KenKen Puzzle Solver Project Report

Name	Id
كريم أيمن محمد عبدالرحمن	20210681
شهد يحيي زكريا	20210471
محمود محمد سمير	20210877
يونان كميل راغب دانيال	20211118
أحمد حسن أحمد عماره	20220017
محمد عادل محمد محمود	202000778

Project Url :

https://github.com/KareemWebDeveloper/KenKen-Solver.git

1. Introduction and Overview

Project Idea and Overview

The goal of this project is to solve KenKen puzzles efficiently by implementing two problem-solving techniques:

1. **Backtracking Search Algorithm** - A systematic approach to explore possible grid solutions.
2. **Genetic Algorithm (GA)** - A heuristic-based evolutionary method that evolves solutions over multiple generations.

The **KenKen Puzzle Solver** provides users with an interactive interface to input puzzles and visualize solutions, enabling an analysis of algorithmic efficiency.

Applications and Similar Projects

Several applications address similar puzzle-solving problems:

- **KenKen Solver Apps**: https://tirl.org/software/kenken/
- **Logic Puzzle Solvers**: Tools like "Logic Grid Puzzle Solver" focus on combinatorial problems using constraint satisfaction algorithms.

These applications primarily aim to automate solving while enabling users to learn from algorithmic processes.

Literature Review

1. **"Solving and Modeling Ken-Ken Puzzle by Using Hybrid Genetics Algorithm"**
    a. This research explores a hybrid genetic algorithm approach to effectively solve KenKen puzzles, demonstrating the algorithm's efficiency in finding solutions.
    b. https://www.researchgate.net/publication/232647839_Solving_and_Modeling_Ken-Ken_Puzzle_by_Using_Hybrid_Genetics_Algorithm

2. **KenKen Puzzle Solver using Backtracking Algorithm**
    a. This paper presents an implementation of a backtracking algorithm to solve KenKen puzzles, detailing the algorithm's design and performance

b. https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2014-2015/Makalah2015/Makalah_IF221_Strategi_Algoritma_2015_016.pdf
3. **Solving Sudoku Puzzles with Genetic Algorithm**
   a. https://nidragedd.github.io/sudoku-genetics/

These references provide foundational concepts and optimization strategies implemented in this project.

2. Proposed Solution & Dataset

Main Functionalities and Features

1. **Puzzle Input**: Users can input a KenKen grid of any size.
2. **Algorithm Selection**: Users can choose between Backtracking and Genetic Algorithm.
3. **Adding Cages**: User Can add cages to their KenKen puzzle grid they want to be solved.
4. **Performance Metrics**: Evaluate efficiency based on solving time, correctness, and algorithm comparison.

Use-Case Diagram:

A use-case diagram can show how users interact with the KenKen solver, including selecting algorithms and viewing solutions.

Dataset

The project does not require an external dataset. Instead, users input custom KenKen puzzles for real-time solving. For testing, standard KenKen grids (4x4, 6x6) are utilized.

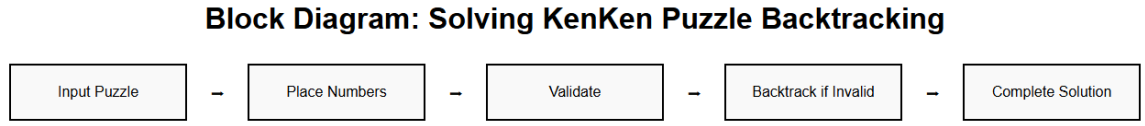
3. Applied Algorithms

Backtracking Search Algorithm

- Backtracking explores possible grid combinations by:
  - Placing numbers row-by-row.

- Checking constraints (sum, product, difference).
- Backtracking when a violation occurs.

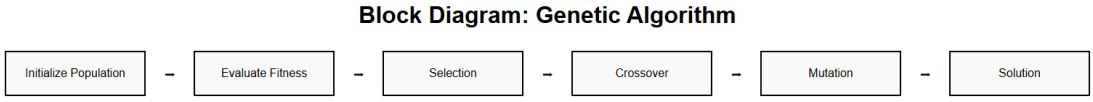
Block Diagram:



Genetic Algorithm

- GA operates by evolving potential solutions:
  - **Population Initialization**: Generate random solutions.
  - **Fitness Evaluation**: Measure correctness (number of satisfied constraints).
  - **Selection**: Choose the fittest individuals.
  - **Crossover & Mutation**: Combine and alter solutions.
  - **Repeat** until optimal/near-optimal solutions are found.

Block Diagram:



4. Experiments & Results

Experiments and Testing

1. **Backtracking Testing**: Tested on standard 4x4 and 6x6 KenKen puzzles.
2. **Genetic Algorithm Testing**: Measured efficiency and accuracy across multiple generations with varying population sizes.

Results

1. **Performance Metrics**:
   a. Backtracking: High correctness, but slower for larger grids.
   b. Genetic Algorithm: Faster for large grids but may yield near-optimal solutions.

5. Analysis, Discussion, and Future Work

Analysis of Results

- **Backtracking**: Works well for small grids with 100% correctness but scales poorly.
- **Genetic Algorithm**: Provides a trade-off between speed and accuracy, excelling for larger puzzles.

Advantages/Disadvantages

- **Backtracking**:
  - *Advantages*: Guaranteed solution.
  - *Disadvantages*: Computationally expensive.
- **Genetic Algorithm**:
  - *Advantages*: Faster for large-scale puzzles.
  - *Disadvantages*: May not find the optimal solution every time.

Future Work

- Optimize the Genetic Algorithm using advanced selection methods.
- Implement parallel processing to enhance backtracking performance.
- Extend the project to solve other combinatorial puzzles like Sudoku and Kakuro.

Conclusion

The KenKen Puzzle Solver successfully integrates **Backtracking** and **Genetic Algorithms** to address combinatorial challenges. The project highlights the strengths and trade-offs of these techniques, providing an interactive tool for both enthusiasts and researchers.