11-2015

Tight Super-Edge-Graceful Labelings of Trees and Their Applications

Alex Collins  
*Georgia State University*, acollins38@gsu.edu

Colton Magnant  
*Georgia Southern University*, cmagnant@georgiasouthern.edu

Hua Wang  
*Georgia Southern University*, hwang@georgiasouthern.edu

Follow this and additional works at: [https://digitalcommons.georgiasouthern.edu/math-sci-facpubs](https://digitalcommons.georgiasouthern.edu/math-sci-facpubs)

Part of the [Mathematics Commons](https://digitalcommons.georgiasouthern.edu/math-sci-facpubs/584)

Recommended Citation

[https://digitalcommons.georgiasouthern.edu/math-sci-facpubs/584](https://digitalcommons.georgiasouthern.edu/math-sci-facpubs/584)

This article is brought to you for free and open access by the Mathematical Sciences, Department of at Digital Commons@Georgia Southern. It has been accepted for inclusion in Mathematical Sciences Faculty Publications by an authorized administrator of Digital Commons@Georgia Southern. For more information, please contact digitalcommons@georgiasouthern.edu.
Tight super-edge-graceful labelings of trees and their applications

Alex Collins\textsuperscript{a}, Colton Magnant\textsuperscript{b}, Hua Wang\textsuperscript{b,∗}

\textsuperscript{a} Department of Mathematics and Statistics, Georgia State University, Atlanta, GA 30303, USA
\textsuperscript{b} Department of Mathematical Sciences, Georgia Southern University, Statesboro, GA 30460, USA

Received 15 September 2012; accepted 23 July 2014
Available online 3 December 2015

Abstract

The concept of graceful labeling of graphs has been extensively studied. In 1994, Mitchem and Simoson introduced a stronger concept called super-edge-graceful labeling for some classes of graphs. Among many other interesting pioneering results, Mitchem and Simoson provided a simple but powerful recursive way of constructing super-edge-graceful trees of odd order. In this note, we present a stronger concept of "tight" super-edge-graceful labeling. Such a super-edge graceful labeling has an additional constraint on the edge and vertices with the largest and smallest labels. This concept enables us to recursively construct tight super-edge-graceful trees of any order. As applications, we provide insights on the characterization of super-edge-graceful trees of diameter 4, a question posed by Chung, Lee, Gao and Schaffer. We also observe infinite families of super-edge-graceful trees that can be generated from tight labelings. Given the direct applications of "tight" super-edge-graceful labeling to the study of super-edge-graceful labelings, we note that it is worthwhile to further examine recursively generated tight super-edge-graceful trees.

Keywords: Graceful labeling; Super-edge-graceful; Trees

1. Introduction

A finite, simple and undirected graph $G$ with $p = |V(G)|$ and $q = |E(G)|$ is called edge-graceful if there are onto functions

$$l : E(G) \to \{1, \ldots, q\}$$

and

$$l^* : V(G) \to \{1, \ldots, p\}$$

such that

$$l^*(v) = \sum_{uv \in E(G)} l(uv) \mod p.$$

Peer review under responsibility of Kalasalingam University.

* Corresponding author.

E-mail addresses: acollins38@gsu.edu (A. Collins), cmagnant@georgiasouthern.edu (C. Magnant), hwang@georgiasouthern.edu (H. Wang).

http://dx.doi.org/10.1016/j.akcej.2015.11.004
0972-8600/© 2015 Kalasalingam University. Production and Hosting by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
When it exists, such a labeling is called an edge-graceful labeling. Introduced by Lo [1] in 1985, this concept has been extensively studied. In particular, many classes of trees of odd orders were shown to be edge-graceful [2–6] as partial results towards the following conjecture:

**Conjecture 1.1** ([7]). All odd-order trees are edge-graceful.

In [3], all trees of odd order of diameter at most four have been shown to be edge-graceful. The concept of super-edge-graceful graphs was introduced in [4].

**Definition 1.** A graph $G$ of order $p$ and size $q$ is super-edge-graceful (SEG) if there exists a bijection $l$ from $E(G)$ to

\[
\left\{ 0, \pm 1, \ldots, \pm \left\lfloor \frac{q}{2} \right\rfloor \right\} \quad \text{if } q \text{ is odd}
\]

and

\[
\left\{ \pm 1, \ldots, \pm \left\lfloor \frac{q}{2} \right\rfloor \right\} \quad \text{if } q \text{ is even}
\]

such that the induced vertex labeling $l^*$

\[
l^*(v) = \sum_{uv \in E(G)} l(uv)
\]

is a bijection from $V(G)$ to

\[
\left\{ 0, \pm 1, \ldots, \pm \left\lfloor \frac{p}{2} \right\rfloor \right\} \quad \text{if } p \text{ is odd}
\]

and

\[
\left\{ \pm 1, \ldots, \pm \left\lfloor \frac{p}{2} \right\rfloor \right\} \quad \text{if } p \text{ is even}.
\]

When it exists, such a labeling is called a super-edge-graceful labeling (SEGL). For trees, it is shown in [4] that SEG implies edge-graceful. Also in [4], the following simple but extremely useful “growing tree algorithm” was provided.

**Proposition 1.2** ([4]). By appending two edges to the same vertex of a SEG tree of odd order, one obtains a new tree that is also SEG.

For a list of the work done on various graph labelings, we recommend the dynamic survey [8]. As a related question to edge-graceful trees, the following was posed in [9]:

**Question 1.1** ([9]). Characterize trees of diameter 4 that are SEG.

All trees of odd order with three even vertices are shown to be SEG in [10]. Most recently, [11] provided explicit SEGL for caterpillars (trees whose removal of leaves results in a path) and some lobsters (trees whose removal of leaves results in a caterpillar) of diameter 4.

