Every Tree is a subtree of Graceful Tree, Graceful Graph and Alpha-labeled Graph

G. Sethuraman* and P. Ragukumar
Department of Mathematics
Anna University
Chennai 600 025, India
sethu@annauniv.edu

Abstract

A function f is a called graceful labeling of a graph G with m edges, if f is an injective function from $V(G)$ to $\{0, 1, 2, \cdots, m\}$ such that when every edge uv is assigned the edge label $|f(u) - f(v)|$, then the resulting edge labels are distinct. A graph which admits graceful labeling is called a graceful graph. A graceful labeling of a graph G with m edges is called an α-labeling if there exists a number λ such that for any edge uv, $min[f(u), f(v)] \leq \lambda < max[f(u), f(v)]$. The Characterization of graceful graphs appears to be a very difficult problem in Graph Theory. In this paper, we prove a basic structural property of graceful graphs, that every tree is a subtree of a graceful graph, an α-labeled graph and a graceful tree, and we discuss a related open problem towards settling the popular Graceful Tree Conjecture.

Mathematics Subject Classification: 05C78;05C05
Keywords: Graceful Tree;Graceful Tree Conjecture;Graceful Tree Embedding;Graceful Labeling;Graph Labeling

1 Introduction

All the graphs considered in this paper are finite and simple. The terms which are not defined here can be referred from [21]. In 1963, Ringel posed his celebrated conjecture, popularly called the Ringel Conjecture [14],

*Corresponding Author

which states that, K_{2n+1}, the complete graph on $2n+1$ vertices can be decomposed into $2n+1$ isomorphic copies of a given tree with n edges. In [10], Kotzig independently conjectured the specialized version of the Ringel Conjecture, that the complete graph K_{2n+1} can be cyclically decomposed into $2n+1$ copies of a given tree with n edges. In an attempt to solve both the Ringel and Kotzig Conjectures, in 1967, Rosa, in his classical paper [15] introduced a hierarchical series of labelings called σ, ρ, β and α labelings as a tool to attack both the Ringel and Kotzig Conjectures. Later, the β-labeling was called graceful labeling by Golomb, and now this term is being widely used. A function f is called a graceful labeling of a graph G with m edges, if f is an injective function from $V(G)$ to $\{0,1,2,\cdots,m\}$ such that, when every edge uv is assigned the edge label $|f(u)-f(v)|$, then the resulting edge labels are distinct. A graph which admits graceful labeling is called graceful graph. A graceful labeling of a graph G with m edges is called α-labeling if there exists a number λ such that for any edge uv, $min[f(u),f(v)] \leq \lambda < max[f(u),f(v)]$. The structural characterization of graceful graphs is one of the most difficult problems in graph theory. In fact, very few results [1–5,8] deal with necesary conditions of graceful graphs, and numerous classes of graceful graphs [6, 9, 11–13, 16, 17, 19, 20] with various special structural properties were identified. However, these results have provided only very little insight about the structural characterization of the graceful graphs. The popular graceful tree conjecture: All trees are graceful, remains open over four decades. In this paper, we provide a basic structural understanding of graceful graphs that every tree is a subtree of a graceful graph, an α-labeled graph and a graceful tree.

2 Main Result

In this section, first we present the Labeling Algorithm, which generates distinct labels on the vertices and edges of a given arbitrary input tree T with m edges. Using the Labeling Algorithm, we present a Graceful Tree Embedding Algorithm, that will generate a graceful tree T^* containing the given input arbitrary tree T as its subtree.

Labeling Algorithm

Input: Arbitrary tree T with m edges

Step 1: Initialization

Identify a longest path P of T. Let d be the length of the longest path P. Let u_0 be the origin of P. Consider the tree

T as a rooted tree having u_0 as its root. Then, T has $d+1$ levels. Arrange the vertices in every level of the rooted tree T in such a way that each vertex of the longest path P always appears as the left most vertex of the respective level of the vertex. Further, arrange the remaining vertices of each level in such a way that the edges of T do not cross each other.
For each r, $1 \le r \le \lfloor \frac{d}{2} \rfloor$, describe the vertices of the $(2r)$th-even level as $u_{2r,1}, u_{2r,2}, \cdots, u_{2r,\gamma_{2r}}$ from left to right such that $u_{2r,1}$ is the vertex of the longest path P in the $(2r)$th level, where γ_{2r} denotes the number of vertices of the $(2r)$th level.
For each r, $1 \le r \le \lceil \frac{d}{2} \rceil$, describe the vertices of the $(2r-1)$th-odd level as $v_{2r-1,1}, v_{2r-1,2}, \cdots, v_{2r-1,\gamma_{2r-1}}$ from left to right such that $v_{2r-1,1}$ is the vertex of the longest path P in the $(2r-1)$th level, where γ_{2r-1} denotes the number of vertices of the $(2r-1)$th level.

Step 2: Labeling Vertices
Step 2.1: Labeling vertices in each even level

$l(u_{0,1}) = l(u_{0,\gamma_0}) = l(u_0) = 0$
For each r, $1 \le r \le \lfloor \frac{d}{2} \rfloor$, define the labels of the vertices of the $2r$th level,
$l(u_{2r,1}) = l(u_{2(r-1),\gamma_{2(r-1)}}) + 1$
$l(u_{2r,i}) = l(u_{2r,i-1}) + 1, 2 \le i \le \gamma_{2r}$

Step 2.2: Assigning weights to the vertices of each odd level

For each r, $1 \le r \le \lceil \frac{d}{2} \rceil$, define the weights of vertices of $(2r-1)$th level, for $1 \le i \le \gamma_{2r-1}$,
$$wt(v_{2r-1,i}) = \begin{cases} w \ge 0, & \text{if } deg(v_{2r-1,i}) = 1 \\ w \ge [l(RMC(v_{2r-1,i})) - l(PR(v_{2r-1,i}))], & \text{if } deg(v_{2r-1,i}) > 1 \end{cases}$$
where $RMC(v_{2r-1,i})$ denotes the right most child of $v_{2r-1,i}$ and $PR(v_{2r-1,i})$ denotes the parent of $v_{2r-1,i}$.
If d is even, then define the labels of the vertices of the last odd level, $(d-1)$th level,
$l(v_{d-1,\gamma_{d-1}}) = l(u_{d,\gamma_d} + wt(v_{d-1,\gamma_{d-1}}) + 1$,
$l(v_{d-1,\gamma_{d-1}-i}) = l(v_{d-1,\gamma_{d-1}-(i+1)}) + wt(v_{d-1,\gamma_{d-1}-i}) + 1$, for $1 \le i \le \gamma_{d-1} - 1$
If d is odd, then define the labels of the vertices of the last odd level, dth level,
$l(v_{d,\gamma_d}) = l(u_{d-1,\gamma_{d-1}}) + wt(v_{d,\gamma_d}) + 1$,
$l(v_{d,\gamma_d-i}) = l(v_{d,\gamma_d-(i-1)}) + wt(v_{d,\gamma_d-i}) + 1$, for $1 \le i \le \gamma_d - 1$

Step 2.3: Labeling of vertices in each odd level

For each r, $0 \leq r \leq \lceil \frac{d}{2} \rceil - 2$, define the labels of the vertices of the $(2r+1)$th odd level,
$l(v_{2r+1,\gamma_{2r+1}}) = l(v_{2r-1,\gamma_{2r-1}}) + wt(v_{2r+1,\gamma_{2r+1}}) + 1$,
$l(v_{2r+1,\gamma_{2r+1}-i}) = l(v_{2r+1,\gamma_{2r+1}-(i-1)}) + wt(v_{2r+1,\gamma_{2r+1}-i}) + 1$,
for $1 \leq i \leq \gamma_{2r+1} - 1$

Step 3: Edge labels of the edges of T

For every edge uv of T, define the edge label
$l'(uv) = |l(u) - l(v)|$.

