Homework 2: Motion Planning Solutions

Problems

In square brackets are the points assigned to each problem.

1. [30 pts] In class, we showed that the deterministic shortest path (DSP) problem and the deterministic finite state (DFS) optimal control problem are equivalent. We used this equivalence to apply ideas from dynamic programming to the shortest path problem. In this problem, we will consider the reverse direction applying shortest path algorithms to the DFS optimal control problem.

   (a) Derive a formulation of the best-first search (Dijkstras) algorithm that applies to the DFS optimal control problem. Obtain a backward version of the algorithm that starts from the terminal time. Discuss how your formulation compares to the dynamic programming algorithm, e.g., does it investigate fewer states?

   (b) Is it possible to define an A^* algorithm for the DFS optimal control problem? What would a heuristic mean in this case? If it is possible, derive a backward version of the A^* algorithm; if not, explain what prevents the algorithm from being used.

   Solutions

   (a) The equivalence of DFS and DSP problem is omitted, since it can be easily found at slides of lecture 4 on Page $4, 5$. Note that in the backward Dijkstra's algorithm 1, the meaning of **Children** is different here. It should be clearly stated out that node j is a child of node i means: $j = (t, x_t), i = (t+1, x_{t+1})$ satisfies $x_{t+1} = f(x_t, u_t)$, because the arrow direction of the graph in the lecture node at page 6 is indicating i is a child of j. The **parent** use in line 8 has a reverse meaning as well. And g-value should be interpreted as remained- cost-to-goal, keeps the lowest cost from τ to each visited node $i \in \mathcal{V}$.

   In most of the case, the Backward Dijkstra's Algorithm (BDA) investigate fewer states than Dynamic Programming Algorithm (DPA). First, DPA needs to "expands" all nodes at each iteration, while BDA only explore nodes from OPEN list in a systematic greedy way. Second shortest path algorithm (includes BDA) usually has early stop rule, which terminates the algorithm once the goal is reached, while DPA has to find out shortest path starting from every nodes. These two reasons make BDA outperform DPA in this particular problem setting.

   (b) It is possible to define A^* for very simple DFS optimal control problem, when states connection is straightforward and cost is easily defined like path length. But in general, this is very difficult. I realize that some people missed the point of the problem. I give 4 points for free for those made response to this problem. In addition, good discussions on different aspects of heuristic design will earn another 1-2 points. Define nontrivial heuristic ($h \neq 0$) on states space is usually not possible. Several reasons are given as follows:

   - Biggest reason is that when system dynamic is involved, a proper edge cost is very hard to define. For example, a simple kinematic model for a differential drive wheel robot with states $s = (x, y, \theta)$. Consider two states $s_0 = (x_0, y_0, \theta_0)$ and $s_1 = (x_0 + \epsilon, y_0, \theta_0)$, where $\epsilon > 0$ is an arbitrary small scalar. They are very "close" in Euclidean norm, i.e. $\|s_1 - s_0\|_2$ is small, however after a second thought you will realize that the real cost is very high because the the robot cannot move in parallel and lots of maneuver is needed to reach s_1 from s_0. This example illustrates the difficulties of define a valid cost.
   - Heuristic admissibility based on distance to goal (vector norm based metric) might not work on non- Euclidean states space.
   - Heuristic development needs to satisfy the consistency constraints but the triangular inequality is again hard to define for general non-Euclidean states space.

Algorithm 1 Backward Dijkstra's Algorithm

OPEN $\leftarrow \{\tau\}$, $g_\tau = 0$, $g_i = \infty$ for all $i \in \mathcal{V} \setminus \{\tau\}$
while OPEN is not empty **do**
  $i = \arg\min_{k \in \text{ OPEN}} g_k$
  Remove i from OPEN
  for $j \in$ "Children" (i) **do** ▷ Here, the **Children** has a different meaning
    if $(g_i + c_{ij}) < g_j$ **and** $(g_i + c_{ij} < g_s)$ **then**
      $g_j \leftarrow (g_i + c_{ij})$
      Parent(j) $\leftarrow i$ ▷ Here, the **Parent** has a different meaning
      if $j \neq s$ **then**
        OPEN $\leftarrow$ OPEN $\cup \{j\}$

