Elias Hussen

**Sudoku report**

[Subject]

School of XXX

Discipline and level of thesis

Degree programme

Vaasa 2024

|  |  |  |  |
| --- | --- | --- | --- |
| **UNIVERSITY OF VAASA**  **School of XXX** | | | |
| **Author:** | Elias Hussen | | |
| **Title of the thesis:** | Sudoku report : [Subject] | | |
| **Degree:** |  | | |
| **Discipline:** |  | | |
| **Supervisor:** |  | | |
| **Year:** | 2024 | **Pages:** |  |

**Contents**

[1 Intruduction 5](#_Toc167899678)

[1.1 Background on Sudoku 5](#_Toc167899679)

[1.2 Importance of Solving Sudoku 5](#_Toc167899680)

[1.3 Brief Overview of Evolutionary Algorithms 5](#_Toc167899681)

[1.4 Objectives of the Project 6](#_Toc167899682)

[2 Problem Definition 7](#_Toc167899683)

[2.1 Description of the Sudoku Problem 7](#_Toc167899684)

[2.2 Complexity and Challenges 7](#_Toc167899685)

[2.3 Specific Sudoku Problems Selected (Easy and Medium Puzzles) 8](#_Toc167899686)

[3 Methodology 9](#_Toc167899687)

[3.1 Overview of Evolutionary Algorithms 9](#_Toc167899688)

[3.2 Detailed Description of the Evolutionary Algorithm Used 9](#_Toc167899689)

[3.3 Implementation Steps in the Jupyter Notebook 11](#_Toc167899690)

[4 Results and Analysis 13](#_Toc167899691)

[4.1 Solution for Each Sudoku Puzzle with Visual Representations 13](#_Toc167899692)

[4.2 Generation Count and Time Taken for Each Puzzle 20](#_Toc167899693)

[4.3 Analysis of the Performance 20](#_Toc167899694)

[4.4 Comparison Between Easy and Medium Puzzles 20](#_Toc167899695)

[5 Conclusion 22](#_Toc167899696)

[5.1 Summary of Findings 22](#_Toc167899697)

[5.2 Recap of Performance 22](#_Toc167899698)

Tables

[**Table 1.** Summary of the generation count and time taken to solve each Sudoku puzzle. 20](#_Toc167899705)

# Intruduction

## Background on Sudoku

Sudoku is a popular puzzle game that originated in Japan in the late 20th century. It consists of a 9x9 grid divided into nine 3x3 subgrids, known as regions, boxes, or blocks. The objective of the game is to fill the grid with digits from 1 to 9, such that each column, each row, and each of the nine 3x3 subgrids contain all the digits from 1 to 9 (Pidd, 2009). Despite its simple rules, Sudoku puzzles can range from easy to extremely challenging, making it an engaging activity for people of all ages.

## Importance of Solving Sudoku

Solving Sudoku puzzles is not only a recreational activity but also has several cognitive benefits. It is known to enhance logical thinking, problem-solving skills, and mental agility. Regular engagement with Sudoku puzzles can improve concentration and memory retention, making it a useful exercise for maintaining cognitive health. Additionally, solving Sudoku has educational benefits, helping students develop critical thinking and numerical skills (Pidd, 2009).

## Brief Overview of Evolutionary Algorithms

Evolutionary algorithms (EAs) are a subset of artificial intelligence techniques inspired by the principles of natural selection and genetics. These algorithms operate by evolving a population of candidate solutions to a given problem through processes analogous to biological evolution, such as selection, crossover, and mutation (Goldberg, 1989). EAs are particularly well-suited for solving complex optimization problems where traditional methods may be infeasible or inefficient. They are widely used in various fields, including engineering, economics, and bioinformatics.

## Objectives of the Project

The primary objective of this project is to apply evolutionary algorithms to solve Sudoku puzzles. Specifically, the project aims to:

1. Implement an evolutionary algorithm to solve Sudoku puzzles of varying difficulty levels.
2. Evaluate the performance of the algorithm on both easy and medium Sudoku puzzles.
3. Analyze the effectiveness and efficiency of the evolutionary algorithm in finding valid solutions.
4. Identify any patterns or trends in the algorithm's performance and suggest potential improvements for future work.

By fulfilling these goals, the project hopes to show how evolutionary algorithms can be used effectively to solve combinatorial problems and advance knowledge of both their advantages and disadvantages in this setting.

