VNU Journal of Science: Comp. Science & Com. Eng. Vol. 30, No. 3 (2014) 1–11

An Algorithm for Graceful Labelings of Certain Unicyclic Graphs

Pambe Biatch' Max[1], Jay Bagga[2], Laure Pauline Fotso[1]

[1]*University of Yaounde I, Yaounde, Cameroon*

[2]*Ball State University, Muncie, Indiana, USA*

Abstract

A graceful labeling of a simple graph G is a one-to-one map f from the vertices of G to the set $\{0, 1, 2, \cdots, |E(G)|\}$, such that when each edge xy is assigned the label $|f(x) - f(y)|$, the resulting set of edge labels is $\{1, 2, \cdots, |E(G)|\}$, with no label repeated. We are interested at Truszczynski's conjecture, that all unicyclic graphs except cycles C_n with $n \equiv 1(mod\ 4)$ or $n \equiv 2(mod\ 4)$, are graceful. Jay Bagga et al. introduced an algorithm to enumerate graceful labelings of cycles and "sun graphs". We generalize their algorithm to enumerate all graceful labelings of a class of unicyclic graphs and provide some experimental results.

Manuscript article: received 24 January 2014, revised 14 March 2014, accepted 25 March 2014

Corresponding author: Jay Bagga, jbagga@bsu.edu

Keywords: Unicyclic graph, Labeling algorithm, Graceful labeling

1. Introduction

Given a simple graph $G = (V, E)$ with the set of vertices $V(G)$ and the set of edges $E(G)$, f is a vertex (resp. edge) labeling of G if it is a mapping from $V(G)$ (resp. $E(G)$) to a set L of labels. If f is an injection, from $V(G)$ to $\{0, 1, \cdots, |E(G)|\}$ and if for all edges xy of $E(G)$, the assigned labels $\left|f(x) - f(y)\right|$ are all distinct, then f is called a graceful labeling. A graph G is graceful if it has a graceful labeling. Rosa [6] called such a labeling a β-valuation. The term graceful labeling was first used by Golomb [5]. Graceful labeling traces its origin in 1967 when Ringel [6] conjectured that every tree T with n edges, decomposes the complete graph K_{2n+1} in $2n + 1$ subgraphs, all isomorphic to T. To our knowledge, Ringel's conjecture is still unsolved. An attempt of solution was made by Rosa [4] who showed that if a tree T with n edges is graceful, then it decomposes the complete graph K_{2n+1} in $2n + 1$ subgraphs, all isomorphic to T. He further conjectured that every tree is graceful. Even though Rosa's conjecture is still open, special classes of trees including caterpillars [6], symmetrical trees [6], trees with at most 4 end-vertices and trees with diameter at most 5 [9] have been shown to be graceful.

Rosa [6] showed that a cycle C_n is graceful for all n except when $n \equiv 1(mod\ 4)$ or $n \equiv 2(mod\ 4)$. This led to the discovery of several classes of unicyclic graceful graphs. Truszczynski [8] conjectured that all unicyclic graphs except the cycles forbidden by Rosa. Bermond [3] conjectured that lobsters are graceful. In this paper, we focus our work on Truszczynski's conjecture and Jay Bagga et al. algorithm [1]. Jay Bagga et al. [1] designed algorithms to enumerate graceful labelings of all graceful cycles and certain classes of graceful unicyclic graphs. We present a generalization of that algorithm and use it to generate graceful labelings of some new classes of unicyclic graphs.

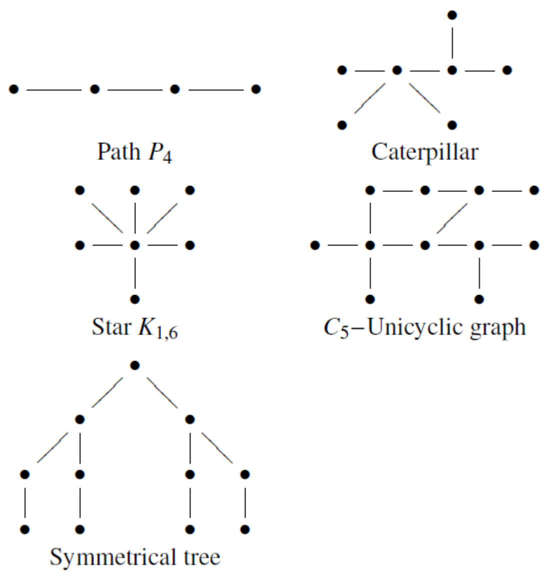


Fig. 1: Some common graphs.

The rest of the paper is organized as follows: Section 2 introduces basic definitions and notation used throughout the paper. Section 3 briefly describes the algorithm of Jay Bagga et al [1], introduces our new algorithm, explains a proof of correctness, and presents some experimental results. We conclude in section 4.

2. Definitions and Notation

In this section, we introduce some definitions and notation. Definitions of common classes of graphs such as paths, stars, caterpillars and unicyclic graphs can be found in standards graph theory books. Figure 1 illustrates some of the common graphs. A C_n−unicyclic graph is one where the cycle has n vertices. We observe that for unicyclic graphs, the number of vertices is equal to the number of edges. A symmetrical tree is a rooted tree in which every level contains vertices of the same degree.