Observation 1: Vertex labels defined in Labeling Algorithm are distinct.

When d is odd, observe from Step 2 of the Labeling Algorithm, that if the labels of the vertices in all the even levels of T and the labels of the vertices in all the odd levels of T are arranged in a sequence as,
$l(u_{0,1})$,
$l(u_{2,1}), l(u_{2,2}), \cdots, l(u_{2,\gamma_2})$,
$l(u_{4,1}), l(u_{4,2}), \cdots, l(u_{4,\gamma_4})$,
$\vdots$
$l(u_{d-1,1}), l(u_{d-1,2}), \cdots, l(u_{d-1,\gamma_{d-1}})$,
$l(v_{d,\gamma_d}), (v_{d,\gamma_d-1}), \cdots, (v_{d,1})$,
$l(v_{d-2,\gamma_d}), (v_{d-2,\gamma_{d-2}-1}), \cdots, (v_{d-2,1})$,
$\vdots$
$l(v_{3,\gamma_3}), (v_{3,\gamma_3-1}), \cdots, (v_{3,1})$,
$l(v_{1,\gamma_1}), (v_{1,\gamma_1-1}), \cdots, (v_{1,1})$.
Then this sequence forms a monotonically increasing sequence.
When d is even, observe from Step 2 of the Labeling Algorithm, that if the labels of the vertices in all the even levels of T and the labels of the vertices in all the odd levels of T are arranged in a sequence as
$l(u_{0,1})$,
$l(u_{2,1}), l(u_{2,2}), \cdots, l(u_{2,\gamma_2})$,
$l(u_{4,1}), l(u_{4,2}), \cdots, l(u_{4,\gamma_4})$,
$\vdots$
$l(u_{d,1}), l(u_{d,2}), \cdots, l(u_{d,\gamma_d})$,
$l(v_{d-1,\gamma_{d-1}}), (v_{d-1,\gamma_{d-1}-1}), \cdots, (v_{d-1,1})$,
$l(v_{d-3,\gamma_{d-3}}), (v_{d-3,\gamma_{d-3}-1}), \cdots, (v_{d-3,1})$,
$\vdots$
$l(v_{3,\gamma_3}), (v_{3,\gamma_3-1}), \cdots, (v_{3,1})$,
$l(v_{1,\gamma_1}), (v_{1,\gamma_1-1}), \cdots, (v_{1,1})$.

Then this sequence forms a monotonically increasing sequence.
Thus, the vertex labels of all the vertices of T are distinct.

Theorem 1. *The edge labels of all the edges of T that defined in the Labeling Algorithm are distinct.*

Proof. By Observation 1, the vertex labels of vertices of T are distinct. Thus, the edge labels of the edges incident at each vertex $v_{2r-1,i}$ for i, $1 \le i \le \gamma_{2r-1}$, of $(2r-1)$th odd level, $1 \le r \le \lceil \frac{d}{2} \rceil$, are distinct.
Claim: The edge labels of all the edges incident at all the vertices in every $(2r-1)$th odd level of T are distinct, for $1 \le r \le \lceil \frac{d}{2} \rceil$.
Consider two (consecutive) vertices $v_{2r-1,(i-1)}$ and $v_{2r-1,i}$ in the $(2r-1)$th odd level of T. Let $y_{i-1} = PR(v_{2r-1,i-1})$ and $y_i = PR(v_{2r-1,i})$. Let $x_{i-1} = RMC(v_{2r-1,i-1})$ and $x_i = RMC(v_{2r-1,i})$. By the vertex labeling, for any vertex $v_{2r-1,i}$ in the $(2r-1)$th odd level, $l(y_i) < l(w)$, for any child w of $v_{2r-1,i}$ and $l(w) < l(x_i)$, for any child other than x_i. That is, if $deg(v_{2r-1,i}) > 1$, $min\{l(N(v_{2r-1,i}))\} = l(y_i)$ and $max\{l(N(v_{2r-1,i}))\} = l(x_i)$. If $deg(v_{2r-1,i}) = 1$, $min\{l(N(v_{2r-1,i}))\} = l(y_i) = max\{l(N(v_{2r-1,i}))\}$. Here, $N(v_{2r-1,i})$ denotes the set of adjacent vertices of $v_{2r-1,i}$ and $l(N(v_{2r-1,i}))$ denotes the set labels of the adjacent vertices of $v_{2r-1,i}$. Thus,
$min\{l'(v_{2r-1,i}w) : w \in N(v_{2r-1,i})\} = l'(v_{2r-1,i}x_i)$ and
$max\{l'(v_{2r-1,i}w) : w \in N(v_{2r-1,i})\} = l'(v_{2r-1,i}y_i)$.
Case 1: $deg(v_{2r-1,i-1}) = 1$
Then, $wt(v_{2r-1,i-1}) = w \ge 0$, we have $l(v_{2r-1,i-1}) = l(v_{2r-1,i}) + w + 1$.
Claim 1: $l'(v_{2r-1,i-1}y_{i-1}) > l'(v_{2r-1,i}y_i)$.
Subcase 1.1: $y_{i-1} = PR(v_{2r-1,i-1}) \ne PR(v_{2r-1,i}) = y_i$
Let k_1 denote the number of vertices between the vertices y_{i-1} and y_i. Then, $l(y_i) = l(y_{i-1}) + k_1 + 1$, $k_1 \ge 0$. We have, $l'(v_{2r-1,i-1}y_{i-1}) - l'(v_{2r-1,i}y_i) = l(v_{2r-1,i-1}) - l(y_{i-1}) - l(v_{2r-1,i}) + l(y_i) = l(v_{2r-1,i}) + w + 1 - l(y_{i-1}) - l(v_{2r-1,i}) + l(y_{i-1}) + k_1 + 1 = w + k_1 + 2 > 0$. Thus, $l'(v_{2r-1,i-1}y_{i-1}) > l'(v_{2r-1,i}y_i)$.
Subcase 1.2: $y_{i-1} = PR(v_{2r-1,i-1}) = PR(v_{2r-1,i}) = y_i$
Then, $l'(v_{2r-1,i-1}y_{i-1}) - l'(v_{2r-1,i}y_i) = l(v_{2r-1,i-1}) - l(y_{i-1}) - l(v_{2r-1,i}) + l(y_{i-1}) = l(v_{2r-1,i}) + w + 1 - l(v_{2r-1,i}) = w + 1 > 0$. Hence Claim 1.
Case 2: $deg(v_{2r-1,i-1}) > 1$
Claim 2: $l'(v_{2r-1,i-1}x_{i-1}) > l'(v_{2r-1,i}y_i)$.
Subcase 2.1: $y_{i-1} = PR(v_{2r-1,i-1}) \ne PR(v_{2r-1,i}) = y_i$
Since $deg(v_{2r-1,i-1}) > 1$, then by the Labeling Algorithm, we have, $wt(v_{2r-1}, i-1) = w \ge [RMC(v_{2r-1,i-1}) - PR(v_{2r-1,i-1})]$ and $l(v_{2r-1,i-1}) = l(v_{2r-1,i}) + w + 1$. Let k_1 denote the number of vertices between the vertices y_{i-1} and y_i. Let k_2 denote the number of vertices to the right side of the vertex y_i in the preceeding level, the $(2r-2)$th even level. Let k_3 denote the number of vertices to the left side of the vertex x_{i-1} in the succeeding level, the $(2r)$th even level. Note that $k_1 \ge 0$, $k_2 \ge 0$ and $k_3 \ge 0$. Then,

