Collatz Hypothesis and Random Increases

Idea: Mikhail Ivanov
Developer: Mikhail Ivanov

There were plenty of ways to solve the problem. One series of solutions is as follows: fix two parameters $w \in \mathbb{Z}_{>0}$ and $k \in [0; 1]$. Let us try to apply the Collatz function to the current number w times. If 1 is reached within these steps, then we apply the Collatz function. If at most kw times we applied the Collatz function to an odd number, then we also apply the Collatz function (intuitively, the less we have to apply the Collatz function to an odd number, the less we multiply it by three and the more we divide it by two, thus on average we better decrease the number shown on the screen). Otherwise, apply the random function.

One reason why this solution might work poorly is that it may sometimes increase an even number. With a probability of 50% it turns odd, and the opportunity to halve it is missed. It's better to first halve the number, and only then to apply the random function. To address this issue, let us combine that (w, k)-strategy with $(1, k), \ldots, (w-1, k)$-strategies: that is, if at least for one positive integer $v \leq w$ in the nearest v iterations of the Collatz function we increase the number at most kw times, then we had better apply Collatz. This updated strategy is already accepted for $w = 11$, $k = 0.37$. To get a pair of well-performing parameters, one might brute-force through random pairs of parameters and check the performance of each one on many random inputs.

Another strategy is to try to construct an (almost) optimal solution. Let us choose a bound N and create a float array dp$[1..N]$ denoting the mathematical expectation of the score of the optimal strategy starting with each number. Initialize it with the score reached by only applying the Collatz function. Then let us traverse the array s times, each time updating each value $dp[i]$ with the minimum of the score achieved by applying Collatz (and continuing optimally), or with the average over $a[3i+1..6i]$, depending on what's smaller. To calculate the average over $a[3i+1..6i]$, we can make use of the running sum (a.k.a. window sum) algorithm. This algorithm is accepted for $N = 10^7$, $s = 19$ (and certainly for many other pairs).