight to form a spanning tree, by considering edges in non decreasing order of weight, breaking ties arbitrarily [5].

Prims algorithm grows a spanning tree from a given tree from a given vertex of a connected weighted graph G, iteratively adding the cheapest edge from a vertex already reached to a vertex not yet reached, finishing when all the vertices of G have been reached [8]. In 1995, Karger et al., proposed the only known linear expected-time algorithm for the restricted random access computation model. That algorithm is a randomized, recursive algorithm and requires the solution of a related problem, that of verifying whether a given spanning tree is minimum [3]. In 2012, Yajing Wang et al., proposed variant formula of the Wiener index edge-weighted trees of order n. The Wiener index of a graph is the sum of the distances between all pairs of vertices. They have provide the minimum, the second minimum, the third minimum, the maximum, and the second maximum values of the Wiener index. Also, they have characterized the corresponding extremal trees [11].

In this paper we provide two methods for generating a spanning tree and minimum weighted spanning from a graph by using the minimum domination number. In first method, we provide a constructive procedure to generate a spanning tree for any graph from its γ - set. In second method, we provide a procedure for generating a minimum weighted spanning tree by using adjacency matrix.

2 Terminology

A spanning tree of G is a subgraph of G that is a tree containing every vertex of G. An edge in a spanning tree T is called a branch of T. An edge of G that is not in a given spanning tree T is called a chord. Adding any chord to a spanning tree T will create exactly one circuit. Such a circuit, formed by adding chord to a spanning tree, is called a fundamental circuit. A spanning forest of a graph G is a forest that contains every vertex of G such that two vertices are in the same tree of the forest when there is a path in G between these two vertices.

A graph G is said to be a weighted graph if its edges are assigned some weight. A minimum weighted spanning tree in a connected weighted graph is a spanning tree that has the smallest possible sum of weights of its edges. A graph G is maximal with some property P provided that G has property P and no proper supergraph of G has property P. An adjacency matrix of a graph G with n vertices that are assumed to be ordered from v_1 to v_n is defined by,

$$A = [a_{ij}]_{n\times n} = \begin{cases} 1, & \text{if there exist an edge between } v_i \text{ to } v_j; \\ 0, & \text{otherwise.} \end{cases}$$

Adjacency Matrix is also used to represent weighted graphs. If $[a_{ij}] = w$, then there is an edge from vertex v_i to vertex v_j with weight w.For details on graph theory parameter we refer to[2].

A set of vertices D in a graph $G = (V, E)$ is a dominating set if every vertex of $V - D$ is adjacent to some vertex of D. If D has the smallest possible cardinality of any dominating set of G, then D is called a minimum dominating set - abbreviated MDS. The cardinality of any MDS for G is called the domination number of G and it is denoted by $\gamma(G)$. γ - set denotes a dominating set for G with minimum cardinality. A connected dominating set of a graph G is a set D of vertices, if D induces a connected subgraph, then it is called a connected dominating set(CDS). The connected domination number of a graph G is the minimum cardinality of a CDS, denoted by $\gamma_c(G)$.[9]

The open neighborhood of vertex $v \in V(G)$ is denoted by

$$N(v) = \{u \in V(G) | (uv) \in E(G)\}$$

while its closed neighborhood is the set $N[v] = N(v) \cup \{v\}$. The private neighborhood of $v \in D$ is denoted by $pn[v, D]$, is defined by

$$pn[v, D] = N(v) - N(D - \{v\}).$$

If a subgraph H satisfies the added property that for every pair u, v of vertices, $uv \in E(H)$ if and only if $uv \in E(G)$, then H is called an induced subgraph of G and is denoted by $< H >$. We indicate that u is adjacent to v by writing $u \perp v$.For details on domination we refer to [9] and [10].

3 Spanning Tree From a γ - set

In this section, we provide a constructive procedure to generate a spanning tree for any graph from its γ - set. In all the figures encircled vertices denotes a γ - set for G.

Theorem 1 *Given any graph G with n vertices, there is a spanning tree T of G such that $\gamma(G) = \gamma(T)$.*

Proof: Let G be a graph with n vertices and let k be the minimum value for which, $D = \{S_1, S_2, \ldots, S_k\}$ is a γ - set for G such that

1. Each $< S_i >, i = 1, 2, \ldots, k$ is maximal and connected.
2. $V < D >= V(S_1) \cup V(S_2) \ldots, V(S_k)$.
3. $E < D >= E(S_1) \cup E(S_2) \ldots, E(S_k)$.
4. $V(S_1) \cap V(S_2) \cap \ldots \cap V(S_k) = \phi$.
5. $E(S_1) \cap E(S_2) \cap \ldots \cap E(S_k) = \phi$.

Case 1

If $|D| = 1$, then $\gamma(G) = \gamma_c(G)$. Consider $< D >$, where D is a connected dominating set for G. If $< D >$ is not a tree, remove suitable edges from $< D >$ to generate a new subgraph of G say D_1 such that $V(D_1) = V(D)$ and $< D_1 >$ is a tree.

If $< D >$ is a tree, then consider $< D >$ itself and label $< D >$ as $< D_1 >$.

Let $V(D_1) = v_1, v_2, \ldots, v_m$ and let $V(V - D_1) = \{u_1, u_2, \ldots, u_s\}$, where $s + m = n$. Since $< D_1 >$ is connected, for all $v_i \in V(D_1), pn[v_i, D_1] \neq \phi$. For all $u_j \in V - D_1$, either $u_j \in pn[v_i, D_1]$ or $u_j \notin pn[v_i, D_1]$ from some $v_i \in D_1$.

