

**A comparative study of Roth-Erev and Modified Roth-Erev
reinforcement learning algorithms for uniform-price
double auctions**

Mridul Pentapalli
Graduate Student (MS Comp. Sc.)
Iowa State University, Ames, IA
March 2008

Presentation Outline

- Overview of the M.S. thesis
- The tested learning algorithms and their importance
- Specific form of the three algorithms
- An example of computational results leading to a greater understanding of the algorithms
- Results of some computational experiments in multi-agent learning contexts

Overview of the M.S. thesis

- Chapter 2: Description of the tested algorithms
- Chapter 3: Five test cases used to compare the learning algorithms
- Chapter 4: Experimental results
- Chapter 5: Mathematical analysis of the algorithms
 - Limits of learning
- Chapter 6: Conclusions and future work
- Appendices
 - Double auction terms and definitions
 - Literature review
 - Implementation of the testbeds

Main points of the thesis

- Computational experiments provide a powerful tool for understanding the behavior of learning algorithms in multi-agent contexts.
- ‘Heat maps’ can help to visualize outcome sensitivities in computational experiments involving intensive parameter sweeps.
- Computational results can point to mathematical theorems

Roth-Erev Reinforcement Learning Algorithms

- Originally developed by A. Roth and I. Erev
(*Games and Economic Behavior* 1995, *American Economic Review* 1998)
- Modified Roth-Erev reinforcement learning (MRE RL) algorithm developed by J. Nicolaisen, V. Petrov and L. Tesfatsion
(*IEEE Transactions on Evolutionary Computation* 2001)
- Variant Roth-Erev RL (VRE RL) algorithm developed by J. Sun and L. Tesfatsion
(*Computational Economics* 2007)

Roth-Erev RL Algorithm

$$q_j(t + 1) = [1 - r]q_j(t) + E_j(e, N, k, t)$$

$$E_j(e, N, k, t) = \begin{cases} \pi_k(t)[1 - e] & \text{if } j = k \\ \pi_k(t)\frac{e}{N-1} & \text{if } j \neq k \end{cases}$$

$q_j(0)$ is the initial propensity of action j at time $t = 0$ (aspiration level)

$q_j(t)$ is the propensity of action j at time t

$\pi_k(t)$ is the reward obtained for taking action k at time t

r is the recency parameter

e is the experimentation parameter

N is the number of actions

From Action Propensities to Action Choice Probabilities for the Roth-Erev RL Algorithm

$$p_j(t) = \frac{q_j(t)}{\sum_{i=0}^{N-1} q_i(t)}$$

$p_j(t)$ is the *choice probability* of action a_j at time t

Modified Roth-Erev RL Algorithm

$$q_j(t + 1) = [1 - r]q_j(t) + E_j^*(e, N, k, t)$$

$$E_j^*(e, N, k, t) = \begin{cases} \pi_k(t)[1 - e] & \text{if } j = k \\ q_j(t)\frac{e}{N-1} & \text{if } j \neq k \end{cases}$$

$q_j(0)$ is the initial propensity of action j at time $t = 0$ (aspiration level)

$q_j(t)$ is the propensity of action j at time t

$\pi_k(t)$ is the reward obtained for taking action k at time t

r is the recency parameter

e is the experimentation parameter

N is the number of actions

From Action Propensities to Action Choice Probabilities for the Modified Roth-Erev RL Algorithm

$$p_j(t) = \frac{q_j(t)}{\sum_{i=0}^{N-1} q_i(t)}$$

$p_j(t)$ is the *choice probability* of action a_j at time t

Variant Roth-Erev RL Algorithm

$$q_j(t + 1) = [1 - r]q_j(t) + E_j^*(e, N, k, t)$$

$$E_j^*(e, N, k, t) = \begin{cases} \pi_k(t)[1 - e] & \text{if } j = k \\ q_j(t)\frac{e}{N-1} & \text{if } j \neq k \end{cases}$$

$q_j(0)$ is the initial propensity of action j at time $t = 0$ (aspiration level)

$q_j(t)$ is the propensity of action j at time t

$\pi_k(t)$ is the reward obtained for taking action k at time t

r is the recency parameter

e is the experimentation parameter

N is the number of actions

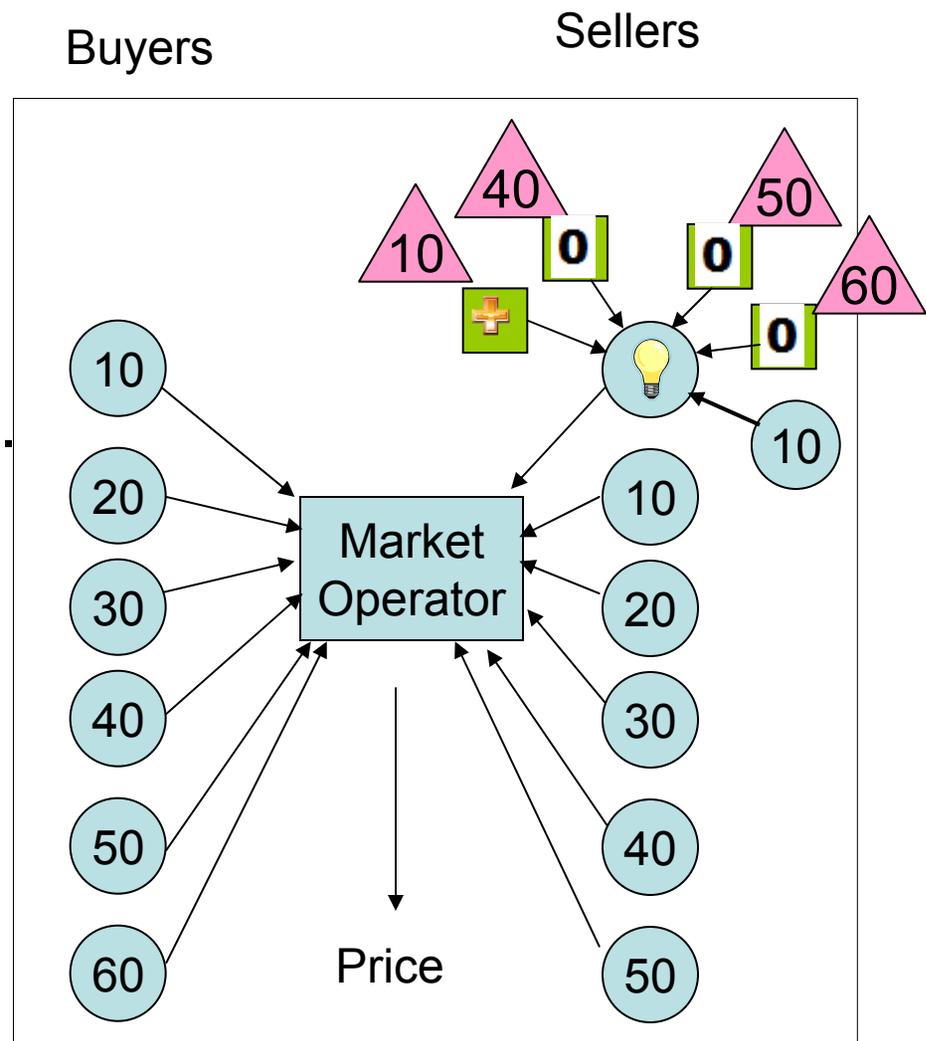
From Action Choice Propensities to Action Choice Probabilities for the Variant Roth-Erev RL Algorithm

$$p_j(t) = \frac{e^{\frac{q_j(t)}{T}}}{\sum_{i=0}^{N-1} e^{\frac{q_i(t)}{T}}}$$

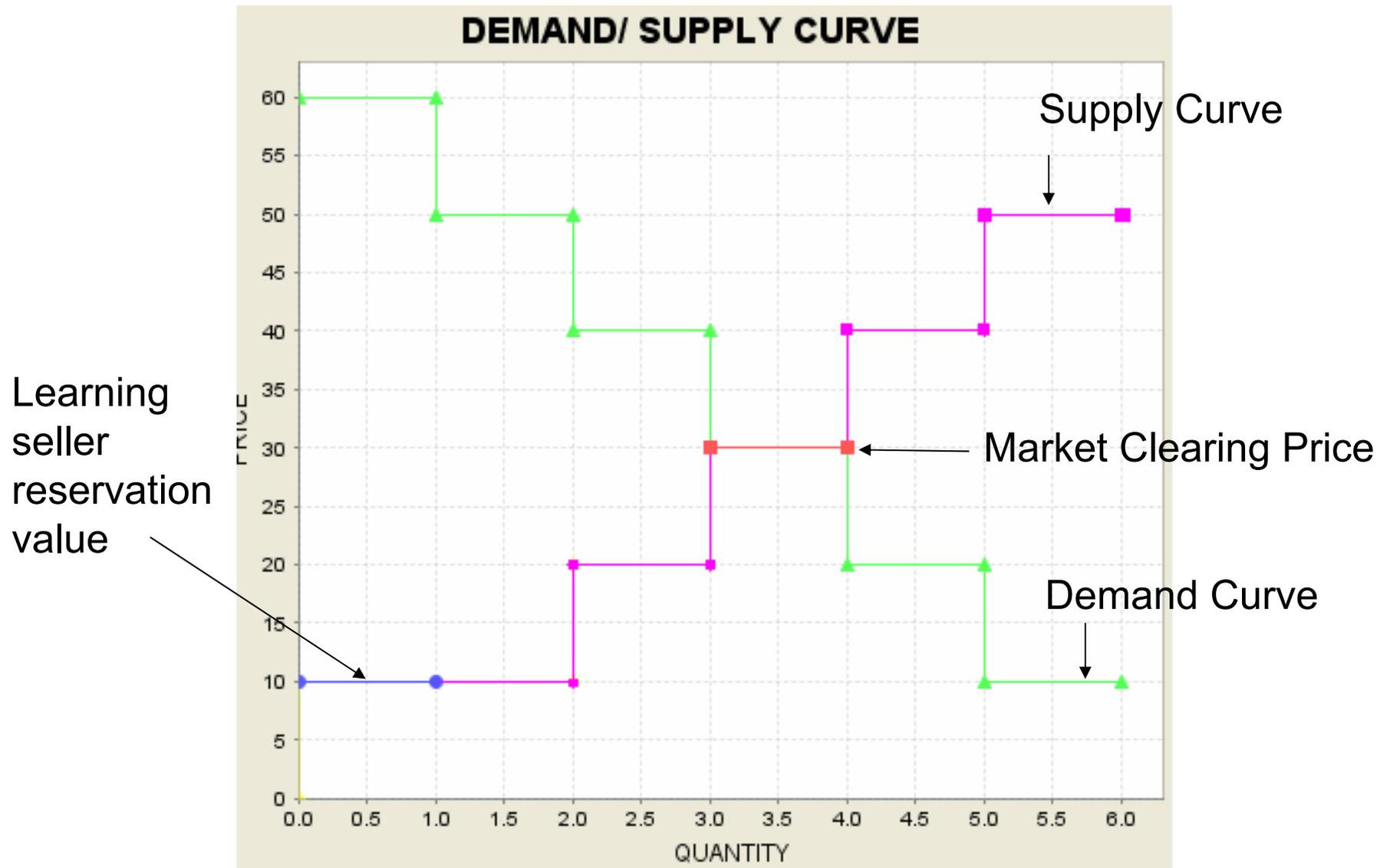
$p_j(t)$ is the *choice probability* of action a_j at time t
 T is the Gibbs-Boltzmann cooling parameter

SimpleModel-I: One Learning Seller with One Profitable Action

- Six sellers and six buyers 
- Only **one** seller  uses reinforcement learning
- The Learning Seller has four sale price choices 
- All sellers/buyers have fixed reservation values (in circles).
- Market operator constructs supply/ demand curves and calculates uniform market clearing price
- Seller Profit $\pi = [\text{Market clearing price} - \text{Reservation value}]$
- Only **one** price choice for the Learning Seller generates positive profit 



True Supply & Demand Curves for SimpleModel-I (True Reservation Values)



Experimental Design for SimpleModel-I

- Initial propensities ($q_j(0)$) are all set equal to the same level taking on one of two values:
 - (i) all with value 1000.0 (Experiment 1: High Initial Propensity)
 - (ii) all with value 1.0 (Experiment 2: Low Initial Propensity)
- Experimentation parameter (e) is varied from 0.0 to 1.0 in increments of 0.1.
- Recency parameter (r) is varied from 0.0 to 1.0 in increments of 0.1.
- 100 runs for each $\{r, e\}$ setting are conducted, with a different initial random seed for each run.
- Each run consists of 1000 market rounds, with the Learning Seller's profit (π) calculated for each round
- For each run, at the end of the 1000th round, the profits of the Learning Seller earned over the entire run are reported along with the action choice probability currently assigned to his best action (i.e., to his only profitable action).

Experimental Design (continued)

		<u>Experimentation parameter (e)</u> →				
		0.0	0.1	0.2	...	1.0
Recency parameter (r)	0.0	100 runs	100 runs	100 runs	• • •	100 runs
	0.1	100 runs				• • •
	.	• • •				• • •
	.					
	1.0	100 runs		• • •	• • •	100 runs

Initial propensity has settings of i) 1000 or ii) 1

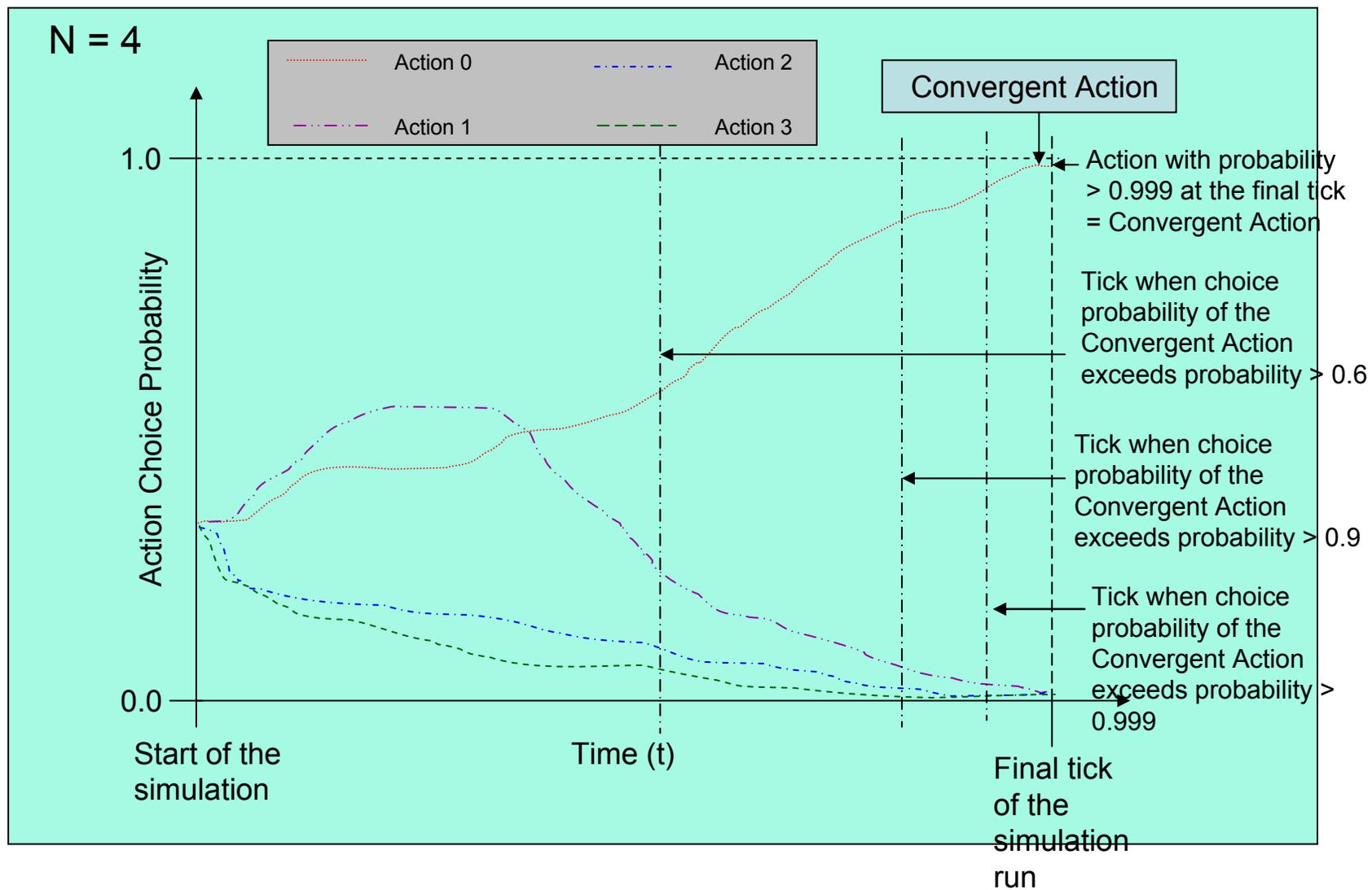
Profit of the Learning Seller for a run = Sum of profits obtained by the Learning Seller across 1000 rounds.

