Cover Times of Random Walks on Finite Graphs

Michael H. Duyzend   Rebecca L. Ferrell
Miranda J. Fix
Department of Mathematics, Carleton College

June 10, 2008

Abstract

The cover time of a random walk on a finite graph is defined to be the number of steps it takes to hit all the vertices of the graph. For our senior integrative exercise in the Department of Mathematics at Carleton College, we investigated the problem of finding whatever information we could (expectation, variance, or exact distribution) about the cover times for random walks on certain types of graphs, in particular, the n-cycle, the star, the "sparkler", and the Petersen graph, deriving new results for the last three graphs. We utilized a variety of techniques to study the cover time, including a general method of exhaustion, gambler's ruin absorption times, recurrence relations, and simulation.

Contents

1 Introduction to the cover time problem **3**

2 The simulator **4**

3 The general method **6**

3.0.1 The extended general method 6

3.0.2 Determining the expectation of the cover time 7

3.0.3 Determining higher moments of the cover time 10

3.0.4 Limitations, restrictions, and words of caution 11

4 The n-cycle **11**

4.1 Reducing the analysis of the n-cycle to the gambler's ruin problem 11

4.2 Expectation and variance of the n-cycle 13

4.3 Exact distribution of the T_k times 15

4.3.1 Approaching gambler's ruin with lattice paths 15

4.3.2 Approaching gambler's ruin with generating functions 21

4.3.3 Approaching gambler's ruin with recurrence relations 22

4.4 Some results for the distribution of C_n 26

5 The star graph **30**

5.1 Expectation of the star 30

5.2 Variance of the star 33

5.3 Generalizing the star to the sparkler 34

6 The Petersen graph **39**

6.1 Applying the general method to the Petersen graph 39

6.1.1 Variance and the general method 43

7 Other possible areas of exploration **44**

8 Acknowledgements **47**

9 Appendix **49**

9.1 Petersen graph expectation equations 49

1 Introduction to the cover time problem

Let G be a finite connected graph. Given some vertex u in G, the degree $\deg(u)$ of u is the number of edges incident to u. The following is a simple random walk on G. At discrete units of time, a particle moves from vertex v to an adjacent vertex with probability $\frac{1}{\deg(v)}$. The *cover time* C is the number of steps required to hit all vertices of the graph. Since C is a random variable, we can study its expectation, variance and even exact distribution for different types of graphs. Exact results for the expected cover time are known for several graphs, including the n-path, the complete graph K_n, the n-cycle, and the star. In general, little is known about the variance or cover time for the exact distribution.

Some useful notions:

- For discrete random variables X and Y we say that X and Y have the **same distribution** and write $X \overset{d}{=} Y$ if $\mathbb{P}(X = k) = \mathbb{P}(Y = k)$ for all k.

- The **expectation** of a random variable X, written $E[X]$, is the sum of the probability of each possible outcome multiplied by the outcome value. The variance $\text{Var}[X]$ is the square of the standard deviation and the fact that $\text{Var}[X] = E[X^2] - E[X]^2$ also holds. A useful property of the variance is that the variance of the sum of independent random variables is the sum of their variances.

- For a graph on n vertices, the **adjacency matrix** is the $n \times n$ matrix where the entry $a_{i,j}$ is equal to 1, if vertex i is adjacent to vertex j, and 0, otherwise.

Some previous results:

- An n-path is defined here as the path consisting of vertices $0, 1, \ldots, n$, where vertex i is adjacent to vertex $i + 1$ for $0 \leq i < n$. The expected cover time for the n-path starting at one of the end vertices is $E[C] = n^2$. (See [5] for a proof using recurrence relations.)

- The complete graph on n vertices, denoted K_n, is the graph in which each of the $\binom{n}{2}$ unordered pairs of distinct vertices is connected by an edge. The expected cover time for K_n is $E[C] = (n-1)\left(1 + \frac{1}{2} + \ldots + \frac{1}{n-1}\right)$. (See [1] for proof using the coupon collector's problem.)

In the following sections we present several methods for investigating the cover time of the n-cycle, star and "sparkler" graphs, and the Petersen graph.

2 The simulator

In order to study the distributions of cover times of random walks, we needed empirical tools to verify our theoretical work. Unable to find an existing program with the functionality we desired, we wrote Java code to simulate many random walks on a given graph and provide us with data. The program (RandomWalk) requires the user to input an adjacency matrix for the graph in question, a starting vertex, and the number of trials desired. RandomWalk outputs the cover time for each random walk in a text file, allowing for easy analysis in a statistical package such as R.

Some details about how RandomWalk works:

- The user must give RandomWalk a text input file with the following information separated by spaces: n (the number of vertices in the graph), the number of trials, and each row of the adjacency matrix for the graph. RandomWalk does *not* presently incorporate any initial verification for the validity of the input text.

- The user must also enter in an appropriate starting vertex (from 0 to $n-1$) when prompted.

- RandomWalk creates a 2-dimensional $n \times n$ integer array and fills it in with the given adjacency matrix entries. It then creates an array of size n of customized linked list objects, each object corresponding to a vertex of the graph. For each linked list object, RandomWalk uses the $n \times n$ integer array to find the degree of each vertex and append nodes to the list containing the numbers of the vertices connected to the given vertex.

- Additionally, RandomWalk stores cutoffs for each node in the linked list in the following manner: if vertex i has degree $\deg(i)$, then the lowest numbered vertex adjacent to vertex i will have cutoff $\frac{1}{\deg(i)}$, the next lowest vertex cutoff $\frac{2}{\deg(i)}$, and so forth, with the last vertex having cutoff $\frac{\deg(i)}{\deg(i)} = 1$.

- For each trial, RandomWalk initializes a boolean array of size n to store whether a vertex has been visited or not. It then starts off at the given initial vertex by generating a random number between 0 and 1 and marking the starting vertex as having been visited in the boolean array. To determine which vertex to visit, RandomWalk compares this number to the cutoff numbers for the vertices adjacent to the starting vertex. If the random number is larger than the cutoff for some vertex, it looks at the next vertex in numerical order, stopping when the random number is smaller than the cutoff. This is the vertex to which the random walk will visit. RandomWalk sets the visited status in the boolean array to 'true' for this new vertex, increments a step counter by one, and checks to see if all entries in the boolean visited array are now 'true'. If so, it ends the walk and records the step count. If not, the walk continues, with a new random number being generated.

- In addition to outputting all the cover times in a text file, RandomWalk displays the mean and variance of cover times in the sample.

RandomWalk is able to easily produce simulations on the order of 10^6 to 10^9 iterations for most of the graphs that we study. This tool proved extremely helpful in the course of our research, allowing us to generate hypotheses about expectation and variance in cover times (the n-cycle, in particular) while confirming theoretical results prone to computational errors (the Petersen graph and "sparkler", in particular).

For the sparkler graph, to test our hypothesis about the relative probabilities with which different length ends of rays are reached, we altered RandomWalk to track this information. Specifically, we changed RandomWalk to require as additional input the number of "special vertices" to monitor and the labels of these vertices. During the course of a walk, rather than stop when the graph is covered, this modified version of RandomWalk stops when one of these "special" vertices is reached and records the number of this vertex. This allowed us to use statistical software to compare the frequencies with which different ray ends of the sparkler were reached.

3 The general method

The General Method described by Blom and Sandell [1] is a method of exhaustion (really a primitive algorithm) to determine the expected value of the cover time on any finite graph. This method can be extended to determine all of the moments of the cover time. Note: The *k-th moment* of a random variable X is defined as $E[X^k]$.

3.0.1 The extended general method

1. Begin with a finite graph.

2. Define new random variables, each the cover time of the graph from a particular state; that is, from a specific configuration of vertices that have already been visited.

3. Write the random variables in terms of each other.
   Ex. $C \stackrel{d}{=} \frac{1}{3}(D+1) + \frac{1}{3}(E+1) + \frac{1}{3}(E'+1)$.

4. Take the random variables to the k^{th} power, $1 \leq k$, where k is the moment of the cover time desired.

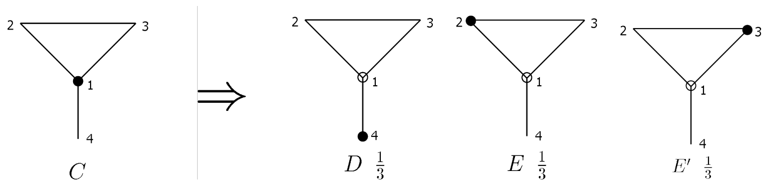
5. Take the expected value of the random variables to the k^{th} power.

6. Solve the generated system of equations for the k^{th} moment.

Note: In order to find the k^{th} moment using this method, it becomes evident that knowledge of the $1^{st}, 2^{nd}, \ldots, k-1^{st}$ moments are necessary.

3.0.2 Determining the expectation of the cover time

The method is best illustrated by a visual example. Consider the graph on four vertices shown below. A state is a particular configuration of vertices that have been visited and is denoted by a bold letter. The cover time from that particular state (a random variable) is denoted by an italicized letter. We start the random walk from the center vertex; call this state **C** and the cover time from this initial state C. We go to state **D** with probability $\frac{1}{3}$, to state **E** with probability $\frac{1}{3}$, and to state **E**$'$ with probability $\frac{1}{3}$. Note that **E** and **E**$'$ are isomorphic states, so E and E' are equal in distribution.



This allows us to write C in terms of the cover times from states **D**, **E**, and **E**$'$.

$$C \stackrel{d}{=} \frac{1}{3}(D+1) + \frac{1}{3}(E+1) + \frac{1}{3}(E'+1).$$

The number 1 is added to the random variables D, E, and E' because, from state **C**, it takes one step to get to state **D**, **E**, or **E**$'$.

We can continue with this strategy by writing the cover times from the states **D**, **E**, and **E**$'$ in terms of the cover times from other states.

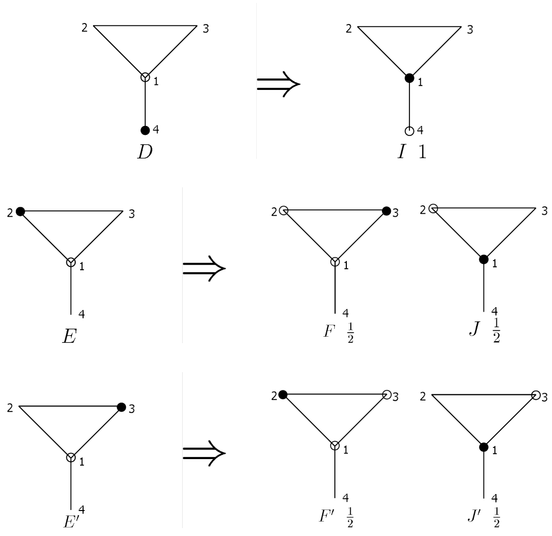


Figure 1: States reachable from $\mathbf{D}$, $\mathbf{E}$, and $\mathbf{E}'$ in graphical form with the cover time from and probability of reaching those states listed below the graphics.

As shown above, from state $\mathbf{D}$, the random walk returns to the center, or state $\mathbf{I}$ with probability 1. So we have that:

$$D \stackrel{d}{=} I + 1$$

From state $\mathbf{E}$ the random walk moves to vertex 3 or state $\mathbf{F}$ with probability $\frac{1}{2}$ and moves back to the center or state $\mathbf{J}$ with probability $\frac{1}{2}$. So we have that:

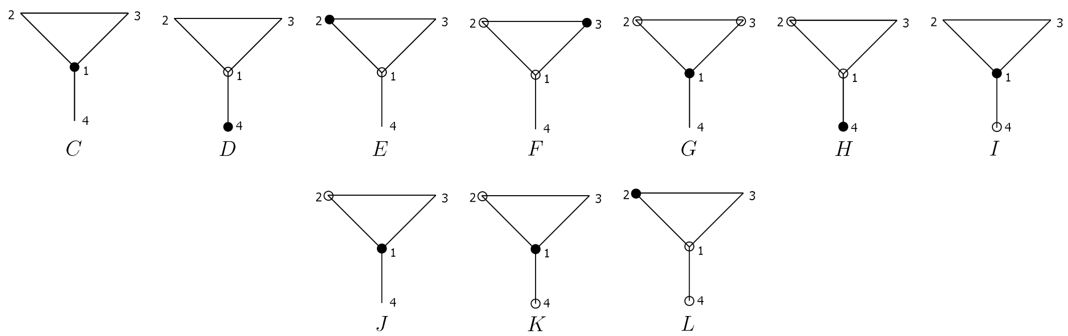
$$E \stackrel{d}{=} \frac{1}{2}(F + 1) + \frac{1}{2}(J + 1)$$

Finally, from state $\mathbf{E}'$, the random walk moves to vertex 2 or state $\mathbf{F}'$ with probability $\frac{1}{2}$ and moves to vertex 1 or state $\mathbf{J}'$ with probability $\frac{1}{2}$. So we have that:

$$E' \stackrel{d}{=} \frac{1}{2}(F' + 1) + \frac{1}{2}(J' + 1)$$

The reason for defining all possible states, even those that are isomorphic to other states, is that the cover times from isomorphic states are only equal in distribution. Since the expected values of random variables with the same distributions are the same, the expected values are equal, for example, $E[F]{=}E[F']$. In this light, we now worry only about non-isomorphic states.

