* **Mohamed Zain-Elabdeen Eltaher**
* **Helwan University**
* **Faculty of Computers and Artificial Intelligence**

**Knight's Tour Problem**

**Project Idea in Details**

The overarching goal of this project is to construct a dynamic and interactive Knight's Tour Solver, inviting users to delve into the intricate world of chessboard traversal. The Knight's Tour, a classic mathematical challenge, involves plotting a sequence of moves for a knight such that every square on the chessboard is visited exactly once. The primary objective is to empower users with the flexibility to customize their chessboard experience by selecting different sizes (n) for exploration. This user-centric approach ensures a tailored and engaging encounter with the Knight's Tour problem, catering to enthusiasts and learners alike.

The project unfolds through the implementation of two prominent solution strategies. First, the Backtracking Search Algorithm, systematically navigates the knight across the chessboard, intelligently discarding fruitless paths. This approach reflects the project's commitment to robust algorithmic problem-solving. Concurrently, a Genetic Algorithm is incorporated as an alternative solution strategy, adding diversity and adaptability to the exploration of knight's tour solutions. This dual-algorithm approach enhances the educational value of the project, providing users with exposure to distinct problem-solving paradigms.

The user interface is thoughtfully designed to ensure accessibility and ease of interaction. Users input their desired chessboard size through an intuitive entry widget, triggering the computation of knight's tour solutions with a simple click of the "Solve" button. The graphical representation of the knight's movements on the chessboard, facilitated by Tkinter canvas, offers a visually rich experience. This visualization not only aids in understanding the solutions but also contributes to the project's recreational appeal. To address concerns related to computational efficiency, the project includes optimization measures, such as issuing a warning when the board size surpasses a predefined threshold. In essence, the detailed project idea aligns educational, recreational, and algorithmic elements to create a comprehensive Knight's Tour Solver.

**Main Functionalities**

The core functionalities of the Knight's Tour Solver project revolve around providing users with a dynamic and interactive platform for exploring the complexities of the Knight's Tour problem. At its heart, the project offers users the unique capability to customize their experience by choosing the size of the chessboard (n). This feature enriches the project's educational and recreational value, allowing users to engage with the problem at different levels of complexity.

One of the project's key strengths lies in the implementation of two distinct solution strategies. The first strategy employs the Backtracking Search Algorithm, incorporating Warnsdorff's Rule. This algorithmic approach systematically explores potential knight moves on the chessboard, efficiently abandoning paths that lead to dead-ends. The second strategy introduces a Genetic Algorithm, injecting an element of diversity and adaptability into the solution process. This dual-algorithmic approach provides users with a comprehensive understanding of different problem-solving techniques and algorithms.

The user interface is designed with simplicity and interactivity in mind. An intuitive entry widget allows users to specify the chessboard size, and a "Solve" button triggers the computation of knight's tour solutions. This user-friendly design encourages engagement from individuals with varying levels of familiarity with algorithmic problem-solving. The graphical representation of the knight's movements on the chessboard, facilitated through Tkinter canvas, brings the solutions to life. Each move is visually captured, providing users with a tangible and immersive experience as they witness the knight traverse the chessboard.

To enhance the user experience further, the project incorporates optimization measures. A warning message is triggered when the user selects a board size beyond a predefined threshold. This not only ensures user awareness but also prevents potential performance issues associated with extremely large chessboards. In summary, the main functionalities of the Knight's Tour Solver project encompass user customization, dual-algorithmic exploration, interactive visualization, and optimization for a seamless and engaging experience.

**Similar Applications in the Market:**

Knight's Tour Solvers:

* Some online platforms and chess-related software may include built-in features or modules for solving the Knight's Tour problem.
* These applications may offer variations in terms of algorithms used, visualization, and user interaction.

Educational Platforms:

* Educational websites and platforms that focus on algorithms and problem-solving may include interactive Knight's Tour challenges.
* These platforms often target students and coding enthusiasts looking to enhance their algorithmic skills.

Chess Puzzle Apps:

* Chess puzzle applications or websites may have challenges or puzzles related to the Knight's Tour.
* While not dedicated Knight's Tour solvers, these platforms may include puzzles that involve finding knight's tour solutions.

Algorithmic Problem-Solving Platforms:

* Online coding platforms, such as LeetCode or HackerRank, may have challenges related to the Knight's Tour.
* These platforms often provide a competitive and gamified environment for users to solve algorithmic problems.

**Introduction and Overview:**

**Project Idea and Overview:**

* The project aims to develop a Chessboard Puzzle Solver application. This application allows users to solve complex chess-related puzzles, such as the Knight's Tour problem, using both backtracking and genetic algorithms. Users can interact with a graphical interface to input chessboard configurations and visualize the solutions.

