# Homework #3. Distributed Temporal Difference Learning

Yongsung Cho

CHI CHIEH WENG

Computer Science, Oregon State University

AI 533: Intelligent Agents & Decision

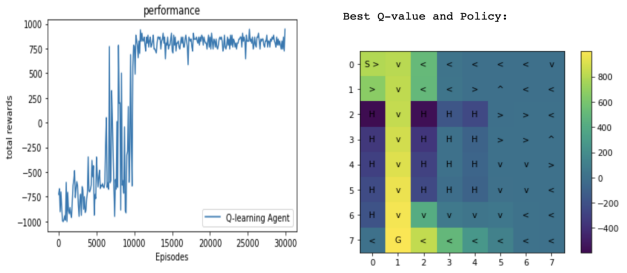
Prof. Alan Fern

Due Nov 12, 2021

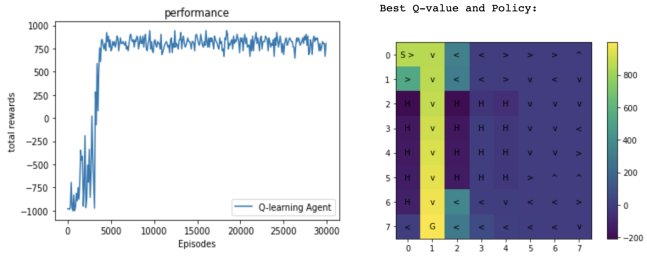
Provide the learning curves for the above experiments. Clearly label the curves by the parameters used.

### Dangerous Map with Q-learning

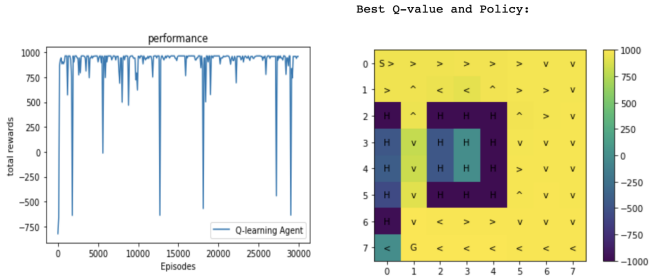
Learning rate: 0.001 / **epsilon: 0.3** / Learning time: 58.98282217979431



