Abstract:

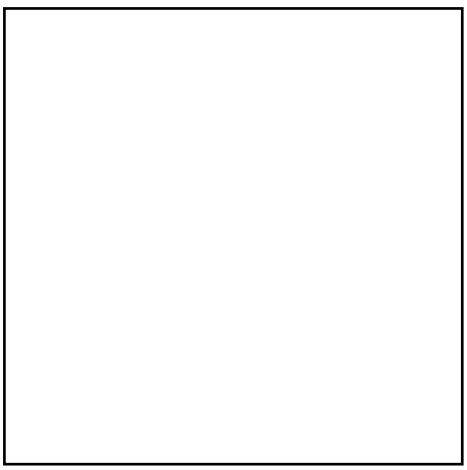
In this report, three reinforcement learning algorithms (value iteration, policy iteration and Q-learning) are used to investigate two Markov decision process problems. Performances are compared and influence of parameters (learning rate and exploration rate) are studied.

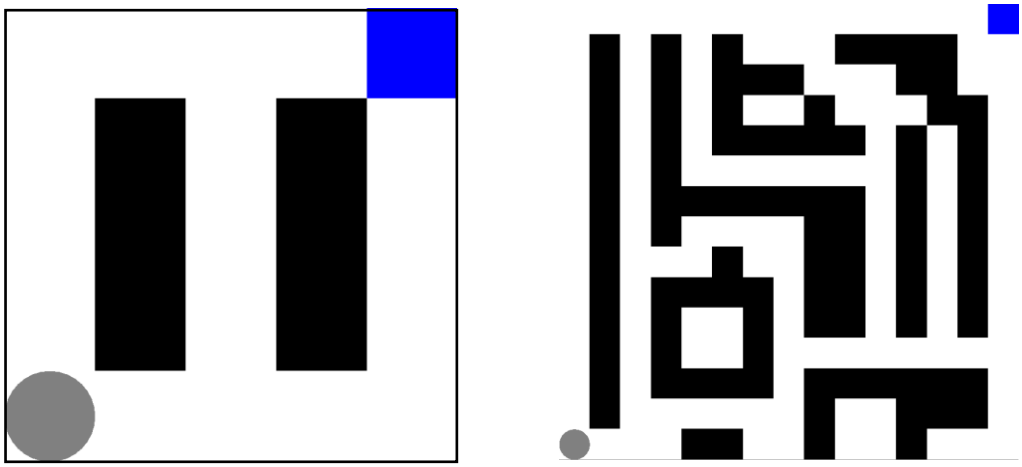
Markov Decision Processes:

A Markov decision process, MDP(S, A, T, R, γ), is specified by five parameters, where S is a finite set of problem state, A is a finite set of actions that could change the state of MDP, T is a random factor determing the possibility of taking the action corretly, R is the immidiate reward of taking the action, finally γ that ranges 0 to 1 is the discount factor adjusting the weights of relative importance of future rewards.By the defination of the MDP, algorithms are used to generate the best policy to maximum the total rewards and give the corresponding action series [1].

In this report, the MDP(S, A, T, R, γ) is implemented by maze solving problems with N by N grids. This sort of problems are interesting to check the ability of reinforcement algorithms finding the best strategy. The application of such problems could be used for pattern recognization, artificial intellegence, decision making and many other related fields.

The states S is defined by the total grid positions in the maze. Action A is defined by allowing movement to up, down, left, right or stay if hits on the wall. T is defined by the probobility (0.8) that taking the right action and taking the wrong actions with probobility of 0.2. R is defined by 100 reward on the destination of the maze and -1 on every other available positions. The discount factor γ is set ot 0.99.





(a) Easy grid world (b) Hard grid world

Fig.1 Two MDP problems

Two sizes (5*5, 15*15) of mazes impemented and studied using Burlap shown as Fig.1. To make easy comparison, the start point is set at the bottom left (grey circle) and the destination is at the top right (blue square). However, value iteration, policy iteration and Q-learning will actually return the best action at every available position.

