

Intro to AI Assignment 2

Nikita Zagainov
DSAI-1

November 20, 2024

1 Implementation Details

The algorithm for solving Sudoku is implemented in `Python` and `C++` programming languages. The `Python` implementation is used for experiments and evaluation on test cases, while the `C++` implementation is used as submission on CodeForces.

For better availability of implementation details, I provide the source code on github: [Link](#)

2 Evolutionary Algorithm Description

2.1 Algorithm Flow

The EA algorithm follows a standard flow consisting of initialization, selection, crossover, mutation, and replacement. Each step will be described in detail below.

2.2 Fitness Function

The fitness function evaluates how close a given Sudoku solution is to being correct. It considers the number of conflicts in rows, columns, and subgrids.

2.3 Variation Operators

The variation operators include crossover and mutation. Crossover combines two parent solutions to produce offspring, while mutation introduces random changes to maintain diversity in the population.

2.4 EA Parameters

The EA parameters include population size, crossover rate, mutation rate, and the number of generations. These parameters are crucial for the algorithm's performance and will be tuned based on experimental results.

3 Experimental Setup

3.1 Test Cases

We evaluated the genetic algorithm on various Sudoku test cases with different numbers of givens and complexity levels. The test cases were divided into easy, medium, hard, and expert levels.

3.2 Evaluation Criteria

The evaluation criteria include the average and maximum population fitness at the final generations. We conducted at least 10 tests for each number of givens, ranging from 20 to 40. For complexity levels, we conducted at least 50 tests for each level.

4 Results and Discussion

4.1 Statistics

We present the statistics demonstrating the average and maximum population fitness at the final generations for different numbers of givens or complexity levels. The results are shown in the following plots.

4.2 Fitness Plot

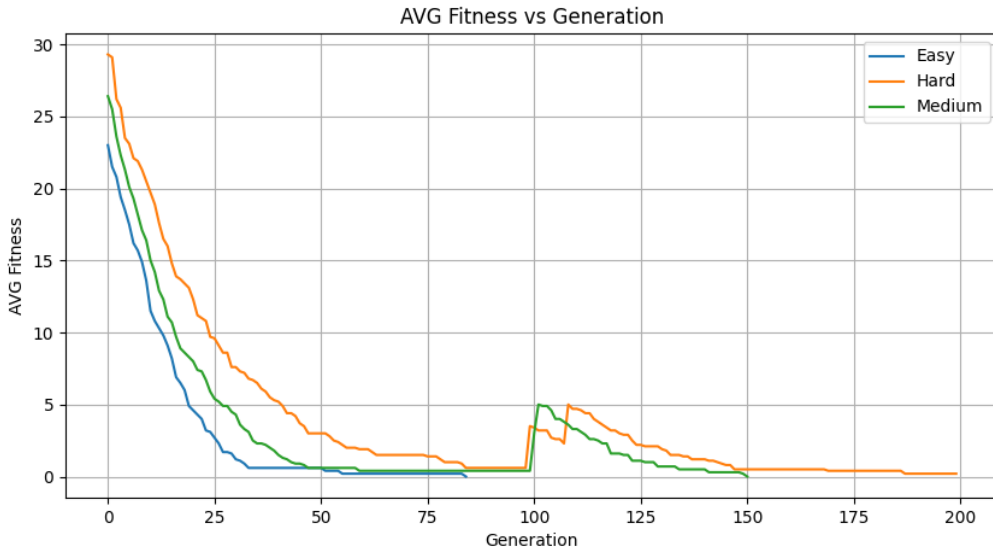


Figure 1: Average fitness value evaluated over 10 different Sudoku puzzles

5 Conclusion

In conclusion, we have described the chosen EA algorithm and its components in detail. We provided statistical analysis and plots demonstrating the algorithm's performance on

Sudoku puzzles with different numbers of givens and complexity levels. The results show that the algorithm performs better with more givens, as expected. Future work could involve further tuning of EA parameters and exploring other variation operators.

6 References

References

- [1] Author. *Title of the Classification Idea*. Journal/Conference, Year.