$l(x_{i-1}) = l(y_i) + k_2 + k_3 + 1$. Therefore,
$l(v_{2r-1,i}) + w + 1 - l(y_i) - k_2 - k_3 - 1 - l(v_{2r-1,i}) + l(y_i) = w - (k_2 + k_3) > 0$.
Note that $w = k_1 + k_2 + k_3 + 2$.

Subcase 2.2: $y_{i-1} = PR(v_{2r-1,i-1}) = PR(v_{2r-1,i}) = y_i$
We have $l(v_{2r-1,i}) + w + 1 - l(y_i) - k_2 - k_3 - 1 - l(v_{2r-1,i}) + l(y_i) = w - (k_2 + k_3) > 0$. Note that $w = k_2 + k_3 + 1$.
Hence Claim 2.

Thus, the edge labels of the edges incident at any two consecutive vertices in every odd level $(2r-1)$th level are distinct, $1 \leq r \leq \lfloor \frac{d}{2} \rfloor$.

Let $l'_{min}(h)$ denote the minimum of all the edge labels of the edges incident at the vertices in the level h. Let $l'_{max}(h)$ denote the maximum of all the edge labels of the edges incident at the vertices in the level h.

Claim 3: $l'_{min}(2r-1) > l'_{max}(2r+1)$, $1 \leq r \leq \lfloor \frac{d}{2} \rfloor$
Observe that $l'_{max}(2r+1)$, the maximum of all the edge labels of the edges incident at the vertices in the level $2r+1$ is obtained from the edge $v_{2r+1,1}u_{2r,1}$.

Case a: $deg(v_{2r-1,\gamma_{2r-1}}) = 1$
Let the parent of the vertex $v_{2r-1,\gamma_{2r-1}}$ be $y_{\gamma_{2r-1}}$. Since the degree of the vertex $v_{2r-1,\gamma_{2r-1}}$ is 1, observe that $l'_{min}(h)$, the minimum of all the edge labels of the edges incident at the vertices in the level $(2r-1)$ is obtained from the edge label $l'(v_{2r-1,\gamma_{2r-1}}y_{\gamma_{2r-1}})$. Also, observe that $l'_{max}(2r+1) = l'(v_{2r+1,1}u_{2r,1})$. Let k_1 denote the number of vertices to the right of the vertex $y_{\gamma_{2r-1}}$ in the even level $(2r-2)$. Note that $k_1 \geq 0$. Then, by Labeling Algorithm, we have $l(u_{2r,1}) = l(y_{\gamma 2r-1}) + k_1 + 1$. Since $deg(v_{2r-1,\gamma_{2r-1}}) = 1$, we have $wt(v_{2r-1,\gamma_{2r-1}}) = w \geq 0$. Therefore, $l(v_{2r-1,\gamma_{2r-1}}) = l(v_{2r+1,1}) + w + 1$. Thus,
$l(v_{2r+1,1}) + w + 1 - l(y_{\gamma_{2r-1}}) - l(v_{2r+1,1}) + l(y_{\gamma_{2r-1}}) + k_1 + 1 = w + k_1 + 2 > 0$.

Case b: $deg(v_{2r-1,\gamma_{2r-1}}) > 1$
Let the right most child of the vertex $v_{2r-1,\gamma_{2r-1}}$ be $x_{\gamma_{2r-1}}$. Since the degree of the vertex $v_{2r-1,\gamma_{2r-1}} > 1$, observe that $l'_{min}(h)$, the minimum of all the edge labels of the edges incident to the vertices in the level $(2r-1)$ is obtained from the edge label $l'(v_{2r-1,\gamma_{2r-1}}x_{\gamma_{2r-1}})$. Also, observe that $l'_{max}(2r+1) = l'(v_{2r+1,1}u_{2r,1})$. Let k_2 denote the number of vertices between the vertices $u_{2r,1}$ and $x_{\gamma_{2r-1}}$ in the even level $(2r)$. Note that $k_2 \geq 0$. Then, by Labeling Algorithm, we have $l(x_{\gamma 2r-1}) = l(u_{2r,1}) + k_2 + 1$. Since $deg(v_{2r-1,\gamma_{2r-1}}) > 1$, we have $wt(v_{2r-1,\gamma_{2r-1}}) = w \geq [RMC(v_{2r-1,\gamma_{2r-1}}) - PR(v_{2r-1,\gamma_{2r-1}})]$. Observe that $w > k_2$. Therefore, $l(v_{2r-1,\gamma_{2r-1}}) = l(v_{2r+1,1} + w + 1$. Thus,
$l(v_{2r+1,1}) + w + 1 - l(u_{2r,1}) - k_2 - 1 - l(v_{2r+1,1}) + l(u_{2r,1}) = w - k_2 > 0$.
Hence Claim 3.

For each r, $1 \leq r \leq \lceil \frac{d}{2} \rceil$, the edge labels of the edges incident with vertices v_i, $1 \leq i \leq \gamma_{2r-1}$ at the $(2r-1)$th odd level are strictly increasing as the index i decreases from γ_{2r-1} to 1. Consequently, the edge labels of the edges incident at the vertices of any odd level $(2r-1)$ strictly increases as r decreases from $\lceil \frac{d}{2} \rceil$ to 1. Thus, the edge labels of all the edges incident at all the vertices in every $(2r-1)$th odd level of T are distinct, for $1 \leq r \leq \lceil \frac{d}{2} \rceil$. Hence edge labels of all the edges of T are distinct. □

Note 1: For a given arbitrary input tree T with m edges, we run the Labeling Algorithm and obtain the labeled output tree T'. For convenience, hereafter the vertices of the output tree T' are referred to by their vertex labels, and the edges are referred to by their edge labels. Thus, for the output tree T' we consider $V(T') = \{0, 1, 2, \cdots, p-1, p, \alpha_1, \alpha_2, \cdots, M\}$ and $E(T') = \{l'(e_1), l'(e_2), \cdots, l'(e_m)\}$.