Given a labeling f of a unicyclic graph G, a *sublabeling* is an ordered union of disjoint subsequences of f. As described in Jay Bagga et al. [1], a labeling $f =< a_1, a_2, \cdots, a_n >$ of C_n can be considered an ordered (circular) sequence. When f is graceful, then for $1 \leq k \leq n$, we get n sublabelings S_k of f, where S_k is the sublabeling of f which produces edge labels $k, k+1, \cdots, n$. We may also consider this sublabeling S_k of f as the ordered union of paths in C_n containing edges with labels k through n. For example, given the graceful labeling $f =< 4, 15, 0, 16, 2, 11, 3, 13, 1, 14, 7, 9, 12, 6, 10, 5 >$ of C_{16}, we have $S_{13} =< 15, 0, 16, 2 >< 1, 14 >$. Thus S_{13} is the ordered union of the two paths P_4 and P_2 with vertices labeled 15-0-16-2 and 1-14, respectively. We also observe that for any graceful labeling f, $S_n =< 0, n >$ and $S_1 = f$.

Adding first (resp. adding last) an element e to a sublabeling S_k of the labeling f results in inserting e at the first (resp. last) position in one of the sequences of S_k. The operation is denoted $addfirst(S_k, e)$ (resp. $addlast(S_k, e)$). For example, adding first the element 2 to the sublabeling $< 4, 5, 9 >$ gives $< 2, 4, 5, 9 >$. Adding last the element 1 to the sublabeling $< 4, 5, 9 >$ gives $< 4, 5, 9, 1 >$. Concatenating two sublabelings S_{k_1} and S_{k_2} results in applying $addlast(S_{k_1}, e)$ repeatedly to the elements e of S_{k_2}. The operation of concatenation is denoted $concat(S_{k_1}, S_{k_2})$. For example $concat(< 4, 5, 2 >, < 8, 0, 1 >) =< 4, 5, 2, 8, 0, 1 >$.

If $f =< a_1, a_2, \cdots, a_n >$ is a graceful labeling of a unicyclic graph G of order n, then the *complementary labeling* $\overline{f}$ of f is given by $\overline{f} =< n - a_1, n - a_2, \cdots, n - a_n >$. Clearly, $\overline{f}$ is also a graceful labeling of G.

3. Enumerating graceful labelings of graphs

In this section, we describe an algorithm for enumeration of graceful labelings of unicyclic graphs and use it to enumerate graceful labelings of unicyclic graphs obtained by identifying an end vertex of a star to a vertex of a cycle, $K_{1,m-1} \oplus C_4$, $3 \leq m \leq 15$.

3.1. Algorithm of Jay Bagga et al. [1]

The algorithm of Jay Bagga et al. finds graceful labelings of a cycle C_n by generating edge labels as it traverses the nodes of an execution tree. Given a cycle C_n , the algorithm

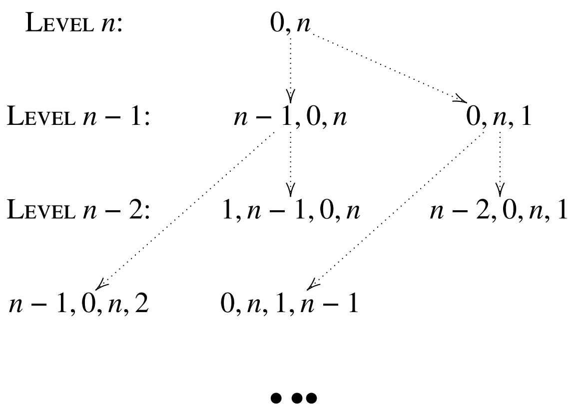


Fig. 2: Nodes of the execution tree of the algorithm of Jay Bagga et al.

starts the computation at level L with $L = n$, where level indicates that it is necessary to find a sublabeling containing two labels a_i and a_j such as $|a_i - a_j| = L$. At level $L = n$ there exists only one sublabeling, namely $< 0, n >$ and hence this is the starting sublabeling. The next step is to find sublabelings for $L = n - 1$. In this case, there are two alternatives: $< n - 1, 0, n >$ and $< 0, n, 1 >$. The algorithm splits the computation into two branches. The left branch uses the sublabeling $< n - 1, 0, n >$ and the right branch uses the sublabeling $< 0, n, 1 >$. The algorithm continues in this way, computing sublabelings for $L = n - 2$ by splitting into several branches each time and recursively calling each branch. The computation for a particular branch continues until either a graceful labeling is found or no graceful labeling is possible. In the last case, a backtracking is performed. Figure 2 shows the nodes of the execution tree from level n to $n - 2$.

Figure 3 shows an example of enumeration of graceful labelings of the cycle C_4 when $f =< 1, 3, 0, 4 >$, $< 3, 0, 4, 2 >$, $< 2, 0, 4, 1 >$, and $< 0, 4, 1, 3 >$ producing respectively the edge labels set $\{2, 3, 4, 3, \}$, $\{3, 4, 2, 1\}$, $\{2, 4, 3, 1\}$ and $\{4, 3, 2, 3\}$. We observe that the labelings $< 1, 3, 0, 4 >$ and $< 0, 4, 1, 3 >$ are not graceful, while $< 3, 0, 4, 2 >$ and $< 2, 0, 4, 1 >$ are graceful.

In the next subsection, we present a generalization of this algorithm which enumerates graceful labelings of some classes of

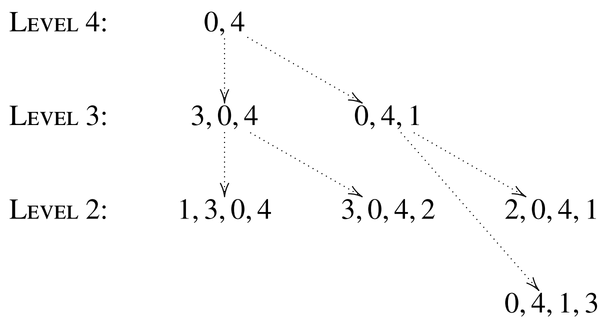


Fig. 3: Execution tree of the enumeration of graceful labelings of C_4.

