Cairo University
Faculty of Computers and Artificial Intelligence

Computational Cognitive Science

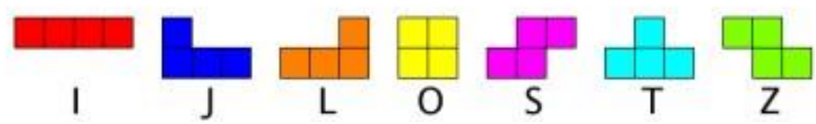
Tetris game report

Name:	ID:
1. Seif Eldin Mahmoud Helmy	**20210169**
2. Amira Bassam Dabbak	**20210799**
3. Adham Adel Mohamed	**20210052**
4. Adham Hany Ahmed	**20210056**
5. Muhannad Ibrahim Yassin	**20210411**
6. Ziad Ahmed Mohamed	**20210143**

Tetris game with genetic algorithm

HOW TO PLAY TETRIS?

Tetris, the iconic falling blocks video game, has captivated players for decades with its simplicity and addictiveness. Its mechanics lend themselves well to computerized gameplay, with each game state easily encapsulated by parameters such as the current score, the upcoming shape, and the occupied spaces on the 20 by 10 grid. Within Tetris, both human and computer players have a finite set of possible moves to consider at any given state. This paper explores Tetris as an optimization problem, presenting a range of optimization techniques applicable to devising efficient Tetris-playing algorithms. Among these techniques is the genetic algorithm, which mimics the process of evolution. The framework outlined herein details the application of a genetic algorithm to Tetris optimization, accompanied by an analysis of the outcomes obtained through its implementation.



- Figure 1. illustrates the move sets for the seven types of Tetris shapes. -

HOW DO GENETIC ALGORITHIMS?

In genetic algorithms, the aim is to identify inputs within a given input space that optimize the output of a specific function. This function, known as the fitness function, assesses the suitability of each candidate input. The algorithm maintains a population of N candidates, usually ranging from hundreds to thousands. At the start of each iteration, the current population represents a generation of candidates. This generation is then used to produce a new population of N candidates for the subsequent generation. Each generation is numbered according to the number of algorithm iterations conducted. The initial generation, Generation 0, comprises N randomly selected inputs from the input space. Subsequent generations are generated by probabilistically selecting candidates from the current generation, with the selection probability determined by their fitness as evaluated by the fitness function.

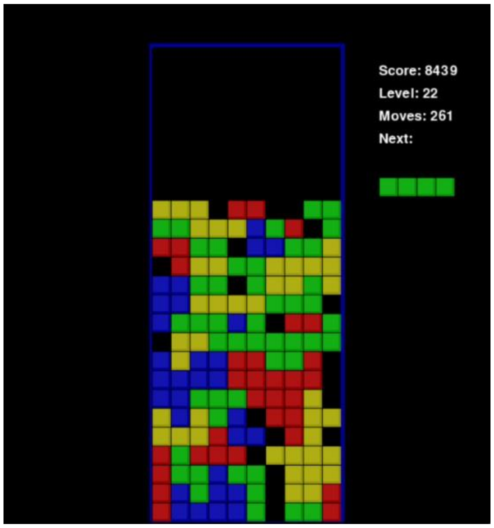
After evaluating all candidates, candidate selection proceeds in three stages, each contributing a certain percentage of candidates to the next generation:

- Selection: Candidates are chosen and transferred to the next generation without alteration.
- Mutation: Candidates are selected, and slight modifications are applied to their inputs before they are included in the next generation.
- Crossover: Candidate pairs are chosen, and their corresponding inputs are combined in a certain manner to create new candidates for inclusion in the next generation.

TETRIS IMPLEMENTATION

utilized for this scenario employs a game board with dimensions of 20 by 10. Players have visibility of both the upcoming shape for the next turn and the current one in play. Simplifying the gameplay, a player's action is limited to rotating the shape and selecting the column on the board from which it will descend, excluding additional movements typical in Tetris. This restriction corresponds to the move set sizes depicted in Figure 1. Notably, this implementation excludes the gradual descent of shape down the board, assuming rapid decision-making by a computer player. It's worth mentioning that this assumption overlooks the potential challenge posed by instant gravity in certain Tetris versions, rendering some moves unfeasible. Moreover, upon clearing at least one line following a move, the game score increases by specific increments: 40 points for clearing one line, 120 points for two lines, 300 points for three lines, and 1200 points for a 'Tetris' i.e., clearing four lines simultaneously.

based on Original BPS scoring system on: https://tetris.wiki/Scoring



- Figure 2. provides a sample depiction of a Tetris state for visualization. -

HOW GENETIC ALGORITHM OPTMIZE TETRIS?