Choose an arbitrary vertex $u_j \in V - D_1, j = 1, 2, \ldots, s$.

If $u_j \in pn[v_i, D_1]$, then construct a new graph D_2 as follows.

• $V(D_2) = V(D_1) \cup \{u_j\}$.

• $E(D_2) = E(D_1) \cup \{u_j v_i\}, i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, s$.

If $u_j \notin pn[v_i, D_1]$, for all $i = 1, 2, \ldots, m$, then u_j is k - dominated, say $u_j \perp v_1, v_2, \ldots, v_k$. In this case, construct D_2 as follows

• $V(D_2) = V(D_1) \cup \{u_j\}$.

• $E(D_2) = E(D_1) \cup \{u_j v_i\}$, where v_i is one of the vertices chosen arbitrarily from $\{v_1, v_2, \ldots, v_k\}$.

In both cases,

• $|V(D_2)| = |V(D_1)| + 1$.

• $|E(D_2)| = |E(D_1)| + 1$.

that is the number of vertices and edges is increased by one. Proceeding this way, for all $u_j \in$

$V - D_1$, we generate a sequence of graphs $D_1, D_2, \ldots$ such that

- Each $< D_i >$ is a tree, $i = 1, 2, \ldots$
- $|V(D_{p+1})| = |V(D_p)| + 1$.
- $|E(D_{p+1})| = |E(D_p)| + 1$.

until we generate a spanning tree for G.

Case 2

If $|D| \geq 2$ and say D has k - components $D = \{S_1, S_2, \ldots, S_k\}$. Verify if each $< S_i >, i = 1, 2, \ldots, k$ is a tree. If not remove an edge from every circuit of each $< S_i >$, so that each $< S_i >$ is a tree.

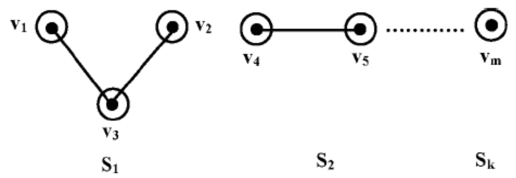

Figure 1: The graph represents the spanning forest that contain vertices in D only.

Let $V(V - D) = \{u_1, u_2, \ldots, u_s\}$. For each $u_j \in V - D$, proceed as in case 1 to generate a spanning forest $X_1 = \{T_1, T_2, \ldots, T_K\}$.

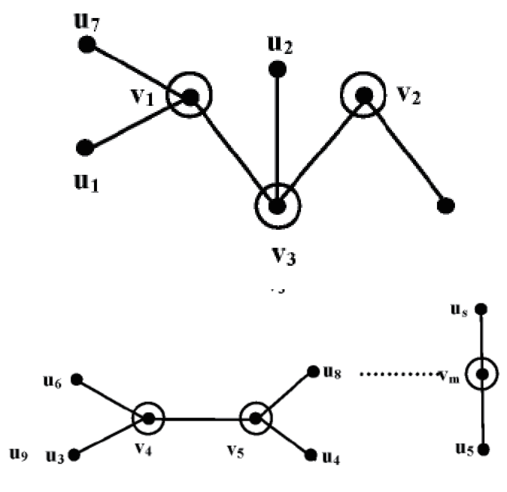

Figure 2: The tree partially constructed by adding edges belonging to $pn[v, D]$.

Let $T_i = \{u_{i1}, u_{i2}, \ldots, u_{imi}\}$, where $m_i = |T_i|$. Since G is a connected graph, for every $T_i \in X_1$, there is at least one $T_j \in X_1$ such that $u_{ip} \perp u_{jq}$, for some $p = 1, 2, \ldots, m_i$ and $q = 1, 2, \ldots, m_j$.

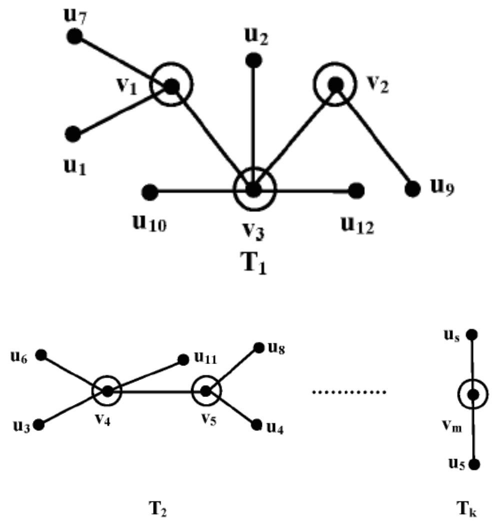

Figure 3: The tree partially constructed by adding edges belonging to non private vertices.

Assume that $T_i, T_j \in X_1$, such that $u_{ip} \perp u_{jq}$ for some $p = 1, 2, \ldots, m_i$ and $q = 1, 2, \ldots, m_j$. Construct a new graph X_2 as follows. $X_2 = \{T_1, T_2, \ldots, T_{i-1}, T_{i+1}, \ldots, T_{j-1}, T_{j+1}, \ldots, T_k\} \cup \{T_i T_j\}$, where

- $|X_2| = |X_1|$ - 1;
- $E(X_2) = E(X_1) \cup (u_{ip} u_{jq})$;
- $V(X_2) = V(X_1)$.

