Supplementary material: Planning with Abstract Markov Decision Processes

Nakul Gopalan[*] ngopalan@cs.brown.edu
Marie desJardins[†] mariedj@umbc.edu
Michael L. Littman[*] mlittman@cs.brown.edu
James MacGlashan[‡§] james@cogitai.com
Shawn Squire[†] ssquire1@umbc.edu
Stefanie Tellex[*] stefie10@cs.brown.edu
John Winder[†] jwinder1@umbc.edu
Lawson L.S. Wong[*] lsw@brown.edu

1 Introduction

In the supplementary material we describe the AMDPs at each level of the hierarchy within Taxi (Dieterrich 2000) and Cleanup World problems (MacGlashan et al. 2015). We outline all the objects present in the states and their attributes as per the OOMDP formulation (Diuk, Cohen, and Littman 2008); the actions available in the AMDP; the assumed transition function; reward functions; set of terminating states and state projection for the hierarchies. When describing the states, each object present within the state is listed, along with their attributes and possible values that the attributes can take. All projections are done from the environment to the current level mentioned.

2 AMDPs for the Fickle Taxi Problem

We will define the AMDPs starting from the highest AMDPs down to the source MDP.

2.1 Level 2 AMDP

This is the AMDP defined at the Root. The objective of the root is solve the entire problem using its subgoals. At this level the Passenger simply teleports between locations without any Cartesian co-ordinates.

1. States, $\tilde{\mathcal{S}}_2$:
   (a) Locations: The respective colors of locations.
   (b) Passenger: Source location color, current location, goal location color, a binary bit to indicate that the passenger has been picked up at least once and is not in a taxi.
2. Actions, $\tilde{\mathcal{A}}_2$: Get Passenger, Put Passenger. These are the subgoal that result in the passenger being picked up at any location and dropped off at the goal location.
3. Transition Function, $\tilde{\mathcal{T}}_2(s, a, s')$: The transition probability for Get to result in the passenger being in the taxi is deterministic. Similarly the transition probability for the passenger to be dropped off at the goal location after the Put action is deterministic. However this transition probability is untrue, as the passenger can change their goal after being picked up. However, this is not an issue as we use an online planner for the Put subtask which replans when the agent lands in an unexpected state.
4. Reward Function, $\tilde{\mathcal{R}}_2(s, a, s')$: Each action for this AMDP earns a 0 reward, unless it reaches the goal of delivering the passenger, when a terminal reward of 1 is given.
5. Termination set, $\tilde{\mathcal{E}}_2$: This is the set of all states in $\tilde{\mathcal{S}}_2$, in which the passenger is present at their desired goal location, after having been picked up at least once. This termination condition is specified according to the environment MDP's goal projected to the root's level.
6. State projection function, F_2: This function extracts the passenger's current location's color, if it is in one of the four locations, else labels the passenger as being on road. It retains the passenger's goal color, source locations color and if they were ever picked up by a taxi. The locations are extracted with only their color attributes.

2.2 Level 1 AMDP

This AMDP is the subgoal of either a Get or a Put AMDP.

1. States, $\tilde{\mathcal{S}}_1$:
   (a) Taxi: Current location of the taxi, if it is not in one of the four locations, the taxi is "on road". A binary bit to indicate that the passenger is in the taxi.
   (b) Locations: The respective colors of locations.
   (c) Passenger: Source location color, goal location color, a binary bit to indicate that the passenger has been picked up at least once.
2. Actions, $\tilde{\mathcal{A}}_1$: Depending on which subgoal spawns the current AMDP, Get Passenger or Put passenger from

[1]Brown University, Providence, RI, 02912
[2]University of Maryland, Baltimore County, Baltimore, MD 21250
[3]Cogitai, Inc.
[4]Work done while at Brown University

level 2, we get get two action sets. The navigate actions are common to both types of AMDPs. However the Pickup action to pick the passenger is available only to the Get Passenger AMDP; and the put down action is available only to the Put AMDP. The navigate actions are: Navigate to Red location, Navigate to Yellow location, Navigate to Green location and Navigate to Blue location.

It is important to note that the Pickup and Putdown actions are grounded in the environment as they are primitive actions. After their execution, the resulting state is project up to represent the state in the current AMDP.