2. [36 pts] In this problem, we will investigate the properties of heuristic functions. As a reminder, a heuristic function is **admissible** if $h(\mathbf{x}_i) \leq$ **dist**$(\mathbf{x}_i, \mathbf{x}_\tau)$ for every node i with coordinates $\mathbf{x}_i$, where **dist**$(\mathbf{x}_i, \mathbf{x}_\tau)$ is the shortest distance from $\mathbf{x}_i$ to the goal $\mathbf{x}_\tau$. A heuristic function h is **consistent** if:

   - $h(\mathbf{x}_\tau) = 0$ for the goal node τ with coordinates $\mathbf{x}_\tau$
   - $h(\mathbf{x}_i) \leq c(\mathbf{x}_i, \mathbf{x}_j) + h(\mathbf{x}_j)$ for every node j with coordinates $\mathbf{x}_i$ and its children j with coordinates $\mathbf{x}_j$

   Finally, a heuristic h is $\epsilon-$**consistent** with $\epsilon \geq 1$ if:

   - $h(\mathbf{x}_\tau) = 0$ for the goal node τ with coordinates $\mathbf{x}_\tau$
   - $h(\mathbf{x}_i) \leq \epsilon c(\mathbf{x}_i, \mathbf{x}_j)$ for every node j with coordinates $\mathbf{x}_i$ and its children j with coordinates $\mathbf{x}_j$

   (a) Prove that if $h^{(1)}$ and $h^{(2)}$ are consistent heuristics, then $h \coloneqq \max\left\{h^{(1)}, h^{(2)}\right\}$ is also consistent.

   (b) Prove that if $h^{(1)}$ and $h^{(2)}$ are consistent heuristics, then $h \coloneqq h^{(1)} + h^{(2)}$ is ϵ-consistent.

   (c) Prove that if h is consistent, then it is also admissible.

   (d) Given an example of an admissible heuristic h that is not consistent. Show explicitly why h violates consistency

   (e) Prove that if the heuristic function used in A^* is consistent, then A^* will not re-open nodes. In other words, show that it is not possible to improve the label g_j of any node j that has already been expanded and placed in the CLOSED list.

   (f) Consider the stage cost $c(\mathbf{x}_i, \mathbf{x}_j) \coloneqq \|\mathbf{x}_i - \mathbf{x}_j\|_2$ and the heuristic function $h(\mathbf{x}_i) \coloneqq \|\mathbf{x}_i - \mathbf{x}_\tau\|_\infty + 0.4\|\mathbf{x}_i - \mathbf{x}_\tau\|_{-\infty}$, where for a vector $\mathbf{x} \in \mathbb{R}^D$ with elements $\mathbf{x}[d]$, the functions $\|\cdot\|_\infty$ and $\|\cdot\|_{-\infty}$ are defined as defined as:

$$\|\mathbf{x}\|_\infty \coloneqq \max_d |\mathbf{x}[d]| \quad \|\mathbf{x}\|_{-\infty} \coloneqq \min_d |\mathbf{x}[d]|$$

   Is h consistent? Is h admissible? Provide a proof or counter-example.

Solutions For convenience, define $h(i) \coloneqq h(\mathbf{x}_i)$ and $c(i,j) \coloneqq c(\mathbf{x}_i, \mathbf{x}_j)$ for all $i, j \in \mathcal{V}$ and let $h_{\max} \coloneqq \max\left\{h^{(1)}, h^{(2)}\right\}$ and $h_{\text{sum}} \coloneqq h^{(1)} + h^{(2)}$

(a) *Proof.* Since $h^{(1)}$ and $h^{(2)}$ are consistent heuristics, then the following holds:

$$h^{(1)}(\tau) = 0 \quad \text{and } h^{(1)}(i) \leq c(i,j) + h^{(1)}(j) \quad \forall j \in \text{childern(i)} \tag{1a}$$

$$h^{(2)}(\tau) = 0 \quad \text{and } h^{(2)}(i) \leq c(i,j) + h^{(2)}(j) \quad \forall j \in \text{childern(i)} \tag{1b}$$

Then we have $h_{\max}(\tau) = \max\left\{h^{(1)}(\tau), h^{(2)}(\tau)\right\} = 0$ and by (1) we have

$$h_{\max}(i) = \max\left\{h^{(1)}(i), h^{(2)}(i)\right\} \leq c(i,j) + \max\left\{h^{(1)}(j), h^{(2)}(j)\right\} = c(i,j) + h_{\max}(j)$$

holds for all $\forall j \in$ childern(i).