In this note, we present a generalization of SEGL that we call a tight super-edge-graceful labeling (tight SEGL). The precise definition of a tight SEGL is provided in Section 2, where we also show a generalized version of Proposition 1.2 that applies to trees of even order. As applications, we show, in Section 3, how Question 1.1 can be reduced to simpler trees through the utilization of tight SEGL. We also make some modest progress towards some open questions from [11] through this approach. In Section 4, we briefly mention how our recursive construction can be used to generate an infinite family of trees (of even order) that have tight SEGL.

2. Tight labeling and the “growing tree algorithm” for even ordered trees

Towards our goal of providing a generalization of Proposition 1.2 for trees of even order, we first define the tight SEGL for trees of even order.
**Definition 2.** A SEGL of a tree $T$ of even order is said to be "tight" if there exist $e = ab, f = cd \in E(T)$ such that

$$l(e) = \left\lfloor \frac{q}{2} \right\rfloor$$

and

$$l(f) = -\left\lfloor \frac{q}{2} \right\rfloor$$

with the vertex labelings

$$\{l^*(a), l^*(b)\} = \left\{ \left\lfloor \frac{p}{2} \right\rfloor, \left\lfloor \frac{p}{2} \right\rfloor - 1 \right\}$$

and

$$\{l^*(c), l^*(d)\} = \left\{ -\left\lfloor \frac{p}{2} \right\rfloor, -\left\lfloor \frac{p}{2} \right\rfloor + 1 \right\}.$$
3. On trees of diameter 4

We restrict our attention to Question 1.1 and trees of diameter four in this section.

First of all, it is easy to see that a tree of diameter four is either a caterpillar or a lobster [11]. As nicely demonstrated in [11], caterpillars of diameter four can be represented as a three-tuple \((x, y, z)\) (Fig. 3).

If such a caterpillar is of even order, then \(x + y + z\) is odd. Take, for example, the case of \(x \geq 1, y \geq 3\) and \(z \geq 3\) each of which being odd. Such a tree can be easily reduced to \((x, y, z) = (1, 3, 3)\) by repeatedly removing two adjacent pendant edges. Fig. 4 displays a tight SEGL of such a tree. Then by applying Proposition 2.1, one can generate a tight SEGL (and hence SEGL) for any tree of this kind.

Remark 1. This approach of “growing tree algorithm” through Proposition 2.1 (for even ordered trees) and Proposition 1.2 (for odd ordered trees) can be applied to all similar cases. We omit the details since explicit labelings were presented in [11], in which the set of all caterpillars of diameter four that possess a SEGL are characterized.

Given any lobster of diameter 4, by following the same approach, the tree can be reduced to a spider (a tree with exactly one vertex of degree \(> 2\)) with legs of length 1 or 2 (Fig. 5).

Unfortunately, such graphs do not necessarily yield a SEGL or tight SEGL. However, by adding a few edges, one can obtain some “semi-elementary” trees that do have a tight SEGL. See, for instance, Fig. 6.

In [11], the explicit labelings of many lobsters of diameter 4 are presented. We would like to mention that they can also be generated from elementary cases, such as Fig. 6, through the “growing tree algorithms”. For the few remaining cases, there are descriptions in Conjectures 3.12–3.14 in [11]. We skip the details of their descriptions but point out that the questions from Conjectures 3.13 and 3.14 can be reduced to the following simpler structure but stronger statement.

Question 3.1. Consider a tree \(T\) of height 2 rooted at \(v\) with \(k + 2\) children \(v_0, v_1, \ldots, v_k, u\) such that \(v_0\) is a leaf, \(v_i\) is of degree 2 for \(1 \leq i \leq k\) and \(u\) is of degree 4. Does there always exist a tight SEGL of \(T\) for \(k \geq 2\)? For example, Fig. 6 shows such a tree with \(k = 2\).
Remark 2. A positive answer to Question 3.1 will imply a tight SEGL for any tree generated from them through the “growing tree algorithm”, among them the ones conjectured in [11].

We have yet to find a general labeling for such trees. However, computer simulations yield a positive answer to Question 3.1 for all small values of \( k \). With such evidence at hand, we conjecture that this question can be answered positively.

In an analogous way, Conjecture 3.12 of [11] can be reduced to the following stronger statement on much simpler structures.

Question 3.2. Consider a tree \( T \) of height 2 rooted at \( v \) with \( 2k + 1 \) children \( v_0, v_1, \ldots, v_{2k} \) such that \( v_0 \) is of degree 3 and \( v_i \) is of degree 2 for \( 1 \leq i \leq 2k \). Does there always exist a tight SEGL of \( T \) for \( k \geq 3 \)? For example, Figs. 7 and 8 show such trees with \( k = 3, 4 \).

Remark 3. When \( k = 1 \) in Question 3.2, it can be seen that such a tree does not have a SEGL. When \( k = 2 \), a SEGL can be easily found but no tight SEGL exists.

4. Trees with tight super-edge-graceful labelings

The “growing tree algorithm” can easily generate infinite families of tight SEG trees of even order. For instance, every tree shown in Figure 3 of [10] of order 8 that has a SEGL also has a tight SEGL.

By applying Proposition 2.1 to these trees, we easily get various trees with tight SEGLs. It seems to be an interesting question to try to provide some characterization of tight SEG trees generated through this venue.
Acknowledgment

This work was partially supported by a grant from the Simons Foundation (#245307 to Hua Wang).

References