# Problem Definition

## Description of the Sudoku Problem

Sudoku is a combinatorial number-placement puzzle where the objective is to fill a 9x9 grid so that each row, column, and the nine 3x3 subgrids contain all digits from 1 to 9 exactly once. The puzzle begins with some cells pre-filled with digits, which serve as clues for the player. The challenge is to complete the grid by filling in the remaining cells while adhering to the rules of the game. This requirement makes Sudoku a classic example of a constraint satisfaction problem (Lewis, 2007).

## Complexity and Challenges

Solving a Sudoku puzzle is categorized as an NP-complete problem, meaning that it requires a lot of computation and that no algorithm is currently available that can solve the problem for every instance in polynomial time. Sudoku puzzles are complicated because you have to meet several requirements at once. The arrangement of one digit influences the other digits that can be placed within the same row, column, and sub-grid, resulting in a highly connected problem space.

Traditional methods such as backtracking, which involves searching through all possible configurations, can solve simpler puzzles efficiently. However, these methods become less effective as the difficulty of the puzzles increases. The primary challenges in solving Sudoku include ensuring that all constraints are met without any conflicts and navigating the extensive search space efficiently. Additionally, achieving computational efficiency is critical, especially for more challenging puzzles where the solution space is vast and complex (Weber, 2017).

## Specific Sudoku Problems Selected (Easy and Medium Puzzles)

This project uses a set of Easy and Medium Sudoku puzzles with varying degrees of difficulty to assess the evolutionary algorithm's performance. There are three medium and three easy puzzles in the collection, each with varying levels of difficulty and restrictions.

Kuva, joka sisältää kohteen sanaristikko, teksti, neliö, Suorakaide

Kuvaus luotu automaattisesti

**Figure 1**. Selected Sudoku puzzles.

Simpler patterns and a greater quantity of pre-filled cells distinguish the easy Sudoku puzzles, which are easier to solve. The medium Sudoku puzzles, on the other hand, have fewer pre-filled cells and call for more sophisticated strategies to determine the right digit placements. This choice enables a thorough evaluation of the evolutionary algorithm's capacity to solve both easier and harder Sudoku puzzles.

# Methodology

## Overview of Evolutionary Algorithms

The optimization methods known as evolutionary algorithms (EAs) are founded on the ideas of genetic evolution and natural selection. These algorithms work especially well for solving intricate puzzles with a large search space, like Sudoku. The first step in the process is to initialize a population of potential solutions. Every candidate, or individual, is assessed according to a predetermined fitness function and represents a potential fix for the issue.

The fitness function evaluates how well a potential solution adheres to the limitations of the problem. The fitness function in Sudoku counts the number of distinct digits in each row, column, and subgrid; a higher score denotes fewer infractions of the game's rules. Then, based on their fitness scores, the selection process selects candidates for reproduction, giving preference to those who are more fit while allowing less fit individuals to contribute to the preservation of genetic diversity.  
  
Recombination, also known as crossover, joins together paired candidates to create offspring with features inherited from both parents. In order to explore new areas of the solution space, this step is essential. In order to prevent the population from becoming overly similar and to investigate novel potential solutions, mutations introduce haphazard changes to the progeny. Through these iterative processes of evaluation, selection, crossover, and mutation, the population evolves over successive generations, gradually improving the quality of the solutions.

## Detailed Description of the Evolutionary Algorithm Used

For this project, a specific evolutionary algorithm was implemented to solve Sudoku puzzles. The algorithm operates on a population of candidate Sudoku solutions and iteratively improves them using the following components:

First, all rows, columns, and subgrids in the Sudoku grid are filled with random numbers to create the initial population, making sure that no Sudoku rules are broken right away. This gives the evolutionary process a variety of starting points.  
  
The number of uni-que digits in each row, column, and subgrid is counted by the fitness function to assess each potential solution. A perfect score is attained when every row, column, and subgrid contains every digit from 1 to 9 exactly once. A higher fitness score denotes fewer rule violations.

Candidates with the highest fitness scores are chosen for reproduction. Fitness-proportionate selection is the method employed, meaning that candidates with higher fitness scores stand a better chance of getting chosen. Better solutions are more likely to be passed down to the following generation as a result of this.

Pairs of parent solutions combine to form offspring during crossover. The columns of the parent grids are switched beyond a randomly chosen crossover point. This aids in blending the traits of the parents, which may produce better results.

Mutation introduces random changes to the offspring. For each empty cell in the grid, there is a probability that its value will be altered to a different digit. This helps maintain diversity in the population and allows the algorithm to explore new areas of the solution space.

## Implementation Steps in the Jupyter Notebook

The evolutionary algorithm is implemented in a structured manner in the Jupyter Notebook:   
The first set of potential answers is created by duplicating the provided Sudoku grid and adding random numbers to the empty spaces. By doing this, a variety of legitimate beginning points for the evolutionary process are provided, ensuring that there are no instantaneous violations of the Sudoku rules.

