دانشگاه شهید بهشتی
۱۳۳۸

Advanced Artificial Intelligence

Mobin Nesari

Student ID: 403422231

Problem 0

0.a Minimum cost to reach (m, n)

The step cost is

$$c((x, y), a) = 1 + \max(x, 0),$$

so the cost depends only on the current x-coordinate and not on y.

To minimize the total cost, all vertical moves should be taken while $x = 0$. Any move made after increasing x to a positive value incurs an additional penalty of $\max(x, 0) > 0$. In particular, a vertical move at $x > 0$ costs at least $1 + x > 1$, whereas the same move at $x = 0$ costs exactly 1.

Thus, an optimal path is to first move north from $(0, 0)$ to $(0, n)$, and then move east from $(0, n)$ to (m, n).

The cost of the n vertical moves is

$$n \cdot (1 + \max(0, 0)) = n.$$

For the horizontal moves, when moving from $x = i - 1$ to $x = i$ for $i = 1, \ldots, m$, the cost is

$$1 + \max(i - 1, 0) = i.$$

Therefore, the total horizontal cost is

$$\sum_{i=1}^{m} i = \frac{m(m+1)}{2}.$$

Hence, the minimum cost required to reach (m, n) is

$$C^*(m, n) = n + \frac{m(m+1)}{2}.$$

Uniqueness: If $m, n > 0$, this path is unique. Any path that performs a vertical move after increasing x above zero incurs strictly higher cost. If $m = 0$ or $n = 0$, the remaining straight-line path is trivially unique.

0.b True/False (Uniform Cost Search on the grid)

- **False.** Uniform Cost Search (UCS) can still terminate on an infinite state space as long as all step costs are nonnegative and the optimal goal cost C^* is finite. Here every action has cost ≥ 1, and (m, n) is reachable with finite cost, so UCS will eventually pop the goal and stop.

- **False.** UCS does not restrict exploration to the rectangle $0 \leq x \leq m$, $0 \leq y \leq n$. It expands states in increasing path-cost order, so it may also expand cheap states outside that rectangle (e.g., with $x < 0$ or $y < 0$) before reaching the goal.

- **True.** Let C^* be the optimal cost to the goal. UCS expands nodes in nondecreasing accumulated cost $g(s)$. Therefore, it will never *expand* any state with $g(s) > C^*$ before terminating (though such states might be generated and placed in the frontier).

0.c True/False (General weighted graphs)

- **True.** Adding edges only adds additional candidate paths while keeping all existing paths. Since shortest-path distance is a minimum over all paths, adding edges can only decrease distances or leave them unchanged, but cannot increase them.
- **False.** UCS is designed for graphs with nonnegative edge costs; negative edges can break its optimality/termination guarantees. Moreover, even if a very negative edge exists, it need not appear in an optimal $s \to t$ path unless it lies on some beneficial path from s to t.
- **False.** Adding a constant c to every edge cost changes a path cost from $C(P)$ to $C(P) + c \cdot |P|$, where $|P|$ is the number of edges in the path. Because different paths can have different numbers of edges, their relative ordering can change, so the optimal path is not necessarily preserved.

Problem 1

1.b Example of a useful / surprising route (UCS)

Start–tag pair. I used the Stanford map (`stanford.pbf`) with the start location at the landmark tagged `landmark=gates` (Gates) and the destination tag `landmark=oval` (The Oval).

Result and why it is desirable. Uniform Cost Search (UCS) found a direct, intuitive route that follows the main east–west corridor for most of the trip and then takes a short northbound segment into the Oval (Figure 1). This route is desirable because it: (i) avoids unnecessary detours, (ii) uses well-connected arterial walkways/roads, and (iii) produces a simple path that matches how a human would typically navigate between these two landmarks.

