Colin Conn

Final Project Proposal

Evolutionary Computing

3/31/2023

For my project, I would like to use and evolutionary algorithm to breed a Pokémon with 6 perfect IVs (“individual values”). In the Pokémon games, each Pokémon is given an IV in each of it’s 6 main statistics (hp, attack, defense, special attack, special defense, and speed). When breeding two Pokémon together, these statistics are passed down from the parents of the original parents. The way that this passing down works is if one of the parents holds a specific item, 5 of these IVs will be randomly chosen from the parents to be passed down to the child, with the fifth being randomly generated.

The goal of this project is showing that EC is applicable in a (possibly new) problem domain. From what I was able to find on the internet there is no EC projects related to the Pokémon series, and I think that a game series that is as popular and math-heavy has a lot of potential for the topic.

My plan for the project is to use an Evolution Strategy like from homework 2. I will represent each Pokémon as an array of six numbers ranging from 0 to 31. For recombination I will randomly choose 5 of the stats from each parent and randomly determine the fifth. The breeding system in the original series doesn’t have a method for mutation (unless randomly generating the sixth stat counts). I think that it makes sense to stick to the source material, so

I'm not planning on doing any traditional mutation. I believe that the sixth stat being randomly generated will be a good source of avoiding premature convergence. For survivor selection I will do a series of tournaments, the winners of which will survive to the next generation.

To test my approach, I want to use mean best fitness. Usually when you are breeding Pokémon you are only worried about getting one child with satisfactory IVs (for use in competitions). For my fitness measure I check how close each value is to 31 and average those out. However, usually a player only cares about either attack or special attack being at a maximum, not both. I would like to consider this in my fitness function as well.