X_2 is also a spanning forest for the connected graph G. We continue this procedure to generate a sequence of trees $X_1, X_2, \ldots$ such that

- Each X_i is a spanning forest for G.
- $|X_{i+1}| = |X_i| - 1$;
- $|V(X_{i+1})| = |V(X_i)|$;
- $|E(X_{i+1})| = |E(X_i)| + 1$.

until we generate a spanning tree for G.

From case 1 and case 2, we conclude that we can generate a spanning tree T of G. D is a dominating set for T also. Moreover $\gamma(T) \geq \gamma(G)$, that is D is a γ- set for T also. Hence $\gamma(G) = \gamma(T)$. □

4 Graph Domination

In this section, we introduce a new kind of minimum dominating set, provide a necessary and sufficient condition for the existence of the set.

Sampathkumar and Kamath[4] define a set $D \subseteq V \cup E$ as a mixed domination set (md - set) if every element not in D is m - dominated by an element of D. The mixed domination number $\gamma_m(G)$ is the minimum cardinality of an md - set. A set $S \subseteq V$ is a ve dominating set (ved - set) if every edge of G is m

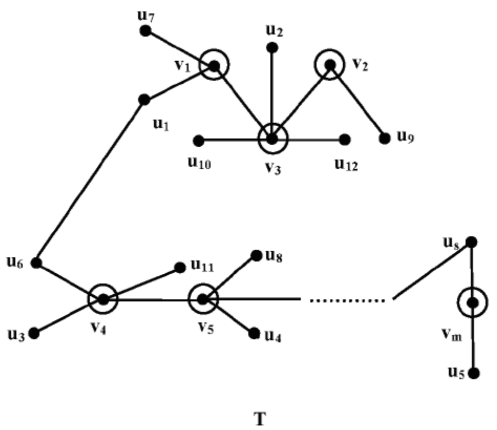

Figure 4: Spanning tree generated from D.

dominated by vertex in S. A ved - set with minimum cardinality is called a γ_{ve} - set.

We introduce a new kind of dominating set called graph domination. A γ- set $D \subseteq V$ is said to graph domination set if D covers all the vertices and edges of G. A γ - set D of G that satisfies this property is denoted by $\gamma_G(G)$.

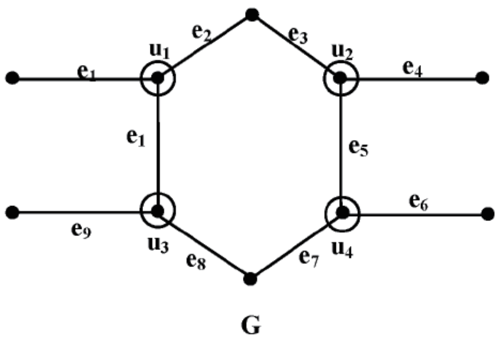

Figure 5: G is Graph domination graph. The γ - set $\{u_1, u_2, u_3, u_4\}$ covers all the edges and vertices of G

Theorem 2 *A γ - set D is a graph domination set if and only if $V - D$ is independent.*

Proof: Let G be any graph and D be a $\gamma_G(G)$ - set for G. Let $V - D = \{u_1, u_2, \ldots, u_m\}$. If $V - D$ is not independent there are some $u_i, u_j \in V - D$ such that $u_i \perp u_j$, that is there is an edge $e = (u_i u_j)$ such that e is not covered by D, a contradiction to the assumption that D is graph domination set.
Conversely assume that $V - D$ is independent. Any edge $e \in E(G)$ has one end vertex in D and other in $V - D$ or both end vertices of e are in D, which implies all the edges in G are covered by D, that is D is a $\gamma_G(G)$ - set. □

Observation 1
Let $v \in V - D$. Then $N(v) \in D$.

Proof: Let $v \in V - D$. If there is one $w \in N(v)$ such that $w \notin D$, then $e = (v, w) \in E(G)$ such that $w, v \in V - D, w \perp v$, a contradiction by theorem [2]. □

Observation 2
$v = pn[u, D]$ if and only if v is pendant.

5 Minimum Weighted Spanning Tree From a γ - set

In the following theorem we provide a method for generating a spanning tree for a weighted graph with n vertices by using a γ - set D, where D is graph domination set.

Theorem 3 *Let G be a weighted connected graph with n vertices and D be a γ - set for G that covers all the edges of G. Then there is a spanning tree T for G such that*

1. *$\gamma(G) = \gamma(T)$.*
2. *T is minimum weighted.*

Proof: Let G be a weighted graph with n vertices and D be a γ - set that covers all the edges of G. By the definition of D,

1. $u, v \in V(G), u \perp v$, at least one of u or v is included in D.
2. $V - D$ is an independent set.

Let $D = \{u_1, u_2, \ldots, u_p\}$ and $V - D = \{v_1, v_2, \ldots, v_q\}$ such that $p + q = n$. We observe that, if $v_j = pn[u_i, D]$, where $i = 1, 2, \ldots, p$ and $j = 1, 2, \ldots, q$ for some $u_i \in D$, then v_j is a pendant vertex. Choose an arbitrary vertex $v_j \in V - D$. Let $e_{ij} = (u_i v_j)$. Consider the vertices in the dominating set D.