Graceful Tree Embedding Algorithm

Input: Any arbitrary tree T

Step 1:
Step 1.1:

Run Labeling Algorithm on input tree T and get the output tree T'.

Step 1.2:

For the tree T', define
Vertex Label Set
$V = V(T') = \{0, 1, 2, \cdots, p-1, p, \alpha_1, \alpha_2, \cdots, \alpha_{q-1} = M\}$,

where the elements of V are the vertex labels of the vertices of the input tree T that defined in the Labeling Algorithm,

Edge Label Set $E = E(T') = \{l'(e_1), l'(e_2), \cdots, l'(e_m)\}$,

where $l'(e_i)$, is the edge label of the edge e_i, for $1 \leq i \leq m$ of T defined in the Labeling Algorithm,

All label set $X = \{0, 1, 2, \cdots, M\}$,
Common label set $I = V \cap E$,
Exclusive vertex label set $\hat{V} = (V - \{0\}) - I$,
Exclusive edge label set $\hat{E} = E - I$ and
Missing vertex label set $\hat{X} = X - V$.
Initiate $T^* \leftarrow T'$,
$V(T^*) \leftarrow V(T')$,
$E(T^*) \leftarrow E(T')$.

Step 2:

While $\hat{X} \neq \phi$, find $min\hat{X} = a$.

Step 3:

If $a \notin \hat{E}$, then consider a new vertex with label a and add a new edge between the vertex with label 0 and the new vertex with label a to T^*.
Update $T^* \leftarrow T^* + (0, a)$,
$V(T^*) \leftarrow V(T^*) \cup \{a\}$,
$E(T^*) \leftarrow E(T^*) \cup \{(0, a)\}$.
Delete a from $\hat{X}$ and go to Step 2.

Step 4:

If $a \in \hat{E}$, then find $min\hat{V} = b$ and find $\beta = a - b$. Consider a new vertex with label a and add a new edge between the vertex labeled β and the new vertex with label a to T^*.
Update $T^* \leftarrow T^* + (\beta, a)$,
$V(T^*) \leftarrow V(T^*) \cup \{a\}$,
$E(T^*) \leftarrow E(T^*) \cup \{(\beta, a)\}$.
Delete a from $\hat{X}$ and delete b from $\hat{V}$ and go to Step 2.

Observation 2: If $\hat{X} \neq \phi$, the Graceful Tree Embedding Algorithm executes Step 2 and $min\hat{X} = a$ is found. This means that there always exist a number of vertices in the current tree T^* with labels $0, 1, 2, \cdots, a-1$. Thus, after the execution of Step 3 or Step 4, the latest updated tree T^* will have $a+1$ vertices with vertex labels $0, 1, 2, \cdots, a-1, a$.

Lemma 1. *The label β defined in Step 4 of Graceful Tree Embedding Algorithm is always a positive integer and it exists as vertex label of a vertex of the current tree T^* that being used in that execution of Step 4.*

Proof. For the convenience, we consider the input tree T as a bipartite graph with bipartition (V_1, V_2), where V_1 consists of all the even level vertices and V_2 consists of all the odd level vertices. We arrange the vertices of V_1 in the increasing order of the label of the vertices positioned in the top to bottom and also we arrange the vertices of V_2 in the decreasing order of the labels of the vertices positioned in the top to bottom and to the right side of the vertices of V_1. V_1 is refered as left side partition and V_2 is refered as right side partition.
Observe that Step 4 of Graceful Tree Embedding Algorithm is executed when $\hat{X} \neq \phi$. Further $a = min\hat{X}$, $b = min\hat{V}$ and $\beta = a - b$ are found in the Step 4 We claim that β is a positive integer. For that we prove $a > b$. Suppose not. Then $a \leq b$. Since $a \in \hat{X}, b \in \hat{V}$ and $\hat{X} \cap \hat{V} = \phi, a \neq b$. Thus,

$a < b$. Since $min\hat{X} > p$, where $p = |V_1|$, and (V_1, V_2) is the bipartition of the vertex set of the input tree T. Therefore, $a > p$. This implies that $b > p$. Then the labels $1, 2, 3, \cdots, p-1$ do not exist as exclusive vertex labels in T since b is the minimum over V^*. This implies that either the labels $1, 2, 3, \cdots, p-1$ do not exist as vertex labels or else, if the labels $1, 2, 3, \cdots, p-1$ exist as vertex labels then the labels $1, 2, 3, \cdots, p-1$ must all exist as edge labels also.

Observe that Step 4 of Graceful Tree Embedding Algorithm is executed when $\hat{X} \neq \phi$. Further $a = min\hat{X}$, $b = min\hat{V}$ and $\beta = a - b$ are found in the Step 4 We claim that β is a positive integer. For that we prove $a > b$. Suppose not. Then $a \leq b$. Since $a \in \hat{X}, b \in \hat{V}$ and $\hat{X} \cap \hat{V} = \phi, a \neq b$. Thus, $a < b$. Since $min\hat{X} > p$, where $p = |V_1|$, and (V_1, V_2) is the bipartition of the vertex set of the input tree T. Therefore, $a > p$. This implies that $b > p$. Then the labels $1, 2, 3, \cdots, p-1$ do not exist as exclusive vertex labels in T since b is the minimum over V^*. This implies that either the labels $1, 2, 3, \cdots, p-1$ do not exist as vertex labels or else, if the labels $1, 2, 3, \cdots, p-1$ exist as vertex labels then the labels $1, 2, 3, \cdots, p-1$ must all exist as edge labels also.

Suppose the labels $1, 2, 3, \cdots, p-1$ do not exist as vertex labels then T must have only one even level having one vertex labeled '0'. (Since by Step 2.1 of Labeling Algorithm, the consecutive labeling are given in the even level vertices.) This implies that there is only one odd level and all the vertices in this odd level must be adjacent to vertex labeled '0'. But the degree of the vertex labeled '0' is one, therefore there is only one vertex in the unique odd level. Thus, the tree must be K_2. Then by Step 2.3 of Labeling Algorithm, the label of the unique vertex, say c, adjacent to the vertex labeled '0' must have the label $w + 1 \geq p$, where $w = wt(c)$. Then we have, $V = \{0, w+1\}, E = \{w+1\}$ and $I = \{w+1\}$. Therefore $\hat{E} = E - I = \phi$. A contradiction to the fact that $\hat{E} \neq \phi$ (since $a \in \hat{E}$).