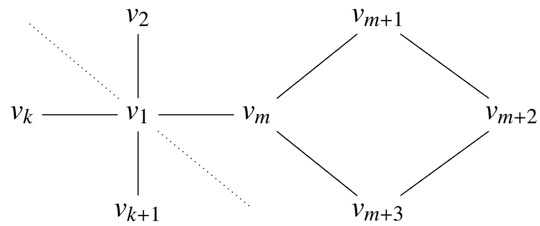


Fig. 4: Unicyclic graphs $K_{1,m-1} \oplus C_4$.

graceful unicyclic graphs.

3.2. New Approach for enumerating Graceful Labelings of unicyclic graphs

Our new approach constructs an execution tree from the root to the leaves like the algorithm of Jay Bagga et al. [1]. We consider the class $K_{1,m-1} \oplus C_4$ of unicyclic graphs composed of a star $K_{1,m-1}$ with m vertices and a cycle C_4 with 4 vertices. Figure 4 shows such a class of unicyclic graphs. Sekar [7] proved that graphs belonging to this class of unicyclic graphs are graceful.

We represent a labeling of a graph of this class by

$$\blacktriangleleft\ s_1, s_2, \cdots, s_m, c_{m+1}, c_{m+2}, c_{m+3}\ \blacktriangleright$$

where s_1 is the label of the central vertex of the star, s_2, s_3,$\cdots$, s_{m-1} are the labels of the peripheral vertices of the star. s_m is the label of the common vertex and c_{m+1}, c_{m+2}, c_{m+3} are the labels of the vertices of the cycle. In other words,

$$f(v_i) = \begin{cases} s_i & \text{if} \quad i \in \{1, 2, \cdots, m\}, \\ c_i & \text{if} \quad i \in \{m+1, m+2, m+3\}. \end{cases}$$

as shown in figure 5.

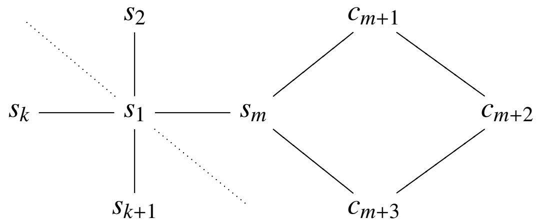


Fig. 5: Labeling of a graph of the class $K_{1,m-1} \oplus C_4$.

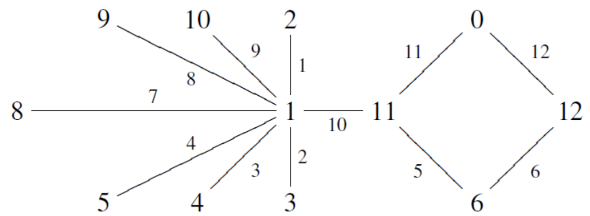


Fig. 6: A graceful labeling of $K_{1,8} \oplus C_4$ produced by our algorithm.

We use three procedures, Common, StarL and CycleL which are called whenever the previously labeled vertex is respectively the common vertex, a vertex in the star or a vertex in the cycle. The main algorithm performs all graceful labelings of a given graceful graph. The label of the common vertex can be any of the vertex labels. The main algorithm proceeds as follows:

i. Either assign 0 to the common vertex, or to a vertex in the star or to a vertex in the cycle.

ii. If the assigned vertex is the common vertex then procedure Common is called to look for edge label n. Otherwise if the labeled vertex is a vertex of the star, procedure StarL is called to look for edge label n. Otherwise CycleL is called to look for edge label n.

iii. End.

Figure 6 illustrates an example of a graceful labeling produced by these procedures. We describe these procedures next.

3.2.1. Description of the procedure Common

The procedure Common enumerates graceful labelings of the unicyclic graph $K_{1,m-1} \oplus C_4$ starting when the label 0 or $m + 3$ is assigned to the common vertex v_m. From a previously labeled vertex, it uses the set of available labels and the edge label l to be produced to label a new vertex in the star or in the cycle. If it successfully labels a vertex in the star or in the cycle, StarL and CycleL are called to look for edge label $l - 1$. If not, the labeling is incomplete and the execution stops.

i. Suppose $l = n$ and the label 0 is assigned to the common vertex. There is just one way of obtaining edge label n: by labeling an adjacent vertex of the common vertex with the highest label l. If the labeled vertex is in the star, it is necessarily s_1, otherwise it can be any of the two neighbors of the common vertex in the cycle.

ii. If the labeled vertex is in the star, we assign to a peripheral vertex a vertex label such that the obtained edge label is $n - 1$. If the labeled vertex is in the cycle, we assign to an adjacent vertex, a vertex label such that the obtained edge label is $n - 1$. The procedure for obtaining edge label $n - 2$ is similar : in the star, we assign to a peripheral vertex a vertex label such that the obtained edge label is $n - 2$; in the cycle, we assign to an adjacent vertex of previously labeled vertex, a vertex label such that the obtained edge label is $n - 2$.

iii. More generally, suppose we have found all edge labels from n down to $k + 1$ and we want to obtain edge label k, for $k = n-3, n-4, \cdots, 2, 1$.
In the cycle, as in the algorithm of Jay Bagga et al, we assign, if possible, to an adjacent vertex of previously labeled vertex, a vertex label such that the obtained edge label is k. In the star, we assign if possible to a peripheral vertex, a vertex label such that the obtained edge label is k. Else the procedure stops.