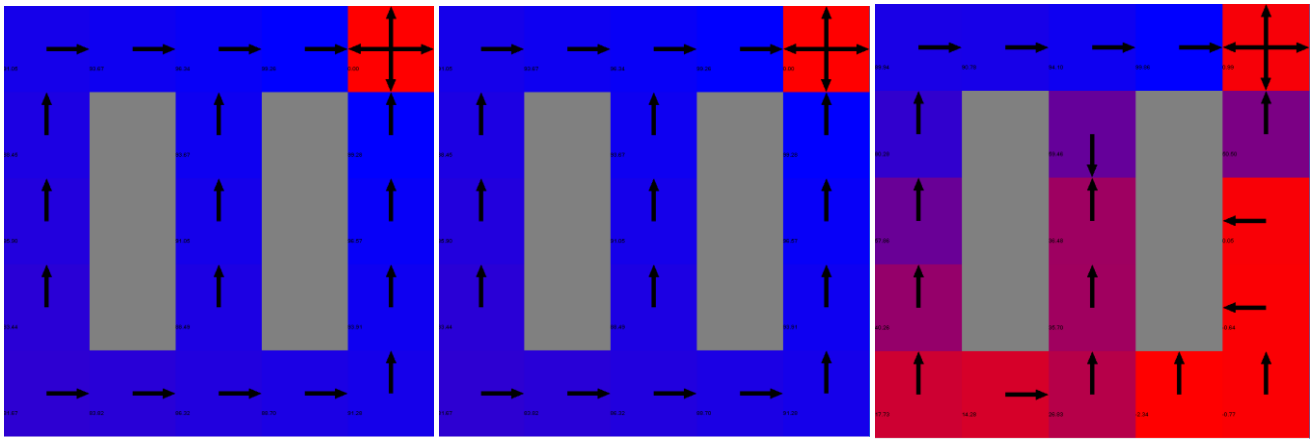
Reinforcement Learning Algorithms:

1. ***Value iteration***: Value iteration interatively update the Bellman equation to maximize the the utility of the states until convergence. Firtly, the utility value is assigned to each state with the immediate reward only. Then, by calculating the utility of nearby states the reward of the terminal will sperate to all domain and thus solve the MDP [1].
2. ***Policy iteration***: MDP is solved by using arbitrary policies and comparing the utility values of those different policies. Then the process is repeated until no improvement can be done for the utility value[1].
3. **Q-learning:** Different from value iteration and policy iteration, Q-learning does not require the domain knowledge thus is a model free algorithm. The agent will walk through all possible positions to update the Q value based on immediate and delayed reward. By this kind of iteration MDP converges to the solution[2].

Part 1. Easy Grid World (5*5):

For all three algorithms, the value iteration, the policy iteration and the Q-learning, 20 iterations are runed for each for this simple MDP. The discount factor is set to 0.99. A middle value 0.5 is selected for Learning rate of Q-learning and the exploration rate is set as default 0.1.

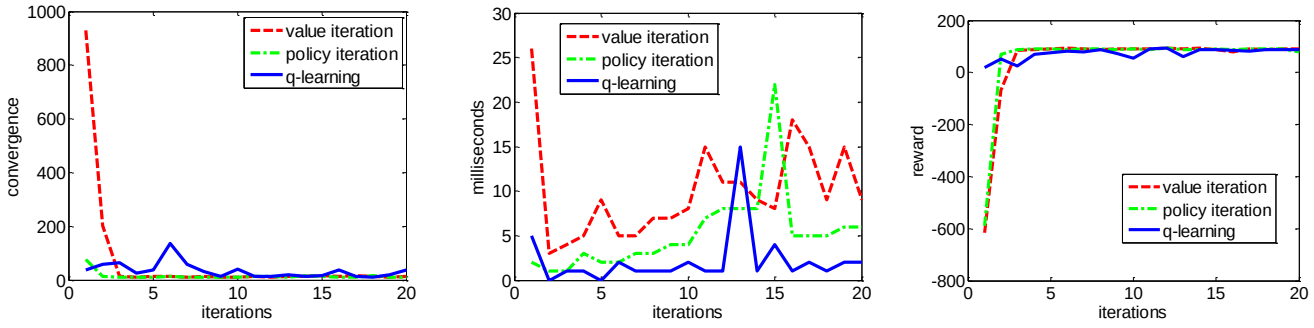
The utility of each position and the corresponding preferable action are shown as Fig.2. Higher utility is colored with blue and the lower with red. Arrow on the grid points out the best action of that state should take to gain higher utility. So, the agent will go to the neighbor grid with highest utility or just stay where it is if no neighbor grid is better than itself.