Total Profit for the Learning Seller per $\{r, e\}$ setting = Sum of profits of the Learning Seller for all runs with a given $\{r, e\}$ setting

Average Total Profit for the Learning Seller = Total profit for the Learning Seller per $\{r, e\}$ setting / Number of runs with the $\{r, e\}$ setting

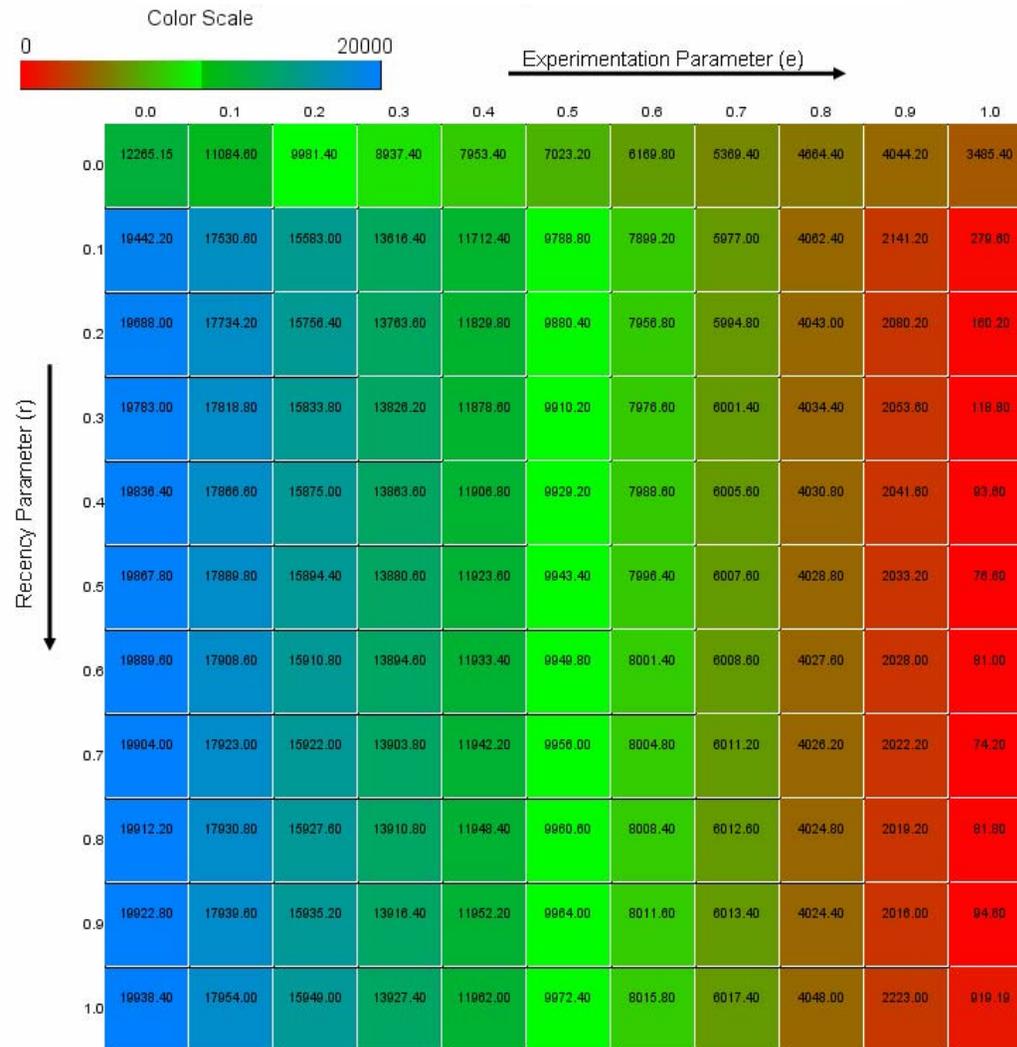
Definition of “Convergent Action”

Sketch of a simulation run showing the action *choice probabilities* over time.



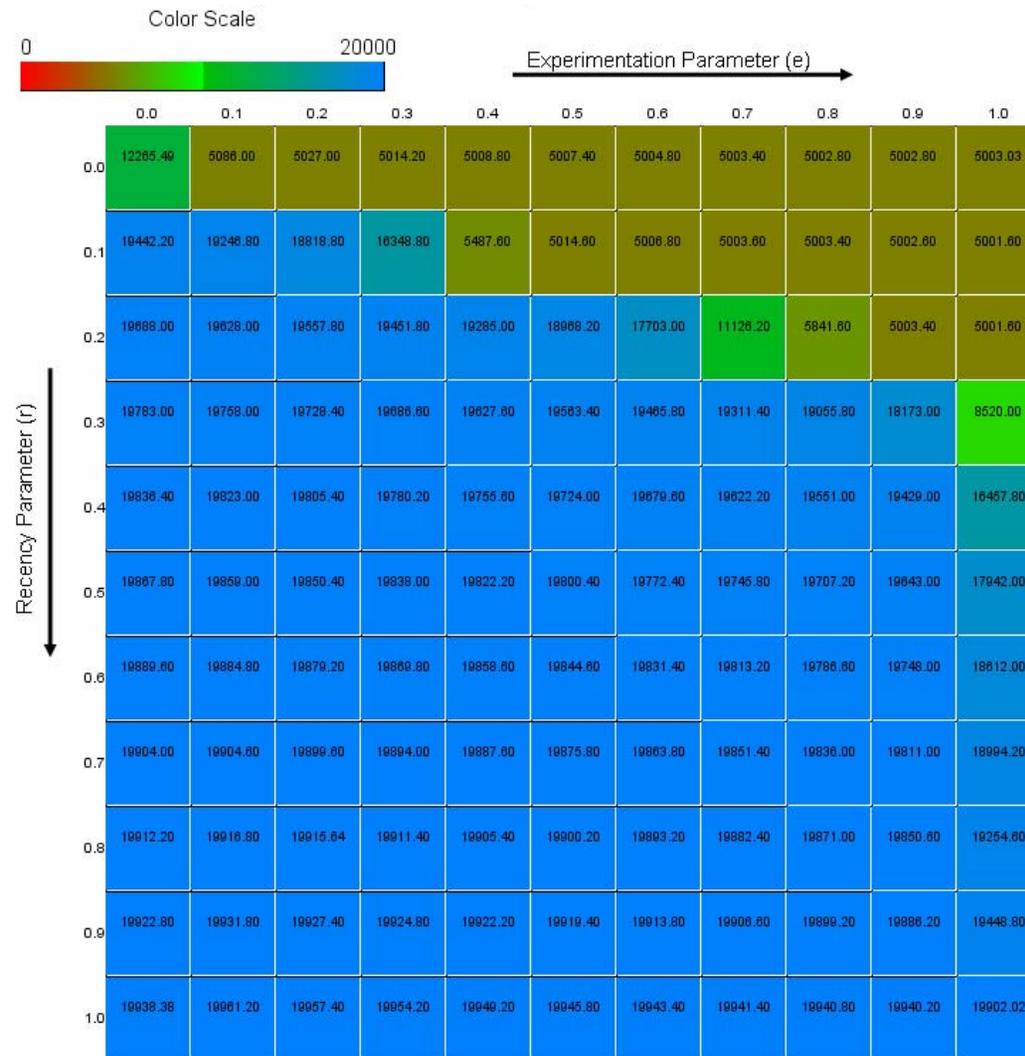
SimpleModel-I: Experiment 1 (High Initial Propensity)

Average Total Profits for the Learning Seller (Roth-Erev RL Algorithm)



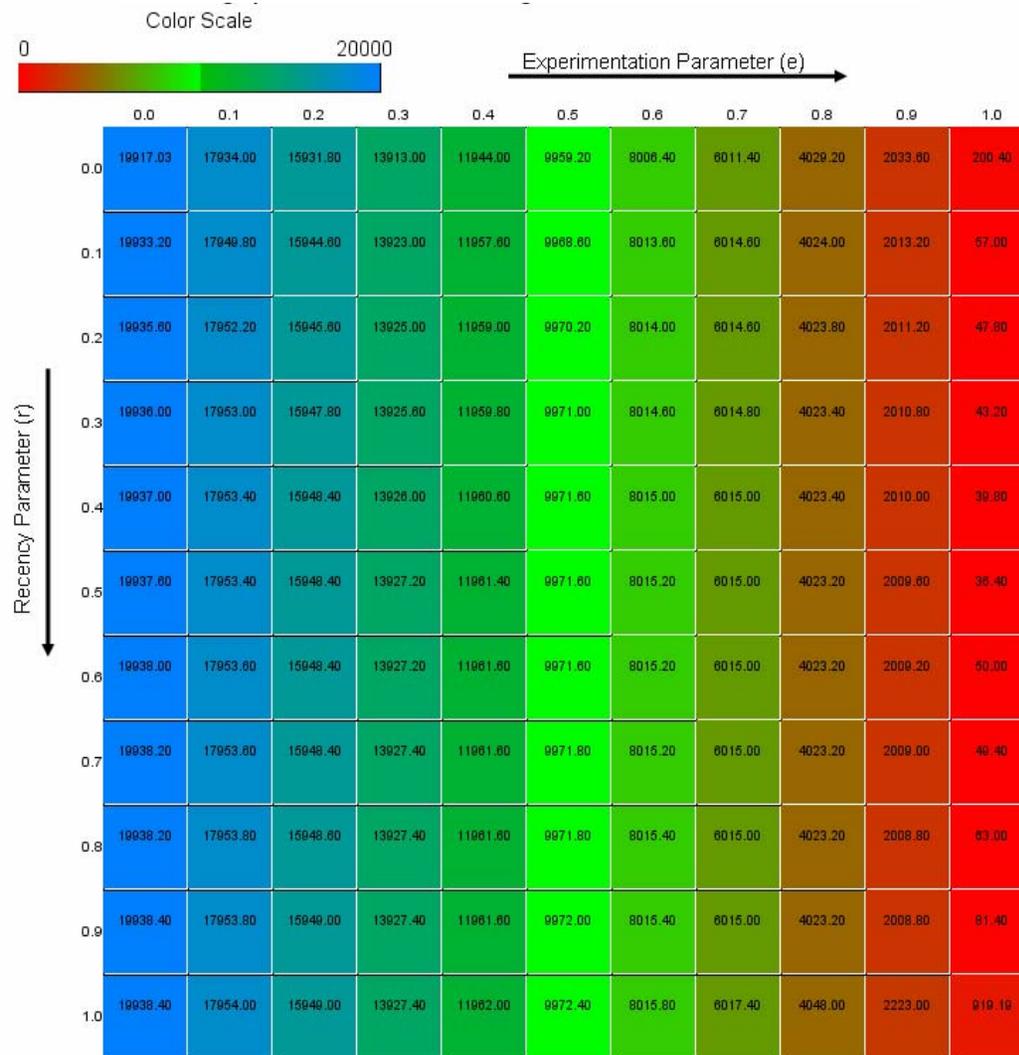
SimpleModel-I: Experiment 1 (High Initial Propensity)

Average Total Profits for the Learning Seller (Modified Roth-Erev RL Algorithm)



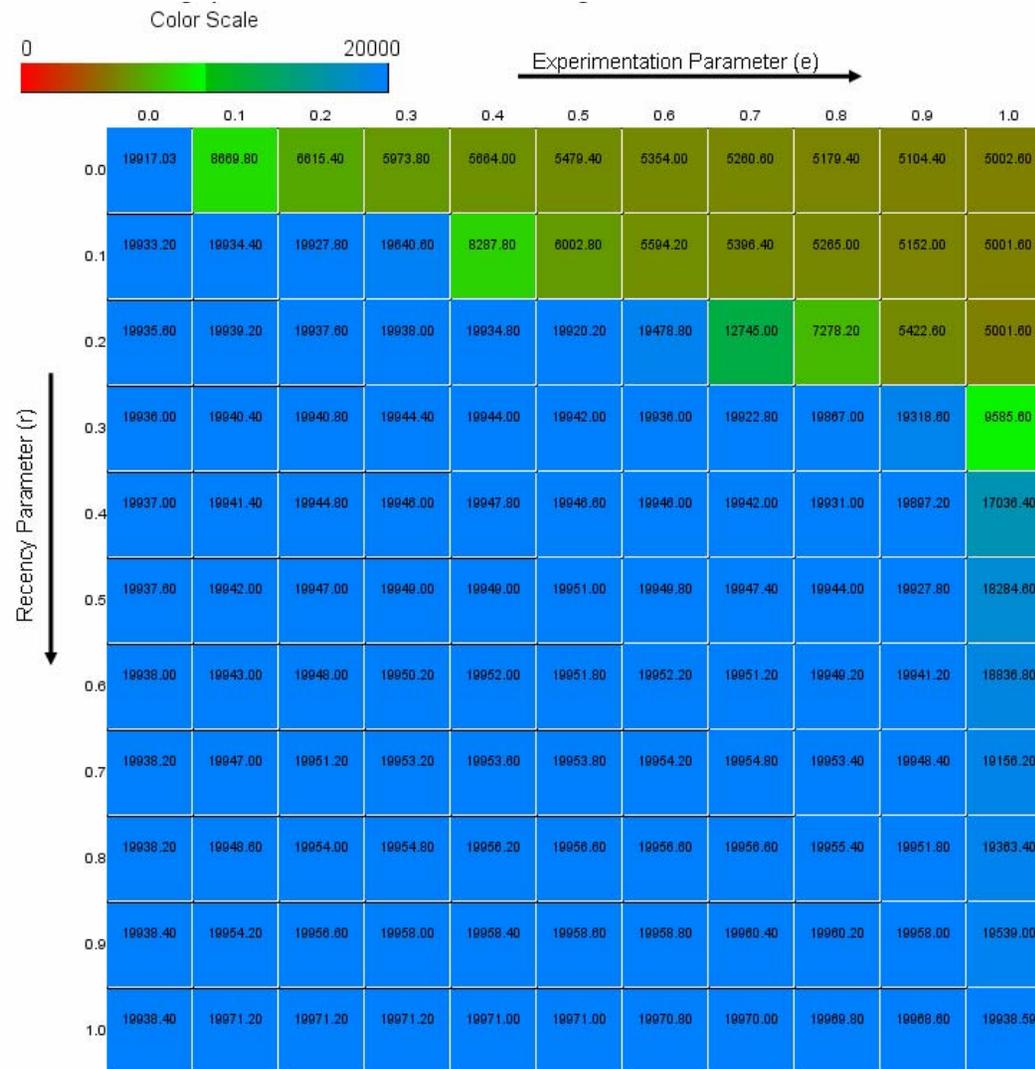
SimpleModel-I: Experiment 2 (Low Initial Propensity)

Average Total Profits for the Learning Seller (Roth-Erev RL Algorithm)



SimpleModel-I: Experiment 2 (Low Initial Propensity)

Average Total Profits for the Learning Seller (Modified Roth-Erev RL Algorithm)



SimpleModel-I with Roth-Erev RL algorithm

Computational Results Suggest a Theorem

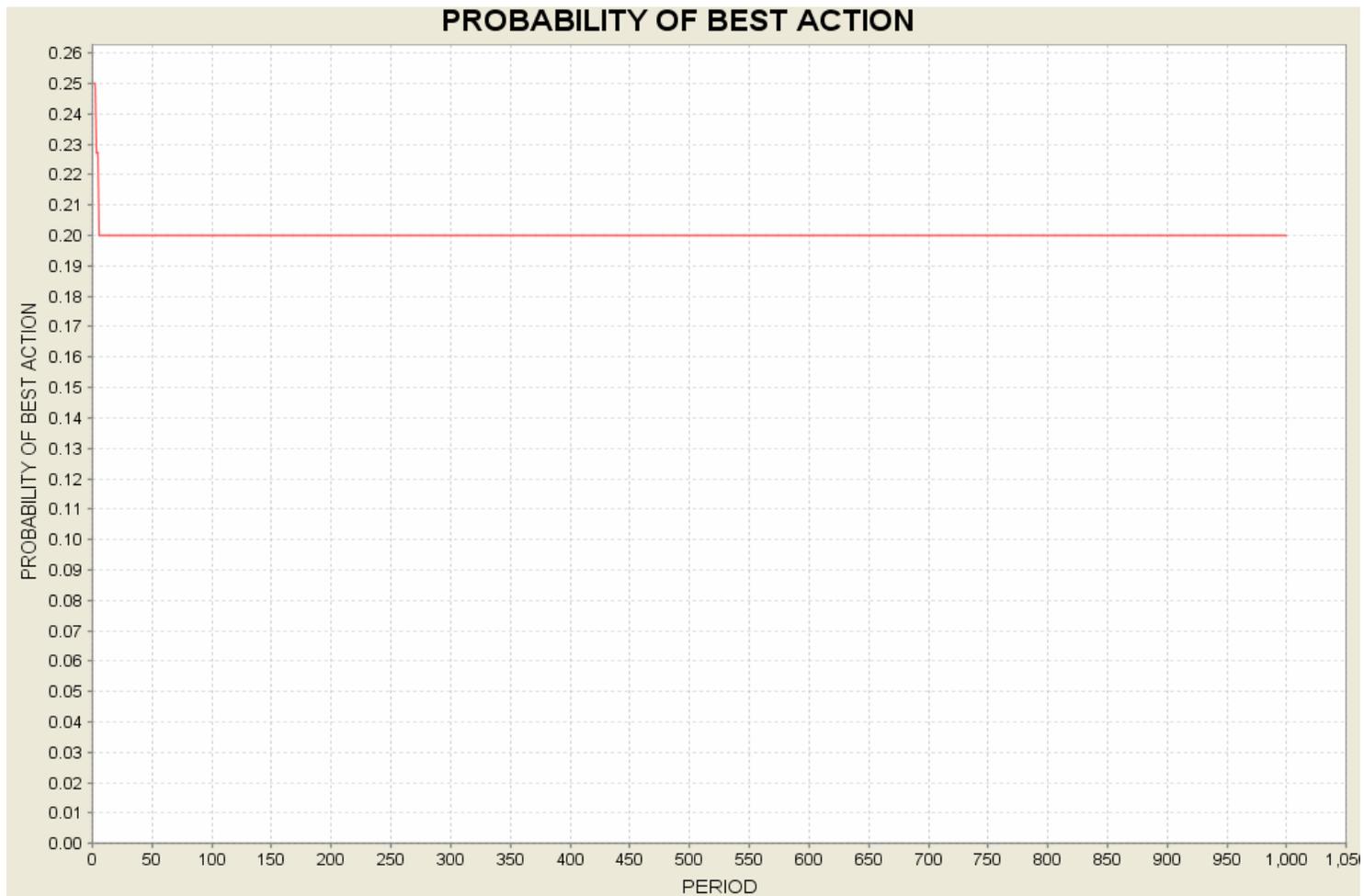
- In most simulation runs the *choice probability* of the best action appears to converge to some probability value, that appears to depend on the recency and experimentation parameters, but appears to be independent of the initial propensity.
- An extract from the XML output of the simulation run is shown for parameter settings:
 - Experimentation $e = 0.8$
 - Recency $r = 0.7$
 - Number of actions $N = 4$
 - Initial propensity for all actions $q_j(0) = 1000$

```
<tick>
  <count>693</count>
  <agent id = "0">
    <propensities>
      <propensity id = "0">0.4680810490953136</propensity>
      <propensity id = "1">0.6241080654604177</propensity>
      <propensity id = "2">0.6241080654604177</propensity>
      <propensity id = "3">0.6241080654604177</propensity>
    </propensities>
    <probabilities>
      <probability id = "0">0.20000000000000012</probability>
      <probability id = "1">0.2666666666666666</probability>
      <probability id = "2">0.2666666666666666</probability>
      <probability id = "3">0.2666666666666666</probability>
    </probabilities>
    <action>2</action>
    <reward>0.0</reward>
  </agent>
</tick>

<tick>
  <count>922</count>
  <agent id = "0">
    <propensities>
      <propensity id = "0">0.002916000000002771</propensity>
      <propensity id = "1">0.0038880000000036928</propensity>
      <propensity id = "2">0.0038880000000036928</propensity>
      <propensity id = "3">0.0038880000000036928</propensity>
    </propensities>
    <probabilities>
      <probability id = "0">0.20000000000000007</probability>
      <probability id = "1">0.2666666666666666</probability>
      <probability id = "2">0.2666666666666666</probability>
      <probability id = "3">0.2666666666666666</probability>
    </probabilities>
    <action>0</action>
    <reward>20.0</reward>
  </agent>
</tick>
```

Graphed Results for a Simulation Run

Choice Probability of best action in a sample simulation run with experimentation = 0.8, recency = 0.7, $N = 4$ and $q_j(0) = 1000$



Theorem Statement

- Suppose the the action domain for a learning agent using Roth-Erev RL includes $N \geq 2$ actions. Suppose all of the initial action choice propensities $q_j(0)$, $j=0, \dots, N-1$, are positively valued and the recency parameter r and experimentation parameter e have values lying strictly between 0 and 1. Finally, suppose the profit for choosing the “best action” 0 is positive and constant and the profit for choosing each other action j is 0.
- If the best action 0 is chosen infinitely many times, then the choice probability of the best action 0 converges to $(1 - e)$ and the choice probability of each of the other $N-1$ actions converges to $e/[N-1]$.

Proof of the theorem

Consider the case in which, for all sufficiently large t , the learning agent always chooses the best action 0. Without loss of generality, it can be assumed that this persistent choice of action 0 starts in time period 1. Following the choice of action 0 at time 1, the initial choice propensity $q_0(0)$ for action 0 is updated as follows:

$$q_0(1) = (1 - r)q_0(0) + \pi_0(1 - e)$$

At time $t = 2$, this becomes

$$q_0(2) = (1 - r)\left((1 - r)q_0(0) + \pi_0(1 - e)\right) + \pi_0(1 - e)$$

At a general time t the formula is

$$q_0(t) = (1 - r)^t q_0(0) + \left((1 - r)^{t-1} + (1 - r)^{t-2} + \dots + (1 - r)^0\right) \pi_0(1 - e)$$

As time $t \rightarrow \infty$, this becomes

$$q_0(\infty) = \frac{1}{1 - (1 - r)} \pi_0(1 - e)$$

or

$$q_0(\infty) = \frac{\pi_0(1 - e)}{r}$$

Proof of the theorem (continued)

Similarly

As $t \rightarrow \infty$,

$$q_j(\infty) = \pi_0 \times \frac{e}{N-1} \times \frac{1}{r} \quad 1 \leq j < N$$

$$\begin{aligned} p_0(\infty) &= \frac{q_0(\infty)}{N-1} \\ &\quad \sum_{i=0} q_i \\ &= \frac{\frac{\pi_0(1-e)}{r}}{\sum_{i=1}^{N-1} \frac{\pi_0(e)}{(N-1)(r)} + \frac{\pi_0(1-e)}{r}} \\ &= (1 - e) \end{aligned} \quad p_j(\infty) = \frac{e}{N-1} \quad 1 \leq j < N$$

Proof of the theorem (continued)

Now suppose that some action j *other* than the best action 0 is also chosen infinitely often. Each time that this action j is chosen, the only effect is to uniformly shrink *all* propensity values by a factor of $[1 - r]$, so that *all* choice probabilities are unaffected by this choice. It follows that probability equations above are the limits for the choice probabilities as long as the best action 0 is chosen infinitely often, whether or not any other action j is chosen infinitely often as well.

Necessary & Sufficient Condition for Strict Increase in the Choice Propensity of a Non-Chosen Action: MRE RL Algorithm

Suppose that the choice propensities for a non-best action j satisfy

$$q_j(1) > q_j(0)$$

\therefore

$$(1-r)q_j(0) + q_j(0) \times \frac{\epsilon}{N-1} > q_j(0)$$

\therefore

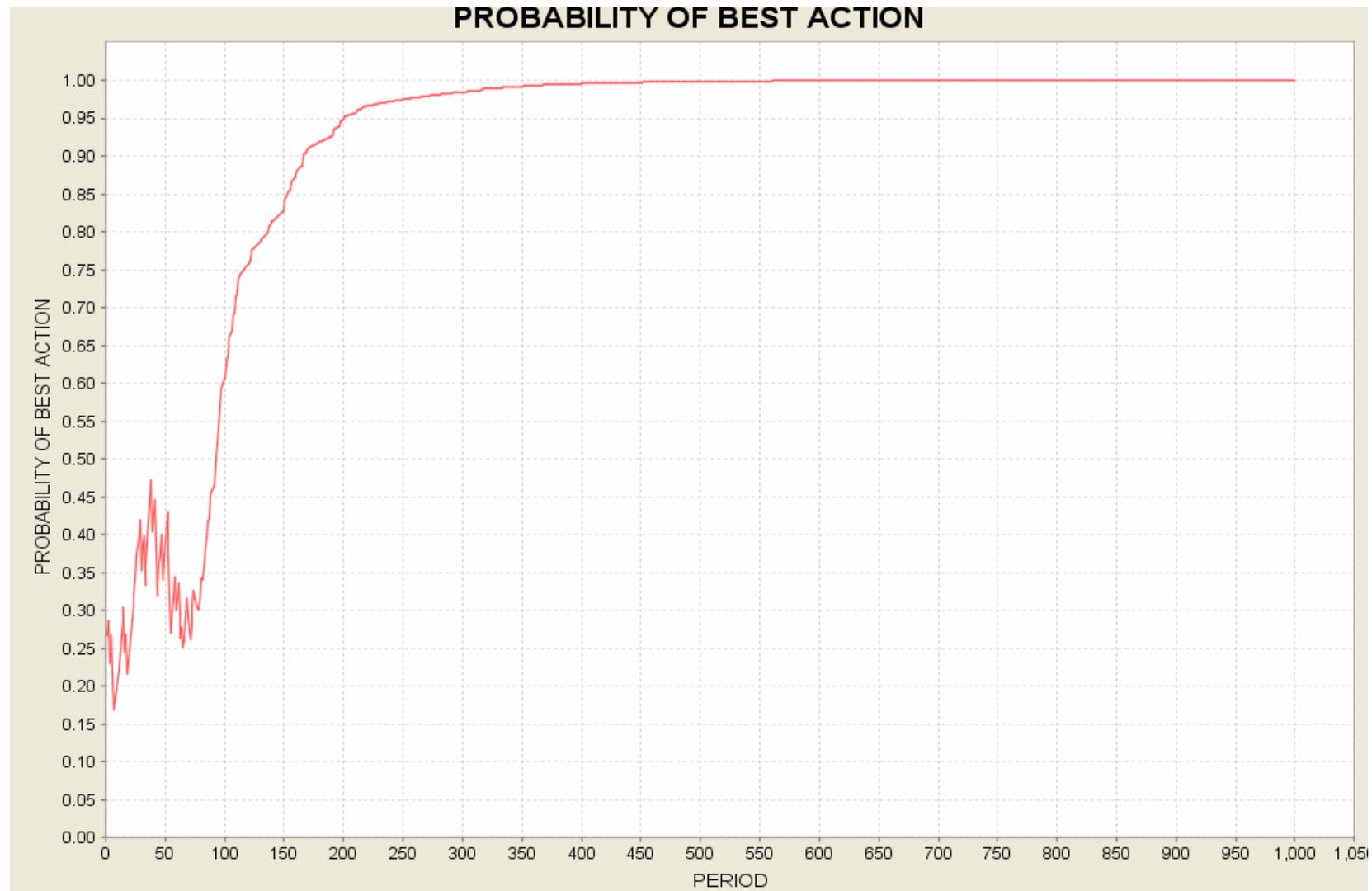
$$(1-r) + \frac{\epsilon}{N-1} > 1$$

\therefore

$$\frac{\epsilon}{N-1} > r$$

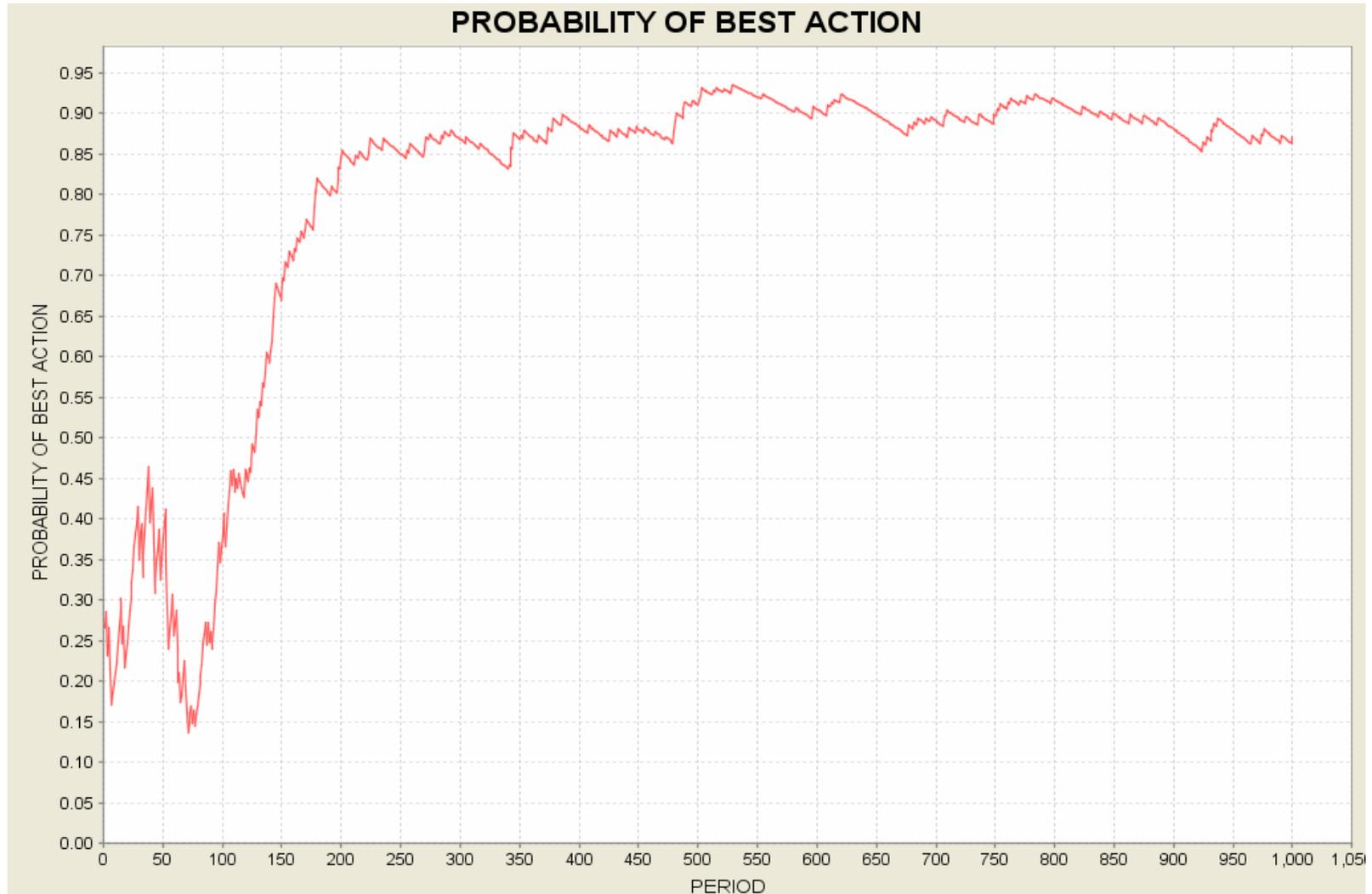
Best Action Choice Probability over Time

$e/[N-1] \leq r$ (MRE RL Algorithm)



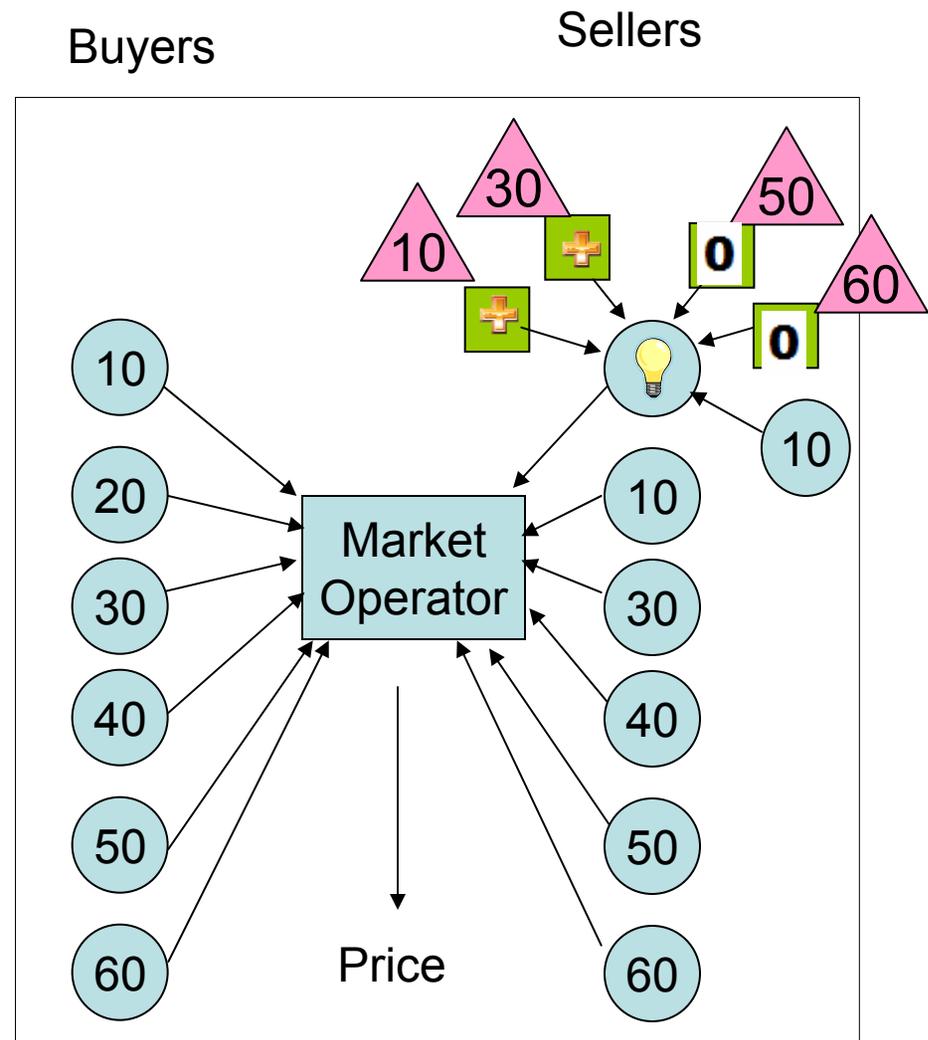
Best Action Choice Probability over Time

when $r < e/[N-1]$ (MRE RL Algorithm)



SimpleModel-II: One Learning Seller with Two Profitable Actions

- Six sellers and six buyers 
- Only **one** seller  uses reinforcement learning
- The Learning Seller has four sale price choices 
- All sellers/ buyers have fixed reservation values (in circles).
- Market operator constructs supply/ demand curves and calculates uniform market clearing price
- Seller Profit $\pi = \text{Market clearing price} - \text{Reservation value}$
- **Two** price choices of the Learning Seller generate positive profits 
- Action  **30** always has the largest profit for the learning agent.



Experimental Design for SimpleModel-II

- Initial propensities ($q_j(0)$) are all set equal to the same level taking on one of two values:
 - (i) all with value 1000.0 (Experiment 1: High Initial Propensity)
 - (ii) all with value 1.0 (Experiment 2: Low Initial Propensity)
- Experimentation parameter (e) is varied from 0.0 to 1.0 in increments of 0.1.
- Recency parameter (r) is varied from 0.0 to 1.0 in increments of 0.1.
- 100 runs for each $\{r, e\}$ setting are conducted, with a different initial random seed for each run.
- Each run consists of 1000 market rounds, with the Learning Seller's profit (π) calculated for each round
- For each run, at the end of the 1000th round, the profits of the Learning Seller earned over the entire run are reported along with the action choice probability currently assigned to his best action (i.e., to his only profitable action).

Experimental Design (continued)

Experimentation parameter (e) →

		0.0	0.1	0.2	...	1.0
Recency parameter (r)	0.0	100 runs	100 runs	100 runs	• • •	100 runs
	0.1	100 runs				• • •
	.	• • •				• • •
	.					
	1.0	100 runs		• • •	• • •	100 runs

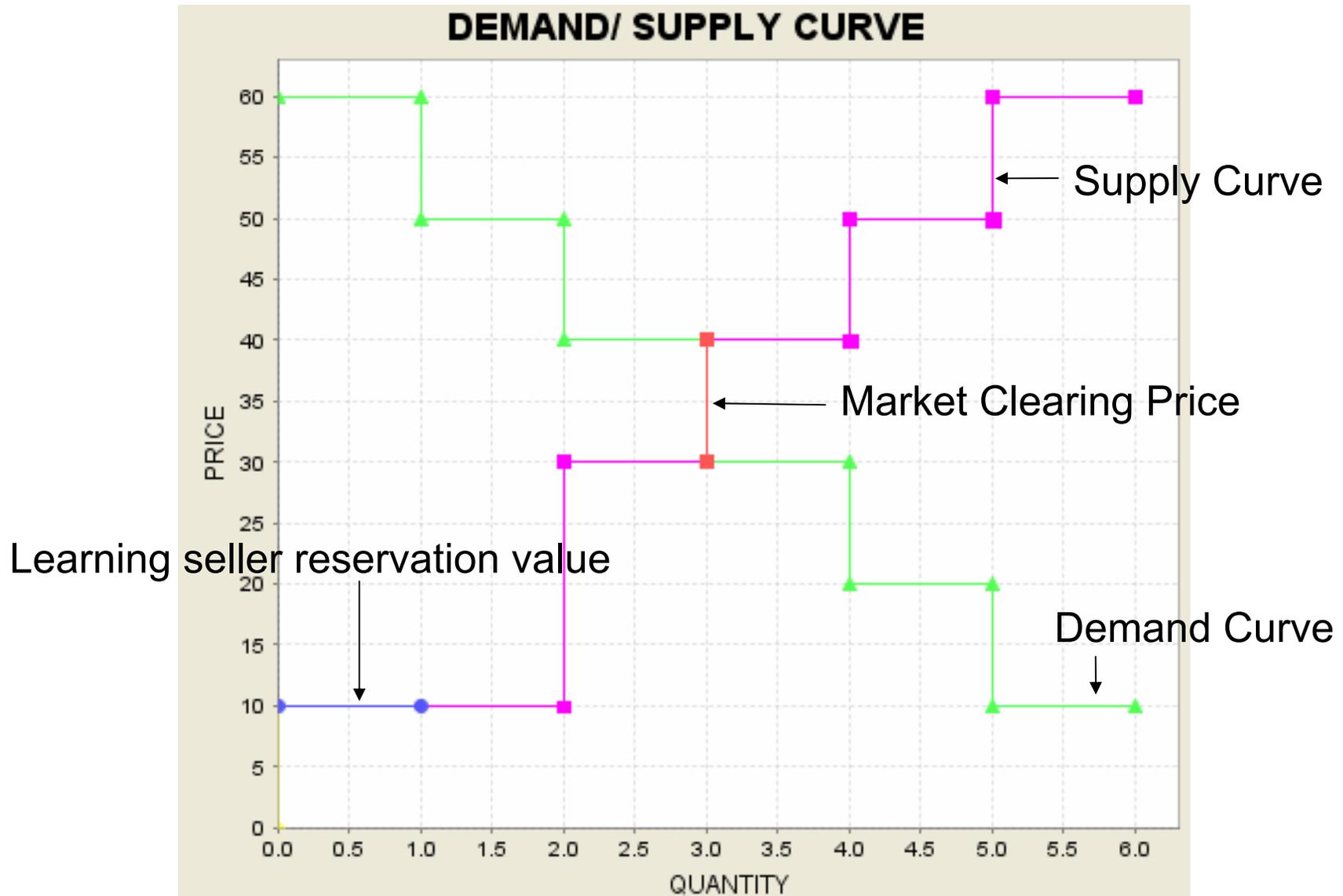
Initial propensity has settings of i) 1000 or ii) 1

Profit of the Learning Seller for a run = Sum of profits obtained by the Learning Seller across 1000 rounds.

Total Profit for the Learning Seller per $\{r, e\}$ setting = Sum of profits of the Learning Seller for all runs with a given $\{r, e\}$ setting

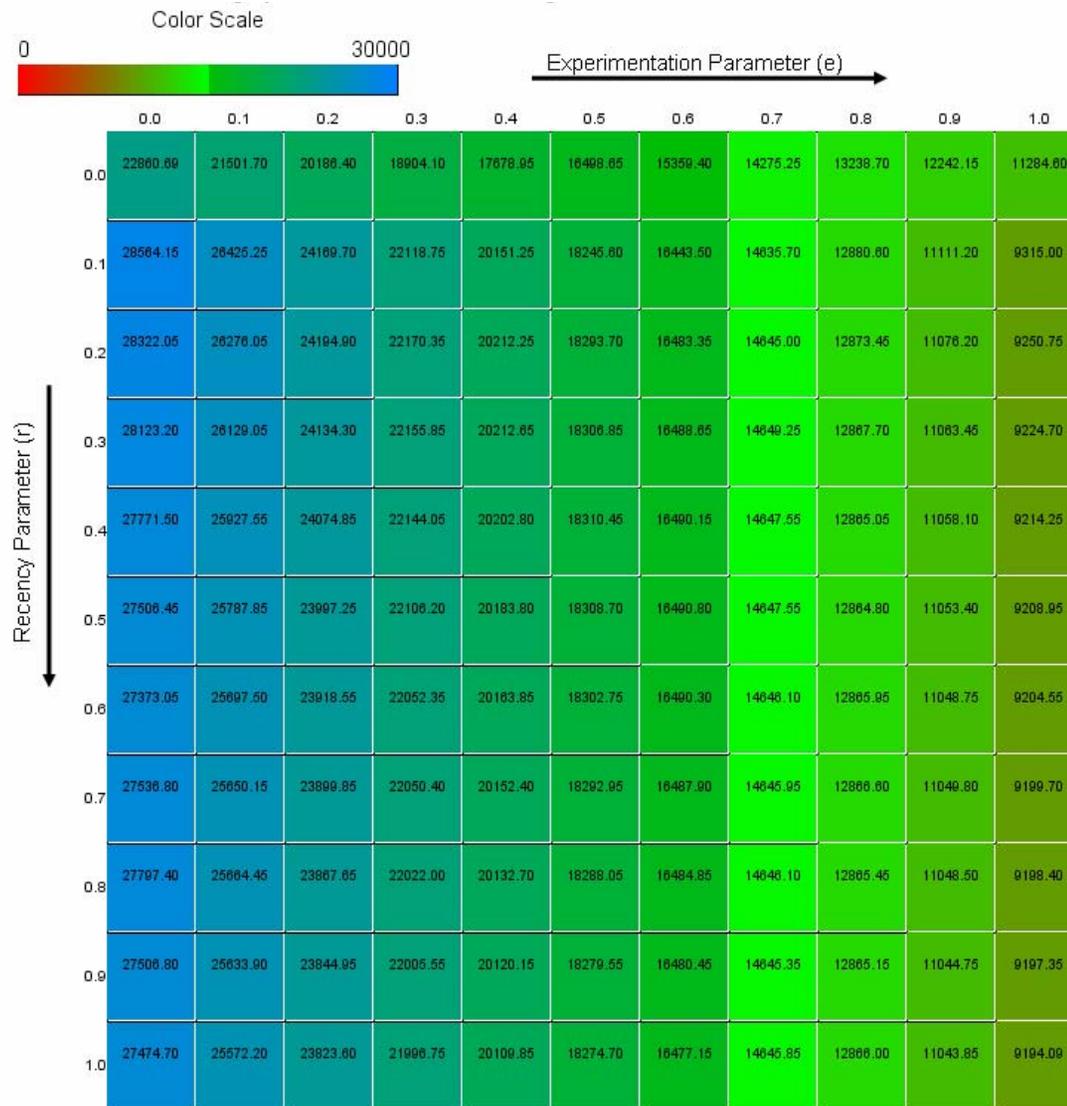
Average Total Profit for the Learning Seller = Total profit for the Learning Seller per $\{r, e\}$ setting / Number of runs with the $\{r, e\}$ setting

True Supply & Demand Curves for SimpleModel-I (True Reservation Values)



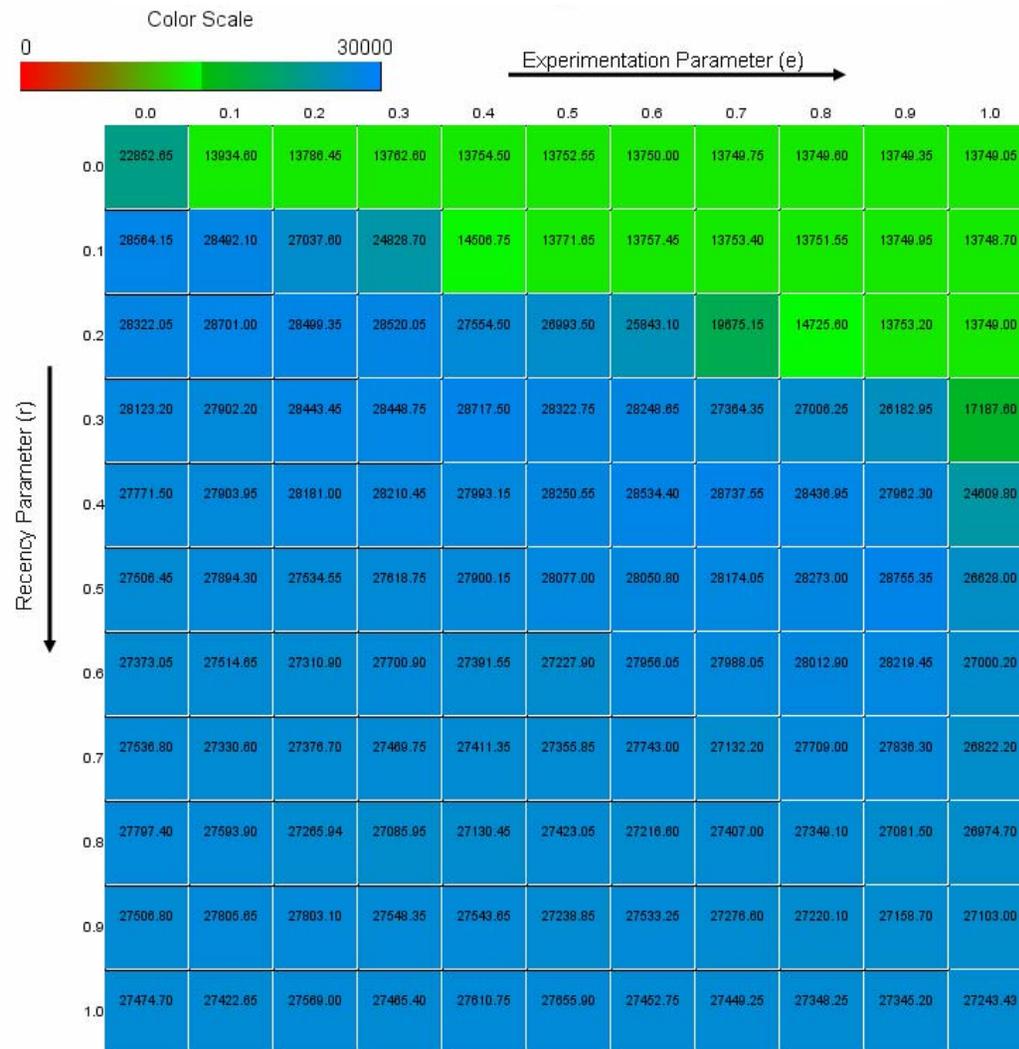
SimpleModel-II: Experiment 1 (High Initial Propensity)

Average Total Profits for the Learning Seller (Roth-Erev RL Algorithm)



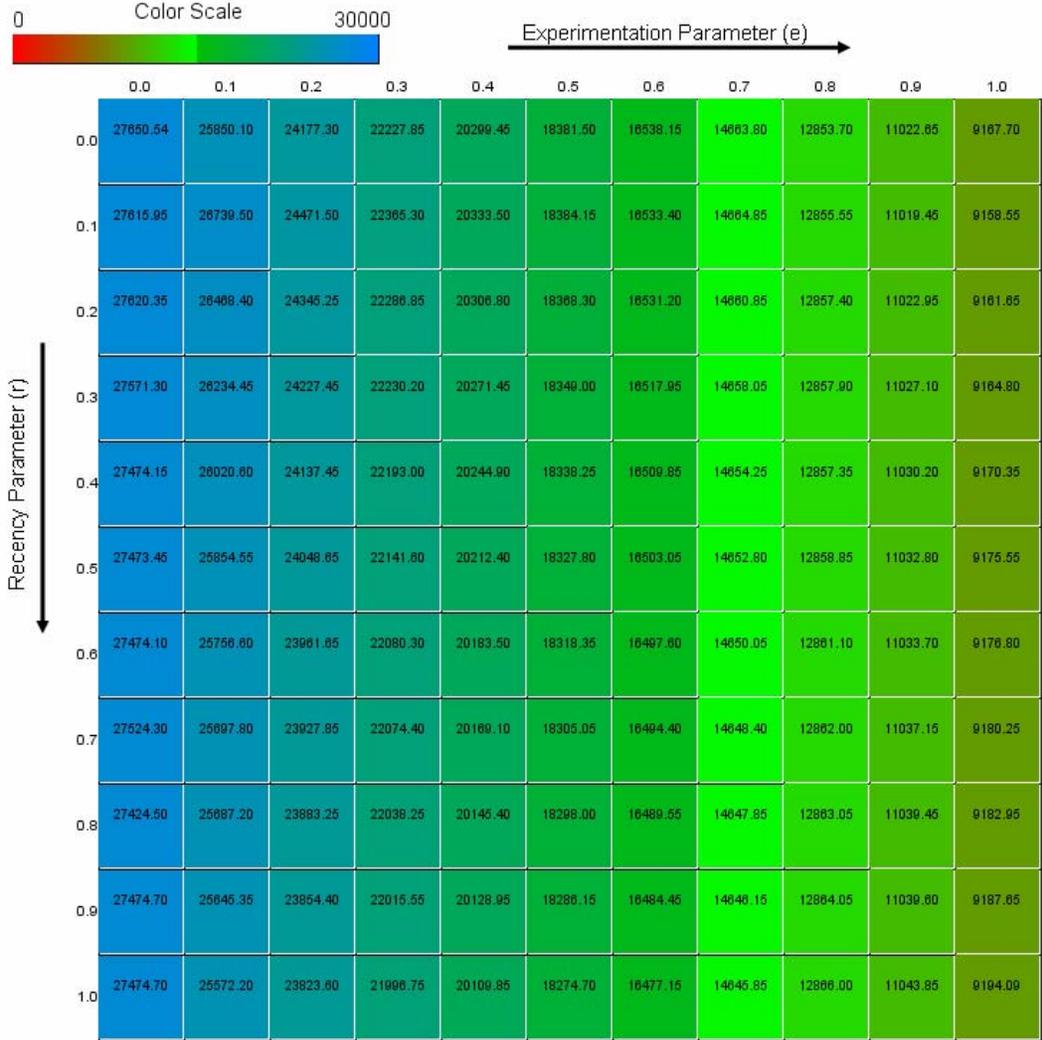
SimpleModel-II: Experiment 1 (High Initial Propensity)

Average Total Profits for the Learning Seller (Modified Roth-Erev RL Algorithm)



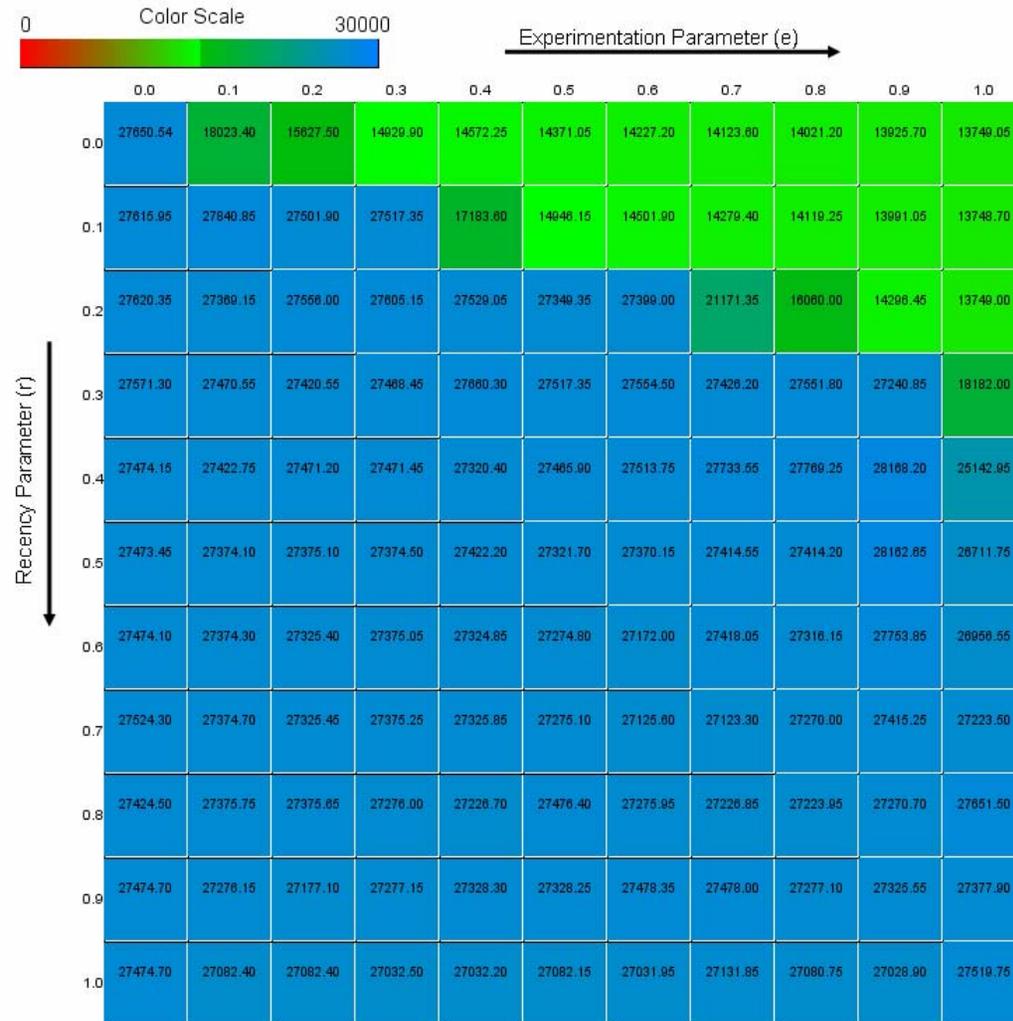
SimpleModel-II: Experiment 2 (Low Initial Propensity)

Average Total Profits for the Learning Seller (Roth-Erev RL Algorithm)



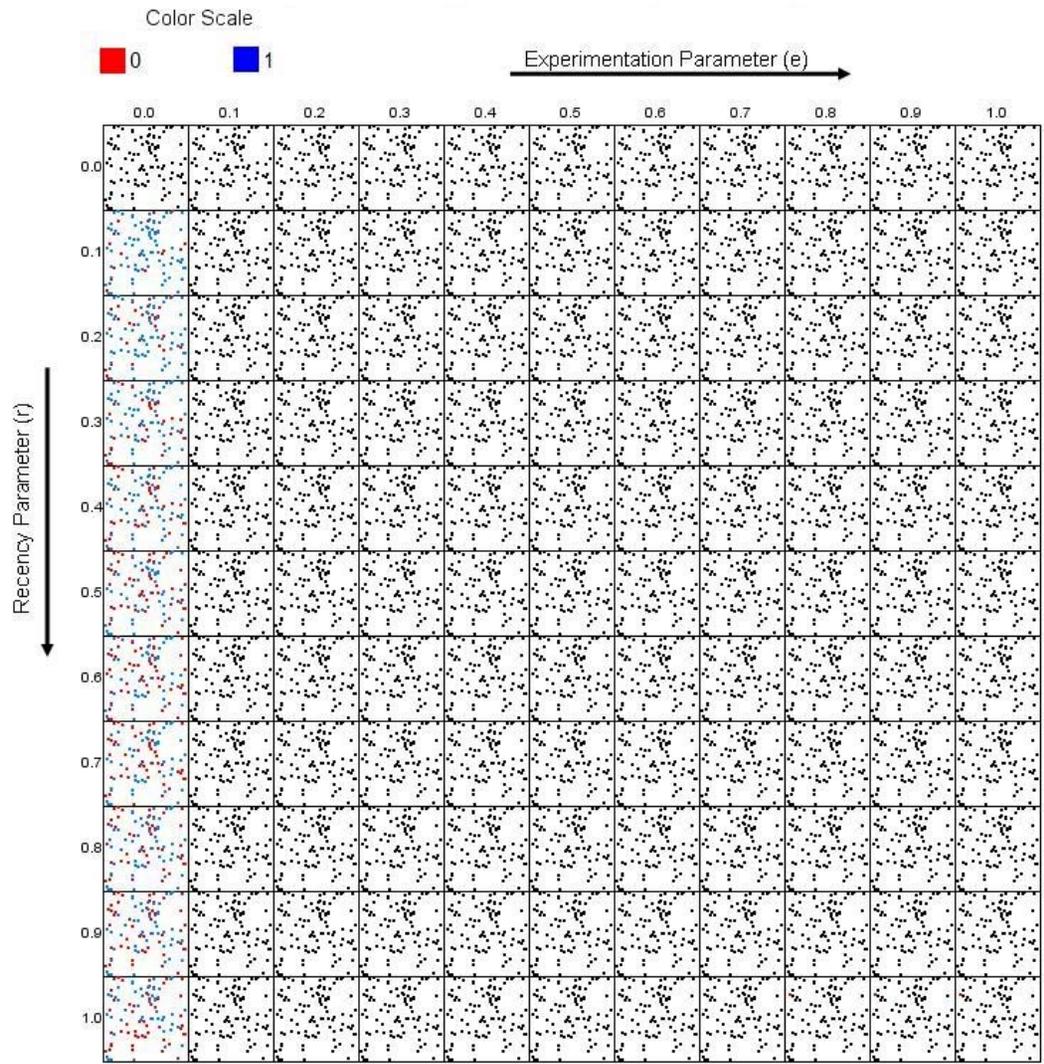
SimpleModel-II: Experiment 2 (Low Initial Propensity)

Average Total Profits for the Learning Seller (Modified Roth-Erev RL Algorithm)



SimpleModel-II: Experiment 1 (High Initial Propensity)

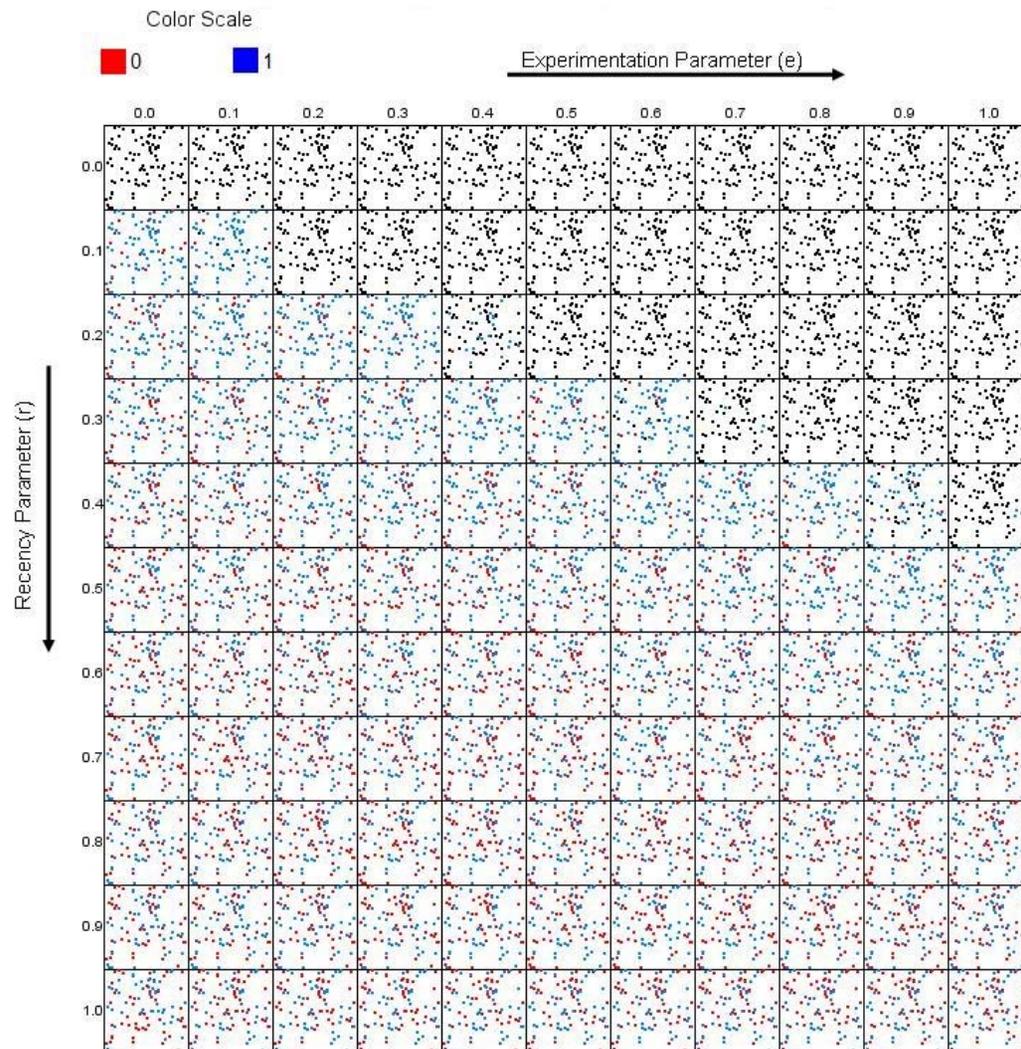
Convergent Action for the Learning Seller (Roth-Erev RL Algorithm)



- A red dot indicates that the simulation converged to action 0.
- A blue dot indicates that the simulation converged to action 1.
- A black dot indicates that the simulation did not to convergence to any action.

SimpleModel-II: Experiment 1 (High Initial Propensity)

Convergent Action for the Learning Seller (Modified Roth-Erev RL Algorithm)

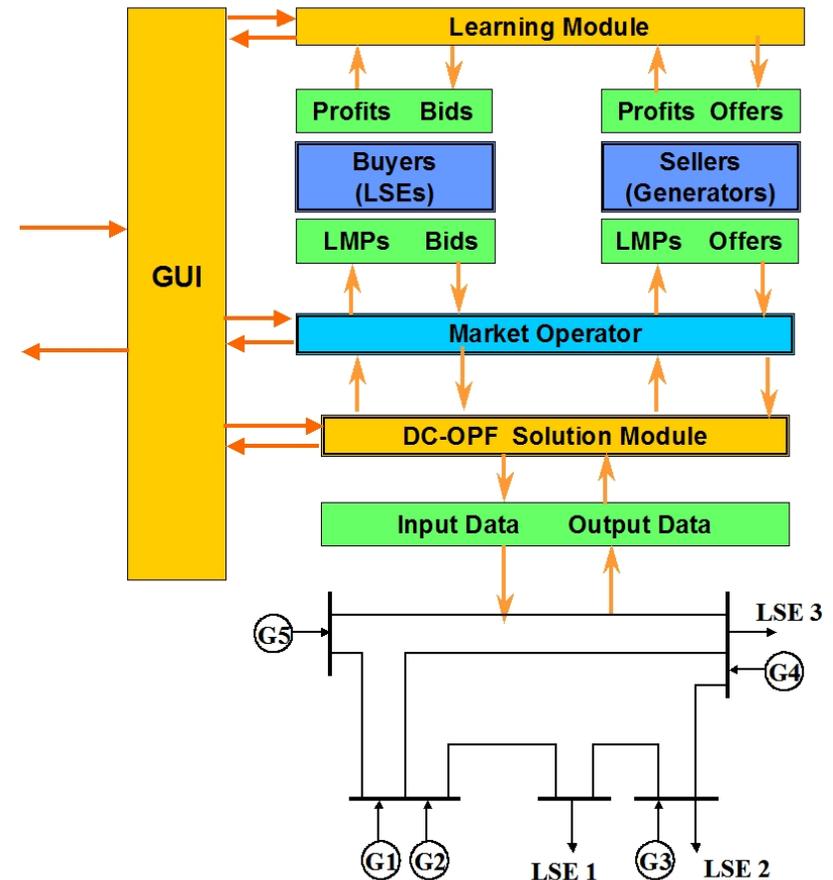


- A red dot indicates that the simulation converged to action 0.
- A blue dot indicates that the simulation converged to action 1.
- A black dot indicates that the simulation did not to convergence to any action.

AMESModel-I: One Learning Generator G5

www.econ.iastate.edu/tesfatsi/AMESMarketHome.htm

- AMESModel-I makes use of a 5-node test case conducted with the AMES Market Package (Li, Sun, Tesfatsion, 2008).
- Generator G5 is a learning seller with 40 action choices (i.e., 40 possible supply offers).
- Each of the four remaining generators (sellers) has only 1 action choice.
- Hence, there is **one learning generator**, and **four non-learning generators**.
- LSEs (buyers) report fixed demand curves to the market operator each day.
- Each generator reports a supply curve to the market operator each day.
- The Market Operator uses daily reported demand/supply curves to solve for daily prices/quantities
- Each gen/LSE uses posted solution to compute its profits for each day.



AMES Market Package

Experimental Design for AMESModel-I

- Two experiments are carried out
 - Experiment 1 (High Initial Propensity for G5): Initial propensity values ($q_j(0)$) = 140,000.0 and a cooling parameter value $T = 35,000$
 - Experiment 2 (Low Initial Propensity for G5): Initial propensity values = 6,000.0 and $T = 1,000.0$
- Experimentation parameter (e) varied from 0.0 to 1.0 in increments of 0.1.
- Recency parameter (r) varied from 0.0 to 1.0 in increments of 0.1.
- 100 runs for each $\{r, e\}$ setting with a different initial random seed for each run.
- Each run consists of 100 market rounds, with the G5's profit (π) calculated for each round
- For each run, at the end of the 100th round the Total Profits obtained by G5 over the run are calculated, and a “Convergent Action” (if any) for G5 is recorded.

Experimental Design (continued)

Experimentation parameter (e) →

		0.0	0.1	0.2	...	1.0
Recency parameter (r)	0.0	100 runs	100 runs	100 runs	• • •	100 runs
	0.1	100 runs				• • •
	.	• • •				• • •
	.					
	1.0	100 runs		• • •	• • •	100 runs

Initial propensity has settings of i) 140,000 with $T = 35,000$ or ii) 6000 with $T = 1000$

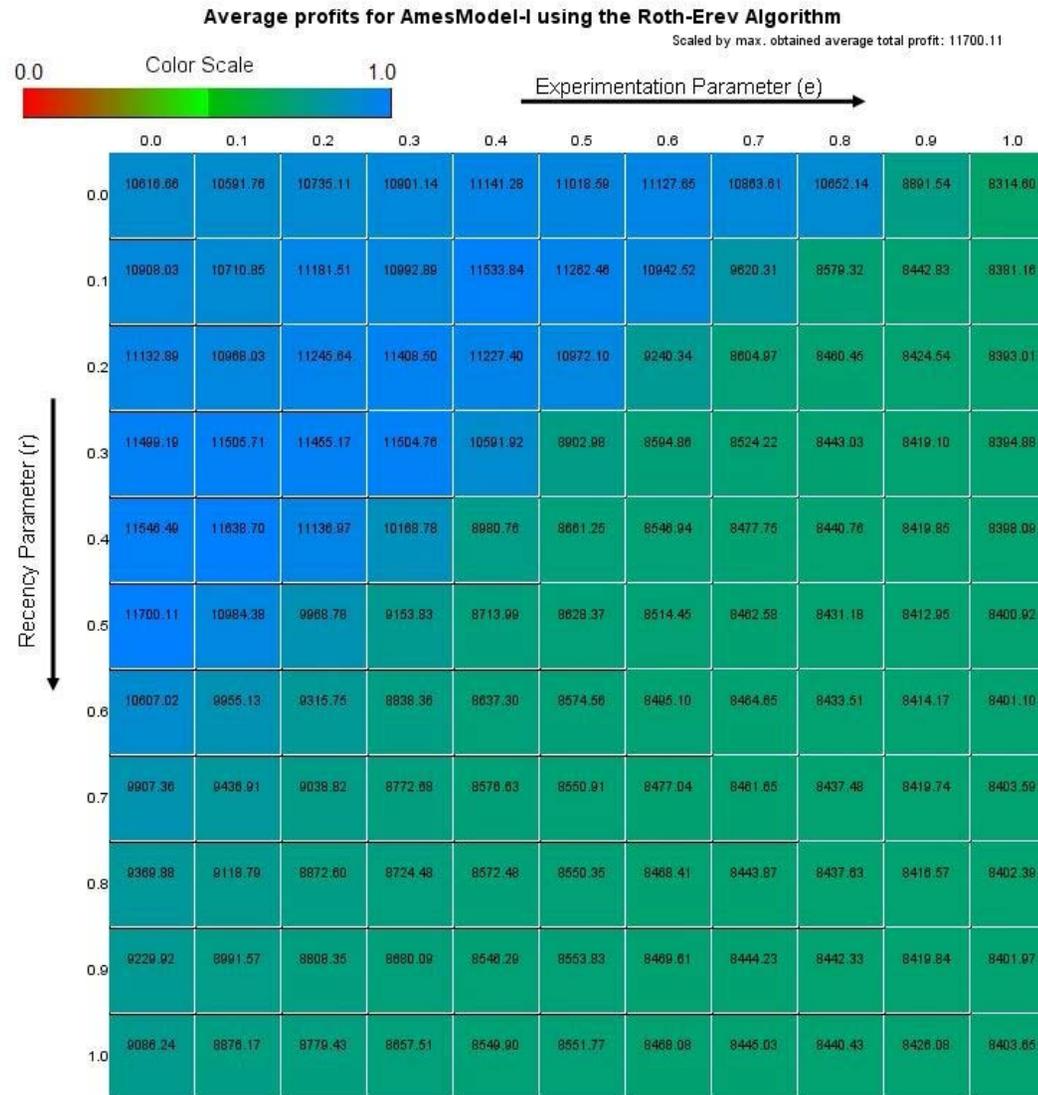
Profit of G5 for a run = Sum of profits for G5 obtained across all 100 rounds.

Total Profits of G5, given a $\{r, e\}$ setting = Sum of all profits for G5 in all runs with this $\{r, e\}$ setting

Average Total Profits = Total Profits divided by number of runs (for G5 for given $\{r, e\}$ setting)

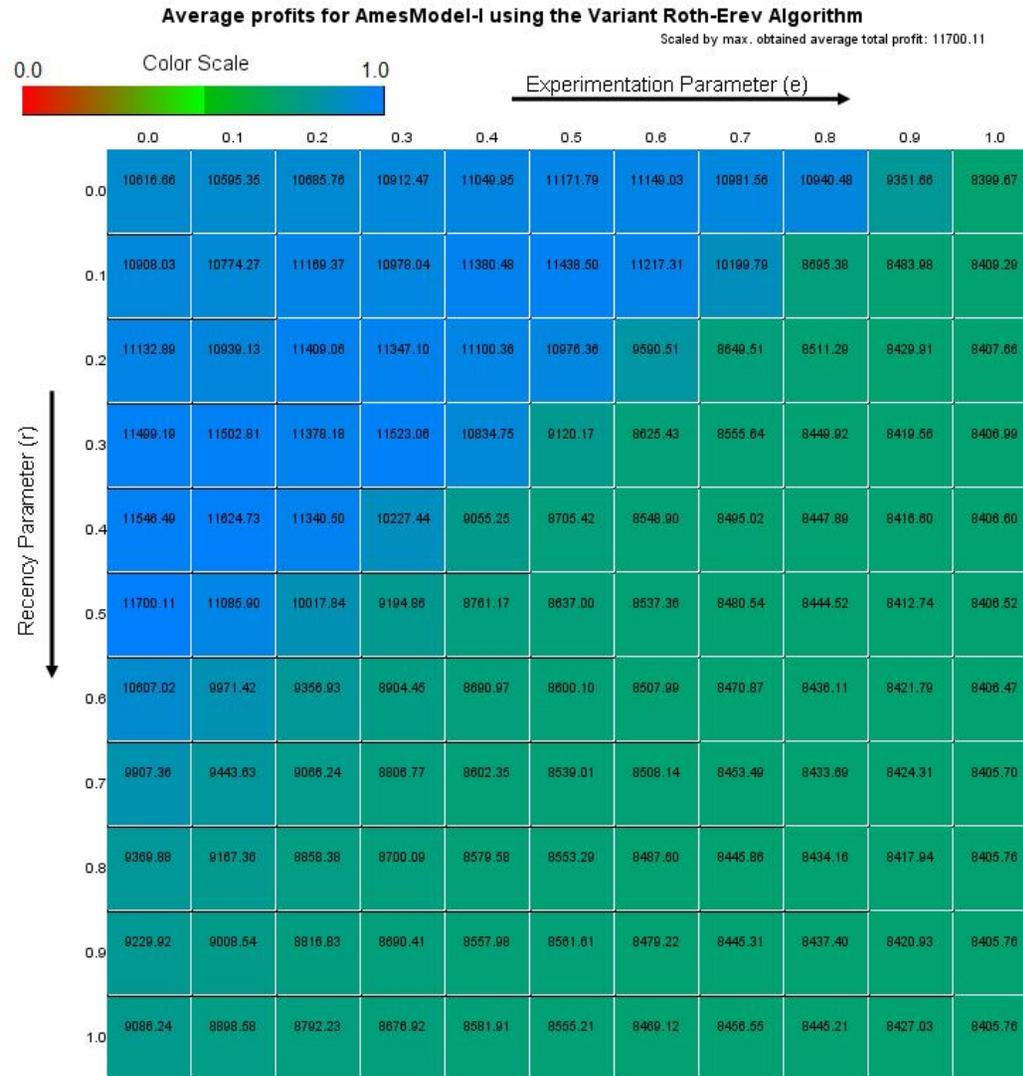
AMESModel-I: Experiment 1 (High Initial Propensity)

Average Total Profits for G5 (Roth-Erev RL Algorithm)



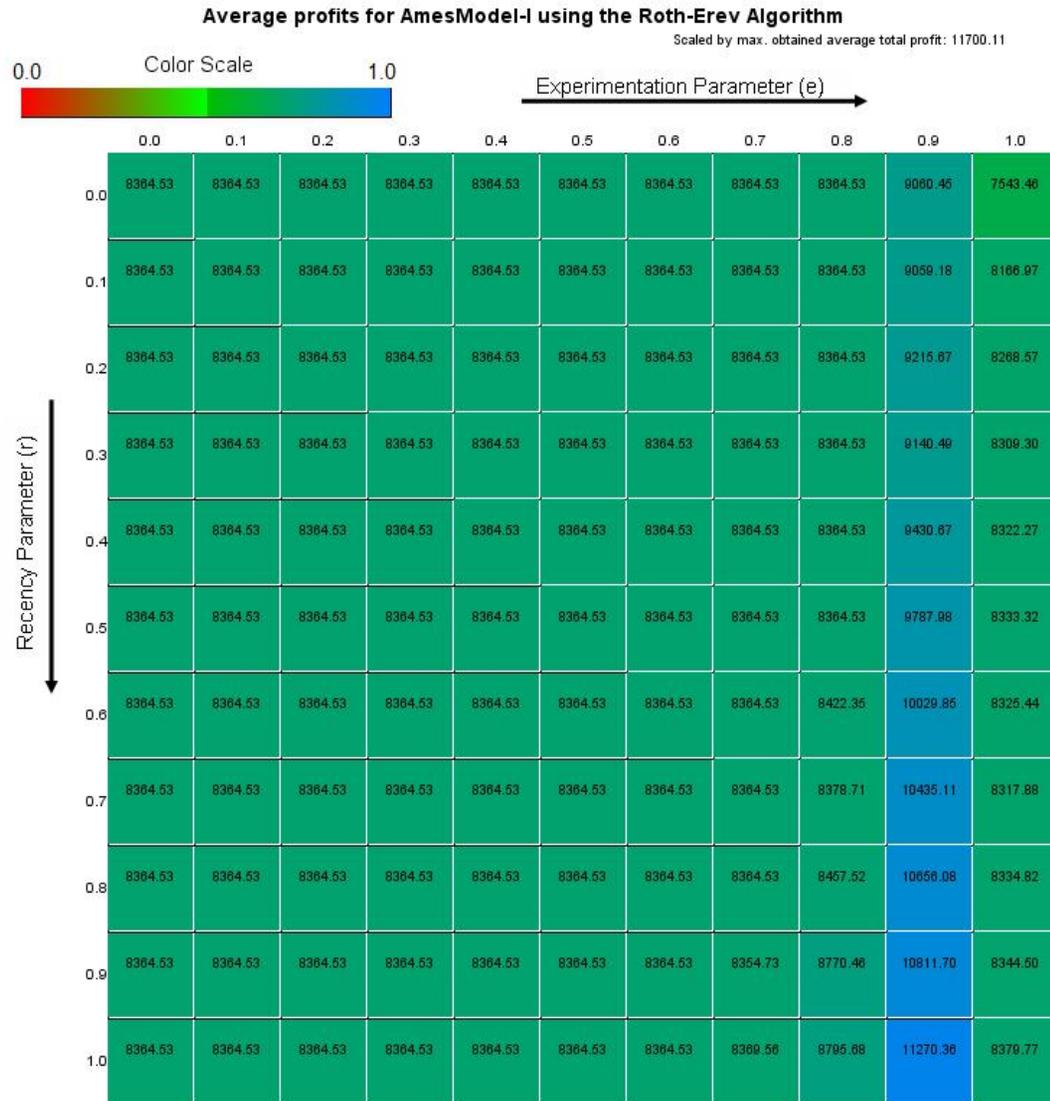
AMESModel-I: Experiment 1 (High Initial Propensity)

Average Total Profits for G5 (Variant Roth-Erev RL Algorithm)



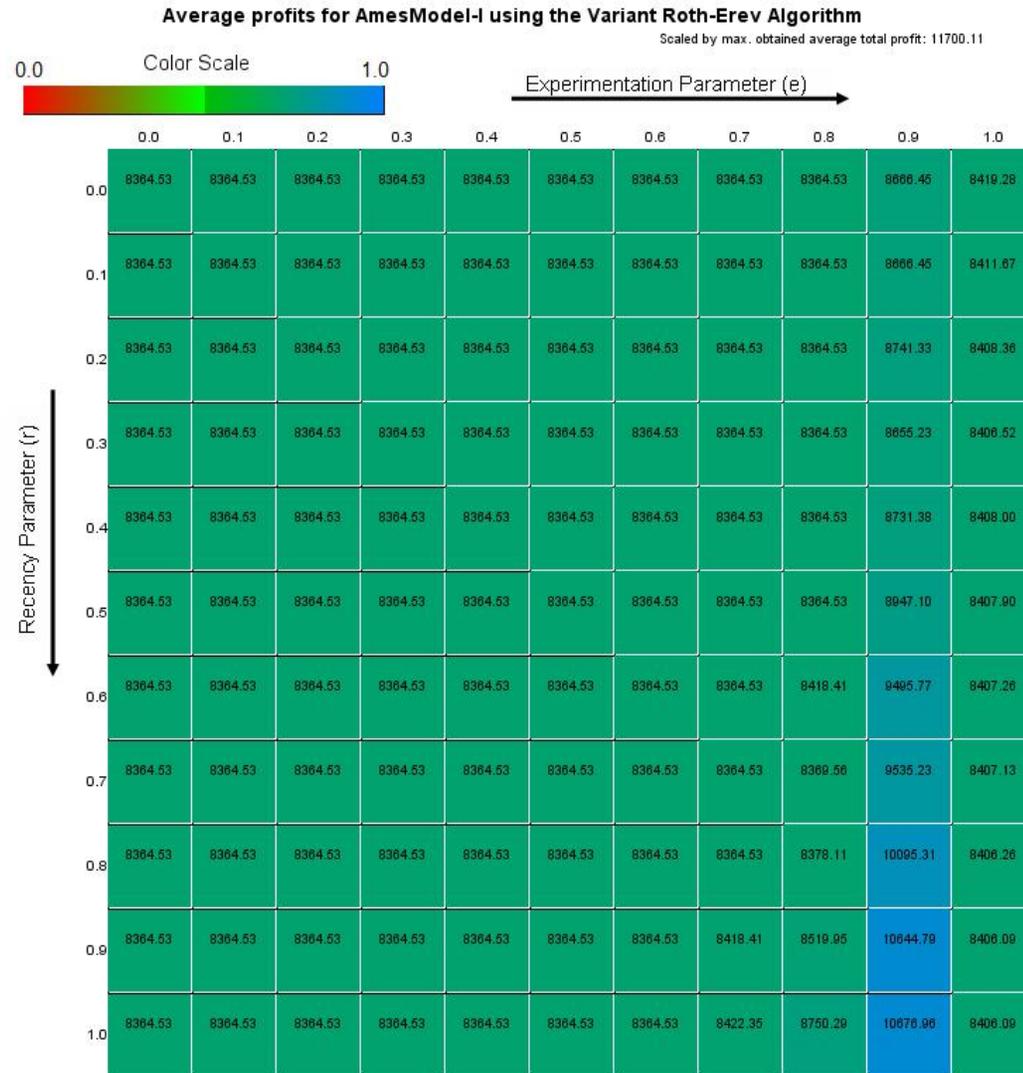
AMESModel-I: Experiment 2 (Low Initial Propensity)

Average Total Profits for G5 (Roth-Erev RL Algorithm)



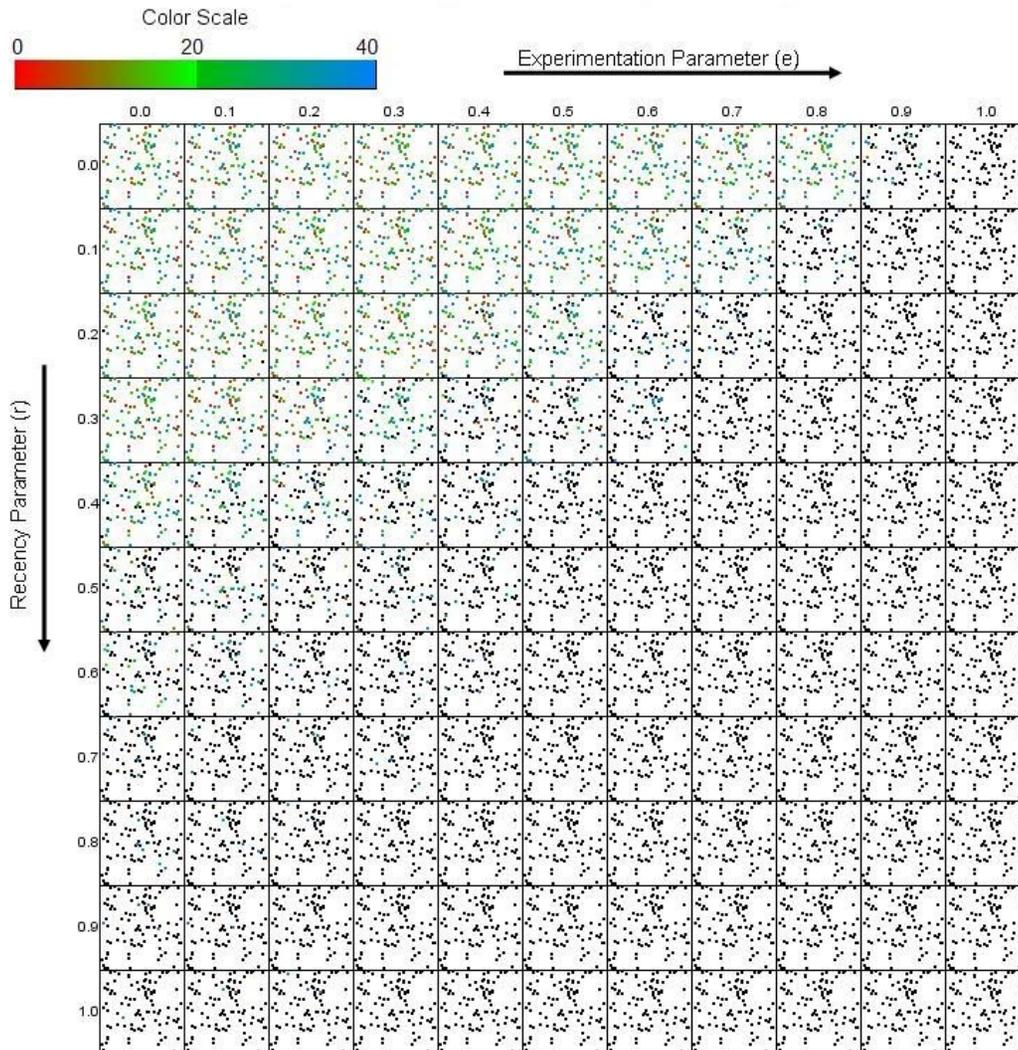
AMESModel-I: Experiment 2 (Low Initial Propensity)

Average Total Profits for G5 (Variant Roth-Erev RL Algorithm)



AMESModel-I: Experiment 1 (High Initial Propensity)

G5's Convergent Actions (Roth – Erev RL Algorithm)

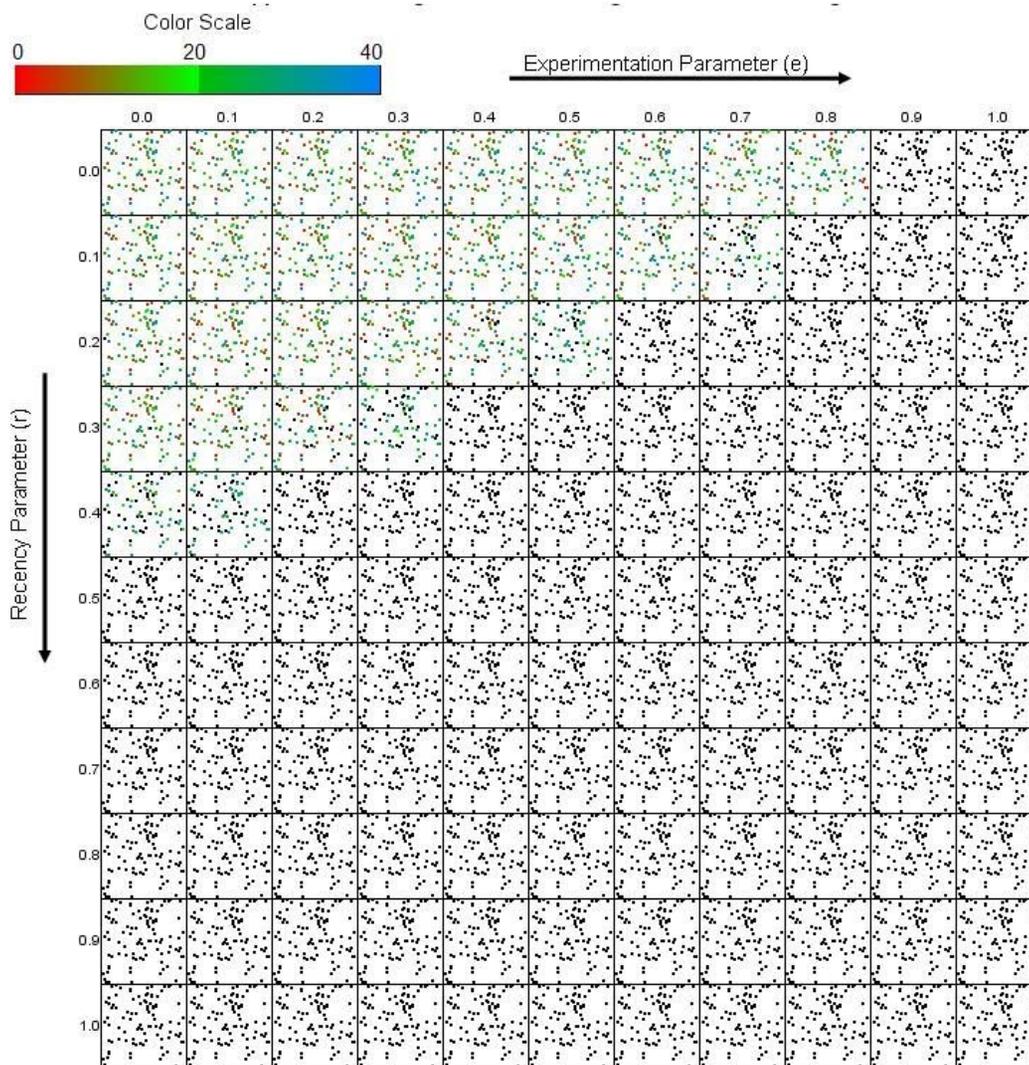


- A run for which none of G5's action choice probabilities exceed .999 at 100th tick is marked with black.
- In all other runs, the color indicates which of G5's 40 actions has a choice probability $> .999$ at the 100th tick.

AMESModel-I: Experiment 1 (High Initial Propensity)

G5's Convergent Actions

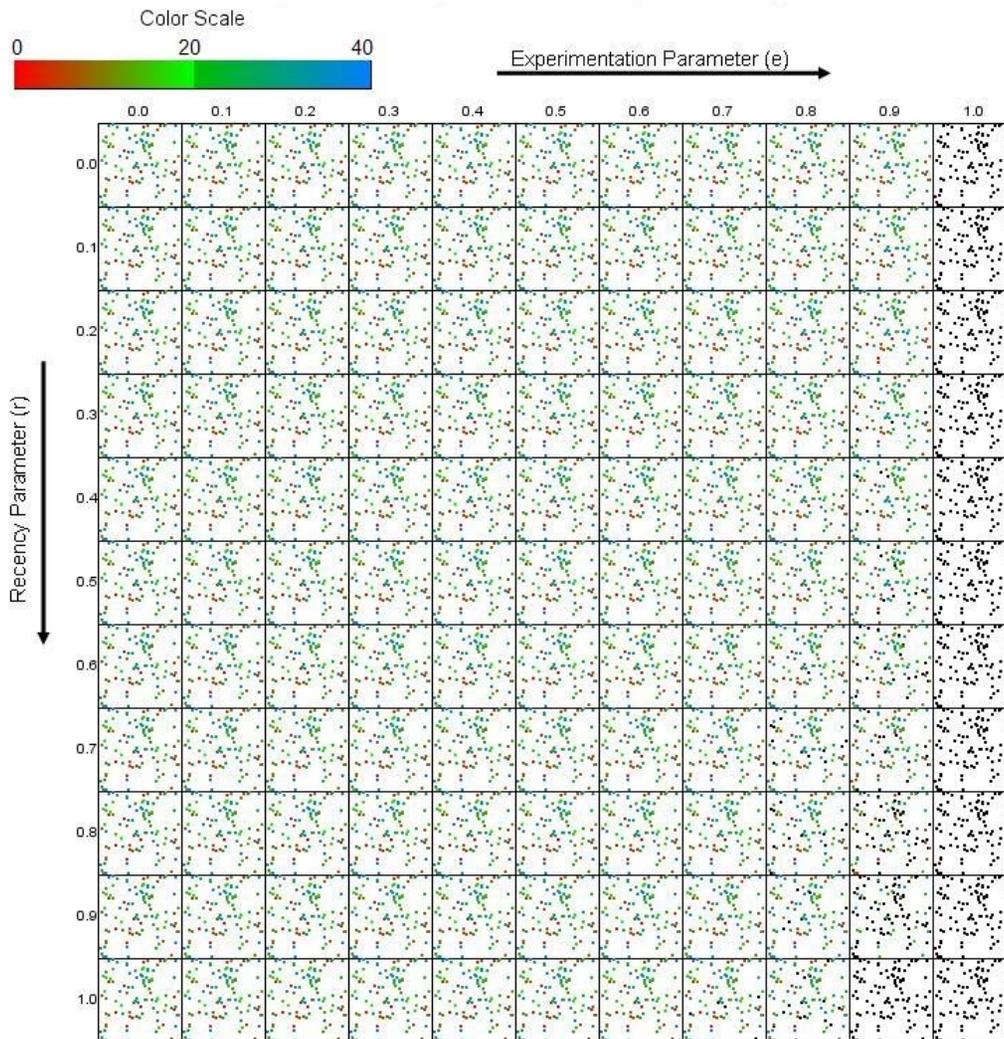
(Variant Roth – Erev RL Algorithm)



- A run for which none of G5's action choice probabilities exceed .999 at 100th tick is marked with black.
- In all other runs, the color indicates which of G5's 40 actions has a choice probability $> .999$ at the 100th tick.

AMESModel-I: Experiment 2 (Low Initial Propensity)

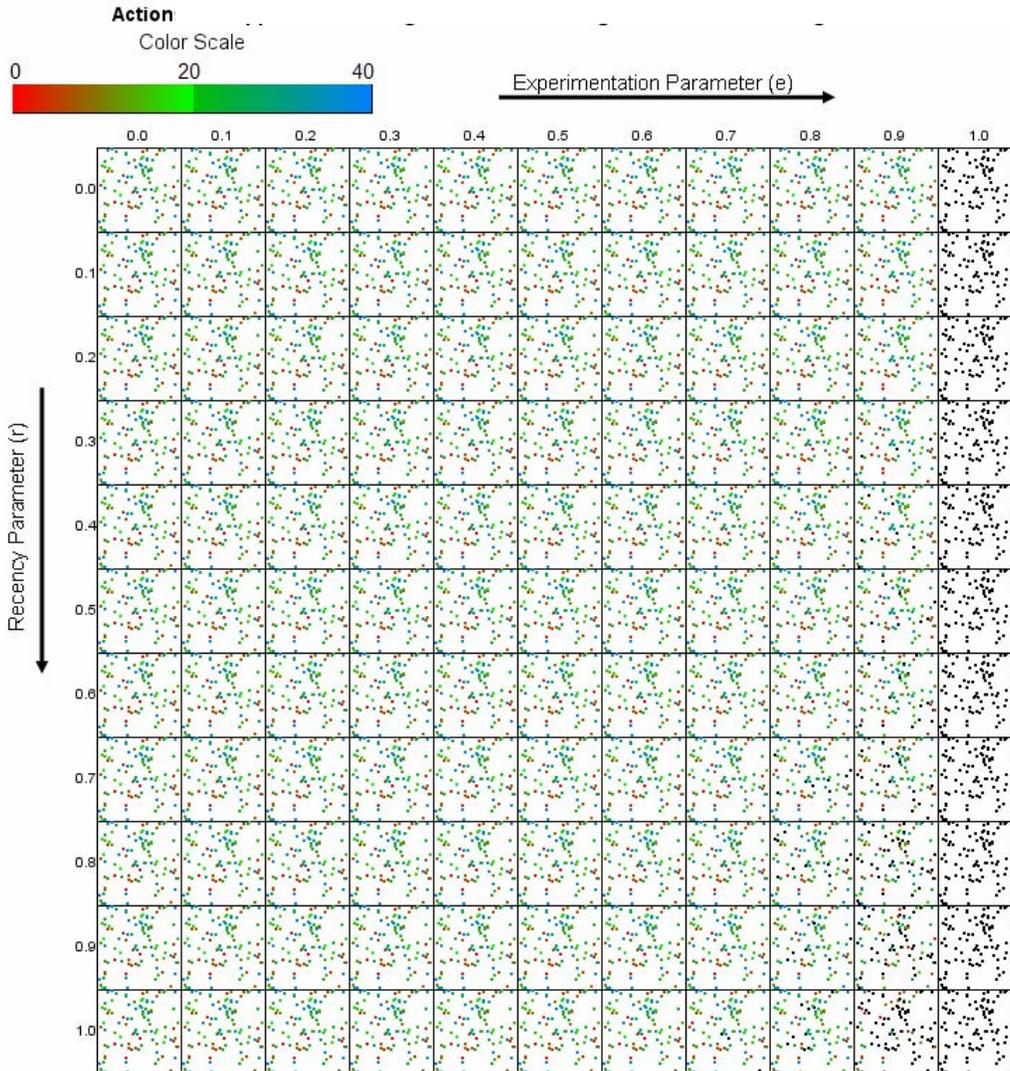
G5's Convergent Actions (Roth – Erev RL Algorithm)



- A run for which none of G5's action choice probabilities exceed .999 at 100th tick is marked with black.
- In all other runs, the color indicates which of G5's 40 actions has a choice probability $> .999$ at the 100th tick.

AMESModel-I: Experiment 2 (Low Initial Propensity)

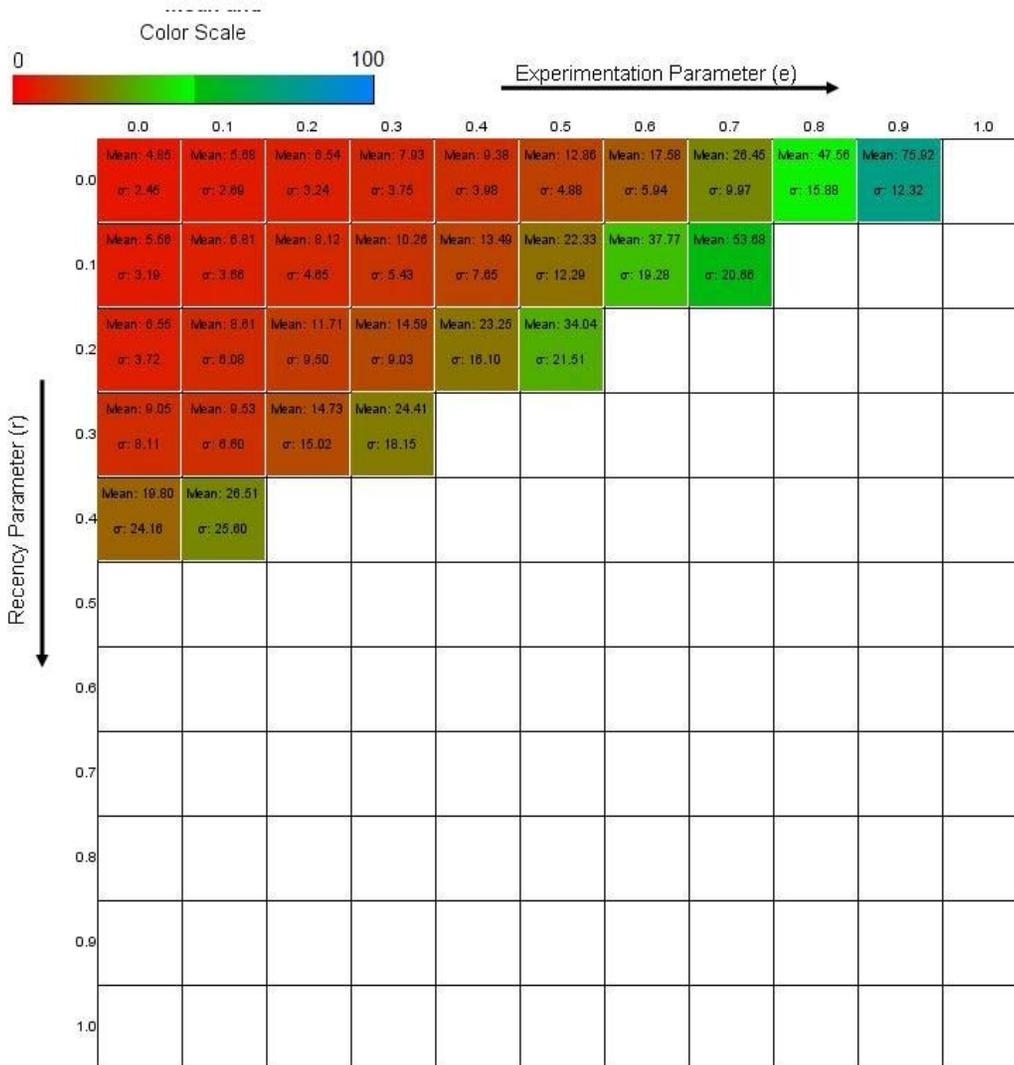
G5's Convergent Actions (Variant Roth – Erev RL Algorithm)



- A run for which none of G5's action choice probabilities exceed .999 at 100th tick is marked with black.
- In all other runs, the color indicates which of G5's 40 actions has a choice probability > .999 at the 100th tick.

AMESModel-I: Experiment 1 (High Initial Propensity)

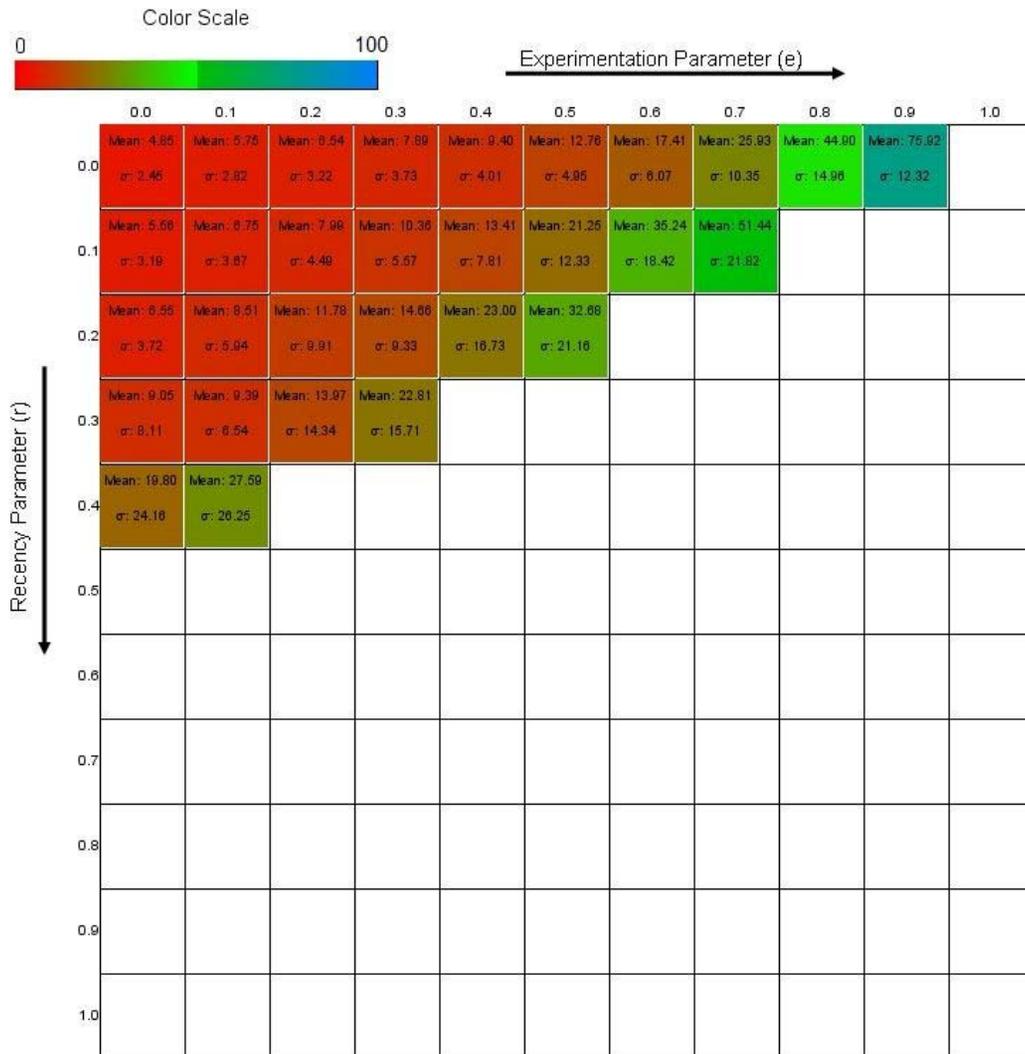
G5's Convergent Action Reaching Probability 0.999 (Roth – Erev RL Algorithm)



- The average time and std. deviation are reported for the Convergent Action of G5 (if any) to reach the threshold probability of 0.999, conditional on given settings $\{r, e\}$ for the recency parameter r and the experimentation parameter e .

AMESModel-I: Experiment 1 (High Initial Propensity)

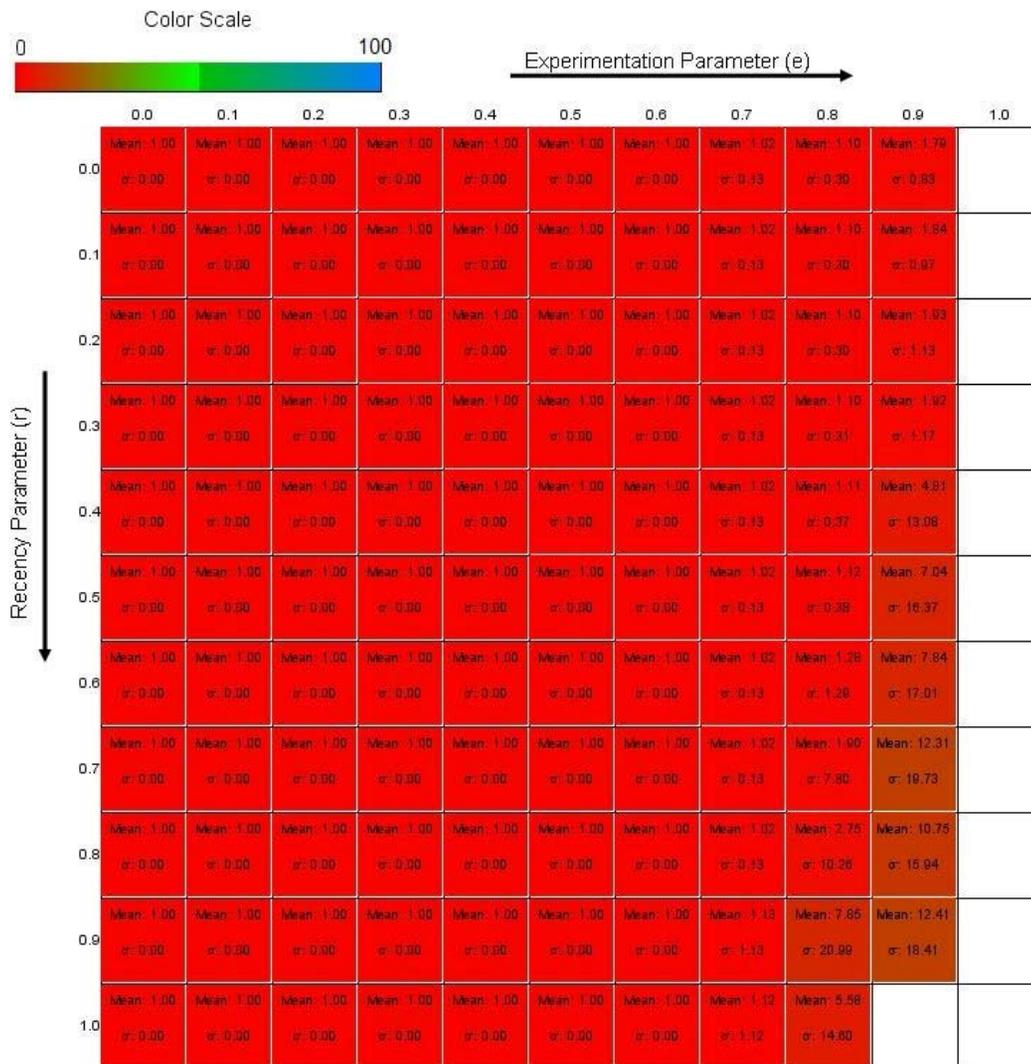
G5's Convergent Action Reaching Probability 0.999 (Variant Roth – Erev RL Algorithm)



- The average time and std. deviation are reported for the Convergent Action of G5 (if any) to reach the threshold probability of 0.999, conditional on given settings $\{r, e\}$ for the recency parameter r and the experimentation parameter e .

