Resource Allocation and Planning for Automated Trash Collection

Robert Moss

CS221—Artificial Intelligence: Principles and Techniques, Stanford University

Stanford
Computer Science

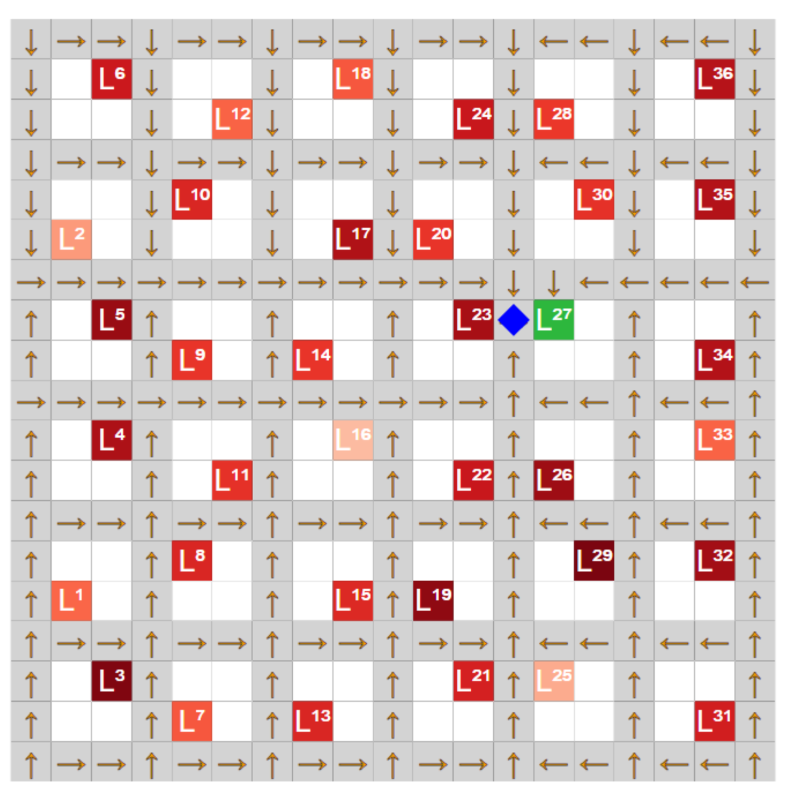
Motivation

- Reduce frequency of **unnecessary garbage truck visits**
  - Limiting emissions output of garbage trucks
  - Saving cost on gas
- Optimize ***when*** to collect trash and ***who*** needs collection
- Solve a **resource allocation** problem in a **dynamic setting**
  - Utilizing online methods to solve in real-time

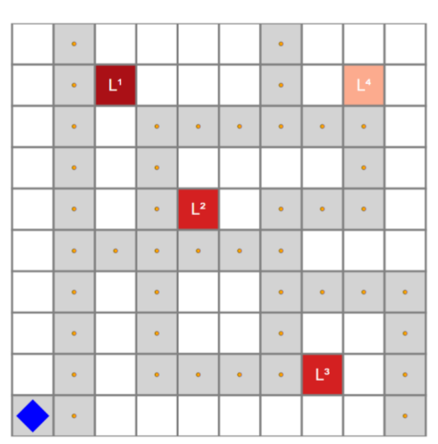
Problem Description

Environment

- Dynamic environment accumulates trash per day (time step)
- Trash is accumulated at certain locations
- Environment modeled as a **10x10** and **19x19** sq. mile grid city
  - **Trash site locations:** Locations to collect trash
    - **Fill-level:** [0-100] current trash accumulation level
    - **Fill-rate:** [1-10] accumulation rate per day (time step)
  - **Roads:** (x, y) coordinates, restricting agent's travel
- The agent is the garbage truck (the blue diamond)
- Trash locations (red/green) are adjacent to roads (gray)
- Trash locations have +/- reward relative to their fill-level



19x19
36 Trash Locations (one per-block)