We proceed to construct a spanning tree T_1 from D as follows

1. $v_j = pn[u_i, D]$.
   - $V(T_1) = V(T_1) \cup v_j$.
   - $E(T_1) = e_{ij}$.

2. v_j is k - dominated, $k \geq 2$, then v_j is adjacent to k vertices in D say $u_1, u_2, \ldots, u_k$. Label the corresponding k - edges as $e_{j1}, e_{j2}, \ldots, e_{jk}$. In this case, $V(T_1) = V(D) \cup \{v_j\}$ and $E(T_1) = e_{jr}$, where e_{jr} is the edge with minimum weight, $r = 1, 2, \ldots, k$. If there exist more than one edge with the same minimum weight, then we can arbitrarily choose any edge.

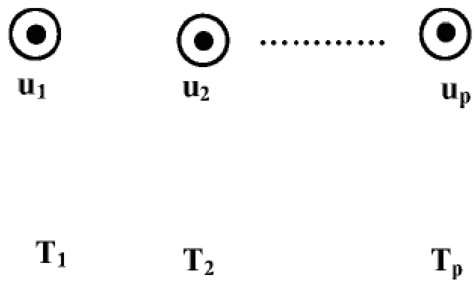

Figure 6: The graph represents the spanning forest that contain vertices in D only.

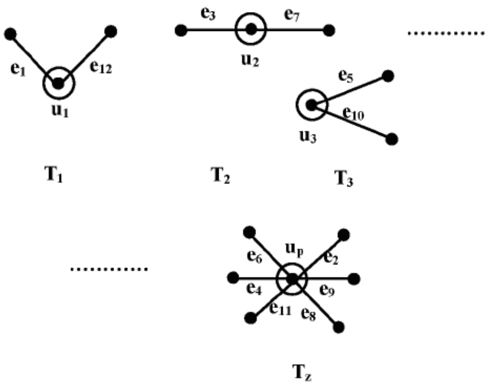

Figure 7: The tree partially constructed by adding minimum weighted edges belonging to $pn[v, D]$.

Continue this procedure for all vertex in $V - D$ to generate a sequence of trees $T_1, T_2, \ldots, T_z$. Let $S_1 = T_1 \cup T_2 \cup \ldots \cup T_z$. $S_1 = (D, V - D)$ is a bipartite graph such that $deg(u_i) \geq 1$, for all $i = 1, 2, \ldots, p$ and $deg(v_j) = 1$, for all $j = 1, 2, \ldots, q$.

1. If S_1 is connected, then S_1 is a spanning tree for G.

2. If S_1 is disconnected, as observed in the proof of the theorem, S_1 is a bipartite graph such that $deg(u_i) \geq 1$, for all $i = 1, 2, \ldots, p$ and $deg(v_j) = 1$, for all $j = 1, 2, \ldots, q$, that is S_1 is a spanning forest. When we add an edge between any two trees T_i and T_j, we do not create any circuit. Since G is connected, there exist at least one $T_i, T_j \in S_1, i \neq j$ such that there is an edge between T_i and T_j. For any $T_i, T_j \in S_1, i \neq j, i, j = 1, 2, \ldots, z$, label the edge between T_i and T_j as e_{ij}. Let $E = \{e_{ij} \mid T_i \perp T_j, i \neq j$, where $i, j = 1, 2, \ldots, z\}$. Arbitrarily choose an edge $e_{xy} \in E$ such that e_{xy} is of minimum weight. Construct a new tree S_2 as follows,
$S_2 = \{T_1, T_2, \ldots, T_{x-1}, T_{x+1}, \ldots, T_{y-1}, T_{y+1}, \ldots, T_z\} \cup \{T_{xy}\}$, where $T_{xy} = T_x \cup T_y \cup e_{xy}$, that is $V(S_2) = V(S_1)$ and $E(S_2) = E(S_2) \cup e_{xy}$.
Continue this process to generate a sequence $S_1, S_2, \ldots$ such that

   (a) $|S_{i+1}| < |S_i|$.
   (b) $|V(S_{i+1})| = |V(S_i)| = V(G)$.
   (c) $|E(S_{i+1})| = |E(S_i)| + 1$.

   until we generate a spanning tree T for G.

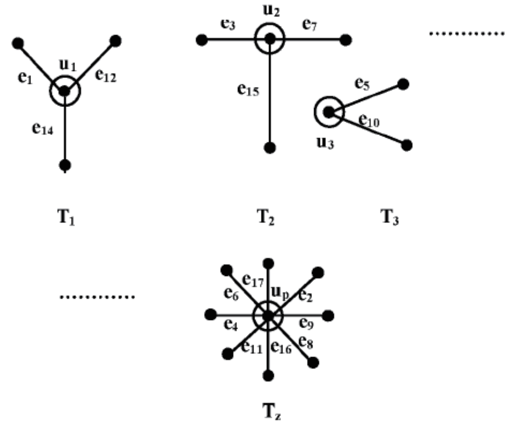

Figure 8: The tree partially constructed by adding minimum weighted edges belonging to non private vertices.

D is a dominating set for T also. Moreover $\gamma(T) \geq \gamma(G)$, that is D is a γ- set for T also. It remains to show that T is a spanning tree with minimum weight. We modify the proof technic for proving that T has minimum weight [6].

