TFT Calculator Proofs

Sergio André López Pereo

December 20, 2023

1 Introduction

Well, first, let's make it as simple as possible. I will assume you are a TFT player with the basic knowledge of the game. As everybody know tft has a shop where you can choose from 5 different champs from the pool to buy. Let's work with just one of those 5 slots. I will assume those are regular shops (no headliner).

1.1 Single slot

Let $P(C_i)$ = The probability that it is a slot of cost i

Also let $P(C_{name})$ = The probability of it being a slot of the desired champ

The first probability is fixed, given by the current user level. And we can se the second one as a Bernoulli's distribution. A Bernoulli's distribution is defined as:

$$f_x(x) = P(X = x) = p^x(1-p)^{1-x}, x = 0, 1$$

The p value is the probability of the Bernoulli's experiment as being positive (where x=1).
Our experiment is positive if:

- The slot is of the desired cost
- The slot has the desired champ

$$\therefore P(C_i \cap C_{name})$$

Now let's about the second event as we already have the probability of the first event.

Let M_{name} = Champs of the same cost left in the pool

Also let K_{name} = Copies of the desired champ left in the pool

So given this information we can say that:

$$P(C_{name}) = \frac{M_{name}}{K_{name}}$$

This happens because if the slot happens to be of the desired cost, it can only have champs of the pool of that cost. So the complete probability of the slot being of a desired cost and a desired champ is:

$$P(C_i \cap C_{name}) = P(C_i)P(C_{name})$$

As they are not exclusive events and

$$\therefore P(C_i \cap C_{name}) = P(C_i)\frac{M_{name}}{K_{name}}$$

$$\therefore p = P(C_i)\frac{M_{name}}{K_{name}}$$

1.2 Shop mechanic

Now let's add the shops mechanic. We can see the shop as a set of 5 different Bernoulli's experiments with the same probabilities. Therefore we can use the Binomial distribution to model this mechanic. A binomial distribution is defined as:

$$f_x(x) = P(X = x) = \binom{n}{x} p^x q^{n-x}, x = 0, 1, \ldots, n$$

Where p is the probability of success of the Bernoulli's experiment; q is the probability of fail of the Bernoulli's experiment ; n is the amount of Bernoulli's experiments and x is the amount of successes of the Bernoulli's experiment's.

Let X be the number of copies of the desired champ we can see in a specific shop; n is 5 ; p is already calculated and q is just the complement of p.

$$\therefore Bin(5, P(C_i)\frac{M_{name}}{K_{name}})$$
$$= \binom{5}{x}(P(C_i)\frac{M_{name}}{K_{name}})^x(1 - (P(C_i)\frac{M_{name}}{K_{name}}))^{5-x}, x = 0, 1, \ldots, 5$$

For simplicity I will call this function just g(x). This p value assumes that the shop champs are calculated all at the same time. If it is calculated one by one then the probability of success will be changing dynamically based on the pool size. As this is not strictly defined by Riot Games I will keep this simple way of calculating it.

1.3 Events

Now let's add our desired events.

$$\text{Let } E_i = \text{Seeing i copies of the champ in a rolldown}$$
$$\text{Also let } T = \text{Quantity of stores}$$
$$\therefore = \frac{\text{\# gold to roll}}{2}$$

We can say that

$$P(E_0) = g(0)^T$$

This is because there are going to be T stores where the champion will not appear. And as the events happening on each shop are independent and not exclusive. The probability of they not having the champion is the multiplication of the probability of that event on each shop. That means g(0) is multiplied T times. There's no possible permutations so no further calculation is needed. Now let's get into the other events.