In all there are ten possible non-isomorphic states (shown below) that the random walk can reach.



And so, dealing with expectations, we can generate the system of ten equations shown below:

$$
\begin{aligned}
E[C] &= \frac{1}{3}(E[D]+1)+\frac{2}{3}(E[E]+1)\\
E[D] &= (E[I]+1)\\
E[E] &= \frac{1}{2}(E[J]+1)+\frac{1}{2}(E[F]+1)\\
E[F] &= \frac{1}{2}(E[G]+1)+\frac{1}{2}(E[F]+1)\\
E[G] &= \frac{2}{3}(E[F]+1)+\frac{1}{3}\\
E[F] &= E[K]+1\\
E[I] &= \frac{1}{3}(E[D]+1)+\frac{2}{3}(E[L]+1)\\
E[J] &= \frac{1}{3}(E[F]+1)+\frac{1}{3}(E[E]+1)+\frac{1}{3}(E[H]+1)\\
E[K] &= \frac{1}{3}(E[L]+1)+\frac{1}{3}(E[H]+1)+\frac{1}{3}\\
E[L] &= \frac{1}{2}(E[K]+1)+\frac{1}{2}
\end{aligned}
$$

Solving this system of equations yields the expected value of the cover time from each of the states. We are most interested in the expected cover time from the initial state **C** which is $E[C] = \frac{142}{15} \approx 9.467$.

3.0.3 Determining higher moments of the cover time

In order to find the k^{th} moment of the cover time, it is necessary to take each random variable to the k^{th} power, then take expectations. In most cases, we are particularly interested in finding the variance of the cover time, which requires knowledge of the **second moment** $E[C^2]$. The first step is to square the random variables defined in terms of each other. For example, upon squaring C, we get:

$$\begin{aligned} C^2 &\stackrel{d}{=} \frac{1}{3}(D+1)^2 + \frac{1}{3}(E+1)^2 + \frac{1}{3}(E'+1)^2 \\ &\stackrel{d}{=} \frac{1}{3}(D^2+2D+1) + \frac{1}{3}(E^2+2E+1) + \frac{1}{3}(E'^2+2E'+1) \end{aligned}$$

and upon taking expected values, we get

$$E[C^2] = \frac{1}{3}(E[D^2]+2E[D]+1) + \frac{2}{3}(E[E^2]+2E[E]+1).$$

As is evident here, the first moments (expectations) of the cover times from particular states are needed in order to determine the second moments. In general, to determine the k^{th} moment of the cover times, it is necessary to know the $k-1^{st}$ moment (which requires knowledge of the $k-2^{nd}$ moment and so on) of the cover times. The method for determining higher moments, then, is iterative.

If we solve the resulting system of equations for the second moments of the cover times started above, we discover that the second moment of the cover time from the initial state **C**, is $E[C^2] = \frac{9923}{75} \approx 132.307$. This in turn allows us to find the variance of C $(E[C^2] - E[C]^2)$, so $\text{Var}[C] = \frac{1921}{45} \approx 42.689$ and so the standard deviation of the cover time C $(\sqrt{\text{Var}[C]})$ is about 6.534.

3.0.4 Limitations, restrictions, and words of caution

The general method seems like a miracle; we now have a method to determine all moments of the cover times. Why do we spend time using other methods to determine the expectation and variance of graphs? There are several reasons outlined below:

- *The general method is a brute force method.* It requires the enumeration of all possible non-isomorphic states a random walk can reach. The number of states is at least as large as the number of non-isomorphic subgraphs of a particular finite graph. This means that for a graph on a small number of vertices, say 10, there can be hundreds of non-isomorphic states.

- *The general method is only practical if significant symmetries or special properties of the graph of study exist.* If a graph has a high degree of symmetry or other special properties, it is possible that a relatively small number of non-isomorphic subgraphs exist, which would render the general method usable.

- *It is not known if the moments of the cover time can be computed in polynomial time.* Computationally this method is not practical for even moderately sized graphs.

The general method is a good tool to have in our toolbox, but we must use it judiciously and cleverly.

4 The n-cycle

In the language of graph theory, an n-cycle is a closed walk on n vertices.

4.1 Reducing the analysis of the n-cycle to the gambler's ruin problem

Let C_n denote the cover time for an n-cycle.

Let T_k denote the time to hit the k^{th} new vertex in this random walk given that we have just hit the $k-1^{st}$ new vertex, $1 \leq k \leq n$. (Necessarily, $T_1 = 0$ and $T_2 = 1$.) T_k is equivalent to the time, in gambler's ruin, for absorption in the ruin ($0) or win ($k) states, given a starting amount of $1 and winning or losing $1 with probability $\frac{1}{2}$ at each trial. More explicitly, this is because once we have just hit the $k-1^{st}$ new vertex, then several conditions must hold:

- The set of vertices that have been hit already must be connected in the cycle (due to the random walk process) and form a path with $k-1$ vertices.
- We must be at the end of this path of $k-1$ vertices. If this weren't the case, then the $k-1^{st}$ vertex encountered was in the middle of the path, so the subgraph spanned by the already visited vertices formed two disconnected components, which contradicts the previous fact.
- Since we are at the end of a path of $k-1$ vertices, the only two vertices that can next be hit are the uncovered vertices at either end of the path. (In the case where $k = n$, then these two vertices are the same vertex, but the ways the walk can proceed to hit the last uncovered vertex are similar to the smaller cases.)
- By labeling our current position to be vertex 1 and the other covered vertices as vertices 2, 3, ..., $k-1$ in order from the current position, then the two uncovered vertices that can be next reached should be labeled with 0 and k.

Therefore T_k is equivalent to the time to hit either vertices 0 or k from vertex 1 in a random walk on a k-path. With this setup, C_n can be thought of the time to hit the first new vertex, plus the time to hit the second new vertex from the first new vertex, plus the time to hit the third new vertex from the second new vertex, and so forth, until we hit the n^{th} new vertex and cover the n–cycle. Thus:

$$C_n = T_1 + T_2 + ... + T_n \tag{1}$$

By this construction, the T_k are independent—the time to reach the k^{th} new vertex from the $k-1^{st}$ new vertex is unaffected by the times to reach any previous vertices. Thus for $n > 3$:

$$\begin{aligned}
\mathbb{P}(C_n = j) &= \mathbb{P}\left(\sum_{i=1}^{n} T_i = j\right) \\
&= \mathbb{P}\left(\sum_{i=1}^{n-1} T_i = m, T_n = j - m \text{ for some } m\right) \\
&= \mathbb{P}\left(C_{n-1} = m, T_n = j - m \text{ for some } m\right) \\
&= \sum_{m=0}^{j} \mathbb{P}\left(C_{n-1} = m\right) \mathbb{P}\left(T_n = j - m\right).
\end{aligned}$$

The last expression is the *convolution* of the cover time distribution for the $(n-1)$-cycle and the gambler's ruin absorption time T_n. Evidently, the distribution of the cover times on the n-cycle is equal to the convolution of the distributions of the cover times of the $(n-1)$-cycle and the gambler's ruin absorption time T_n, at least when $n > 3$. With this framework, the distribution of cover times of an n-cycle is reduced to finding the distribution of the gambler's ruin absorption times and convolving the resulting sequences.

4.2 Expectation and variance of the n-cycle

From (1) we see that $E[C_n] = E[T_1] + \cdots + E[T_n]$. Additionally, since the T_k's are independent, $\mathrm{Var}[C_n] = \mathrm{Var}[T_1] + \cdots + \mathrm{Var}[T_n]$. To analyze these summands, we first restrict to looking at the k-path. For a given k-path, let X_i denote, on a k-path, the time to reach either vertices 0 or k from vertex i. Clearly $T_k = X_1$, so $E[T_k] = E[X_1]$ and $\mathrm{Var}[T_k] = \mathrm{Var}[X_1] = E[X_1^2] - E[X_1]^2$. Thus, we need to find $E[X_1]$ and $E[X_1^2]$. We will use recurrence relations to find these values.

First, we have that $X_0 = X_k = 0$ and for $1 \leq i \leq k-1$, $X_i \stackrel{d}{=} \frac{1}{2}(X_{i-1}+1)+\frac{1}{2}(X_{i+1}+1)$. Taking the expectations of X_i, then, we obtain:

- $E[X_0] = E[X_k] = 0$
- $E[X_i] = \frac{1}{2}(E[X_{i-1}] + 1) + \frac{1}{2}(E[X_{i+1}] + 1)$.

The solution to the recurrence defined by these two equations is $E[X_i] = ki - i^2$. Setting $i = 1$, then, we obtain that $E[T_k] = k - 1$.

We thus have the expected cover time of the cycle:

$$E[C_n] = \sum_{k=1}^{n} E[T_k] = \sum_{k=1}^{n} (k-1) = \binom{n}{2}.$$

Now that we have found the first moment of T_k, we can examine the second moment. Squaring the distributional relationships we had set up for the X_i above we find that $X_0^2 = X_k^2 = 0$ and for $1 \leq i \leq k-1$, $X_i^2 \stackrel{d}{=} \frac{1}{2}(X_{i-1}+1)^2 + \frac{1}{2}(X_{i+1}+1)^2$. Thus:

- $E[X_0^2] = E[X_k^2] = 0$
- $E[X_i^2] = \frac{1}{2}(E[X_{i-1}^2] + 2E[X_{i-1}] + 1) + \frac{1}{2}(E[X_{i+1}^2] + 2E[X_{i+1}] + 1)$.

We can use our solution for $E[X_i]$ to solve the recurrence relation defined by the last two equations. The solution to this second recurrence is thus (with the help of *Mathematica*) $E[X_i^2] = \frac{1}{3} i \left(i^3 - 2i^2 k + ik^2 + 2i - 2k\right)$, so $E[T_k^2] = \frac{1}{3}\left(k^3 - 4k + 3\right)$ and $\mathrm{Var}[T_k] = E[T_k^2] - E[T_k]^2 = \frac{1}{3}\left(k^3 - 4k + 3\right) - (k-1)^2$, so:

$$\begin{aligned}\mathrm{Var}[C_n] &= \sum_{k=1}^{n} \mathrm{Var}[T_k] \\ &= \sum_{k=1}^{n} \left(\frac{1}{3}\left(k^3 - 4k + 3\right) - (k-1)^2 \right) \\ &= 2\binom{n+1}{4}.\end{aligned}$$

4.3 Exact distribution of the T_k times

4.3.1 Approaching gambler's ruin with lattice paths

One way to approach the exact distribution of the T_k's is using lattice paths. Recall that T_k is the time it takes to hit the k^{th} new vertex on a random walk on the n-cycle given that we've just hit (and are still currently on) the $k-1^{st}$ new vertex. This is equivalent to the time it takes to hit one of the ends of a path of length k, starting at vertex 1. To facilitate our lattice path arguments, we will relabel the k-path so that the vertices are numbered from -1 to $k-1$ so absorption occurs when we hit vertex -1 or $k-1$. We will refer to the old labeling as the *standard* labeling and the new labeling as the *new* labeling.

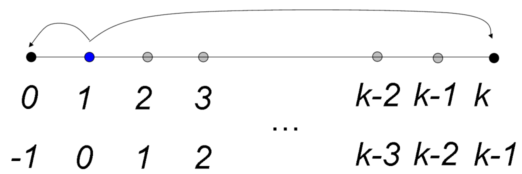


Figure 2: The k-path with standard labels (above) and new labels (below).

What we want is the probability distribution for T_k; that is, the probability of absorption at either end of the path given the walk takes exactly j steps. In order to find this for a given T_k, we need the total number of ways to be absorbed in exactly j steps. Since each step is taken with probability $\frac{1}{2}$, we multiply the total number of ways to be absorbed in exactly j steps by $\left(\frac{1}{2}\right)^j$ to generate the probability of absorption in exactly j steps for a given T_k.

To find the total number of ways to be absorbed, consider the Cartesian coordinate system where the the x-coordinate shows the number of steps the walk has taken and the y-coordinate shows the vertex that random walk is on. The walk can move diagonally from (x, y) to either the $(x+1, y+1)$ or $(x+1, y-1)$. The number of ways to be absorbed by the $k-1^{st}$ vertex (using the new system) is equal to the number of lattice paths from the origin to $(j, k-1)$ that don't *cross* the lines $y = k-2$ or $y = 0$ except on the j^{th} (last) move. (Otherwise absorption would occur before the j^{th} move.) Similarly, the number of ways to be absorbed by the -1^{st} vertex is equal to the number of lattice paths from the origin to $(j, -1)$, with the same restrictions. Since the j^{th} (last) move is pre-determined, these quantities are equal to the number of lattice paths from the origin to $(j-1, k-2)$ and $(j-1, 0)$, respectively, with the same restrictions.