In Tetris optimization, the player's ability can indeed be evaluated based on two key factors:

- Survival Ability: This relates to the player's capability to make many moves without losing. It involves strategic placement of shapes to prevent the game board from filling up and reaching the top. Players with strong survival ability can prolong the game and make more moves, increasing their chances of achieving higher scores.
- Scoring Efficiency: This aspect assesses how efficiently the player can score points by clearing lines. Clearing multiple lines at once, preferably by forming tetrises (four lines cleared simultaneously), is highly desirable. Efficient players can maximize their score output by strategically positioning shapes to create opportunities for line clears.

To optimize performance in Tetris, players often need to balance these two factors. While prioritizing survival is crucial for staying in the game and making more moves, efficiently scoring points is equally important for achieving high scores. Therefore, a successful Tetris player typically demonstrates proficiency in both survival strategies and scoring techniques, aiming to strike a balance between making moves and maximizing score output.

FEATURE SET

the selection of the feature set plays a crucial role in developing a player capable of consistently achieving long game durations with high efficiency. Previous studies have proposed various ideas for potential features, which were amalgamated with novel concepts to form a list of 14 prospective features. However, not all features contribute positively to the player's efficiency, but we just used only 9 of them.

To prevent overfitting on weight vectors and shape sequences, slight modifications were made to the problem. The range of values for each weight was constrained to [-10, 10], after a lot of experiments we saw that optimal weights ranged from [-10, 10].

Additionally, evaluations were conducted over 10 experiments with distinct random seeds.

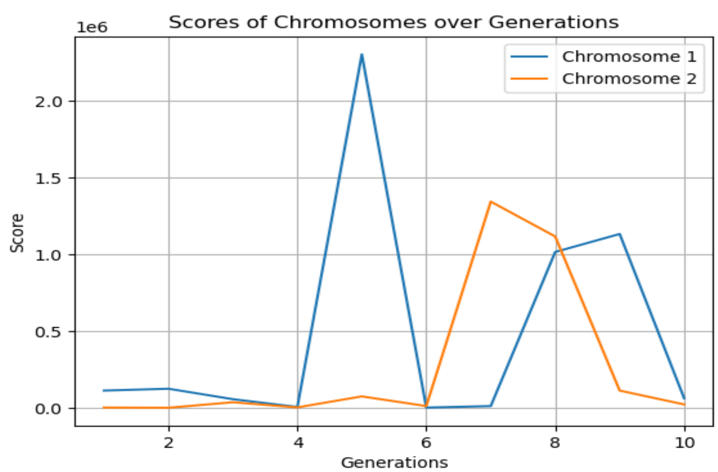
After many iterations and experiments we got our 9 best features:

The nine features as follows:

1. NumberOfRemovedLines - number of lines currently cleared during the game.
2. HolesNumber - number of holes on the board.
3. BlockingHolesNumber - number of holes which been blocked by blocks on the board.
4. PieceSides - number of sides of the current Tetris piece that are in contact with other pieces or the board.
5. FloorSides - number of sides of the Tetris piece that are in contact with the bottom floor.
6. WallSides - number of sides of the Tetris piece that are in contact with the walls.
7. ColumnsHasHoles – number of columns that has at least one hole.
8. bumpiness - sum of absolute height differences between the columns.
9. AggregateHeight – sum of all heights on the board.

TRAIN RESULTS

Following the computation of the optimal feature set with a size of 9, the genetic algorithm underwent 10 generations with a population size of 12 and 500 iterations. The candidate selection process involved three phases: Selection, Mutation, and Crossover, each contributing 50%, 20%, and 75% of the new candidates, respectively. This allocation ensured a balanced exploration of candidate solutions, with selection favoring promising individuals, mutation introducing diversity, and crossover facilitating the exchange of genetic information among candidates. We got our best two chromosomes which seen in Figure 3,



- Figure 3. provides the progress of the best two chromosomes in all generations. -

after 600 iterations we got the best weights:

[13.25994862, -5.27623236, -0.5, 7.69699829, 5.66449111, 5.18324878, -0.13187432, -3.9823942, 2.97403985]

Let’s see our best two chromosomes’ weights and scores.

```
Best Choromosom 1:  [13.25994862, -5.27623236, -0.5, 7.69699829, 5.66449111, 5.18324878, -0.13187432, -3.9823942, 2.97403985]
Score:  2304945
```

```
Best Choromosom 2:  [11.835122, -3.276212, 1.23179163, 5.66829, 3.61241, 3.12168, -4.212, -0.392, -0.2355]
Score:  1344635
```

TEST RESULTS

The best player of generation 5 was selected for additional testing. which the player plays 600 games of Tetris, we got the Final Best score, our winning criteria is to exceed 1 million:

```
Best Test Score 1421345
```

REFERENCES

Beating the world record in Tetris (GB) with genetics algorithm | by Duc Anh Bui | Towards Data Science

Building A Tetris Bot Part 2: Genetic Algorithms | by Alvin Lin | Medium

238_poster.pdf (stanford.edu)

Coding a Tetris AI using a Genetic Algorithm – Lucky's Notes (wordpress.com)