MANAGEMENT SCIENCE
DOI 10.1287/mnsc.1080.0870ec
pp. ec1–ec19



e-companion

ONLY AVAILABLE IN ELECTRONIC FORM

# Electronic Companion—"Effect of Delays on Complexity of Organizational Learning" by Hazhir Rahmandad

*Management Science*, DOI 10.1287/mnsc.1080.0870.

**Electronic Companion for:**

# Effect of Delays on Complexity of Organizational Learning

**Hazhir Rahmandad, hazhir@vt.edu**



The e-companion material includes a detailed example to elaborate on the learning mechanisms, instructions for downloading and running the learning models, process for finding optimal policies, graph of learning time as a function of delays, analysis of dominant strategies, sensitivity to exploration, details of analysis on initial cognitive maps and adaptation in changing environments, and detailed results of sensitivity to parameters and payoff functions. The general discussions are reported in the paper and the e-companion lays out the detailed procedures and results for different analyses.

## 1) Learning Example

As discussed in the paper, the simulated organization at each period takes an action by randomly selecting between the N=3 available actions, based on probabilities specified in equations 3 and 4. The payoff, $r_t$, is then realized based on equation 1, and the updating of Q function according to equation 2 happens afterwards. Through this process of exploration and adjustment, the organization learns about the value of different state-action pairs and moves towards the optimum strategy. In the base-case, the optimum strategy consists of cycles of a=0 for K consecutive periods, followed by one period of a=2. If an organization follows this optimum policy, it will receive payoffs of 0 for K periods followed by a period with the payoff of 1. As a result the maximum possible average payoff for an organization is 1/(K+1).

An example would elaborate the workings of the model better. Lets assume we are working in the delay condition of K=3 and initially the organization is in the state (0,2,1), that is, it has taken the actions 1, 2, and 0 in the periods t-1, t-2, and t-3. The current state-action pairs available to the organization include [(0,2,1),0], [(0,2,1),1], and [(0,2,1),2], which we assume to have Q values of 0.3, 0.7, and 2 respectively. With $e_w$=1, the probabilities of taking the three actions would be 0.1 ($= \frac{0.3^1}{0.3^1 + 0.7^1 + 2^1}$), 0.23, and 0.67. The organization randomly takes the action a=2 and observes the payoff of 1. The new state of the organization is (2,1,2) and we assume the Q values of state-action pairs from this state are 1, 1.5, and 3. The organization now updates the Q value of the last state-action pair (the calculations assume γ=0.5):

$$Q_{t+1}((0,2,1),2) := (1-\alpha)Q_t((0,2,1),2) + \alpha(r_t + \gamma \max_{a'} Q_t((2,1,2),a')) = (1-\alpha)2 + \alpha(1+3\gamma) = 2.5$$

## 2) Model Implementation

The model is implemented in Anylogic ™ simulation environment, version 5.5. The simulation models for different delay conditions and different experiments conducted in the paper are available in a .zip file from the author's website at:

http://filebox.vt.edu/users/hazhir/www/papers/Q-Learning-DelayedBehavior-Sep07.zip

You can also find a Java applet with the learning model for delay condition of K=4 at:

http://filebox.vt.edu/users/hazhir/www/papers/Q-Learning-DelayedBehaviorDelay4 Applet.html

To run the applet your browser needs to be able to run Java code. Some Web browsers have built-in capability of displaying Java™ applets; some require Java™ plug-in to be installed. The installation is done only once, before you view the first applet. Normally, the browser automatically detects whether the plug-in needs to be installed and offers to download and install the plug-in for your particular platform and browser version. If this does not happen, and you are unable to view Java™ applets, please consult your browser/OS manufacturer.

To open the Anylogic model source code, download and install the Anylogic™ software (You need the version 5 of the software. A free 15-day trial is available from http://www.xjtek.com/support/download/evaluation5/). Unzip the file "Q-Learning-

DelayedBehavior-Sep07.zip", available from the above website, in one folder. Open any of the simulation models for different delay conditions to replicate different experiments reported in the paper. The list of all objects and procedures in the Q-learning model is displayed in the left hand column; select any object to inspect its formulation. Run the model by clicking on the run button (or pressing F5), which compiles the model and brings up an interface. You can inspect the behavior of the model both through the interface accompanying the model (which is similar to what you see in the Java™ applet), or by browsing different variables and graphing them in Anylogic™ run mode. To do the latter, go to "root" tab in run time mode, where you can see all model variables and can inspect their runtime behavior.

You can also re-run the model with the parameter settings of your interest. Just set the new parameters (including the 4 parameters of the model and the number of replications), then click on the run button in the applet.

## 3) Optimal Policy

The optimal policy can be found using a dynamic program or a computational framework. Given the continuous nature of the task we use the latter. In fact the Q-Learning algorithm, when left to openly explore different state-action pairs, is guaranteed to converge to correct values for different state-action values (Watkins and Dayan 1992). Those values can then be used to graph what action is the best from each state (similar to figure 4 in the paper). The closed cycle that shapes through this process points to the optimum policy.

## 4) Convergence Time as a Function of Delays

The graph below follows the discussions in the paper regarding the relationship between the convergence time and the length of delays.

