Stanford | ENGINEERING
Computer Science

Multi-Stop Navigation with Congestion

Alexander Toews, Nivedita Rahurkar, Xuyi Guo {artoews, rahurkar, xuyguo}@stanford.edu

Problem Description

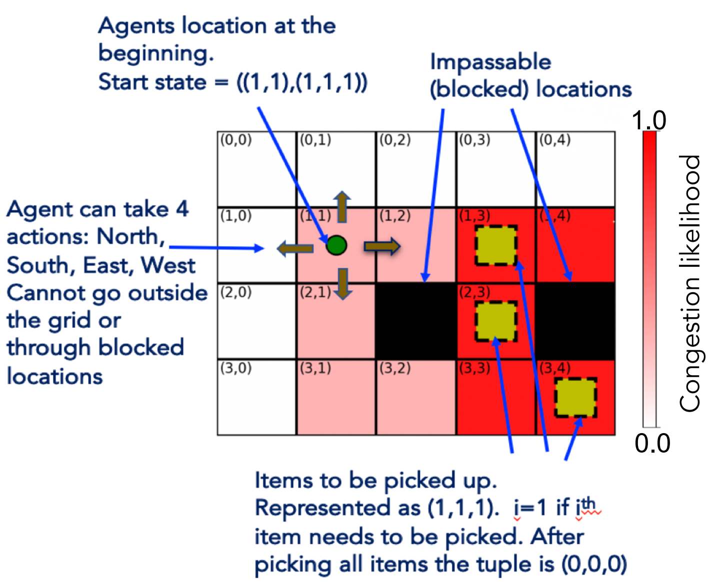
How do you go about collecting groceries in a busy supermarket? What is the best route for delivering packages during rush hour? We study the familiar problem of finding an efficient path through many checkpoints in a chaotic, congested world.

We investigate policies for an agent to navigate through multiple unordered checkpoints in a time-efficient manner. Random congestion complicates the problem.

Model

We model the world as a Markov Decision Process where we discretize both time and space.

State: agent location and list of remaining items.
Action: *attempt* to move to an adjacent square. The agent will not move if the destination happens to be congested at that time.
Reward: singular positive reward for collecting all items and returning to the start. The reward is discounted by time so as to favor faster routes (γ=0.99).

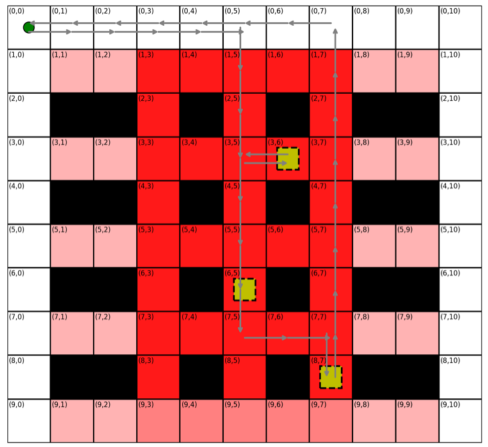


Congestion: represented as independent Bernoulli random variables at each grid space. Any attempt to move to grid space j fails with probability p_j. <u>Congestion probabilities p_j are NOT known a priori.</u>

Methods

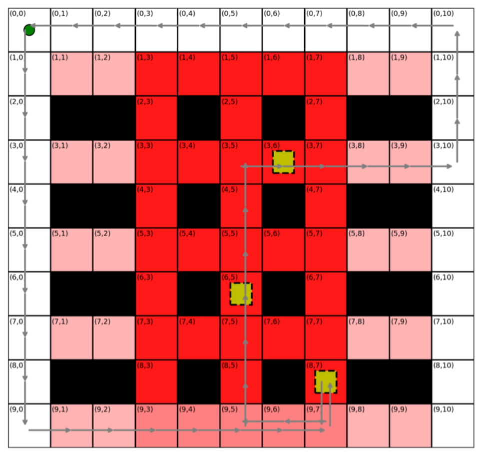
➢ **Baseline:** greedy ordering and basic search.

1. Identify nearest item by Manhattan distance.
2. Plan route to item *ignoring congestion* by running Uniform Cost Search on a congestion-free map.
3. Iterate.