3. Transition Function, $\tilde{\mathcal{T}}_1(s, a, s')$: The navigate actions are modeled deterministic, however in the source environment they are not, as the passenger can change their destination when navigating. Because the planners used are online, they replan if an unexpected transition occurs. A more precise model will only help the planner solve the task better, we designed crude transition and reward functions to test the capabilities of AMDPs. The navigate actions do not change the relationship between a taxi and the passenger, since it does not have methods to pick or drop a passenger off. The Pickup action is deterministic and the passenger is picked up if the taxi is at the passenger's location. The Putdown action is also deterministic, and the passenger can only be dropped off when the passenger is at their goal location.
4. Reward Function, $\tilde{\mathcal{R}}_1(s, a, s')$: Each action earns a reward of 0. When the passenger is within the taxi for the Get AMDP, a terminal reward of 1 is given, and when the passenger is dropped off at the goal location for the Put AMDP a terminal reward of 1 is given.
5. Termination set, $\tilde{\mathcal{E}}_1$: For a Get AMDP, this is the set of states, in $\tilde{\mathcal{S}}_1$, in which the passenger is within the taxi; and for the Put AMDP this is when the passenger has been dropped to their goal location.
6. State projection function, F_1: This function extracts the taxi's current location's color, if it is in one of the four locations, else labels the passenger as being on road. It retains the passenger's goal color, source locations color and if they were ever picked up by a taxi. The locations extracted with only their color attributes. The taxi also extract the the binary bit to indicate if the passenger is present in the taxi from the environment state.

2.3 Level 0 AMDP

This is the lowest AMDP which is spawned by the Navigate AMDPs.

1. States, $\tilde{\mathcal{S}}_0$:
   (a) Taxi: Cartesian co-ordinates (x, y) for taxi's position. A binary bit to indicate that the passenger is in the taxi.
   (b) Locations: Cartesian co-ordinates of the locations, and their respective colors.
   (c) Passenger: Cartesian co-ordinates, source location color, goal location color, a binary bit to indicate that the passenger has been picked up at least once
2. Actions, $\tilde{\mathcal{A}}_0$: North, South, East, West. These describe the taxi's movement actions available when navigating .
3. Transition Function, $\tilde{\mathcal{T}}_0(s, a, s')$: We use the transition dynamics for the source MDP directly here. The movement actions succeed with a probability of 0.8. With probability 0.1 the taxi slips in the each of the two directions orthogonal to the desired movement action. Further, when the passenger has been picked up, after a step of movement, the passenger can change their destination with a probability of 0.3 equally divided between all locations. With probability 0.7 the passenger does not change their location.
4. Reward Function, $\tilde{\mathcal{R}}_0(s, a, s')$: The agent get a terminal reward of 1 when the navigate task is complete.
5. Termination set, $\tilde{\mathcal{E}}_0$: This is the set of all states in which the taxi has been navigated to the goal location of the Navigate AMDP.
6. F_0: There is an identity projection, that is, we keep the entire source MDP state at this level, as these actions are directly getting grounded in the source MDP.

2.4 Source MDP

1. States, $\mathcal{S}$:
   (a) Taxi: Cartesian co-ordinates (x, y) for taxi's position. A binary bit to indicate that the passenger is in the taxi.
   (b) Locations: Cartesian co-ordinates of the locations, and their respective colors.
   (c) Passenger: Cartesian co-ordinates, source location color, goal location color, a binary bit to indicate that the passenger has been picked up at least once
2. Actions, $\mathcal{A}$: North, South, East, West, Pickup, Dropoff. These describe the taxi's movement actions, and actions of the passenger being picked up and dropped off.
3. Transition Function, $\mathcal{T}(s, a, s')$: The movement actions succeed with a probability of 0.8. With probability 0.1 the taxi slips in the each of the two directions orthogonal to the desired movement action. Further, when the passenger has been picked up, after a step of movement, the passenger can change their destination with a probability of 0.3 equally divided between all locations. With probability 0.7 the passenger does not change their destination.
4. Environment Reward Function, $\mathcal{R}(s, a, s')$: The source MDP has a true reward function. Each action earns a penalty of -1. When the passenger is dropped off to their desired goal location, after being

picked up at least once, the agent earns a reward of 20.