To be more specific, you can enumerate all four situations with two cases directly follows from (1).

- $h_{\max}(i) = h^{(1)}(i)$ and $h_{\max}(j) = h^{(1)}(j)$
- $h_{\max}(i) = h^{(2)}(i)$ and $h_{\max}(j) = h^{(2)}(j)$
- $h_{\max}(i) = h^{(1)}(i)$ and $h_{\max}(j) = h^{(2)}(j)$

$$h_{\max}(i) = h^{(1)}(i) \leq c(i,j) + h^{(1)}(j) \leq c(i,j) + h^{(2)}(j) = c(i,j) + h_{\max}(j) \tag{2}$$

- $h_{\max}(i) = h^{(2)}(i)$ and $h_{\max}(j) = h^{(1)}(j)$

$$h_{\max}(i) = h^{(2)}(i) \leq c(i,j) + h^{(2)}(j) \leq c(i,j) + h^{(1)}(j) = c(i,j) + h_{\max}(j) \tag{3}$$

□

(b) *Proof.* Sum the two equations in (1), then h_{sum} is $\epsilon-$consistent heuristics with $\epsilon = 2$. □

(c) Prove that if h is consistent, then it is also admissible.

Proof. Proof by contradiction. Assume that h is consistent, but it is not admissible. Then, from definition of admissible, we know there exists at least one node $i \in \mathcal{V}$ such that $h(i) > \mathbf{dist}(\mathbf{x}_i, \mathbf{x}_\tau)$. If $i \neq \tau$ and i is connected to τ, i.e. there exist finite-length path from i to τ, then there must exist node j as children node of i on one of shortest paths (you must mention this even if you are using induction) from i to τ. Hence,

$$h(i) > \mathbf{dist}(\mathbf{x}_i, \mathbf{x}_\tau) = c_{ij} + \mathbf{dist}(\mathbf{x}_j, \mathbf{x}_\tau) \geq c_{ij} + h(j) \tag{4}$$

which contradicts the assumption that h is consistent. Let us consider remained corner cases to complete the proof. If $i = \tau$, then $h(\tau) > \mathbf{dist}(\mathbf{x}_\tau, \mathbf{x}_\tau) = 0$ which leads to contradiction. If no i and τ is not connected, then $\mathbf{dist}(\mathbf{x}_i, \mathbf{x}_\tau) = \infty$, cannot exist i such that $h(i) > \mathbf{dist}(\mathbf{x}_i, \mathbf{x}_\tau)$. □

(d) Consider the following graph with only three nodes $1, 2, 3$, starting node $s = 1$ and terminal node $\tau = 3$, edge cost and heuristic is shown in Fig 1. The heuristic is admissible which can be easily verified. but $h(1) > c(1,2) + h(2)$. So this heuristic is not consistent.

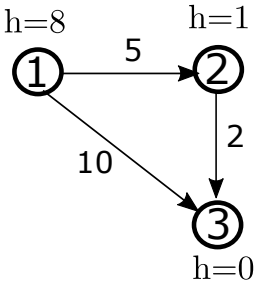

Figure 1: Example of admissible but inconsistent heruristic

(e) *Proof.* Prove by contradiction. Suppose heuristic function h used in A^* is consistent. A^* will re-open at least one nodes, i.e., it is possible to improve label of some nodes that have already been expanded and placed in the CLOSED list.

Suppose there exist $i \in$ has been expanded and already been put in CLOSED list whose label g_i can be improved. It is clear that node i has been extracted from OPEN list at early time. Define OPEN list at that iteration named $\mathcal{O}_{t_0}$, and let $\mathcal{O}_t$ be OPEN list after that iteration if $t > t_0$. Use above notation we have two statements:

- **Statement 1** $f(i) \leq f(i')$ for all $i' \in \mathcal{O}_{t_0}$
- **Statement 2** There exists $j \in \mathcal{O}_{t_1}$ where $t_1 \geq t_0$, when extracted from OPEN list, it can re-open i in CLOSED list, i.e., i is a child of j and $g(i) > g(j) + c(j,i)$.