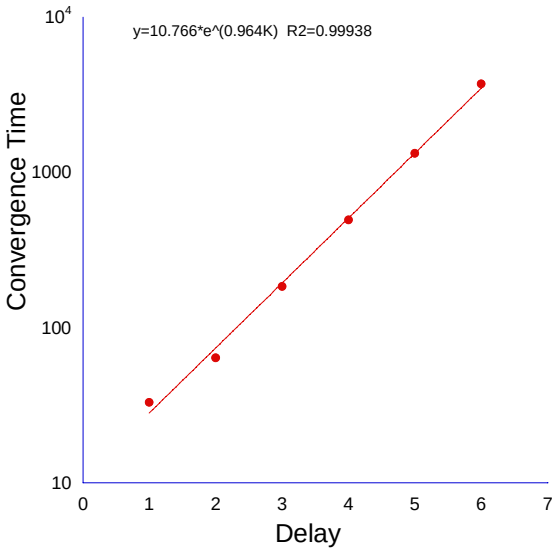
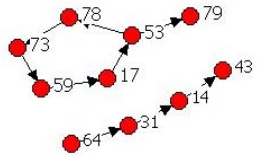


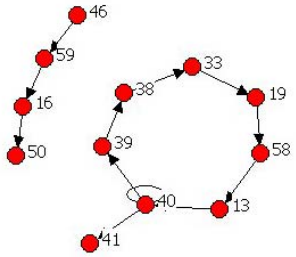

**Figure EC.1 The convergence time as a function of delay length. The best fitting line on a logarithmic scale along with the equation for that line is also shown.**

## 5) Analysis of Dominant Strategies

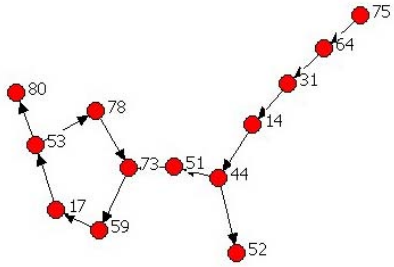
To learn more about the emerging dominant strategies we run additional analysis with delay size of K=4. Simulating the learning organizations for a longer time to allow for observation of steady state behavior (40000 periods) we record the number of visits to different state-action pairs. We then map the dominant strategies by noting what action-state pairs have been visited most. Specifically, we sort the action-state pairs according to the number of visits they received in the simulation and pick out state-action pairs with more than 1000 visits, or until we have reached a dominant cycle, whichever comes faster. Figure EC.2 shows the emerging dominant strategies in ten randomly selected simulations. In all cases a dominant strategy cycle has emerged where all state-action pairs on the cycle have been visited more than 1000 times. Note that only 5.3% of state-action pairs are visited this frequently (in a sample of one hundred simulations) and the majority (78%) of state-action pairs are visited less than ten times. One of the simulations (number 9, highlighted with patterned states) has converged to the optimal strategy.


78
79
53
73
17
59
43
14
31
64

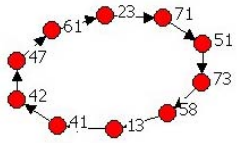
1


46
59
16
50
33
38
19
39
58
40
13
41

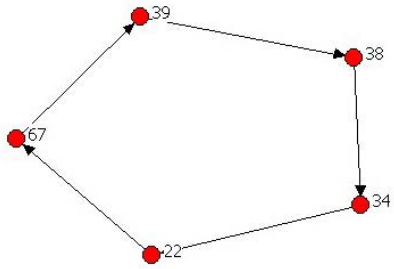
2


75
64
31
80
78
14
53
73
51
44
17
59
52

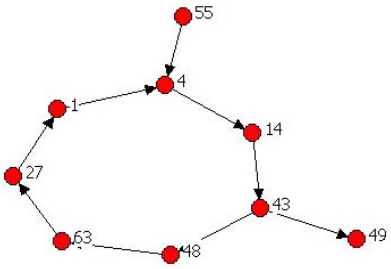
3


23
71
61
51
47
73
42
58
41
13

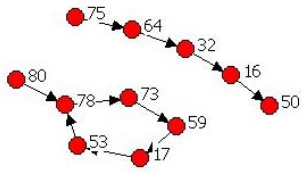
4


39
38
67
34
22

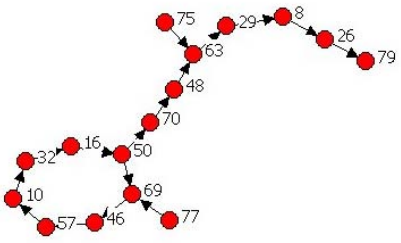
5


55
4
1
14
27
43
63
48
49

6


75
64
32
16
80
50
78
73
59
53
17

7


75
29
8
26
63
79
48
70
16
32
50
10
69
57
46
77

8

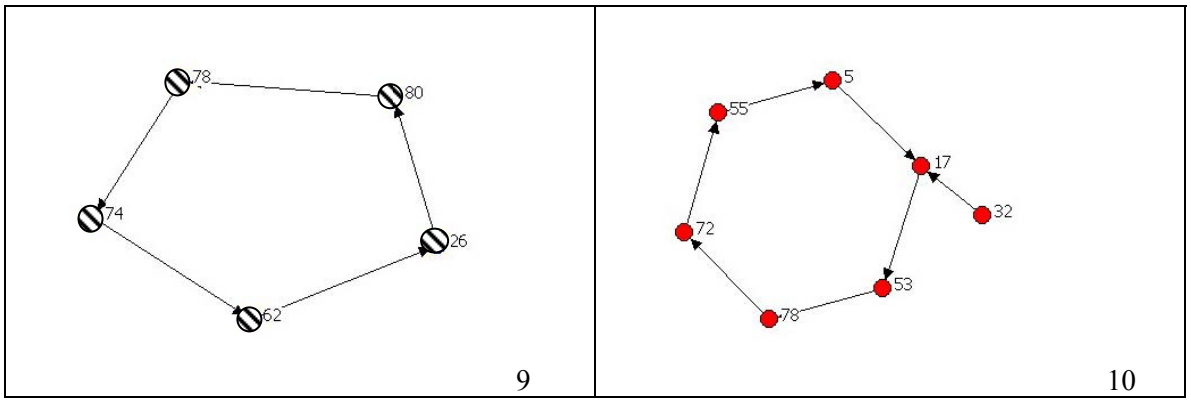



**Figure EC.2 Sample of dominant strategies in simulations with K=4.**

Two competing factors determine the size of the dominant strategy cycles. On the one hand, shorter strategy cycles are promoted by the stochastic nature of exploration. Dominance of a cycle requires the organization to remain on that path for multiple rounds, form internally consistent cognitive maps about the superiority of that strategy to other alternatives, and lock into cycling through that strategy later in the life cycle of the organization. Every step on a cycle harbors the possibility of deviation from that cycle and therefore the breakdown of the strategy. Therefore longer cycles are more prone to such random deviations, and emerge less frequently. On the other hand, the number of potential cycles that could exist increases with the size of cycles. There are only three cycles of size one (a cycle of size one is really a single dominant action that will take the organization to the starting state; in this example the state-actions ((0,0,0,0), 0), ((1,1,1,1), 1), and ((2,2,2,2), 2)). Another three potential cycles of size two can be found (e.g. ((1,2,1,2),1) and ((2,1,2,1), 2)). The number of cycles of higher lengths increases exponentially, where each increase in the potential length increases the number of potential cycles N=3 times. Therefore there are many more lengthy cycles that could potentially emerge to dominate the strategy. Besides these two factors, the payoffs obtainable from different strategic cycles will impact their strategic value. Our experiments suggest that as a result of these factors most dominant cycles are of sizes slightly bigger than the delay length. For example in the cases reported above there is only one dominant cycle of length one which is reinforced by a cycle of length seven (case 2). In half the cases the dominant cycles are of length five, including the optimum strategy (Cases 1, 3, 5, 7, 9); and a dominant cycle of length six and three of size seven are observed. Only one longer cycle (length ten) has come to dominate in our sample.

Overall, the emergent dominant strategies are quite diverse in the exact states they visit, even though they are all emerging in simulated organizations that are learning about the same problem with exactly same parameters. The evaluation bias creates a path dependent process of reinforcing arbitrary strategies that seem to give a good payoff initially. These strategies however have some common characteristics: they are dominated by cycles of length close to the delay size. Smaller cycles fail to generate positive payoffs in our setting (except for the cycle of ((1,1,1,1), 1), which did emerge as dominant in one of our simulations) and therefore are not reinforced. Longer cycles breakdown given the stochastic explorative moves that take the organization away from those cycles.

The complexity of the dominant cycles speaks to the emergence of successful organizational routines. In dealing with temporally complex problems, organizations converge to routines that are lengthy enough to allow for observation of positive payoffs. Longer routines, however, are harder to emerge given the chances that the organization deviates in any of the multiple steps on the routine, and therefore fails to update its cognitive map to appreciate the full value of that routine.

## 6) Exploration Exploitation Trade-off

We repeated the experiments using an exploration function in which for the first 60% of simulation time the algorithm chose between different (already tried) actions with the same probability, and then switched to using the exploration function dictated by equations 3 and 4. We used the same parameters; $\gamma$=0.5, $e_u$=0.5, $\mu$=15; across different delay conditions for the purpose of these simulations. Large mu value ($\mu$=15) is selected so that exploitative action is favored once exploration is completed. In case of additive capability in payoff (See the sensitivity to payoff functions below), larger mu values are needed for getting the optimal performance in light of closeness of the real values of different states. Table EC.1 reports the performance of learning model under these settings. Convergence time is tied to the time when the switch from explorative to exploitative action happens (60% of simulation time) and therefore is not reliable.

**Table EC.1- Metrics of performance and heterogeneity are reported for different delay conditions**

| **Delay Size** | **1** | **2** | **3** | **4** | **5** | **6** |
|---|---|---|---|---|---|---|
| **Convergence Time** | 86 | 270 | 604 | 2846 | 5848 | 15541 |
| **Optimal Fraction** | 0.951 | 0.973 | 0.973 | 0.963 | 0.886 | 0.841 |
| **Performance Gain** | 99.3 | 99.6 | 99.5 | 99.7 | 97.4 | 96.0 |
| **Performance Heterogeneity** | 0.11 | 0.09 | 0.14 | 0.20 | 0.61 | 1.01 |
| **Simulation Time** | 150 | 450 | 1000 | 4000 | 8000 | 24000 |

Other exploration functions exist that could also find the optimum strategy. In fact as long as all the states are visited a large number of time as a result of exploration (so that their values are estimated with reducing potential for bias), the algorithms can guarantee convergence to optimal policy (Sutton and Barto 1998).

