2022 HAWAII UNIVERSITY INTERNATIONAL CONFERENCES
SCIENCE, TECHNOLOGY & ENGINEERING, ARTS, MATHEMATICS & EDUCATION JUNE 7 - 9, 2022
PRINCE WAIKIKI RESORT, HONOLULU, HAWAII

ON K-CORDIAL LABELING

ZENG, HONG BIAO
DEPARTMENT OF COMPUTER SCIENCE
FORT HAYS STATE UNIVERSITY
HAYS, KANSAS

On k-Cordial Labeling

Hong Biao Zeng

Department of Computer Science
Fort Hays State University
Hays, KS 67601

Abstract

In this paper we introduce a concept so called k-cordial labeling to generalize the concept of cordial labeling. We establish the connection between k-cordial labeling and Fibonacci cordial labeling. We propose a faster brute force search algorithm to check the existence of Fibonacci cordial labeling. Finally, we extend the results to Tribonacci cordial labeling.

1 Introduction

In this paper, we only consider finite simple connected undirected graphs. We denote the vertex set and the edge set of a graph G as V(G) and E(G) respectively. We denote the size of set S as $|S|$. We adopt the definitions of cordial labeling and Fibonacci cordial labeling from [1] and [2] as follows.

Definition 1.1. *A function $f : V(G) \longrightarrow \{0, 1\}$ is called a binary vertex labeling of a graph G and f(v) is called label of the vertex v of G under f. For an edge $e = uv$, the induced edge labeling $f^* : E(G) \longrightarrow \{0, 1\}$ is given by $f^*(e) = |f(u) - f(v)|$. We denote $v_f(i)$, i = 0 or 1 as the number of vertices with label i; we denote $e_f(i)$, i = 0 or 1 as the number of edges with induced label i. (see [1] or [2])*

Definition 1.2. *A binary vertex labeling f of a graph G is called cordial labeling if $|v_f(0) - v_f(1)| \leq 1$ and $|e_f(0) - e_f(1)| \leq 1$. A graph G is cordial if it admits cordial labeling.*

Definition 1.3. *An injective function $f : V(G) \longrightarrow \{F_0, F_1, F_2, ..., F_n\}$ where F_j is the jth Fibonacci number (j = 0, 1, 2, ..., n) is said to be Fibonacci cordial labeling if the induced function $f^* : E(G) \longrightarrow \{0, 1\}$ defined by $f^*(uv) = (f(u) + f(v)(mod2)$ satisfied the condition $|e_f(0) - e_f(1)| \leq 1$. A graph which admits Fibonacci cordial labeling is called Fibonacci cordial graph.*

In Definition 1.3 above, the Fibonacci numbers are defined by the linear recurrence relation $F_n = F_{n-1} + F_{n-2}, n \geq 2$ with initial conditions $F_0 = 0, F_1 = 1$.

Cahit [2] introduced the concept of cordial labeling. Rokad and Ghodasara [1] introduced the concept of Fibonacci Cordial Labeling.

An important remark: Although in Definition 1.3, it is not stated that $n = |V(G)|$, however, in all theorems and examples given in [1], $n = |V(G)|$. So the number of Fibonacci numbers is just one more than the number of vertices of the graph. To avoid confusion, we redefine Fibobacci cordial labeling as follows in the rest of this paper.

Definition 1.4. *An injective function $f : V(G) \longrightarrow \{F_0, F_1, F_2, ..., F_n\}$ where F_j is the jth Fibonacci number (j = 0, 1, 2, ..., n, $n = |V(G)|$) is said to be Fibonacci cordial labeling if the induced function $f^* : E(G) \longrightarrow \{0, 1\}$ defined by $f^*(uv) = (f(u) + f(v)(mod2)$ satisfied the condition $|e_f(0) - e_f(1)| \leq 1$. A graph which admits Fibobacci cordial labeling is called Fibonacci cordial graph.*

Comparing with Definition 1.3 cited from [1], we added one more condition $n = |V(G)|$ in Definition 1.4. All theorems and examples in paper [1] still hold with this new definition of Fibonacci Cordial Labeling.