First, let us show that future node to expand will has a bigger f value, i.e. $f(i_t) \geq f(i_{t_0})$ where $i_t \in \mathcal{O}_t$ and $i_{t_0} \in \mathcal{O}_{t_0}$, where $t \geq t_0$. From **Statement 1**, if $t = t_0$ then $f(i_t) \geq f(i)$ for all $i_t \in \mathcal{O}_{t_0}$. If $t > t_0$, consider new node k been explored from some node in $\mathcal{O}_{t_0}$ (node k parent is in $\mathcal{O}_{t_0}$), its f value is greater than any node $i_t \in \mathcal{O}_{t_0}$ as shown in below

$$\begin{aligned} f(k) &= g(k) + h(k) && && (5)\\ &= g(i_t) + c(i_t, k) + h(k) && k \in \text{Children}(i_t) && (6)\\ &\geq g(i_t) + c(i_t, k) + [h(i_t) - c(i_t, k)] && h \text{ is consistent} && (7)\\ &= f(i_t) && && (8) \end{aligned}$$

Use this approach iteratively, we can show that all future expanding nodes will have a bigger f value, i.e. $f(i_t) \geq f(i)$ for all $i_t \in \mathcal{O}_t \setminus \{i\}$ where $t \geq t_0$.
Next from **Statement 2**, we have

$$f(i) = g(i) + h(i) > [g(j) + c(j, i)] + h(i) \geq g(j) + h(j) = f(j) \tag{9}$$

where the second inequality comes from h is consistent. Further note that $j \in \mathcal{O}_t \setminus \{i\}$, which leads to a contradiction. □

(f) h is not admissible, and it is not consistent neither. We will construct low dimension counter-example when $d = 2$, and a counter-example is given after some basic analysis. For convenience, consider simple graph with only two connected nodes, i and terminal node τ with coordinates x_i with coordinates (ξ_1, ξ_2) and $x_\tau = (0, 0)$, where $\xi_1 > \xi_2 > 0$. If heuristic function h is admissible then h should satisfies $h(\tau) = 0$ and $h(x_i) \leq \mathbf{dist}(\mathbf{x}_i, \mathbf{x}_\tau) = c(x_i, x_\tau)$, which implies

$$h^2(x_i) \leq c^2(x_i, x_\tau) \implies (\xi_1 + 0.4\xi_2)^2 \leq \xi_1^2 + \xi_2^2 \implies \xi_2(0.8\xi_1 - 0.84\xi_2) \leq 0$$

Therefore, if we choose $0.8\xi_1 > 0.84\xi_2$, then h is not admissible, neither consistent. For example, choose $\xi_1 = 10, \xi_2 = 1$.

3. [42 pts] Suppose that the charge of your cell phone's battery takes on values $[n] := 1, \ldots, n$ from better to worse. Your browsing and chatting experience has a decreasing reward $r(i)$ in $i \in [n]$. Also, the worse your battery state is, the more likely it is to get worse. Formally, $P(i, j)$ describes the probability of transitioning from state i to state j and the quantity $\sum_{j=l}^{n} P(i, j)$ is increasing in i for all $l \in [n]$ (stochastic dominance). You have the option to recharge your cell phone, which resets its state back to 1 with electricity costs of c. Unfortunately, your cell phone company is not great at making chargers and your charger only works with probability $q \in (0, 1)$. If it happens that the charger does not work, the battery state remains unchanged and the electricity cost is 0.
Solution:

(a) Problem can be formulated as minimizing following infinite-horizon problem $V^*(x) = \min_\pi V^\pi(x) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \ell(x_t, \pi(x_t)) | x_0 = x]$ where state space is $\chi = \{1, 2, \ldots, n\}$, and for each state the control space either charges or not, i.e. for all states in $\chi = \{1, 2, ..., n\}$ like x we have $\mathcal{U}(x) = \{R = recharge, N = no\ recharge\}$, and the transition distribution could be described as follows:

$$p_f(x'|x, N) = P(x, x')$$
$$p_f(x'|x, R) = \begin{cases} x' = x & \text{with probability } 1 - q \\ x' = 1 & \text{with probability } q \end{cases}$$

And the stage cost is:

$$\ell(x, N) = -r(x)$$
$$\ell(x, R) = \begin{cases} -r(x) & \text{with probability } 1 - q \\ -r(x) + c & \text{with probability } q \end{cases}$$