3.2.2. Description of the procedure CycleL

The procedure *CycleL* labels the vertices of the cycle. It is a modified version of the algorithm of Jay Bagga et al. [1]. It uses the available vertex labels, the previously labeled vertices and the edge label l to be produced to look for edge label $l-1$. The difference with the algorithm of Jay Bagga et al is that: if the previously labeled vertex is the common vertex, it calls the procedure common to look for the edge label $l-1$ instead of recursively calling itself as with the other vertices of the cycle. If CycleL fails in finding the edge label l, the execution stops and a backtrack is performed. In the line 3 of the algorithm *CycleL*, S represents a subsequence in S_c. Rank is the index of the subsequence in the sublabeling.

3.2.3. Description of the procedure StarL

The procedure *StarL* labels the vertices of the star. It uses the available vertex labels, the previously labeled vertices, the label of the central vertex and the edge label l to be produced to look for edge label $l-1$. If *StarL* is called for the first time, there are two cases. In the first case, the label 0 has been assigned to a vertex of the cycle. Then the previously labeled vertex can only be the common vertex; in this case, the central vertex is assigned a label such that the induced edge label is l. In the other case (the algorithm started with the assignment of the label 0 to the central vertex of the star), independently of the previously labeled vertices, *StarL* searches to assign a label to a peripheral vertex such that the induced edge label is l, this is done as follows: if the peripheral vertex to be labeled is the common vertex, it calls the procedure *Common* to look for edge label $l-1$; otherwise *StarL* is recursively called to look for the edge label $l-1$. If *StarL* fails in finding the edge label l, the execution stops and a backtrack is performed.

3.2.4. Main Algorithm and detailed description of the procedures

The following variables are used in the main algorithm and the procedures: L is the set of available vertex labels; m is the number of vertices of the star; S_s is a sublabeling containing labels of the vertices of the star; S_c is a sublabeling containing labels of vertices of the cycle; l is the value of the edge label to be produced; l_a is an edge label which is automatically calculated when all the vertices of the cycle are labeled. S_s and S_c indicate the vertices already labeled in the star and in the cycle, respectively.

Algorithm 1: CycleL

Input : L, m, S_s, S_c, l, l_a
Output: L (updated), S_c (updated) {∗ *S_c is a concatenation of sequences* ∗}

1 **begin**
2 Possibility = $\emptyset$
3 **for** $w \in L$ **do**
4 **for** $S \in S_c$ **do**
5 **if** $|w - first(S)| = l$ **then**
6 {∗ *The function first (resp. last) returns the first (resp. last) element of a sequence* ∗}
7 Possibility = Possibility $\cup\{(w, first, rank)\}$
8 **if** $|w - last(S)| = l$ **then**
9 Possibility = Possibility $\cup\{(w, last, rank)\}$
10 **if** *the number of elements of S_c is 4* {∗ All the vertices of the cycle are labeled ∗} **then**
11 $l_c = |S_c(1) - S_c(4)|$
12 **else**
13 $l_c = l_a$
14 **for** *all $(v, position, rank) \in Possibility$* **do**
15 $S_d = new(S_c)$ {∗ *A new sublabeling S_d is created and elements of S_c are copied in S_d* ∗}
16 **if** *position = first* **then**
17 $addfirst(S_d(rank), v)$ {∗ *$S_l(i)$ returns the ith sequence of the sublabeling S_l* ∗}
18 **else**
19 $addlast(S_d(rank), v)$
20 **if** *all the vertices of the cycle have not been labeled* **then**
21 Call CycleL $(L \setminus \{v\}, m, S_s, S_d, l-1, l_a)$
22 Call Commom $(L \setminus \{v\}, m, S_s, S_d, l-1, l_c)$
23 **end**

Example 1. *Consider the unicyclic graph $K_{1,2} \oplus C_4$ in figure 7. The application of the main algorithm, illustrated in figure 9, produces the following result:*

- $(Start_1)$ *shows the labeling of the common vertex with 0.*
  $(Start_2)$ *presents the labeling of the central vertex of the star with 0. There are two*

Algorithm 2: Main-Algorithm

```
Input : m
begin
    S_s =< >
    S_c =< >
    l = m + 3
    initialize(L, l)
    addfirst(S_s, 0)
    Call StarL (L, m, S_s, S_c, l, -1)
    S_s =< >
    addfirst(S_c, 0)
    Call CycleL (L, m, S_s, S_c, l, -1)
    S_s =< >
    S_c =< >
    Call Common (L, m, S_s, S_c, l, -1)
end
```

Algorithm 3: StarL

```
Input : L, m, S_s, S_c, l, l_a
Output: L (updated), S_s (updated)
begin
    Possibility = ∅
    for w ∈ L do
        if |w − first(S_s)| = l then
            {* The function first returns the first element
            of a sequence *}
            Possibility = Possibility ∪{w}
    for all v ∈ Possibility do
        S_d = new(S_s) {* A new sublabeling S_d is created
        and elements of S_s are copied in S_d *}
        addlast(S_d, v)
        if all the vertices of the star are not labeled then
            Call StarL (L \ {v}, m, S_d, S_c, l − 1, l_a)
        Call Commom (L \ {v}, m, S_d, S_c, l − 1, l_a)
end
```

Algorithm 4: Common

```
Input : L, m, S_s, S_c, l, l_a
Output: Labeling f, the concatenation of S_s and S_c // f is
        graceful or not
begin
    if All the edge labels have been produced then
        Set f =< > // The empty sequence
        for all labels v in S_s do
            addlast(f, v)
        for all label v in S_c do
            addlast(f, v)
        if f is graceful then
            Output f
    else
        if there exists a vertex of the star that is not
        labeled then
            if l ≠ l_a then
                Call StarL (L, m, S_s, S_c, l, l_a)
            else
                Call StarL (L, m, S_s, S_c, l − 1, l_a)
        if there exists a vertex of the cycle that is not
        labeled then
            Call CycleL (L, m, S_s, S_c, l, l_a)
end
```

