Jodie Simkoff
Reinforcement Learning HW
November 25, 2017

Introduction

Broadly speaking, reinforcement learning algorithms train agents to behave in a certain way in their given environment so as to maximize the reward that they obtain. The goal of any reinforcement learning algorithm is for the agent to learn an optimal policy by repeatedly interacting with the environment. Algorithms for reinforcement learning may involve models of the dynamics of the environment, which are fully or only partially known; alternatively, they may be model-free. In a model-free RL setting, the agents can only learn about the environment and improve their policies by interacting with it. Examples of model-free RL algorithms include Q-learning and SARSA.

Because of its generality and flexibility, reinforcement learning can be used to tackle many different classes of problems from scheduling to control to gaming. A simple example of a task that can be completed by a RL-trained agent is collecting litter from a sidewalk while avoiding obstacles. This is the task that will be explored in this work, using a Q-learning framework.

Methods

The problem explored in this work is as follows: the agent must navigate down a sidewalk (represented as a grid of dimension W x L) while achieving the following objectives:

(i) Reaching the end of the sidewalk
(ii) Collecting litter along the way (positive reward)
(iii) Avoiding obstacles (negative reward)

The litter and obstacles are generated randomly for a given sidewalk.

At any given time the state of the agent is defined as its x- and y- coordinates in the sidewalk. Assume the agent starts at the west side of the sidewalk and that the end of the sidewalk is to the east. We do not allow the agent to walk backward, so in a given state, there are five actions available to the agent: move north, move northeast, move east, move southeast, and move south. When an agent moves to a new position in the sidewalk, it may encounter either a piece of litter for which it is rewarded, or an obstacle for which it is penalized. The goal is to reach the end of the sidewalk while accruing litter and avoiding obstacles.

A Q-learning algorithm is used to arrive at an optimal policy for meeting these objectives. The first approach used is to train three modules separately: each consists of an agent which is only concerned with achieving one of the three stated objectives. Once trained, these agents can be used cooperatively and their performance assessed.

In this framework, the policy at a given state is evaluated as follows, where the subscript i denotes one module:

$$a(s) = \max\big(Q_i(s,a)\big) \qquad (1)$$

Q_i is a matrix of dimension WxLx5 for this case where there are 5 available actions per state. It maps a state-action pair, (s,a) to a real number which represents the expected reward for taking the given action in the given state.

Once an action is performed during the learning phase, the Q-matrix is incremented according to the update law:

$$Q_i(s,a) = (1-\alpha_i)Q_i(s,a) + \alpha_i(r_i + \gamma_i(\max_a Q_i(s',a')) \qquad (2)$$

The parameters α and γ are, respectively, the learning rate and the discount factor, and subscript *i* denotes the module. The learning rate determines the extent to which new information changes the policy. As α tends to 1, the new information completely overrides the old Q value for that state-action pair, while if this parameter tends to 0 then the new information will have no effect on the Q value. The discount factor represents the importance of future rewards; note that it multiplies the expected reward in the *next* state which the agent moves to after taking the current action.

For the case of training the modules separately, the reward values are set as follows:

Module	r_i
Litter collection	+3 if litter 0 otherwise
Obstacle avoidance	-1 if obstacle 0 otherwise
Reaching the end	+ 10 if end (x^+ - x) otherwise

Table 1. Reward scheme

These can of course be tuned according to the relative importance of the objectives. Learning rates of 0.9 and discount factors of 0.5 were used for all simulations and found to work well. The Q matrix was initialized with random positive values drawn from a uniform distribution. Also note that the first 20% of training episodes were completed by choosing actions randomly by sampling a scaled uniform distribution. This is to allow the agent to explore the space sufficiently and avoid missing optimal paths due to the random initialization of the board.

After the separate training of the modules, they were used cooperatively using the following best-average policy:

$$a(s) = \max(mean_i\big(Q_i(s,a)\big) \qquad (3)$$

Note that the training process was still continued for a small number of episodes while the modules are used together, i.e. the individual modules' Q matrices are still updated according to the previously stated law. This allows reduction in inefficient movements that might result due to conflicting objectives when the modules are first integrated.

An alternative approach which will also be explored is the use of a single module consisting of just one matrix which is concerned with achieving all three of the objectives.

Finally, an environment with litter-heavy and obstacle-heavy regions will be constructed, and we will explore the effect of region-dependent module activation.