10x10
4 Trash Location

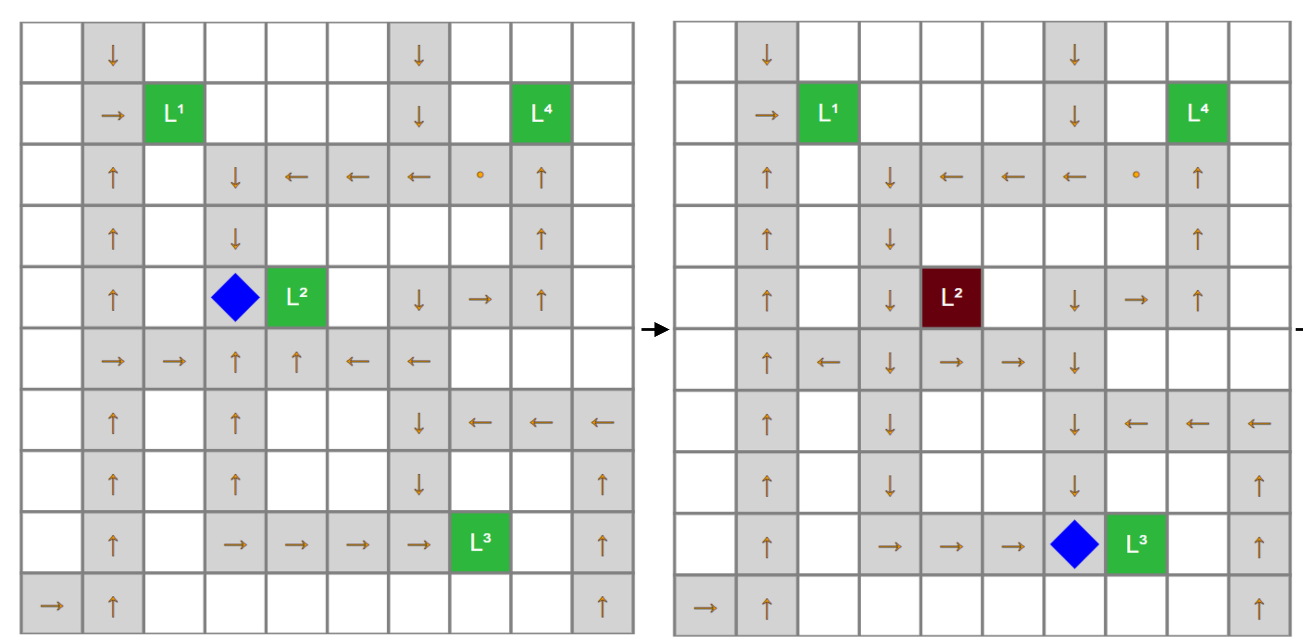
Legend
- **Agent** (blue diamond)
- **Roads** (gray)
- **Trash site locations** (red/green)
  - **Fill-level** (indicated by color)
    - **Red** = low fill-level, negative reward
      - Lighter red means reward closer to 0
    - **Green** = high fill-level, positive reward
- **Arrows/Dots** (orange)
  - Indicates policy action (dot = `nothing`)