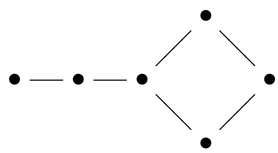


Fig. 7: Unicyclic graph $K_{1,2} \oplus C_4$.

branches: 6_1 *(Labeling of the peripheral vertex of the star with 6. The procedure StarL is called to look for edge label 5) and* 6_2 *(Labeling of the common vertex with 6. The procedure Common is called to look for edge label 5).*
($Start_3$) *presents the labeling of a vertex of the cycle with 0. There are two branches:* 6_3 *(Labeling of the common vertex with 6. The procedure CycleL is called to look for edge label 5) and* 6_4 *(Labeling of a vertex of the cycle with 6. The procedure CycleL is called to look for edge label 5).*

- (5_1) *shows the labeling of the common vertex of the cycle with 5. The procedure Common is called to look for edge label 4.*
  (5_2) *shows the labeling of the common vertex of the cycle with 5. The procedure Common is called to look for edge label 4.*
  (5_3) *shows the labeling of a vertex of the cycle with 1. There are two branches:* 4_1 *(Labeling of the common vertex with 4. The procedure StarL is called to look for edge label 3) and* 4_2 *(Labeling of the common vertex with 5. The procedure Common is called to look for edge label 3).*

- *And so on...*

Figure 8 shows the execution tree of the main algorithm. In this execution tree, $X \rightarrow Y$ *means that node Y emanates from node X. At a leaf of the execution tree, a backtracking is performed*

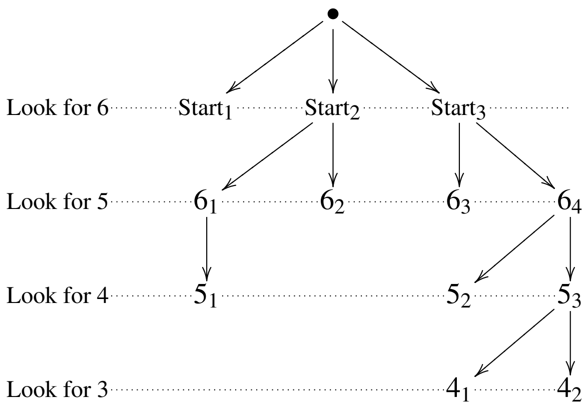


Fig. 8: Nodes of the execution tree.

or procedure Common is applied. For example, consider the node $(Start_1)$ *in figure 9, procedure Common is applied on it. Here the vertex label set is* $L = \{0, 1, 2, 3, 4, 5, 6\}$ *and figure 10 illustrates the execution of the procedure common on the node* $(Start_1)$*:*

- $(Start_1)$ *shows the labeling of the common vertex with 0. There are two branches:* 6_1 *(labeling of the central vertex of the star with 6) and* 6_2 *(labeling of an adjacent vertex of the common vertex in the cycle with 6).*
- (6_1) *produces the branches* 5_1 *(labeling of the peripheral vertex of the star with 1) and* 5_2 *(labeling of an adjacent vertex of the common vertex, in the cycle, with 5).*
  (6_2) *produces the branches* 5_3 *(labeling of the central vertex of the star with 5),* 5_4 *(labeling of an adjacent vertex of the common vertex, in the cycle, with 5) and* 5_5 *(labeling of a vertex in the cycle with 1).*
- *And so on...*

At the end of the execution of the procedure Common, we have 4 graceful labelings:

- $\blacktriangleleft\ 6, 1, 0, 4, 2, 3\ \blacktriangleright$,
- $\blacktriangleleft\ 6, 3, 0, 5, 1, 2\ \blacktriangleright$,
- $\blacktriangleleft\ 5, 1, 0, 6, 3, 2\ \blacktriangleright$,
- $\blacktriangleleft\ 5, 1, 0, 6, 4, 3\ \blacktriangleright$.