When we construct the spanning tree T, we include edges one by one. Let the edges be labeled as $e_1, e_2, \ldots, e_{n-1}$. Let S be a minimum weighted spanning tree of G chosen to have as many edges in common with T as possible. We shall prove that $S = T$ by the method of contradiction. Suppose that $S \neq T$, then T has at least one edge which is not in S. Let e_k be the first edge chosen by the theorem which is in T

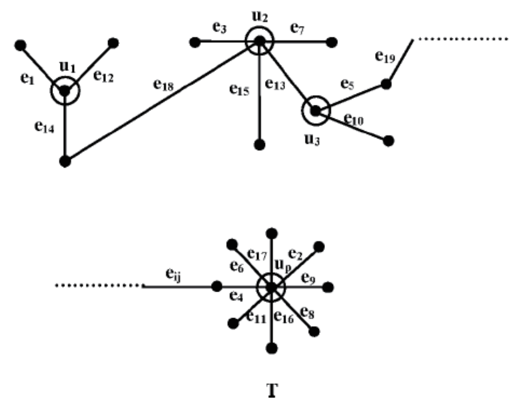

Figure 9: Minimum weighted spanning tree generated from D.

but not in S, that is the edges $e_1, e_2, \ldots, e_{k-1}$ are in S and e_k is the k^{th} - edge such that e_k is not in S.

Case 1 $e_k \in S_1$

$e_k \in S_1$, which implies $e_1, e_2, \ldots, e_{k-1}$ are edges whose one end vertex is in D and the other in $V - D.e_k$ is also an edge to be included such that one end vertex is in D and the other in $V - D$, say $e_k = (uv)$, where $u \in D, v \in V - D$.

Let T_i denote the sub tree created after the addition of the i^{th} edge $e_i, 1 \leq i \leq n-1$. By the theorem one end of e_k is in T_{k-1} and the other is not, that is $u \in T_{k-1}$ and $v \notin T_{k-1}$. Since u and v are in the tree S there is a unique path P in S connecting u and v and P does not involve e_k. Also since P is path from u to v there is an edge e^* such that $e^* = (u^*v)$. Since G is graph domination graph and $v \in V - D, N(v) \in D$, that is there is an edge e^* such that e^* = $(u^*v), u^* \in D$, where $u^* \in D$ and $w(e^*) \geq w(e_k)$, since otherwise, e^* has less weight than e_k and the theorem would have incorporated e^* and not e_k as the kth edge. Now the path P in S together with e_k gives a cycle in G. So, if we replace the edge e^* in S with the edge e_k, we still have a connected subgraph with n vertices and $n-1$ edges. In other words, replacing e^* in S with e_k, gives a new spanning tree R. Since $w(e^*) \geq w(e_k)$, the weight of R is not greater than that of S and so R must a be minimum weighted spanning tree. However R has one more edge in common with T than S has namely the edge e_k, a contradiction to the assumption that S was chosen to be a minimum weighted spanning tree with as many edges in common with T as possible.

Case 2 $e_k \notin S_1$.

Let $e_k \in S_q, 2 \leq q \leq z$. In this case the bipartite graph S_1 is already created, S_1 is a forest and e_k is the first edge chosen by the theorem which is in T but not in S, that is the bipartite graph S_1 is common to S and T.

$e_k \in S_q$ means that, $e_1, e_2, \ldots, e_{k-1}$ are edges in the partially constructed tree T and e_k is the first edge which does not belong to S. Since $e_k \notin S_1$, let $T_r = \{T_1 \cup T_2 \cup \ldots \cup T_{k-1}\} \cup \{k-2$ edges added between the trees $T_1, T_2, \ldots, T_{k-1}\}$. Let $T_s = T_r \cup T_{k+1} \cup \ldots \cup T_z$. T_s is a forest such that $V(T_s) = V(T)$ and $E(T_s) = E(T_r) \cup E(T_{k+1}) \cup \ldots \cup E(T_z)$. $e_k = (uv)$ is an edge with smallest weight such that $u \in T_i, v \in T_j, i \neq j, i, j = r, k+1, \ldots, z$.

Since $e_k \notin S$, adding e_k to S creates a fundamental circuit Γ in S. Not all the edges of Γ belong to T_s [else addition of e_k will not create a fundamental circuit]. Let P be the path from u to v in Γ, not including e_k. P is not completely contained in any $T_i, i = r, k+1, \ldots, z$. Therefore, there exist at least one edge e^* belonging to Γ such that $e^* \neq e_k$ and $w(e^*) \geq w(e_k)$, since otherwise $R = S - \{e\} \cup \{e_k\}$ is a spanning tree for G with weight less than S, a contradiction. R is a connected subgraph with n vertices and $n-1$ edges. Since $w(e^*) \geq w(e_k)$, the weight of R is not greater than that of S and so R must be minimum weighted spanning tree. However R has one more edge in common with T than S has namely the edge e_k. This contradicts the assumption that S was chosen to be a minimum weighted spanning tree with as many edges in common with T as possible.

This contradiction as arisen from the assumption that $S \neq T$. Hence $S = T$.

Hence T is a spanning tree for G such that $\gamma(T) = \gamma(G)$ and T is minimum weighted. □

Example 4 *We have the following graphs*

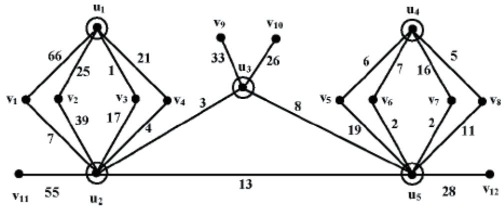

Figure 10: Weighted Graph

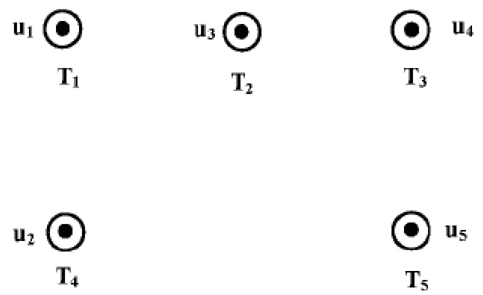

Figure 11: The graph represents the spanning forest that contain vertices in $D = \{u_1, u_2, u_3, u_4, u_5\}$ only.

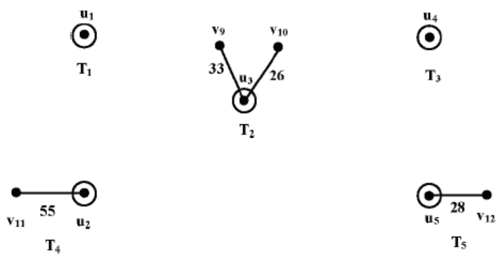

Figure 12: The tree partially constructed by adding minimum weighted edges belonging to $pn[u_2, D]$,$pn[u_3, D]$ and $pn[u_5, D]$, where $pn[u_2, D] = \{v_{11}\}$, $pn[u_5, D] = \{v_{12}\}$, $pn[u_3, D] = \{v_1, v_9\}$.

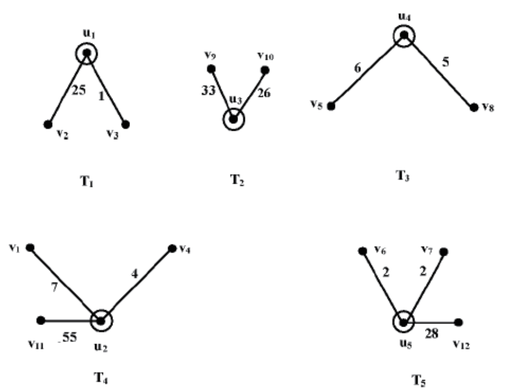

Figure 13: The tree partially constructed by adding minimum weighted edges belonging to non private vertices.

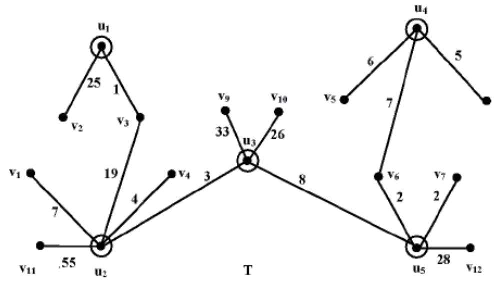

Figure 14: Minimum weighted spanning tree generated from D.The weight of this spanning tree is equal to 229.

6 Adjacency Matrix Using γ - set

Let G be a weighted connected and graph domination graph with n vertices. Let $D = \{u_1, u_2, \ldots, u_k\}$, and $V - D = \{v_1, v_2, \ldots, v_m\}, k + m = n$. Let X be the adjacency matrix of G. For comfort of discussion, let us arrange the rows and columns of X as follows.

1. The first k - rows and columns of X corresponds to the vertices in D.

2. The remaining $n - k$ rows and columns corresponds to the vertices of $V - D$.

3. Define X as follows,

$$X = [x_{ij}]_{n \times n} = \begin{cases} a_{ij}, & \text{if } v_i \perp v_j; \\ 0, & \text{otherwise.} \end{cases}$$

where a_{ij} represents the edge value from v_i to v_j

$$X = \left(\begin{array}{c|c} X_{11} & X_{12} \\ \hline X_{21} & X_{22} \end{array} \right)$$

where X_{11} represents adjacency between vertices in D, X_{12} and X_{21} between vertices in D and $V - D$ and X_{22} between vertices in $V - D$. In a graph domination graph, since $V - D$ is independent, X_{22} is a null matrix.

The first k - rows of X represents the vertices dominated by the corresponding vertex, that is every non - zero entry in the first k - rows represents the number of vertices dominated by each vertex in D. So, the number of non - zero entries in each column of X_{12} represents the number of vertices dominating the corresponding vertex. In other words, a column with exactly one non - zero entry specifies that the corresponding vertex is a private neighborhood, that is it is a pendant vertex.

The following matrix represent the adjacency matrix of G as given in Fig. 10.

$X = [x_{ij}]_{n\times n} =$

$$\left(\begin{array}{ccccc|cccccccccccc}
0 & 0 & 0 & 0 & 0 & 66 & 25 & 1 & 21 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 3 & 0 & 13 & 7 & 39 & 17 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 55 & 0\\
0 & 3 & 0 & 0 & 8 & 26 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 33 & 26 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 & 7 & 16 & 5 & 0 & 0 & 0 & 0\\
0 & 13 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 19 & 2 & 2 & 11 & 0 & 0 & 0 & 28\\
\hline
66 & 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
25 & 39 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
1 & 17 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
21 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 6 & 19 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 7 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 16 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 5 & 11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 33 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 26 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 55 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 28 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right)$$

Step. 1

Chose the smallest entry in each column of X_{12}.