## 7) Impact of Initial Cognitive Maps on Adaptation

In this analysis we examine the impact of initial cognitive maps on organizational adaptation. We consider four scenarios. To set the initial values in three of these scenarios we use the following equation:

$Q_0(s,a)=(1-C)\text{Uniform}(0,M)+CQ^*(s,a)$ (EC.1)

Here $Q_0$ and $Q^*$ are the initial and correct value of Q function (correct values of Q function satisfy the bellman optimality condition, and can be obtained by running the algorithm in exploratory mode for a long time). M is the average correct value for the Q function, $M = \frac{\sum_{\forall s\in S, a\in A} Q^*(s,a)}{|S.A|}$, and is used to normalize the random term in initial value of Q function and C is a constant between 0 and 1 that controls the weight of randomness in initial values. We consider the following four scenarios: a) When all initial Q function values are one (which is higher than the real Q-value for any state-action under current parameter settings) b)When initial mental maps are completely random (C=0) c)When initial mental maps are partially correlated with correct maps (C=0.5) d)When initial mental maps are correct (C=1)

The summary of metrics is reported in table EC.2. The overall performance is very similar across all these conditions, except for much faster convergence to optimal strategy when correct mental maps are originally specified (case d). Note that with $\mu$=5 (base case parameter settings are used), enough explorative deviation from the optimal path takes place that even with correct cognitive map, the optimal policy is not always pursued and random deviations increase performance heterogeneity significantly.

**Table EC.2 The performance of simulated organizations with different initial cognitive maps**

| Case | a) Fixed initial Q =1 | b) Random C=0 | c)Correlated C=0.5 | d) Correct C=1 |
|---|---|---|---|---|
| **Convergence Time** | 1695 | 1830 | 1712 | 343 |
| **Optimal Fraction** | 0.109 | 0.120 | 0.111 | 0.117 |
| **Performance Gain** | 86 | 84 | 86 | 83 |
| **Performance Heterogeneity** | 1.45 | 1.58 | 1.46 | 1.46 |

## 8) Adaptation in Changing Environments

In this analysis we compared two scenarios for K=4. In the first scenario we simulated 1000 organizations, each starting from one random payoff function, learning with base case parameters until time 2000, when an environmental shift changes the payoff function randomly. The organization continues to use its old cognitive map to guide its adaptation and exploration in this new environment until time 4000. The initial payoff function is randomly chosen for each organization according to the general payoff function introduced in s*ensitivity to other multiplicative payoff functions* section (S10 below), by randomly selecting $P_i$ values (0 or 1) for i=0…4. The environmental shift is modeled with another random selection of payoff function at time 2000. The initial random selection of payoff function allows for comparison of performance before and after the change.

In the second scenario we follow the exact same procedure, except that the exploration is reset at the time of environmental shift. That is, noting the shift, the organization decides to become more open to alternatives it has already tried and categorized as ineffective. This was

implemented by using a new equation for tendency to explore known alternatives (equation 5 in the paper). Thus, at time 2001, equation 5 switched to:

$$e_w = \frac{\ln(t-2000)}{\ln(T-2000)}\mu \qquad \text{(EC.2)}$$

The result is a reset of exploration to its starting value, before inertia again builds up and the organization reduces its explorative tendencies.

The results of this analysis are reported in the graphs below. Figure EC.3-a shows the first scenario (no change in exploration pattern) and EC.3-b shows the scenario with resetting of exploration. The specific performance metrics are also reported in the graphs.

The similarity of performance under the two scenarios suggests that renewing exploration is not able to overcome the challenge posed by old cognitive maps. These maps are internally consistent, that is, they have emerged to value some closed loop, often suboptimal, strategy cycle. The additional exploration in scenario b is not enough to overcome the evaluation bias trap created by these old cognitive maps.

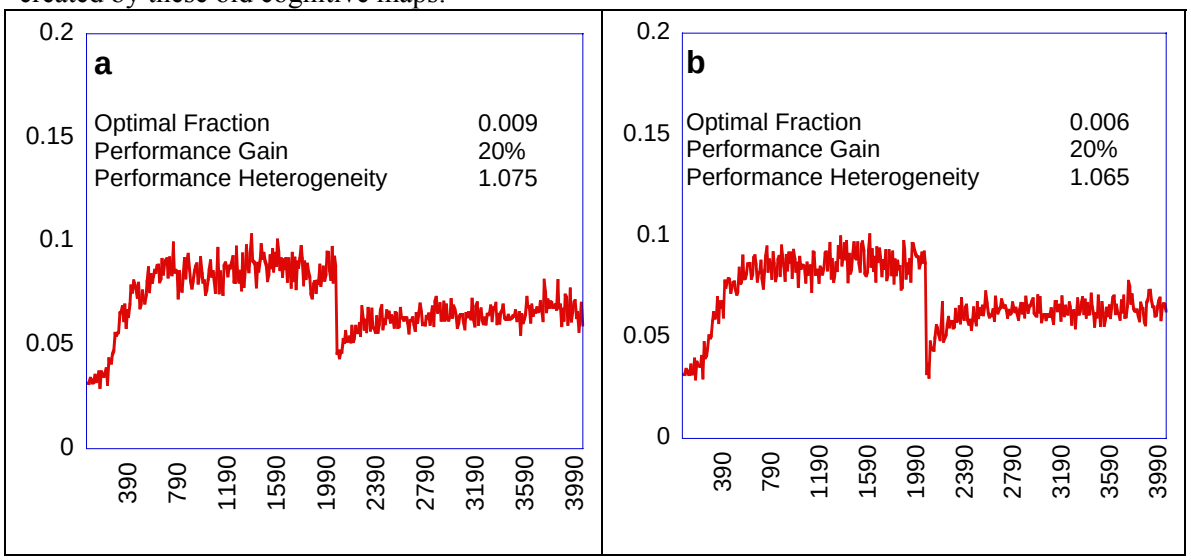



**Figure EC.3 The average performance of the organizations when an environmental shift changes the payoff function in the middle of the simulation. Different performance metrics are reported on the graphs. a) When the organization continues with the old exploration tendency b) When the exploration is set back to the level of a new organization. The curves are very similar throughout the simulation time. Resetting the exploration does not help the organization overcome the traps created by faulty cognitive maps. The old maps quickly dominate the exploration pattern in the new**

**environment and therefore lead to performance levels worse than those achieved in the first environment (because the maps were partially effective in the old environment, but have even less value in the new one).**

## 9) Sensitivity to Learning Parameters

We conducted two sets of sensitivity experiments with respect to learning parameters.