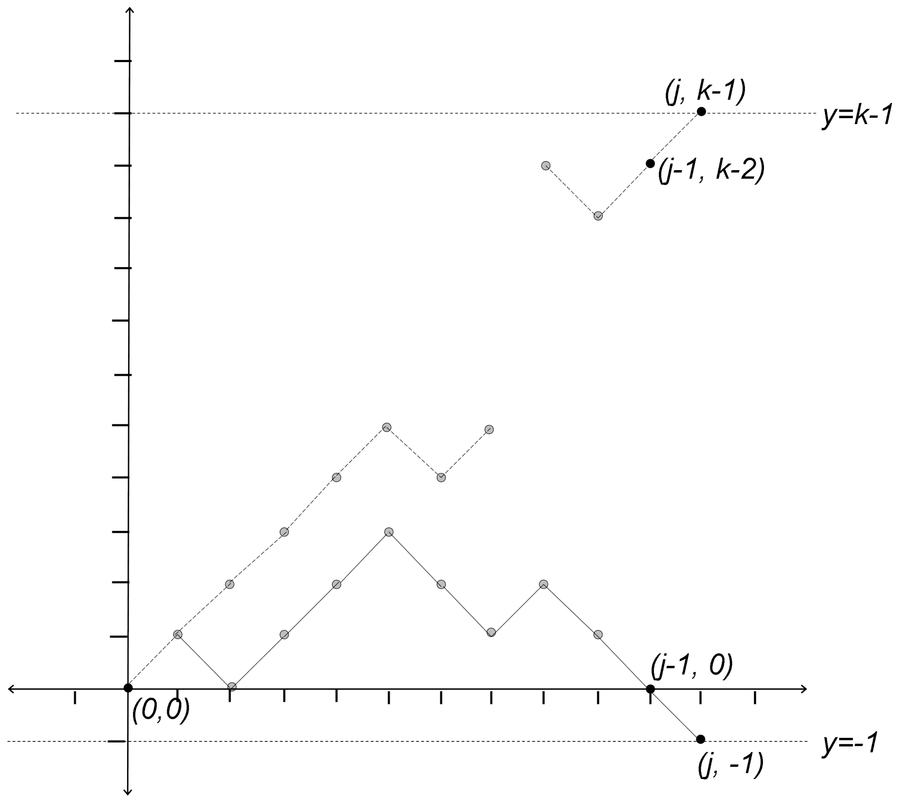


Figure 3: Two possible lattice paths representing absorption in exactly j steps at $k-1$ and -1. Note that the last moves are predetermined.

Many results from combinatorics apply directly to lattice paths that begin at the origin and move only in the positive x and positive y directions. We can make the lattice paths of the type shown in Figure 3 in the common form using the transformation $(x, y) \to \left(\frac{x-y}{2}, y + \frac{x-y}{2}\right)$.

In the new coordinate system,

$$\begin{aligned}
y = -1 &\quad\longrightarrow\quad y = x - 1 \\
y = k - 1 &\quad\longrightarrow\quad y = x + k - 1 \\
(0,0) &\quad\longrightarrow\quad (0,0) \\
(j-1, 0) &\quad\longrightarrow\quad \left(\frac{s-1}{2}, \frac{s-1}{2}\right) \\
(j-1, k-2) &\quad\longrightarrow\quad \left(\frac{s-k+1}{2}, k - 2 + \frac{s-k+1}{2}\right)
\end{aligned}$$

Also note that in this coordinate system, the length of one unit in the previous coordinate system is $\frac{\sqrt{2}}{2}$.

The number of paths from $(0,0)$ to $p = (p_1, p_2)$ where $p \in \mathbb{Z}^2$ that do not *cross* the lines $y = x + d$ and $y = x + c$ where $c, d \in \mathbb{Z}$ and $d > c$ is:

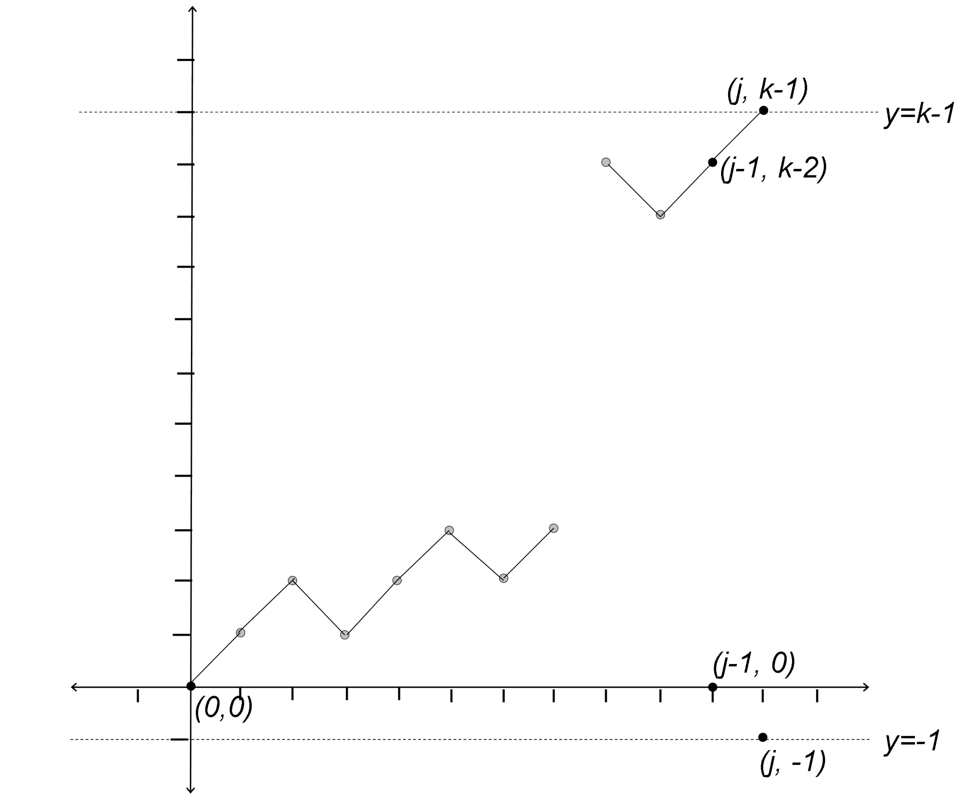
$$\sum_{s \in \mathbb{Z}} \left\{ \binom{p_1 + p_2}{p_1 - s(d - c + 2)} - \binom{p_1 + p_2}{p_1 + s(d - c + 2) + c - 1} \right\} \tag{2}$$

provided that $p_1 + c \leq p_2 \leq p_1 + d$ and $c \leq 0$. A generalized form of this formula can be found in [3], the proof of which can be found in [4] p. 9.

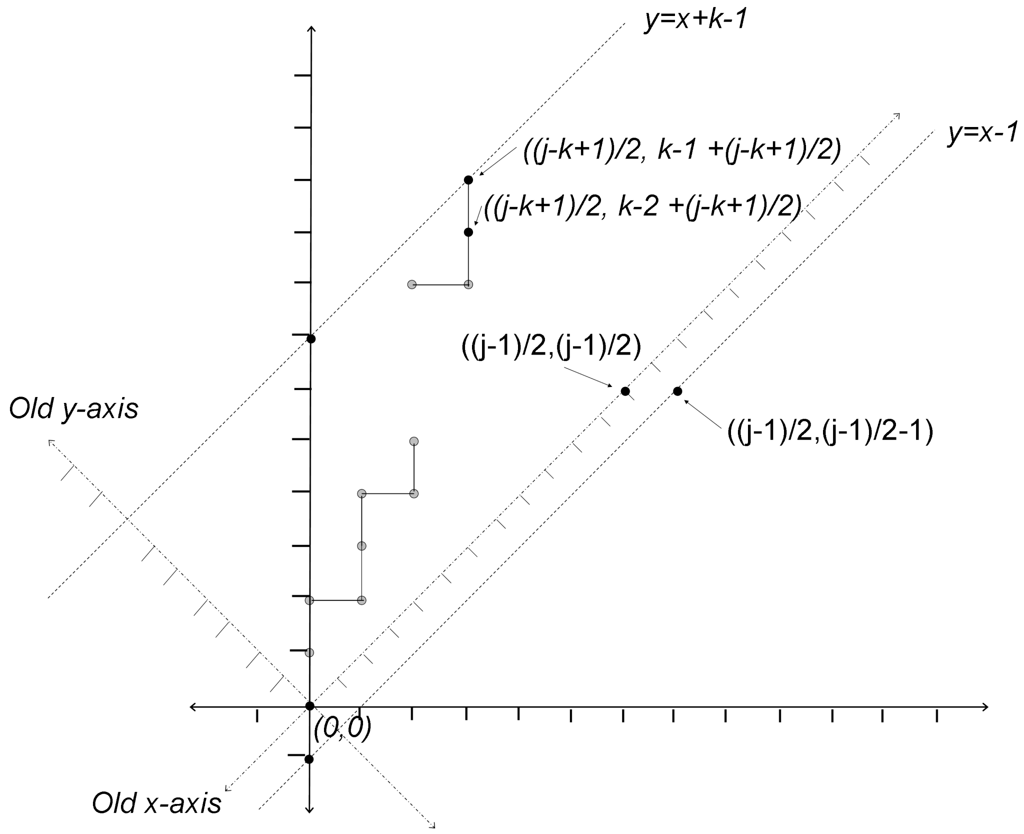
Refering to Figure 4, we see that for a given path of length k, the total number of ways to be absorbed in exactly j steps, denoted $T[k, j]$, is equal to the number of lattice paths (with the only possible moves in the positive x and y directions) from $(0,0)$ to $(\frac{j-1}{2}, \frac{j-1}{2})$ plus the number of lattice paths from $(0,0)$ to $\left(\frac{j-k+1}{2}, k - 1 + \frac{j-k+1}{2}\right)$. Then, using (2), we have

$$T[k,j] = \sum_{s \in \mathbb{Z}} \left\{ \binom{j-1}{\frac{j-1}{2} - sk} - \binom{j-1}{\frac{j-1}{2} + sk - 1} \right\} + \sum_{s \in \mathbb{Z}} \left\{ \binom{j-1}{\frac{j-k+1}{2} - jk} - \binom{j-1}{\frac{j-k+1}{2} + sk - 2} \right\} \tag{3}$$

The left sum in this equation is equal to the number of ways to be absorbed by the



(a) Old Coordinate System



(b) New Coordinate System

Figure 4: Transformation from Old (a) to New (b) coordinate systems.

left end of the path ($p_1 = \frac{j-1}{2}$ and $p_2 = \frac{j-1}{2}$) and the right sum the number of ways to be absorbed by the right end of the path ($p_1 = \frac{j-k+1}{2}$ and $p_2 = k-2+\frac{j-k+1}{2}$). Also in our case, $d = k-2$ and $c = 0$.

In order to actually compute these values using a program like *Mathematica*, we need restrictions and bounds on the values that k and j can take. First, we'll consider parity. Referring to Figure 3, we see that a lattice path can hit $y = k-1$ on the j^{th} move if and only if j and $k-1$ have different parities.

Note that if we only make upward steps, it will take $k-1$ steps to reach $y = k-1$. So the minimum number of steps to reach $y = k-1$ is $k-1$. Every time the walk steps downward an upward step must cancel it out. Hence, the total number of steps must be $k-1$ plus some even number of steps, so the parity of j must be the same as the parity of $k-1$. This implies that the parities of k and j must be different. Similarly, paths can hit $y = -1$ if and only if j is odd. The same number of upward steps as downward steps is necessary to return to line $y = 0$, then the path must always take an additional step to hit $y = -1$.

To express this, we let:

$$T[k,j] = T[k,j]_{odd} + T[k,j]_{diff}$$

where $T[k,j]_{odd}$ is the left summation and $T[k,j]_{diff}$ is the right summation in equation (3).

To compute $T[k,j]$ we also need bounds on each of the sums. To get these bounds, we use the fact that for $\binom{x}{y}$, $x \geq y$, and $y \geq 0$. Distributing out the summations and applying these inequalities, we get:

$$
\begin{aligned}
T[k,j]_{odd} &= \begin{cases} \sum_{s=\lceil \frac{-(j-1)}{2k} \rceil}^{\lfloor \frac{j-1}{2k} \rfloor} \binom{j-1}{\frac{j-1}{2}-sk} - \sum_{s=\lceil \frac{(3-j)}{2k} \rceil}^{\lfloor \frac{j+1}{2k} \rfloor} \binom{j-1}{\frac{j-1}{2}+sk-1} & \text{if } j \text{ odd} \\ 0 & \text{otherwise} \end{cases} \\
T[k,j]_{diff} &= \begin{cases} \sum_{s=\lceil \frac{3-j-k}{2k} \rceil}^{\lfloor \frac{j-k+1}{2k} \rfloor} \binom{j-1}{\frac{j-k+1}{2}-sk} - \sum_{s=\lceil \frac{1-j+k}{2k} \rceil}^{\lfloor \frac{j+k-1}{2k} \rfloor} \binom{j-1}{\frac{j-k+1}{2}+sk-1} & \text{if parity } j,\ k \text{ different} \\ 0 & \text{otherwise} \end{cases} \\
T[k,j] &= T[k,j]_{odd} + T[k,j]_{diff}
\end{aligned}
$$

Since each step occurs with probability $\frac{1}{2}$, and the walk makes j steps, the probability distribution function for a particular T_k, where we want to know the probability of absorption in *at most* γ steps is:

$$\mathbb{P}(T_k \leq \gamma) = \sum_{j=1}^{\gamma} \left(T[k,j] * \left(\frac{1}{2}\right)^j \right)$$

As mentioned, since the T_k are independent and since $C_n = T_1 + T_2 + \ldots + T_n$ for $n > 3$, we can convolve the distributions of the T_k to get the exact distribution of the C_n.