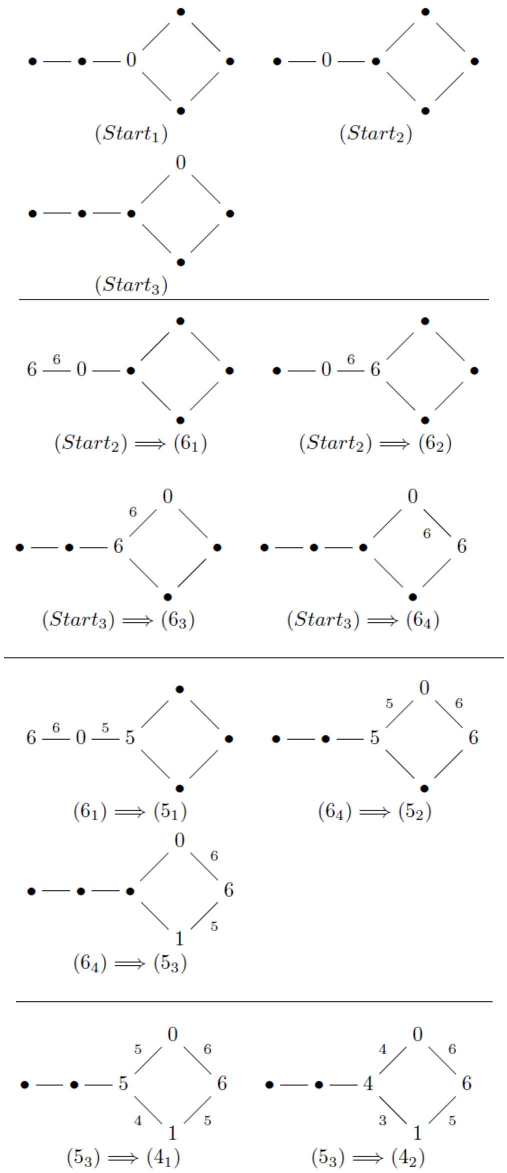


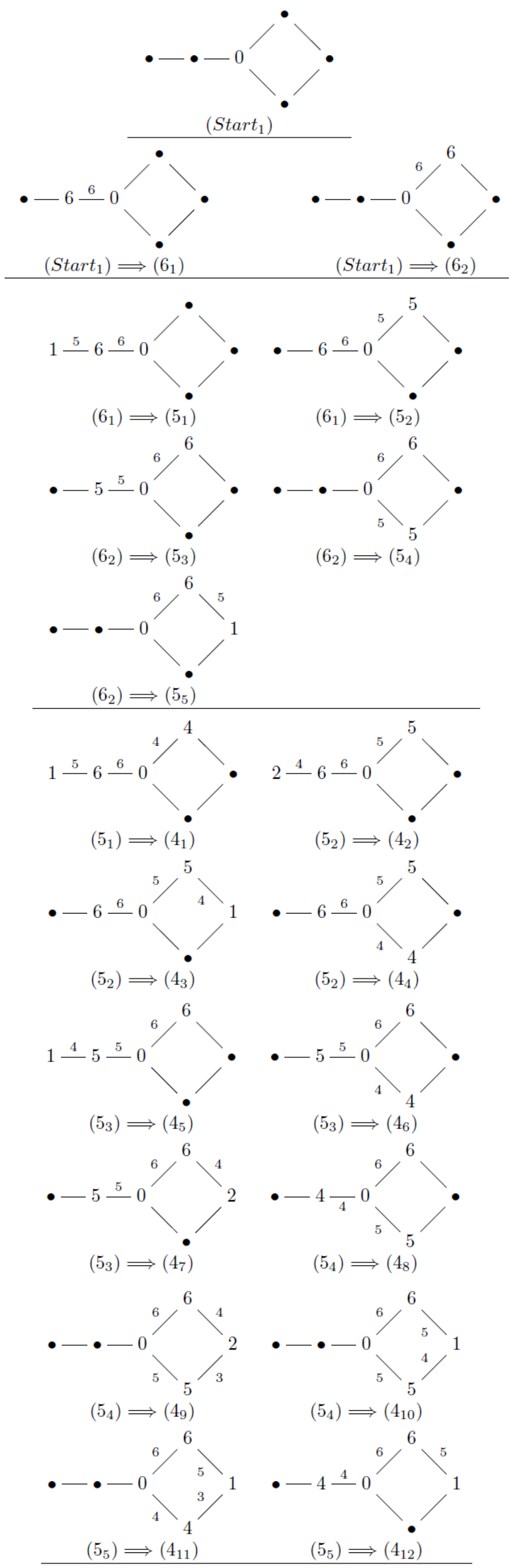
Fig. 9: Execution of procedure Main-Algorithm on $K_{1,2} \oplus C_4$.

3.3. *Correctness of the algorithm*

In this section we present a proof of the correctness of the algorithm.

Theorem 1. *The algorithm achieves a graceful labeling* $f = \blacktriangleleft\ s_1, s_2, \cdots, s_m, c_{m+1}, c_{m+2}, c_{m+3}\ \blacktriangleright$ *of* $K_{1,m-1} \oplus C_4$ *exactly once.*

Proof 1. *We prove it by induction on the sublabeling* S_k*. A sublabeling* S_k *of* f *is the union*



And so on.

Fig. 10: Execution of procedure Common on the node $Start_1$.

of those subsequences of f that produce edge labels from n down to k. For every sublabeling S_k of f ($1 \leq k \leq n$), our algorithm achieves S_k exactly once.

The algorithm starts by looking for the edge label n. Thus for the base case, $k = n$, $S_n = \blacktriangleleft 0, n \blacktriangleright$ and the algorithm achieves it at level n at the the root of the execution tree. Suppose that the algorithm achieves S_{k+1} exactly once, let prove that S_k is also achieved exactly once. Suppose that in S_k, the edge label k is obtained by vertex labels l_x (assigned to vertex x)and l_y (assigned to vertex y) in f, so that $|l_x - l_y| = k$. There are many cases (= is part of S_i in all these cases):

- *xy is an interior edge of a path (illustrated in figure 11(a)).*
- *xy is a pendant edge of a path (illustrated in figure 11(b)).*
- *e is the edge of the path P_2 elsewhere in the graph (illustrated in figure 11(c)).*
- *xy is the edge of the path P_2 intersecting another path of S_{k+1} (illustrated in figure 11(d)).*

When the algorithm tries to achieve S_{k+1}, it uses exactly one of the four cases described. Thus, from n down to k the algorithm achieves S_k exactly once. Then for $k = 1$, $S_1 = f$ and by induction we can conclude that the algorithm achieves exactly once.

3.4. Experimental results

We implemented our algorithm to enumerate graceful labelings of some unicyclic graphs $K_{1,m-1} \oplus C_4$, $3 \leq m \leq 15$. Table 1 contains all the graceful labelings of $K_{1,2} \oplus C_4$. Figure 12 illustrates the graceful labeling of $K_{1,2} \oplus C_4$ in line 9. The last column of table 2 gives the total number of graceful labelings of $K_{1,m-1} \oplus C_4$. We observe that since a unicyclic graph G of order n has n edges, exactly one of the vertex labels from the set $\{0, 1, 2, \cdots, n\}$ is missing from any graceful labeling of G. As shown in the results by Jay Bagga et al. [2], the study of missing labels is of interest. In table 2, the element on the intersection

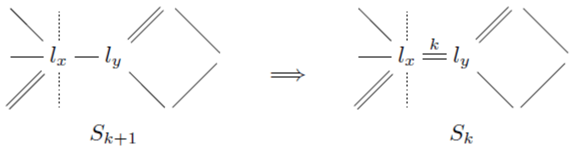


Figure 11(a): xy is an interior edge of a path.

