Module 2: Search-Based Ranger and Drone Patrol Routing

This module presents the design and implementation of a search-based route optimization framework for wildlife protection patrols. The system computes optimal paths for a ranger or drone to travel from a fixed start node to an alert node within a grid-based environment. Environmental factors, thermal alerts, migration zones, and historical poaching densities dynamically influence edge costs. Two search algorithms—Uniform Cost Search (UCS) and A* Search—are used to compute and compare optimal patrol paths.

2.1 Grid-Based Environment Representation

The wildlife reserve is modeled as a 5×5 grid, resulting in 25 nodes, labeled N0 to N24. Each node corresponds to a physical zone with coordinates (r, c) representing row and column positions in the grid. This grid is fully connected in the four primary directions: up, down, left, and right. Diagonal movement is restricted because the patrol model follows a 4-direction movement structure (up, down, left, right), similar to navigating through discrete adjacent zones in a reserve. Under this movement model, Manhattan distance correctly represents the minimum steps required to reach the goal. Since the heuristic must reflect the underlying movement rules of the graph, allowing diagonals would change the grid geometry and invalidate the Manhattan metric. Therefore, the movement design and the heuristic remain aligned by disallowing diagonal transitions.

- Each node stores attributes such as: terrain type, poaching density, migration zone membership, high-risk assignment, and thermal activity level.
- Each edge represents a direct traversable path between neighboring nodes.
- All base edge weights start as 1 before environmental adjustments.

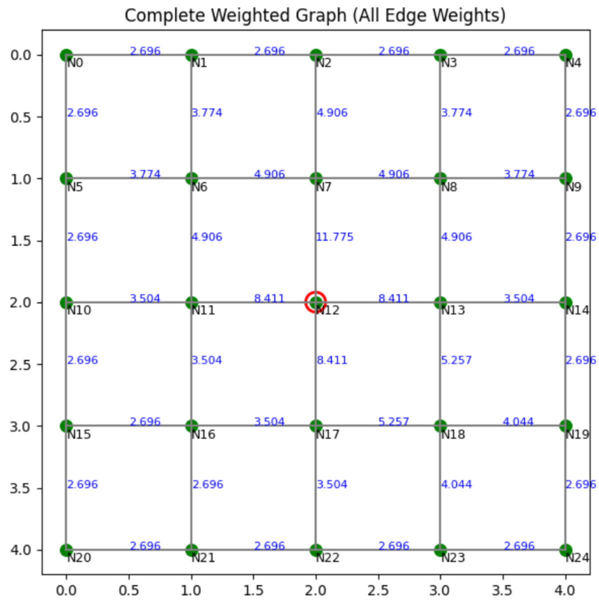

Figure 1: Grid-Based Patrol Graph with All Edge Weights and the alert node

2.2 Dynamic Edge Weight Model

Each edge weight $w(u, v)$ is computed using a multi-factor environmental model. This formulation allows the routing algorithm to evaluate the relative difficulty, risk, and traversal cost across different regions of the reserve. The computation occurs in two stages:

1. **Additive adjustment:** The base traversal cost is increased by the terrain penalty of the destination node.
2. **Multiplicative scaling:** Environmental, behavioural, and risk-based factors are then applied.

The adjusted base cost is:

$$\text{AdjustedBase} = \text{BaseCost} + P_{\text{terrain}}$$

The final edge weight is:

$$\begin{aligned} w(u, v) = \text{AdjustedBase} &\times F_{\text{visibility}} \times F_{\text{weather}} \times F_{\text{thermal}} \\ &\times F_{\text{season}} \times (1 + p_u)(1 + p_v) \\ &\times F_{\text{migration}} \times F_{\text{risk}} \times F_{\text{drone}}. \end{aligned}$$

Below is the breakdown of each contributing factor.

2.2.1 Terrain Penalty

Each terrain type imposes an **additive** penalty:

Terrain	Penalty
Flat	0
Rocky	2
Forest	3
River	4

In this implementation, all nodes use flat terrain ($P_{\text{terrain}} = 0$), but the system supports future extensions.

2.2.2 Visibility (Time of Day)

$$F_{\text{visibility}} = \begin{cases} 0.9 & \text{day,} \\ 1.3 & \text{night} \end{cases}$$

Night-time traversal is slower due to reduced visibility.