Finally, if k is large, the values for the sequence $T[k,1], T[k,2], \ldots, T[k,j]$ approach the sequence of Catalan numbers, with zeros interspersed between each. As k gets large, the upper boundary on our lattice paths (in the new coordinate system, Figure $(4b)$), $y = x + k - 1$, essentially dissapears, since it is impossible to cross it except with a very large number of steps. The boundary that still holds is that we cannot cross the line $y = x$. The number of lattice paths from the origin to the point (n, n) which do not cross the line $y = x$ is known to be $\frac{1}{n+1}\binom{2n}{n}$, the n^{th} Catalan number. Using our parity argument, we can hit this line (be absorbed at the left end of the k-path) only if j is odd which is the reason for the interspersed zeros.

4.3.2 Approaching gambler's ruin with generating functions

William Feller [2] supplies an argument for developing the generating function for gambler's ruin times in the third edition of his classic *An Introduction to Probability Theory and Its Applications*, Chapter 14, Sections 4 and 5. The sketch of his argument is as follows, with the notation modified for our context in examining the exact distribution of the time T_k:

For a k-path, let $u_{i,j+1}$ be the probability that, from vertex i, we hit vertex 0 in exactly $j+1$ steps. Then if $2 \le i \le k-2$ and we take 1 step (left or right, each with probability $\frac{1}{2}$),

$$u_{i,j} = \frac{1}{2}u_{i-1,j} + \frac{1}{2}u_{i+1,j}. \tag{4}$$

We set boundary conditions as follows:

- $u_{0,j} = u_{k,j} = 0$ for $j \ge 1$
- $u_{0,0} = 1$
- $u_{i,0} = 0$ for $i \ge 1$

Now, we form the generating function $U_i(x) = \sum_{j=0}^{\infty} u_{i,j}x^j$. By multiplying each side of (4) by x^{j+1}, and summing over all j, we obtain

$$U_i(x) = \frac{1}{2}xU_{i+1}(x) + \frac{1}{2}xU_{i-1}(x), \tag{5}$$

and similarly for the boundary equations, $U_0(x) = 1$ and $U_k(x) = 0$.

Feller states that we can find solutions to (5) by looking at solutions of the form $U_i(x) = \lambda^i(x)$. Feller finds two such solutions, $\lambda_1^i(x)$ and $\lambda_2^i(x)$, and rewrites $U_i(x)$ in terms of these solutions. He then derives explicit expressions for $U_i(x)$ by exploring the form of $U_i(x)$ and using a method of partial fractions with various complex substitutions.

The end result of these manipulations is that

$$u_{i,j} = \frac{1}{k} \sum_{m=1}^{k-1} \cos^{j-1}\left(\frac{\pi m}{k}\right) \sin\left(\frac{\pi m}{k}\right) \sin\left(\frac{\pi i m}{k}\right).$$

(We omit the details.)

This is an explicit expression for the probability of hitting vertex 0 from vertex i in exactly j steps. Thus, with $i = 1$, $u_{1,j}$ gives us the probability of hitting vertex 0 from vertex 1 in exactly j steps. With $i = k - 1$, $u_{k-1,j}$ is the probability of hitting vertex 0 from vertex $k - 1$ in exactly j steps, which is by symmetry of the random walk equal to the probability of hitting vertex k from vertex 1 in exactly j steps. Thus, the probability of hitting either vertex 0 or vertex k in exactly j steps from vertex 1 is $u_{1,j} + u_{k-1,j}$. Plugging in these values for i and simplifying, we finally obtain:

$$P(T_k = j) = \frac{1}{k} \sum_{m=1}^{k-1} \cos^{j-1}\left(\frac{\pi m}{k}\right) \sin\left(\frac{\pi m}{k}\right) \left(\sin\left(\frac{\pi m}{k}\right) + \sin\left(\frac{\pi (k-1) m}{k}\right)\right).$$

The exact distribution of C_n is obtained by convolving the above sequence of probabilities, as explained earlier.

4.3.3 Approaching gambler's ruin with recurrence relations

We now give a different approach for studying the exact distribution. Again on the k-path, let $p(i, j)$ be the probability, starting at vertex i, of reaching one of the end vertices in exactly j steps. We can similarly define the following recurrence relation:

$$p(i,j) = \frac{1}{2}p(i-1,j-1) + \frac{1}{2}p(i+1,j-1) \text{ for } 1 \le i \le k$$

with boundary conditions:

$$p(0,j) = \begin{cases} 1 & \text{if } j = 0 \\ 0 & \text{else} \end{cases}$$

$$p(k,j) = \begin{cases} 1 & \text{if } j = 0 \\ 0 & \text{else} \end{cases}$$

Recall that in gambler's ruin we start at vertex 1. For a given k-path, let s_j be the the number of ways starting at vertex 1 to reach an end vertex in exactly j steps. Since the probability of a given way to reach an end vertex in j steps is $(\frac{1}{2})^j$.

$$s_j = p(1,j) \cdot 2^j$$

Using *Mathematica* we can generate the sequences (s_j) for a given k, shown below for $1 \le j \le 18$ and $2 \le k \le 10$.

k	**Value of j**																	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	0	2	0	4	0	8	0	16	0	32	0	64	0	128	0	256	0
5	1	0	1	1	2	3	5	8	13	21	34	55	89	144	233	377	610	987
6	1	0	1	0	3	0	9	0	27	0	81	0	243	0	729	0	2187	0
7	1	0	1	0	2	1	5	5	14	19	42	66	131	221	417	728	1341	2380
8	1	0	1	0	2	0	6	0	20	0	68	0	232	0	792	0	2704	0
9	1	0	1	0	2	0	5	1	14	7	42	34	132	143	429	560	1429	2108
10	1	0	1	0	2	0	5	0	15	0	50	0	175	0	625	0	2250	0

Several interesting patterns emerge. We observe:

$$
\begin{aligned}
\text{for } k=4,\ s_j &= 2^{\frac{j-1}{2}} \text{ for } j \text{ odd} \\
\text{for } k=5,\ s_j &= F_{j-3} \text{ for } j \geq 3, \text{ where } F_i \text{ is the } i^{th} \text{ Fibonacci number} \\
\text{for } k=6,\ s_j &= 3^{\frac{j-3}{2}} \text{ for } j \geq 3 \text{ odd} \\
\text{as } k \to \infty,\ s_j &\to \text{ the sequence of Catalan numbers alternating with zeroes}^{1}
\end{aligned}
$$

For small k it is not difficult to justify these relations for s_j. Let R denote a step to the right and L a step to the left. Each path will be denoted by a sequence of L's and R's. It is clear that for $k = 2$, $s_j = 2$ for $j = 1$ and zero everywhere else, since if we start at vertex 1 then we will reach either vertex 0 or 2 in the first step.

In the case $k = 3$, for each path of j L's and R's we obtain a path of $j + 2$ steps by prepending an RL to the given sequence. Similarly, each path of $j + 2$ steps must begin

[1] This pattern was noticed by Eric Egge and Lily Thiboutot, and proved by Jonah Ostroff (Carleton College). We omit the argument for why this is so here.

with an RL, and we get a path of j steps by deleting the RL. This establishes a bijection giving that all the s_j's are the same. Thus $s_j = 1$ for $j \geq 1$.

For $k = 4$ we find the corresponding recurrence relation, $s_{j+2} = 2s_j$ with initial condition $s_1 = 1$ by creating bijections between paths that take j steps and paths that take $j+2$ steps. For each path of j steps we obtain a path of $j+2$ steps by either *(a)* prepending RL or *(b)* inverting the sequence and prepending RR. Note that *(a)* works for any k, since it brings us back to vertex 1 having taken an extra 2 steps. On the other hand, *(b)* works because RR brings us to vertex 3, which is the mirror image of vertex 1, and from there we follow the mirror image of a path that works for vertex 1 in j moves. Since RL and RR are the only ways to start the sequence of moves, and we assume that we found all the possible paths for s_j, it follows that these are the only possible paths for s_{j+2}. For the other direction, each path of $j+2$ steps begins either with RL or RR. In the former case we obtain a path of j steps by deleting the RL. In the latter case we delete the RR and invert the sequence of L's and R's to obtain a path of j steps.

For $k = 5$ we find the recurrence relation, $s_j = s_{j-1} + s_{j-2}$ for j large enough, which is recursively equivalent to the relation, $s_j = s_{j-2} + s_{j-3} + s_{j-2}$. From a sequence of moves in $j-2$ steps, we obtain a sequence of moves in j steps by *(a)* prepending RL and *(b)* removing the first R and prepending RRL; from a sequence of moves in $j-3$ steps, obtain a sequence of moves in j steps by *(c)* inverting the sequence and prepending RRR. This gives us sequences starting with RL, RRL and RRR. The third method works because it takes us in 3 steps to vertex 4, the mirror image of vertex 1 and from there we follow the mirror image of a path that works for vertex 1 in $j-3$ steps.

For $k = 6$ we find the recurrence relation $s_{j+2} = 3s_j$. From a sequence of moves in j steps we obtain a sequence of moves in $j+2$ steps by a prepending RL; *(b)* removing the first R and prepending RRL; and *(c)* inverting the sequence, removing the first L and prepending RRR. This gives us sequences starting with RL, RRL, and RRR. The second method works because it effectively inserts a RL after the first move to the right, which brings us to the same vertex as a single R but with two extra steps. The third method works because it effectively brings us to vertex 5, i.e. the mirror image of vertex 1 and proceeds from there with a total of two extra steps.

Now let c_j denote the number of ways starting at a given vertex to cover the n-cycle in exactly j steps. Recall that for $n > 3$ the distribution of the cover times on the n-cycle is equal to the convolution of the distributions of the cover times of the $(n-1)$-cycle and the gambler's ruin absorption time T_n. So we can convolve the s_j sequences to obtain the c_j sequences, which are given below for $1 \le j \le 17$ and $3 \le n \le 8$.

n	**Value of j**																
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
3	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
4	0	0	2	2	6	6	14	14	30	30	62	62	126	126	254	254	510
5	0	0	0	2	2	8	10	26	36	78	114	224	338	626	964	1718	2682
6	0	0	0	0	2	2	10	12	40	52	146	198	506	704	1696	2400	5554
7	0	0	0	0	0	2	2	12	14	54	70	218	304	832	1222	3068	4680
8	0	0	0	0	0	0	2	2	14	16	70	88	308	414	1270	1790	5036

To find the probabilities of covering the n-cycle in in j steps, divide the $(n, j)^{th}$ entry of the table by 2^j. This gives the exact distribution of the cover time for the n-cycle.

4.4 Some results for the distribution of C_n

To get an idea of what the distributions for C_n actually look like, we ran simulations of random walks on the n-cycle for $4 \le n \le 8$. The results of these simulations are shown in the Figures 5-9 and in Table 1.

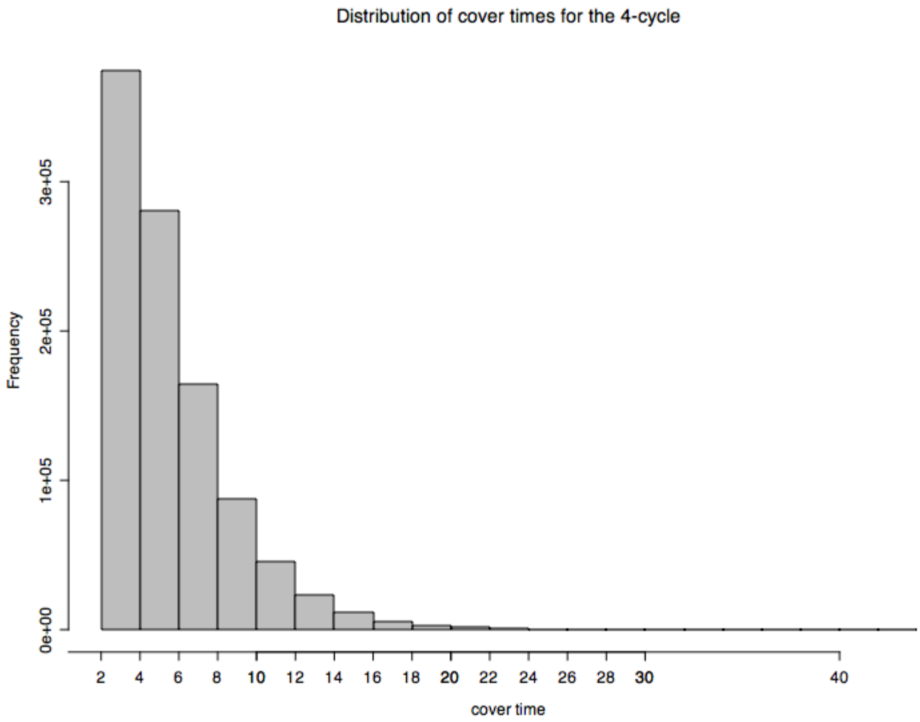


Figure 5: Simulated distribution of the 4-cycle with $1,000,000$ trials.

