8) adding exploitations factors measures to check the selection pressure:

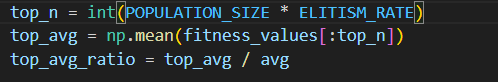
1. Fitness variance measure this function measure the fitness variance for each generation



We simply check the variance for the fitness sample for each generation to check convergence (high selection pressure) or diversity in solutions, low variance imply the first and high one imply the second

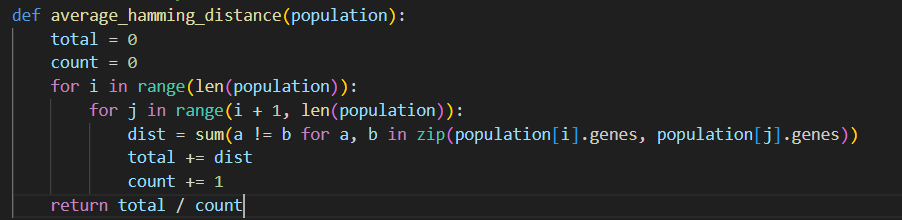
1. Top average selection ratio

Added this measure to check how much the algorithm favor to select the top individuals(solutions) of the population



9) Adding explorations measure to check the genetic diversification of the algorithm, We implemented 3 measure functions :

1. the distance between the individuals(solutions) in each genenration (hamming distance)



For each solutions we count the number of different genomes at each position this measure function is computationally costly because it compare each solution with all the other solutions which take O(n^2), higher hamming distance indicate higher search space (more exploration).

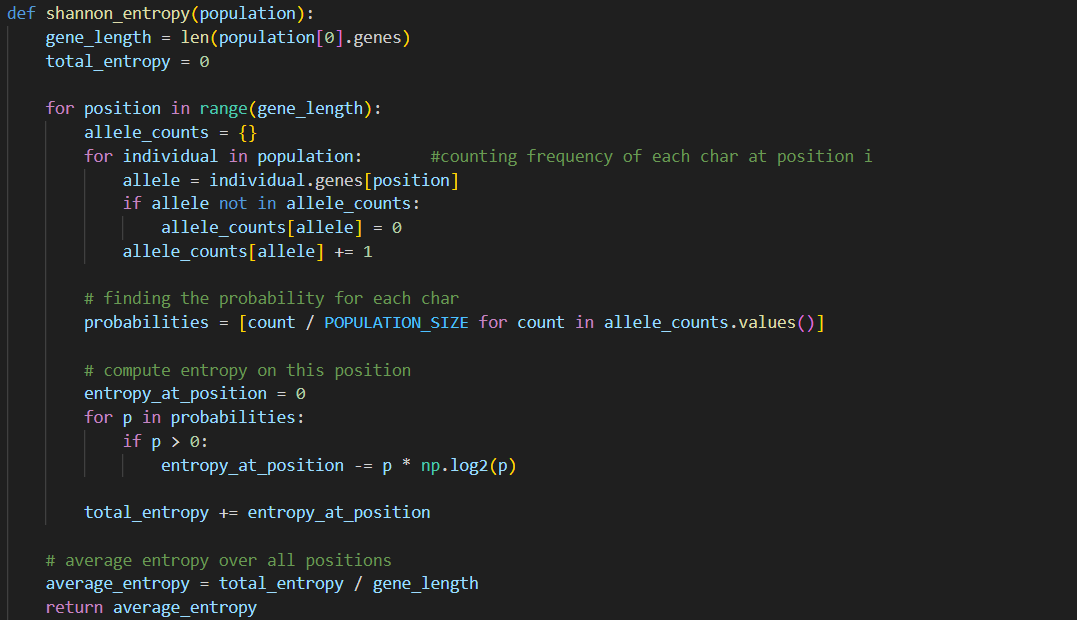
1. Number of unique chromosomes on index i across the population

A computer code with blue and yellow text

AI-generated content may be incorrect.

This measurement method check variation of chromosomes on each position to check for diversity in each position in the gene

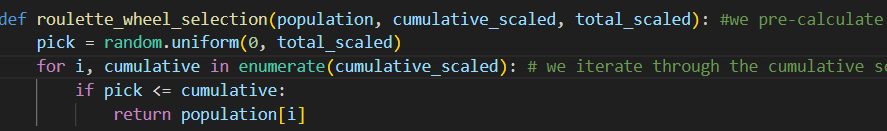
1. Shannon Entropy measure



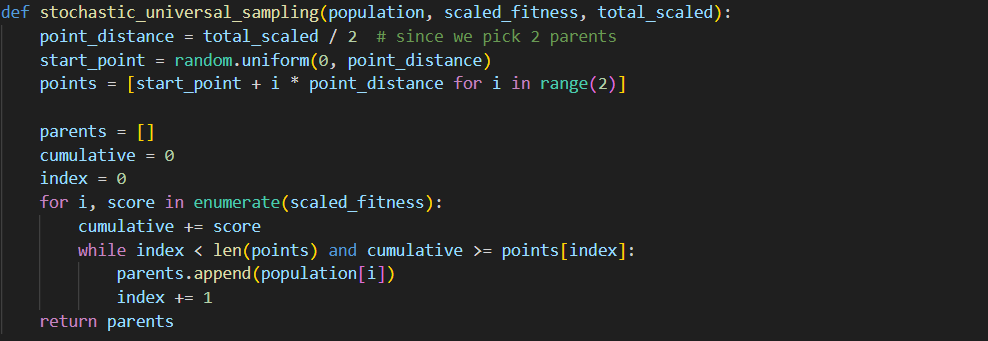
This function measure the amount of uncertainty for each position in the gene then give an average uncertainty outcome over all the gene positions higher entropy mean higher uncertainty which imply more random space (diversity)

10) Adding parent selection methods

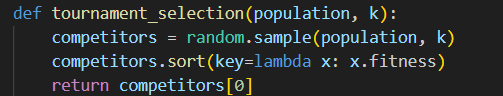
1. RWS + linear scaling: each individual take a portion of the populations space based on it’s fitness then we randomly pick one giving higher chance for more fit parents while maintaining chance for other less fitted parents for exploration

this method does have moderate selection pressure and preserve a fair diversity chance but it’s a little bit noisy because it could ignore good candidate or get stuck with lucky winners

1. SUS + linear scaling: similar to RWS but less randomness this time we choose 2 parents randomly with the more fit parents having more chance to be selected and return them

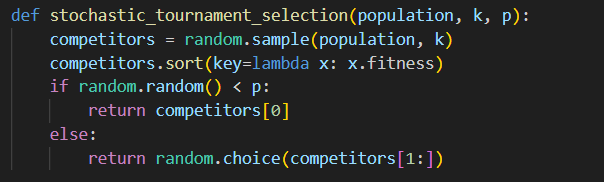


1. Tournament Selection(Deterministic) : in this parents selecting method we randomly choose a sample with size k from the population and return the best fit individual



This method does highly exploit selection pressure the weaker individuals will be eliminated fast focusing only on the best fit which will hinder diversity

1. Stochastic Tournament Selection: in this method we choose a sample K individuals and return the best fit out of them with a constant probability or else we return a random induvial out of the them



In this method we added a probabilistic constant to prevent high selection pressure and ensure diversity and fairness which will lead to better results than deterministic specially at escaping local optimums.