Suppose the labels $1, 2, 3, \cdots, p-1$ exist as vertex labels then the labels $1, 2, 3, \cdots, p-1$ must all exist as edge labels also. As $1, 2, 3, \cdots, p-1$ also must appear as edge labels, then structure of the input tree T must be either a labeled double star or a labeled star as given in Figure 1 or Figure 2.

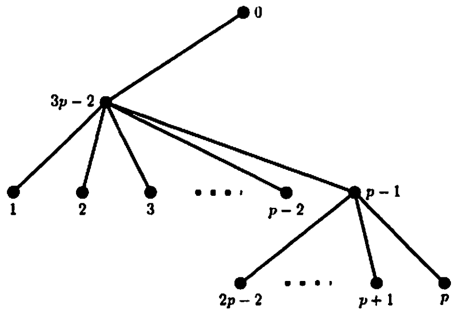

Figure 1: Forcible structure of T as a labeled double star

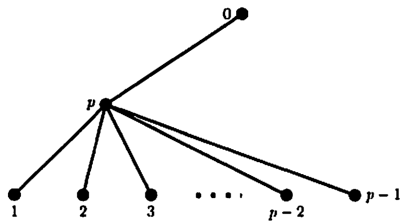

Figure 2: Forcible structure of T as a labeled star

When T is a double star, then by Labeling Algorithm, T shoud have been labeled as shown in Figure 3.

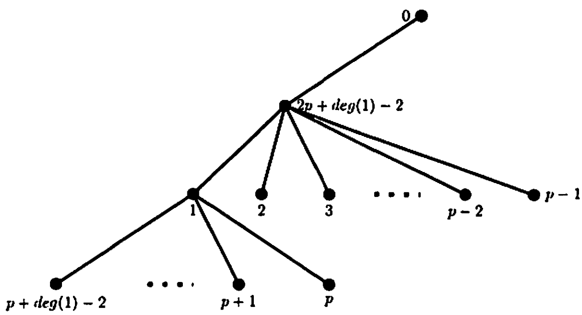

Figure 3: Vertex Labeling for Double Star by Labeling Algorithm

Therefore, T can not be a double star. When T is a star, then T must have the labeling by Labeling Algorithm as given in Figure 2. Then

$V = \{0,1,2,\cdots,p\}$, $E = \{1,2,\cdots,p\}$ and $I = \{1,2,\cdots,p\}$. Therefore, $\hat{E} = E - I = \phi$. But $\hat{E} \neq \phi$, since $a \in \hat{E}$. Hence $b < a$. Therefore $a - b > 0$. Hence β is a positive integer. Since $a = min\hat{X}$, the current tree T^* should contain all the vertex labels $0, 1, 2, \cdots, a-1$. As $\beta = (a-b) < a$, β must be a label of a vertex in that current tree T^*. □

Theorem 2. *The output tree T^* generated by Graceful Tree Embedding Algorithm for an input arbitrary tree T is graceful and contains the input arbitrary tree T as its subtree.*

Proof. For an input arbitrary tree T, obtain the output tree T^* generated by the Graceful Tree Embedding Algorithm. Consider the sets $V, E, I, X, \hat{V}, \hat{E}, \hat{X}$ that are defined in Step 1.2 of Embedding Algorithm. By Step 1.2 of Graceful Tree Embedding Algorithm, we have $\hat{X} = X - V$. Then, we have $X = \hat{X} \cup V$. Since $\hat{E} \subset \hat{X}$, we can write $X = \hat{X} \cup V = ((\hat{X} - \hat{E}) \cup \hat{E}) \cup V$. Observe that by definition of $\hat{E}$, $\hat{E} \cap V = \phi$, $\hat{E} \cap (\hat{X} - \hat{E}) = \phi$ and $V \cap (\hat{X} - \hat{E}) = \phi$. That is, the sets $(\hat{X} - \hat{E}), \hat{E}$ and V are mutually disjoint. Note that V consists of all the vertex labels of T. $\hat{E}$ consists of the edge labels of T that are not vertex labels of T. $\hat{X} - \hat{E}$ consists of the members of X which are neither the vertex labels of T nor the edge labels of T. Consider $a = min\hat{X}$ obtained by an execution of Step 2 of Graceful Tree Embedding Algorithm. If $a \notin \hat{E}$, then by Step 3 of Graceful Tree Embedding Algorithm, the vertex label a is obtained in the updated tree T^* by adding the new edge $(0, a)$ to the current tree T^*. Also a is removed from $\hat{X}$. Since a was removed from $\hat{X}$, the vertex label a will never be obtained again.

If $a \in \hat{E}$, then by Step 4 of Graceful Tree Embedding Algorithm, the vertex label a is obtained in the updated tree T^* by adding the new edge (β, a) in the current tree T^*, where $\beta = a - b$ and $b = min\hat{V}$. Since a is removed from $\hat{X}$, the vertex label a will never be obtained again.

Thus, after executing Step 3 of Graceful Tree Embedding Algorithm $|\hat{X} - \hat{E}|$ times and Step 4 of Graceful Tree Embedding Algorithm $|\hat{E}|$ times, T^* contains all the vertex labels $0, 1, 2, \cdots, M$. Observe that all the vertex labels obtained from Embedding Algorithm are distinct and belong to $X - V$. By Theorem 1, all the vertex labels of T are also distinct. Thus, vertex labels of all the vertices of T^* are distinct and the final updated tree T^* has $M + 1$ vertices with vertex set $V(T^*) = \{0, 1, 2, \cdots, M\}$ (where a vertex of T^* is refered by its corresponding label).

We can write the set $X - \{0\} = \hat{X} \cup (V - \{0\}) = (\hat{X} - \hat{E}) \cup E \cup \hat{V}$. Observe that the sets $(\hat{X} - \hat{E}), \hat{E}, \hat{V}$ and I are mutually disjoint. The elements in $\hat{E}$ and I are already existing as edge labels in T. Consider, $min\hat{X} = c$, obtained at an (any) execution of Step 2 of the Embedding Algorithm. If $a \notin \hat{E}$, then by Step 3 of Graceful Tree Embedding Algorithm, the edge label a is obtained in the updated tree T^* by adding the new edge $(0, a)$ to

the current tree T^* and a is removed from $\hat{X}$. Since a was removed from $\hat{X}$, the edge label a will never be obtained again. If $a \in \hat{E}$, then a unique $b \in \hat{V}$, where $b = min\hat{V}$ is found in an execution of Step 4 of Graceful Tree Embedding Algorithm, and the edge label b is obtained in the updated tree T^* from the new edge (β, a) which was added to the current tree T^*, where $\beta = a - b$ and b is removed from $\hat{V}$. Also a is removed from $\hat{X}$. Since a is removed from $\hat{X}$ and b is removed from $\hat{V}$ the edge label b will never be obtained again. Since $|\hat{V}| = |\hat{E}|$, whenever $a \in \hat{E}$, corresponing unique $b \in \hat{V}$ is found. We see that every element of $\hat{V}$ is obtained as edge label in the final updated tree T^*.
Thus, after executing Step 3 of Graceful Tree Embedding Algorithm $|\hat{X} - \hat{E}|$ times and Step 4 of Graceful Tree Embedding Algorithm $|\hat{V}|(= |\hat{E}|)$ times in the final updated tree T^*, the edge labels belong to $(X - \{0\}) - E$ are all obtained as distinct edge labels. As T^* was initiated with m edges having distinct edge labels belong to the set E, the final updated tree T^* has distinct edge labels $1, 2, \cdots, M$ for its M edges. Thus, the final updated tree T^* is graceful. □