$$P(E_1) = \binom{T}{1} g(1)g(0)^{T-1}$$

The reasoning behind this is kind of the same as the equation before. There is 1 shop where it appears the champion and T-1 shops where it doesn't appear. So we have a multiplication of 1 g(1) and T-1 g(0)'s. But this is only one case. In fact that single shop with the champion can be any of those shops. So we need to multiply that by the amount of possible permutations. This are not simple permutations, in fact, they are permutations of groups with indistinguishable members. To solve this kind of problem we need to use the multinomial coefficient. This coefficient is defined as

$$\binom{n}{k_1, k_2, \ldots, k_m} = \frac{n!}{k_1! k_2! \ldots k_m!}$$

Where

$$\sum_{i=1}^{m} k_i = n$$

So in this case we have two groups. One with T-1 members where all of them are g(0). And one where it's single member is g(1). So the amount of permutations of this is

$$\binom{T}{T-1,1} = \frac{T!}{(T-1)!1!} = \binom{T}{1}$$

Now

$$P(E_2) = \binom{T}{1} g(2)g(0)^{T-1} + \binom{T}{2} g(1)^2 g(0)^{T-2}$$

Here we have two cases. One where 1 shop has 2 copies of the desired champ. And one where 2 shops have 1 copy of the champ. This is kinda the same as having to get all the possible ways to sum 2 with 1 and 2. The second case has a different coefficient cause:

$$\binom{T}{T-2,2} = \frac{T!}{(T-2)!2!} = \binom{T}{2}$$

$$P(E_3) = \binom{T}{1} g(3)g(0)^{T-1} + \frac{T!}{(T-2)!} g(1)g(2)g(0)^{T-2} + \binom{T}{3} g(1)^3 g(0)^{T-3}$$

The second and third coefficient are differents because

$$\binom{T}{T-2,1,1} = \frac{T!}{(T-2)!1!1!} = \frac{T!}{(T-2)!}$$

$$\binom{T}{T-3,3} = \frac{T!}{(T-3)!3!} = \binom{T}{3}$$

And the respective cases are

$$\{(3), (1,2), (1,1,1)\}$$

$$P(E_4) = \binom{T}{1} g(4)g(0)^{T-1} + \binom{T}{2} g(2)^2 g(0)^{T-2} + \frac{T!}{(T-2)!} g(1)g(3)g(0)^{T-2}$$
$$+ \frac{T!}{(T-3)!2!} g(1)^2 g(2)g(0)^{T-3} + \binom{T}{4} g(1)^4 g(0)^{T-4}$$

Cases

$$\{(4), (2,2), (1,3), (1,1,2), (1,1,1,1)\}$$

$$P(E_5) = \binom{T}{1} g(5)g(0)^{T-1} + \frac{T!}{(T-2)!} g(1)g(4)g(0)^{T-2} + \frac{T!}{(T-2)!} g(2)g(3)g(0)^{T-2}$$
$$+ \frac{T!}{(T-3)!2!} g(1)g(2)^2 g(0)^{T-3} + \frac{T!}{(T-3)!2!} g(3)g(1)^2 g(0)^{T-3} + \frac{T!}{(T-4)!3!} g(2)g(1)^3 g(0)^{T-4}$$
$$+ \binom{T}{5} g(1)^5 g(0)^{T-5}$$

Cases

$$\{(5), (1,4), (2,3), (1,2,2), (1,1,3), (1,1,1,2), (1,1,1,1,1)\}$$

1.3.1 Added store complexity

There's only one thing left. The dynamically changing pool between stores. Between this stores the pool changes as soon as you start buying the unit you want. That makes it less likely to hit it once you start buying it. So the order in which this stores appear matter. So I have to consider some cases in this where I can make this offset of buyed units.

The single shops with the required amount of units stay the same as we assume that in the stores the hand is calculated at the same time. When have X amount of shops with just one copy it is pretty easy. I just need to re-calculate k once one of them appears. When we have 2 stores with different amount of copies I need to offset differently when one of them appears first. If I have a (2,3) case. Once the first shop with two copies appear, I recalculate k and m with an offset of 2. This makes k=k-offset and m=m-offset. Otherwise I recalculate k and m with an offset of 3.

Cases like (1,1,2) and (1,1,1,2) are more complex than that. And honestly I'm not sure how to avoid calculating all the possible scenarios and changing the offset based on the case. And right now I don't want to do it. Also I don't like algorithms that computational expensive. In this specific cases plus (2,2,1) and (1,1,3) I will assume the

probabilities will not change dynamically. This will make the probability of E_4 and E_5 be higher than the real probability. But I believe it is for a negligible amount. Let me know if you have ideas to circunvent this problem!