Smart Traffic Light Controller
AI-Based Intelligent Traffic Light Control System

1. Introduction

Traffic congestion is a major challenge in modern cities. Traditional traffic light systems rely on fixed timing strategies that fail to adapt to dynamic traffic conditions.

This project introduces an AI-based Smart Traffic Light Controller that formulates traffic signal control as a state-space search problem, utilizing classical AI search algorithms to optimize traffic flow in real time.

2. Objective

The objective is to develop a GUI-based smart traffic light controller that:

- Minimizes traffic congestion

- Dynamically adapts to traffic conditions

- Uses AI search algorithms for decision-making

3. Problem Definition

The system controls a four-way intersection (North, South, East, West).

At each discrete time step, vehicles arrive at the intersection and the controller must decide whether to HOLD the current signal phase or SWITCH to the opposite phase.

The primary goal is to minimize the cumulative waiting time of vehicles over a finite planning horizon.

4. State Representation

Each state is represented as:

(n, s, e, w, phase, time_in_phase)

Where n, s, e, and w represent the number of waiting vehicles in each direction.

5. Actions

- HOLD: Maintain the current green signal phase.
- SWITCH: Change to the opposite phase, subject to minimum green time constraints.

6. Constraints

A minimum green time constraint is enforced to prevent unrealistic rapid signal switching.

7. Transition Model

Two transition models are used:

- Planning Transition: A deterministic model used for search and planning.
- Simulation Transition: A stochastic model that simulates real-world traffic variability.

8. Cost Function

The cost of a state is defined as the total number of waiting vehicles:

cost = n + s + e + w

9. Heuristic Function

The heuristic function used in the A* algorithm is defined as:

heuristic = max(n, s, e, w)

This heuristic estimates congestion severity and guides the search efficiently.

10. Algorithms Used

- Breadth First Search (BFS)
- Depth First Search (DFS)
- Uniform Cost Search (UCS)
- A* Search

- Iterative Deepening Depth First Search (IDDFS)

11. Planning Horizon

A fixed planning horizon is used to limit search depth and ensure computational feasibility.

At each time step, the algorithm explores future states up to the defined horizon and executes only the first optimal action.

12. Performance Metrics

The system evaluates performance using the following metrics:

- Total waiting vehicles
- Average waiting vehicles
- Number of signal switches
- Average computation time per decision

13. GUI Features

- Real-time visualization of traffic flow and signal states
- Step-by-step execution of search algorithms
- Start, Pause, and Reset controls
- Adjustable simulation speed
- Display of performance metrics during execution
- Although explored nodes are not explicitly visualized graphically, their effect is reflected through algorithm behavior and terminal output

14. Sample Input

Example initial state:

(10, 8, 2, 1, "NS", 5)

15. Sample Output

- Real-time GUI visualization of the intersection

- Terminal output displaying performance statistics and algorithm decisions

16. Conclusion

This project demonstrates how classical AI search algorithms can be effectively applied to intelligent traffic control systems.

The integration of search-based decision-making with real-time visualization enhances both system performance and interpretability.

Preparation:

Yassin tamer

Ezzeldin Mohamed

Yassin Mohamed

Aly ahmed

Mohamed wael