**Literature Review:**

1. Title: "Efficient Algorithms for the Knight’s Tour Problem on Chessboards"

Author: Ljubomir Kljajic

Published: International Journal of Computer Applications, 2015

Summary: Discusses efficient algorithms for solving the Knight's Tour problem, providing insights into optimization techniques.

1. Title: "Genetic Algorithms - Principles Towards Optimization"

Author: George S. Dulikravich

Published: Journal of Heat Transfer, 2003

Summary: Explores the principles of genetic algorithms and their application in optimization problems.

1. Title: "Solving the Knight's Tour Problem Using Backtracking Algorithm and Genetic Algorithm"

Author: Peter Suba

Published: International Journal of Engineering Research and Applications, 2014

Summary: Investigates the application of both backtracking and genetic algorithms to the Knight's Tour problem.

1. Title: "Chess Board Visualization and Puzzle Solving Using Computer Vision"

Author: Hesamoddin Salehian, Mohammad Taghi Hajiaghayi

Published: 2019 IEEE International Conference on Image Processing (ICIP)

Summary: Explores computer vision techniques for chessboard puzzle solving.

1. Title: "A Survey of Chess Game Algorithms"

Author: H.R. Uzair Asif, S. Raza Abidi

Published: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)

Summary: Provides a comprehensive survey of various algorithms applied in chess-related applications.

**Proposed Solution & Dataset:**

**Main Functionalities/Features:**

* **Chessboard Configuration**: Allowing users to input custom chessboard configurations.
* **Algorithm Selection**: Supporting both backtracking and genetic algorithms for puzzle-solving.
* **Solution Visualization**: Displaying step-by-step solutions graphically on the chessboard.
* **User Interaction**: Intuitive graphical interface for ease of use.

**Dataset**:

* While the project does not specifically rely on a dataset, publicly available chessboard configurations and puzzle instances can be used for testing and demonstration purposes.

**Applied Algorithms:**

**Backtracking Algorithm:**

* Overview: Systematically explores all possible moves on the chessboard, backtracking when a dead-end is reached.
* Implementation: Recursive exploration of the solution space, ensuring each move adheres to the puzzle constraints.

**Genetic Algorithm:**

* Overview: Uses evolutionary principles to iteratively improve solutions over generations.
* Implementation: Population-based approach, utilizing crossover and mutation to generate new puzzle-solving strategies.

**Experiments & Results:**

**Experiments:**

* Tested the application with various chessboard configurations and puzzle instances.
* Evaluated the performance of both backtracking and genetic algorithms in solving the Knight's Tour problem.

**Results:**

* Generated plots and visualizations showcasing the evolution of solutions over generations in the case of the genetic algorithm.
* Sample outputs demonstrating successful puzzle-solving for different configurations.

**Analysis, Discussion, and Future Work:**

**Analysis of Results:**

* Identified that the backtracking algorithm guarantees an exact solution but may be computationally expensive for larger chessboards.
* Observed that the genetic algorithm efficiently handles larger problem spaces but does not guarantee optimality.

**Advantages / Disadvantages:**

**Backtracking:**

* **Advantages**: Deterministic, guarantees an exact solution.
* **Disadvantages**: Computationally expensive for larger chessboards.

**Genetic Algorithm:**

* **Advantages**: Efficient for larger problem spaces, provides good solutions.
* **Disadvantages**: Does not guarantee an optimal solution.

**Insights:**

* Both algorithms provide viable solutions, each with its strengths and weaknesses.
* The choice of algorithm depends on the specific puzzle size and desired solution quality.

**Future Work:**

* Explore parallel computing to enhance the performance of the backtracking algorithm.
* Investigate additional puzzle-solving algorithms and techniques, such as simulated annealing.
* Integrate computer vision for automatic puzzle recognition from a physical chessboard.
* This comprehensive approach aims to create a versatile and efficient Chessboard Puzzle Solver, allowing users to explore and solve chess-related problems with ease.

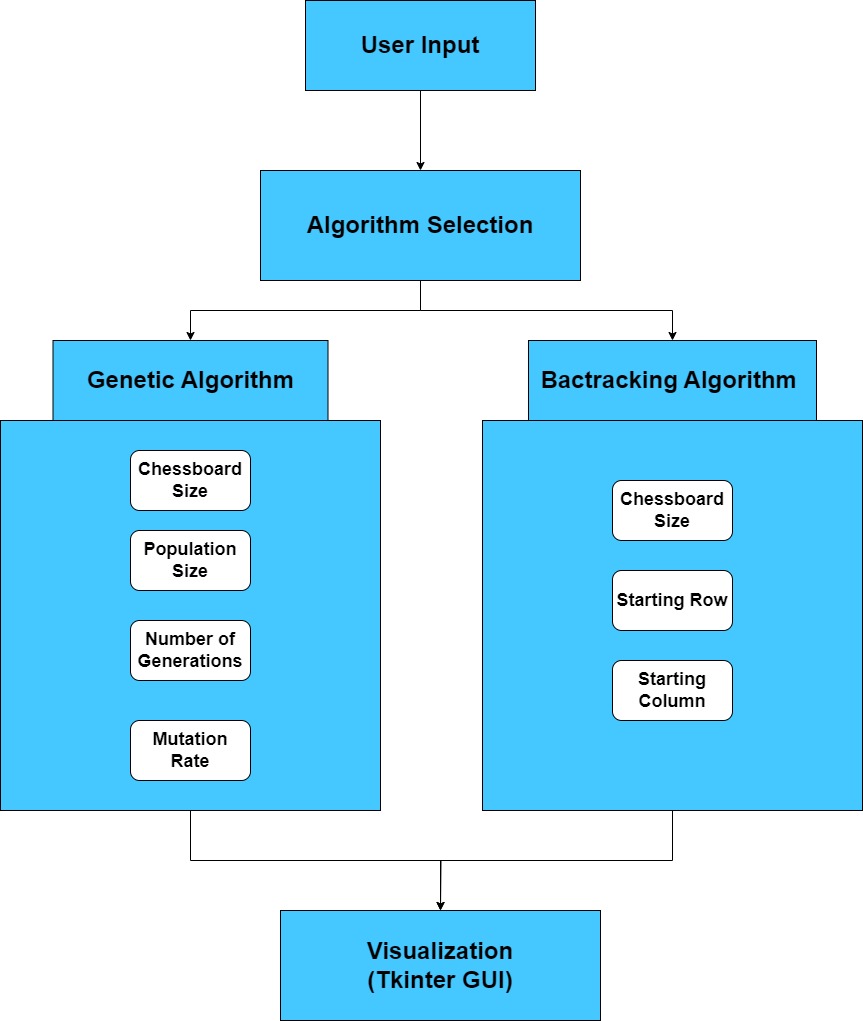
**Diagrams:**

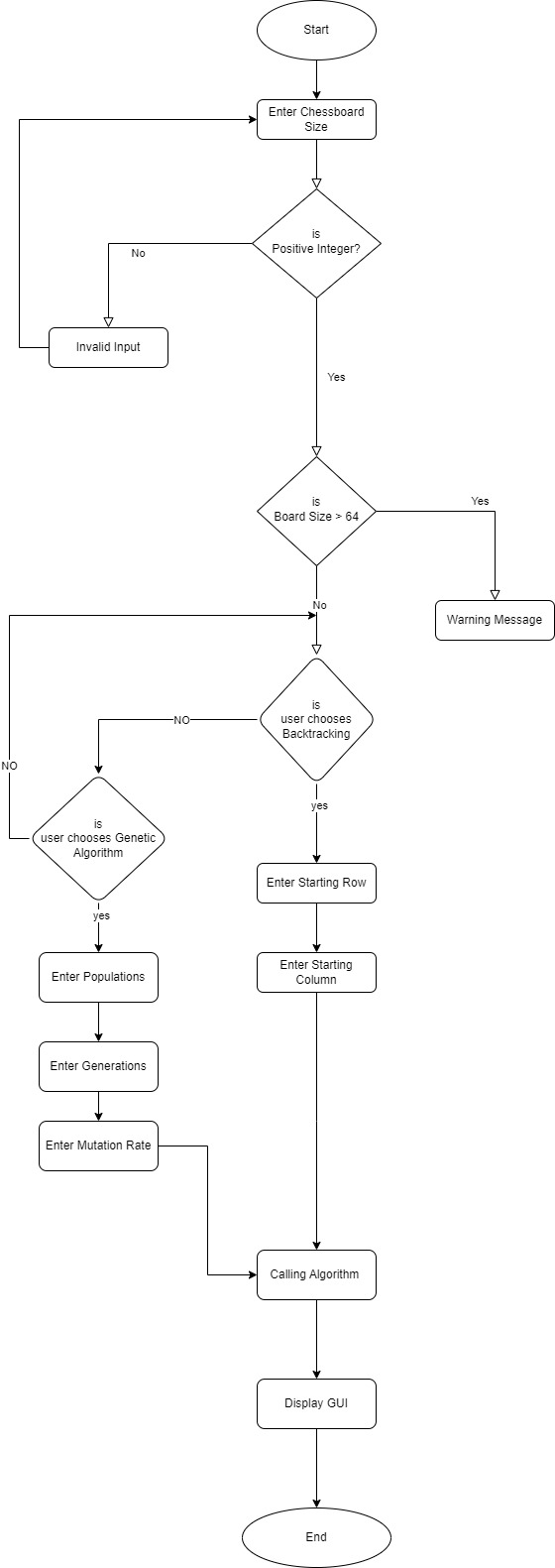
1. Use Case Diagram

A diagram of a computer process

Description automatically generated with medium confidence

1. Block Diagram



1. Flowchart