We introduce a new concept so called k-cordial labeling as follows.

Definition 1.5. *A binary vertex labeling of a graph G is called k-cordial labeling if $|v_f(0) - v_f(1)| = k$ and $|e_f(0) - e_f(1)| \leq 1$, here k is a non-negative integer. A graph G is k-cordial if it admits k-cordial labeling.*

By the definition, a cordial labeling is just a special case of k-cordial labeling for $k = 0$ or $k = 1$.

2 Main Result

The following lemma states an obvious fact of Fibonacci numbers.

Lemma 2.1. *Given the set of the first $n+1$ Fibonacci numbers $S = \{F_0, F_1, ..., F_n\}$, if we denote e_n as number of evens in S and o_n as number of odds in S, then*

$$o_n - e_n = \begin{cases} m-1 & \text{if } n = 3m \\ m & \text{if } n = 3m+1 \\ m+1 & \text{if } n = 3m+2 \end{cases} \qquad (1)$$

Proof. It is obvious that Fibonacci numbers have pattern E, O, O, E, O, O, E, ..., where E stands for even and O stands for odd. The lemma can be easily verified. □

Theorem 2.2 (Main Theorem)**.** *A graph G of n vertices is a Fibonacci cordial graph if and only if G is a k-cordial graph where*

$$k = i \ where \begin{cases} i \in \{m-2, m\}, & if\ n = 3m \\ i \in \{m-1, m+1\}, & if\ n = 3m+1 \\ i \in \{m, m+2\}, & if\ n = 3m+2 \end{cases} \tag{2}$$

Proof. By definition, a graph G of n vertices is a Fibonacci cordial graph if and only if we can label n vertices with n Fibonacci numbers chosen from set $S = \{F_0, F_1, ..., F_n\}$. Hence, there is one and only one Fibonacci number in set S that will not be used for labeling. So the difference between number of vertices labeled with even Fibonacci numbers and the number of vertices labeled with odd Fibonacci numbers will be one away from $o_n - e_n$ in Lemma 2.1.

Given a Fibonacci cordial labeling of a graph, we relabel the graph as follows: if a vertex is labeled by an odd Fibonacci number, then we relabel it to 1; if a vertex is labeled by an even Fibonacci number, then we relabel it to 0. This new labeling will not change the induced labeling of edges. Hence, we get a k-cordial labeling, here k is defined in equation 2.

On the other hand, if we are given a k-cordial labeling where k is defined in equation (2), then we can relabel all vertices whose labels are 1 by mutually distinct odd Fibonacci numbers from set S and relabel all vertices whose labels are 0 by mutually distinct even Fibonacci numbers from set S. If we do so, the induced labels of edges will not change and hence the new labeling is a Fibonacci cordial labeling. □

The Theorem 2.2 has an important application which we will address in the next section.

3 Application of Main Theorem

Although there are many papers, such as [3, 4, 5], which studied algorithms to search for certain labeling for an arbitrary graph, these algorithms are essentially brute force search algorithms. If we use brute force algorithm to search for Fibonacci cordial labeling, the worst case (if the labeling not exists) running time will be $O(m(n+1)!)$ for a graph with n vertices and m edges since we have $(n+1)!$ ways to label vertices and for each labeling, we need to calculate induced labeling for m edges.

Theorem 3.1. *The worst case running time to search for a k-cordial labeling of a graph with n vertices and m edges is $O(mC(n, \frac{n+k}{2}))$. Here C(a, b) is combination "a choose b".*

Proof. Assume that we have vertices $v_1, v_2, ..., v_n$. Notice that if we change all labels of 0 to 1 and all labels of 1 to 0 simultaneously, the induced labeling will not be changed. So without loss of generality, we can assume that we have k more vertices that are labeled with 1 than 0. Assume that there are x vertices that are labeled with 1, then we have $x-k$ vertices that are labeled with 0. Hence $(x-k)+x=n$. Therefore, $x=\frac{n+k}{2}$. Notice that under the assumption, $\frac{n+k}{2}$ is an integer.

