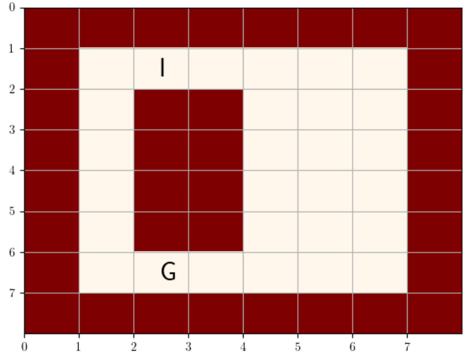
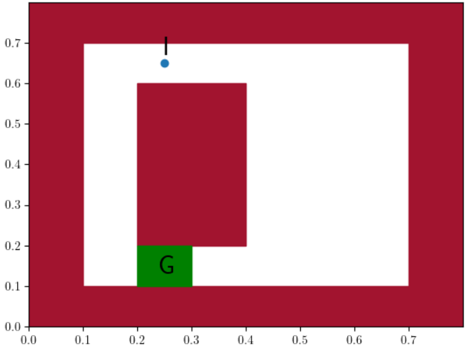
TEXAS
The University of Texas at Austin

Chance-Constrained Motion Planning

Nikitha Gollamudi
Apurva Patil

Problem Formulation





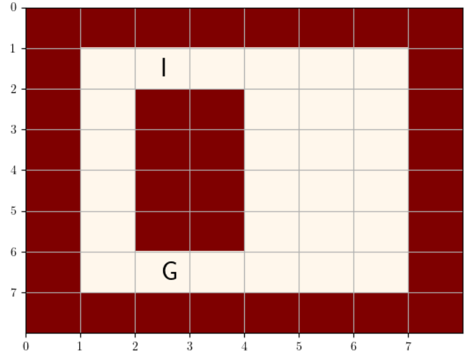
Problem 1 (Risk-constrained motion planning problem):

$$\arg\max_{\pi} \mathbb{E}_{\pi}\left[U_{t_0} \mid S_{t_0} = I\right] \quad \text{s.t.} \;\; P_{fail} < \Delta.$$

Problem 2 (Risk-minimizing motion planning problem):

$$\arg\max_{\pi} \left\{\mathbb{E}_{\pi}\left[U_{t_0} \mid S_{t_0} = I\right] - \eta \cdot P_{fail}\right\}.$$

Problem Formulation



Transition Dynamics

Unsafe states: $\mathcal{S}_u$

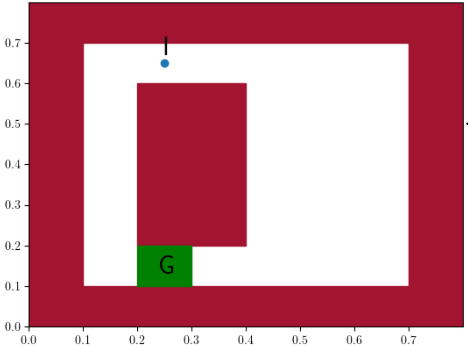
Safe states: $\mathcal{S}_s$

Terminal states: $\mathcal{S}_u + G$

Action space: $\mathcal{A} = \{N, W, S, E\}.$

Action N:

$$P(S^N \mid S, N) = 0.9$$
$$P(S^{NW} \mid S, N) = 0.05$$
$$P(S^{NE} \mid S, N) = 0.05.$$



Transition Dynamics

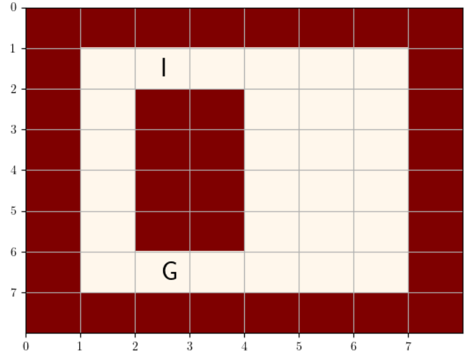
Action E:

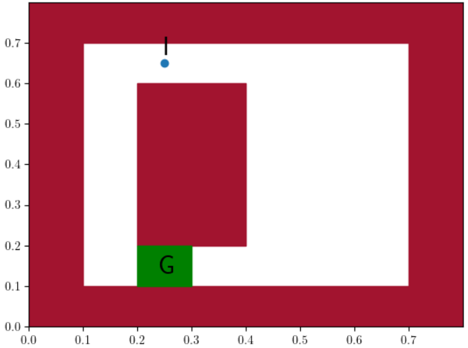
$$x_1(t+1) = x_1(t) + \beta_1 + n_1(t), \qquad n_1(t) \sim \mathcal{N}\left(0, \sigma^2\right),$$
$$x_2(t+1) = x_2(t) + n_2(t), \qquad n_2(t) \sim \mathcal{N}\left(0, \sigma^2\right).$$

Action N:

$$x_1(t+1) = x_1(t) + n_1(t), \qquad n_1(t) \sim \mathcal{N}\left(0, \sigma^2\right),$$
$$x_2(t+1) = x_2(t) + \beta_2 + n_2(t), \qquad n_2(t) \sim \mathcal{N}\left(0, \sigma^2\right).$$

Problem Formulation





Problem 2 (Risk-minimizing motion planning problem):

$$\arg\max_{\pi} \left\{ \mathbb{E}_{\pi}\left[U_{t_0} \mid S_{t_0} = I\right] - \eta \cdot P_{fail} \right\}.$$

$$P_{fail} = \mathbb{E}_{\pi}\left[\mathbb{1}_{S_{t_f} \in \mathcal{S}_u} \mid S_{t_0} = I\right]$$

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi}\left[U_{t_0} - \eta \cdot \mathbb{1}_{S_{t_f} \in \mathcal{S}_u} \mid S_{t_0} = I\right]$$
$$= \arg\max_{\pi} \mathbb{E}_{\pi}\left[U'_{t_0} \mid S_{t_0} = I\right].$$

where,

$$U'_{t_0} = U_{t_0} - \eta \cdot \mathbb{1}_{S_{t_f} \in \mathcal{S}_u}.$$

Risk Estimation of π^*

$$v_{\pi^*}(I) = \mathbb{E}_{\pi^*}\left[U_{t_0} \mid S_{t_0} = I\right].$$

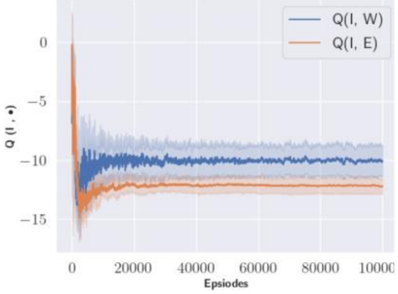
$$\lambda = 0, \quad R^G = 0, \quad \eta = 1.$$

$$U_{t_0} = \mathbb{1}_{S_{t_f} \in \mathcal{S}_u}$$

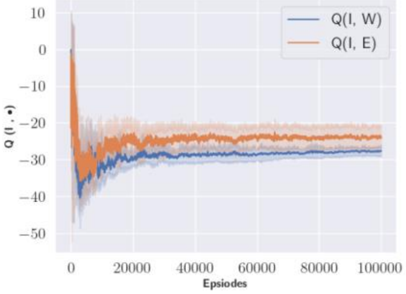
$$v_{\pi^*}(I) = \mathbb{E}_{\pi^*}\left[\mathbb{1}_{S_{t_f} \in \mathcal{S}_u} \mid S_{t_0} = I\right].$$

$$P_{fail} = \mathbb{E}_{\pi}\left[\mathbb{1}_{S_{t_f} \in \mathcal{S}_u} \mid S_{t_0} = I\right]$$

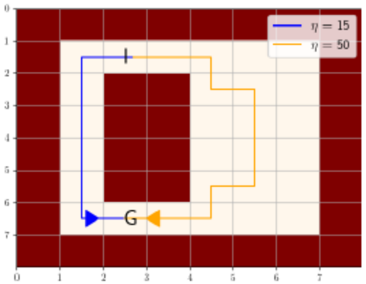
Discrete State Space: Policy Synthesis



(a) $\eta = 15$



(b) $\eta = 50$



No noise trajectories for $\eta = 15$ and $\eta = 50$

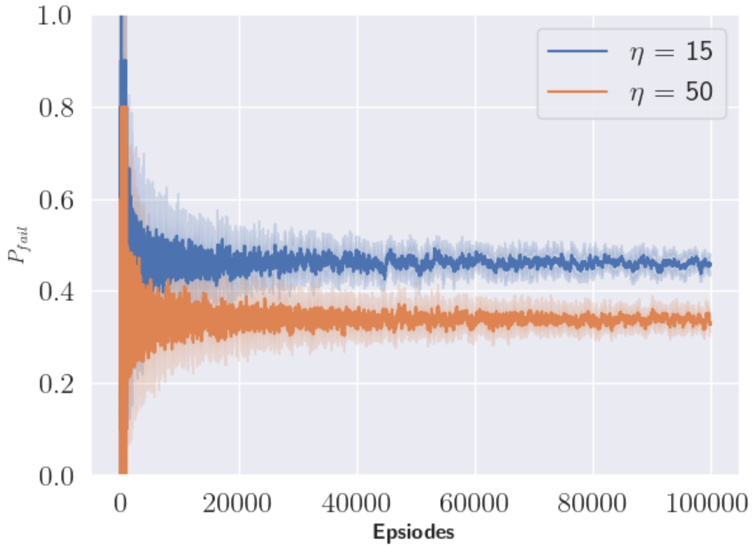
Algorithm 1: Modified Q-learning

Parameters: step size $\alpha \in (0,1]$, $\epsilon > 0$, rewards $\lambda, R^G, \eta > 0$, $\gamma = 1$.

Initialize $Q(s,a)$, $\forall\ s \in \mathcal{S}, a \in \mathcal{A}$, arbitrarily except that $Q(s,.) = 0$, $\forall\ s \in \mathcal{S}_u$, and $Q(G,.) = 0$.
Loop for each episode:
- $S_{t_0} \leftarrow I$
- **Loop for each step of the episode:**
  - Choose A_t from S_t using ϵ-greedy policy derived from Q
  - Take action A_t and observe R_{t+1}, S_{t+1}
  - **if** $S_{t+1} = G$ **then**
    - $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + R^G - Q(S_t, A_t)]$
    - **break**
  - **end**
  - **else if** $S_{t+1} \in \mathcal{S}_u$ **then**
    - $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} - \eta - Q(S_t, A_t)]$
    - **break**
  - **end**
  - **else**
    - $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)]$
    - $S_t \leftarrow S_{t+1}$
  - **end**
- **end**

end

Discrete State Space: Risk Estimation of π^*



Convergence of P_{fail} for $\eta = 15$ and $\eta = 50$ with one standard deviation.

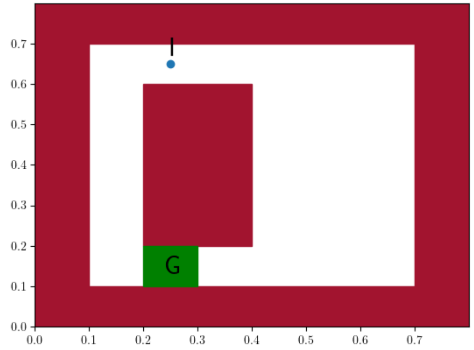
Algorithm 2: Modified TD(0)

Input : π^* synthesized from Algorithm 1
Parameters: step size $\alpha \in (0, 1]$, $\gamma = 1$