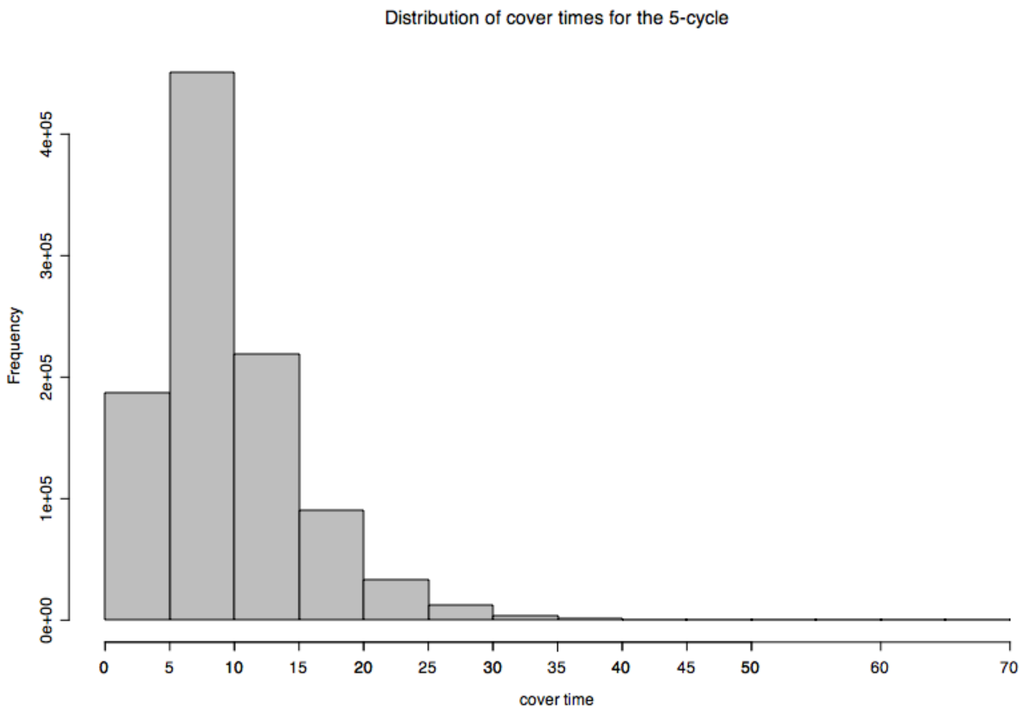


Figure 6: Simulated distribution of the 5-cycle with $1,000,000$ trials.

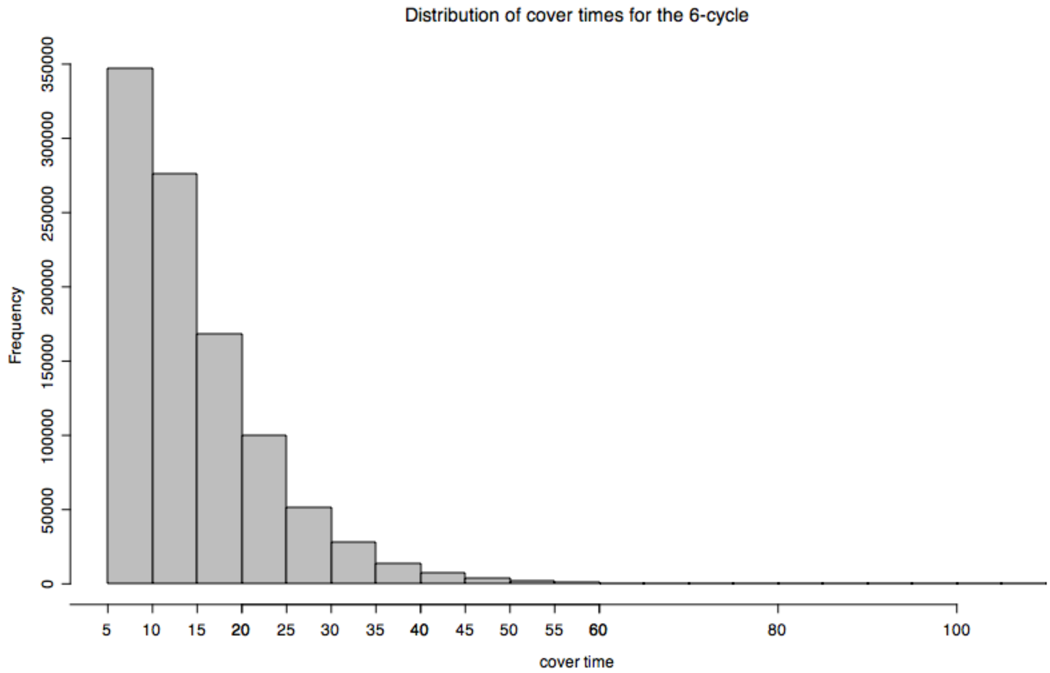


Figure 7: Simulated distribution of the 6-cycle with $1,000,000$ trials.

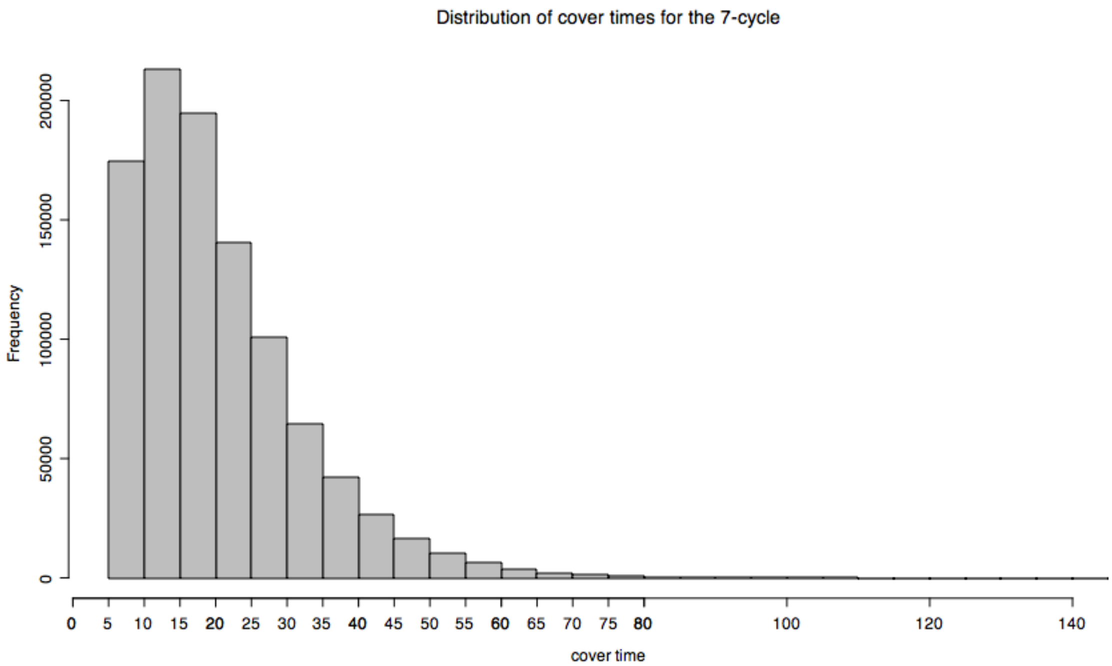


Figure 8: Simulated distribution of the 7-cycle with $1,000,000$ trials.

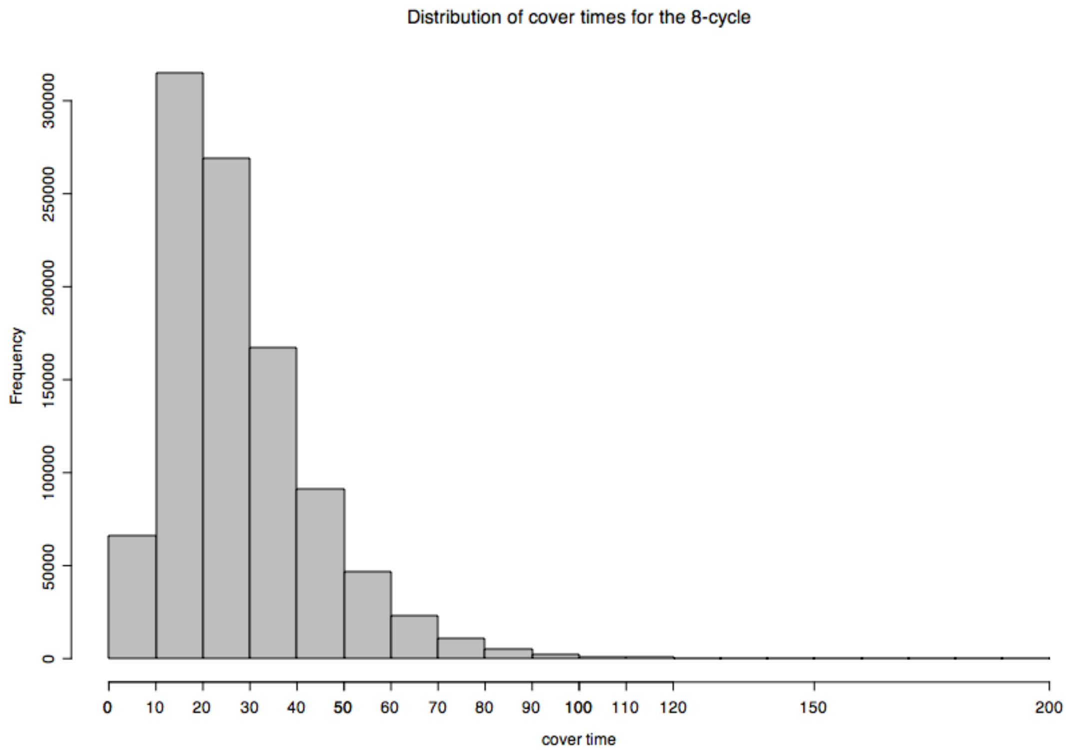


Figure 9: Simulated distribution of the 8-cycle with $1,000,000$ trials.

n	$E[C_n]_{sim}$	$E[C_n]_{calc}$	$\mathrm{StDev}[C_n]_{sim}$	$\mathrm{StDev}[C_n]_{calc}$
4	6.004	6	3.165	3.162
5	10.002	10	5.478	5.477
6	14.995	15	8.353	8.367
7	20.979	21	11.827	11.832
8	28.019	28	15.868	15.875

Table 1: Simulated and calculated values for $E[C_n]$ and $\mathrm{Var}[C_n]$.

5 The star graph

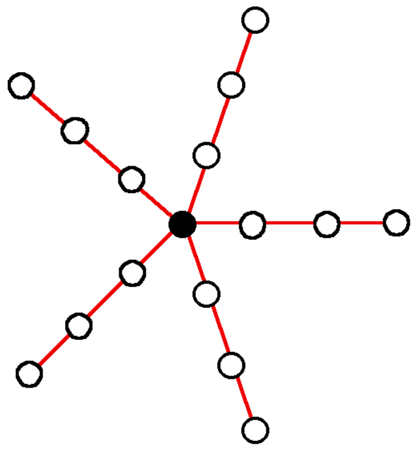


Figure 10: Example of a star graph, $r = 5$, $v = 2$

The star is a special type of tree graph consisting of r rays emanating from a central vertex, each ray of length v. Figure 10 illustrates a star with $r = 5$ and $v = 2$.

With the star, hitting the end of a ray implies that all the vertices on that ray have been covered. Thus, we can simplify the analysis of the cover time of the star by examining only the times between hitting ends of different rays. We use an argument outlined in [1] to find the expectation of the cover time of the star, and extend this structure to find the variance as well.

5.1 Expectation of the star

Let F_i denote the time to reach the i^{th} new ray end of the star (that is, the i^{th} distinct ray end to be covered), starting from the center after having returned from the $i-1^{st}$ new ray end. Let G_i denote the time to reach the center of the star after hitting the i^{th} ray end.