To find a labeling, we first assign x many ones and $n-x$ many zeros to n vertices. There are $C(n,x)$ many different ways to do so. Then for each assigned labeling, we calculate induced labeling of m edges and check if the condition $|e_f(0)-e_f(1)|\leq 1$ holds. The worst running time hence is $O(mC(n,x))$. □

For k-cordial labeling that corresponds to Fibonacci cordial labeling of graph G, k is about $\frac{|V(G)|}{3}$ by Theorem 2.2. However, we need to check out for two values of k, according to Theorem 2.2. If we take the ceiling to get control from above, the running time shall be $O(2mC(n,\lceil\frac{n}{3}\rceil))$ where n = V(G) and m = E(G). Since both running times have m = E(G) as factor, we only compare $(n+1)!$ and $2C(n,\lceil\frac{n}{3}\rceil)$ to see the difference of running time. It is easy to see that $\lim_{n\to\infty}\frac{2C(n,\lceil\frac{n}{3}\rceil)}{(n+1)!}=0$. The following table shows such a comparison.

n	$(n+1)!$	$2C(n,\lceil\frac{n}{3}\rceil)$
8	362880	112
9	3628800	168
10	39916800	420
11	479001600	660
12	6227020800	990

In the proof of the main theorem 2.2, we already see how to get a Fibonacci cordial labeling from a k-cordial labeling with appropriate value of k. Since the running time to get a k-cordial labeling is much faster, instead of searching for a Fibonacci cordial labeling directly, we search for a k-cordial labeling with value k defined in equation (2). The same idea shall be easily extended to Tribonacci cordial labeling.

4 Extension of Main Result

We start with the definition of Tribonacci numbers.

Definition 4.1. *Tribonacci numbers are defined by recurrence relation* $T_n=T_{n-1}+T_{n-2}+T_{n-3}$ *with initial conditions* $T_0=0,T_1=0,T_3=1$.

Tribonacci numbers are just like Fibonacci numbers except instead of starting with two initial values, the Tribonacci sequence starts with three initial values and each term afterwards is the sum of three preceding terms. The first few Tribonacci numbers are 0, 0, 1, 1, 2, 4, 7, 13, 24, 44, 81, 149, 274, 504, 927, 1705.

Now we can similarly define Tribonacci cordial labeling.

Definition 4.2. *An injective function* $f : V(G) \longrightarrow \{T_0, T_1, T_2, ..., T_n\}$ *where* T_j *is the jth Tribonacci number (j = 0, 1, 2, ..., n,* $n = |V(G)|$*) is said to be Tribonacci cordial labeling if the induced function* $f^* : E(G) \longrightarrow \{0, 1\}$ *defined by* $f^*(uv) = (f(u) + f(v))(mod2)$ *satisfied the condition* $|e_f(0) - e_f(1)| \leq 1$*. A graph which admits Tribonacci cordial labeling is called Tribonacci cordial graph.*

We have the following lemma for Tribonacci numbers which is similar to Lemma 2.1

Lemma 4.1. *Given the set of the first n+1 Tribonacci numbers* $S = \{T_0, T_1, ..., T_n\}$*, if we denote* e_n *as number of evens in S and* o_n *as number of odds in S, then*

$$e_n - o_n = \begin{cases} 1 & \text{if } n = 4m \\ 2 & \text{if } n = 4m+1 \\ 1 & \text{if } n = 4m+2 \\ 0 & \text{if } n = 4m+3 \end{cases} \tag{3}$$

We have the following theorem, which is similar to Theorem 2.2, for Tribonacci cordial labeling.