***Parameter variation-*** In the first set we analyzed how sensitive the learning performance is to the parameter values of the learning algorithm. For this purpose we changed the three major parameters of the learning algorithm according to table EC3. Wide parameter ranges are chosen to get a comprehensive perspective (Note that $\gamma$ and $e_u$ should be between 0 and 1). We changed one parameter at a time, keeping the rest at their base value. For each parameter setting we simulated 1000 organizations and averaged their final performance. Figure EC.4 reports the results.

**Table EC.3 The parameter ranges for the sensitivity analysis on the parameters of learning algorithm. Sensitivity step specifies the increments of change in the parameter between the maximum and the minimum values.**

| Parameter | Minimum | Maximum | Sensitivity Step |
|---|---|---|---|
| Unknown Exploration ($e_u$) | 0.05 | 0.95 | 0.1 |
| Discount Rate ($\gamma$) | 0.05 | 0.95 | 0.1 |
| Exploitation ($\mu$) | 1 | 10 | 1 |

The results suggest that the learning performance is largely insensitive to the parameter values. In fact for medium levels of all the parameters, the behavior is almost identical. Very large values of "Discount Rate" tend to increase the difference in value of explored and unexplored pathways, and reinforce evaluation bias. They therefore reduce performance by promoting quick abandonment of stepping-stone states that do not payoff quickly. Low levels of Exploitation parameter also reduce long-term performance by not allowing the organization to use the evolving cognitive maps. The parameter values used in the base case experiments throughout the paper use robust values in the mid range of this spectrum.

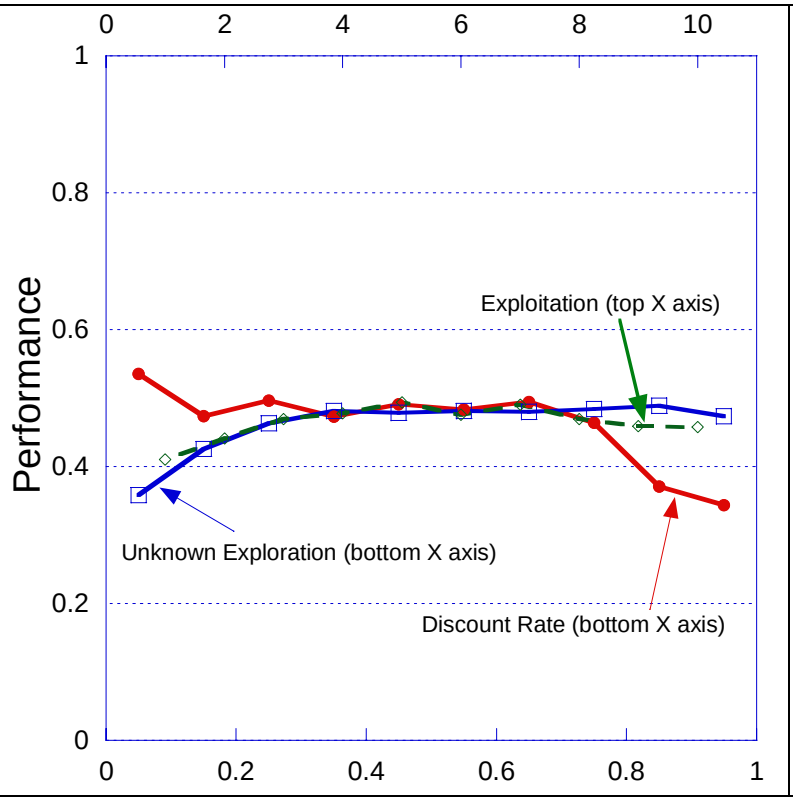



**Figure EC.4 The average final performance as a fraction of maximum achievable for the 1000 simulated organizations (K=3) in parameter variation experiments. The bottom X axis specifies the values of "Unknown Exploration" ($e_u$) and "Discount Rate" ($\gamma$). The top axis reports "Exploitation" ($\mu$) values.**

***Parameter optimization***- In the second set of experiments we optimized the parameters of the model to maximize the final learning performance over the long-run in different delay conditions. The logic behind this conceptualization is that by comparing the best learning performance in each delay condition, we are closer to a fair comparison between different models because similar parameter settings may have different qualitative implications for learning under different delay conditions. Moreover, by optimizing the parameters we err on the side of showing conservative impacts of delays on learning, and thus obtaining stronger results for the main hypothesis in the paper. Of course, such optimization is not behaviorally realistic (real organizations can not access such optimization results prior to interacting with an environment) and therefore should be considered more as a test of robustness of the overall results than a behaviorally plausible scenario.

Table EC.4 reports the parameters found through the maximization of average final interval performance for each condition over 100 simulations, using OptQuest

optimization engine. Values of $e_u$ suggest that the organization is fairly aggressive in exploring new actions. In the presence of the same number of new and already explored actions, new actions are explored with a chance higher than 0.3. $\mu$ values suggest that by the end of the simulation time the organization acts aggressively to exploit the best alternatives found so far ($\mu>1$) and therefore the state-action with highest Q(s,a) is very likely to be chosen at each step towards the end of simulation.

