**MASTER DEGREE PROGRAM IN DATA SCIENCE AND ADVANCED ANALYTICS**

**Computational Intelligence for Optimization**

**Project Report**

**Sudoku Solver**

[[Github Link](https://github.com/Dioguini97/CIFO_GroupU)]



May, 2022

Group U

Maiia Tagunova, m20210984

Guzel Bayazitova,m20210699

Diogo Alexandre Espada Almeida Pereira, m20210657

Bruna Barbosa Vieira de Almeida Duarte, m20210669

# 1. INTRODUCTION

Sudoku is a logic-based number-placement puzzle, in which a player needs to fill and complete the grid using the numbers from 1 to 9. Each puzzle contains predefined numbers on the grid, regardless of the difficulty, so that the player can use the clues to find the missing digits in each row and column.

There are 3 essential rules that need to be followed:

* Each row and column must contain without any repetition numbers from 1 to 9. The placement of the digits is irrelevant
* A regular 9 x 9 grid is divided into 9 smaller blocks of 3 x 3, which are called nonets. The numbers from 1 to 9 can only occur one per nonet, which creates an additional restriction for the number positioning
* The sum of every row, column and block must equal 45, this condition helps to find a missing value or establish if any duplicates are present.

Each Sudoku puzzle can have only one solution if the player correctly follows the rules. Trying to guess the solution is not permitted, as it is a logical game.

# 2. SELECTION METHODS

Text here.

# 2.1. FPS

Text here.

# 2.2. Ranking

Text here.

# 2.3. Tournament

Text here.

***Figure 1:***

# 3. FITNESS FUNCTION

Text here.

# 4. MUTATION OPERATORS

Text here.

# 5. CROSSOVER OPERATORS

Text here.

# 6. MAXIMIZATION AND MINIMIZATION?

# Text here.

# 7. STATISTICAL VALIDATION

Text here.

# 8. CONCLUSION

Text here.

# 9. DIVISION OF LABOR

Everyone did everything.

## 

Why did you choose the representation you did?

How did you design the fitness function? Did you try using different fitness functions, to see the impact on your GA?

Which configurations worked best together? How many did you try, and how did you determine the "best" one?

Do different operators affect the convergence of your GA?

Have you implemented elitism? Does the inclusion or exclusion of elitism impact your GA?

Are your implementations abstract and work with both minimization and maximization?

Do you get good results for the project you chose? What could be improved?

## 

# 