Introduction to Artificial Intelligence: Assignment 2.

Mashenkov Timofei B23-ISE-05
t.mashenkov@innopolis.university

December 1, 2024

1 Evolutionary Algorithm

1.1 Flow

1. *Initialization:* Generate an initial population of chromosomes.
2. *Selection:* Sort the population based on fitness and select the top-performing chromosomes.
3. *Crossover:* Combine the selected chromosomes to create new offspring.
4. *Mutation:* Introduce random changes in the offspring to maintain diversity.
5. *Replacement:* Replace the least fit chromosomes with the new offspring.
6. *Termination:* Repeat steps 2-5 until a stopping criterion is met (solution with 0 fitness value is found).

1.2 Algorithm Overview

The evolutionary algorithm (EA) consists of the following components, they have been tested for a long time, so the ranges of values that were used in testing are given:

- **Population Size:** 2000 chromosomes (sudoku tables).
- **Mutation Rate:** 3-90%.
- **Generations:** 500–2000 generations.
- **Stagnation Threshold:** 40–100 generations without improvement.

1.3 Initial Population

The initial population is generated by filling the undefined cells (hyphens) with random numbers in available (remain after removing predefined) range of numbers, ensuring that the predefined numbers remain unchanged.

1.4 Fitness Function

The fitness function evaluates the quality of a chromosome. Since numbers in rows are unique, the fitness is determined by counting the number of duplicates in columns and subgrids:

$$\text{Fitness} = \text{Number of Column Duplicates} + \text{Number of Subgrid Duplicates}$$

A perfect solution has a fitness value of 0.

1.5 Crossover Operation

Crossover combines two parent chromosomes by selecting the best elements from each parent to create a new offspring. The selection is based on the local fitness of each cell.

1.6 Mutation Operation

Mutation introduces random changes in the chromosome to maintain diversity in the population. It swaps two random cells in a row, provided they are not predefined.

1.7 Selection Process

The selection process involves sorting the population based on fitness and choosing the top-performing chromosomes for the next generation. The population is replenished with new chromosomes generated from the best solutions.

2 Experimental Setup

The algorithm was tested on Sudoku puzzles with varying numbers of givens (20 to 40). For each number of givens, at least 10 tests were conducted. The performance was measured in terms of average and maximum fitness values over the final generations.

3 Results

The algorithm successfully solved Sudoku puzzles across different complexity levels. The plots in References section show the average fitness, maximum fitness and average time for each complexity level.

- **Easy Difficulty:**
  - average time 3.6205454545454545
  - maximum fitness 55.45454545454545
  - average fitness 22.845454545454544
- **Medium Difficulty:**
  - average time 6.5760000000000005
  - maximum fitness 61.25
  - average fitness 25.15
- **Hard Difficulty:**
  - average time 6.489999999999999
  - maximum fitness 63.266666666666666
  - average fitness 26.283333333333335

4 Conclusion

The evolutionary algorithm effectively solves Sudoku puzzles by iteratively improving the population of candidate solutions. The results demonstrate the algorithm's capability to handle puzzles of varying difficulty levels. Future work could explore more sophisticated mutation and crossover operators to enhance convergence speed.

5 References

References

[1] https://sudoku2.com/sudoku-tips/how-sudoku-difficulty-is-measured/, 2023.

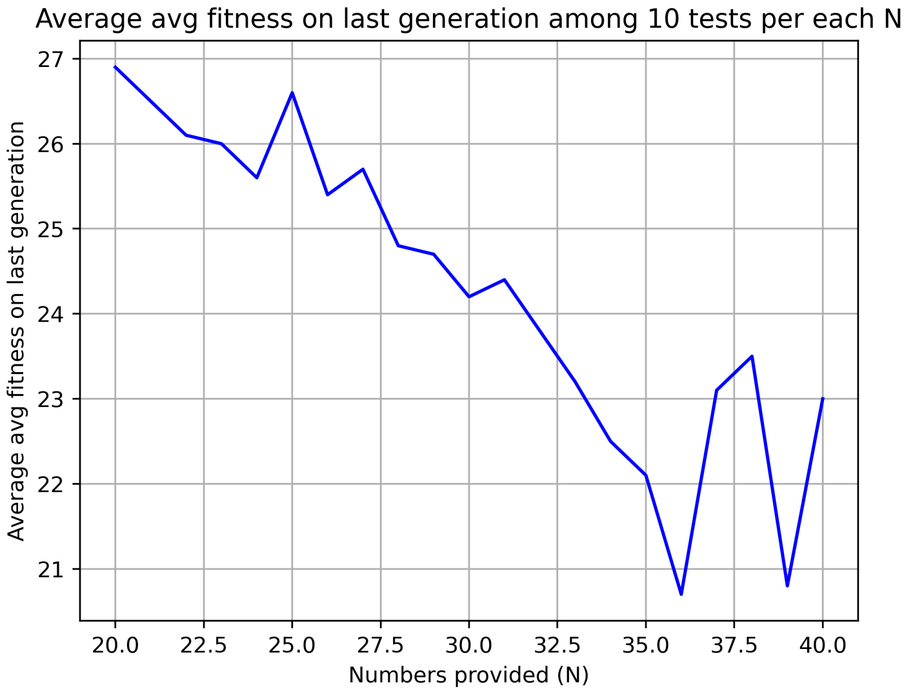

Figure 1: Average fitness.

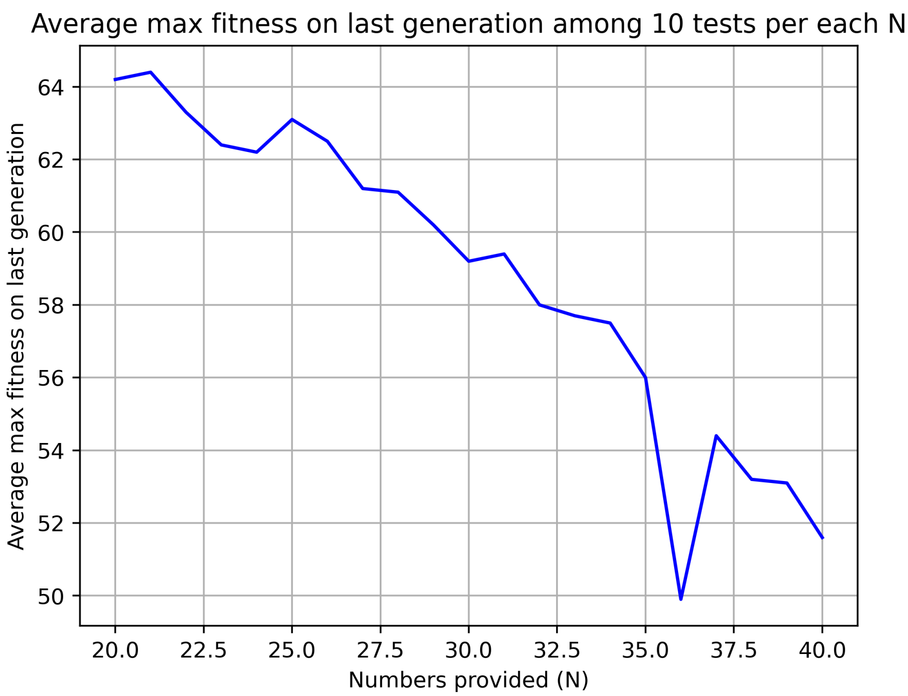

Figure 2: Maximum fitness.

Figure 3: Execution time.