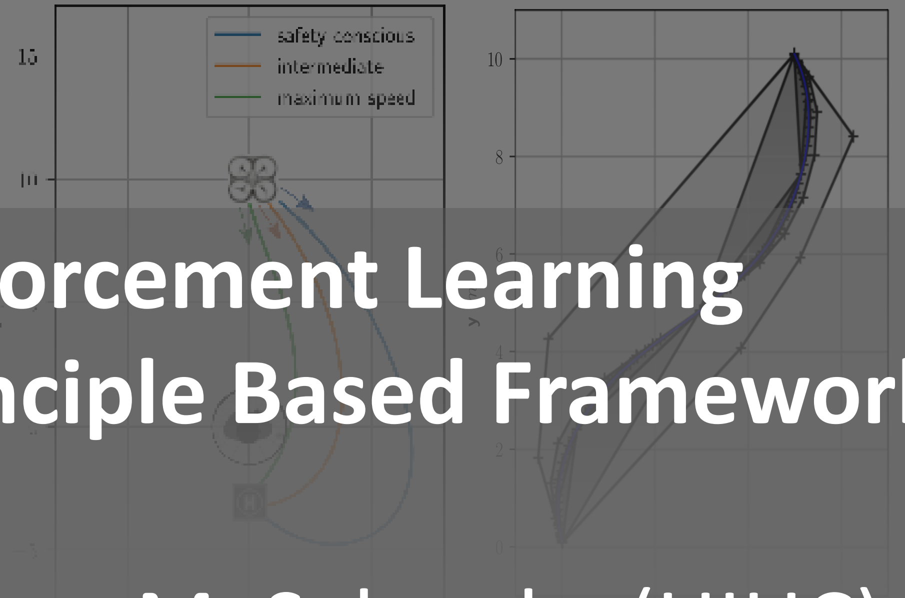
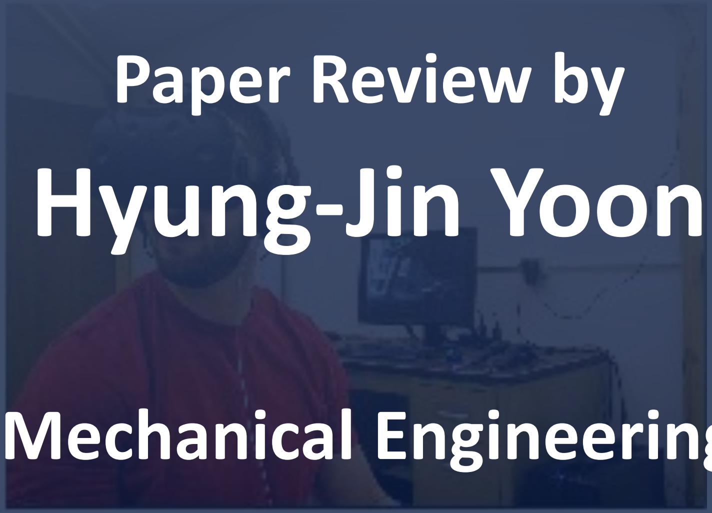
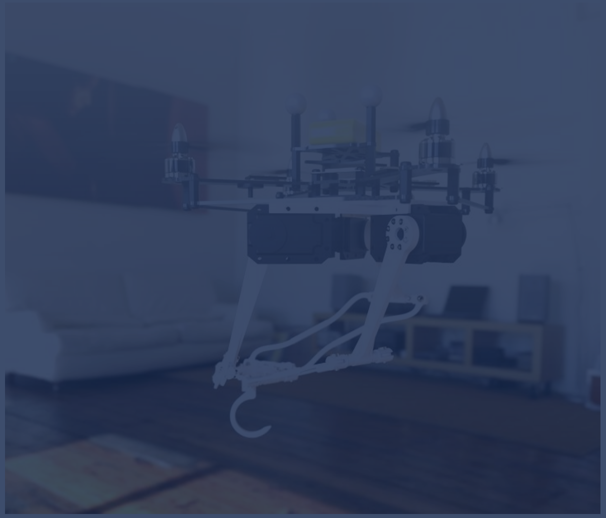