Modified Labeling Algorithm

Consider the Labeling Algorithm.
Retain Steps 1 and 2.1.
Modify Step 2.2 by defining the weight as given below.
For each r, $1 \leq r \leq \lfloor \frac{d}{2} \rfloor$, define the weights of vertices of (2r+1)th level, for $1 \leq i \leq \gamma_{2r+1}$,

$$wt(v_{2r+1,i}) = \begin{cases} 0, & \text{if } deg(v_{2r+1,i}) = 1 \\ l(RMC(v_{2r+1,i})) - l(PR(v_{2r+1,i})), & \text{if } deg(v_{2r+1,i}) > 1 \end{cases}$$

Apply the new definition of weight in Step 2.3.
Retain Step 3 as in the Labeling Algorithm.

Labeling Algorithm with these modifications is called Modified Labeling Algorithm. The output tree thus obtained from the Modified Labeling Algorithm is also denoted by T'.

Note 2: For a given arbitrary input tree T with m edges, we run the Modified Labeling Algorithm and obtain the labeled output tree T'. For convenience, hereafter the vertices of output tree T' are referred to by their vertex labels, and the edges are referred to by their edge labels. Thus, for the output tree T' we consider $V(T') = \{0, 1, 2, \cdots, p-1, p, \alpha_1, \alpha_2, \cdots, M\}$ and $E(T') = \{l'(e_1), l'(e_2), \cdots, l'(e_m)\}$.

Observation 3 From Observation 1, it follows that the vertex labels defined by the Modified Labeling Algorithm are distinct, and from

Theorem 1, it also follows that the edge labels defined by the Modified Labeling Algorithm are also distinct.

Observation 4 In the Modified Labeling Algorithm, it is obvious that the weight of a vertex in the odd level is less than the number of vertices in all the even levels. Thus, the difference between the labels of two consecutive vertices in any odd level is strictly less than the number of vertices in all the even levels of T.

α-labeled Graph Embedding Algorithm
Input: Arbitrary tree T

Step 1:

Run the Modified Labeling Algorithm on input tree T and get the output tree T'.

Step 2:

Define $V_1(T') = \{0, 1, 2, \cdots, p-1\}$ and $V_2(T') = \{p, \alpha_1, \alpha_2, \cdots, M\}$, where M is the maximum of the vertex labels of the vertices of T', the elements in $V_1(T')$ are the vertex labels of the vertices that are in the even levels, and the elements in $V_2(T')$ are the vertex labels of the vertices that are in the odd levels. Define $E(T') = \{l'(e_1), l'(e_2), \cdots, l'(e_m)\}$, the set of edge labels of the edges of T' as defined in the Modified Labeling Algorithm.
Define $E' = \{1, 2, 3, \cdots, M\}$ and $E^* = E' - E$.
Initiate $G^* \leftarrow T'$,
$V(G^*) \leftarrow V(T')$,
$E(G^*) \leftarrow E(T')$.

Step 3:

While $E^* \neq \phi$, find $minE^* = x$.

Step 4:

Define $S = \{y \in V_2(T') | y \geq x\}$. Find $minS = \hat{y}$.

Step 5:

Find $z = \hat{y} - x$. Then add a new edge between the vertex labeled z in $V_1(T')$ and vertex labeled $\hat{y}$ in $V_2(T')$ so that the edge label x is induced.
Update $G^* \leftarrow G^* + (\hat{y}, z)$,
$E(G^*) \leftarrow E(G^*) \cup \{(\hat{y}, z)\}$.
Delete x from E^* and go to Step 3.

Graceful Graph Embedding Algorithm

Input: Arbitrary tree T

Step 1:

Run the Modified Labeling Algorithm on input arbitrary tree T and get the output tree T'.

Step 2:

Define $V_1(T') = \{0, 1, 2, \cdots, p-1\}$ and $V_2(T') = \{p, \alpha_1, \alpha_2, \cdots, M\}$, where M is the maximum of the vertex labels of the vertices of T', the elements in $V_1(T')$ are the vertex labels of the vertices that are in the even levels, and the elements in $V_2(T')$ are the vertex labels of the vertices that are in the odd levels. Define $V(T') = V_1(T') \cup V_2(T')$. Define $E(T) = \{l'(e_1), l'(e_2), \cdots, l'(e_m)\}$, the set of edge labels of the edges of T' as defined in the Modified Labeling Algorithm. Define $E' = \{1, 2, 3, \cdots, M\}$ and $E^* = E' - E$.
Initiate $G \leftarrow T'$,
$V(G) \leftarrow V(T')$,
$E(G) \leftarrow E(T')$.

Step 3:

While $E^* \neq \phi$, find $minE^* = x$.

Step 4:

If $x \in V(T')$, then introduce a new edge between the vertex labeled 0 and x so that the edge label induced in the new edge is x.
Update $G \leftarrow G + (0, x)$,
$E(G) \leftarrow E(G) \cup \{(0, x)\}$.
Delete x from E^* and go to Step 3.

Step 5:

If $x \notin V(T')$, then Define $S = \{y \in V_2(T') | y \geq x\}$. Find $minS = \hat{y}$. Find $z = \hat{y} - x$. Then add a new edge between the vertex labeled z in $V_1(T')$ and vertex labeled $\hat{y}$ in $V_2(T')$ so that the edge label x is induced.
Update $G \leftarrow G + (\hat{y}, z)$,
$E(G) \leftarrow E(G) \cup \{(\hat{y}, z)\}$.
Delete x from E^* and go to Step 3.