**Table EC.4- The simulation parameters for the analysis of impact of delays on learning. The three parameters defining the learning algorithm are derived to have the best learning performance in that setting. Simulation time is the length of simulation used and final interval is the length of the final interval of simulation used to find the *final performance percentage*s. The final intervals are selected to reduce noise in final performance measures, without sampling for the performance metric from the periods before steady-state performance.**

| **Delay Size** | **1** | **2** | **3** | **4** | **5** | **6** |
|---|---|---|---|---|---|---|
| $\gamma$ | 0.02 | 0.25 | 0.02 | 0.42 | 0.21 | 0.16 |
| $e_u$ | 0.30 | 0.53 | 0.65 | 0.30 | 0.64 | 0.64 |
| $\mu$ | 7.48 | 6.87 | 7.48 | 4.19 | 5.88 | 6.63 |
| **Simulation Time** | 100 | 150 | 300 | 1000 | 3000 | 10000 |
| **Final interval** | 30 | 30 | 60 | 100 | 200 | 300 |

Table EC.5 and Figures EC.5 and EC.6 report the results of this analysis. Overall, the results show very similar patterns as those observed in the base case. The overall performance is slightly better but longer time is needed to achieve it (mainly because the objective function had no penalty for waiting to find the optimum performance). The relationship between convergence time and delay length is also consistent with the original hypothesis of the study and has an exponential coefficient of 2.8 (very close to theoretically predicted value of 3), See Figure EC.6.

| Table EC.5- Metrics of performance and heterogeneity are reported for different delay conditions | | | | | | |
|---|---|---|---|---|---|---|
| **Delay Size** | **1** | **2** | **3** | **4** | **5** | **6** |
| **Convergence Time** | 64 | 97 | 252 | 870 | 2690 | 9065 |
| **Optimal Fraction** | 0.764 | 0.3 | 0.055 | 0.025 | 0.023 | 0.013 |
| **Performance Gain** | 81 | 47 | 45 | 28 | 26 | 28 |
| **Performance Heterogeneity** | 0.49 | 0.87 | 1.05 | 1.13 | 1.33 | 1.87 |

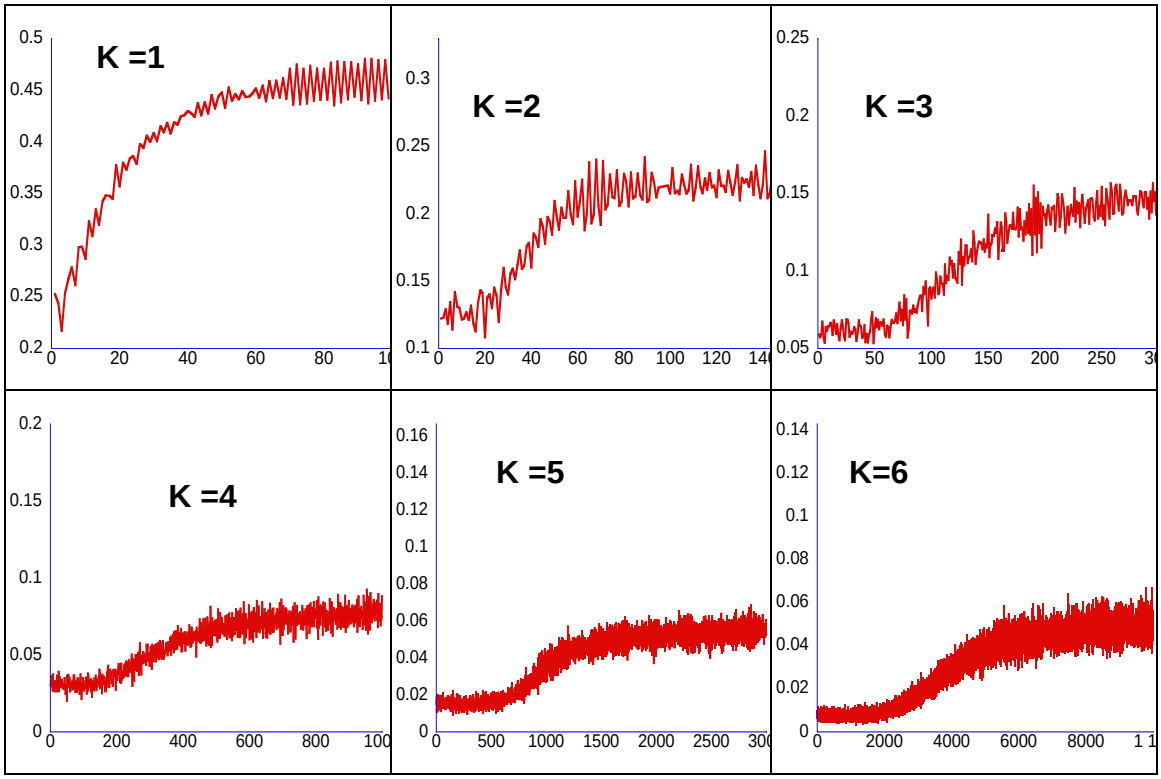



**Figure EC.5- Average performance of the learning organizations for different delay conditions. Averages of 1000 simulated organizations are shown. Y axes are extended to the optimum long-term performance. Note that X axes range from 100 to 10,000 periods.**