➢ **Oracle**

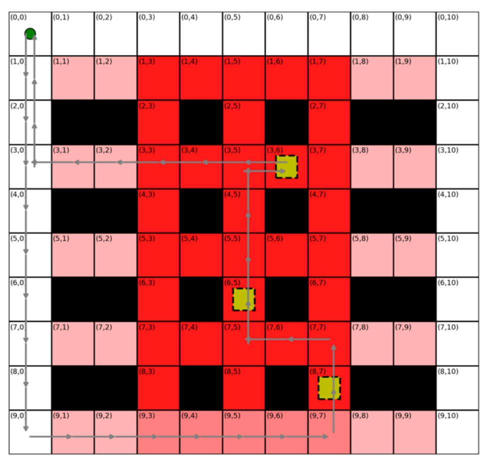
Value iteration *peeking at congestion probabilities.* Final converged policy is <u>optimal</u>.



$$V_{\text{opt}}^{(t)}(s) \leftarrow \max_{a\in\text{Actions}(s)} \sum_{s'} T(s,a,s')[\text{Reward}(s,a,s') + \gamma V_{\text{opt}}^{(t-1)}(s')]$$

➢ **Q-Learning**

Agent learns about the environment over many trials. During learning phase, ϵ-greedy policy is used to balance state exploration against exploiting what the agent already knows.

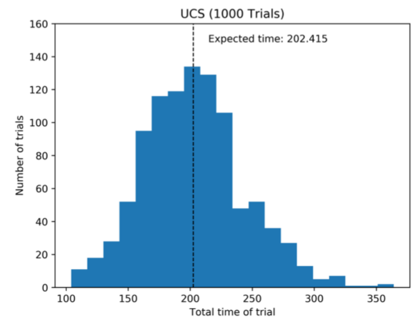


On each (s, a, r, s'):

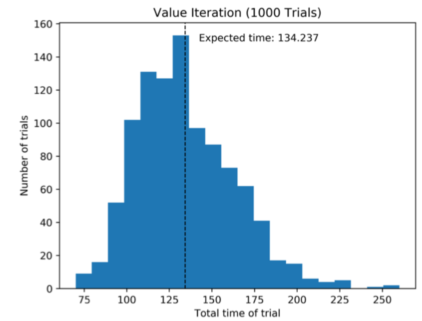
$$\hat{Q}_{\text{opt}}(s,a) \leftarrow (1-\eta)\hat{Q}_{\text{opt}}(s,a) + \eta(r + \gamma \max_{a'\in\text{Actions}(s')} \hat{Q}_{\text{opt}}(s',a'))]$$

Results

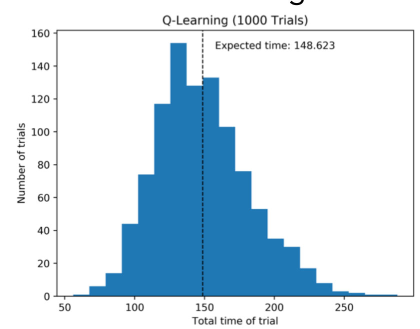
Baseline



Oracle



Q-Learning



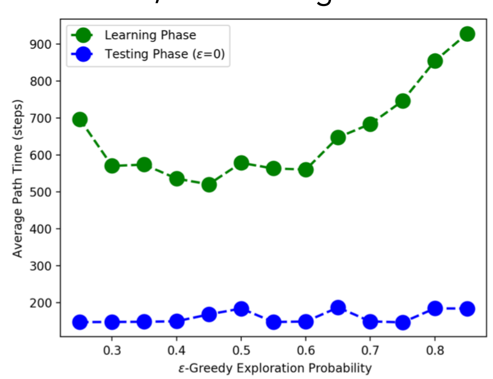
➢ **Baseline:** Expected time of 202

➢ **Oracle:** 180 iterations, expected time of 134

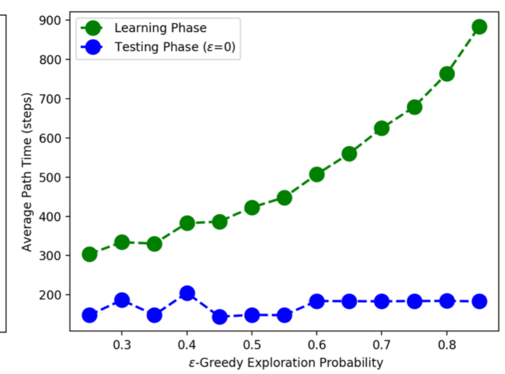
➢ **Q-Learning:** Average learning trial time is 514, expected test trial time is 148

Analysis

1,000 Learning Trials



10,000 Learning Trials



Epsilon-Greedy: Graphs comparing the average path times during the learning and test phases of Q-Learning when the number of learning trials is 1,000 (left) and 10,000 (right). Testing different values of epsilon we find ϵ = 0.5 strikes a good balance between learning time and testing time.

Discussion

In future work, we would like to make the problem more challenging by increasing the number of items and introducing a more realistic model of congestion. We would also be interested in exploring features that improve the efficiency of Q-Learning.