Lemma 2. *The z defined in Step 5 of the α-labeled Graph Embedding Algorithm as well as in Step 5 of Graceful Graph Embedding Algorithm always exists as a vertex label in $V_1(T)$ and also z is strictly less than p, the number of vertices in all the even levels of the input tree T.*

Proof. From Step 5 of the α-labeled Graph Embedding Algorithm, z is defined by $y - x$. We prove that $y - x < p$. Suppose not. Then, $y - x \geq p$. Then, identify the largest vertex label from $V_2(T)$ (say y_1) such that $y_1 < x$. By the definition of vertex labeling in the Modified Labeling Algorithm, $y - y_1 < p$. That is, $y < p + y_1$. From the assumption, we have $x + p \leq y$. Therefore, $p + x < p + y_1$ or $x < y_1$, a contradiction to our choice of y_1 such that $y_1 < x$. Hence $z < p$. □

Theorem 3. *The output graph G^* generated by α-labeled Graph Embedding Algorithm is an α-labeled graph containing the given input arbitrary tree as its spanning tree.*

Proof. For an input arbitrary tree T, obtain the output tree T' generated by the Modified Embedding Algorithm. Then note that the vertex labels of the vertices of T' are distinct and the edge labels of the edges of T' are also distinct by Observation 3. By Step 2 of the α-labeled Graph Embedding Algorithm, the vertex set of the output graph $V(G^*) = V(T')$. Thus vertex labels of the vertices of $V(G^*)$ are distinct and the vertex labels are assigned from the set $\{0, 1, 2, \cdots, M\}$. Observe that from Steps 3 to 5, the edge labels belong to E^* are all obtained as distinct edge values of the new edges which are added to the current graph G^* and G^* is updated. Thus the edge labels $1, 2, 3, \cdots, M$ are all obtained in the final updated graph G^*, since $E' = E(T') \cup E^*$ and $E(T') \cap E^* = \phi$. Hence G^* is graceful. Further observe that G^* is bipartite and every edge is having one vertex label in $V_1(T')$ and other vertex label in $V_2(T')$. Thus for every edge $e = xy$ in G^*, either $x \leq p - 1$ and $p - 1 < y$ or $y \leq p - 1$ and $p - 1 < x$. Hence the labeling thus obtained in α-labeled Graph Embedding Algorithm is an α-labeling. Hence G^* is an α-labeled graph. □

Theorem 4. *The output graph G generated by Graceful Graph Embedding Algorithm is graceful and containing the given input arbitrary tree as its spanning tree.*

Proof. For an input arbitrary tree T, obtain the output tree T' generated by the Modified Embedding Algorithm. Then note that the vertex labels of the vertices of T' are distinct and the edge labels of the edges of T' are also distinct by Observation 3. By Step 2 of the Graceful Graph Embedding Algorithm, the vertex set of the output graph $V(G) = V(T')$. Thus vertex labels of the vertices of $V(G)$ are distinct and the vertex labels are assigned from the set $\{0, 1, 2, \cdots, M\}$. Observe that from Steps 3 to 5, the edge labels belong to E^* are all obtained as distinct edge values of the new edges which are added to the current graph G and G is updated. Thus the edge labels $1, 2, 3, \cdots, M$ are all obtained in the final updated graph G since $E' = E(T') \cup E^*$ and $E(T') \cap E^* = \phi$. Hence G is graceful. □

3 Discussion

Theorems 2, 3 and 4 respectively imply the basic structural property of graceful graphs that every tree is a subtree of a graceful graph, an α-labeled graph and a graceful tree. In particular, Graceful Tree Embedding Algorithm generates a graceful tree from the given arbitrary tree. This is a new approach in constructing graceful trees from an arbitrary tree. One can observe that the Graceful Tree Embedding Algorithm generates a graceful tree from a given arbitrary tree by adding a sequence of new pendant edges to the given arbitrary tree. Thus, it is tempting to ask the question,

> *If G is a graceful tree and v is any one degree vertex of G, is it true that $G - v$ is graceful?*

If this question is answered affirmatively, then those additional edges of the input arbitrary tree T introduced for constructing the graceful tree T^* by the Graceful Tree Embedding Algorithm could be deleted in some order so that the given arbitrary tree T becomes graceful. This would imply that the Graceful Tree Conjecture is true.

Illustration

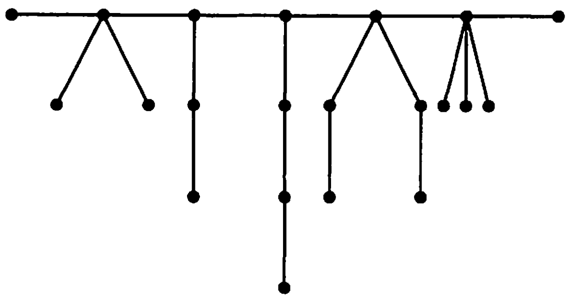

Figure 4: Input Tree T

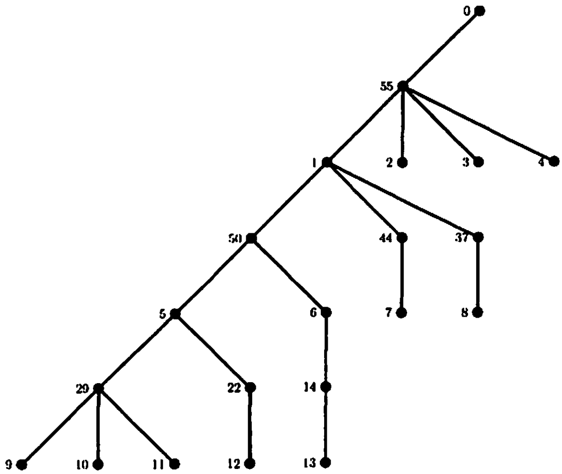

Figure 5: Vertex Labeled Tree T' for the Input Tree T

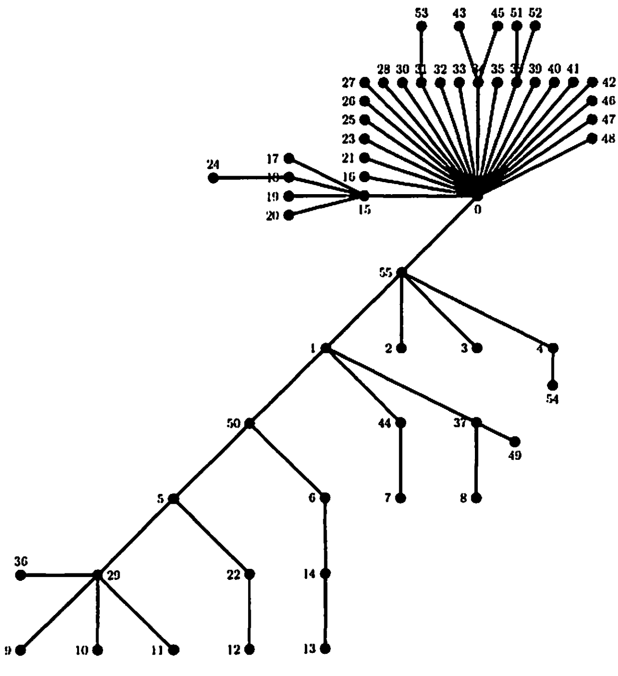

Figure 6: Graceful Tree T^* obtained from the input tree T by Graceful Tree Embedding Algorithm