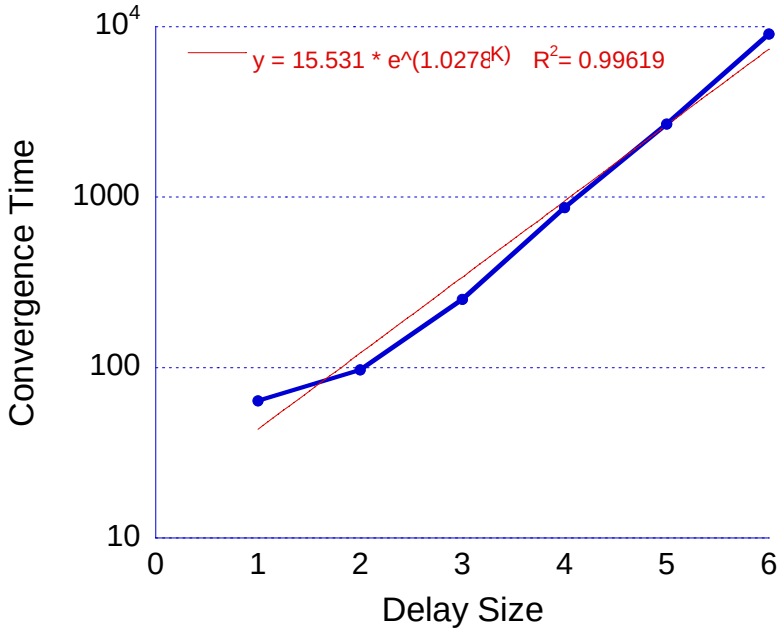



**Figure EC.6- Convergence time as a function of delay size, graphed on a logarithmic scale. Also shown are the best fit to the curve, the fit equation, and the R-square for the fit.**

## 10) Sensitivity to Payoff Function

***Additive Payoff Function-*** The first set of experiments for sensitivity to payoff function use the following payoff function:

$$r_t = c_K a_t \sum_{i=1}^{i=K} (N - a_{t-i} - 1) \qquad \text{EC.3}$$

$$c_K = \frac{1}{K(N-1)^2} \qquad \text{EC.4}$$

The main characteristic of this function is the additive function (rather than multiplicative one) between actions from the previous periods. The payoff at any period is bounded between 0 and 1. The optimal performance with this function can be achieved from following the policy of a=0 for K periods followed by K periods of a=2[1], which will lead to the average payoff per period of $\frac{K+1}{K(N-1)^2}$. Note that for K=1 this function is identical with the base case function reported in the body of paper, therefore we only repeat the analysis for K=2-6.

In order to have a stronger test for impact of delays on learning with this payoff function, we used the optimized parameter form (rather than same parameters for all models; similar to

[1] An additional optimal policy exists for K=3, where one period of a=2 is followed by one period of a=0.

analysis reported under sensitivity to parameter values). Table EC.6 represents the parameter values used for the simulations in this setting (similar to table EC.4 above). Figure EC.7 depicts the overall performance and table EC.7 reports the performance metrics for this set of experiments.

**Table EC.6- The simulation parameters for the analysis of impact of delays on learning. Longer simulation times were chosen to allow for potentially later convergence to optimal policy.**

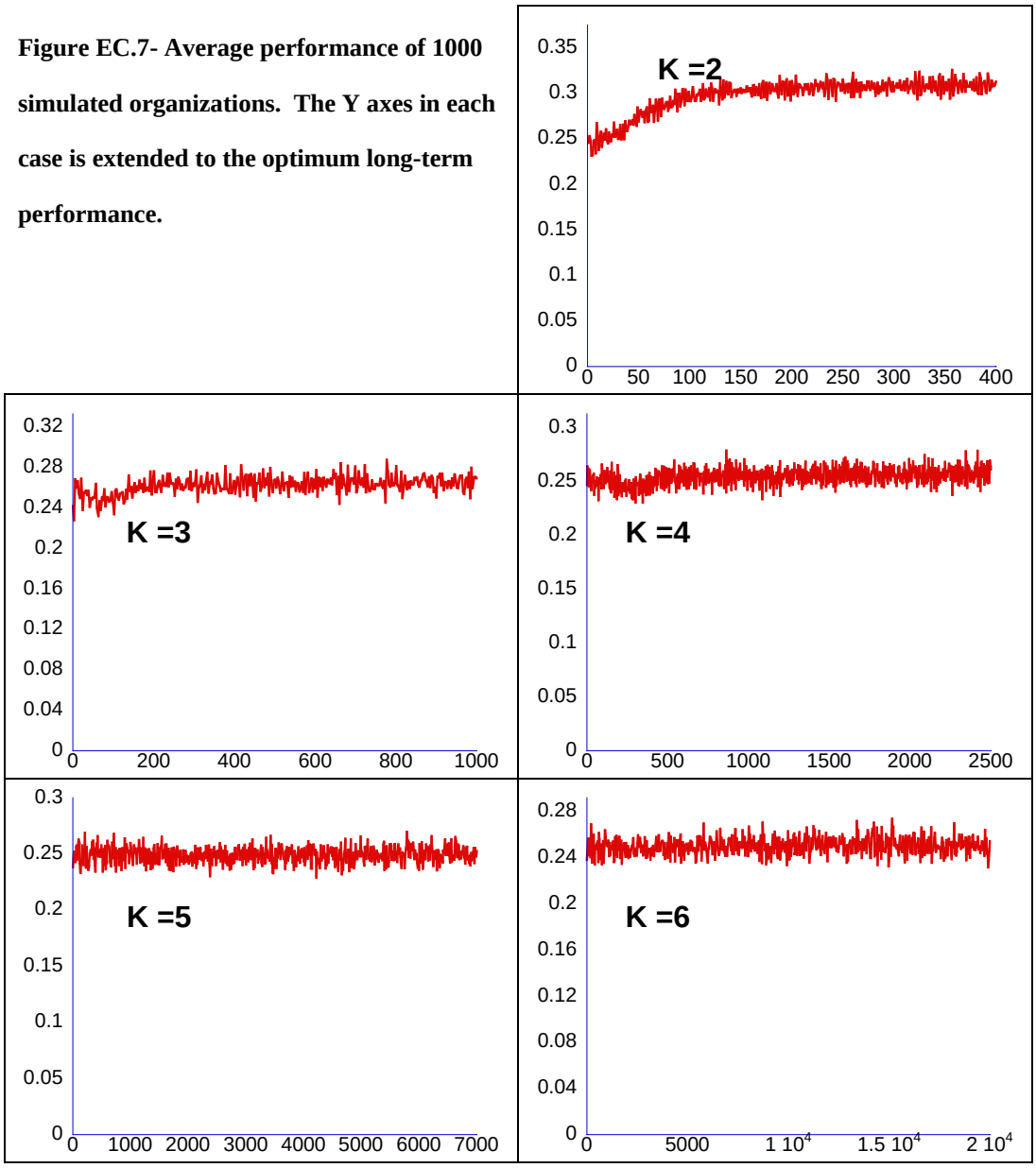
| **Delay Size** | **2** | **3** | **4** | **5** | **6** |
|---|---|---|---|---|---|
| $\gamma$ | 0.49 | 0.82 | 0.87 | 0.73 | 0.54 |
| $e_u$ | 0.31 | 0.88 | 0.99 | 0.80 | 0.64 |
| $\mu$ | 8.58 | 5.17 | 4.47 | 0.23 | 0.19 |
| **Simulation Time** | 450 | 1000 | 4000 | 8000 | 24000 |

.

**Table EC.7- The performance metrics for sensitivity analysis on payoff function. Zero values for convergence time suggest that performance for K=5 and 6 has not changed significantly and therefore the set of organizations converge to the initial performance even after a few thousand periods of experimentation**

| **Delay Size** | **2** | **3** | **4** | **5** | **6** |
|---|---|---|---|---|---|
| **Convergence Time** | 111 | 91 | 271 | 0 | 0 |
| **Optimal Fraction** | 0.009 | 0.031 | 0.008 | 0.001 | 0.000 |
| **Performance Gain** | 49.1 | 19.1 | 8.0 | -0.2 | 6.2 |
| **Performance Heterogeneity** | 0.86 | 0.89 | 1.38 | 1.42 | 2.26 |

**Figure EC.7- Average performance of 1000 simulated organizations. The Y axes in each case is extended to the optimum long-term performance.**




***Sensitivity to other multiplicative payoff functions***- A general category of multiplicative payoff functions can be defined as:

$$r_t = \frac{\prod_{i=0}^{i=K}((N - a_{t-i} - 1)^{p_i} a_{t-i}^{(1-p_i)})}{(N-1)^{K+1}}, \quad P_i \in \{0,1\} \qquad \text{(EC.5)}$$

With different values for $P_i$, one can configure this function so that past actions have different types of impacts on the current payoff. For example the base case payoff function used in our study used $P_0$=0 and $P_i$=1 for the rest of the actions (or (0,1,1,1,..) payoff function).

Despite its generality, all different payoff functions of this form can be summarized in a few distinct arrangements for the purpose of our study. These arrangements are dictated by the shape of strategy cycles that can emerge under each payoff function. These cycles create multiple symmetries across possible payoff functions. For example (1,1,1,1,0) is the same as (1,1,1,0,1) because over the long run only the sequence of actions matter. It is also the same as (0,0,0,0,1) because a renaming of variables will simply bring us to this new configuration, when it comes to learning. Following a similar logic for K=4, only four such arrangements exist: PF3=(1,1,1,1,1), PF0=(1,1,1,1,0), PF2=(1,1,1,0,0), and PF1=(1,0,1,0,0). Any other multiplicative function of the general form in equation EC.5 is identical in terms of learning behavior to one of these four configurations.

The PF0 configuration is already studied in the base case, therefore in the sensitivity analysis we run the model, using the base parameter values, for the other three combinations. Table EC.8 reports the results. Given the base case parameter ($\mu$=5), in the PF3 condition, even after arrival at the optimum policy, enough random explorative moves are made that the optimal fraction does not exceed 0.19 and performance heterogeneity remains very high.

**Table EC.8- The performance metrics for sensitivity to different multiplicative payoff function forms. Results are reported for 1000 organizations in K=4 condition. Payoff function forms (PF1, PF2, PF3) are defined above.**

| **Payoff Function Type** | **PF1** | **PF2** | **PF3** |
|---|---|---|---|
| **Convergence Time** | 419 | 369 | 2601 |
| **Optimal Fraction** | 0.034 | 0.027 | 0.19 |
| **Performance Gain** | 34% | 32% | 90% |
| **Performance Heterogeneity** | 1.31 | 1.23 | 2.42 |

## References:

Sutton, R. S. and A. G. Barto (1998). Reinforcement Learning: An Introduction. Cambridge, The MIT Press.

Watkins, C. J. C. H. and P. Dayan (1992). "Q-Learning." Machine Learning **8**(3-4): 279-292.