COUNTING RANDOM WALK LABELINGS OF GRAPHS

SARAH HE, JOHN C. WIERMAN; JOHNS HOPKINS UNIVERSITY

INTRODUCTION

Consider a simple, undirected, connected graph G on n vertices. A graph labeling is an assignment of labels, which are usually integers, to either the vertices of a graph, the edges, or both. Common labeling methods include graceful labelings, harmonious labelings, magic labelings, and anti-magic labelings [5].

Fang et. al. [1] first introduced the concept of a random walk labeling under the name successive vertex ordering, where a successive vertex ordering is defined to be a linear ordering of the vertices such that for every *i*, the first *i* vertices induce a connected subgraph of G. Fried and Mansour [3] coined the term for such a process as random walk labelings when they calculated the number of random walk labelings for the king's graph and on a $2 \times n$ grid graph. For the latter, the expression involved sums of inverses of binomial coefficients, leading to new combinatorial identities and confirming a conjecture of Bala. Fried and Mansour [2] then studied random walk labelings of the wheel, fan, barbell, lollipop, tadpole, friendship, and snake graphs, proving various combinatorial identities. They then calculated the number of random walk labelings of perfect trees, combs, and double combs, the torus $C_2 \times C_n$ [4].

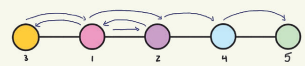
Our motivations for the study of random walk labelings for the purpose of discovering new integer sequences in the Online Encyclopedia of Integer Sequences (OEIS) as well as for deriving new combinatorial identities and interpretations for entries already existing in the OEIS.

In particular, we begin by examining the total number of random walk labelings for a caterpillar graph with two-legs, and expand this method to other graph families by considering modifications on the backbone of the caterpillar and mapping any labeling to that of a caterpillar.

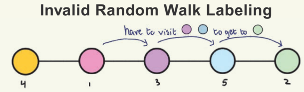
RANDOM WALK LABELINGS

What is a random walk labeling? As the name suggests, random walk labelings are based on random walks, wherein a walker can walk randomly as long as they are traveling along the edges of the graph. Each previously unvisited vertex is subsequently labeled in increasing order until all vertices are labeled.

Valid Random Walk Labeling

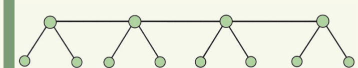


Invalid Random Walk Labeling

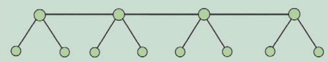


CATERPILLAR GRAPHS

We first consider random walk labelings of caterpillar graphs. A caterpillar graph is one where there is a path of *n* vertices, called the "backbone," with additional vertices connected to the backbone by edges called "legs." Each leg vertex is connected to the backbone through a single edge and cannot connect to multiple backbone vertices. That is, each leg vertex is distance 1 from the central path. Additionally, leg vertices cannot connect to vertices attached to any other backbone vertices, i.e. each leg vertex is degree 1.



STANDARD CATERPILLAR



Labelings Starting from the Endpoint of the Backbone

$$t_n = \frac{(n(j+1)-1)!}{(j+1)^{n-1}(n-1)!}$$

Each backbone vertex has *j* corresponding legs, thus there are *(j+1)n* vertices in total to label, with the first one labeled being the endpoint. Of the remaining $(j+1)n - 1$ vertices, if there were no restrictions there would be *((j+1)n-1)!* ways to label them; however we have the restriction that legs cannot be labeled until after the corresponding backbone vertex has been labeled. This means that for each group of backbone and leg vertices, only *1/(j+1)* of the permutations are valid in our sequence, meaning that *1/((j+1)n−1)* of the permutations are valid. Similarly, the backbone vertices must be labeled in increasing order, so only one out of *(n − 1)!* permutations of the remaining unlabeled backbone vertices is valid.

Total Labelings

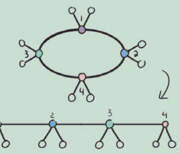
When we start at the backbone at the *k-th* vertex (or one of the legs), we can partition the caterpillar into the left and right caterpillars, with the spit at *k*, and apply t_n. Observe that if we start the labeling at one of the legs of the *k*-th backbone vertex, necessarily this backbone vertex is labeled second. Across all possible starting vertices, we have

$$\sum_{k=1}^{n}[\binom{3n-1}{3k-1} \cdot t_k \cdot t_{n--k} + 2 \cdot \binom{3n-2}{1, 3k-3, 3n-3k} \cdot t_{k--1} \cdot t_{n--k}]$$