Executing step 1 generates a spanning forest, that is stage 1 and 2 of theorem [3], is executed.

Let $A = [X_{12}]$, that is A is a sub matrix of X of order $k \times n - k$. v_i along with vertices (corresponding to the non - zero entries) picked by step - 1 forms the spanning forest generated at step - 2, of theorem[3].

$$A = \begin{pmatrix}
66 & \mathbf{25} & \mathbf{1} & 21 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
\mathbf{7} & 39 & 17 & \mathbf{4} & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{55} & 0\\
26 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{33} & \mathbf{26} & 0 & 0\\
0 & 0 & 0 & 0 & \mathbf{6} & 7 & 16 & \mathbf{5} & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 19 & \mathbf{2} & \mathbf{2} & 11 & 0 & 0 & 0 & \mathbf{28}
\end{pmatrix}$$

In the above matrix the bold vertices are ones picked by step 1. After executing this matrix we can get the spanning forest as given in Fig. 13.

Any row of A, with zero entry by step - 1 indicates that the corresponding vertex is an isolated vertex in the partially constructed spanning forest $\{T_1, T_2, \ldots, T_z\}, z \leq k$. Let

$$V(T_1) = \{x_{11}, x_{12}, \ldots, x_1p_1\},$$
$$V(T_2) = \{x_{21}, x_{22}, \ldots, x_2p_2\},$$
$$\ldots\ldots$$
$$V(T_z) = \{x_{z1}, x_{z2}, \ldots, x_zp_z\}.$$

Every set of $V(T_i)$ contains at least one element in D, which implies $|V(T_i)| \geq 1$, for all $i = 1, 2, \cdots, z$. Let $Z = \{V(T_1), V(T_2), \ldots, V(T_z)\}$.

Arbitrarily choose any $V(T_i)$ from Z. We mean to say that, we start from the component $V(T_i)$ which contains the vertices $x_{i1}, x_{i2}, \ldots, x_ip_i$. We need to pick the next edge by considering the edge with minimum weight leaving this current component $V(T_i)$, that is we need to pick an edge with smallest weight adjacent to any of the vertices in $X - V(T_i)$. Since we are looking for an edge from $V(T_i)$ to $X - V(T_i)$. Consider the submatrix $X_1 = [x_{ij}]_{n\times n-p_i} = \{X-$ the columns corresponding to the vertices in $V(T_i)\}$. Connecting T_i to an another component T_j by choosing an edge with minimum weight can be enabled by choosing a smallest entry corresponding to the rows $x_{i1}, x_{i2}, \ldots, x_ip_i$ (excluding the entries picked corresponding to step 1).

$$X_1 = \left(\begin{array}{cccc|cccccccccc}
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{66} & \mathbf{21} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\
0 & 3 & 0 & 13 & 7 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 55 & 0\\
3 & 0 & 0 & 8 & 26 & 0 & 0 & 0 & 0 & 0 & 33 & 26 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 6 & 7 & 16 & 5 & 0 & 0 & 0 & 0\\
13 & 8 & 0 & 0 & 0 & 0 & 19 & 2 & 2 & 11 & 0 & 0 & 0 & 28\\
\hline
7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
\mathbf{39} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\
\boldsymbol{17} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\
4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 6 & 19 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 7 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 16 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 5 & 11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 33 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 26 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
55 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 28 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right)$$

In the above matrix, we start from the bold rows $\{u_1, v_2, v_3\}$ and select the smallest entry in these rows. The smallest entry is represent by bold and italic, that is 17.

Step.2

Start from the vertex set $V(T_i)$, that is $x_{i1}, x_{i2}, \ldots, x_{ipi}$. Connect this component to another component in Z (say $V(T_j)$) by choosing a smallest entry in the rows $x_{i1}, x_{i2}, \ldots, x_ip_i$ (excluding the entries picked in step 1) in matrix X_1 that is we choose an edge (e_{ij}) with minimum value that connect two components of the partially constructed spanning tree by step 1.

By applying step 2, we can get the following matrix.

$$X_2 = \left(\begin{array}{ccc|ccccccc}
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\
\boldsymbol{3} & \mathbf{0} & \mathbf{13} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\
0 & 0 & 8 & 0 & 0 & 0 & 0 & 33 & 26 & 0\\
0 & 0 & 0 & 6 & 7 & 16 & 5 & 0 & 0 & 0\\
8 & 0 & 0 & 0 & 19 & 2 & 2 & 11 & 0 & 28\\
\hline
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\
0 & 6 & 19 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 7 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 16 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 5 & 11 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
33 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
26 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\
0 & 0 & 28 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right)$$

In the above matrix, the bold rows represents two forest combined, that is $\{u_1, v_2, v_3\}$ and $\{u_2, v_1, v_4, v_{11}\}$. The smallest entry is represented by bold and italic, that is 3.

Once step 2 is executed the number of components of the spanning forest is reduced by one. Let $Z_1 = \{V(T_1), V(T_2), \ldots, V(T_{i-1}), V(T_{i+1}), V(T_{j-1}), V(T_{j+1}), \ldots, V(T_z)\} \cup \{V(T_iT_j)\}$, where $V(T_iT_j) = V(T_i) \cup V(T_j) \cup \{e_{ij}\}$, that is Z_1 is the newly constructed spanning forest after executing step 2 once.

Step.3

Choose an arbitrary vertex set from Z_1 and continue as in step - 2 to generate Z_2 and the corresponding matrix X_2.

$$X_3 = \left(\begin{array}{cc|ccccc}
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{13} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \boldsymbol{8} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
0 & 0 & 6 & 7 & 16 & 5 & 0 \\
0 & 0 & 19 & 2 & 2 & 11 & 28 \\
\hline
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
6 & 19 & 0 & 0 & 0 & 0 & 0 \\
7 & 2 & 0 & 0 & 0 & 0 & 0 \\
16 & 2 & 0 & 0 & 0 & 0 & 0 \\
5 & 11 & 0 & 0 & 0 & 0 & 0 \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
0 & 28 & 0 & 0 & 0 & 0 & 0
\end{array}\right)$$

In the above matrix, the bold rows represents two forest combined, that is $\{u_1, u_2, v_1, v_2, v_3, v_4, v_{11}\}$ and $\{u_3, v_9, v_{10}\}$. The smallest entry is represented by bold and italic, that is 8.

Step.4

Continue the process to generate $Z_1, Z_2, \ldots$ and a sequence of matrices $X_1, X_2, \ldots$ until all n vertices have been connected by $n-1$ edges.

$$X_4 = \left(\begin{array}{c|cc}
\mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} \\
0 & 6 & 5 \\
\mathbf{0} & \mathbf{19} & \mathbf{11} \\
\hline
\mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} \\
6 & 0 & 0 \\
\boldsymbol{7} & \mathbf{0} & \mathbf{0} \\
\mathbf{16} & \mathbf{0} & \mathbf{0} \\
5 & 0 & 0 \\
\mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0}
\end{array}\right)$$

In the above matrix, the bold rows represents two forest combined, that is $\{u_1, u_2, u_3, v_1, v_2, v_3, v_4, v_9, v_{10}, v_{11}\}$ and $\{u_5, v_6, v_7, v_{12}\}$. The smallest entry is represented by bold and italic, that is 7.

After executing this matrix we can get a minimum weighted spanning tree T.

$$\left(\begin{array}{ccccc|cccccccccccc}
0 & 0 & 0 & 0 & 0 & 66 & \boldsymbol{25} & \boldsymbol{1} & 21 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \boldsymbol{3} & 0 & 13 & \boldsymbol{7} & 39 & 17 & \boldsymbol{4} & 0 & 0 & 0 & 0 & 0 & 0 & \boldsymbol{55} & 0 \\
0 & 3 & 0 & 0 & \boldsymbol{8} & 26 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \boldsymbol{33} & \boldsymbol{26} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \boldsymbol{6} & 7 & 16 & \boldsymbol{5} & 0 & 0 & 0 & 0 \\
0 & 13 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 19 & \boldsymbol{2} & \boldsymbol{2} & 11 & 0 & 0 & 0 & \mathit{28} \\
\hline
66 & 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
25 & 39 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & \boldsymbol{17} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
21 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 6 & 19 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \boldsymbol{7} & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 16 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 5 & 11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 33 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 26 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 55 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 28 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right)$$

In the above matrix, the smallest entries are represented by bold and italic, that is 7, 25, 1, 4, 6, 2, 2, 5, 33, 26, 55, 28, 17, 3, 8 and 7. The weight of this spanning tree is equal to 229. This weight is equal to the weight of spanning tree as given in Fig. 14.

7 Conclusion

This paper provides a new method of generating a minimum weighted spanning tree from a γ - set. For each distinct γ - set we can generate a minimum weighted spanning tree. So the maximum possible distinct spanning trees that can be generated is equal to the number of distinct γ - set possible for any graph domination graph. This method can be adopted to find minimum weighted spanning trees for graph domination graph and to generate graph domination trees.

References:

[1] C. F. Bazlamaccl, K. S. Hindi, Minimum - weight spanning tree algorithms a survey and empirical study, *Computer & Operations Research*, Vol.28, 2001, pp.767-785.

[2] D. B. West, *Introduction to graph theory*, second ed., Prentice - Hall, New Jersey (2007).

[3] D. R. Karger et. al., A randomized linear-time algorithm to find minimum spanning trees, *Journal of the Association for Computing Machinery*, Vol.42, 1995, issue 2: pp.321-328.

[4] E. Sampathkumar and S. S. Kamath, Mixed domination in graphs, *Sankhya*, Vol.54, 1992, pp.399-402.

[5] J. B. Kruskal, On the shortest spanning subtree of a graph and the traveling salesman problem, *Proc. Am. Math. Soc.*, Vol.7, 1956, pp.48-50.

[6] J. Clark, D. A. Holtan, *A First look at Graph Theory*, World Scientific Publishing Co. Pte Ltd, (1991).

[7] J. Nesetril et. al, Otakar Boruvka on minimum spanning tree problem Translation of both the 1926 papers, comments, history, *Discrete Mathematics*, Vol.233, 2001, pp.3-36.

[8] R. C. Prim, Shortest connection networks and some generalizations, *Bell Syst. Tech. J.*, Vol.36, 1957, pp.1389-1401.

[9] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater, *Fundamentals of Domination in Graphs*, Marcel Dekker, New York, (1998).

[10] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater, *Domination in Graphs - Advanced Topics*, Marcel Dekker, New York, (1998).

[11] Yajing Wang et al., The wiener index for weighted trees, *WSEAS Transactions on Mathematics*, Vol.11, 2012, Issue 12, pp.1127-1136.