# Parameterized MDPs and Reinforcement Learning Problems - A Maximum Entropy Principle Based Framework

Amber Srivastava (UIUC) and Srinivasa M. Salapaka (UIUC)





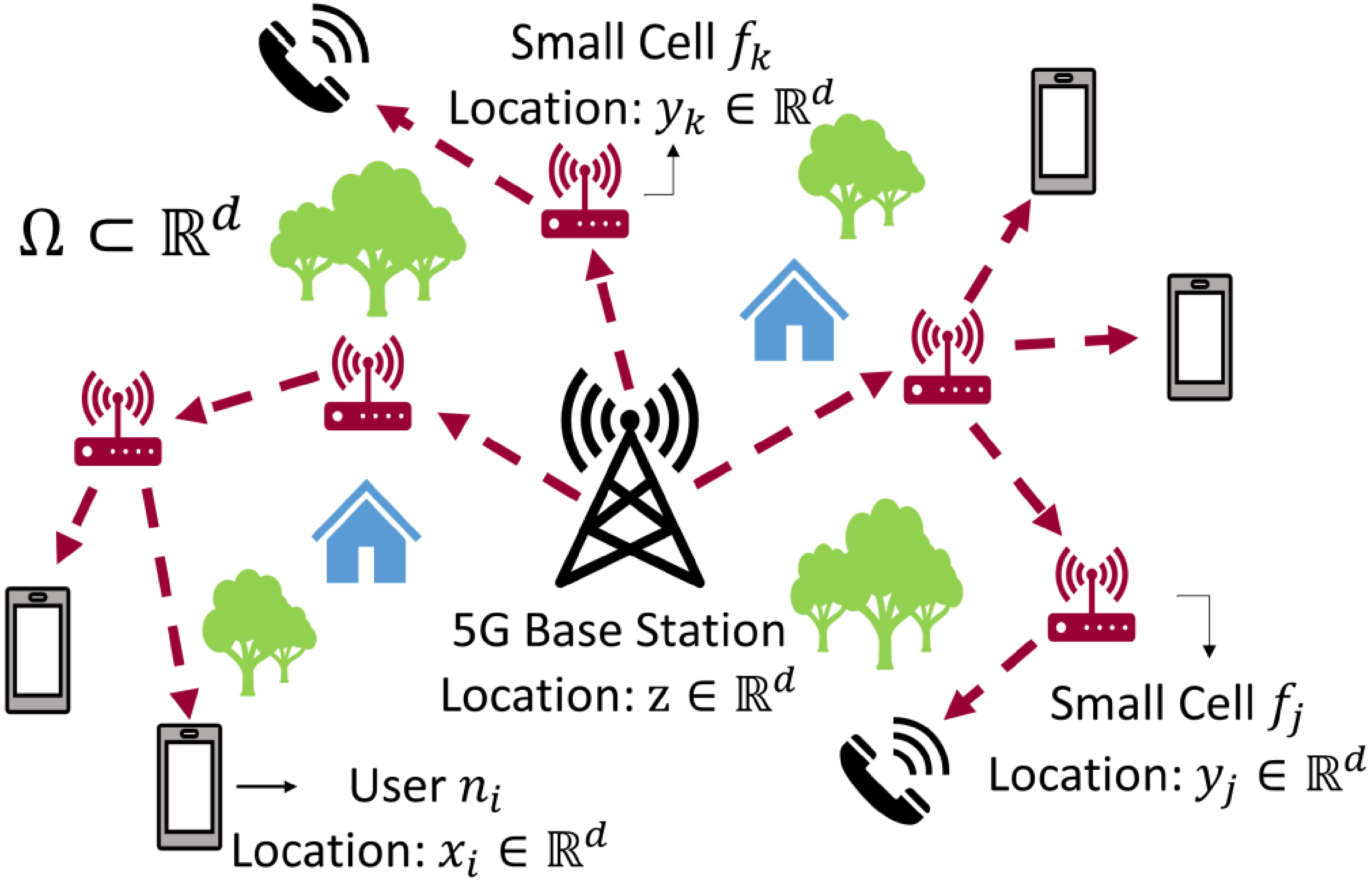
Paper Review by

**Hyung-Jin Yoon**

**Mechanical Engineering**

**University of Nevada, Reno**

# A Motivating Example – Parametrized MDP




**5G Small Cell Network.**

- **Determine the locations of the small cells**
- **Design the communication path**