A fitness function that counts the number of unique digits in each row, column, and subgrid is used to evaluate each candidate solution. The number of distinct digits raises the fitness score, and a maximum score denotes a workable Sudoku solution:

Kuva, joka sisältää kohteen teksti, Fontti, kuvakaappaus

Kuvaus luotu automaattisesti

**Figure 2.** Fitness function for the Easy 1 Sudoku puzzle.

Candidates are selected for reproduction based on their fitness scores. The selection process ensures that fitter candidates have a higher probability of being chosen, but it also allows less fit individuals to contribute, maintaining genetic diversity.

Selected pairs of parent solutions undergo crossover, where a random point is chosen, and the columns of the parent grids are swapped beyond this point. This results in two new offspring grids that combine traits from both parents:

Kuva, joka sisältää kohteen teksti, Fontti, kuvakaappaus, algebra

Kuvaus luotu automaattisesti

**Figure 3.** Crossover function for the Easy 1 Sudoku puzzle.

Offspring are subjected to mutation, in which their values are randomly altered. To ensure that the final grid is still valid, random cells are chosen and their values are changed to other possible di-gits.   
  
The algorithm gradually increases the fitness of the solutions by evolving the population over several generations through these iterative steps. Until a working Sudoku solution is discovered or a predetermined number of generations is reached, the process is repeated. With the use of evolutionary algorithms, this approach makes it possible to solve the Sudoku puzzle effectively and efficiently by navigating the challenging search space.

# Results and Analysis

## Solution for Each Sudoku Puzzle with Visual Representations

The evolutionary algorithm was used to solve a number of Easy and Medium-level Sudoku puzzles with differing degrees of difficulty. The visual representations of the solved Sudoku puzzles are shown below, along with information on how long it took to solve each puzzle and how many generations were needed.

The algorithm solved Easy Sudoku 1 in 0 generations, taking 0.02 seconds. The visual representation of the solved puzzle is shown below:

Kuva, joka sisältää kohteen neliö, kuvio, Suorakaide, kuvakaappaus

Kuvaus luotu automaattisesti

**Figure 4.** Solved Easy 1 Sudoku puzzle.

The algorithm solved Easy Sudoku 2 in 229 generations, taking 13.08 seconds. The visual representation of the solved puzzle is shown below:

Kuva, joka sisältää kohteen neliö, kuvio, kuvakaappaus, Suorakaide

Kuvaus luotu automaattisesti

**Figure 5.** Solved Easy 2 Sudoku puzzle.

The algorithm solved Easy Sudoku 3 in 69 generations, taking 255.22 seconds. The visual representation of the solved puzzle is shown below:

Kuva, joka sisältää kohteen kuvakaappaus, neliö, kuvio, Suorakaide

Kuvaus luotu automaattisesti

**Figure 6.** Solved Easy 3 Sudoku puzzle.

The algorithm solved Medium Sudoku 1 in 741 generations, taking 33 minutes. The visual representation of the solved puzzle is shown below:

Kuva, joka sisältää kohteen kuvakaappaus, neliö, kuvio, Suorakaide

Kuvaus luotu automaattisesti

**Figure 7.** Solved Medium 1 Sudoku puzzle.

The algorithm solved Medium Sudoku 2 in 123 generations, taking 416.18 seconds. The visual representation of the solved puzzle is shown below:

Kuva, joka sisältää kohteen neliö, kuvakaappaus, kuvio, Suorakaide

Kuvaus luotu automaattisesti

**Figure 8.** Solved Medium 2 Sudoku puzzle.

The algorithm solved Medium Sudoku 3 in 346 generations, taking 960.55 seconds. The visual representation of the solved puzzle is shown below:

Kuva, joka sisältää kohteen neliö, kuvio, kuvakaappaus, Suorakaide

Kuvaus luotu automaattisesti

**Figure 9.** Solved Medium 3 Sudoku puzzle.

## Generation Count and Time Taken for Each Puzzle

The table below summarizes the generation count and time taken to solve each Sudoku puzzle:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | **Puzzle** | **Generations** | **Time (sec or min)** | **Population Size** | **Max Generations** | | --- | --- | --- | --- | --- | | Easy Sudoku 1 | 0 | 0.02sec | 100 | 1000 | | Easy Sudoku 2 | 229 | 13.08sec | 500 | 15000 | | Easy Sudoku 3 | 69 | 4min 15sec | 25000 | 50000 | | Medium Sudoku 1 | 741 | 33 min | 25000 | 50000 | | Medium Sudoku 2 | 123 | 6min 56sec | 25000 | 50000 | | Medium Sudoku 3 | 346 | 16 min | 25000 | 50000 |   **Table 1.** Summary of the generation count and time taken to solve each Sudoku puzzle. |