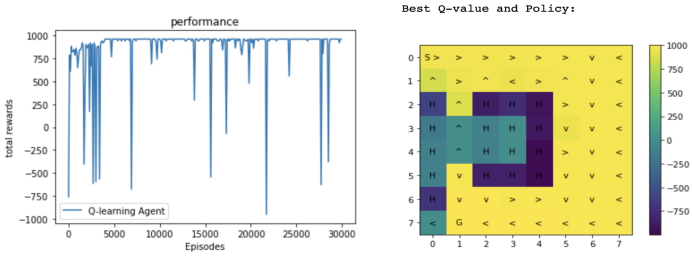

Learning rate: 0.001 / **epsilon: 0.05** / Learning time: 28.47514510154724




**Learning rate: 0.1** / epsilon: 0.3 / Learning time: 30.579301834106445

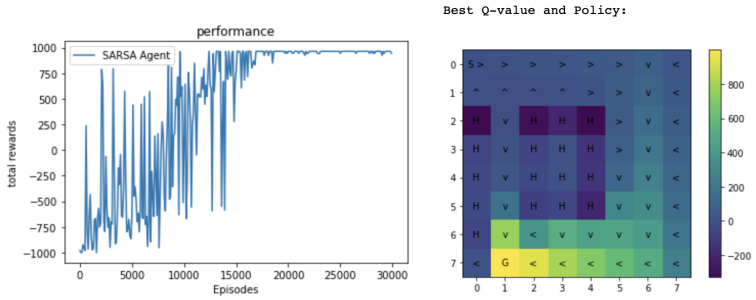



**Learning rate: 0.1** / epsilon: 0.05 / Learning time: 30.92876625061035

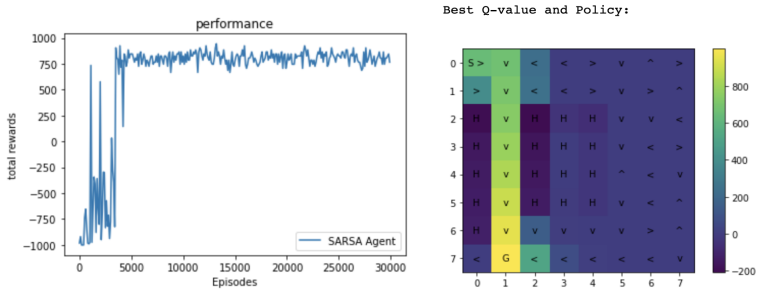


## Dangerous Map with SARSA

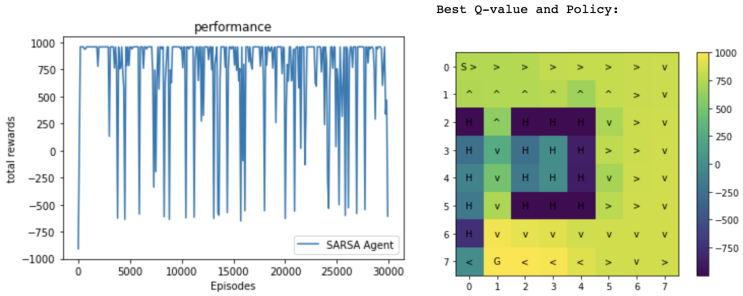
Learning rate: 0.001 / **epsilon: 0.3** / Learning time: 86.35592794418335



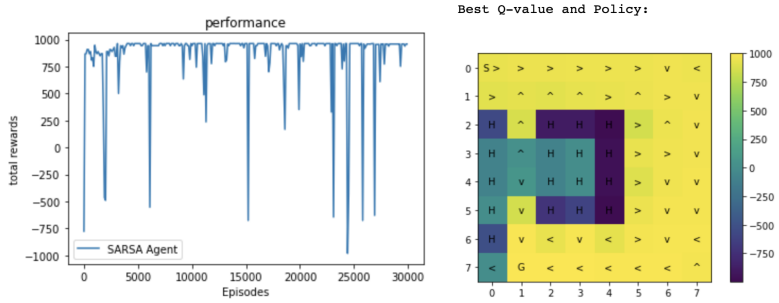

Learning rate: 0.001 / **epsilon: 0.05** / Learning time: 22.65689492225647




**Learning rate: 0.1** / epsilon: 0.3 / Learning time: 67.61502122879028

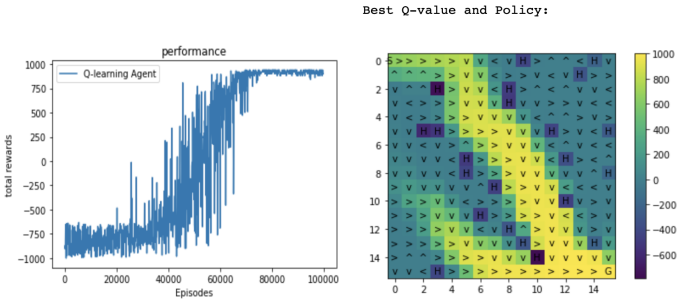



**Learning rate: 0.1** / epsilon: 0.05 / Learning time: 31.83169412612915

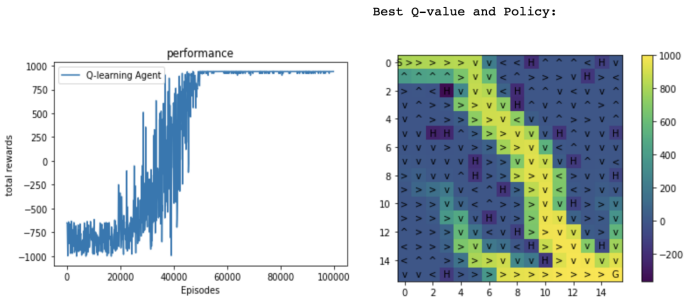


## Map16x16 with Q-learning

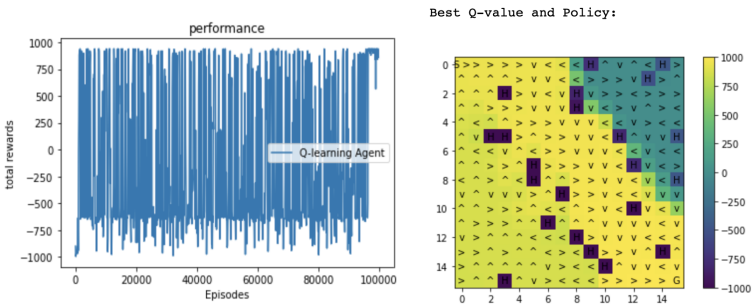
Learning rate: 0.001 / **epsilon: 0.3** / Learning time: 514.8704888820648



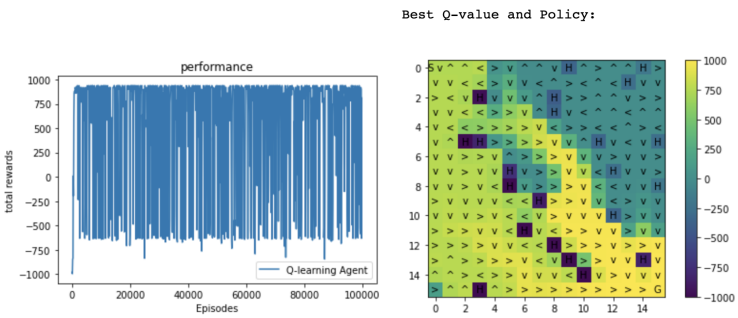

Learning rate: 0.001 / **epsilon: 0.05** / Learning time: 393.44849371910095




**Learning rate: 0.1** / epsilon: 0.3 / Learning time: 704.2829818725586

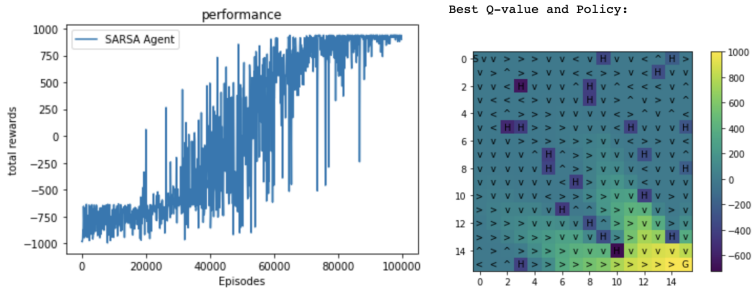



**Learning rate: 0.1** / epsilon: 0.05 / Learning time: 435.89066100120544

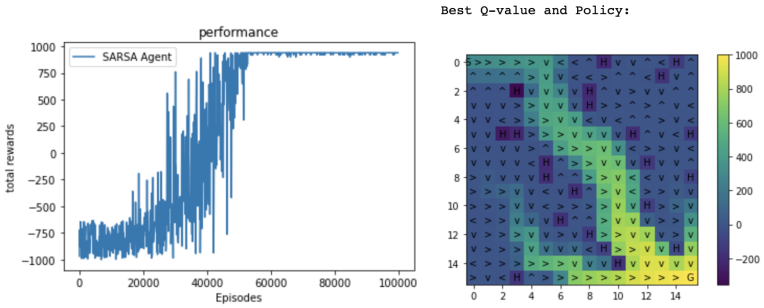


## Map16x16 with SARSA

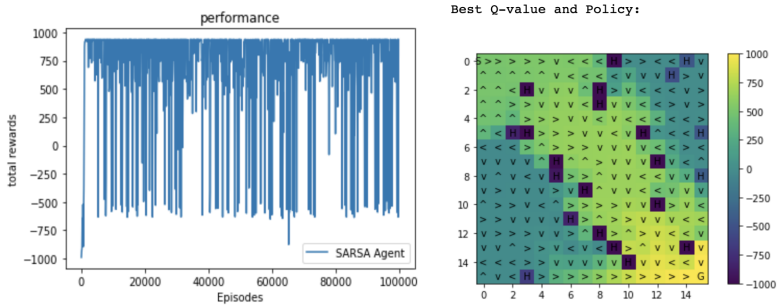
Learning rate: 0.001 / **epsilon: 0.3** / Learning time: 577.456444978714



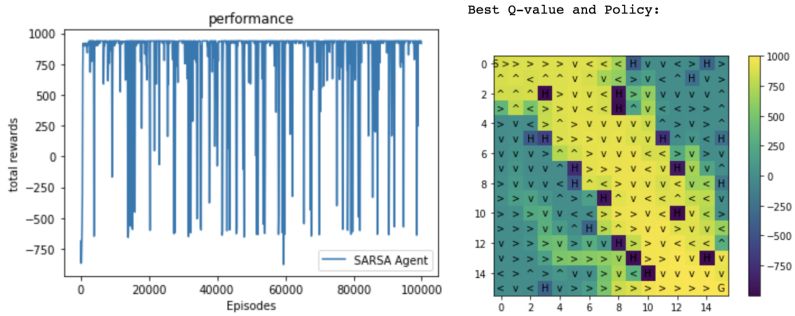

Learning rate: 0.001 / **epsilon: 0.05** / Learning time: 380.06924533843994




**Learning rate: 0.1** / epsilon: 0.3 / Learning time: 300.59545493125916

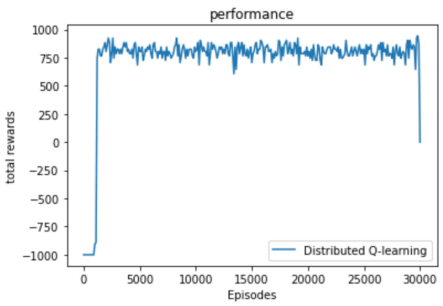


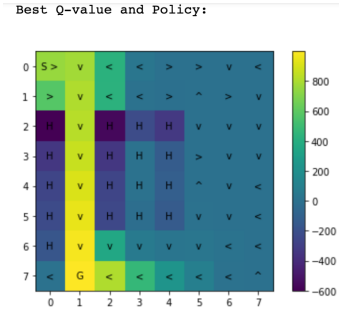

**Learning rate: 0.1** / epsilon: 0.05 / Learning time: 206.45802187919617



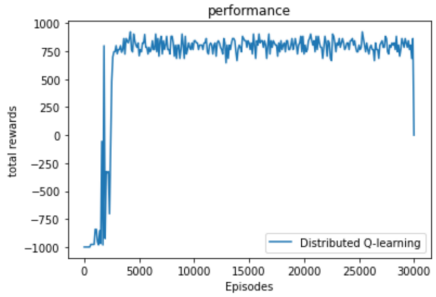
**Dangerous Map Distributed Q-learning Agent (Learning rate:** 0.001 / **epsilon**: 0.3)

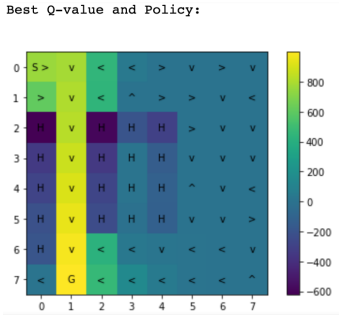
**Collector workers = 2, Evaluator workers = 4, Learning time: 35.28325390815735**