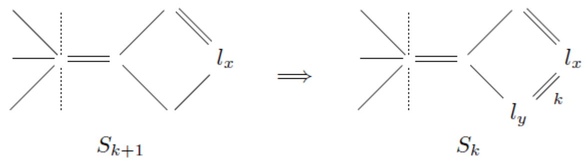


Figure 11(b): xy is a pendant edge of a path.

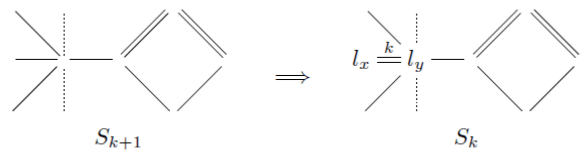


Figure 11(c): e is the edge of the path

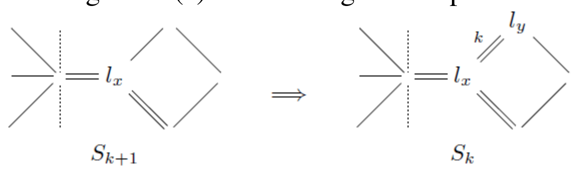


Figure 11(d): xy is the edge of the path P_2 intersecting

Fig. 11: Correctness of the algorithm.

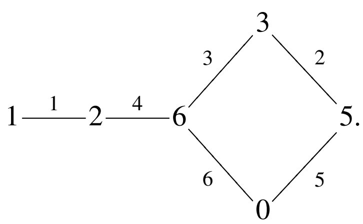


Fig. 12: Graph $K_{1,2} \oplus C_4$ with labeling on line 9. of table 1.

of column l and row G gives the number of times the vertex label l is missing from the graceful labelings of G. For example, the value 12 in column 8 and row $S_9 \oplus C_4$ is the number of times the vertex label 8 is missing from the 82 graceful labelings of $K_{1,8} \oplus C_4$. Clearly, this table is symmetric about the middle column or columns confirming the fact that the for every labeling with a missing label a, the complementary labeling has

Table 1. 26 Graceful labelings of $K_{1,2} \oplus C_4$.

N^{O}	Graceful labeling					
1.	2	3	4	1	6	0
2.	5	4	1	6	0	3
3.	3	2	1	6	0	4
4.	4	3	2	6	0	5
5.	1	2	5	3	6	0
6.	3	4	5	2	6	0
7.	1	5	6	0	2	3
8.	1	5	6	0	3	4
9.	***2***	***1***	***6***	***0***	***5***	***3***
10.	4	3	6	0	5	2
11.	0	6	5	3	4	1
12.	0	6	5	2	3	1
13.	0	5	6	4	1	2
14.	0	5	6	3	4	2
15.	0	5	6	3	1	2
16.	6	1	0	2	5	4
17.	6	1	0	3	2	4
18.	6	1	0	3	5	4
19.	6	3	0	2	1	5
20.	5	1	0	2	3	6
21.	5	1	0	3	4	6
22.	4	5	0	3	1	6
23.	2	3	0	4	1	6
24.	6	0	1	3	2	5
25.	6	0	1	4	3	5
26.	0	3	6	4	5	1

the missing label $n - a$.

In table 3, a dot on the intersection of column l and row G indicates that the label l is assigned to the central vertex of the star in G. For example, the dot on the intersection of column 2 and line $K_{1,9} \oplus C_4$ indicates that 2 is assigned to the central vertex of $K_{1,9}$. Table 3 shows that for $6 \le m \le 15$ and for any graceful labeling of $K_{1,m-1} \oplus C_4$, the central vertex cannot have a label in the set $\{4, 5, \cdots, m-1\}$. We next show that this result holds for all $m \ge 6$.

Theorem 2. *For $m \ge 6$ and for any graceful labeling of $K_{1,m-1} \oplus C_4$, the central vertex cannot have a label in the set $\{4, 5, \cdots, m-1\}$.*

Proof 2. *Suppose f is a graceful labeling of $K_{1,m-1} \oplus C_4$. We observe that for each of the edge labels $m + x$, for $0 \le x \le 3$, the vertex labels*

Table 2. Number of Graceful labelings of $K_{1,m-1} \oplus C_4$, $3 \le m \le 15$.

Unicyclic Graph	Label i 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Total
$K_{1,2} \oplus C_4$	0	4	8	2	8	4	0													**26**
$K_{1,3} \oplus C_4$	0	7	7	6	6	7	7	0												**40**
$K_{1,4} \oplus C_4$	0	6	6	8	8	8	6	6	0											**48**
$K_{1,5} \oplus C_4$	0	6	4	5	9	9	5	4	6	0										**48**
$K_{1,6} \oplus C_4$	0	6	7	3	10	4	10	3	7	6	0									**56**
$K_{1,7} \oplus C_4$	0	8	5	5	9	8	8	9	5	5	8	0								**70**
$K_{1,8} \oplus C_4$	0	7	8	4	12	5	10	5	12	4	8	7	0							**82**
$K_{1,9} \oplus C_4$	0	9	7	7	11	10	9	94	10	11	7	7	9	0						**106**
$K_{1,10} \oplus C_4$	0	9	10	4	16	5	15	4	15	5	16	4	10	9	0					**122**
$K_{1,11} \oplus C_4$	0	10	8	7	14	8	11	11	11	11	8	14	7	8	10	0				**138**
$K_{1,12} \oplus C_4$	0	10	12	5	19	6	15	6	16	6	15	6	19	5	12	10	0			**162**
$K_{1,13} \oplus C_4$	0	12	10	8	17	11	13	11	13	13	11	13	11	17	8	10	12	0		**190**
$K_{1,14} \oplus C_4$	0	11	12	8	22	7	20	5	20	6	20	5	20	7	22	8	12	11	0	**216**

Table 3. Labels of central vertex of the star ($K_{1,m-1} \oplus C_4$ with $3 \le m \le 15$).