total random walk labelings. After some algebraic manipulation, this simplifies to

$$\frac{2^{n-1} \cdot (3n+1)!}{3^n \cdot n! \cdot (3n-1)}$$

GENERAL METHOD

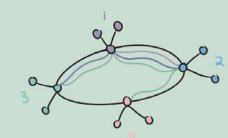


We can generalize the method for caterpillar graphs to other graphs. Consider a graph on n vertices with a central structure--henceforth referred to as the backbone-- with a total of B_n random walk labelings. If we attach *j* legs to each vertex of the central structure, the total number of random walk labelings is:

$$B_n \cdot (t_{n,j} + j \cdot ((j+1)n-2)_{j-1} \cdot t_{n-1,j})$$

The derivation is as follows. Consider any arbitrary random walk labeling of a "caterpillar" graph that starts along the backbone. If we look at just the label of the backbone vertices of each segment, and place the segments in increasing order of such labels, we effectively map the graph to a standard caterpillar structure. Visually, we can consider an unraveling of the random walk performed.

If we instead had a labeling that started at one of the legs, we can similarly map the structure to that of the standard caterpillar by increasing order of the backbone labels. Note that by the structure of legs, we know that if the first label is assigned to a leg, the second vertex labeled is necessarily the corresponding backbone vertex. The remaining legs of this first segment can then be labeled with any of the remaining labels.



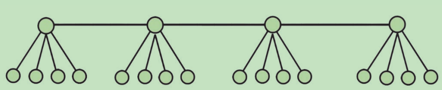
J-LEGGED CATERPILLAR

Similar to the two-legged caterpillar, we have for random walk labelings of the backbone [3]:

$$B_n = 2^{n-1}$$

Note that if we have *j* legs instead of 2, we modify t_n slightly to reflect the total *n(j+1)* vertices and *(j+1)* vertices within each segment, such that

$$t_{n,j} = \frac{(n(j+1)-1)!}{(j+1)^{n-1}(n-1)!}$$

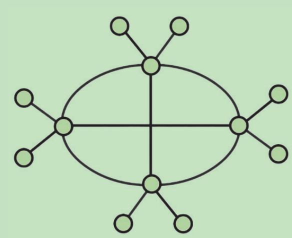


4-legged caterpillar with backbone of length 4

COMPLETE GRAPH

For a complete graph, as shown by Sela and Mansour [3], all backbone vertices are accessible from any other backbone vertex, therefore

$$B_n = n!$$

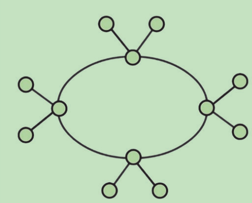


2-legged caterpillar with complete graph backbone of length 4

CYCLE

Note that there are *n* segments (or backbone vertices). Starting at any backbone vertex, we can at each step consider moving to the next unlabeled backbone vertex on the left or on the right, until we reach the last backbone vertex to be labeled [3].

$$B_n = n \cdot 2^{n-2}$$



2-legged caterpillar with cycle backbone of length 4

NUMERICAL RESULTS AND FUTURE DIRECTIONS

- For the standard two-legged caterpillar, t_n matches the integer sequence generated by OEIS A052502, which counts the number of permutations σ of [3n] without fixed points such as σ^3=Id.
- All other integer sequences derived here do not appear in the OEIS, meaning they are new integer sequences to be added.
- We can also derive the generating functions for our sequences
- Instead of modifications on the backbone, we may consider modifications on the legs
    - if legs could be connected to each other
    - if we attach more pendant vertices to each leg, e.g. having a path as a leg

REFERENCES

[1] Lixing Fang, Hao Huang, J´anos Pach, G´abor Tardos, and Junchi Zuo. Successive vertex orderings of fully regular graphs. Journal of Combinatorial Theory, Series A, 199:105776, 2023.

[2] Sela Fried and Toufik Mansour. Further results on random walk labelings. Discrete Applied Mathematics, 353:211–221, 2024.

[3] Sela Fried and Toufik Mansour. Graph labelings obtainable by random walks. The Art of Discrete and Applied Mathematics, 7(1), 2024.

[4] Sela Fried and Toufik Mansour. Random walk labelings of perfect trees and other graphs. Quaestiones Mathematicae, 48(6):879–901, 2025.

[5] Joseph A Gallian. A dynamic survey of graph labeling. Electronic Journal of Combinatorics, 25:DS6, 2022.

JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING