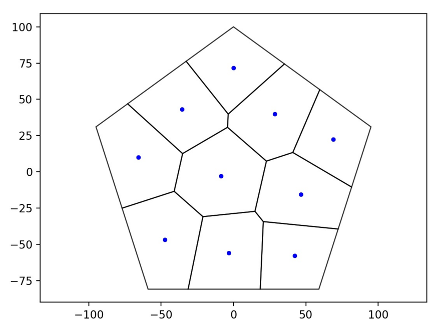

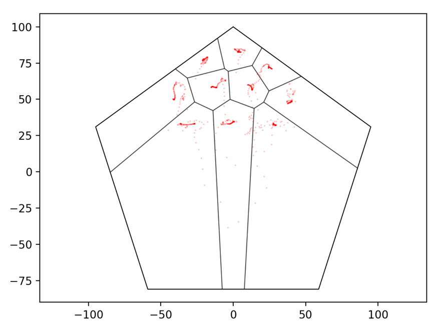

Multi-Agent Area Coverage Control using Reinforcement Learning

Adekunle Adepegba, Suruz Miah, Davide Spinello
Presented by Simon Hu

UC San Diego
JACOBS SCHOOL OF ENGINEERING
Electrical and Computer Engineering

Problem Statement

Consider a group of N homogeneous agents moving in a compact environment $\Omega \subset \mathbb{R}^2$ where the dynamics of the agent are given by $\dot{p_i} = g(p_i, u_i)$ where $p_i = (x_i, y_i) \in \mathbb{R}^2$ is the agent position and $u_i = (u_{x_i}, u_{y_i})$ represents the control input. The goal is to find a configuration of agent positions $p = (p_1, p_2, \ldots, p_N$ such that the cost index

$$\mathcal{H}(p,t) = \int_\Omega \max_{i=1,\ldots,N} f_i(\|p_i - q\|)\phi(q,t)\,\mathrm{d}q$$

is maximized.

Voronoi Partitions

The Voronoi partition of Ω is given by $\mathcal{V} = \bigcup \mathcal{V}_i$ where each $\mathcal{V}_i$ is given by

$$\left\{ q \in \Omega \mid \|q - p_i\| \leq \left\|q - p_j\right\|, \forall j \neq i \right\}.$$

The mass $m_{\mathcal{V}_i}$ and center of mass $c_{\mathcal{V}_i}$ of each V_i are given by

$$m_{\mathcal{V}_i} = \int_{\mathcal{V}_i} \phi\,\mathrm{d}q, c_{\mathcal{V}_i} = \frac{1}{m_{\mathcal{V}_i}} \int_{\mathcal{V}_i} q\phi\,\mathrm{d}q$$

Then the cost index $\mathcal{H}$ can be rewritten as

$$\mathcal{H}(p,t) = \sum_{i=1}^{n} \int_{\mathcal{V}_i} f_i(\|p_i - q\|)\phi(q,t)\,\mathrm{d}q.$$

[Cortés et al., 2004] showed that the optimal partition of Ω that maximizes $\mathcal{H}$ is the centroidal voronoi partition.

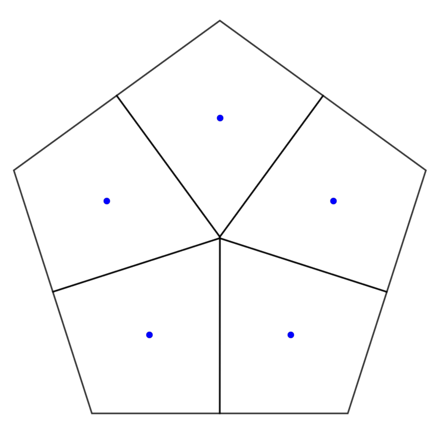

Figure 1: Example of a centroidal voronoi partition, for $\phi(q,t) = 1$.

TD3 Algorithm

Algorithm 1 Twin-Delayed Actor-Critic DDPG

```
Initialize critic networks Q_{θ1}, Q_{θ2} and actor network π_φ with random parameters θ1, θ2, φ.
Initialize the target networks θ'_1 ← θ1, θ'_2 ← θ2, φ' ← φ.
Initialize a replay buffer R.
for t = 1 to T do
    Select action with exploration noise a ~ π(s) + ε, ε ~ N(0, σ) and record the reward r and new state s'.
    Store the tuple (s, a, r, s') into R.
    Sample minibatches of N transitions (s, a, r, s') from R.
    Smooth the target policy according to ã ← π_{φ'}(s) + ε, ε ~ clip(N(0, σ̃), −c, c).
    Perform double Q-learning and clip the results according to y ← r + γ min_{1,2} Q_{θ'_i}(s', ã).
    Update the critics according to θ_i ← min_{θ_i} N^{-1} Σ(y − Q_{θ_i}(s, a))^2.
    if t mod d then
        Update φ by the deterministic policy gradient ∇_φ J(φ) = N^{-1} Σ ∇_a Q_{θ1}(s, a)|_{a=π_φ(s)} ∇_φ π_{φ(s)}.
        Update the target networks according to θ'_i ← τθ_i + (1 − τ)θ'_i, φ' ← τφ + (1 − τ)φ'.
    end if
end for
```

The TD3 algorithm is an improvement on the SACDDPG algorithm, which is more vanilla, and prevents overestimation of the value function by decoupling the action selection and Q-value update.

TD3 reduces variance by updating the policy at a lower frequency than the Q-function updates.

A regularization strategy is introduced by adding a small amount of clipped random Gaussian noise to the selected action and then averages it over minibatches.

Extensions

Placeholder.

Experimental Setup

Placeholder.

Results

Placeholder.

References

[1] C. Nowzari and J. Cortés, "Self-triggered coordination of robotic networks for optimal deployment," *Automatica*, vol. 48, pp. 1077–1087, June 2012.

[2] J. Cortés, S. Martínez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, pp. 243–255, April 2004.

[3] A. Adepegba, S. Miah, and D. Spinello, "Multi-agent area coverage control using reinforcement learning," 2016.

[4] "Deep deterministic policy gradient," *Deep Deterministic Policy Gradient - Spinning Up Documentation.*