Isaac Abboudi

ArithmeticPuzzle Write-up

March 1 2023

I used a lot of Hashmaps and sets to map the character to digits. I used Math.random() to generate random numbers. I defined an int elite to be the cap of selected solutions, I found that half was a good benchmark. Fitness was defined by finding the absolute value of the difference between (augend + addend) - sum for the given mapping. The fitness scores were scaled based on the totalFitness of the entire population.

To mutate, I iterated over the population, and if Math.random() was less than the mutationProbability then I picked a random char to map to a random digit. That new mapping was saved as an independent child of its ‘parent’ that it's based on.

For crossover I iterated over the set of elite mappings and for each pair, generated a child mapping based on a random crossover point.

I found that a selection threshold in the 25-35% range worked the best. I didn't find much difference between the tournament and roulette selection types. I also found that as crossover rates increased it did not improve performance, but as mutation increased performance went up a bit. I think that has to do with the fact that mutation affected the ‘non-elite’ mappings, whereas the crossover wasn’t adding much significant improvement within a group of stronger solutions.