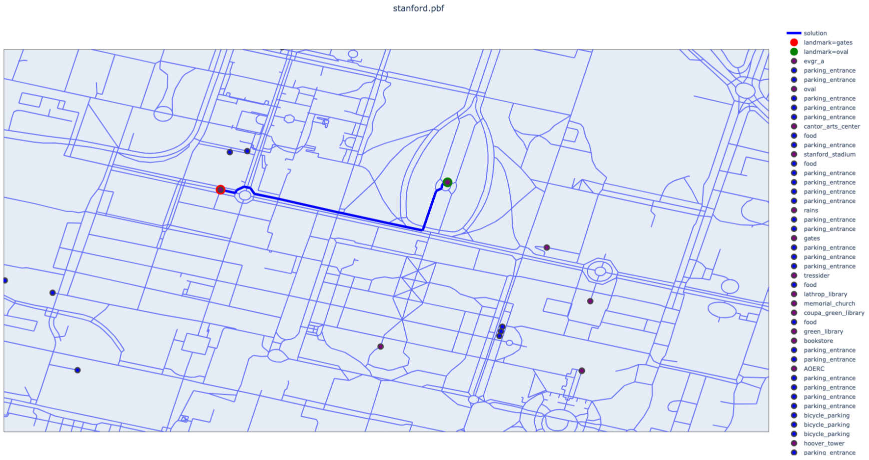

Figure 1: Desirable UCS route on the Stanford map: start at `landmark=gates` (red) and reach a node matching `landmark=oval` (green).

1.b Example of an undesirable route (UCS)

Start–tag pair. Using the Stanford map (`stanford.pbf`), I selected a start location tagged `landmark=oval` and a destination tagged `landmark=bookstore`. Uniform Cost Search was used with the given edge-cost model.

Result and why it is undesirable. Figure 2 shows two routes between the same start and destination. The blue path is the route returned by UCS, which is optimal under the given cost function. However, from a human perspective this route is undesirable: it contains many turns, follows a visually complex zig-zag pattern, and is harder to reason about or remember.

In contrast, the green path represents a more natural route that a human would likely prefer. It follows longer straight segments and major walkways, even though its total cost under the model is slightly higher. Humans often favor such routes because they are simpler, easier to follow, and cognitively less demanding.

Moreover, frequent turning introduces real-world costs that are not captured by the search problem formulation. Paths with many turns typically require additional braking and acceleration, which increases fuel consumption and wear, and may reduce comfort and safety. Since these factors are not included in the edge-cost model, UCS correctly optimizes the specified objective but produces a path that is misaligned with human preferences.

This example highlights a limitation of purely cost-based shortest-path planning: optimizing a simplified cost function can yield routes that are mathematically optimal but practically undesirable. Incorporating additional costs for turns, route complexity, or comfort could help align algorithmic solutions with human expectations.

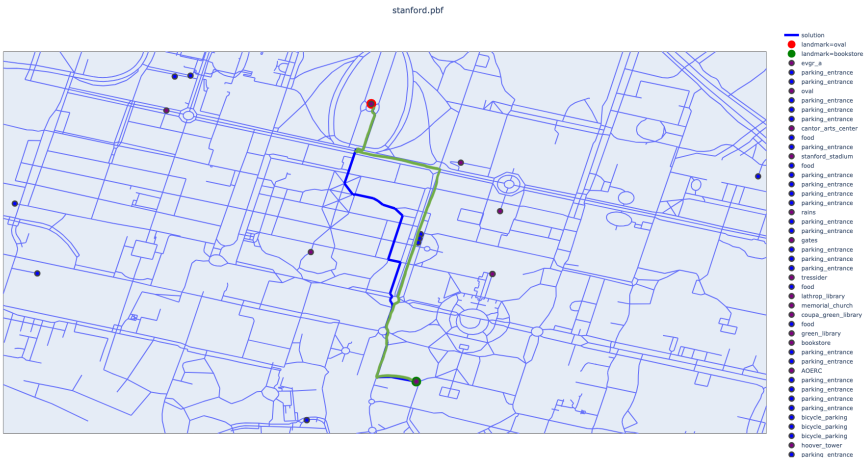

Figure 2: Undesirable UCS route (blue) compared with a more natural, human-preferred route (green) between `landmark=oval` and `landmark=bookstore`.

1.c Potential negative externalities and mitigation

If route-planning systems are widely adopted, they can create negative externalities because they optimize an individual user's travel objective (e.g., time or distance) without

fully accounting for system-level and social costs.
Impacts on users.

- *Over-reliance and reduced situational awareness:* frequent use can reduce users' ability to navigate independently and adapt when the system fails or is inaccurate.
- *Safety and accessibility mismatches:* an "optimal" route under the model may ignore factors such as lighting at night, steep terrain, unsafe crossings, or accessibility needs, increasing risk for some users.
- *Privacy risks:* accurate routing often relies on collecting fine-grained location traces, which can expose sensitive information about routines, home/work locations, and habits if stored or shared.