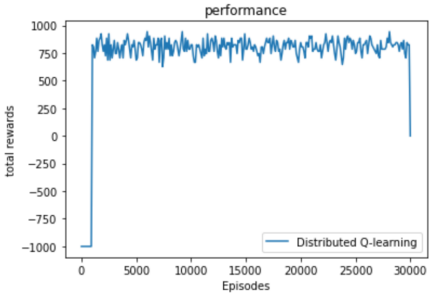



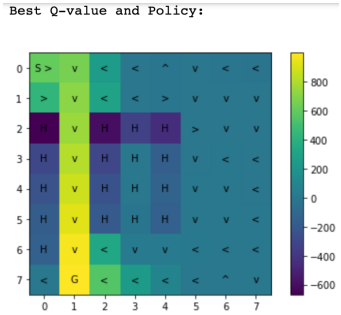

**Collector workers = 4, Evaluator workers = 4, Learning time: 33.861584186553955**



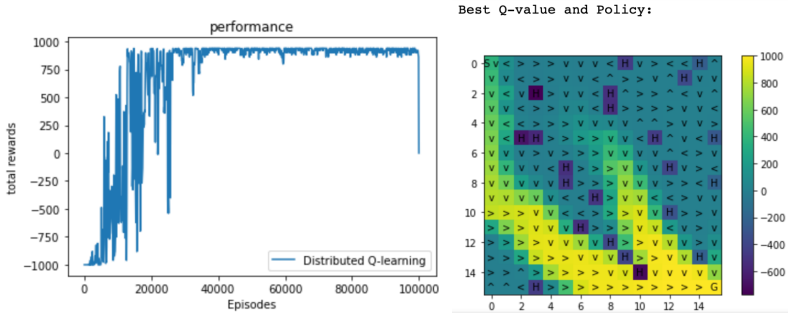

**Collector workers = 8, Evaluator workers = 4, Learning time: 10.278455972671509**



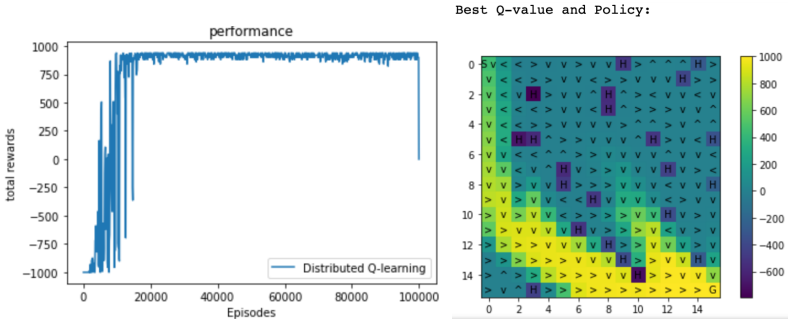



**Map16x16 Distributed Q-learning Agent (Learning rate:** 0.001 / **epsilon**: 0.3**)**

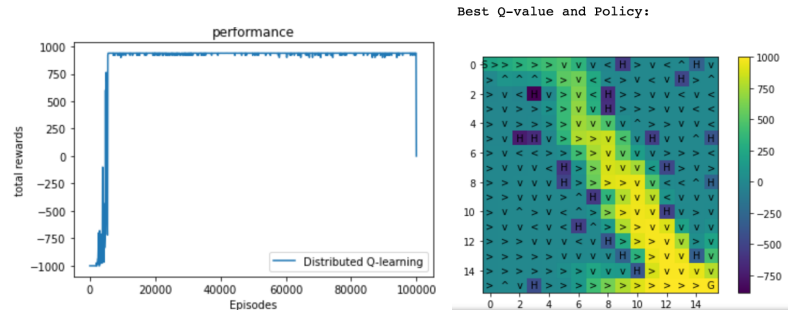
**Collector workers = 2, Evaluator workers = 4, Learning time: 88.79792404174805**




**Collector workers = 4, Evaluator workers = 4, Learning time: 63.04349493980408**




**Collector workers = 8, Evaluator workers = 4, Learning time: 61.284079790115356**



1. **Did you observe differences for SARSA when using the two different learning rates? If there were significant differences, what were they and how can you explain them?**

   When the learning rate is 0.001, the total reward will gradually increase as the episode increases, and compared with the higher learning rate, the fluctuation between each episode is very small. When the learning rate is 0.1, the total reward will reach the highest value in a short time and fluctuate greatly. Therefore, in the Q value and strategy mapping, the mapping with a higher learning rate will result in a higher Q value in each grid.

2. **Repeat (1) for Q-Learning.**

   As shown in SARSA, when the learning rate is 0.001, total reward gradually increases with increasing episodes. The high episode shows a small fluctuation and shows a relatively constant reward value. However, when the learning rate is 0.1, overall high reward values are shown, but some episodes show very low reward values and the overall graph shows high fluctuation. In Temporal Difference learning, Q-value is updated by adding a noisy value to the current Q-value value. When the learning rate is 0.1, the learning rate is too high, and the overall Q-value tends to be high. Compared with SARSA, Q-learning converges to a high reward value in relatively short episode trials.

3. **Did you observe differences for SARSA when using different values of $\epsilon$? If there were significant differences, what were they and how do you explain them?**

   When the epsilon value is 0.3, the Greedy policy is selected with a probability of 0.7, and the random policy is selected with a probability of 0.3. Using a low epsilon value like 0.05 increases the probability of choosing a greedy policy. It means that the total reward is more focused on the exploit than the exploration, so a small trial is needed to increase the total reward to the optimal value. Moreover, since I don't randomly select a policy, the fluctuation of the overall graph tends to decrease.

4. **Repeat (3) for Q-Learning.**

   As a result of comparing the different epsilon values, as shown in SARSA, the tendency to reach rapidly from the low epsilon value to the optimized value was shown. Furthermore, because of the low epsilon value, the greeny policy is mainly selected. Because the probability of randomly selecting a policy is also reduced, the fluctuation of the graph is also reduced. This also means that the reward value is rarely increased except

for optimal policy. Therefore, when we look at the Q-value policy, we can see that it shows a very low reward value outside the optimal route.