5. Termination set, $\mathcal{E}$: This is the set of all states in which the passenger is present at their desired goal location, after having been picked up at least once.

3 AMDPs for the Cleanup World Problem

We define, once again, the AMDPs starting from the highest AMDPs down to the source MDP. This section is for the AMDPs over the original Cleanup Domain. The changes between this domain and continuous Cleanup Domain will be presented in the next section.

3.1 Level 2 AMDP

This is the AMDP defined at the Root. The objective of the root is solve the entire problem using its subgoals, that is to move the required object into the goal room.

1. States, $\tilde{\mathcal{S}}_2$:
   (a) Agent: The agent has color attribute corresponding to the room it is in.
   (b) Room: Rooms have an attribute for their color.
   (c) Objects: Objects have attributes of shape, color, and the color of the room they are in.
2. Actions, $\tilde{\mathcal{A}}_2$: At this level there are two types actions that can be grounded to different objects and rooms. The First type of action is Move object(i) to room(j) action moves the specified object to the specified room, which can be grounded to different rooms and different different objects. Further, is the Move to room(j) action, where the agent it self moves to the room specified.
3. Transition Function, $\tilde{\mathcal{T}}_2(s, a, s')$: The transition probability for both actions are deterministic depending on the design of the rooms. Generally a room has multiple doors, so the agent can get to any room needed. However room configurations can be designed for which this transition would not be true. However, we can still plan with these approximate models.
4. Reward Function, $\tilde{\mathcal{R}}_2(s, a, s')$: Each action for this AMDP earns a 0 reward, unless the desired object is moved to the desired room, when a terminal reward of 1 is given.
5. Termination set, $\tilde{\mathcal{E}}_2$: This is the set of all states in $\tilde{\mathcal{S}}_2$, in which the desired object is present in its goal room.
6. State projection function, F_2: This function extracts the agent's current room, the object's current room, and the room's color attribute. Further the objects are specified with their color and shape attributes. There are propositional functions that confirm if an agent is within a room, allowing such a state abstraction.

3.2 Level 1 AMDP

This AMDP is the subgoal of either a Move Object to Room AMDP or a Move agent to Room AMDP.

1. States, $\tilde{\mathcal{S}}_1$:
   (a) Agent: The agent has a relational attribute corresponding to the door or room it is in.
   (b) Room: Rooms have an attribute for their color, and a relational attribute to the doors they connect to.
   (c) Door: There are doors at this level which can be locked, further the doors maintain the relational attributes with rooms they connect to. The lock attribute is unknown until the agent tries to open the door, after which the attribute is stochastically chosen.
   (d) Objects: Objects have attributes of shape, color, and the relational attribute to the door or room they are in.
2. Actions, $\tilde{\mathcal{A}}_1$: At this level there are four types actions that can be grounded to different objects and rooms: Move object(i) to door(j), Move object(i) to room(j), Move agent to room(j) and Move agent to door(j). They can be grounded with different rooms and objects.
3. Transition Function, $\tilde{\mathcal{T}}_1(s, a, s')$: The transition functions at this level are stochastic depending on whether the doors are open. If the doors are closed then Moving to a door can fail, however then the locked door attribute would be known.
4. Reward Function, $\tilde{\mathcal{R}}_1(s, a, s')$: Each action earns a reward of 0. Unless the object or agent depending upon the parent MDP is in the desired room, which results in a terminal reward of 1.
5. Termination set, $\tilde{\mathcal{E}}_1$: For a Move object(i) to room(j) AMDP, the termination sets are all states in which the object(i) is present in room(j). For a Move to room(j) AMDP the termination set is the set of states in which the agent is present in room(j).
6. State projection function, F_1: This function extracts the agent's current room or door, the object's current room or door, the room's color attribute, and the door's lock attribute. Further the objects are specified with their color and shape attributes. There are propositional functions that confirm if an agent is within a room or door, allowing such a state abstraction.

3.3 Level 0 AMDP

This is the lowest AMDP which can be spawned by Move to door(i), Move to room(i), Move object(i) to room(j) and Move object(i) to door(j) actions.

1. States, $\tilde{\mathcal{S}}_0$:
   (a) Agent: The agent has Cartesian co-ordinates, and a cardinal direction that the agent is facing.