Model

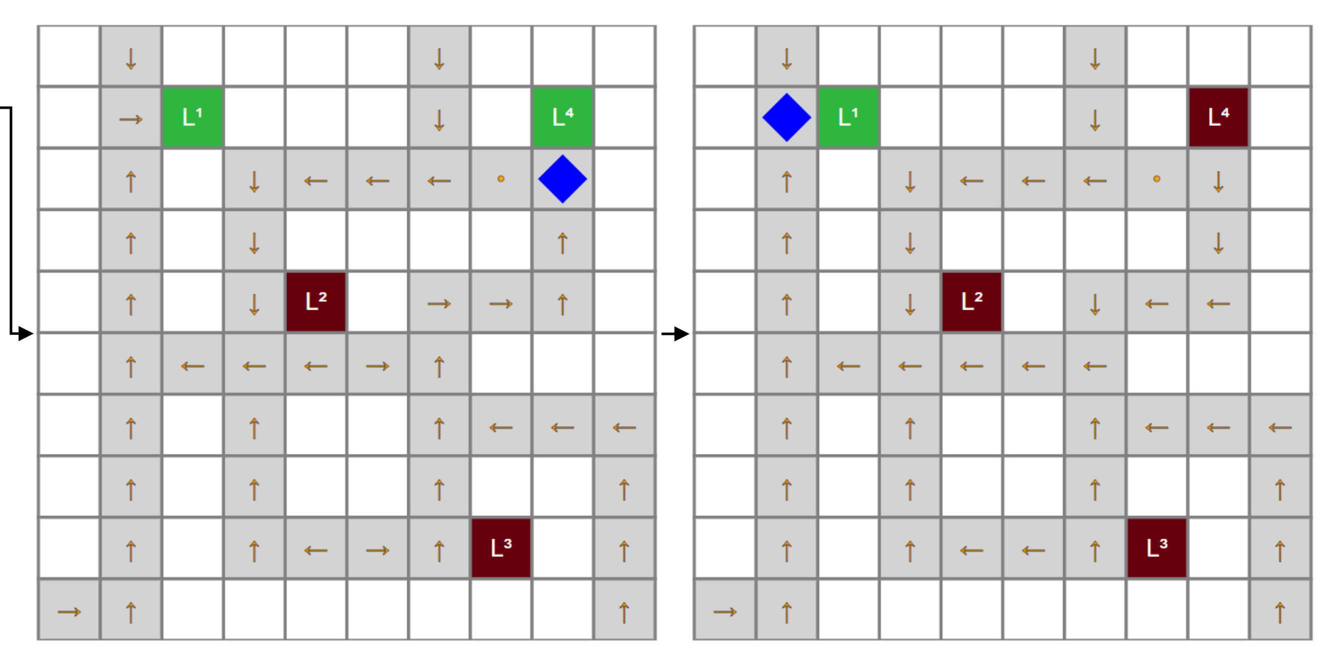
Markov Decision Process (MDP)

Trash problem modeled as a Markov Decision Process

- **State space:**
  - X-coordinate in grid [1-10] or [1-19]
  - Y-coordinate in grid [1-10] or [1-19]
    - (x, y) limited to roads and trash locations
    - State space size = 48 for the 10x10 grid
    - State space size = 316 for the 19x19 grid
- **Action space:**
  - *nothing, up, down, left, right*
    - *nothing* waits for more environment information
- **Transition function:**
  - Deterministic: agent can freely travel the roads
  - *nothing* stays in current state with probability 1

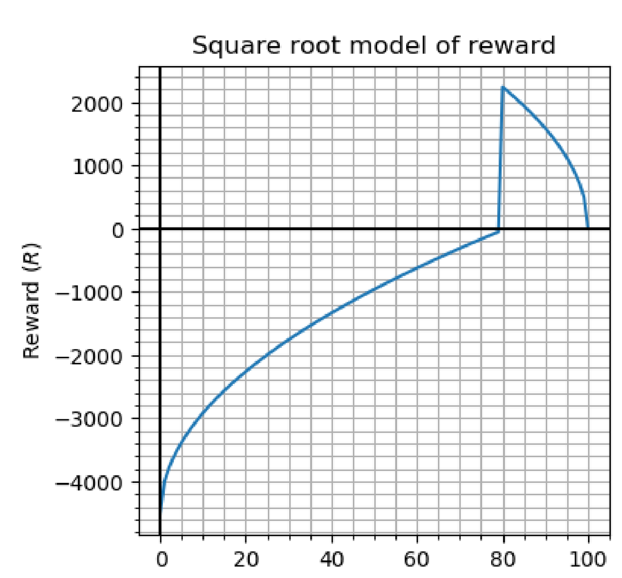


$t = 70$ (Location 2) $t = 70$ (Location 3)