First, observe that the following holds for the cover time C of the star:

$$C \stackrel{d}{=} F_1 + G_1 + F_2 + G_2 + \ldots + F_{r-1} + G_{r-1} + F_r. \tag{6}$$

Additionally, note that all the F_i and G_i random variables are independent, and that the G_i are identically distributed and equal in distribution to the cover time of a random walk on a walk on a v-path, starting at one end.

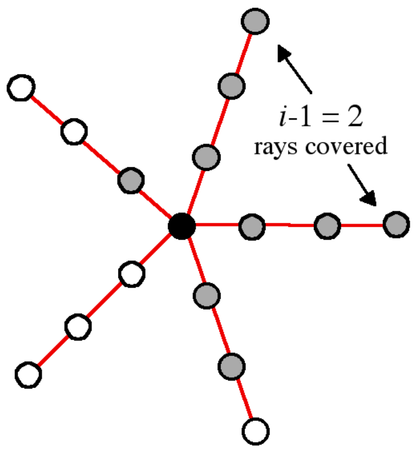


Figure 11: A star with $i-1=2$ ray ends already covered

Suppose we have hit the $i-1^{st}$ new ray end and have returned to the center, as illustrated in Figure 11 (with gray vertices representing visited vertices and the black vertex representing the current location of the walk.)

Then, in finding the distribution of F_i, there are two cases for the next ray end we hit:

- The next ray end we hit is one we have not yet hit.
- The next ray end we hit is one we have already hit.

The latter situation occurs with probability $\frac{i-1}{r}$, since we have already hit $i-1$ out of the r ray ends and all rays are the same length, which means that the former situation occurs with the complementary probability $\frac{r-i+1}{r}$.

In the former situation, we need only to consider the time to get from the center to the ray end (call this H_i), since the next ray end we hit is one we have not yet hit. In the latter situation, we must account for the time to reach the ray end (call this H_i', the time to return to the center from the ray end (call this G_i'), and the time to actually hit the next *new* ray end from the center (call this F_i'). We thus get:

$$F_i \stackrel{d}{=} \frac{r-i+1}{r}(H_i) + \frac{i-1}{r}(H_i' + G_i' + F_i'). \tag{7}$$

Clearly F_i and F_i' are equal in distribution, so $E[F_i] = E[F_i']$. H_i and H_i' are equal in distribution, but they are also equal in distribution to the time it takes to travel from one end of a path of length v to the other end, since walking from the center to a ray end is equivalent to walking down a path—we have r paths to choose from initially, but this does not affect the time to reach the end of any one of them. G_i' is obviously equal in distribution to the time to walk down a v-path, as is G_i. For simplicity, then, when we take the expectations of G_i, G_i', H_i, or H_i', we will say they are equal to $E[G]$, where G represents the time to walk from one end of a v-path to the other.

Taking the expectation of F_i:

$$E[F_i] = \frac{r-i+1}{r}(E[H_i]) + \frac{i-1}{r}(E[H_i'] + E[G_i'] + E[F_i']).$$

We use the previously described substitutions and solve for $E[F_i]$ to obtain:

$$E[F_i] = \frac{r+i-1}{r-i+1}E[G].$$

Since G is the time it takes to walk from one end of a v-path to another, $E[G]$ is just the expectation of that time, which we know from [1] to be v^2.

Thus:

$$E[F_i] = \frac{r+i-1}{r-i+1}v^2.$$

Using (6), then:

$$
\begin{aligned}
E[C] &= E[F_1 + G_1 + F_2 + G_2 + \ldots + F_{r-1} + G_{r-1} + F_r] \\
&= E[F_1] + E[G_1] + E[F_2] + E[G_2] + \ldots + E[F_{r-1}] + E[G_{r-1}] + E[F_r] \\
&= \sum_{i=1}^{r} E[F_i] + (r-1)E[G] \\
&= \sum_{i=1}^{r} \frac{r+i-1}{r-i+1} v^2 + (r-1)v^2 \\
&= v^2 \left(-1 + r + \sum_{i=1}^{r} \frac{r+i-1}{r-i+1} \right) \\
&= v^2 \left(-1 + \sum_{i=1}^{r} \left(1 + \frac{r+i-1}{r-i+1} \right) \right) \\
&= v^2 \left(-1 + \sum_{i=1}^{r} \frac{2r}{r-i+1} \right) \\
&= v^2 \left(-1 + 2r \sum_{i=1}^{r} \frac{1}{r-i+1} \right) \\
&= v^2 \left(-1 + 2r \sum_{i=1}^{r} \frac{1}{i} \right).
\end{aligned}
$$

5.2 Variance of the star

The variance of the star can be obtained using the relationship established in (7). Examining F_i^2, we find that:

$$
F_i^2 \stackrel{d}{=} \frac{r-i+1}{r}(H_i)^2 + \frac{i-1}{r}(H_i' + G_i' + F_i')^2.
$$

Expanding the squared terms, taking the expectation of F_i^2, and using the independence of G_i, G'_i, H_i, and H'_i, we have that:

$$\begin{aligned}
E[F_i^2] &= \frac{r-i+1}{r}(E[H_i^2]) + \frac{i-1}{r}(E[H_i'^2] + E[G_i'^2] + E[F_i'^2] + \\
&\quad 2E[H'_i]E[G'_i] + 2E[H'_i]E[F'_i] + 2E[G'_i]E[F'_i]) \\
E[F_i^2] &= \frac{r-i+1}{r}(E[G^2]) + \frac{i-1}{r}(E[G^2] + E[G^2] + E[F_i^2] + 4E[G]^2 + 2E[G]E[F_i]) \\
\frac{r-i+1}{r}E[F_i^2] &= E[G^2] + \frac{i-1}{r}\left(E[G^2] + 2v^4 + 4\cdot\frac{r+i-1}{r-i+1}v^4\right) \\
E[F_i^2] &= \frac{r+i-1}{r-i+1}E[G^2] + \frac{i-1}{r-i+1}\left(4\cdot\frac{3r-i+1}{r-i+1}v^4\right).
\end{aligned}$$

We need $E[G2]$. This can be found using recurrence relations. If for a v-path, we let Y_i denote the time to hit vertex v from vertex i, then $E[G^2] = E[Y_0^2]$. We have that $Y_0 = Y_1 + 1$, $Y_i \stackrel{d}{=} \frac{1}{2}(Y_{i-1}+1) + \frac{1}{2}(Y_{i+1}+1)$ (for $1 \le i \le v-1$), and $Y_v = 0$, so we can find both the first and second moment of Y_i from these relations. We find that $E[Y_0^2] = E[G^2] = \frac{1}{3}(5v^4 - 2v^2)$.

Now we can determine the variance of the star:

$$\begin{aligned}
\mathrm{Var}[C] &= \mathrm{Var}[F_1] + \mathrm{Var}[G_1] + \ldots + \mathrm{Var}[F_{r-1}]\,\mathrm{Var}[G_{r-1}] + \mathrm{Var}[F_r] \\
&= \sum_{i=1}^{r}\left(E[F_i^2] - E[F_i]^2\right) + \sum_{i=1}^{r-1}\left(E[G_i^2] - E[G_i]^2\right) \\
&= \sum_{i=1}^{r}\left(E[F_i^2] - E[F_i]^2\right) + (r-1)\left(E[G^2] - E[G]^2\right) \\
&= \frac{2}{3}v^2 - \frac{2}{3}v^4 - \frac{4}{3}rv^2(1+2v^2)\sum_{i=1}^{r}\frac{1}{i} + 4r^2v^4\sum_{i=1}^{r}\frac{1}{i^2}.
\end{aligned}$$

5.3 Generalizing the star to the sparkler

The "sparkler" graph is a modified star, with $r-1$ (short) rays of length v and one (long) ray of length $v+c$. Figure 12 illustrates a sparkler with $r = 5$, $v = 2$ and $c = 2$.

The argument for the expected cover time of the sparkler is similar to that for the

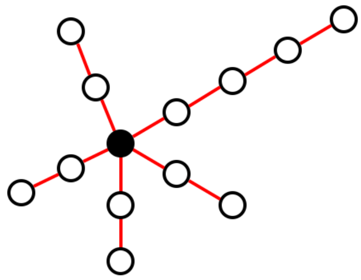


Figure 12: Example of a sparkler graph, $r = 5$, $v = 2$, $c = 2$

star, with an added conditioning step because of the differences in ray length. Suppose we are at the central vertex having hit $i-1$ distinct ray ends. Let $hr[i]$ be the expected time to hit the i^{th} new ray end and return to the center. Also let $h[r]$ be the expected time to hit the last new ray end given that we have already hit $r-1$ ray ends. Note that once we have hit the last ray end we need not return to the center. Thus

$$E[C] = hr[1] + hr[2] + \ldots + hr[r-1] + h[r]$$

To find $hr[i]$ we must condition on whether or not we have been to the long ray end. We define the following conditional probabilities:

$$\begin{aligned} p[i] &= \mathbb{P}(\text{hit } i-1 \text{ short ray ends} \mid \text{hit } i-1 \text{ ray ends}) \\ 1-p[i] &= \mathbb{P}(\text{hit } i-2 \text{ short ray ends and the long ray end} \mid \text{hit } i-1 \text{ ray ends}) \\ s &= \mathbb{P}(\text{hit a particular short ray end} \mid \text{hit a ray end}) \\ g &= \mathbb{P}(\text{hit the long ray end} \mid \text{hit a ray end}) \end{aligned}$$

So: $(r-1)s + g = 1$.

Given that we have hit $i-1$ distinct ray ends, either we have been to $i-1$ short ray ends with probability $p[i]$, or we have been to the long ray end and $i-2$ short ray ends with complementary probability $1-p[i]$. For each of these possibilities, we can either hit a new ray end immediately or hit an old ray end and return to the center to repeat the process. Recall that the expected time to cover a v-path (short ray) starting at one of the ends is v^2, and similarly the expected cover time for a $(v+c)$-path (long ray) starting at one end is $(v+c)^2$.

For $1 < i < r$,

$$hr[i] = p[i] \cdot t1[i] + (1 - p[i]) \cdot t2[i],$$

where $t1[i]$ is the remaining time to hit the i^{th} new ray end given that we have hit $i-1$ short ray ends, and $t2[i]$ is the remaining time to hit the i^{th} new ray end given that we have hit the long end and $i-2$ short ends. $t1[i]$ can be expressed by the recurrence

$$t1[i] = (r-i)s \cdot 2v^2 + g \cdot 2(v+c)^2 + (i-1)s \cdot (2v^2 + t1[i]).$$

This is because if we hit a new ray end on the first attempt, it will be a short ray end with probability $(r-i)s$, thus requiring $2v^2$ steps to reach the end and return to the center, or the long ray end with probability g, thus requiring $2(v+c)^2$ steps to reach the end and return to the center. On the other hand, we will hit a previously visited (short) ray end with probability $(i-1)s$, thus requiring $2v^2$ steps to hit that end and return to the center, plus the remaining time to hit a new ray end, which is again $t1[i]$.

Similarly, $t2[i]$ can be expresssed by the following recurrence relation:

$$t2[i] = (r-i+1)s \cdot 2v^2 + g \cdot (2(v+c)^2 + t2[i]) + (i-2)s \cdot (2v^2 + t2[i]),$$

since in this case the long ray end has already been visited.

The case $i = r$ must be treated separately. Recall that $h[r]$ is the time to hit the r^{th} new ray end, thus covering the sparkler. So

$$h[r] = p[r] \cdot t1[r] + (1 - p[r]) \cdot t2[r],$$

where $t1[r]$ and $t2[r]$ are expresssed by the following recurrence relations:

$$\begin{aligned} t1[r] &= g \cdot (v+c)^2 + (r-1)s \cdot (2v^2 + t1[r]) \\ t2[r] &= s \cdot v^2 + g \cdot (2(v+c)^2 + t2[r]) + (r-2)s \cdot (2v^2 + t2[r]), \end{aligned}$$

since we need not return to the center after hitting the r^{th} new ray end.

Next we derive an expressions for $p[i]$. Recall that $p[i] = \mathbb{P}($ hit $i-1$ short ray ends$|$ hit $i-1$ ray ends$)$, and the complementary probability $1-p[i] = \mathbb{P}($hit $i-2$ short ray ends$|$ hit $i-1$ ray ends$)$. Imagine an urn with r balls, of which $r-1$ are labeled S and one is labeled L. Assign probabilities $s = \mathbb{P}(S)$ to each of the S-balls and $g = \mathbb{P}(L)$ to the L-ball, with $(r-1)s+g = 1$. Then the probability that we choose $i-1$ S-balls when sampling without replacement is

$$p[i] = \prod_{k=1}^{i-1}(r-k) \cdot \frac{s^{i-1}}{\prod_{k=0}^{i-2}(1-ks)}$$

where the first product counts the number of ordered lists of $i-1$ S-balls from the $r-1$ available. The second factor results because the first ball chosen has probability $s = \frac{s}{1}$ to be an S-ball. Given that the first ball was an S-ball, the 2^{nd} ball chosen has probability $\frac{s}{1-s}$ of being an S-ball. Similarly, the conditional probability that the 3^{rd} ball chosen is an S-ball is $\frac{s}{1-2s}$, and so on. Note that s and g are precisely the probabilities we defined earlier, *i.e.* s is the probability, given that we are at the end of some ray, that this ray is short and g is the probability, given that we are at the end of some ray, that this ray is long. Thus $(r-1)s + g = 1$.