# State value function of the Parametrized MDP

$$J^{\mu}_{\zeta\eta}(s) = \mathbb{E}_{p_\mu}\left[\sum_{t=0}^{\infty} \gamma^t c(x_t(\zeta), u_t(\eta), x_{t+1}(\zeta)) | x_0 = s\right] \quad (22)$$

is minimized $\forall\ s \in \mathcal{S}$, where $x_t(\zeta)$ denotes the state $x_t \in \mathcal{S}$ with the associated parameter $\zeta_{x_t}$ and $u_t(\eta)$ denotes the action $u_t \in \mathcal{A}$ with the associated action parameter value $\eta_{u_t}$.

**The optimization decision variables:**

- $\mu$ action policy ← traditional MDP
- $\zeta$ state transition parameter, e.g., location of small cell in cellular network.
- $\eta$ action parameter.

# Intro. Maximum Entropy Principle

## Related Works in RL:

- Entropy regularization: maximize entropy for exploration, i.e., $-\sum_a \mu(a|s) \log \mu(a|s)$

## Maximum Entropy Principle (MEP):

- The MEP states that for a random variable $X$ with a given prior information, the most **unbiased** probability distribution given prior data is the one that **maximizes** the Shannon **entropy**.

- Suppose we have an equation to solve, i.e., $E[f(X)] = F$. Then the MEP can be used as follows:

$$\max_{\{p_{\mathcal{X}}(x_i)\}} \quad H(\mathcal{X}) = -\sum_{i=1}^{n} p_{\mathcal{X}}(x_i) \ln p_{\mathcal{X}}(x_i)$$

$$\text{subject to} \quad \sum_{i=1}^{n} p_{\mathcal{X}}(x_i) f_k(x_i) = F_k \quad \forall\ 1 \leq k \leq m \qquad (1)$$

where $F_k$, $1 \leq k \leq m$, are known expected values of the functions $f_k$. The above optimization problem results into Gibbs' distribution [39] $p_{\mathcal{X}}(x_i) = ([\exp\{-\sum_k \lambda_k f_k(x_i)\}]/[\sum_{j=1}^{n} \exp\{-\sum_k \lambda_k f_k(x_j)\}])$, where $\lambda_k$, $1 \leq k \leq m$, are the Lagrange multipliers corresponding to the inequality constraints in (1).

**We can apply the MEP to solve Bellman's equation (that can be written as $E[f(X)] = F$) for RL problems.**

# Solution of MDP with the MEP.

$$\omega = (u_0, x_1, u_1, x_2, u_2, \ldots, x_T, u_T, x_{T+1}, \ldots)$$

$$J^\mu(s) = \mathbb{E}_{p_\mu}\left[\sum_{t=0}^{\infty} \gamma^t c(x_t, u_t, x_{t+1}) \Big| x_0 = s\right] \quad \forall\, s \in \mathcal{S}$$

**We want to minimize the discounted sum of the MDP costs.**

**We will try to get most unbiased solution (hopefully better than local optima) using MEP.**

$$\max_{\{p_\mu(\cdot|s)\}:\mu\in\Gamma} \quad H^\mu(s) = -\sum_{\omega\in\Omega} p_\mu(\omega|s) \log p_\mu(\omega|s)$$

$$\text{subject to} \quad J^\mu(s) = J_0.$$

**And the Lagrangian corresponding to the constrained optimization**

$$V_\beta^\mu(s) = J^\mu(s) - 1/\beta H^\mu(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t c_{x_t x_{t+1}}^{u_t} + \frac{1}{\beta}\left(\log \mu_{u_t|x_t} + \log p_{x_t x_{t+1}}^{u_t}\right) \Big| x_0 = s\right]$$

# Solution of MDP with the MEP.

$$V^{\mu}_{\beta}(s) = J^{\mu}(s) - 1/\beta H^{\mu}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t c^{u_t}_{x_t x_{t+1}} + \frac{1}{\beta}\left(\log \mu_{u_t|x_t} + \log p^{u_t}_{x_t x_{t+1}}\right)\Big|x_0 = s\right]$$