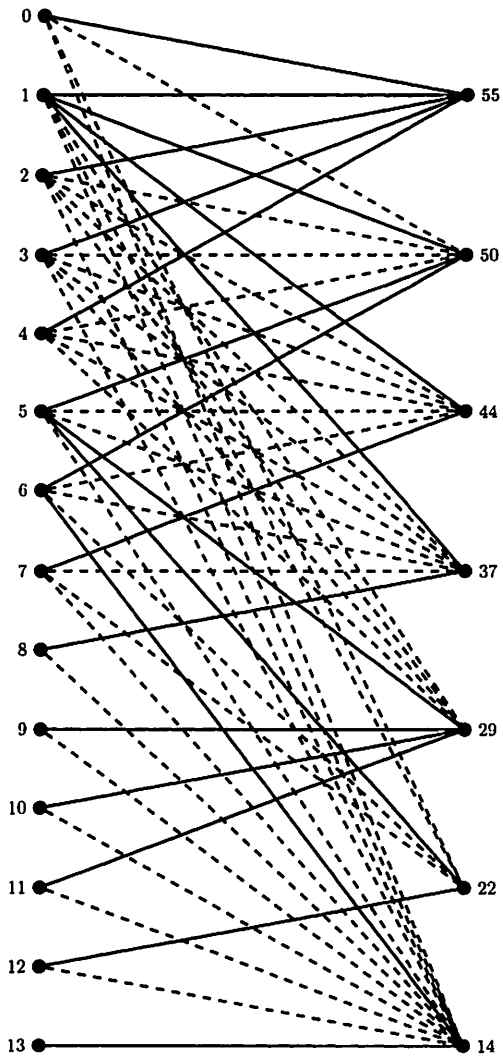

Figure 7: Alpha-Labeled Graph G^* obtained from the input tree T by α-labeled Graph Embedding Algorithm

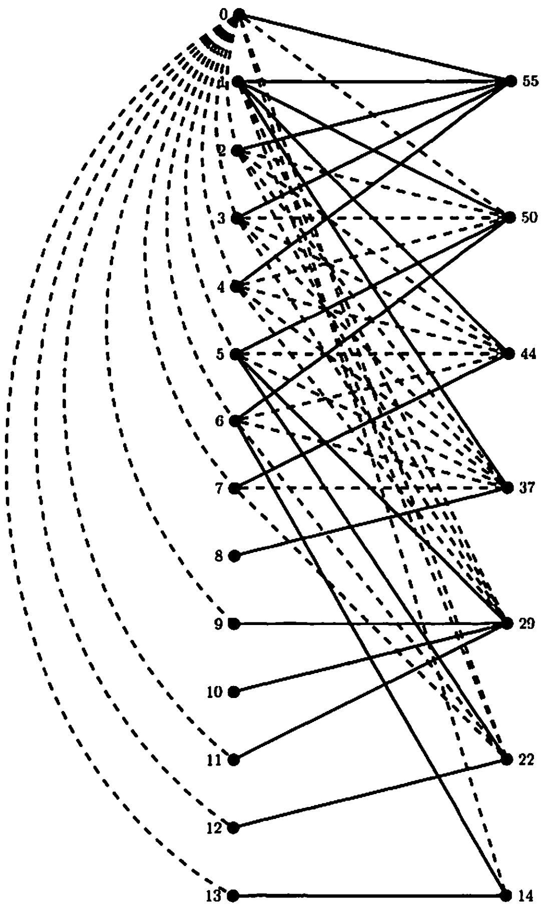

Figure 8: Graceful Graph G obtained from the input tree T by Graceful Graph Embedding Algorithm

Acknowledgments

The Authors would like to thank the referee for the valuable comments. The second author thankfully acknowledges Centre for Research, Anna University, Chennai, for the Anna Centenary Research Fellowship under the grant Ref:CR/ACRF/Jan.2011/31.

References

[1] B.D. Acharya, S.B. Rao, S. Arumugam, *Embedding and NP-Complete problems for Graceful Graphs, Labelings of Discrete Structures and Applications*, B.D. Acharya, S. Arumugam, Alexander Rosa, eds., (2008), 57-62, Narosa Publishing House, New Delhi.

[2] Bermond J.C, *Graceful graphs, radio antennae and French windmills*, Graph Theory and Combinatorics, Pitman, London, (1979), 18-37.

[3] Bloom G.S, *A chronology of the Ringel-Kotzig conjecture and the continuing quest to call all trees graceful*, Ann. N.Y. Acad. Sci., 326, (1979), 35-51.

[4] Burzio M. and Ferrarese G, *The subdivision graph of a graceful tree is a graceful tree*, Disc. Math, 181, (1998), 275-281.

[5] Cahit. I, *Graceful labelings of rooted complete trees*, 2002, preprint.

[6] Douglas S. Jungreis and Michael Reid, *Labeling Grids*, Ars Combinatoria, 34, (1992), 167-182.

[7] J.A.Gallian, *A Dynamic Survey of Graph Labeling*, The Electronic Journal of Combinatorics, 18, (2011), #DS6.

[8] Golomb S.W, *How to number a graph*, Graph Theory and Computing R.C. Read, ed., Academic Press, New York, 1972, 23-37.

[9] Jeba Jesintha.J and Sethuraman G, *All arbitrary fixed generalized banana trees are graceful*, Math. Comput. Sci, 5, (2011),1,51-62.

[10] Kotzig A, *Decompositions of a complete graph into 4k-gons (in Russian)*, Matematicky Casopis, 15, (1965), 229-233.

[11] Koh K.H., Rogers D.G. and Tan T, *Two theorems on graceful trees*, Discrete Mathematics, 25, (1979), 141-148.

[12] Ng H.K, *Gracefulness of a class of lobsters*, Notices AMS, 7, (1986), 825-05-294.

[13] Pavel Havier and Alfonz Havier, *All trees of diameter five are graceful*, Discrete Mathematics, 233, (2001), 133-150.

[14] Ringel G, *Problem 25, in Theory of Graphs and its Applications*, Proc. Symposium Smolenice, Prague, (1963) page-162.

[15] Rosa A, *On certain valuations of the vertices of a graph*, Theory of graphs, (International Symposium, Rome, July 1966), Gordon and Breach, N.Y. and Dunod Paris, (1967),349-355.

[16] Sethuraman G. and Venkatesh S, *Decomposition of complete graphs and complete bipartite graphs into α-labeled trees*, Ars Combinatoria, 93, (2009), 371-385.

[17] Sethuraman G. and J. Jeba Jesintha, *All banana trees are graceful*, Advanced Applied Discrete Mathematics, 4, (2009), 1, 53-64.

[18] Van Bussel F.,*Relaxed graceful labelings of trees*, The Electronic Journal of Combinatorics, 9, (2002), #R4.

[19] W. Fang, *A Computational Approach to the Graceful Tree Conjecture*, arXiv:1003.3045v1[cs.DM]

[20] Wang J.G., Jin D.J., Lu X.G. and Zhang D., *The gracefulness of a class of lobster trees*, Math. Comput. Modelling, 20, (1994), 105-110.

[21] West D.B., *Introduction to Graph Theory*, Prentice Hall of India, 2nd Edition, 2001.