Impacts on non-users.

- *Traffic spillover and nuisance:* large-scale guidance can divert vehicles onto residential streets, increasing noise, pollution, and accident risk for pedestrians and cyclists.
- *Unequal distribution of burden:* optimization may repeatedly route through neighborhoods with fewer protections or enforcement, concentrating externalities on specific communities.
- *Interference with public services:* spillover congestion can slow buses, emergency vehicles, and local deliveries, degrading service quality for others.

Mitigation strategy. A practical mitigation is to incorporate *system-level constraints and penalties* into the cost function. For example, the planner can penalize or restrict cut-through routing on residential roads (especially near schools or during certain hours) and incorporate safety/accessibility features as additional costs. Another complementary approach is *load balancing* among near-optimal routes (e.g., randomizing within a small cost margin) to avoid funneling all users onto the same "shortcut." Finally, privacy harms can be reduced via *data minimization* (collect only what is necessary), aggregation, or on-device computation when possible.

Problem 2

2.b State-space size with unordered waypoints

Assume the map contains n distinct locations and k waypoint tags. A state in the unordered-waypoint shortest-path problem must encode:

- the agent's current location, and
- which subset of the k waypoint tags has already been visited.

For each of the n possible locations, there are 2^k possible subsets of visited waypoint tags. Therefore, the total number of distinct states is at most

$$n \cdot 2^k.$$

In the worst case, Uniform Cost Search may need to explore all such states before reaching the goal.

2.c Route visualization with two unordered waypoints

Figure 3 shows a route computed using Uniform Cost Search with two unordered waypoints. The start location is tagged `landmark=gates`, the final destination is `landmark=oval`, and the required waypoints are `landmark=memorial_church` and `landmark=bookstore`. The algorithm is free to visit the two waypoints in any order, as long as both are visited before reaching the final destination.

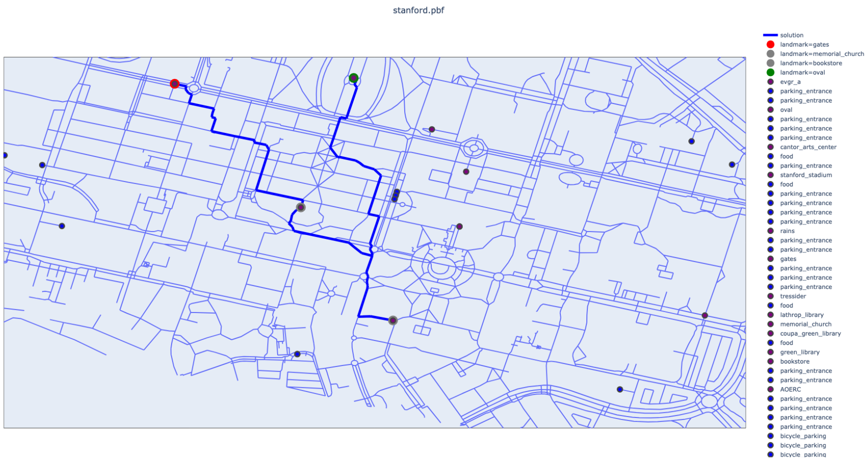

Figure 3: UCS route from `landmark=gates` to `landmark=oval` with unordered waypoints `memorial_church` and `bookstore`.

Observed route behavior. The generated path visits the waypoints in an order determined purely by the accumulated edge costs, not by geographic intuition or visual simplicity. As a result, the route contains several sharp turns and detours, even though the start, waypoints, and goal lie within a relatively compact region of the map. The path doubles back through certain areas after reaching a waypoint instead of following a more globally smooth trajectory.

Unexpected properties. A notable property of the solution is that it prioritizes local cost optimality between successive waypoint states, rather than producing a single globally intuitive route. Even though the waypoints are unordered, the resulting path may appear inefficient or convoluted to a human observer. This behavior is expected: the state space encodes both location and the set of visited waypoints, and UCS expands states strictly by cost, without regard to route simplicity or visual continuity.