(a) Value iteration (b) Policy iteration (c) Q-learning

Fig.2 Utility and action yielded by different algorithms

As showed in Fig.2, all of three algorithms gives best policy to find the destination. Value iteration and policy iteration can give the best action for every grid, whereas the Q-learning only show the correct action on some of states.



(a) Convergence (b) Time cost (c) Total reward

Fig.3 Comparison of different algorithms

As shown in Fig.3, value iteration and policy iteration converge pretty fast (3-4 iterations) whereas the q-learning needs more than 10 iterations to converge. The q-learning also shows fluctuations due to the use of greedy algorithms to prevent local optimal.

For the time cost, generally value iteration and policy iteration are at same level. The value iteration consumes a bit more than policy iteration, but much more than the q-learning. The time cost grows with iteration for all those three algorithms.

Similar to convergence, value iteration and policy iteration reach the highest total reward within 3-4 iterations whereas q-learning takes around 8 iterations. The randomness is showed here as well.

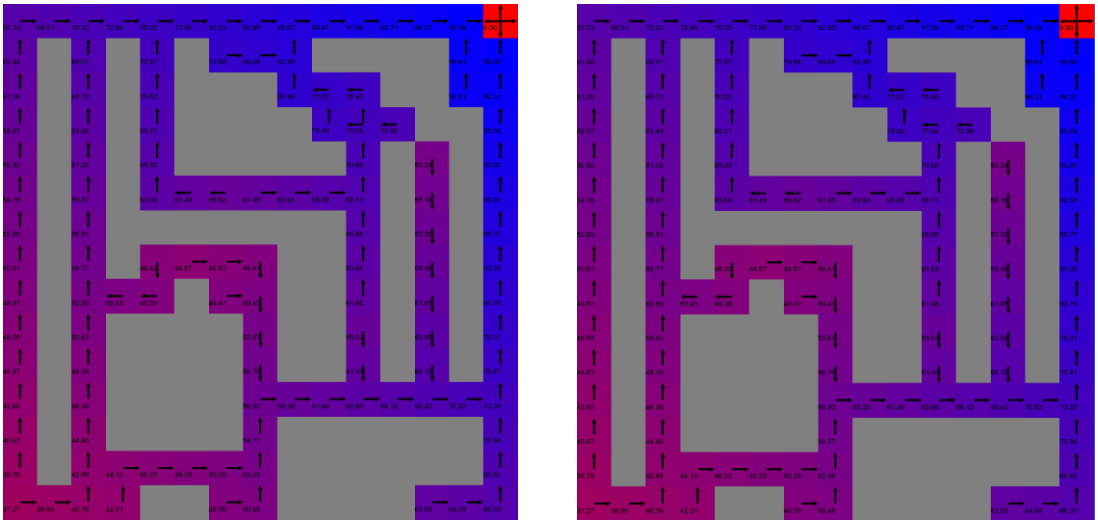
Part 2. Hard Grid World (15*15):

For the second MDP, we use a larger maze with the same model as before. We set many more complicated paths deliberately such as dead end, circle, zig-zag walls and so on. So, we could see how do those algorithms solve such hard problem. The discount rate is set to 0.99. The learning rate and the exploration rate is set to 0.5 and 0.1 respectively for Q-learning, same as in the easy grid world. All algorithms will run for 50 iterations.

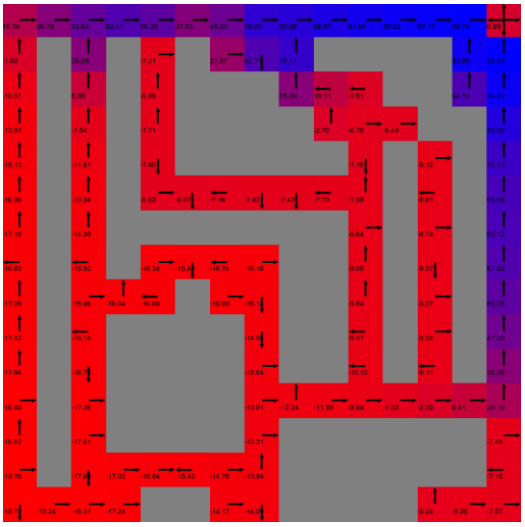
Shown as Fig.4, again, value iteration and policy iteration give the best policy within 50 iterations. Meanwhile all the grids in the domain are labeled with the best actions for highest reward. The whole maze is colored with deep blue/purple showing that every state is of a high utility. The lattice closer to the destination has a deeper blue with higher value, which is resulted from the fact that the reward of the destination is propagate to the whole domain, just as defined in the algorithms.

However, the q-learning does not give the right solution for this problem within 50 iterations. The majority of the domain is colored with red (low utility) and the actions are sort of randomly suggested. Only the grids near to the destination are properly labeled with the arrows. It could indicate that

q-learning needs more steps to iterate than value iteration and policy iteration, which is reasonably because each step for q-learning is much easier than the other two and the greedy algorithm could cause q-learning step down the optima.



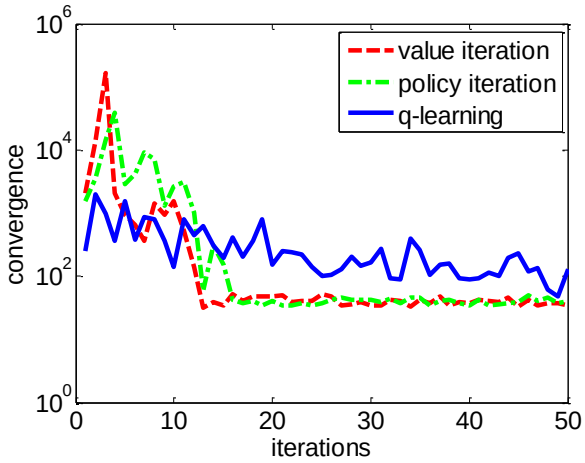
(a) Value iteration (b) Policy iteration



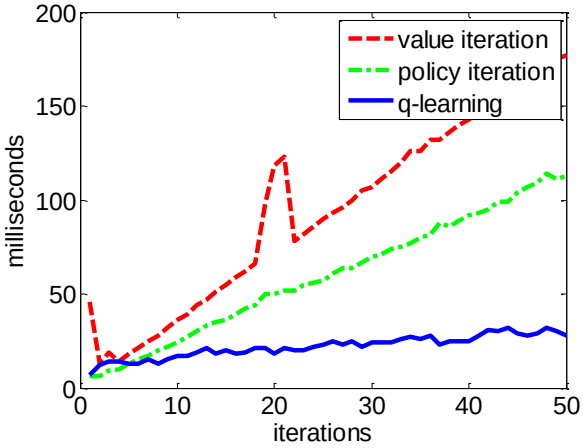
(c) Q-learning

Fig.4 Utility and action yielded by different algorithms

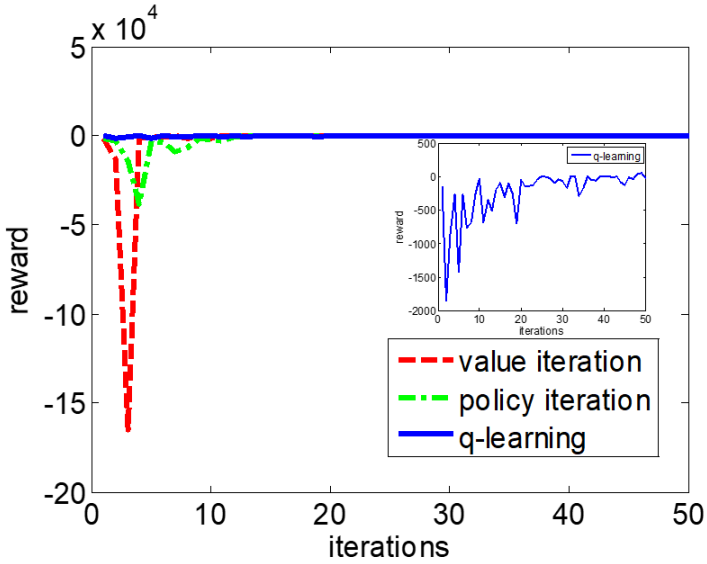
Fig.5 shows that value iteration and policy iteration could converge fast (around 15 steps) even with this complicate MDP. However, the q-learning fails to converge within 50 iterations. Also, value iteration and policy iteration show good stability whereas the q-learning shows randomness due to the use of greedy algorithms.



(a) Convergence



(b) Time cost



(c) Total reward

Fig.5 Comparison of different algorithms

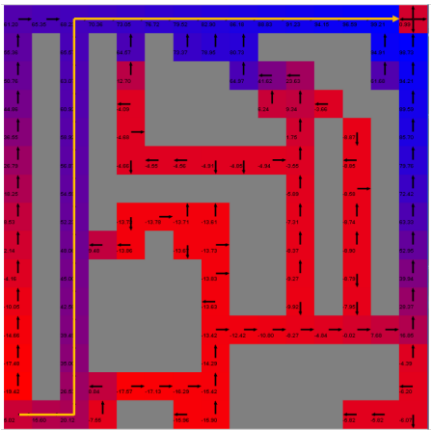
For this more difficult MDP, the time cost is more clearly showed for comparison. For the same amount of iterations, value iteration costs most time, policy iteration costs the second and q-learning costs the less. For each algorithm, time cost grows linearly with the iterations.

Finally, value iteration and policy iteration can converge to the highest total reward within around 15 iterations. Whereas q-learning fails on doing that.

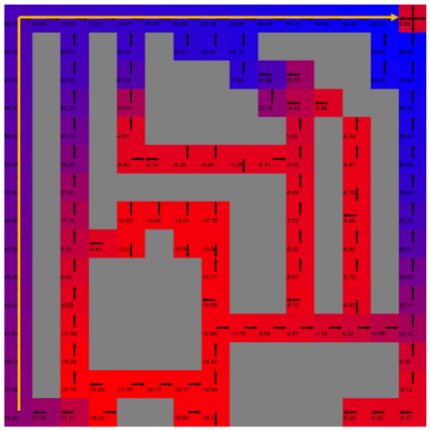
Part 3. Parameter tuning for Q-learning:

As discussed in part 1 and part 2, we concluded that q-learning needs more iterations to converge than value iteration and policy iteration. However, the time cost is much lower for q-learning. Meanwhile, we would also like to invesitage how the learning rate and exploration rate will affect the the results.

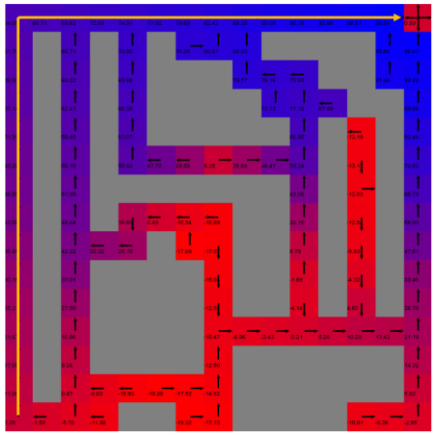
Thus in this part we will try the combinations of different learning rate (0.1, 0.5, 0.9) and exploration rate (0.1, 0.5, 0.9) for the second MDP for 500 iterations.