(b) Room: Rooms have an attribute for their color, and their boundary co-ordinates specified by the top-left and the bottom right corners.

(c) Door: The doors have a lock attribute, and it position attribute again with top-left and bottom right corner of the door.

(d) Objects: Objects have attributes of shape, color, and the Cartesian co-ordinates of the box.

2. Actions, $\tilde{\mathcal{A}}_0$: North, South, East, West and Pull. These describe the agent's movement actions, and actions of the passenger available when navigating .

3. Transition Function, $\tilde{\mathcal{T}}_0(s, a, s')$: The transition dynamics at this level are stochastic, since the doors can be open or closed with equal probability. The doors can be opened or closed if the agent tries to move into a door location.

4. Reward Function, $\tilde{\mathcal{R}}_0(s, a, s')$: The agent gets a terminal reward of 1 when the move task is complete with or without object depending on the AMDP.

5. Termination set, $\tilde{\mathcal{E}}_0$: This is the set of all states in which the object or the agent is present in the door or room specified by the AMDP's goal condition. The states in which the goal doors are locked are in the termination set too.

6. F_0: There is an identity projection, that is, we keep the entire source MDP state at this level, as these actions are directly getting grounded in the source MDP.

3.4 Source MDP

The source MDP differs from the Level 0 AMDP only in its termination set and reward function.

1. States, $\mathcal{S}$:

(a) Agent: The agent has Cartesian co-ordinates, and a cardinal direction that the agent is facing.

(b) Room: Rooms have an attribute for their color, and their boundary co-ordinates specified by the top-left and the bottom right corners.

(c) Door: The doors have a lock attribute, and it position attribute again with top-left and bottom right corner of the door.

(d) Objects: Objects have attributes of shape, color, and the Cartesian co-ordinates of the box.

2. Actions, $\mathcal{A}$: North, South, East, West and Pull. These describe the discrete actions the agent can take.

3. Transition Function, $\mathcal{T}(s, a, s')$: The transition dynamics at this level are stochastic, since the doors can be open or closed with equal probability. The doors can be opened or closed if the agent tries to move into a door location.

4. Reward Function, $\mathcal{R}(s, a, s')$: The agent gets a terminal reward of 1 when the entire move to room or move object to room task is complete.

5. Termination set, $\mathcal{E}$: The states in which the object or agent is present in the goal state is in the set of termination states.

4 AMDPs for the Continuous Cleanup World Problem

We define the AMDPs starting from the highest AMDPs down to the source MDP. The first two levels of the hierarchy are the shared between the two tasks.

4.1 Level 3 AMDP

This is the AMDP defined at the Root, and is exactly the same AMDP as defined in Section 3.1, except in this hierarchy the AMDP is a level higher because the lowest level is continuous.

4.2 Level 2 AMDP

This is the AMDP by the tasks of Move to room(i) or Move object(j) to room(i). It is exactly the same AMDP as defined in Section 3.2, except in this hierarchy the AMDP is a level higher because the lowest level is continuous.

4.3 Level 1 AMDP

The AMDPs at this level are spawned by Move to door(i), Move to room(i), Move object(i) to room(j) and Move object(i) to door(j) actions. It is the last continuous AMDP in this hierarchy. Only the actions of the agent have changed from the AMDP defined in Section 3.3

1. States, $\tilde{\mathcal{S}}_1$:

(a) Agent: The agent has Cartesian co-ordinates, and a cardinal direction that the agent is facing.

(b) Room: Rooms have an attribute for their color, and their boundary co-ordinates specified by the top-left and the bottom right corners.

(c) Door: The doors have a lock attribute, and it position attribute again with top-left and bottom right corner of the door.

(d) Objects: Object have attributes of shape, color, and the Cartesian co-ordinates of the box.

2. Actions, $\tilde{\mathcal{A}}_1$: Move Forward a cell, Move Back a call, Turn Clockwise by $\pi/2$ and Turn Anticlockwise by $\pi/2$.

3. Transition Function, $\tilde{\mathcal{T}}_0(s, a, s')$: The transition dynamics at this level are stochastic, since the doors can be open or closed with equal probability. The doors can be opened or closed if the agent tries to move into a door location.