$t = 70$ (Location 4) $t = 70$ (Location 1)

- **Reward function:**
  - Piecewise square root function of fill-level: $R(l)$
    - Cross-point at fill-level of 80 (threshold)
  - Encourages collecting trash near threshold
  - Penalizes collecting trash relative to fill-level
  - Reward decreases as fill-level gets close to full



$$R(l) = \begin{cases} 5e3\left(\sqrt{\dfrac{l}{\text{MAX\_FILL}}} - 0.9\right) & l < \text{threshold}, \\ 5e3\left(\sqrt{\dfrac{\text{MAX\_FILL} - l}{\text{MAX\_FILL}}}\right) & \text{otherwise.} \end{cases}$$

Simulation Approach/Algorithms

Simulation step

```
for t in 1:MAX_TIME # set to 98 days or 14 weeks (baseline is weekly)
    • Propagate environment forward # i.e. accumulate trash
    • Solve MDP to generate a policy # i.e. optimal collection path
    • Follow the policy to collect trash # i.e. collect rewards
end
```

Algorithms

- **Value iteration, VI** (model-based):
  - Full knowledge of state-space
  - 10,000 max iterations (converges around 200)
  - Bellman residual set to 1e-6 (convergence threshold)
  - Guaranteed to converge to the *optimal policy*
- **Q-Learning** (model-free):
  - Does not learn reward function directly (model-free)
  - No exploration strategy (e.g., no epsilon greedy)
  - Converges to optimal policy
- **Sarsa-λ** (model-free):
  - Eligibility traces seemed appropriate for efficient policies
  - Investigated, but deemed too slow and unnecessary given value iteration converges to optimal policy

Challenges/Assumptions

Efficiency

- Requires solving for optimal policy in real-time
  - Selection of algorithms are limited
  - Obsolete challenge as the policy is learned daily
  - Larger state-spaces could degrade value iteration

Assumptions

- Fill-level is reported precisely to the agent (via sensors)
- Garbage truck has unlimited fill capacity
  - Collects trash at each site ready for pick up (per day)
- Locations are adjacent to roads