## Analysis of the Performance

The evolutionary algorithm performed differently in each of the puzzles. The nearly instantaneous solution of Easy Sudoku 1 shows that the algorithm is capable of handling extremely basic puzzles. Easy Sudoku 2 and Easy Sudoku 3 demonstrated the variation even within the easy category, requiring a notably greater number of generations and time.   
Due to its higher complexity, Medium Sudoku 1 took the longest to solve out of all the medium puzzles—33 minutes. The solution times for Medium Sudoku 2 and Medium Sudoku 3 were more moderate, suggesting that the algorithm's performance is affected by both the initial configuration and the intrinsic difficulty of the puzzle.

Also another thing we can notice is that when population size was raised to 25000, it took more time to solve the puzzles by the algorithm:

**Figure 10.** Algorithm’s computation time analysis.

## Comparison Between Easy and Medium Puzzles

It is evident from comparing the performance results of medium and easy puzzles that solving medium puzzles typically took more generations and time. Nevertheless, the variation in performance did not follow a strict linear relationship with the predetermined level of difficulty of the puzzle. The efficiency of the algorithm seems to be greatly influenced by the particular limitations and starting circumstances of every puzzle.

Overall, the evolutionary algorithm proved that it could efficiently solve Sudoku puzzles that ranged from easy to medium. The wide range of generation counts and solution times emphasizes how crucial it is to take into account the unique features of each puzzle when assessing the algorithm's performance. Subsequent efforts may concentrate on refining the algorithm to enhance uniformity and decrease solving durations for increasingly intricate riddles.

# Conclusion

## Summary of Findings

This project investigated the use of evolutionary algorithms to solve Sudoku puzzles, showcasing the algorithm's effectiveness in solving puzzles with different degrees of difficulty. The robustness and adaptability of the algorithm were demonstrated when it successfully solved every selected Sudoku puzzle, which is based on the principles of natural selection.

The outcomes showed a discernible difference in performance between the various puzzles. Easy Sudoku 1, for example, solved almost instantly, demonstrating the effectiveness of the algorithm in simpler configurations. On the other hand, Easy Sudoku 3 took a lot longer and needed more generations, indicating that the algorithm's performance can differ greatly even within the same difficulty level.

Compared to easy Sudoku puzzles, medium Sudoku puzzles typically presented a higher challenge, requiring more strategies and longer solution times. Specifically, Medium Sudoku 1 took the longest, which is consistent with its higher level of complexity. This variability highlights how the algorithm's efficiency is affected by the initial puzzle configuration and constraints.   
Even though the algorithm worked well, there was still potential for improvement, especially when it came to solving more difficult puzzles. The significant amount of time needed to solve some medium puzzles suggests that more optimization is necessary to improve the algorithm's performance.

## Recap of Performance

We thoroughly tested the evolutionary algorithm on three medium and three easy Sudoku puzzles. The performance differed significantly between these puzzles. Easy Sudoku 1 was solved in 0.02 seconds, demonstrating the algorithm's effectiveness in handling simple puzzles. Easy Sudoku 3, on the other hand, took 69 generations and 255.22 seconds to solve, demonstrating the algorithm's flexibility while also pointing out areas where performance could be improved.  
  
The Medium Sudoku puzzles proved to be more difficult; solving Medium Sudoku 1 took 33 minutes. This is indicative of the medium puzzles' increased complexity and quantity of constraints. While Medium Sudoku 2 and Medium Sudoku 3 required 123 and 346 generations, respectively, it is clear that the algorithm is robust, but efficiency improvements are still needed.

Overall, the evolutionary algorithm performed well in solving Sudoku puzzles; however, the complexity and initial configuration of each puzzle had an impact on the algorithm's performance. The disparity in generation times and solution times highlights the necessity of additional optimization to improve consistency and efficiency across a range of puzzle difficulties. This project successfully illustrated how evolutionary algorithms can be used to solve combinatorial puzzles like Sudoku, offering insightful information about their advantages and possible areas for improvement.

# References

Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley.

Lewis, R. (2007). Metaheuristics can solve Sudoku puzzles. Journal of Heuristics, 13(4), 387-401.

Pidd, M. (2009). Tools for Thinking: Modelling in Management Science. John Wiley & Sons.

Weber, J. M. (2017). A comparative analysis of metaheuristics for solving Sudoku puzzles. Computational Intelligence and Neuroscience, 2017.