|  |  |  |  |
| --- | --- | --- | --- |
| Name | ID | Level | Department |
| احمد ايمن محمد عبدالغفار | 20210032 | 3 | Cs |
| ابانوب ابراهيم صلاح | 20210002 | 3 | Cs |
| ابوبكر خالد ابوبكر | 20210011 | 3 | Cs |
| احمد ابراهيم عبدالله | 20210016 | 3 | Cs |
| بسملة حسن جاد | 20210223 | 3 | Cs |
| حبيبة احمد جلال | 20210274 | 3 | Cs |

Project Report: KenKen Puzzle Solver

Introduction and Overview:

Project Idea and Overview:

The project aims to solve KenKen puzzles, a mathematical puzzle that combines elements of Sudoku and arithmetic operations. It employs a genetic algorithm approach to find the solution to these puzzles.

Similar Applications:

<https://www.kenkenpuzzle.com/>

Literature Review:

<https://www.academia.edu/download/87608757/94.pdf>

<https://www.researchgate.net/profile/Alaa-Morsy-3/publication/364338952_A_BLACK_WIDOW_OPTIMIZATION_ALGORITHM_FOR_SOLVING_KENKEN_PROBLEM/links/634bb7dbff870c55ce29082c/A-BLACK-WIDOW-OPTIMIZATION-ALGORITHM-FOR-SOLVING-KENKEN-PROBLEM.pdf>

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1682239/>

<https://onlinelibrary.wiley.com/doi/abs/10.1002/ps.7917>

<https://dl.acm.org/doi/abs/10.1145/3017680.3017765>

Proposed Solution & Dataset:

Main Functionalities/Features:

Algo1:

The software generates KenKen puzzle solutions based on the input grid size and cage configurations. It utilizes a genetic algorithm to evolve a population of potential solutions to solve the puzzle.

Algo2:

solves kenken by backtrack

Applied Algorithms:

Genetic Algorithm:

Creation of an initial population of potential solutions, Evaluation of fitness based on uniqueness and adherence to cage constraints, Selection, crossover, and mutation to generate new solutions.

Backtrack algorithm:

This Python class, Kenken, is designed to solve Kenken puzzles using a backtracking algorithm. The algorithm enforces constraints on rows, columns, and cage operations to find a valid solution. The class is initialized with a Kenken puzzle defined in an input file, and the solve method is called to find a solution. The solution (if found) or an indication of unsolvability is then printed.

Experiments & Results:

Genetic:

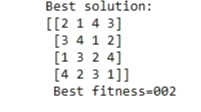
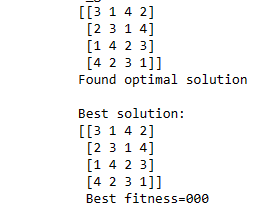
Experimentation and Testing:

# The algorithm underwent testing with various puzzle configurations and population sizes.

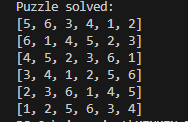
# Each generation's best fitness, the evolution of solutions, and convergence patterns were observed.

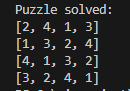
examples of output:

Sol1: Sol2:



Backtrack:





Discussion and Future Work:

Advantages/Disadvantages:

genetic:

# The genetic algorithm tackles KenKen puzzles.

# The approach may struggle with extremely large puzzle sizes or complex constraints.

# it falls at local optima

Backtrack:

# The code solves from 3 to 8 relatively fast.

# The approach may struggle with puzzles greater than 8

Behavior Analysis and Future Modifications:

genetic:

# The algorithm's behavior is influenced by population size, mutation rate, and crossover strategies.

# Future work may involve solving the puzzle without falling at local optima with fitness equals zero

Backtrack:

# Future work may involve changing algorithm to make it solve puzzles greater than 8 relatively fast

Github Repository Link :

https://github.com/Abanoub1111/Project-AI.git