4. Reward Function, $\tilde{\mathcal{R}}_1(s, a, s')$: The agent gets a terminal reward of 1 when the move task is complete with or without object depending on the AMDP.

5. Termination set, $\tilde{\mathcal{E}}_1$: This is the set of all states in which the object or the agent is present in the door or room specified by the AMDP's goal condition. The

states in which the goal doors are locked are in the termination set too.

6. F_1: The agent and the objects have a discrete (x,y) co-ordinate based on the grid cell it is present in. The agent also has a direction which is one of the cardinal directions based on the distance of the direction of the continuous from the cardinal directions. The rooms and doors just get their top-left and bottom-right locations extracted based on their grid locations.

4.4 Level 0 AMDP

This is the AMDP defined by Move Forward a cell, Move Back a call, Turn Clockwise by $\pi/2$ and Turn Anticlockwise by $\pi/2$. This AMDP is continuous. We used closed loop planners at this level to move by a cell or to turn by $\pi/2$. However a specifying a closed loop plan for the entire task is almost impossible.

1. States, $\tilde{\mathcal{S}}_0$:
   (a) Agent: The agent has continuous Cartesian co-ordinates, and a continuous direction in radians that the agent is facing with respect to the global North.
   (b) Room: Rooms have an attribute for their color, and their boundary co-ordinates specified by the top-left and the bottom right corners.
   (c) Door: The doors have a lock attribute, and it position attribute again with top-left and bottom right corner of the door.
   (d) Objects: Objects have attributes of shape, color, and continuous Cartesian co-ordinates of the objects.
2. Actions, $\tilde{\mathcal{A}}_0$: Move Forward, Move Back, Turn Clockwise and Turn Anticlockwise. The actions are parameterized to move by 0.1 m and 0.1 radians, but this can be arbitrary.
3. Transition Function, $\tilde{\mathcal{T}}(s, a, s')_0$: The transition dynamics at this level are stochastic, since the doors can be open or closed with equal probability. The doors can be opened or closed if the agent tries to move into a door location.
4. Reward Function, $\tilde{\mathcal{R}}(s, a, s')_0$: The agent gets a terminal reward of 1 when the move by cell or turn by $\pi/2$ tasks are complete.
5. Termination set, $\tilde{\mathcal{E}}_0$: This is the set of all states in which the agent has moved by a cell length or turned by $\pi/2$ according to the task specified.
6. F_0: There is an identity state abstraction, that is the states are the same between this level and the environment.

4.5 Source MDP

Only the reward function and the termination set differs between this level and Level 0.

1. States, $\mathcal{S}$:
   (a) Agent: The agent has continuous Cartesian co-ordinates, and a continuous direction in radians that the agent is facing with respect to the global North.
   (b) Room: Rooms have an attribute for their color, and their boundary co-ordinates specified by the top-left and the bottom right corners.
   (c) Door: The doors have a lock attribute, and it position attribute again with top-left and bottom right corner of the door.
   (d) Objects: Objects have attributes of shape, color, and continuous Cartesian co-ordinates of the objects.
2. Actions, $\mathcal{A}$: Move Forward, Move Back, Turn Clockwise and Turn Anticlockwise. The actions are parameterized to move by 0.1 m and 0.1 radians, but this can be arbitrary.
3. Transition Function, $\mathcal{T}(s, a, s')$: The transition dynamics at this level are stochastic, since the doors can be open or closed with equal probability. The doors can be opened or closed if the agent tries to move into a door location.
4. Reward Function, $\mathcal{R}(s, a, s')$: The agent gets a terminal reward of 1 when the entire moving object to goal task is complete.
5. Termination set, $\mathcal{E}$: This is the set of all states in which moved the required object to the goal room.

References

Dieterrich, T. 2000. Hierarchical reinforcement learning with the MAXQ value function decomposition. *JAIR* 13:227–303.

Diuk, C.; Cohen, A.; and Littman, M. 2008. An object-oriented representation for efficient reinforcement learning. In *ICML*.

MacGlashan, J.; Babes-Vroman, M.; desJardins, M.; Littman, M.; Muresan, S.; Squire, S.; Tellex, S.; Arumugam, D.; and Yang, L. 2015. Grounding english commands to reward functions. In *Robotics: Science and Systems*.