Limitations. This example highlights several limitations of the approach. First, the cost function does not penalize frequent turning, backtracking, or path complexity, even though such features negatively impact usability and comfort. Second, because UCS treats each waypoint subset as a distinct state, the algorithm may revisit the same physical locations multiple times under different waypoint-visited conditions, leading to visually redundant paths. Finally, the planner does not consider higher-level human preferences such as minimizing turns or following major corridors, which can cause the generated route to diverge from what a person would naturally choose.

Overall, while the route is optimal under the specified cost model and waypoint constraints, it demonstrates how shortest-path optimization with unordered waypoints can produce solutions that are correct but unintuitive, motivating the use of richer cost functions or heuristic guidance.

2.d Unordered waypoints, driver well-being, and risks

Allowing unordered waypoints in route planning can improve driver well-being in several ways. By giving the system flexibility in the order of stops, drivers can avoid unnecessary backtracking, reduce total driving time, and lower cognitive load when managing multiple destinations. This can decrease stress and fatigue, especially in ride-sharing or delivery settings where drivers must visit many locations in a single trip. Unordered waypoints also allow routes to adapt more naturally to real-time conditions such as traffic, road closures, or personal preferences (e.g., taking breaks earlier in the route).

However, the same feature can be misused. Platforms could exploit unordered waypoint optimization to maximize efficiency at the expense of drivers, dynamically reordering stops to increase workload, extend shifts, or prioritize platform profits over driver comfort and safety. In extreme cases, drivers may feel pressured to follow algorithmically optimized routes that are legally optimal but physically demanding or unsafe.

There are also risks associated with collecting driver-specific data. Learning personalized waypoint preferences or performance-based routing requires detailed logs of driver behavior, locations, and schedules. Such data can reveal sensitive information about routines and habits and may be used for excessive monitoring, unfair evaluation, or discriminatory treatment. Mitigating these risks requires transparency about data use, limits on retention, and giving drivers meaningful control over how routing and personalization features are applied.

Problem 3

3.d When the waypoint-ignoring heuristic gives no asymptotic speedup

Figure 4 illustrates a case where the waypoint-ignoring heuristic provides little to no asymptotic improvement over Uniform Cost Search. In this example, the start location and the destination are connected primarily through a single narrow corridor with very limited branching alternatives. Although intermediate points exist nearby, the underlying road network effectively constrains all reasonable paths to follow the same main route.

In this setting, ignoring waypoint constraints yields a heuristic that closely matches the true remaining path cost for most states. However, because the graph structure itself offers almost no alternative routes, the search space is essentially linear. As a result, both UCS and A* with the waypoint-ignoring heuristic expand states in nearly the same order and explore a similar number of states before reaching the goal.

This example demonstrates that heuristic speedup depends not only on heuristic accuracy but also on the structure of the state space. When the environment lacks significant branching or large irrelevant regions, even an admissible and informative heuristic may fail to reduce the asymptotic number of expanded states.

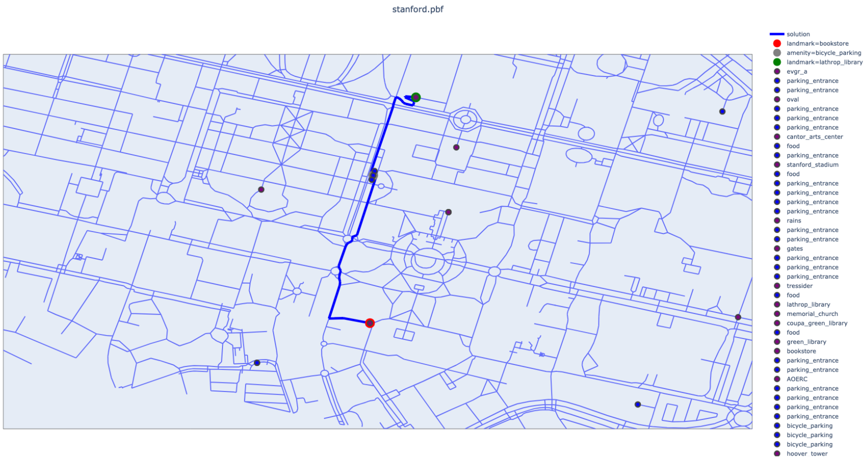

Figure 4: Example where the waypoint-ignoring heuristic offers no asymptotic speedup. The start and goal are connected through a largely linear corridor, causing UCS and A* to behave similarly.