A* Algorithm with Mandatory Stopovers

1 Algorithm Summary

The A* algorithm with mandatory stopovers extends the classical A* pathfinding algorithm to handle scenarios where specific intermediate locations must be visited before reaching the final destination.

1.1 Main Logical Flow

The algorithm operates in three main phases:

1. **Stopover Selection Phase**: Iteratively select the closest unvisited stopover using heuristic distance estimation
2. **Pathfinding Phase**: Apply standard A* algorithm to find optimal path from current location to selected stopover
3. **Final Phase**: Once all stopovers are visited, apply A* to reach the final goal

1.2 Key Components

- S = Set of mandatory stopovers
- V = Set of visited stopovers (initially empty)
- $h(a, b)$ = Heuristic function estimating distance from location a to location b
- $current\_goal$ = Dynamic target location for each iteration

1.3 Algorithm Structure

Algorithm 1 A* with Mandatory Stopovers

$V \leftarrow \emptyset$
$current\_location \leftarrow START$
while $V \neq S$ **do**
    $current\_goal \leftarrow \arg\min_{s \in S \setminus V} h(current\_location, s)$
    Run A*$(current\_location, current\_goal)$
    $current\_location \leftarrow current\_goal$
    $V \leftarrow V \cup \{current\_goal\}$
end while
Run A*$(current\_location, FINAL\_GOAL)$
Reconstruct complete path through all segments

1.4 Optimality Properties

- Each individual path segment is optimal (guaranteed by A*)
- Overall path optimality depends on stopover visiting order
- Greedy heuristic-based selection provides computational efficiency
- Trade-off between optimality and computational complexity