2.2.3 Weather Effects

$$F_{\text{weather}} = \begin{cases} 1.0 & \text{clear,} \\ 1.2 & \text{foggy,} \\ 1.4 & \text{rainy} \end{cases}$$

2.2.4 Thermal Propagation from Alert Node

Thermal intensity is based on user input. This work uses **Propagation Model B**:

- Alert node: High thermal level
- Adjacent neighbors: Medium thermal level
- All remaining nodes: Low thermal level

Multipliers:

$$F_{\text{thermal}} = \begin{cases} 1.0 & \text{low}, \\ 1.3 & \text{medium}, \\ 1.6 & \text{high} \end{cases}$$

2.2.5 Seasonal Influence

$$F_{\text{season}} = \begin{cases} 1.2 & \text{dry}, \\ 1.5 & \text{monsoon}, \\ 1.0 & \text{winter} \end{cases}$$

2.2.6 Poaching Density Factor

Poaching density influences traversal risk through a multiplicative factor:

$$F_{\text{poaching}} = (1 + p_u)(1 + p_v)$$

where p_u and p_v denote poaching densities at nodes u and v.

2.2.7 Migration Zones

Nodes N6, N7, and N8 are wildlife migration corridors and introduce a penalty:

$$F_{\text{migration}} = 1.4$$

2.2.8 Default High-Risk Zones

Historically dangerous nodes (N12 and N18) contribute:

$$F_{\text{risk}} = 1.5$$

2.2.9 Drone Advantage

In drone mode, heavy edges receive a cost reduction:

$$F_{\text{drone}} = \begin{cases} 0.7 & \text{if } w > 3, \\ 1.0 & \text{otherwise.} \end{cases}$$

2.3 Search Algorithms

We use two classical search algorithms: In uninformed techniques: Uniform Cost Search (UCS) and in Informed techniques: A* Search. Both guarantee an optimal path because all edge weights are positive and the heuristic is admissible.

2.3.1 Uniform Cost Search (UCS)

UCS explores nodes by increasing cumulative path cost. It guarantees optimality but may expand many nodes.

- Expands lowest-cost frontier node first.
- Inefficient when a heuristic can guide the search.
- Serves as the baseline for comparison.

2.3.2 A* Search

A* search extends UCS using a heuristic function. Its evaluation function is:

$$f(n) = g(n) + h(n)$$

where:

- $g(n)$ is the cost accumulated from the start node to n,
- $h(n)$ is the heuristic estimate of the cost from n to the goal.

For this module, the heuristic used is the **Manhattan distance**:

$$h(n) = |r_n - r_{\text{goal}}| + |c_n - c_{\text{goal}}|$$

because movement is allowed only in the four cardinal directions (up, down, left, right). This choice aligns exactly with the structure of the underlying grid.

Heuristic Admissibility and Consistency A heuristic is called **admissible** if it never overestimates the true cost to reach the goal:

$$h(n) \leq h^*(n) \quad \text{for all nodes } n$$

where $h^*(n)$ is the actual minimal remaining path cost from n.
In this grid, Manhattan distance is admissible because:

- Each move changes the row or column by exactly 1.
- No diagonal movement is allowed.
- The minimum number of moves required to reach the goal is exactly the Manhattan metric.

Thus, Manhattan distance *never* overestimates the true shortest path in a 4-direction grid.

The heuristic is also **consistent** (monotonic), meaning:

$$h(n) \leq c(n, n') + h(n')$$

for every neighbor n' of n.
This holds because:

- Moving from a node to any neighbor changes Manhattan distance by at most 1.
- Edge costs are always positive and they are always greater than 1.

Therefore, A* using Manhattan distance satisfies both admissibility and consistency. This ensures:

- A* will always find the optimal solution.
- The search never needs to revisit nodes.
- The number of expanded nodes is minimized compared to UCS.

Since both UCS and A* search the same weighted graph, they produce the **same optimal path and cost**. However, A* reaches the answer much faster due to its heuristic guidance.

2.4 Algorithm Comparison

Feature	UCS	A*
Heuristic	No	Yes (Manhattan)
Optimal	Yes	Yes
Speed	Slower	Faster
Nodes Expanded	High	Low
Final Path	Same	Same

A* explores far fewer nodes due to heuristic guidance, while UCS explores blindly. However, both yield the same optimal route and total cost.

2.5 Visualization and Interpretation

The final route visualization includes:

- Node labels (N0–N24)
- Green node markers
- Blue edge-weight labels
- Orange optimal path
- Red ring marking the alert node

Interpretation:

- Edges near the alert node increase in cost due to thermal propagation.
- Migration zone nodes exhibit higher traversal penalties.
- Seasonal and weather multipliers directly influence path choice.
- A* prioritizes nodes nearer to the target, improving efficiency.