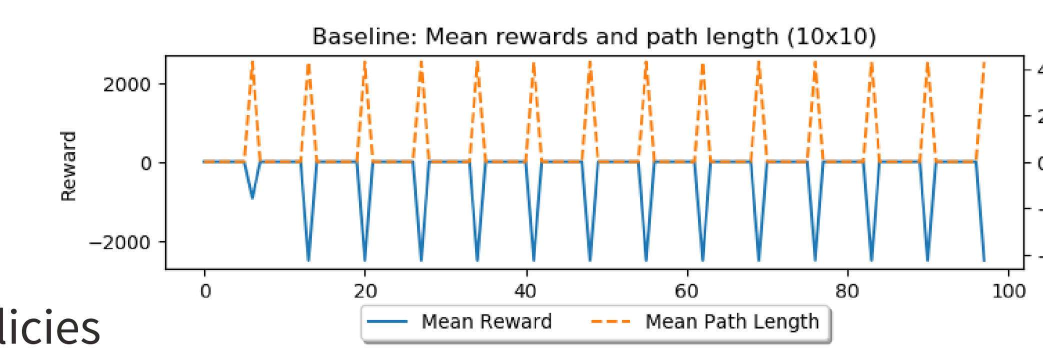
Analysis/Results

Value iteration vs. Q-learning

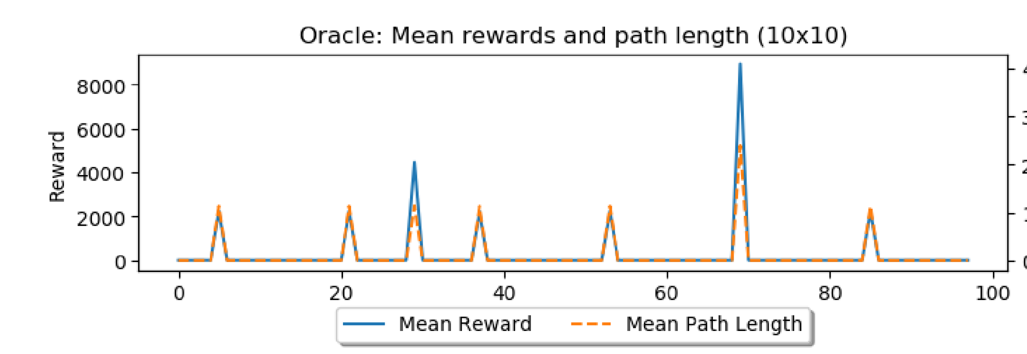
- Value iteration and Q-learning converged to optimal policy
  - Q-learning policy *identical* to value iteration, yet *slower*
    - Q-learning about 15x slower than value iteration
    - Value iteration took 1.8 seconds to solve all 98 days
  - Thus, further results are limited to **value iteration**

Comparison Results (10x10 Grid)

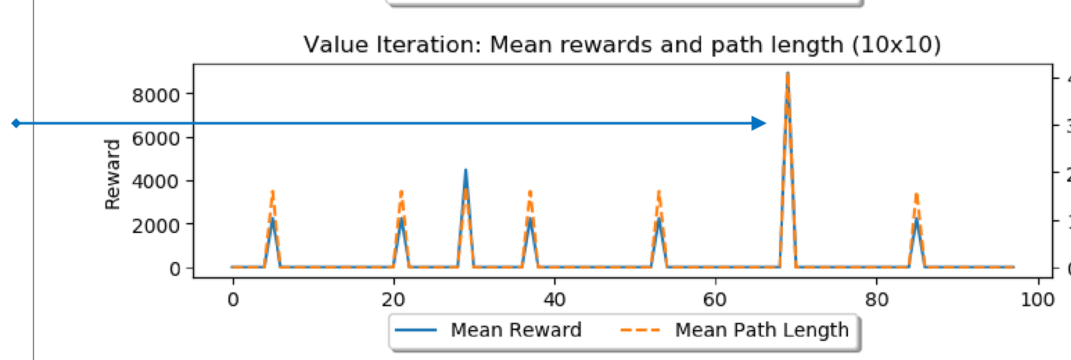
- **Baseline** picks up trash at every location *once a week*
  - *Manhattan distance* used as the Baseline path length
- **Oracle** picks up trash exactly when threshold is met
  - *Euclidean distance* used as the Oracle path length



Mean reward
-340.25
Mean path length
6.14 miles



Mean reward
250.99
Mean path length
0.94 miles



Mean reward
250.99
Mean path length
1.41 miles

Emissions Results (19x19 Grid)

- **Baseline:** $\mu(Reward) = -344.73$, $\mu(Path) = 93.14$ miles
- **Oracle:** $\mu(Reward) = 2286.56$, $\mu(Path) = 11.72$ miles
- **Value iteration:** $\mu(Reward) = 2286.56$, $\mu(Path) = 15.52$ miles

- Value iteration performs as effectively as the Oracle in terms of collecting trash when the threshold is met (i.e. rewards are identical)

- Based on garbage truck CO_2 emissions data and MPG from [1] and current CA diesel gas prices from [2]:
  - **Value iteration** compared to **Baseline**:
    - Reduces fuel consumption and CO_2 emissions by **83%**

References

1. Gurdas S. Sandhu, H. Christopher Frey, Shannon Bartelt-Hunt & Elizabeth Jones (2015) In-use activity, fuel use, and emissions of heavy-duty diesel roll-off refuse trucks, *Journal of the Air & Waste Management Association*, 65:3, 306-323, DOI: 10.1080/10962247.2014.990587
2. https://www.eia.gov/dnav/pet/pet_pri_gnd_dcus_sca_w.htm