*Theorem 1:* The free-energy function $V^{\mu}_{\beta}(s)$ in (7) satisfies the following recursive Bellman equation:

$$V^{\mu}_{\beta}(s) = \sum_{a,s' \in \mathcal{A},\mathcal{S}} \mu_{a|s} p^{a}_{ss'} \left(\bar{c}^{a}_{ss'} + \frac{\gamma}{\beta} \log \mu_{a|s} + \gamma V^{\mu}_{\beta}(s')\right) \quad (8)$$

where $\mu_{a|s} = \mu(a|s)$, $p^{a}_{ss'} = p(s'|s,a)$, and $\bar{c}^{a}_{ss'} = c(s,a,s') + \gamma/\beta \log p(s'|s,a)$ for simplicity in notation.

# Solution of MDP with the MEP.

*Theorem 1:* The free-energy function $V^{\mu}_{\beta}(s)$ in (7) satisfies the following recursive Bellman equation:

$$V^{\mu}_{\beta}(s) = \sum_{a,s' \in \mathcal{A},\mathcal{S}} \mu_{a|s} p^{a}_{ss'} \left( \bar{c}^{a}_{ss'} + \frac{\gamma}{\beta} \log \mu_{a|s} + \gamma V^{\mu}_{\beta}(s') \right) \quad (8)$$

where $\mu_{a|s} = \mu(a|s)$, $p^{a}_{ss'} = p(s'|s,a)$, and $\bar{c}^{a}_{ss'} = c(s,a,s') + \gamma/\beta \log p(s'|s,a)$ for simplicity in notation.

Now, the optimal policy satisfies $[\partial V^{\mu}_{\beta}(s)/\partial \mu(a|s)] = 0$, which results into Gibb's distribution

$$\mu^{*}_{\beta}(a|s) = \frac{\exp\{-(\beta/\gamma)\Lambda_{\beta}(s,a)\}}{\sum_{a' \in \mathcal{A}} \exp\{-(\beta/\gamma)\Lambda_{\beta}(s,a')\}}, \text{ where} \quad (9)$$

$$\Lambda_{\beta}(s,a) = \sum_{s' \in \mathcal{S}} p^{a}_{ss'} \left( \bar{c}^{a}_{ss'} + \gamma V^{*}_{\beta}(s') \right) \quad (10)$$

**Fixed point iteration**

$$[T\Lambda_{\beta}](s,a) = \sum_{s' \in \mathcal{S}} p^{a}_{ss'} \left( c^{a}_{ss'} + \frac{\gamma}{\beta} \log p^{a}_{ss'} \right) - \frac{\gamma^2}{\beta} \sum_{s' \in \mathcal{S}} p^{a}_{ss'} \log \sum_{a' \in \mathcal{A}} \exp\left\{ -\frac{\beta}{\gamma} \Lambda_{\beta}(s',a') \right\}.$$

**Theorem 2 states that the map is contraction**

# Solution of MDP with the MEP.

**Algorithm 1:** Model-Free Reinforcement Learning

**Input:** $N$, $\nu_t(\cdot,\cdot)$, $\sigma$; **Output:** $\mu^*$, $\bar{\Lambda}^*$
**Initialize:** $t=0$, $\Psi_0=0$, $\mu_0(a|s)=1/|\mathcal{A}|$.
**for** *episode* $=1$ *to* $N$ **do**
  $\beta=\sigma\times$ *epsiode*; reset environment at state $x_t$
  **while** *True* **do**
    sample $u_t\sim\mu_t(\cdot|x_t)$; obtain cost $c_t$ and $x_{t+1}$
    update $\Psi_t(x_t,u_t)$, $\mu_{t+1}(u_t|x_t)$ in (14) and (9)
    break if $x_{t+1}=\delta$; $t\leftarrow t+1$

**By increasing $\beta$ along the episodes, the policy becomes more exploitive from explorative.**

$$\Psi_{t+1}(x_t,u_t)=(1-\nu_t(x_t,u_t))\Psi_t(x_t,u_t)+\nu_t(x_t,u_t)\left[c^{u_t}_{x_t x_{t+1}}-\frac{\gamma^2}{\beta}\log\sum_{a'\in\mathcal{A}}\exp\left\{\frac{-\beta}{\gamma}\Psi_t(x_{t+1},a')\right\}\right]$$