Finally, we argue that

$$\begin{aligned} s &= \frac{v+c}{(r-1)(v+c)+v}, \text{ and} \\ g &= \frac{v}{(r-1)(v+c)+v}. \end{aligned}$$

Let p be the probability of hitting an S-leaf first given that we have hit a leaf of the sparkler. We will find p by conditioning on the first time we are at the v^{th} position on some ray, so either *(a)* we are at the end of a short ray, with probability $\frac{r-1}{r}$, or *(b)* we are at the v^{th} position of the long ray, with probability $\frac{1}{r}$. In *(a)*, we have reached an S-leaf first. In *(b)*, in order to reach an S-leaf first we must return to the center, start the walk again and hit an S-leaf before the L-leaf. This scenario is equivalent to gambler's ruin where we are on a $(v+c)$-path and looking for the probability, given that we are at vertex v, of hitting 0 before $v+c$. By a classical result (e.g., see Feller [2]), this probability is $1-\frac{v}{v+c}=\frac{c}{v+c}$. So this gives the following recurrence for p:

$$p=\frac{r-1}{r}\cdot 1+\frac{1}{r}\cdot\frac{c}{v+c}\cdot p.$$

Solving for p, we find

$$p=\frac{(r-1)(v+c)}{(r-1)(v+c)+v}.$$

Since this is the probability of hitting *any* S-leaf first, and s is the probability of hitting a *particular* S-leaf first, $p=(r-1)s$ and therefore $s=\frac{v+c}{(r-1)(v+c)+v}$. Similarly, $g=1-p=\frac{v}{(r-1)(v+c)+v}$.

Now we can put it all together. The expected cover time for the sparkler is

$$\begin{aligned}E[C] &= h[r]+\sum_{i=1}^{r-1}hr[i]\\ &= \frac{v(2c^2r+c(3+r+2r^2)v+r(1+2r)v^2)}{c(r-1)+rv}-\frac{(-1)^r c(c+2(-1+r)v)(r-1)!}{RF[-r+\frac{c}{c+v},-1+r]}\\ &\quad+\sum_{i=2}^{r-1}\frac{-2v(c+rv)(1-\frac{c(\frac{c}{c+v}-r-1)!RF[1-r,i-1]}{(c+v)(i-r+\frac{c}{c+v}-1)!})}{i-r-1},\end{aligned}$$

where $RF[a,b]$ is the rising factorial $a(a+1)\dots(a+b-1)$. We observe that when $c=0$ our formula reduces to the expected cover time of the star, and when $r=2$, we obtain the expected cover time for a $(2v + c)$-path starting at vertex v.

6 The Petersen graph

6.1 Applying the general method to the Petersen graph

It is feasible to explore the expected cover time of the Petersen graph with the general method due to the many isomorphic subgraphs (relatively small number of non-isomorphic subgraphs) generated by the random walk process.

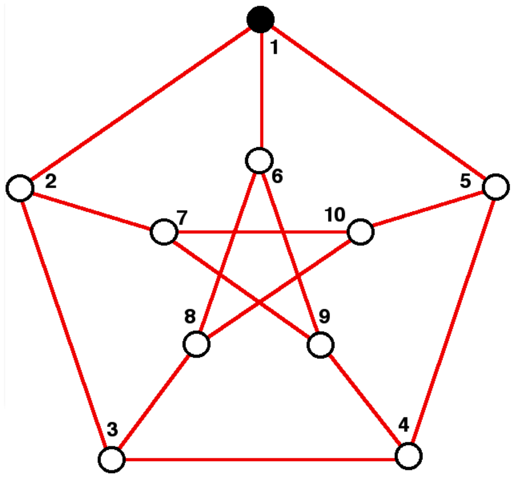


Figure 13: Our "canonical" labeling of the Petersen graph

First, we will use a "canonical" labeling of the graph, as shown in Figure 13. Let $c[\mathcal{A}]$ denote the *expected* remaining cover time of a random walk presently at vertex 1 that has visited the vertices in the list $\mathcal{A}$. For example, $c[1, 2, 3, 5]$ indicates the expected remaining cover time for a random walk that has visited vertices 2, 3, and 5 and is presently at vertex 1. This notation specifies a walk history; $c[1, 2, 3, 5, 6]$ corresponds to a walk configuration in which we are currently at vertex 1 and have previously visited vertices 2, 3, 5, and 6. We will say that the walk is "presently located" at vertex 1, in this case. (Note that not all subsets of $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ form valid configurations: $c[1, 2, 4, 10]$ is not a reachable state, for instance, as vertices 4 and 10 are not adjacent to

1 or 2.)

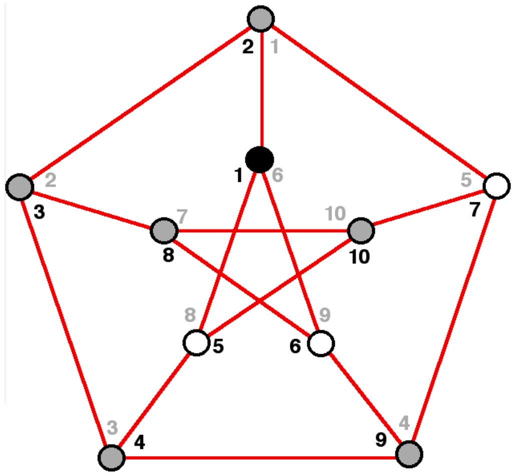


Figure 14: A relabeled walk configuration on the Petersen graph. Gray vertices represented visited vertices, the black vertex is the current location of the walk, gray numbers represent our canonical labeling, and black numbers represent the new labeling.

Why force the random walk configuration to always be at vertex 1 with this notation? In using the general method to solve for the expected cover time of a random walk, we encounter many walk configurations that are isomorphic, but may not initially appear to be. The symmetries of the Petersen graph allow any given walk configuration presently located at a particular vertex to be rewritten as a walk presently located at any other vertex through an appropriate relabeling of the graph. See Figure 14 for an example of a walk relabeled to be presently at vertex 1. We make ourselves aware of the many isomorphic walk configurations and avoid making redundant equations by using relabeling to assume all walks are located at vertex 1. (While the details of our bookkeeping procedures seem tedious, it is only through the careful management of isomorphic walk configurations that we can tackle the cover time of the Petersen graph, which may have thousands of walk states if we approach the problem without a system.)

By similar reasoning, we can relabel any walk configuration with two or more visited vertices to include both vertices 1 and 2: if we've visited two or more vertices, the walk is located at some vertex (relabel this vertex 1) and has visited at least one adjacent vertex (relabel this vertex 2). This pattern does not extend to more than two vertices, though, as with three vertices, we cannot relabel the walk specified by $c[1, 2, 5]$ as one containing vertices 1, 2, and 3.

Another way to reduce the number of equations involved in the general method is to systematically identify isomorphic walk configurations. We will always strive to use terms that contain the longest initial subsequence of consecutive numbers (e.g., opting for $c[1, 2, 3, 4, 9]$ over $c[1, 2, 3, 8, 10]$) and then the smallest number for the next vertex (e.g., opting for $c[1, 2, 5]$ over $c[1, 2, 6]$). Finding the "minimal" isomorphic labeling using these criteria is fairly straightforward once we draw the walk on the canonical labeling, look at the subgraph induced by the already covered vertices, and try to find the longest path from vertex 1 or a 5-cycle.

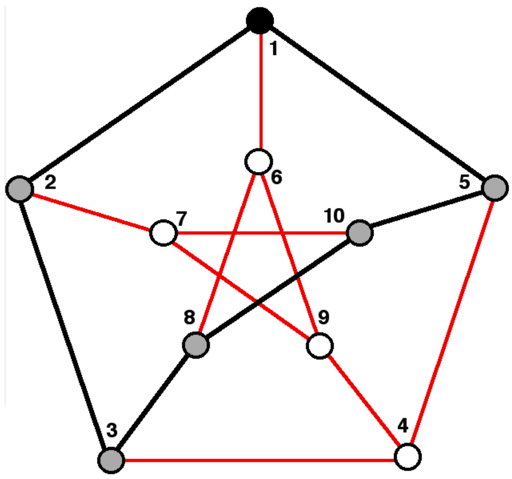


Figure 15: The walk configuration associated with $c[1, 2, 3, 5, 8, 10]$

For example, the graph in Figure 15 corresponding to $c[1, 2, 3, 5, 8, 10]$ has a number of 4-paths, though no 5-cycle. The reader can verify that we can choose one of these 4-paths and relabel it to have vertices 1, 2, 3, and 4, and then the other vertices are forced to be labeled vertices 6, and 9. Therefore $c[1, 2, 3, 5, 8, 10]$ is equivalent to $c[1, 2, 3, 4, 6, 9]$, but we will only use the latter in our equations.

As another example, the graph in Figure 16 corresponding to $c[1, 2, 3, 5, 6, 8]$ contains

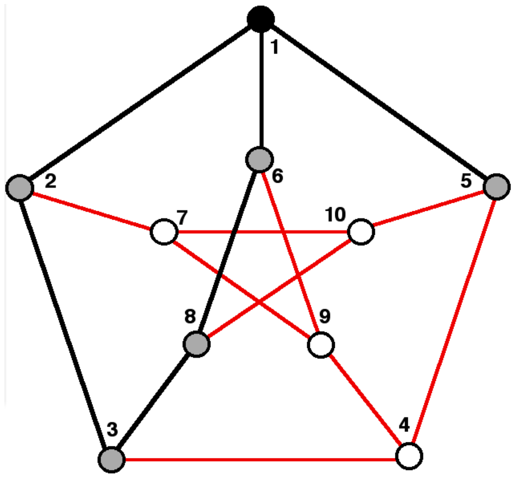


Figure 16: The walk configuration associated with $c[1,2,3,5,6,8]$

a 5-cycle in the induced subgraph. We can relabel this cycle to have vertices 1, 2, 3, 4, and 5 in it, and the vertex not in the induced subgraph is forced to be vertex 6. Thus a more minimal isomorphic term is easily seen to be $c[1,2,3,4,5,6]$, which we will use in our equations.

To actually derive the equations relating the remaining expected cover times from various states, we begin the walk at vertex 1 and examine $c[1]$. From vertex 1, we can walk to vertices 2, 5, or 6. These possible walks (1 to 2, 1 to 5, and 1 to 6) are all isomorphic, so we can say that their expected remaining cover times are identical. Further, the expected remaining cover time for the walk from 1 to 2 is equivalent to the walk that has been to 2 and is currently at 1, so for all three of the possible walks, we will designate the expected remaining cover time as $c[1,2]$. Now, it took one step to go from 1 to 2 (or 5 or 6), so we have that $c[1] = c[1,2] + 1$.

Continuing with the next unknown, suppose we are in the random walk configuration suggested by $c[1,2]$–that is, we have visited vertex 2 and are presently at vertex 1. From vertex 1, we can again visit vertices 2, 5, or 6. If we visit vertex 2 (with probability $\frac{1}{3}$, then we are again in the configuration associated with $c[1,2]$. If we visit vertices 5 or 6 (each with probability $\frac{1}{3}$), then the walk configurations are now equivalent to being at vertex 1 and having visited vertices 2 and 3, which is associated with $c[1,2,3]$. Thus, $c[1,2] = \frac{1}{3}(c[1,2]+1) + \frac{2}{3}(c[1,2,3]+1) = \frac{1}{3}c[1,2] + \frac{2}{3}c[1,2,3] + 1$.

From the configuration given by $c[1,2,3]$ we can again visit vertices 2, 5 or 6 from vertex 1. Visiting vertex 2 thus puts us in a configuration equivalent to that given by $c[1,2,5]$. Visiting vertex 5 from $c[1,2,3]$ is equivalent to $c[1,2,3,4]$. Visiting vertex 6 is likewise equivalent to $c[1,2,3,4]$, so $c[1,2,3] = \frac{1}{3}c[1,2,3] + \frac{2}{3}c[1,2,3,4] + 1$.

We continue this process, systematically generating new equations by examining the $c[\mathcal{A}]$ terms we encounter. This process terminates when there are no new walk configuration states used in a previous equation to examine. After extensive equation double-checking, this method yielded 46 non-isomorphic incomplete walk configurations, and thus 46 linear equations. These equations are displayed in full in the appendix. We used *Mathematica* to solve this system of equations for $c[1]$, the cover time of the graph starting from any vertex, which was found to be $\frac{11964221}{393484}$, or about 30.406 steps. In a simulation with 10 million trials, we found an average cover time of 30.403 for the Petersen graph, which agrees with this result to two decimal places.