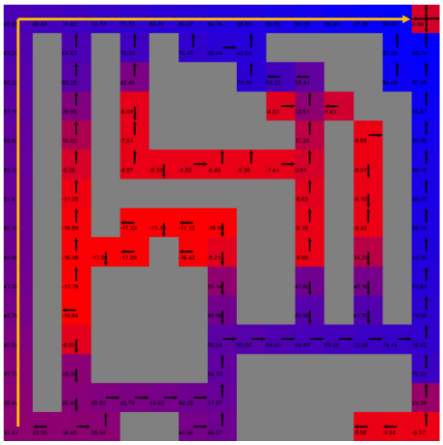
(a) L0.1 E.01



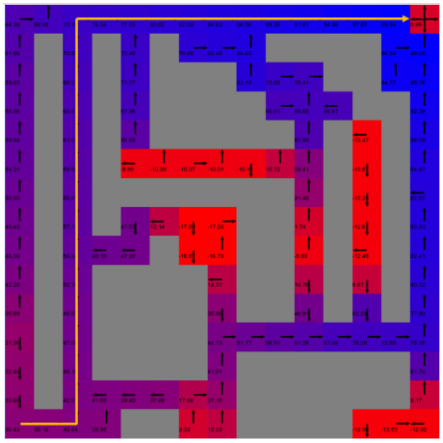
(b)L0.1 E0.5



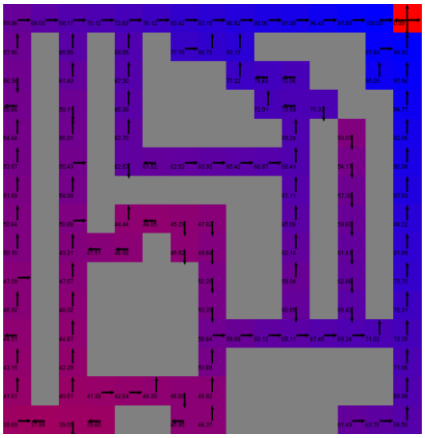
(C)L0.1 E0.9



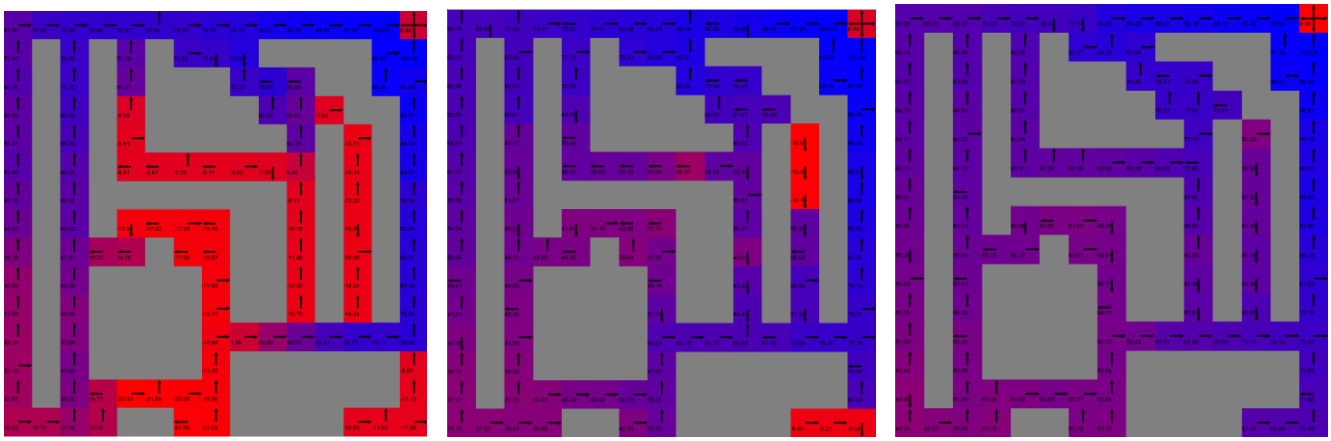
(e) L0.5 E.01



(f)L0.5 E0.5



(g)L0.5 E0.9

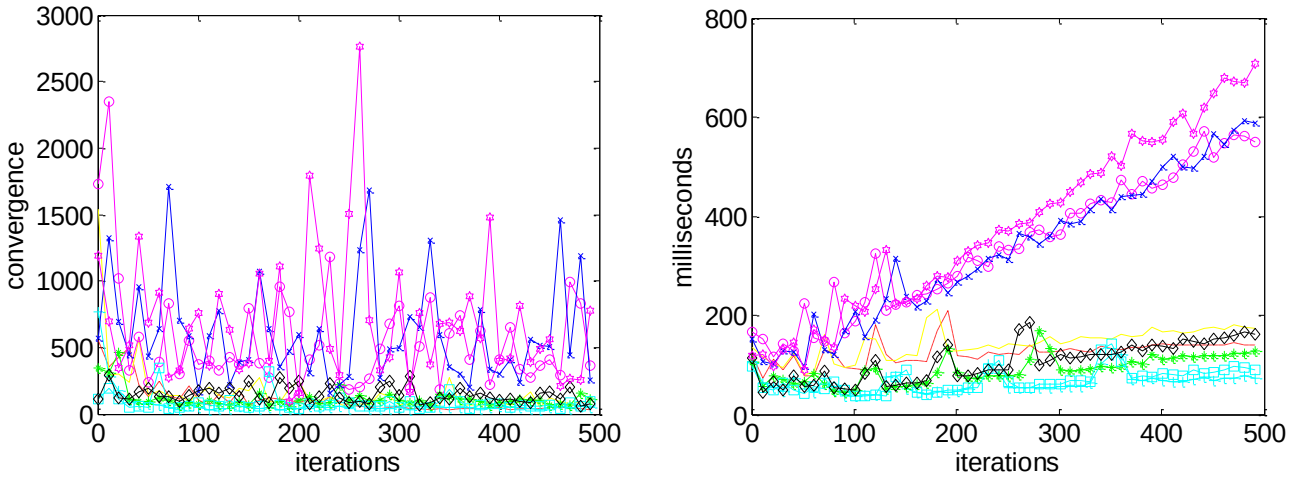


(e) L0.9 E.01 (f)L0.9 E0.5 (g)L0.9 E0.9

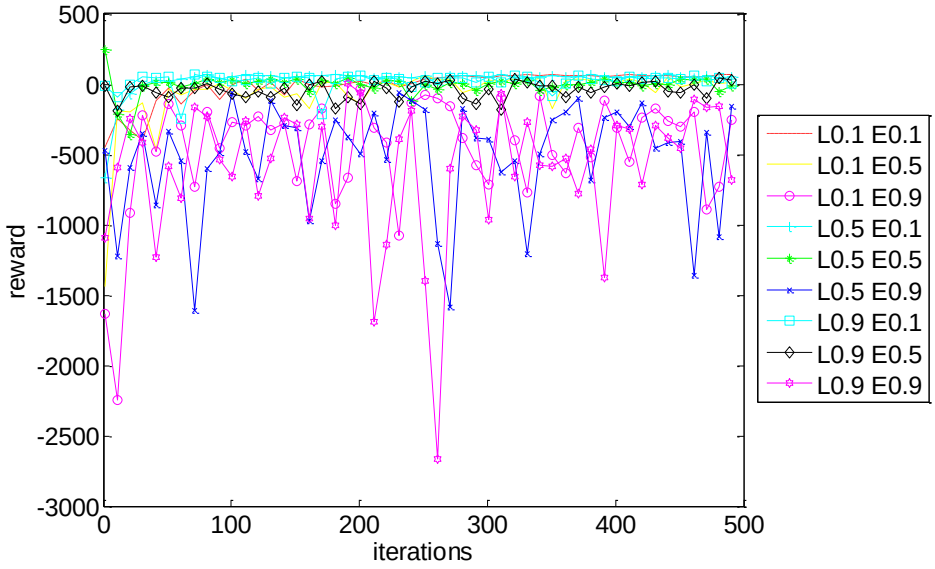
Fig. 6 Q-learning with different learning rate (L) and exploration rate (E)

Fig.6 shows the utilidy and actions for cases with different combinations of learning rate and exploration rate. Firstly, only cases (a) – (f) give the right path (yellow line) to the destination. All the other cases may give high utility for every lattice, however some actions on the right routines are wrong.

For exploration rate, higher value results in higher utility for the domain, as we can see that the whole domain is in deeper blue. That is reasonable because case with higher exploration rate tends to wander in the whole domain. But that could also cause agent to wander the domain randomly without giving the otpimum path.