**The left converges to** $\Lambda_\beta(s,a)=:[T\Lambda_\beta](s,a)$.

$$\mu^*_\beta(a|s)=\frac{\exp\{-(\beta/\gamma)\Lambda_\beta(s,a)\}}{\sum_{a'\in\mathcal{A}}\exp\{-(\beta/\gamma)\Lambda_\beta(s,a')\}}$$

**Instead of ARGMIN in Q learning, SOFTMIN with the parameter $\beta$ is used.**

# Parametrized MDP with the MEP.

**Algorithm 3:** Parameterized Reinforcement Learning

**Input:** $\beta_{\min}$, $\beta_{\max}$, $\tau$, $T$, $\nu_t$; **Output:** $\mu^*$, $\zeta$, $\eta$

**Initialize:** $\beta = \beta_{\min}$, $\mu_t = \frac{1}{|\mathcal{A}|}$, and $\zeta$, $\eta$, $G_\zeta^\beta$, $G_\eta^\beta$, $K_\zeta^\beta$, $L_\eta^\beta$, $\bar{\Lambda}_\beta$ to 0.

**while** $\beta \leq \beta_{\max}$ **do**

  Use Algorithm 1 to obtain $\mu^*_{\beta,\zeta\eta}$ at given $\zeta$, $\eta$, $\beta$.

  Consider *env1*($\zeta$,$\eta$), *env2*($\zeta'$, $\eta'$); set $\zeta' = \zeta$, $\eta' = \eta$

  **while** $\{\zeta_s\}$, $\{\eta_a\}$ *converge* **do**

    **for** $\forall s \in \mathcal{S}$ **do**

      **for** *episode* $= 1$ *to* $T$ **do**

        reset *env1*, *env2* at state $x_t$,

        **while** *True* **do**

          sample action $u_t \sim \mu^*(\cdot|x_t)$.

          *env1:* obtain $c_t$, $x_{t+1}$.

          *env2:* set $\zeta_s' = \zeta_s + \Delta\zeta_s$, get $c_t'$, $x_{t+1}$.

          find $G_{\zeta_s}^{t+1}(x_t)$ with $\frac{\partial c_{x_t x_{t+1}}^{u_t}}{\partial \zeta_s} \approx \frac{c_t' - c_t}{\Delta\zeta_s}$.

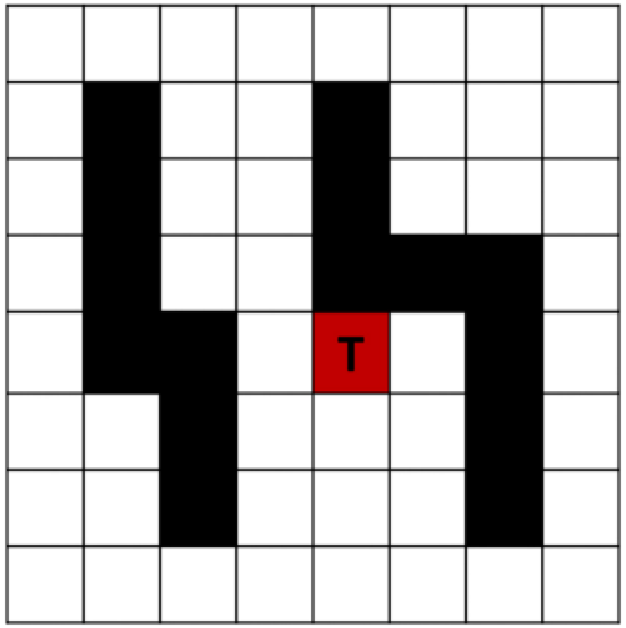
          break if $x_{t+1} = \delta$; $t \leftarrow t + 1$.

