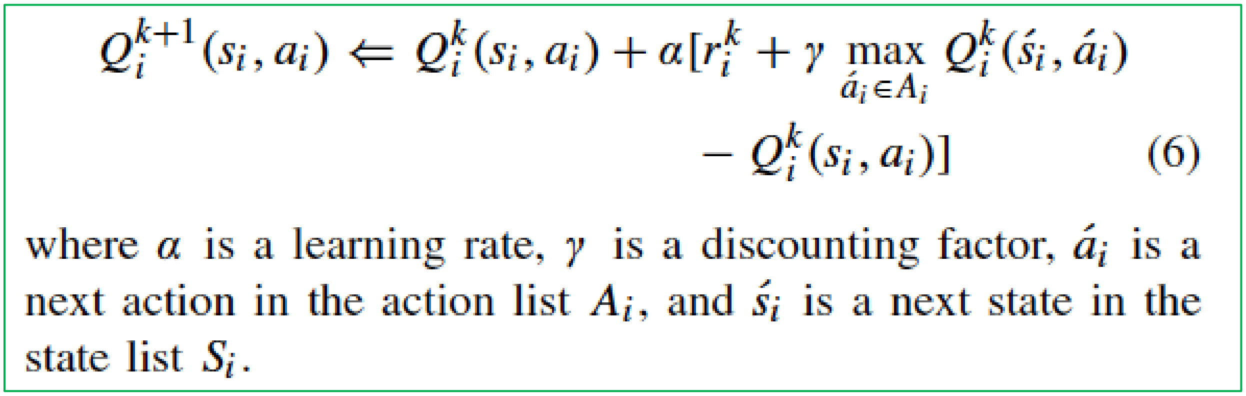
Project 2 (100 points): Multi-robot Cooperative Reinforcement Learning

Project Deadline: November 25th 2021. Submit your project package with report and source codes into Canvas/Webcampus.

Project parameters:

Assume we have 10 mobile robots (N =10) randomly distributed in the area of 50x50. Your project assignment is to implement 2 reinforcement learning algorithms: Independent Q-learning and Cooperative Q-Learning as follows:

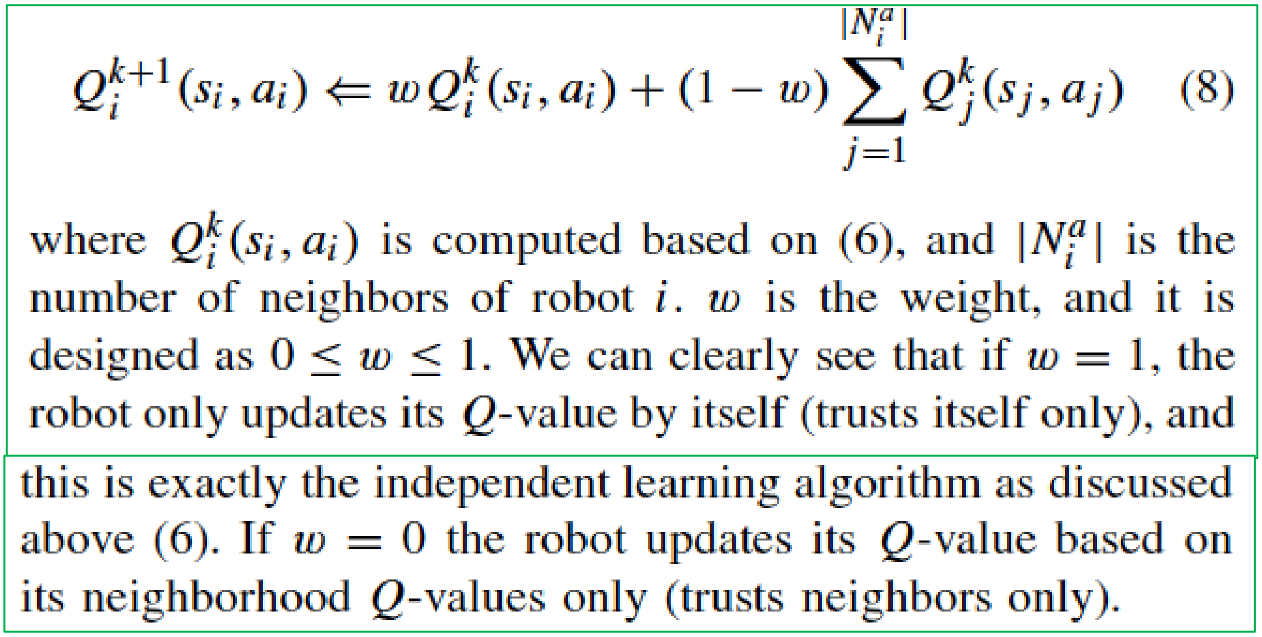
1. Independent Q-Learning



$$Q_i^{k+1}(s_i, a_i) \Leftarrow Q_i^k(s_i, a_i) + \alpha[r_i^k + \gamma \max_{\acute{a}_i \in A_i} Q_i^k(\acute{s}_i, \acute{a}_i) - Q_i^k(s_i, a_i)] \tag{6}$$

where α is a learning rate, γ is a discounting factor, $\acute{a}_i$ is a next action in the action list A_i, and $\acute{s}_i$ is a next state in the state list S_i.

2. Cooperative Q-Learning



$$Q_i^{k+1}(s_i, a_i) \Leftarrow w Q_i^k(s_i, a_i) + (1 - w) \sum_{j=1}^{|N_i^a|} Q_j^k(s_j, a_j) \tag{8}$$

where $Q_i^k(s_i, a_i)$ is computed based on (6), and $|N_i^a|$ is the number of neighbors of robot i. w is the weight, and it is designed as $0 \leq w \leq 1$. We can clearly see that if $w = 1$, the robot only updates its Q-value by itself (trusts itself only), and this is exactly the independent learning algorithm as discussed above (6). If $w = 0$ the robot updates its Q-value based on its neighborhood Q-values only (trusts neighbors only).

- You can explore parameters in both learning algorithms above by yourself. Since Equation (8) returns a large number, you can scale it down by multiplying with a small constant (0.001).

- There is another way to define weight: w = Ni/(Ni+1). You can swap the order of w and 1-w.
- a_i and a_j are the same, or $a_i = a_j = a$ -> rewrite Equation (8)

Other parameters: The communication range of each robot, r = 30. Time step, Delta_t = 0.003 (This parameter is optional and you can change them.)

Assume each robot can select one of the following actions (four safe places to escape predator/enemy) at each time step:

$qt_1 = [540, 400];$

$qt_2 = [540, 0];$

$qt_3 = [300, 400];$

$qt_4 = [300, 0];$

To implement one of these 4 actions, the robot will need a Proportional controller, called P controller as:

$u_i =- k_p (q_i - qt_k)$ where $i = 1, 2, 3, \ldots N$ and $k = 1, 2, 3, 4$

The motion dynamics of each robot is as:

$$\begin{cases} \dot{q}_i = p_i \\ \dot{p}_i = u_i \end{cases}$$

In the discretized form, the motion dynamics of each robot can be written as

$q_i (k) = q_i (k-1) + Delta\_t *p_i (k-1) + Delta\_t * Delta\_t *u_i (k-1)/2;$

$p_i (k) = [q_i (k)- q_i (k-1)]/Delta\_t.$

The robots can move together and avoid collision by using the flocking algorithm 2 in your project 1.

<u>Project Requirement:</u>

1. (40 points) Implement both cooperative and independent reinforcement Q-learning algorithms, respectively as stated above (more detail about these two algorithms you can see in Lecture 10). The cooperative Q-Learning algorithm should be able to allow the robots to learn from itself and its neighbors so that they can agree on the same action to go to the same safe place (e.g., to escape

predator/enemy) in order to maximize its reward (the reward is defined by number of its neighbors in the robot’s communication range, r).

2. (10 points) Plot the trajectory of the robots in the first, second and last learning episodes.
3. (10 points) Plot the individual reward of each robot in cooperative Q-learning and independent Q-learning over learning episodes. You can have two separate figures: one for cooperative Q-learning and one for independent Q-learning.
4. (10 points) Plot the total reward (sum of the reward) of all robots in cooperative Q-learning vs. independent Q-learning over learning episodes.
5. (10 points) Plot the action selection index of each robot cooperative Q-learning and independent Q-learning over learning episodes. Assume the actions are encoded/indexed by numbers 1, 2, 3, 4 respectively. You can have two separate figures: one for cooperative Q-learning and one for independent Q-learning.
6. (10 points) Plot the average of delta Q in cooperative Q-learning vs. independent Q-learning over learning episodes.
7. (10 points) Change the parameter, *w*, in the cooperative Q-Learning algorithm and observe/show the results. Then explain your observations based on the plots/figures of the results.