Calculating Artifact Roll Outcomes

Brandon Fowler

April 21, 2026

This document provides an overview of the approach taken to calculate all possible outcomes for an artifact in Genshin Impact.

Contents

1 Sub-Stat Probability

Unique combinations are computed and the probability of each corresponding permutation is calculated based on the stat weights.

The formula for the likelihood of a permutation, where w_i is the weight of sub-stat i, is:

$$\left(\frac{w_1}{Total}\right)\left(\frac{w_2}{Total - w_1}\right)\left(\frac{w_3}{Total - w_1 - w_2}\right)\ldots$$

For a given unique combination, all of these probabilities are added together for all permutations that correspond to that unique combination.

The sub-stat probability is then the sum of probabilities of combinations that are valid given the inputted constraints,

$$SubstatProbability = \sum_{valid} p$$

2 Roll Probability

The probability for a given sub-stat outcome is:

$$p\left(\frac{1}{InitialValueOutcomes}\right)\left(\frac{1}{RollOutcomes}\right)\left(\frac{1}{RollValueOutcomes}\right)$$

Where:

- p is the probability of the stat combination.
- $InitialValueOutcomes$ the number of value outcomes for the initial states.
- $RollOutcomes$ is the number of possible roll outcomes.
- $RollValueOutcomes$ the number of value outcomes for the rolled stats.

Since the sub-stat probability is separately defined above, we have:

$$RollProbability \times SubstatProbability = p\left(\frac{1}{InitialValueOutcomes}\right)\left(\frac{1}{RollOutcomes}\right)\left(\frac{1}{RollValueOutcomes}\right)$$

$$RollProbability \times \sum_{valid} p = p\left(\frac{1}{InitialValueOutcomes}\right)\left(\frac{1}{RollOutcomes}\right)\left(\frac{1}{RollValueOutcomes}\right)$$

$$RollProbability = \left(\frac{p}{\sum_{valid} p}\right)\left(\frac{1}{InitialValueOutcomes}\right)\left(\frac{1}{RollOutcomes}\right)\left(\frac{1}{RollValueOutcomes}\right)$$

Let:

- a - The number of sub-stats
- a_r - The number of rollable sub-stats
- b - The number of roll values
- n - The number of sub-stat rolls

Then, there are $a = 4$ sub-stats to roll initially, $a^* = 4$ sub-stats choose from each time for each roll, and $b = 4$ roll values a roll can have (70%, 80%, 90%, and 100%).

- $InitialValueOutcomes = b^a = 4^4$ - There are 4 sub-stats to assign base values to
- $RollOutcomes = a_r^n = 4^n$ - There are n sub-stats rolls that each choose a sub-stat
- $RollValueOutcomes = b^n = 4^n$ - The n sub-stats rolls needs value assigned

Thus, the total number of possibilities is:

$$(b^{a+n})(a_r^n) = (4^{4+n})(4^n)$$

2.1 Roll Permutations

To preserve the base of 4, each guaranteed roll is counted as $\frac{4}{\#ofguaranteedrolls} = \frac{4}{2} = 2$.

To reduce computation, the computed 4^n roll permutations are combined into combinations, giving each effective outcome a value $RollPermutations$.

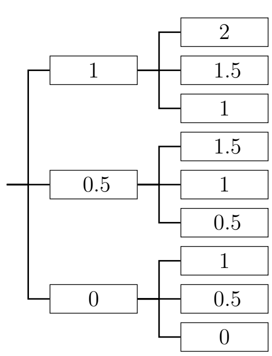
2.2 Roll Values

For each of these, the number of good outcomes is found based on the possible roll values.

For n rolls, there are $4^{a+1}4^{b+1}4^{c+1}4^{d+1}$ possible roll value combinations where $a+b+c+d = n$.

A method computes the roll value distribution of $x = a+1, \ldots, d+1$. It iterates x times, considering the 4 possible roll values that can extend the current computed values each time. To avoid computing a full 4^x outcomes, after each stage, paths with the same value are combined.

For example, assume the roll values are 0, 0.5, and 1. After the second iteration, we have:



Giving us $3^2 = 9$ outcomes to bring to the next iteration.

But instead, they can be grouped as so: 2 (1x), 1.5 (2x), 1 (3x), 0.5 (2x), 0 (1x). This leaves only 5 outcomes to bring to the next iteration. This is repeated for all of the x iterations.

The general method to compute the number of outcomes above the goal also uses this method, but for the 4 stat iterations.

This gives each outcome $RollValuePossibilities_i$ for $i = 1, \ldots, 4$.

2.3 Total Probability

The probability associated with a single outcome is now:

$$\frac{p \times RollPermutations \times \prod_{j=1}^{4} RollValuePossibilities_j}{\sum_{valid} p \times RollOutcomes \times InitialValueOutcomes \times RollValueOutcomes}$$

So, the total probability is:

$TotalProbability$

$$= \sum_{i=1}^{|combos|} \left(\frac{p \times RollPermutations \times \prod_{j=1}^{4} RollValuePossibilities_j}{\sum_{valid} p \times RollOutcomes \times InitialValueOutcomes \times RollValueOutcomes} \right)$$

$$= \frac{\sum_{i=1}^{|combos|} p_i \times RollPermutations_i \times \prod_{j=1}^{4} RollValuePossibilities_{i,j}}{\sum_{valid} p \times RollOutcomes \times InitialValueOutcomes \times RollValueOutcomes}$$

$$= \frac{\sum_{i=1}^{|combos|} p_i \times RollPermutations_i \times \prod_{j=1}^{4} RollValuePossibilities_{i,j}}{\sum_{valid} p \times (4^n)^2(4^4)}$$

2.4 Generating Outcomes

An overview of the code used to generate all outcomes is thus:

```
Sum = 0
foreach (s_1, s_2, s_3, s_4 → p) of GenerateCombos do
  foreach (r_1, r_2, r_3, r_4 → RollPermutations) of RollCountsDist do
    States = Map(0 → 1)
    for i = 1 ... 4 do
      NextStates = Map()
      foreach (State → Ways) of States do
        foreach (statSum_i → RollValuePossibilities_i) of ValueOfRollsDist(r_i + 1) do
          NextState = State + statSum_i × weight_i
          NextStates[NextState] += Ways × RollValuePossibilities_i
        end
      end
      States = NextStates
    end
    foreach (State → Ways) of States do
      if State is within goal then
        Sum += p × RollPermutations × Ways
      end
    end
  end
end
return Sum ÷ (∑_valid p × (4^4)(4^n)^2))
```

Note that at the end, $Ways$ is just the product of the $RollValuePossibilities_i$, $i = 1, \ldots, 4$, for the outcome.

The distribution viewer provides a way to visualize $RollCountsDist$ and $ValueOfRollsDist$.

2.5 Varying Initial Stat Count

Since artifacts have a chance to have 4 or 5 rolls (corresponding to 3 or 4 initially enabled stats), the actual probability is,

$$TotalProbability_{n=4} \times (1 - P(n = 5)) + TotalProbability_{n=5} \times P(n = 5)$$

So, the above pseudo-code is called twice for each calculation.