Results

Training of the three modules

First, three modules were trained separately on a randomly generated sidewalk, with an initial randomly populated Q matrix, 20 episodes following a random policy, and 80 additional episodes following the current optimal policy for that episode.

The results for the first sidewalk ("Sidewalk 1") are plotted in Figures 1-3. In each figure, the top plot shows the first (random) episode is shown. Then, the 30th episode (which is the 10th non-random training episode, following the max-Q optimal policy) is shown. Finally, the 100th episode is shown. Note that by the end of training, the litter collection and obstacle avoidance modules do very well at their respective tasks, while neither avoiding obstacles in the former case nor attempting to pick up litter in the latter. Also note that the end-reach module walks the sidewalk in the minimum number of steps, L, by the end of training (not necessarily in a straight line, since moving forward diagonally is rewarded in the same was as moving forward in a straight line). Reporting of scores is delayed until the following section on integration of the modules.

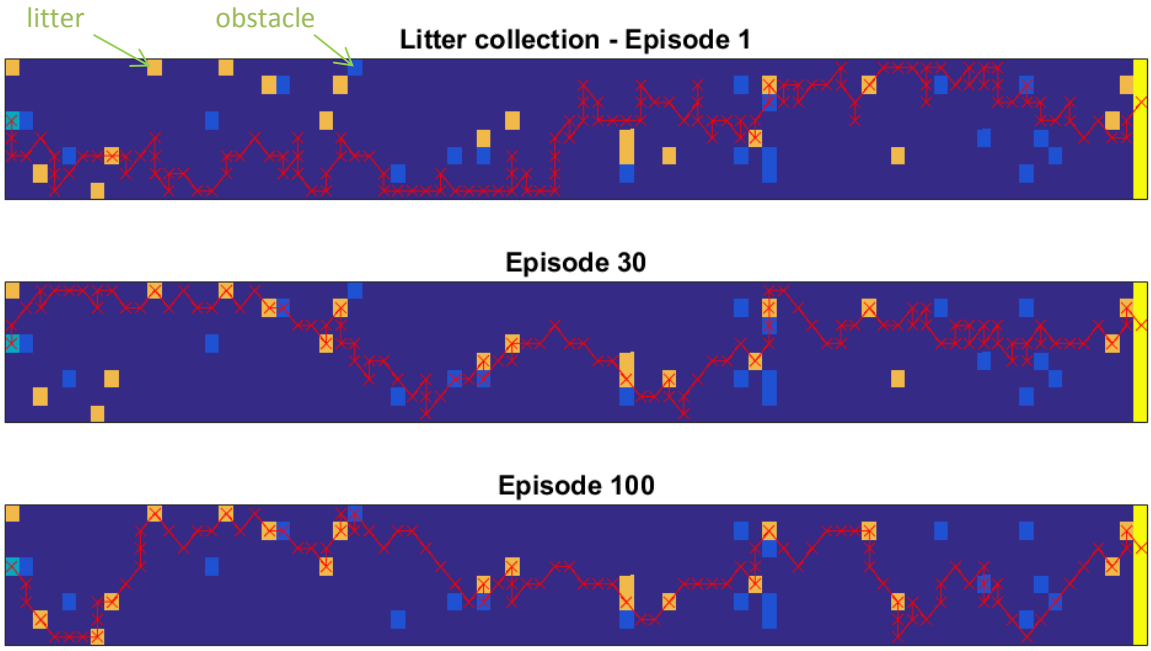

Figure 1. Litter collection module training. Litter in gold, obstacles in light blue, end in bright yellow, path in red.

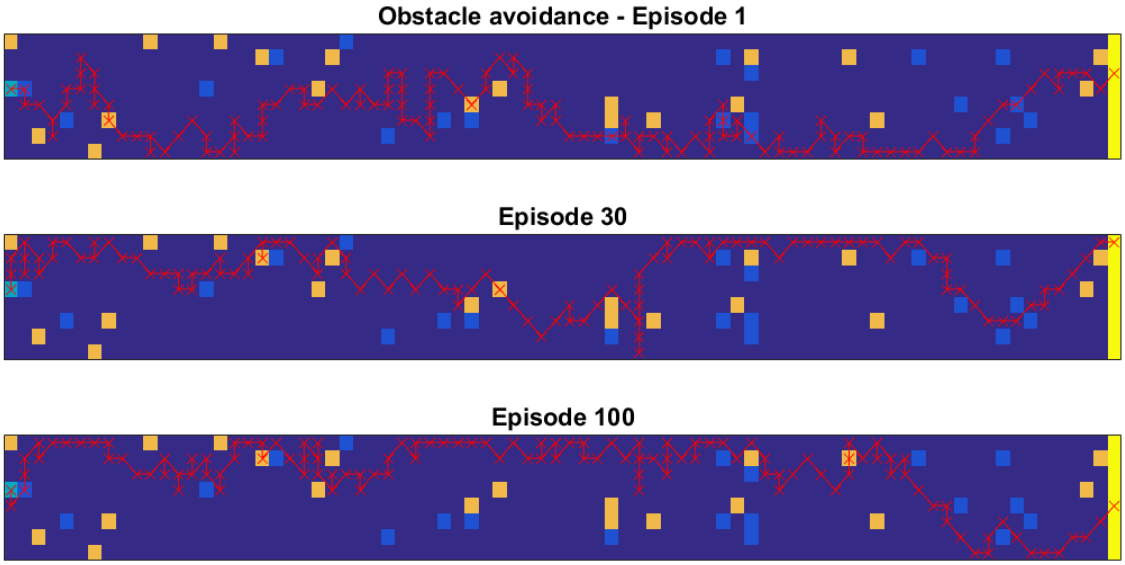

Figure 2. Obstacle avoidance module training

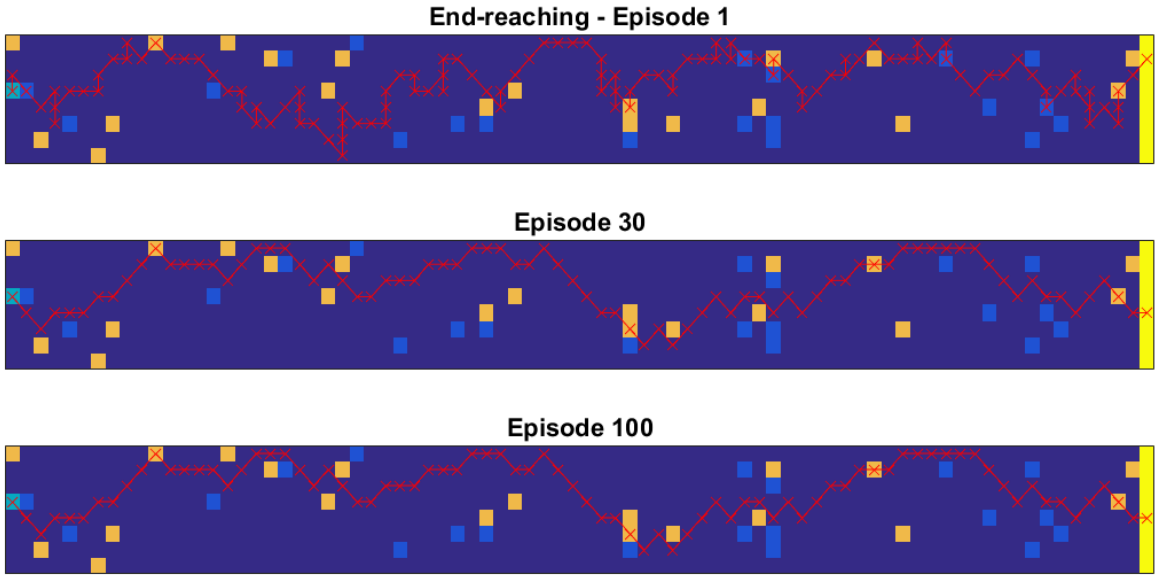

Figure 3. End-reaching module training

Integration of the modules

Next, the module was integrated and the policy at each state was set to the best *average* Q-value action (see equation 3). The agents were allowed to train for an additional 10 episodes together. In the first episode (Figure 4-top), the agent does very well in total score, but the path is somewhat tortuous due to

the competing litter collection and obstacle avoidance objectives. By the 10th trial (Figure 4-bottom), the path is more direct, and while the total score is slightly lower than in the first integrated episode, the score per step is optimized. The evolution of the score and score per step is shown in Figure 5.

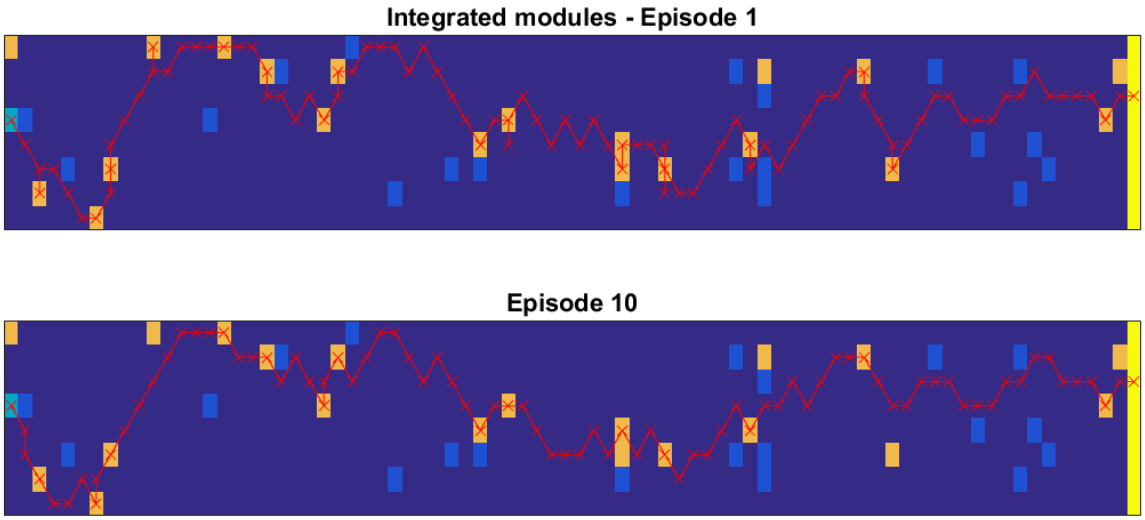

Figure 4. Integration of the three separately trained modules using best average Q-value policy

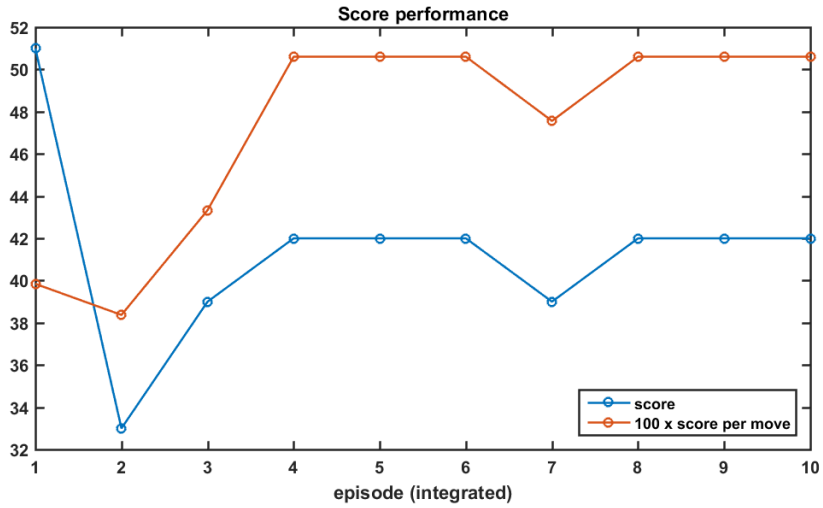

Figure 5. Score and Score per move of the integrated (cooperating) modules on Sidewalk 1. About 10 cooperative training episodes are needed for the performance to level out.

The performance of the integrated agents for two alternate, randomly generated sidewalks ("Sidewalk 2" and "Sidewalk 3") is shown in Figure 6. We can conclude that this reinforcement learning framework works well independent of particular sidewalk configurations.

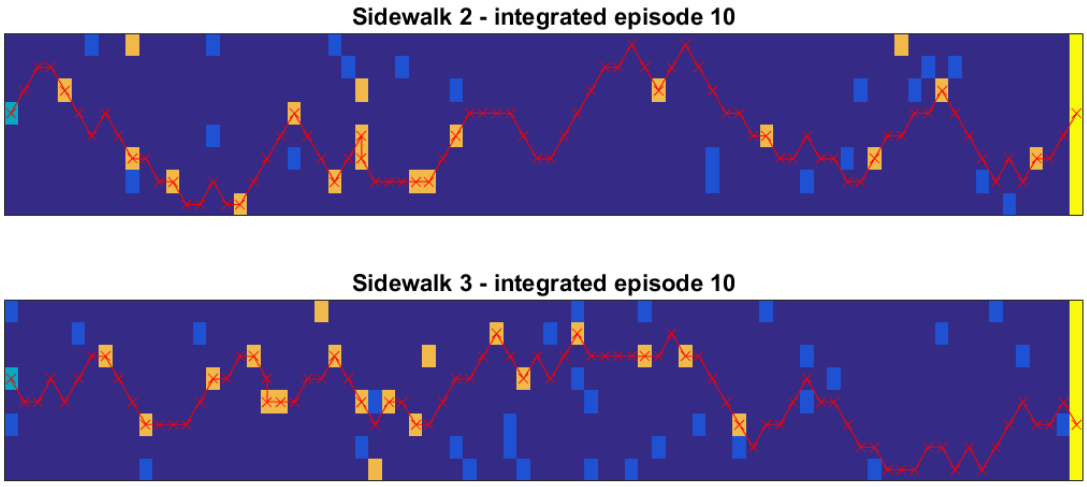

Figure 6. Two additional sidewalks

Comparison with single-module performance

An alternate approach to training separate modules for litter collection, obstacle avoidance and reaching the end is to train a single module. This is possible since the environment and objectives being considered are fairly simple. In this case, there is a single Q-matrix, which is still updated according to equation (2) but all three rewards/penalties are assigned as in Table 1 when encountered by the agent. Such an agent was trained for 100 episodes on Sidewalk 1. The reward values and all training parameters were kept the same as in the case of the separate modules.

The performance of the two approaches, as seen in Figure 7, is comparable, suggesting that for this simple exercise, a single module might be sufficient (and is less computationally demanding in terms of training). Of course, as the complexity of the environment and objectives increases, multiple modules start to make sense and may provide increased flexibility (i.e. may be turned on or off, as explored in the following section).

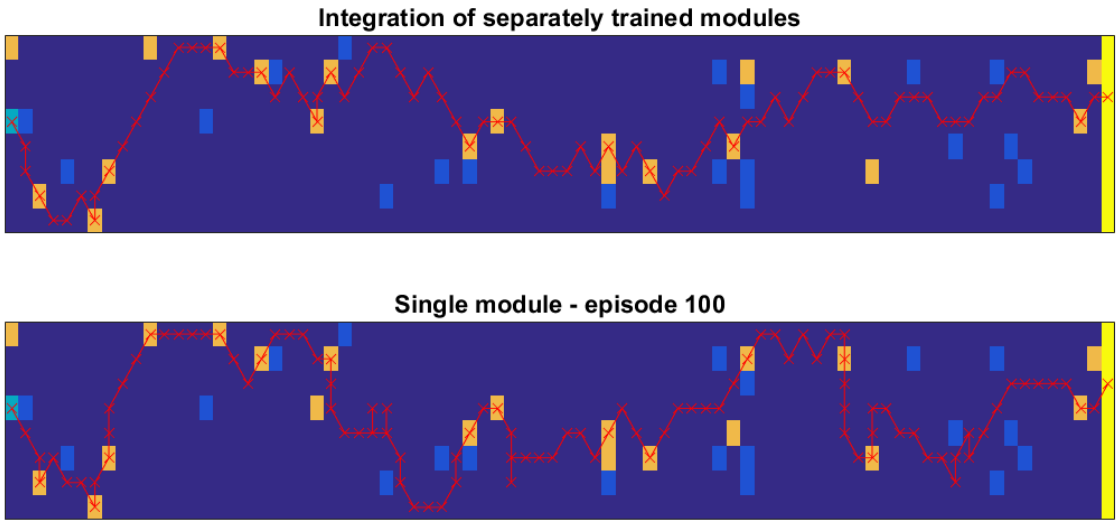

Figure 7. Comparison of integrated, separately trained modules and a single multi-objective module (Sidewalk 1): Performance is very similar for reduced computational effort when using one module.

Region-dependent module activation

Finally, Sidewalk 3 (see Figure 6-bottom) has the property that the west half of the path has a higher relative reward density, while the east half of the path has a higher relative obstacle density. To test whether region-dependent module activation can confer performance improvements, we implement the following module-activation policy taking into consideration the local environment at a given grid position:

$$\boldsymbol{For\ a\ 3\ block\ radius\ of\ feasible\ moves}:$$

$$If\ \%\ litter \geq 75\% \rightarrow\ turn\ off\ obstacle\ module$$

$$Else\ If\ \%\ litter\ \leq 25\% \rightarrow\ turn\ off\ litter\ module$$

$$Else\ If\ no\ litter\ or\ obstacle\ \rightarrow use\ previous\ mode\ module(s)$$

$$Else \rightarrow run\ all\ modules$$

Note that the end-reaching module is left on in all cases.