Since the expected of stage cost is effective so the problem is equivalent to the similar problem with stage cost of:

$$\ell(x, N) = -r(x)$$
$$\ell(x, R) = -r(x) + cq$$

So the bellman equation will be:

$$V^*(x) = \min_{u \in \mathcal{U}(x)} [\ell(x, u) + \gamma \sum_{x' \in \chi} p_f(x'|x, u) V^*(x')] \tag{10}$$

$$= \min_{u \in \{R, N\}} [\ell(x, u) + \gamma \sum_{i=1}^{n} p_f(i|x, u) V^*(i)] \tag{11}$$

$$= \min[-r(x) + qc + q\gamma V^*(1) + (1-q)\gamma V^*(x), -r(x) + \gamma \sum_{i=1}^{n} P(x, i) V^*(i)] \tag{12}$$

(b) Through value iteration algorithm, i.e., for $V_0(x)$ we can initialize it such that it will be increasing in x, now we will prove via induction on t that $V_t(x)$ will stay increasing for all t:
Induction base: this step is trivial as we have chosen $V_0(x)$ such that it would be increasing in x.
Induction hypothesis: lets assume for $t = k$ we know $V_k(x)$ is increasing in x.
Induction step:
Lets define $q_{i,j} = \sum_{l=j}^{n} P(i, l)$, (note that $q_{i,1} = 1$) then we have $P_{i,j} = q_{i,j} - q_{i,j+1}$:

$$\sum_{j=1}^{n} P(x, j) V_k(j) = \sum_{j=1}^{n} (q_{x,j} - q_{x,j+1}) V_k(j) \tag{13}$$

$$= \sum_{j=1}^{n} q_{x,j} V_k(j) - \sum_{j=1}^{n} q_{x,j+1} V_k(j) \tag{14}$$

$$= \sum_{j=1}^{n} q_{i,j} V_k(j) - \sum_{j=2}^{n} q_{x,j} V_k(j-1) \tag{15}$$

$$= q_{x,1} V_k(1) + \sum_{j=2}^{n} q_{x,j} V_k(j) - \sum_{j=2}^{n} q_{x,j} V_k(j-1) \tag{16}$$

$$= V_k(1) + \sum_{j=2}^{n} q_{x,j} (V_k(j) - V_k(j-1)) \tag{17}$$

based on induction hypothesis $V_k(j)$ is increasing in j so all terms $V_k(j) - V_k(j-1)$ are non-negative, and from stochastic dominance $q_{x,j}$ is increasing in x (and it is non negative since is sum of probabilities), so $V_k(1) + \sum_{j=2}^{n} q_{x,j}(V_k(j) - V_k(j-1))$ is increasing in x, so $\gamma \sum_{j=1}^{n} P(x, j) V_k(j)$ is increasing in x, on the other hand since $V_k(j)$ is increasing in j, so $qc + q\gamma V^*(1) + (1-q)\gamma V^*(x)$ is increasing in x, since minimum of two increasing function is increasing, so $\min[qc + q\gamma V_k(1) + (1-q)\gamma V_k(x), \gamma \sum_{i=1}^{n} P(x, i) V_k(i)]$ is increasing in x, $-r(x)$ is increasing in x, since sum of two increasing function is increasing, so $V_{k+1}(x) = -r(x) + \min[qc + q\gamma V_k(1) + (1-q)\gamma V_k(x), \gamma \sum_{i=1}^{n} P(x, i) V_k(i)]$ is increasing in x.

Since $V_t(x)$ is increasing in x for all t, so $V^*(x) = V_\infty(x)$ is increasing in x.

(c) In part b we proved that $V^*(x) = \min[-r(x) + qc + q\gamma V^*(1) + (1-q)\gamma V^*(x), -r(x) + \gamma \sum_{i=1}^{n} P(x, i) V^*(i)]$, so $-r(x) + \gamma \sum_{i=1}^{n} P(x, i) V^*(i) \geq V^*(x)$. Since $r(x) \geq 0$ so $\gamma \sum_{i=1}^{n} P(x, i) V^*(i) \geq V^*(x)$. $\gamma \sum_{i=1}^{n} P(x, i) V^*(i) - (1-q)\gamma V^*(x) \geq (1 - (1-q)\gamma) V^*(x)$. Since $(1-q)\gamma < 1$, and $V^*(x)$ is increasing in x, so if q is close enough to one then $[-r(x) + \gamma \sum_{i=1}^{n} P(x, i) V^*(i) \geq V^*(x)] - [-r(x) + (1-q)\gamma V^*(x)]$ will be increasing in x, since to choose optimal policy, it must be compared with $qc + q\gamma V^*(1)$ which is constant, so the optimal policies are of threshold nature.