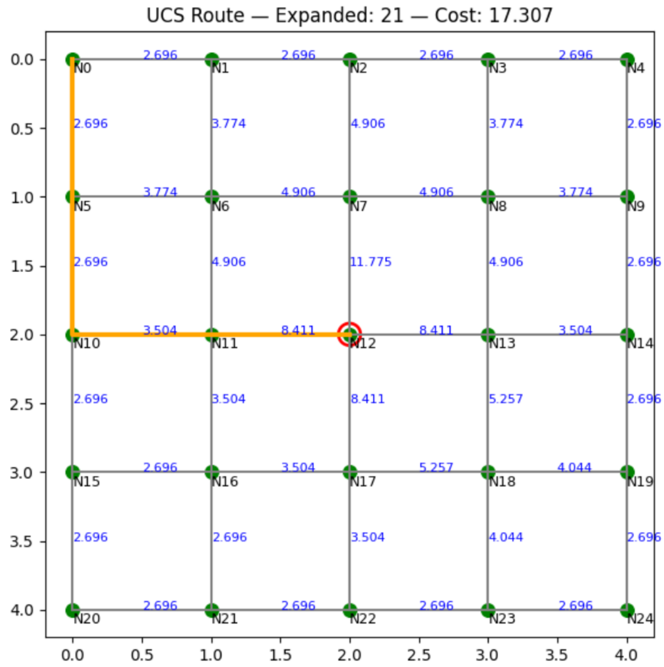

Figure 2: UCS Path, Total Cost, and Nodes Expanded

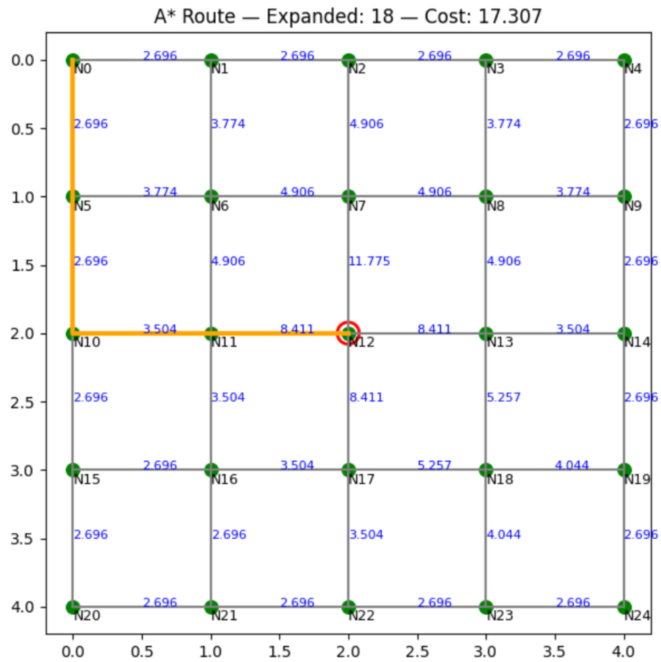

Figure 3: A* Path, Total Cost, and Nodes Expanded

2.6 Summary

This module successfully develops a complete system for intelligent patrol routing:

- Grid-based graph construction
- Multi-factor weighted edges reflecting real-world constraints
- Implementation of UCS and A* search algorithms
- Detailed comparison based on optimal cost and node expansions
- Visualization for interpretability

The model creates a realistic and extensible foundation for ranger and drone routing in wildlife conservation areas.

2.7 Individual Contributions

Since the system consists of three major components—(i) graph construction and unified weight modelling, (ii) Uniform Cost Search implementation, and (iii) A* search implementation—the work was divided equally between the two contributors so that each member completed tasks of comparable complexity and significance.

Lokesh Lingam 25CS06005:

- Collaboratively developed the 5×5 grid graph structure, Worked on designing the dynamic edge-weight model influenced by the risk factors.
- Implemented the Uniform Cost Search (UCS) algorithm and added node-expansion tracking for performance evaluation.
- Documented UCS behaviour and the visualization of the model.

Rahul Dewangan 25CS06008

- Collaboratively developed the 5×5 grid graph structure, Worked on designing the dynamic edge-weight model influenced by the risk factors.
- Implemented the A* search algorithm using Manhattan distance and verified heuristic admissibility.
- Documented the A* search approach and contributed to the comparative analysis with UCS.