In Figure 8, the performance of the integrated modules (top) and the integrated modules with region-dependent activation (middle) are compared at the 10th integrated episode on Sidewalk 3. The bottom plot displays the mode of operation at a given point in the agent's path: mode 1 focuses only on litter collection and reaching the end, mode 2 focuses only on obstacle avoidance and reaching the end, and mode 3 has all modules active.

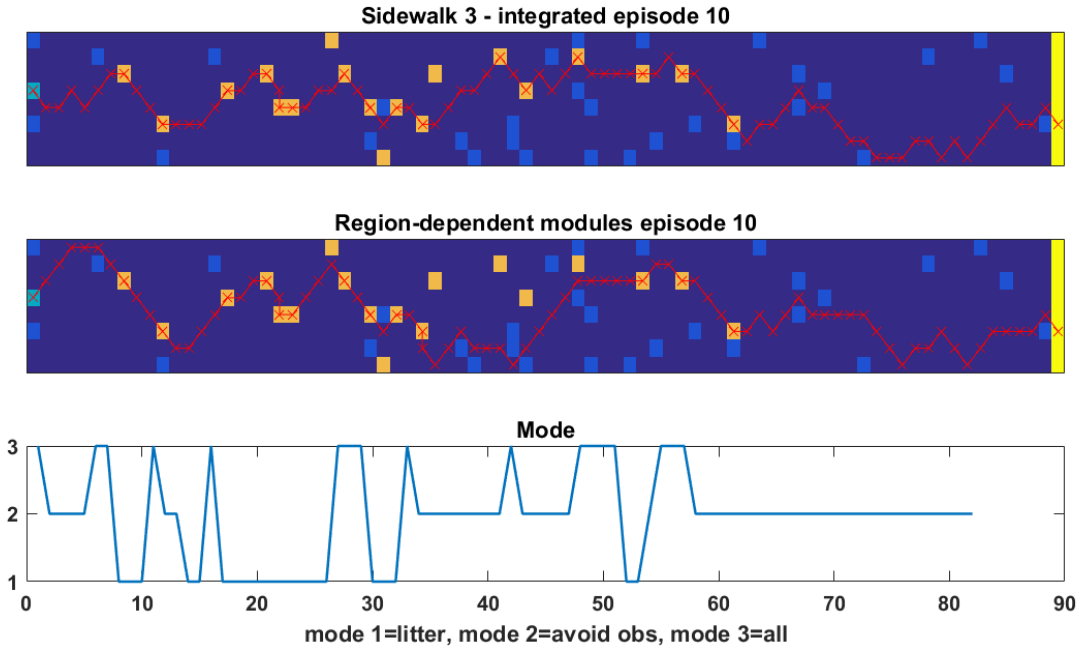

Figure 8. Region-dependent module activation leads to missed litter clean-up opportunity in the x = 35 to x = 50 region

The results of this experiment suggest that, for this environment and associated objectives, such a scheme actually leads to *poorer* performance. For much of the sidewalk, the paths are nearly identical. However, starting around x=35 in the horizontal direction, the agent enters an obstacle-heavy region and accordingly enters mode 2. As such, it is only preoccupied with obstacle avoidance for the next 15 or so moves, and misses the opportunity to move northeast and collect several pieces of litter. When all modules are working, there is enough incentive, based on the litter module's training, so that the agent does move up and collect the litter.

These results are not surprising since the environment is quite simple and the integrated module is already performing well. Also, it has been trained sufficiently long that in regions of low litter density, the Q values for that module are all about the same. In the case of a much more complex environment with a much greater number of modules with competing objectives, switching off certain modules as dictated by the local environment might still prove useful in avoiding unnecessary movements associated with unimportant modules; similarly, it might reduce the length of training necessary to achieve the same results if the Q matrices are initialized randomly.

Conclusions

To conclude, a simple Q-learning implementation is sufficient, with enough training and reasonable learning parameters, to accomplish the task of moving an agent across a sidewalk while collecting litter and avoiding obstacles. The success of the method is not dependent on the specific sidewalk

configuration. A single module focusing on all three objectives may perform similarly to one in which single-objective modules cooperate and the best-average action is taken. On the other hand, switching modules on and off per the requirements of the local environment was not seen to be beneficial and rather led to missed opportunity for this quite simple environment. This approach might prove useful in more complex situations, however.