(a) Convergence (b) Time cost

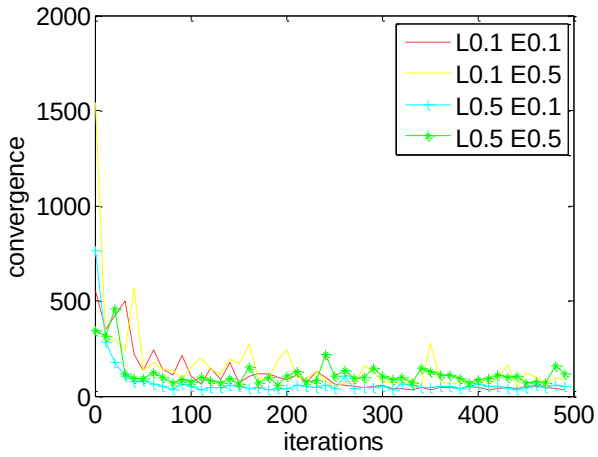


(c) Total reward

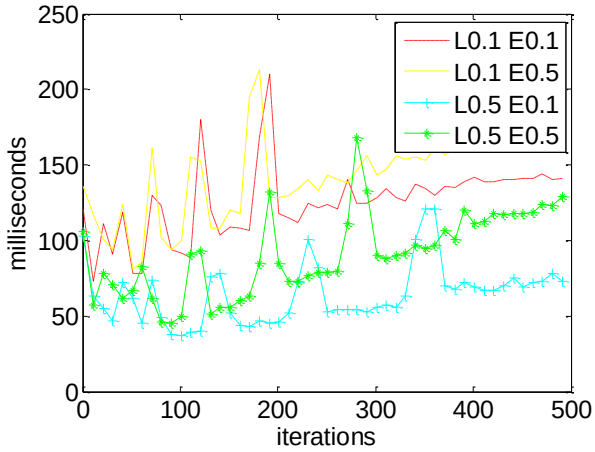
Fig.7 Comparison of Q-learning with different learning rate and exploration rate

As shown in Fig.7, for cases with high expolration rate (0.9), the convergence and reward curves show much fluctuations due to the high tendcy to take the “wrong” action. Also, it takes longer for cases using high exploration rate.

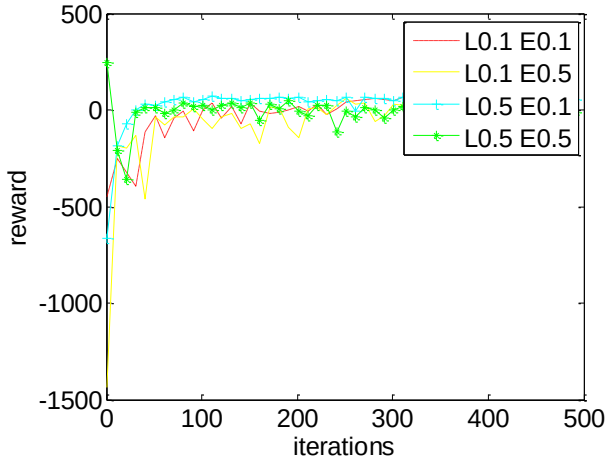
To make a more clear comparison we plot the best four cases in the Fig.7. As we can see that the case with learning rate = 0.5 and exploration rate = 0.1 is the best one regarding to convergence, time cost as well as reward.



(a) Convergence



(b) Time cost



(c) Total reward

Fig.8 Comparison of Q-learning with different learning rate and exploration rate

Summary:

In this report we applied 3 reinforment algorithms to 2 MDP problems (mazes). The first maze is quite strightforward with 5*5 grids. The second one is much harder with 15*15 grids and deadends, circles.

Value iteration: With discount factor = 0.9 is used for both mazes, value iteration solves the problems quite easily and gives the right actions on all grids after a few iterations.

Policy iteration: It is pretty similiar to value iteration in terms of the convergence and reward for both MDP. The time cost is lower dealing the complex one.

Q-learning: It requires much more iterations to converge comparing to value iteration and policy iteration, however the time cost for each step is less. In general, it could also solve the MDP problems with enough iterations. Learning rate and exploration rate are two important parameters for solving. A high learning rate (0.9) can never gives the optimal path. A high exploration rate (0.9) will cost more time and divergence. In current cases, the easy grid world could be easily solved. After tuning, we found that learning rate = 0.5 and the exploration rate = 0.1 might be the best choice for the hard grid world problem.

Reference:

[1] https://en.wikipedia.org/wiki/Markov_decision_process

[2] https://en.wikipedia.org/wiki/Q-learning