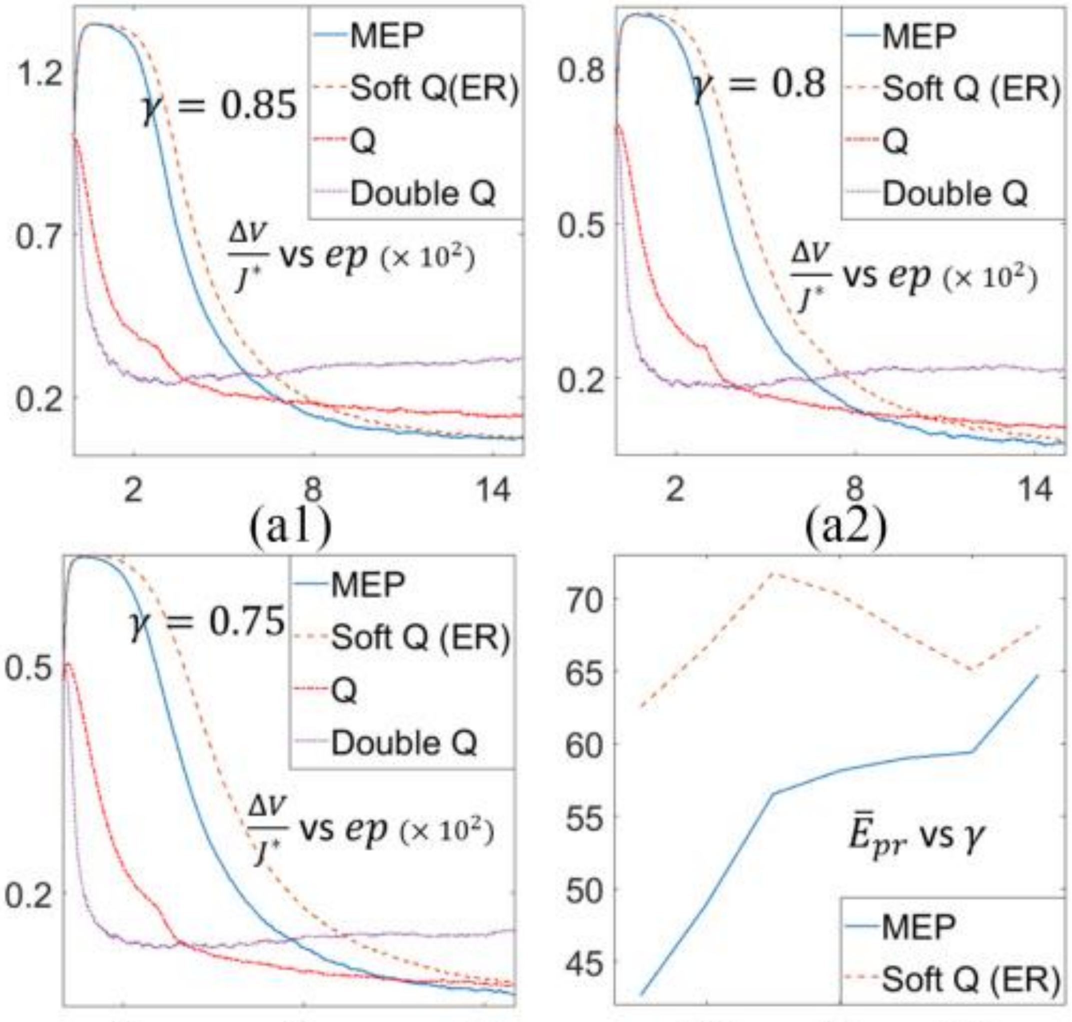
    Similarly learn $G_{\eta_a}^\beta$. Update $\{\zeta_s\}$, $\{\eta_a\}$ in (23).