Theorem 4.2. *A graph G of n vertices is a Tribonacci cordial graph if and only if G is a k-cordial graph where*

$$k = i \text{ where} \begin{cases} i \in \{0, 2\}, & \text{if } n = 4m \\ i \in \{1, 3\}, & \text{if } n = 4m+1 \\ i \in \{0, 2\}, & \text{if } n = 4m+2 \\ i \in \{1\}, & \text{if } n = 4m+3 \end{cases} \tag{4}$$

Proof. For a graph G of n vertices, we have $n+1$ Tribonacci numbers to do the labeling. One of these $n+1$ Tribonacci numbers will not be used since we only have n vertices. Therefore, the difference between numbers of vertices labeled with even Tribonacci numbers and numbers of vertices labeled with odd Tribonacci numbers will be one away from the numbers in equation (3) in Lemma 4.1.

Given a Tribonacci cordial labeling, if we replace the labels of odd Tribonacci numbers with 1 and the labels with even Tribonacci numbers with 0, then we will not change the induced labeling of edges. Hence, we get a k-cordial labeling where k is defined in equation (4).

On the other hand, if we are given a k-cordial labeling where k is defined in this theorem, then we can relabel all vertices whose labels are 1 by mutually distinct odd Tribonacci numbers from set $S = \{T_0, T_1, ..., T_n\}$ and relabel all vertices whose labels are 0 by mutually distinct even Tribonacci numbers from set S. If we do so, the induced labels of edges will not change and hence the new labeling is a Tribonacci cordial labeling. □

If we search for a Tribonacci cordial labeling solution of a graph with n vertices and m edges using brute force algorithm, the worst case running time is $O(m(n+1)!)$. Just like we search a k-cordial labeling to find a Fibonacci labeling, we can search at most two of a 0, 1, 2 or 3-cordial labeling to find a Tribonacci labeling. The worst running time for finding two k-cordial labeling (k = 0, 1, 2, 3) is at most $2mO(C(n, \lceil \frac{n}{2} \rceil)$. Since m are common factor in running time, we only compare $(n+1)!$ with $2C(n, \lceil \frac{n}{2} \rceil)$. It is easy to see that $\lim_{n\to\infty} \frac{2C(n, \lceil \frac{n}{2} \rceil)}{(n+1)!} = 0$. The following table is a comparison for running time.

n	$(n+1)!$	$2C(n, \lceil \frac{n}{2} \rceil)$
8	362880	140
9	3628800	252
10	39916800	504
11	479001600	924
12	6227020800	1848

Again, instead of searching for a Tribonacci cordial labeling, we shall search for a k-cordial with appropriate value k. Then we transform this k-cordial labeling into a Tribonacci cordial labeling as we did in the proof of Theorem 4.2

5 Conclusion

We defined k-cordial labeling. We established the connections between k-cordial labeling and Fibonacci cordial labeling. We then extended the work to establish the connection between k-cordial labeling and Tribonacci cordial labeling.

By introducing and applying the concept of k-cordial labeling, we can expect better running time on searching for a Fibonacci cordial labeling and for a Tribonacci cordial labeling.

Future work can be to establish the connection between k-cordial labeling and m-bonacci cordial labeling. Intuitively, we can see that the similar result shall hold. We will address this topic somewhere else.

References

[1] A.H.Rokad and G.V.Ghodasara. Fibonacci cordial labeling of some special graphs. *Annals of Pure and Applied Mathematics*, 11(1):133–144, 2016.

[2] Cahit. Cordial graphs, a weaker version of graceful and harmonic graphs. *Ars Combinatoria*, 23:201–207, 1987.

[3] Parham Azimi Kourosh Eshghi. Applications of mathematical programming in graceful labeling of graphs. *Journal of Applied Mathematics*, pages 1–8, 2004.

[4] Auparajita Krishnaa. Computer modelling of graph labellings. *proc. National Conference on Mathematical and Computational Models*, pages 293–301, 2001.

[5] Auparajita Krishnaa. On the use of computers in graph labeling. *International Journal of Computer Science and Communication*, 3(1):191–197, 2012.