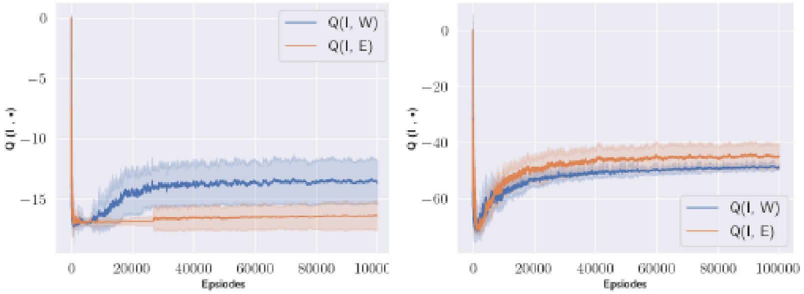
Initialize $V(s)$, $\forall\ s \in \mathcal{S}$ arbitrarily except that $V(s) = 0$, $\forall\ s \in \mathcal{S}_u$, and $V(G) = 0$.
Loop for each episode:
  $S_{t_0} \leftarrow I$
  Loop for each step of the episode:
    $A_t \leftarrow \pi^*(S_t)$
    Take action A_t and observe S_{t+1}
    if $S_{t+1} = G$ **then**
      $V(S_t) \leftarrow V(S_t) - \alpha[V(S_t)]$
      break
    end
    else if $S_{t+1} \in \mathcal{S}_u$ **then**
      $V(S_t) \leftarrow V(S_t) + \alpha[1 - V(S_t)]$
      break
    end
    else
      $V(S_t) \leftarrow V(S_t) + \alpha[\gamma V(S_{t+1}) - V(S_t)]$
      $S_t \leftarrow S_{t+1}$
    end
  end
end

Continuous State Space: Function Approximation

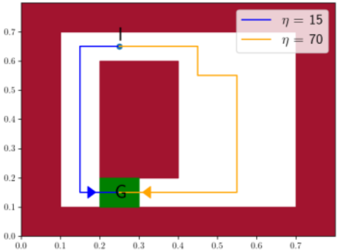
- Linear function approximation
- Polynomial features: $\boldsymbol{x}(s) = \begin{bmatrix} 1 & x_1 & x_2 & x_1 x_2 \end{bmatrix}^T$
- Tile coding



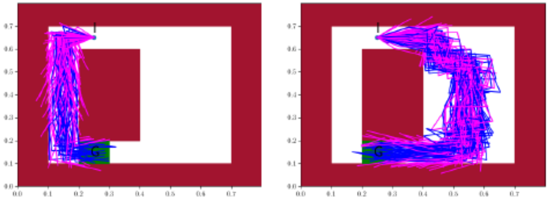
Continuous State Space: Policy Synthesis



(a) $\eta = 15$ (b) $\eta = 70$



No noise trajectories for $\eta = 15$ and $\eta = 50$



(a) $\eta = 15$ (b) $\eta = 70$

Fig. 7. 100 sample trajectories generated using π^* for $\eta = 15$ and $\eta = 70$. The trajectories are color-coded; magenta paths go into the unsafe region $\mathcal{S}_u$, while blue paths go to the goal region $\mathcal{S}_G$.

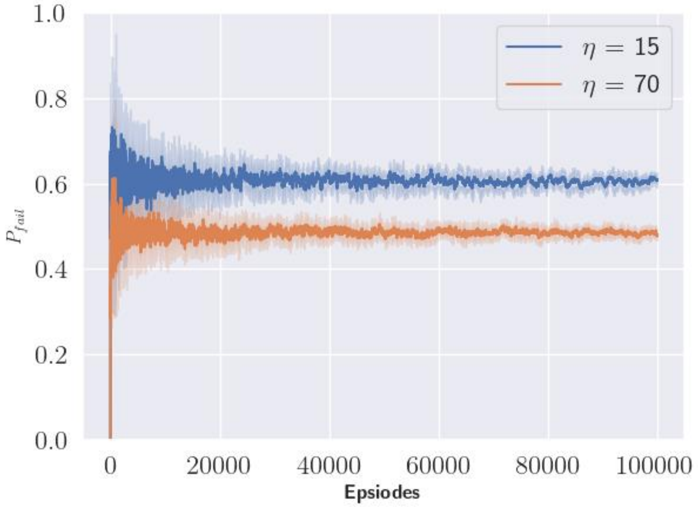
Algorithm 3: Modified semi-gradient SARSA

Input : a differentiable linear action-value function parameterization $\hat{q}(s, a, \boldsymbol{w}) = \boldsymbol{w}^T \boldsymbol{x}(s, a)$

Parameters: step size $\alpha \in (0, 1]$, $\epsilon > 0$, rewards $\lambda, R^G, \eta > 0$, $\gamma = 1$.