  $\beta \leftarrow \tau\beta$

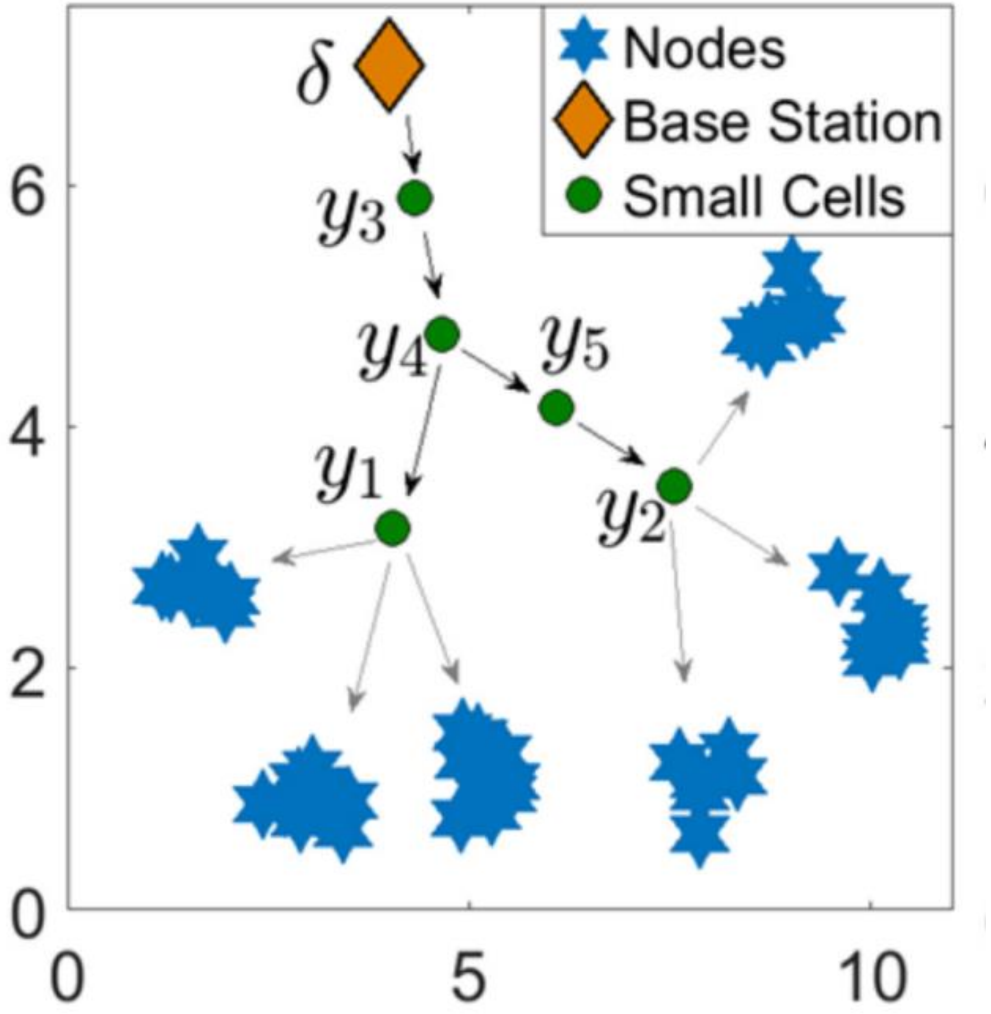
**Use perturbation to estimate gradient (Kiefer–Wolfowitz algorithm?) and update the parameters.**

# Numerical example



# Numerical example




*Simultaneously Determining the Unknown Parameters and Policy in Parameterized MDPs:* We design the 5G small cell network (see Fig. 1) both when the underlying model ($c^a_{ss'}$ and $p^a_{ss'}$) is known (using Algorithm 2) and as well as unknown (using Algorithm 3). In our simulations, we randomly distribute 46 user nodes $\{n_i\}$ at $\{x_i\}$ and the base station $\delta$ at $z$ in the domain $\Omega \subset \mathbb{R}^2$ as shown in Fig. 4(a). The objective is to determine the locations $\{y_j\}_{j=1}^5$ (parameters) of the small cells $\{f_j\}_{j=1}^5$ and determine the corresponding communication routes (policy). Here, the state space of the underlying MDP is $\mathcal{S} = \{n_1, \ldots, n_{46}, f_1, \ldots, f_5\}$ where the locations $y_1, \ldots, y_5$ of the small cells are the unknown parameters $\{\zeta_s\}$ of the MDP, the action space is $\mathcal{A} = \{f_1, \ldots, f_5\}$, and the cost function $c(s, a, s') = \|\rho(s) - \rho(s')\|_2^2$ where $\rho(\cdot)$ denotes the spatial location of the respective states. The objective is to simultaneously determine the parameters (unknown small cell locations) and the control policy (communication routes in the 5G network).

Thank you for your attention!

**Questions and Comments?**