Unicyclic graph	Label i 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$K_{1,2} \oplus C_4$	•	•	•	•	•	•	•												
$K_{1,3} \oplus C_4$	•	•	•	•	•	•	•	•											
$K_{1,4} \oplus C_4$	•	•	•	•	•	•	•	•	•										
$K_{1,5} \oplus C_4$	•	•	•	•			•	•	•	•									
$K_{1,6} \oplus C_4$	•	•	•	•				•	•	•	•								
$K_{1,7} \oplus C_4$	•	•	•	•					•	•	•	•							
$K_{1,8} \oplus C_4$	•	•	•	•						•	•	•	•						
$K_{1,9} \oplus C_4$	•	•	•	•							•	•	•	•					
$K_{1,10} \oplus C_4$	•	•	•	•								•	•	•	•				
$K_{1,11} \oplus C_4$	•	•	•	•									•	•	•	•			
$K_{1,12} \oplus C_4$	•	•	•	•										•	•	•	•		
$K_{1,13} \oplus C_4$	•	•	•	•											•	•	•	•	
$K_{1,14} \oplus C_4$	•	•	•	•												•	•	•	•

that are the end points for that edge have labels $m+x+y$ *and* y*, where* $0 \leq x+y \leq 3$*. Now suppose, on the contrary, that the central vertex has a label that belongs to the set* $\{4, 5, \cdots, m-1\}$*. Then none of the edges with labels* $m + x$ *(*$0 \leq x \leq 3$*) can be edges on the star. In other words, these edge labels are on the edges of* C_4*. Hence the edges of the star have labels* $1, 2, \cdots, m-1$*. If the central vertex has a label in the set* $\{5, \cdots, m-2\}$*, then the edge label* $m-1$ *is impossible since the only vertex labels that achieve this are* $m-1+z$ *and* z *for* $0 \leq z \leq 4$*. Hence the central vertex must have label* 4 *or* $m-1$*.*

Since these are complementary labels in f *and* $\overline{f}$*, it is enough to consider one of them. So assume that the central label is 4. Then to achieve the edge label* $m-1$*, the common vertex must have label* $m+3$*. This forces the label* 0 *on a cycle vertex adjacent to the common vertex. Also to achieve the label* $m-2$ *on an edge of the star, a peripheral vertex must have label* $m+2$*. This in turn forces the labels* 1 *and* $m+1$ *on the remaining two vertices of the cycle, with* 1 *adjacent to* $m+3$*. Since the only ways to achieve edge label* $m-3$ *on the star is to have a peripheral vertex label* 1 *or* $m+1$*, we get a contradiction. This proves the result.*

An easy generalization of the above argument leads to the following general result, which we state below. We omit the proof.

Theorem 3. *For* $n \geq 4$*, for* $m \geq n+2$ *and for any graceful labeling of* $K_{1,m-1} \oplus C_n$*, the central vertex cannot have a label in the set* $\{n, n+1, \cdots, m-1\}$*.*

4. Conclusion

An attempt of generalization of the algorithm of Jay Bagga et al. [1] brought us to introduce a new algorithm which enumerates graceful labelings of graceful unicyclic graphs $K_{1,m-1} \oplus C_4$, a star $K_{1,m-1}$ with m vertices sharing a common vertex with the cycle C_4. For this algorithm which is linked to the structure of $K_{1,m-1} \oplus C_4$, there are three starting points, a vertex in the cycle, a vertex in the star and the common vertex. The different starting points ensure that the common vertex can have any label from the set of labels. We implemented our algorithm to enumerate graceful labelings of unicyclic graphs $K_{1,m-1} \oplus C_4$, $3 \leq m \leq 15$. Experimental results illustrate that the values of the label of the common vertex belong to the set $\{0, 1, 2, 3, 4, n-3, n-2, n-1, n\}$.

In our future work, we will seek to derive general characteristics of graceful unicyclic graphs which could lead to a more general proof of the conjecture of Truszczynski.

References

[1] Jay Bagga, Adrian Heinz, M. Mahbubul Majumber, An Algorithm for Graceful Labelings of Cycles, Congressus Numerantium 186 (2007), 57-63.

[2] Jay Bagga, Adrian Heinz, M. Mahbubul Majumber, Properties of Graceful Labelings of Cycles, Congressus Numerantium, 188 (2007), 109-115.

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

[4] G. Chartrand and L. Lesniak, Graphs & Digraphs, Chapman & Hall CRC, New York, (1996).

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

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

[7] C. Sekar, Studies in Graph Theory, Ph. D. Thesis, Madurai Kamaraj University, 2002.

[8] M. Truszczynski, Graceful unicyclic graphs, Demonstatio Mathematica, 17 (1984) 377-387.

[9] S. L. Zhao, All trees of diameter four are graceful, Graph Theory and its Applications: East and West (Jinan, 1986), 700-706, Ann. New York Acad. Sci., 576, New York Acad. Sci., New York, 1989.