6.1.1 Variance and the general method

With the full solution for the $c[\mathcal{A}]$ terms now at our disposal thanks to *Mathematica*, we can easily examine the second moment of the cover time for the Petersen graph. Using the same scheme as above, we now let $d[\mathcal{A}]$ refer to the expected squared remaining cover time (second moment) from a walk that has visited the vertices in the set $\mathcal{A}$ and is currently at vertex 1. Then if $c[\mathcal{A}] = \frac{1}{3}(c[\mathcal{B}]+1) + \frac{1}{3}(c[\mathcal{C}]+1) + \frac{1}{3}(c[\mathcal{D}]+1)$ (that is, we can get to walk configurations $\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$ from $\mathcal{A}$ by traveling to vertices 2, 5, and 6, respectively), it follows that $d[\mathcal{A}] = \frac{1}{3}(d[\mathcal{B}] + 2c[\mathcal{B}] + 1) + \frac{1}{3}(d[\mathcal{C}] + 2c[\mathcal{C}] + 1) + \frac{1}{3}(d[\mathcal{D}] + 2c[\mathcal{D}] + 1)$. The $c[\mathcal{B}], c[\mathcal{C}], c[\mathcal{D}]$ terms are already known from the earlier solution, so now we have 46 linear equations in terms of the unknown $d[\mathcal{A}]$s. Using *Mathematica* to solve this system, we find that $d[1]$ is $\frac{42416487705755}{38707414564}$. Thus from the formula for variance using second moment and squared expectation, the variance in cover times of random walks on the Petersen graph is $\frac{26523366686179}{154829658256}$, or about 171.307. Therefore the standard deviation in cover times of the Petersen graph is about 13.088 steps. In a simulation with 10 million trials, we found a sample standard deviation of approximately 13.086.

Higher moments can be found in a similar fashion, using previous moments. We did not perform this analysis, but one could obtain the skewness and kurtosis in the distribution of cover times of a random walk on the Petersen graph as such.

7 Other possible areas of exploration

In the course of our work, we approached other graphs which we decided not to pursue further in the short research period that we had since the recurrence relations and methods necessary to solve the expectation and variance of the cover time seemed quite challenging. What follows is a list of other graphs of interest:

- The "flower" graph — m triangular "petals" joined at a central vertex. The "flower"

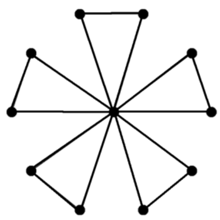


Figure 17: The flower graph with 5 petals.

graph F_n is a central vertex with n triangles (which we call petals) attached. There are $2n + 1$ vertices and $3n$ edges. A "petal" is one of the triangles attached to the center vertex. A "leaf" is any vertex except for the center. A "leaf" is covered if one of the vertices of a petal is covered but the other is not.

Let $C_{a,b}$ be the expected cover time given that we're at the center vertex and given that a petals and b leaves have been covered. So, the overall expected cover time is $C_{0,0}$.

Let $L_{a,b}$ be the expected cover time given that we're at a leaf in an uncovered petal and given that a petals and b leaves have been covered. Finally, let $P_{a,b}$ be the expected cover time, given that we're at a vertex in a covered petal and given that a petals and b leaves have been covered. So, we have that:

$$\begin{aligned} C_{0,0} &= 1 + L_{0,1} \text{ and} \\ L_{0,1} &= \frac{1}{2}(1 + C_{0,1}) + \frac{1}{2}(1 + P_{1,0}), \text{ and} \\ P_{1,0} &= \frac{1}{2}(1 + P_{1,0}) + \frac{1}{2}(1 + C_{1,0}), \text{ etc.} \\ \text{Also, } C_{0,1} &= \frac{1}{2n}(1 + P_{0,1}) + \frac{1}{2n}(1 + P_{1,0}) + \frac{2n-2}{2n}(1 + P_{0,2}). \end{aligned}$$

The last equality follows since if 1 leaf has been covered and we're at the center, then we can either go back to that 1 leaf (with probability $\frac{1}{2n}$) or go to the other leaf of its petal (with probability $\frac{1}{2n}$) or go to some new petal and thus hit a new leaf (with remaining probability $\frac{2n-2}{2n}$). Using these equations, we can set up some recurrence relations for these three variables in general. When we tried to do this, we got recurrence relations in terms of multiple self-referencing variables which programs like *Mathematica* cannot solve. More insight is needed to reduce these recurrence relations to something easily solvable.

- The "China Buffet" graph — two complete graphs joined at a central vertex. The

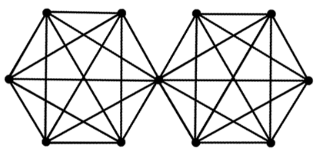


Figure 18: The China Buffet graph with $k = 6$.

China Buffet graph, named for a Chinese restaurant in Northfield, MN, USA, seemed good to pursue since the expected cover time for the complete graph on k vertices is well known and related intimately to the coupon collector's problem. To determine the expectation of the cover time, we need to condition on the number of vertices visited in a particular half of the graph and the half of the graph that the walk is currently in.

- The "infinity" graph – two cycles joined at a central vertex. Now that we have the

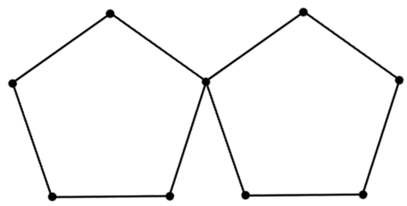


Figure 19: The infinity graph with $n = 5$.

exact distribution of the cover time of the n-cycle, it seems plausible that we could determine the expectation of the cover time for two cycles joined at one vertex. For this graph, we run into the same issue as we did with the China Buffet graph. We need to condition on the number of vertices visited in a particular half of the graph and the half of the graph that the walk is currently in.

- The banana graph — A star graph with stars at each of the ends. The Banana

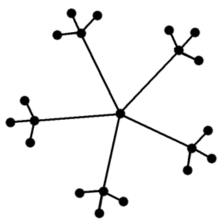


Figure 20: The banana graph with $R = 5$, $r = 3$, $V = 1$, and $v = 1$.

graph is a star with R rays of length V. At each ray end of the star, we add another

r rays with v vertices per ray. This graph is a tree and so looks interesting.

8 Acknowledgements

We wish to particularly thank Robert P. Dobrow, PhD. at Carleton College for his tremendous help, mentorship, and support throughout this project. We would also like to thank David Liben-Nowell for insight into applications of this work; Jonah Ostroff for some comments on Catalan Numbers; David Lonoff for some help with TEXing; and the many other professors, students, and others who supported us and our project.

References

[1] Blom, G. and Sandell, D. (1992) Cover Times for Random Walks on Graphs. *Math. Scientist* **17**, 111-119.

[2] Feller, W. (1968) *An Introduction to Probability Theory and Its Applications,* Vol. I, 3rd Edn. Wiley, New York.

[3] Krattenthaler, C. and Mohanty, S.G. (1993) On lattice path counting by major and descents. *Europ. J. Combin.* **14**, 43-51.

[4] Mohanty, S.G. (1979) *Lattice Path Counting and Applications,* Academic Press, New York.

[5] Wilf, H.S. (1989) The Editor's Corner: The White Screen Problem *American Mathematical Monthly* **96**, 704-707.

9 Appendix

9.1 Petersen graph expectation equations

$$
\begin{aligned}
c1 &= c12+1 \\
c12 &= \frac{1}{3}(c12+1)+\frac{2}{3}(c123+1) \\
c123 &= \frac{1}{3}(c125+1)+\frac{2}{3}(c1234+1) \\
c125 &= \frac{2}{3}(c123+1)+\frac{1}{3}(c1237+1) \\
c1234 &= \frac{1}{3}(c12345+1)+\frac{1}{3}(c1235+1)+\frac{1}{3}(c12349+1) \\
c1237 &= \frac{1}{3}(c1256+1)+\frac{2}{3}(c12348+1) \\
c12345 &= \frac{2}{3}(c12345+1)+\frac{1}{3}(c123479+1) \\
c1235 &= \frac{1}{3}(c1235+1)+\frac{1}{3}(c1234+1)+\frac{1}{3}(c12347+1) \\
c12349 &= \frac{1}{3}(c12346+1)+\frac{1}{3}(c123469+1)+\frac{1}{3}(c123457+1) \\
c1256 &= c1237+1 \\
c12348 &= \frac{2}{3}(c123458+1)+\frac{1}{3}(c12357+1) \\
c123479 &= \frac{1}{3}(c123456+1)+\frac{1}{3}(c1234579+1)+\frac{1}{3}(c1234579+1) \\
c12347 &= \frac{1}{3}(c12356+1)+\frac{1}{3}(c123458+1)+\frac{1}{3}(c123489+1) \\
c12346 &= \frac{1}{3}(c123510+1)+\frac{1}{3}(c12349+1)+\frac{1}{3}(c123457+1) \\
c123469 &= \frac{2}{3}(c123469+1)+\frac{1}{3}(c12345710+1) \\
c123457 &= \frac{1}{3}(c123456+1)+\frac{1}{3}(c123458+1)+\frac{1}{3}(c1234789+1) \\
c123458 &= \frac{1}{3}(c123457+1)+\frac{1}{3}(c123458+1)+\frac{1}{3}(c1234579+1) \\
c12357 &= \frac{1}{3}(c12356+1)+\frac{1}{3}(c12348+1)+\frac{1}{3}(c123478+1) \\
c123456 &= \frac{2}{3}(c123457+1)+\frac{1}{3}(c123479+1) \\
c1234579 &= \frac{1}{3}(c1234568+1)+\frac{1}{3}(c1234579+1)+\frac{1}{3}(c123457810+1)
\end{aligned}
$$

$$
\begin{aligned}
c12356 &= \frac{2}{3}(c12347+1)+\frac{1}{3}(c12357+1)\\
c123489 &= \frac{1}{3}(c123467+1)+\frac{1}{3}(c1234578+1)+\frac{1}{3}(c1234568+1)\\
c12345710 &= \frac{2}{3}(c1234568+1)+\frac{1}{3}(c123478910+1)\\
c1234789 &= \frac{1}{3}(c1234567+1)+\frac{1}{3}(c12345789+1)+\frac{1}{3}(c12345679+1)\\
c1234679 &= \frac{1}{3}(c1234679+1)+\frac{1}{3}(c1234568+1)+\frac{1}{3}(c12345679+1)\\
c123478 &= \frac{1}{3}(c123567+1)+\frac{2}{3}(c1234589+1)\\
c1234568 &= \frac{1}{3}(c12345710+1)+\frac{2}{3}(c1234579+1)\\
c123457810 &= \frac{2}{3}(c12345679+1)+\frac{1}{3}(c1234578910+1)\\
c123510 &= \frac{1}{3}(c12346+1)+\frac{1}{3}(c12346+1)+\frac{1}{3}(c1234710+1)\\
c123467 &= \frac{1}{3}(c123569+1)+\frac{1}{3}(c123489+1)+\frac{1}{3}(c1234578+1)\\
c1234578 &= \frac{1}{3}(c1234567+1)+\frac{1}{3}(c1234589+1)+\frac{1}{3}(c12345789+1)\\
c123478910 &= \frac{1}{3}(c123456710+1)+\frac{2}{3}(c123456789+1)\\
c1234567 &= \frac{1}{3}(c1234567+1)+\frac{1}{3}(c1234578+1)+\frac{1}{3}(c1234789+1)\\
c12345789 &= \frac{1}{3}(c12345678+1)+\frac{1}{3}(c12345789+1)+\frac{1}{3}(c123456789+1)\\
c12345679 &= \frac{1}{3}(c12345679+1)+\frac{2}{3}(c123457810+1)\\
c123567 &= \frac{1}{3}(c123567+1)+\frac{2}{3}(c123478+1)\\
c1234589 &= \frac{2}{3}(c1234578+1)+\frac{1}{3}(c12345679+1)\\
c1234578910 &= \frac{2}{3}(c123456789+1)+\frac{1}{3}(1)\\
c1234710 &= \frac{1}{3}(c123569+1)+\frac{1}{3}(c1234568+1)+\frac{1}{3}(c12348910+1)\\
c123569 &= \frac{2}{3}(c123467+1)+\frac{1}{3}(c1234710+1)\\
c123456710 &= \frac{2}{3}(c12345678+1)+\frac{1}{3}(c123478910+1)\\
c123456789 &= \frac{1}{3}(c1234578910+1)+\frac{2}{3}(c123456789+1)\\
c12345678 &= \frac{1}{3}(c123456710+1)+\frac{2}{3}(c12345789+1)
\end{aligned}
$$

$$\begin{aligned}
c12348910 &= \frac{1}{3}(c12346710+1)+\frac{2}{3}(c12345678+1)\\
c12346710 &= \frac{1}{3}(c12348910+1)+\frac{1}{3}(c12356910+1)+\frac{1}{3}(c12345678+1)\\
c12356910 &= \frac{1}{3}(c12346710+1)+\frac{2}{3}(c12346710+1)
\end{aligned}$$