

# Robot Courier Delivery Puzzle

## Problem Domain:

I designed a Robot Courier Delivery Puzzle on a 2D grid. The environment contains obstacles, a robot start location, multiple packages located at distinct coordinates and a single drop off location. Packages must be delivered in a predefined order (package 0 then 1..). The robot can carry at most one at a time. Actions include moving one cell in the four cardinal directions (cost=1), picking up a package at 0 cost. This domain offers classic puzzles because of the ordered pick up/deliver constraint and the interplay between operations and navigation.

## Formal Definition

State:  $(r, c, \text{next\_index}, \text{carrying})$  where  $\text{next\_index} \in [0..n]$  is the index of the package that must be delivered next and  $\text{carrying}$  is a boolean indicating whether the robot currently holds that package. The start state is  $(r_{\text{start}}, c_{\text{start}}, 0, \text{False})$ . The goal is  $\text{next\_index} == n$  and  $\text{carrying} == \text{False}$ .

## Actions

- Move N/S/E/W: if the target cell is inside the grid and not an obstacle. Cost=1.
- Pickup: allowed when the robot is on the location of  $\text{packages}[\text{next\_index}]$  and not carrying. Cost=0.
- Deliver: allowed when robot is carrying and at the drop cell. Advances  $\text{next\_index}$ . Cost=0.
- Transition model: deterministic updates of position and flags based on actions.
- Step cost: movement costs 1; pickup/deliver cost 0. Total cost equals number of grid moves.

## Algorithms implemented:

1. Uniform-Cost Search(UCS) - an uninformed algorithm that expands nodes in order of path cost( $g$ ). Correct for variable-cost search and returns an optimal solution under our cost model.
2. A\* Search - informed algorithm using the following admissible heuristic:
  - If  $\text{carrying} == \text{True}$ :  $h(\text{state}) = \text{Manhattan}(\text{position}, \text{drop})$
  - If  $\text{carrying} == \text{False}$ :  $h(\text{state}) = \text{Manhattan}(\text{position}, \text{package}[\text{next\_index}]) + \text{Manhattan}(\text{package}[\text{next\_index}], \text{drop})$ . This is admissible because Manhattan distances are lower bound on the number of moves required on a 4-connected grid.

## Evaluation:

- I ran both algorithms on a 7×9 sample map with three packages in sequence.
- Typical measurements printed by the program: solution cost, path length, nodes expanded, max frontier size, runtime.

## Analysis and justification:

- Domain choice rationale: The ordered-delivery constraint produces a state space structure that makes the travel-and-task aspect non-trivial and educational to analyze with search algorithms.
- Algorithm choice rationale: A combination of UCS (uninformed) and A\* (informed) demonstrates performance differences and allows evaluation of admissibility and optimality.
- Heuristic justification: Heuristic is a sum of Manhattan distances relevant to the immediate tasks. It is admissible because movements are at least unit Manhattan steps; pickup/deliver are zero-cost operations and are accounted for.