Initialize value-function weights $\boldsymbol{w}_{t_0} \in \mathbb{R}^2$ arbitrarily (e.g.. $\boldsymbol{w}_{t_0} = \mathbf{0}$)
Loop for each episode:
    $S_{t_0} \leftarrow I$
    Choose A_{t_0} from S_{t_0} using ϵ-greedy policy derived from $\hat{q}(S_{t_0}, ., \boldsymbol{w}_{t_0})$
    Loop for each step of the episode:
        Take action A_t and observe R_{t+1}, S_{t+1}
        if $S_{t+1} \in \mathcal{S}_G$ **then**
            $\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t + \alpha[R_{t+1} + R^G - \hat{q}(S_t, A_t, \boldsymbol{w}_t)]\boldsymbol{x}(S_t, A_t)$
            break
        end
        else if $S_{t+1} \in \mathcal{S}_u$ **then**
            $\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t + \alpha[R_{t+1} - \eta - \hat{q}(S_t, A_t, \boldsymbol{w}_t)]\boldsymbol{x}(S_t, A_t)$
            break
        end
        else
            Choose A_{t+1} from S_{t+1} using ϵ-greedy policy derived from $\hat{q}(S_{t+1}, ., \boldsymbol{w}_t)$
            $\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t + \alpha[R_{t+1} + \gamma\,\hat{q}(S_{t+1}, a_{t+1}, \boldsymbol{w}_t) - \hat{q}(S_t, A_t, \boldsymbol{w}_t)]\boldsymbol{x}(S_t, A_t)$
            $S_t \leftarrow S_{t+1}$
            $A_t \leftarrow A_{t+1}$
        end
    end
end

Continuous State Space: Risk Estimation of π^*



Convergence of P_{fail} for $\eta = 15$ and $\eta = 50$ with one standard deviation.

Algorithm 4: Modified semi-gradient TD(0)

Input : π^* synthesized from Algorithm 3, a differentiable linear state-value function parameterization $\hat{v}(s, \boldsymbol{w}) = \boldsymbol{w}^T \boldsymbol{x}(s)$

Parameters: step size $\alpha \in (0, 1]$, $\gamma = 1$

Initialize value-function weights $\boldsymbol{w}_{t_0} \in \mathbb{R}^2$ arbitrarily (e.g., $\boldsymbol{w}_{t_0} = \boldsymbol{0}$)
Loop for each episode:
  $S_{t_0} \leftarrow I$
  Loop for each step of the episode:
    $A_t \leftarrow \pi^*(S_t)$
    Take action A_t and observe S_{t+1}
    if $S_{t+1} \in \mathcal{S}_G$ **then**
      $\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t - \alpha\, \hat{v}(S_t, \boldsymbol{w}_t)\, \boldsymbol{x}(S_t)$
      break
    end
    else if $S_{t+1} \in \mathcal{S}_u$ **then**
      $\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t + \alpha[1 - \hat{v}(S_t, \boldsymbol{w}_t)]\boldsymbol{x}(S_t)$
      break
    end
    else
      $\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t + \alpha[\gamma \hat{v}(S_{t+1}, \boldsymbol{w}_t) - \hat{v}(S_t, \boldsymbol{w}_t)]\boldsymbol{x}(S_t)$
      $S_t \leftarrow S_{t+1}$
    end
  end
end

Nikitha Gollamudi: nikithag@utexas.edu
Apurva Patil: apurvapatil@utexas.edu