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

ON THE PROBLEM OF A MAGIC CANDY JAR

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

WEBER, BILL
DEPARTMENT OF MATHEMATICS
FORT HAYS STATE UNIVERSITY
HAYS, KANSAS

On the Problem of a Magic Candy Jar

Abstract

An interesting candy jar problem has been investigated. Several recur-rence relations have been discovered and verified. The format of a closed formula of these recurrence relations has been established. A computer simulation is reported.

1 Introduction

Consider the following "Magic Candy Jar" problem. There is a "Magic Candy Jar" that contains infinity many well mixed candies. Each has one of five distinct colors. Suppose a person randomly picks one candy at a time from the jar. The person will stop as soon as he/she gets all five colors. What is the expected number of draws needed?

The assumption of "infinity many well mixed", just asserts that for each pick, each color has 1/5 equal chance to be picked.

This is not a new problem. It is also known as the "Coupon Collector Problem" [1]. It has also been referred to as the "Dixie Cup Problem" as in [3] and [2].

In this paper, we approached this alternative "Magic Candy jar" problem in a different way. Let $C_k(n)$ denote the number of ways to draw n candies from the "magic candy jar" with k different colors of candies such that all colors are presented and the color of last draw is unique.

We first established the recurrence relation of $C_k(n)$. Then we found the closed formula for $C_3(n), C_4(n)$ and the format of the closed formula of $C_k(n)$ for any $k \geq 5$. With this format, the closed formula of $C_k(n)$ can be calculated with the help of computer software for any $k \geq 5$.

Finally, we presented the solution of original "Magic Candy Jar" problem and a computer simulation to support the problem solution.

2 Recurrence Relation

Let $C_k(n)$ denote the number of ways to draw n candies from the "magic candy jar" with k different colors of candies such that all colors are presented and the color of the last draw is unique. Note that $C_k(n) = 0$ if $n < k$.
Let's first prove several obvious facts.

Lemma 2.1. *For all* $n \geq 2, C_2(n) = 2$

Proof. We have two colors, say A and B. The last draw will end with one of the colors and before that, all candies' color shall be the other color. So there are two ways. Namely, the first n-1 candies have color A and last candy has color B; or the first n-1 candies have color B and last candy has color A. □

To get $C_k(n)$ for $k \geq 3$, we need the following lemma.

Lemma 2.2. *For any* $k \geq 3$*, we have*

$$C_{k+1}(n) = (C_k(n-1)+C_k(n-2)\times k+C_k(n-3)\times k^2+...+C_k(k)\times k^{n-k-1})\times(k+1)$$

$$= (k+1)\sum_{i=0}^{n-k-1} C_k(n-1-i)k^i$$

Proof. Imagine that we draw the last candy first. The last candy has (k+1) ways to be drawn. After drawing the last candy, in order to get all colors and the last candy's color is unique, we need to have the first n-1 candies to have the rest of the k colors. We can divide this into the following $n - k - 1$ cases:

- We don't get all k colors until draw n-1. We have $C_k(n-1)$ different ways to do so.
- We don't get all k colors until draw n-2. We have $c_k(n-2) \times k$ different ways to do so since the draw n-1 has k different ways to do it.
- We don't get all k colors until draw n-3. We have $C_k(n-3) \times k^2$ different ways to do so since the draw (n-1) and (n-2) has k^2 different ways to do it.
- ...
- We don't get all k colors until draw $n - (n - k)$, i.e. we get all k colors in first k draws.

We can see the pattern that if we don't get all k colors until draw n - i, then we have $C_k(n - i) * k^{i-1}$ since for last (i-1) candies, each of them can be any of these k colors. The lemma follows after we add up the result of all cases. □

We coded this recurrence relation in Python. Here are several results when $k = 3, 4, 5$

$$C_3(3) = 6; C_3(4) = 18; C_3(5) = 42; C_3(6) = 90; C_3(7) = 186; C_3(8) = 378;$$

$$C_4(4)=24; C_4(5)=144; C_4(6)=600; C_4(7)=2160; C_4(8)=7224;$$
$$C_5(5)=120; C_5(6)=1200; C_5(7)=7800; C_5(8)=42000;$$

We claim the following closed formula

Theorem 2.3.
$$C_3(n)=3\times 2^{n-1}-6$$

Proof. By lemma 2.2, we have

$$\begin{aligned}C_3(n)&=\left(C_2(n-1)+C_2(n-2)\times 2+C_2(n-3)\times 2^2+...C_2(2)\times 2^{n-3}\right)\times 3\\&=(2+2^2+2^3+...+2^{n-2})\times 3\\&=(2^{n-1}-2)\times 3\\&=-6+\frac{3}{2}\times 2^n\end{aligned}$$

□

We can similarly get the closed formula for k = 4.

Theorem 2.4.
$$C_4(n)=12-6\times 2^n+\frac{4}{3}\times 3^n$$

Proof. By lemma 2.1 and Theorem 2.3, we have

$$\begin{aligned}C_4(n)&=\left(C_3(n-1)+C_3(n-2)\times 3+C_3(n-3)\times 3^2+...+C_3(3)\times 3^{n-4}\right)\times 4\\&=((3\times 2^{n-2}-6)+(3\times 2^{n-3}-6)\times 3+...+(3\times 2^2-6)\times 3^{n-4})\times 4\\&=\left(\sum_{i=1}^{n-3}2^{n-1-i}\times 3^i-6\times\sum_{i=0}^{n-4}3^i\right)\times 4\\&=\left(\sum_{i=0}^{n-1}2^{n-1-i}\times 3^i-2^{n-1}-2\times 3^{n-2}-3^{n-1}-6\times\sum_{i=0}^{n-4}3^i\right)\times 4\\&=\left(3^n-2^n-2^{n-1}-2\times 3^{n-2}-3^{n-1}-6\times\frac{3^{n-3}-1}{2}\right)\times 4\\&=12-6\times 2^n+\frac{4}{3}\times 3^n\end{aligned}$$

□

3 Computation in General

To compute the closed formula in general, we first prove the following lemma.

Lemma 3.1. *For any fixed integer $k > 2$, positive integers a and b, and integer $n > k$, we have*

$$\sum_{i=0}^{n-k-1} a^i b^{n-1-i} = \alpha a^n + \beta b^n$$

where α and β are independent on n.

Proof.

$$\sum_{i=0}^{n-k-1} a^i b^{n-1-i}$$

$$= \sum_{i=0}^{n-1} a^i b^{n-1-i} - \sum_{i=n-k}^{n-1} a^i b^{n-1-i}$$

$$= \frac{a^n - b^n}{a-b} + a^n \sum_{i=n-k}^{n-1} \frac{b^{n-1-i}}{a^{n-i}}$$

If we substitute the index i with $j = i - (n-k)$, then $n - i = k - j$ and $n - i - 1 = k - j - 1$. When $i = n - k$, $j = 0$ and when $i = n - 1$, $j = k - 1$. Hence, the above expression continues as

$$= \frac{a^n - b^n}{a-b} + a^n \sum_{j=0}^{k-1} \frac{b^{k-j-1}}{a^{k-j}} = \alpha a^n + \beta b^n$$

where $\alpha = \frac{1}{a-b} + \sum_{j=0}^{k-1} \frac{b^{k-j-1}}{a^{k-j}}$ and $\beta = -\frac{1}{a-b}$. It is easy to see that both α and β only depend on k, a, and b. □

Now we can have the following main result.

Theorem 3.2. *$C_k(n)$ has a closed formula in the form of*

$$C_k(n) = \sum_{i=1}^{k-1} a_{k,i} \times i^n$$

where $a_{k,1}, a_{k,2}, ..., a_{k,k-1}$ are constants that are independent on n.

Proof. We prove by mathematical induction to k. By Theorem 2.3 and Theorem 2.4, we know that this theorem is true for k = 3 and k = 4. Now assume that it is true for k, i.e.

$$C_k(n) = \sum_{i=1}^{k-1} a_{k,i} \times i^n$$

for all $n \geq k$, where $a_{k,i}$ are constants independent on n, then by Lemma 2.2 and Lemma 3.1, we have

$$C_{k+1}(n) = (k+1) \sum_{i=0}^{n-k-1} C_k(n-1-i) \times k^i$$

$$= (k+1) \sum_{i=0}^{n-k-1} (\sum_{j=1}^{k-1} a_{k,j} \times j^{n-1-i}) \times k^i$$

$$= (k+1)(\sum_{j=1}^{k-1} a_{k,j} (\sum_{i=0}^{n-k-1} j^{n-1-i} \times k^i))$$

$$= (k+1)(\sum_{j=1}^{k-1} a_{k,j}(\alpha_j j^n + \beta_j k^n)$$

$$= \sum_{i=1}^{k} a_{k+1,i} \times i^n$$

for some constants $a_{k+1,i}$, where $i = 1, 2, ..., k+1$, that are independent on n. The last step is just combining the like terms with respect to j^n for $j = 1, 2, ..., k$. □

Theorem 3.2 gives us a way to compute the closed formula in general. We can first use Lemma 2.2, along with a computer program, to calculate $C_k(k), C_k(k+1), ..., C_k(2k-2)$. Then we simply solve the linear system

$$Ax = b$$

where A is a $(k-1) \times (k-1)$ matrix with entry $A_{i,j} = i^{i-1+k}$. b is a column vector whose ith row element is $C_k(k+i-1)$. Here the matrix and vector index all start from 1.

There are many algorithms/tools to solve this system. In theory, $x = A^{-1}b$ will be the a_k in the closed formula of $C_k(n)$.

Using k = 5 as an example, we have

Corollary 3.2.1.

$$C_5(n) = -20 + 15 \times 2^n - \frac{20}{3} \times 3^n + \frac{5}{4} \times 4^n$$

Proof. By Lemma 3.2, we have

$$C_5(n) = a_1 + a_2 \times 2^n + a_3 \times 3^n + a_4 \times 4^n$$

Using a software tool such as Wolfram Alpha, we can solve the system $Ax = b$ where

$$A = \begin{bmatrix} 1 & 2^5 & 3^5 & 4^5 \\ 1 & 2^6 & 3^6 & 4^6 \\ 1 & 2^7 & 3^7 & 4^7 \\ 1 & 2^8 & 3^8 & 4^8 \end{bmatrix}$$

$$b = \begin{bmatrix} 120 \\ 1200 \\ 7800 \\ 42000 \end{bmatrix}$$

We have $a_1 = -20; a_2 = 15; a_3 = \frac{20}{3}; a_4 = \frac{5}{4}$

□

4 Computer Program Simulation

Now we can answer the original problem: what is the expected number of candies to draw before a person gets all colors drawn?

Theorem 4.1. *The expected number of draws to get all k colors is*

$$\sum_{n=k}^{\infty} n \frac{C_k(n)}{k^n}$$

Proof. Denote X as the random variable whose value is the number of draws to get all colors. Then X can have values k, k+1, ..., n, ... where

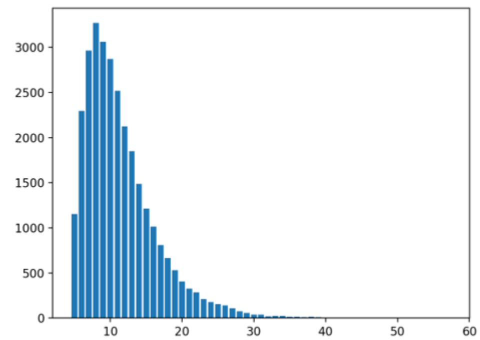
$$P(X = n) = \frac{C_k(n)}{k^n}$$

The probability is true because there are k^n different ways to draw since each draw has k different outcome colors. Then by definition of expectation, we conclude the formula in the theorem. □

We wrote a Python program to verify the result. First, we coded the formula in the program and summed to n = 100. We also simulated 30000 times to randomly draw the candy from a "Magic Candy Jar". The following table shows the results of these two approaches for k = 2, 3, 4, 5, 6, 7, 8. Here k is the number of colors in the "Magic Candy Jar".

	2	3	4	5	6	7	8
Formula	3	5.5	8.333	11.416	14.699	18.149	21.741
Simulation	2.9978	5.5388	8.3178	11.369	14.732	18.196	21.686

The following histogram supports the conclusion. It is the result of 30000 simulations on k = 5. The horizontal axis is the number of draws to get all 5 colors from one simulation. The vertical axis is the number of simulations



5 Future Work

We are going to work on establishing the recurrence relation between coefficients of $C_k(n)$ and $C_k(n+1)$. Basically, If $C_k(n)$ has a closed formula in the form of

$$C_k(n) = \sum_{i=1}^{k} a_i \times i^n$$

and $C_{k+1}(n)$ has close formula in form of

$$C_{k+1}(n) = \sum_{i=1}^{k+1} b_i \times i^n$$

then, we shall be able to represent b_i in terms of a_i. We will address this somewhere else.

References

[1] Wikipedia coupon collector's problem. https://en.wikipedia.org/wiki/Coupon_collector%27s_problem.

[2] William Feller. *An Introduction to Probability Theory and Its Applications, vol. I.* John Wiley & Sons, 1968.

[3] Donald J. Newman and Lawrence Shepp. The double dixie cup problem. *American Mathematical Monthly*, 67(1), 1960.