|  |  |
| --- | --- |
| Yomna Taher Abdallah | 20190624 |
| Adel Abdel-monem Arfa | **20190280** |

**Task 1**

1. **Initialization:**

* **A population of chromosomes is randomly generated. Each chromosome represents a solution or candidate in the search space.**
* **The size of the population is determined by the**

**)pop\_size( parameter.**

* **The length of each chromosome is determined by the )chrom\_len( parameter.**
* **The values of the chromosomes are randomly chosen from the )sample\_space( range.**

1. **Fitness Evaluation:**

* **The fitness function is defined to evaluate the performance of each chromosome in the population.**
* **The )fitness\_functio( calculates the fitness score for each chromosome by running the Tetris environment (env) with the given chromosome.**
* **The fitness score represents the quality or suitability of a chromosome as a solution.**
* **The fitness scores for all chromosomes in the population are stored.**

1. **Selection and Crossover:**

* **The chromosomes in the population are ranked based on their fitness scores.**
* **The top-ranking chromosomes, representing the fittest individuals, are selected for crossover to produce new offspring.**
* **The crossover rate (crossover\_rate) determines the proportion of top-ranking chromosomes used for crossover.**
* **The selected chromosomes undergo crossover, which involves exchanging genetic information between pairs of chromosomes to create new combinations.**
* **In the provided code, the crossover operation swaps the genetic material of the first half of one chromosome with the first half of another chromosome.**

1. **Mutation:**

* **Mutation introduces random changes in the genetic material of the chromosomes to explore new areas of the search space.**
* **The mutation rate (mutation\_rate) determines the probability of a gene in a chromosome being mutated.**
* **The mutation process in the code involves randomly selecting genes in the chromosomes and adding a random value from a range (-2 to 2) to those genes.**

1. **Next Generation:**

* **The new offspring generated through crossover and mutation, along with some of the top-ranking chromosomes from the previous generation, form the next generation population.**
* **The previous generation chromosomes not selected for crossover are discarded.**
* **The new population undergoes fitness evaluation, and the process of selection, crossover, and mutation is repeated for a specified number of generations.**

1. **Termination:**

* **The process continues for a specified number of generations (num\_generations) or until a termination condition is met.**
* **In the provided code, the termination condition is the completion of a certain number of mutation rounds (num\_mutation), where a new mutation is applied to the population.**
* **The final population is the result of the evolutionary process.**

**In summary, the algorithm starts with an initial population of randomly generated chromosomes. It repeatedly evaluates the fitness of the chromosomes, selects the fittest ones for crossover, applies mutation to introduce variability, and generates the next generation. The process continues for multiple generations, allowing the population to evolve and potentially converge towards better solutions.**

**Features :**

**(agg\_height / n\_holes / bumpiness / row\_transitions / score / num\_pits / n\_cols\_with\_holes / cleared)**

**The two best chromo :**

1. **array([-65.74792661, -34.75471346, -27.49264635, -34.75471346, -78.82680045, -65.77022113, -35.54476263, -9.71014655])**
2. **array([-66.65968282, -34.75471346, -27.49264635, -34.75471346, -76.13964912, -78.40617174,**

**-78.58290884, -64.79227041])**

**seed :72**

**(Train):**

**score for first chromo: 138400**

**score for second chromo: 138400**

**(Test):**

**Score for first chromo: 157600**

**Score for second chromo: 157600**