5. **For the map "Dangerous Hallway" did you observe differences in the policies learned by SARSA and Q-Learning for the two values of epsilon (there should be differences between Q-learning and SARSA for at least one value)? If you observed a difference, give your best explanation for why Q-learning and SARSA found different solutions.**

In learning rate 0.001 and epsilon 0.3, Q-learning showed higher reward value in the shortest path, and in SARSA, reward value in the far-return route is relatively higher than the fastest route. I think the reason is due to the difference between the algorithms of Q-learning and SARSA. In Q-learning, each change in each state is used to select a new action value using an explore/exploit policy. However, in SARSA, the Q-function is updated with the selected action, and the corresponding action is selected as the next action. These choices mean that the policy may not go to the optimal path. In the Dangerous Hallway map, there is the shortest path, but this is dangerous, and the direction of the route is limited. However, since there are many routes on the far-return way, SARSA, which randomly selects the action and selects the action as the following action, calculates the high reward for the far-return path. Furthermore, I think there is a high reward for the far-return route because SARSA updates Q-function more often for the far-return way, which can come to multiple paths.

7. **Show the value functions learned by the distributed methods for the best policies learned with $\epsilon$ equal to 0.3 and compare to those of the single-core method. Run the algorithm for the recommended number of episodes for each of the maps. Did the approaches produce similar results?**

In the Dangerous map, the value function of the single-core method and distributed methods was almost similar. However, when the collector workers were 2 and 4 in MAP16, a value function was slightly different from the single-core method. In the single-core method, the value function of the diagonally heading straight to the goal was shown, but in the distribution method, the value function was higher in the lower left side with another route from start to destination.

8. **Provide and compare the timing results for the single-core and distributed experiments, including the time to do the evaluations during learning. Describe the trends you observe as the number of workers increases.**

In Dangerous map, the learning time was about 58 when single-core was used. However, when the distributed Q-learning agent was used, and the collector workers were 2, the learning time decreased to 35, and the learning time decreased with the increase of collector workers. As a result, learning time was reduced to 10 when eight collector workers were used, about one-sixth of that of single-core 58. Similarly, the learning time was about 514 when single-core was used in MAP16. Because the state, action, and episode number is high, it takes longer than the dangerous map. When the value function is calculated using the distributed Q-learning agent in MAP16, the learning time is reduced to about 88 using two collector workers. Like the tendency shown in the dangerous map, the learning time decreases as the number of collector workers increases. As a result, when collector workers were 8, learning time fell to 61.

9. **Provide and compare the timing results for the single-core and distributed experiments with the evaluation procedure turned off. That is, here you only want to provide timing results for the learning process without any interleaved evaluation. Compare the results with (8).**

DH:

Trial 1:

| | Distributed experiment | | | Single-core | |
|---|---|---|---|---|---|
| | collector workers=2 | collector workers=4 | collector workers=8 | Q-learning | SARSA |
| Learning time | 8.382742166519165 | 5.717622995376587 | 3.7572391033172607 | 69.443106174469 | 118.63066101074219 |

Trial 2:
Distributed Q-learning agent with 4 evaluator: 4.406655550003052

Distributed Q-learning agent with evaluation procedure turned off: 22.59319519996643

Q-learning agent with single-core: 40.28834819793701

## MAP16:

Trial 1:

| | Distributed experiment | | | Single-core | |
|---|---|---|---|---|---|
| | collector workers=2 | collector workers=4 | collector workers=8 | Q-learning | SARSA |
| Learning time | 91.00746893882751 | 55.12092995643616 | 48.25921010971069 | 744.7377121448517 | 773.7636840343475 |

Trial 2:

Distributed Q-learning agent with 4 evaluator: 32.06893444061279

Distributed Q-learning agent with evaluation procedure turned off: 54.1111562252044

Q-learning agent with single-core: 371.1701476573944

If the evaluation procedure is turned off, the learning time can be increased, as shown in the above results. This suggests that the existence of an evaluation procedure affects the overall learning time. In (8) question, the higher the number of collector workers, the less learning time. Likewise, the learning time tends to decrease as the number of evaluation procedures increases.