AMESModel-I: Experiment 2 (Low Initial Propensity)

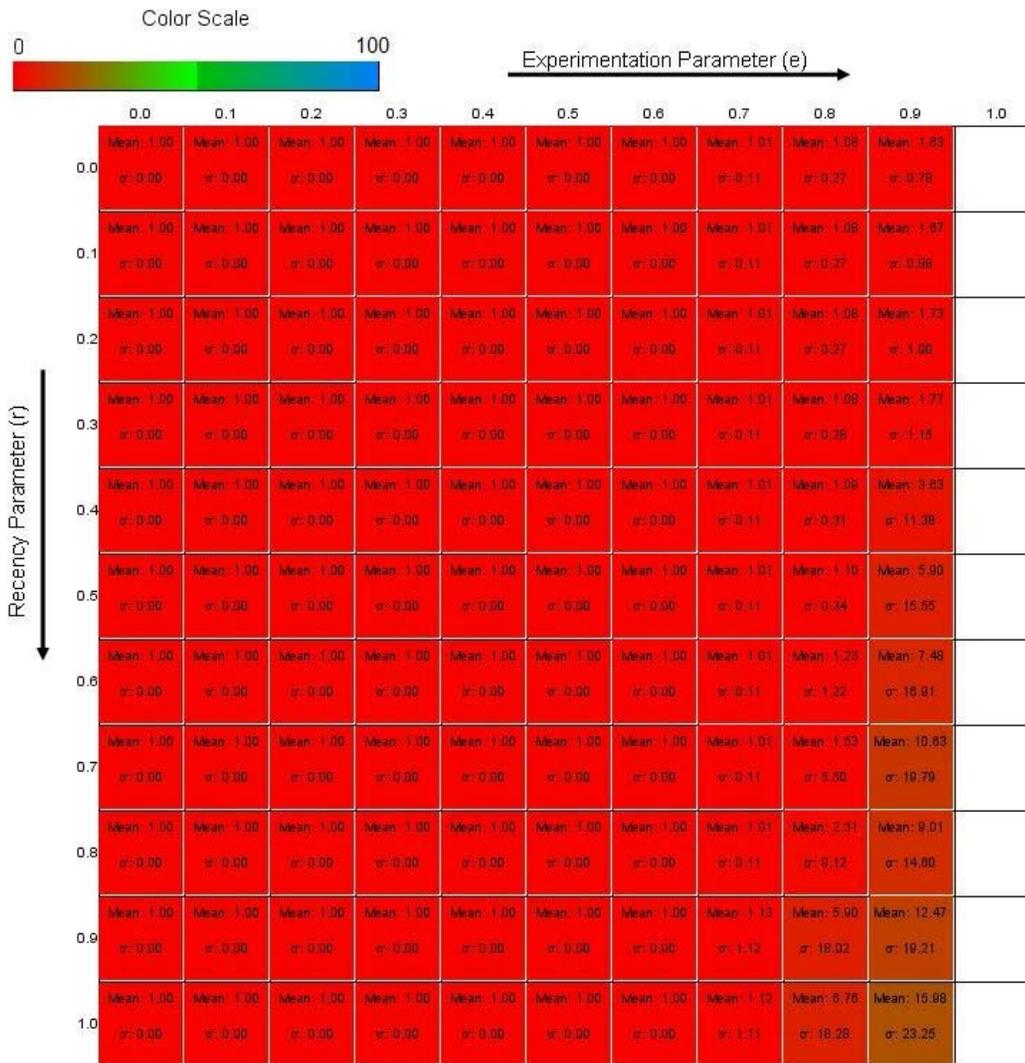
G5's Convergent Action Reaching Probability 0.999 (Roth – Erev RL Algorithm)



- The average time and std. deviation are reported for the Convergent Action of G5 (if any) to reach the threshold probability of 0.999, conditional on given settings $\{r, e\}$ for the recency parameter r and the experimentation parameter e .

AMESModel-I: Experiment 2 (Low Initial Propensity)

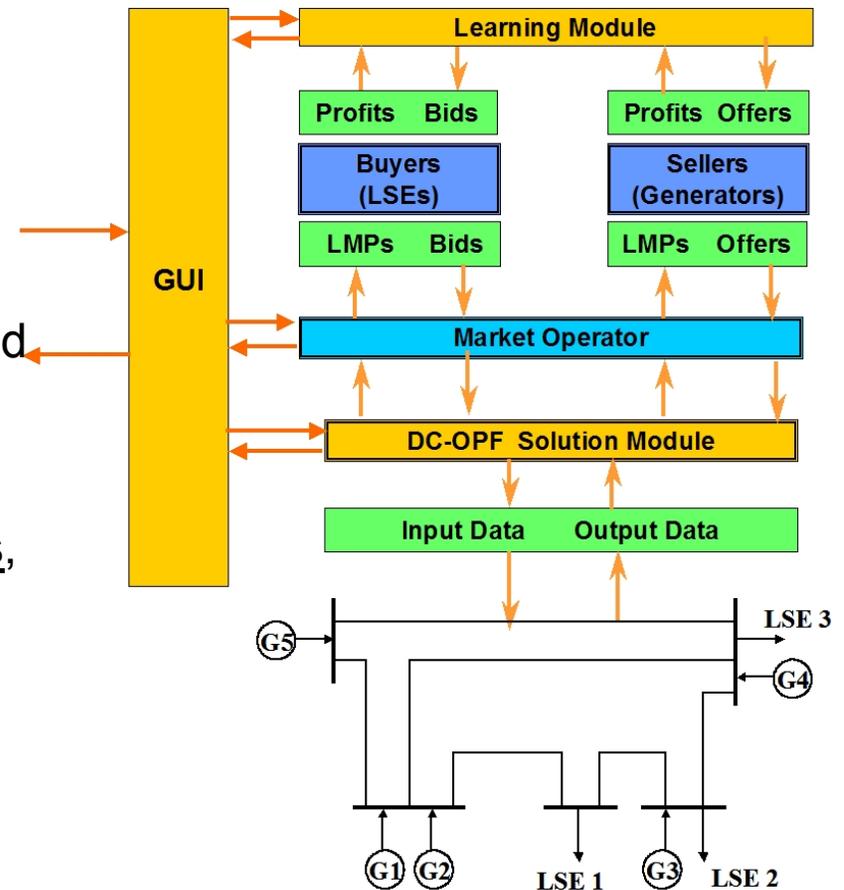
G5's Convergent Action Reaching Probability 0.999 (Variant Roth – Erev RL Algorithm)



- The average time and std. deviation are reported for the Convergent Action of G5 (if any) to reach the threshold probability of 0.999, conditional on given settings $\{r, e\}$ for the recency parameter r and the experimentation parameter e .

AMESModel-II: Two learning generators G4 and G5

- AMESModel-II makes use of a 5-node test case conducted with the AMES Market Package (Li, Sun, Tesfatsion, 2008).
- Generator G5 is a learning seller with 40 action choices (i.e., 40 possible supply offers).
- Generator G4 is now a learning seller with 40 action choices, with a VRE RL component. In all simulation runs, for G4, $\text{recency} = 0.04$, $\text{experimentation} = 0.97$, and $q_j(0)$ is same as that of G5 for all actions j .
- Each of the three remaining generators (sellers) has only 1 action choice.
- Hence, there are **two learning generators**, and **three non-learning generators**.
- LSEs (buyers) report fixed demand curves to the market operator each day.
- Each generator reports a supply curve to the market operator each day.
- The Market Operator uses daily reported demand/supply curves to solve for daily prices/quantities
- Each gen/LSE uses posted solution to compute its profits for each day.



AMES Market Package

Experimental Design for AMESModel-II

- Two experiments are carried out
 - Experiment 1 (High Initial Propensity for G5 and G4): Initial propensity values ($q_j(0)$) = 140,000.0 and a cooling parameter value $T = 35,000$
 - Experiment 2 (Low Initial Propensity for G5 and G4): Initial propensity values = 6,000.0 and $T = 1,000.0$
- Experimentation parameter (e) varied from 0.0 to 1.0 in increments of 0.1 for G5.
- Recency parameter (r) varied from 0.0 to 1.0 in increments of 0.1 for G5.
- 100 runs for each $\{r, e\}$ setting with a different initial random seed for each run.
- Each run consists of 100 market rounds, with the G5's profit ($\tau\pi$) calculated for each round
- For each run, at the end of the 100th round the Total Profits obtained by G5 over the run are calculated, and a “Convergent Action” (if any) for G5 is recorded.

Experimental Design (continued)

Experimentation parameter (e) →

		0.0	0.1	0.2	...	1.0
Recency parameter (r)	0.0	100 runs	100 runs	100 runs	• • •	100 runs
	0.1	100 runs				• • •
	.	• • •				• • •
	.					
	1.0	100 runs		• • •	• • •	100 runs

Initial propensity has settings of i) 140,000 with $T = 35,000$ or ii) 6000 with $T = 1000$

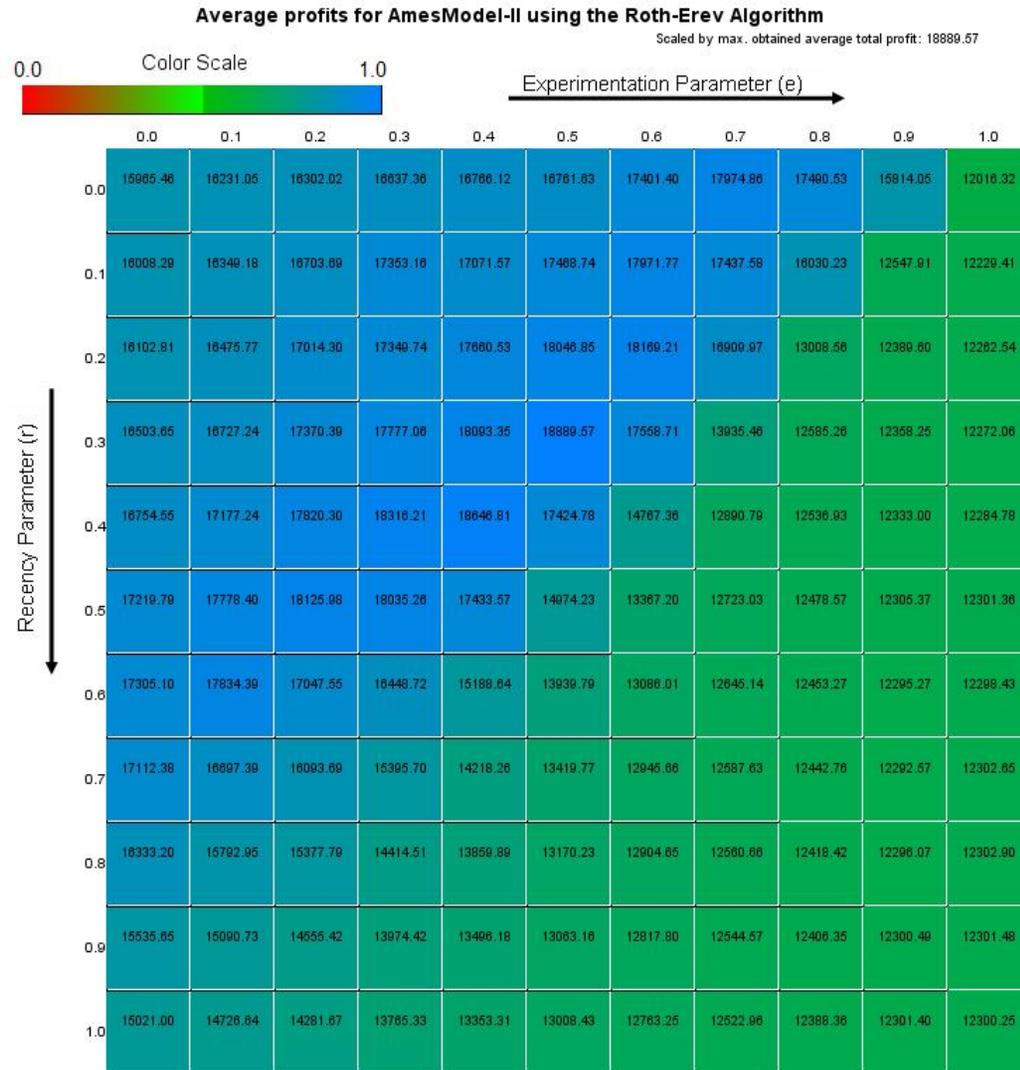
Profit of G5 for a run = Sum of profits for G5 obtained across all 100 rounds.

Total Profits of G5, given a $\{r, e\}$ setting = Sum of all profits for G5 in all runs with this $\{r, e\}$ setting

Average Total Profits = Total Profits divided by number of runs (for G5 for given $\{r, e\}$ setting)

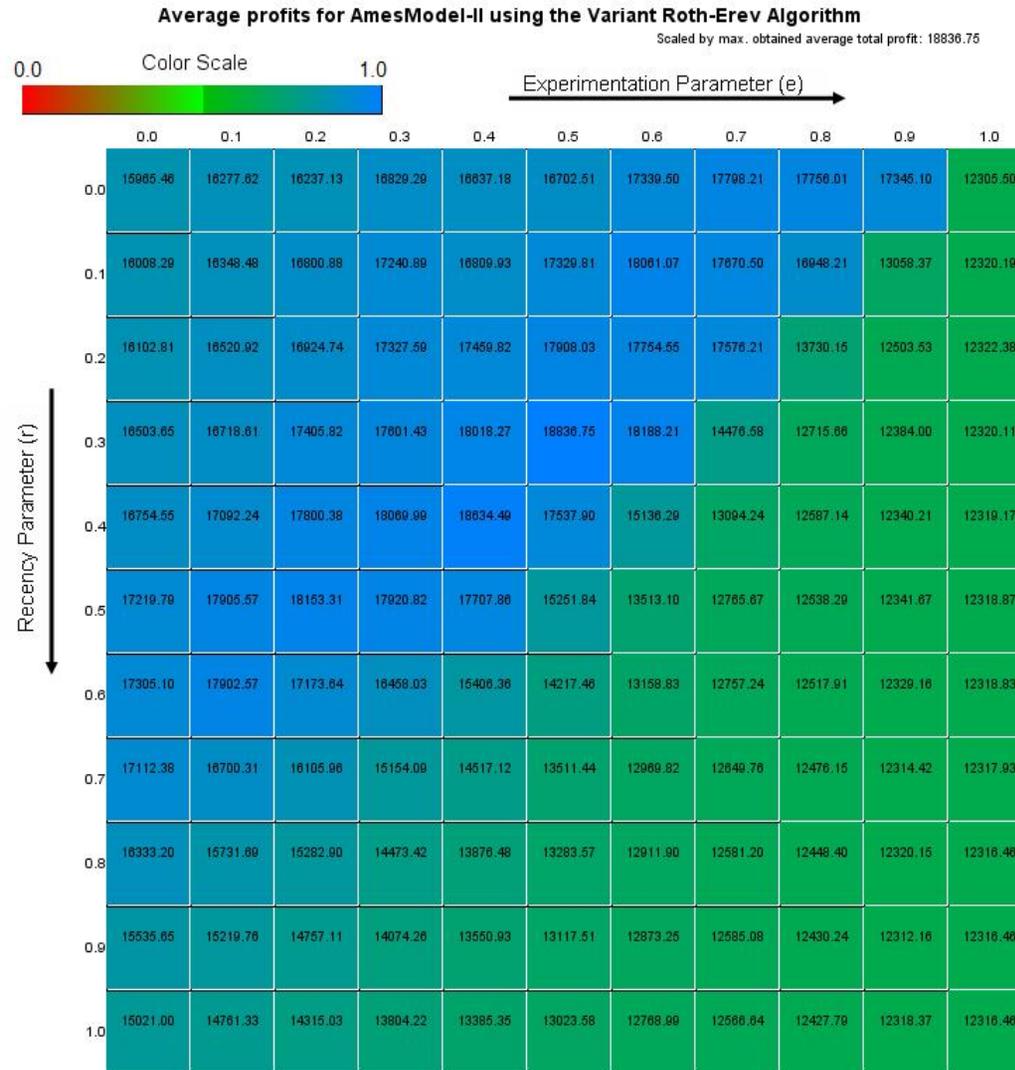
AMESModel-II: Experiment 1 (High Initial Propensity)

Average Total Profits for G5 (Roth-Erev RL Algorithm)



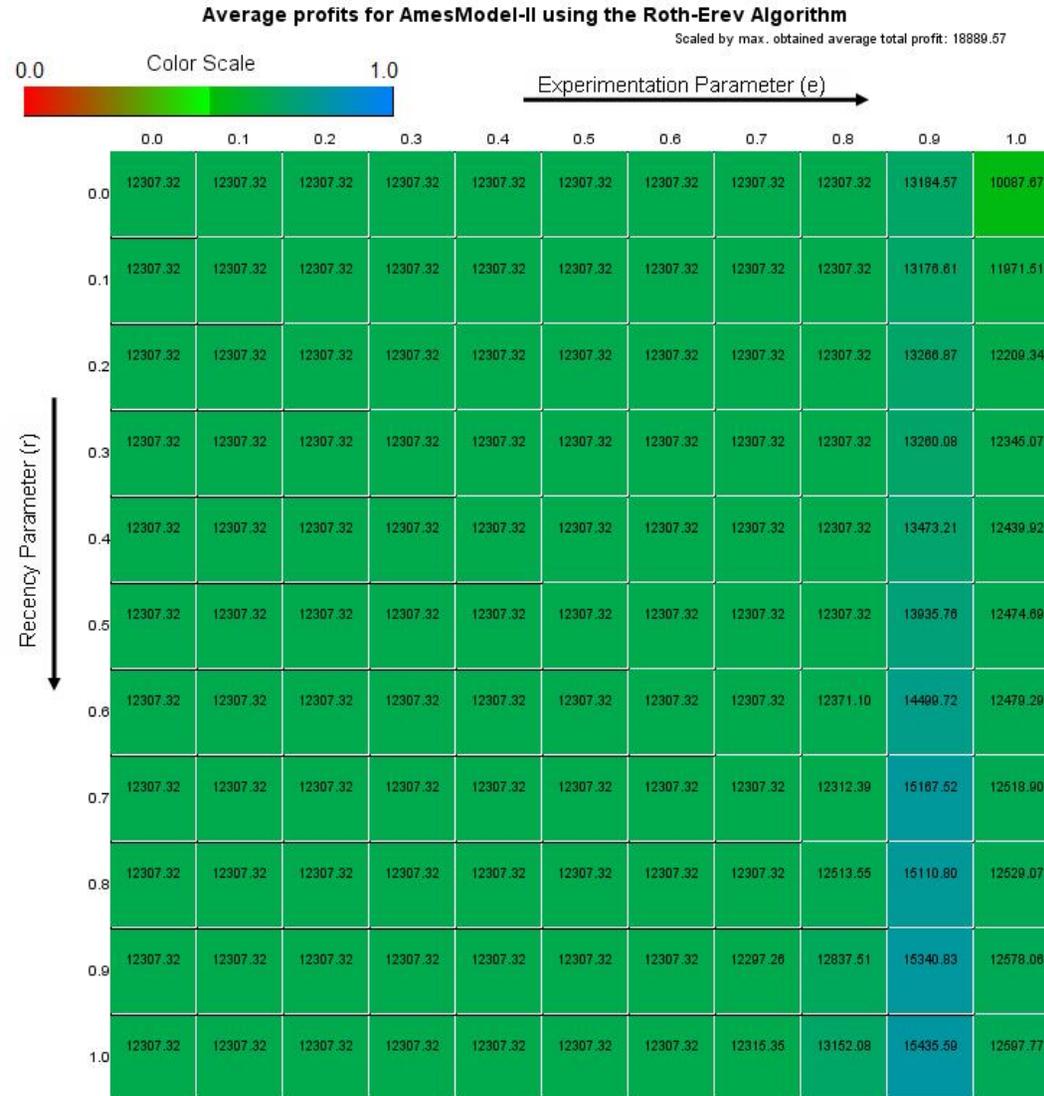
AMESModel-II: Experiment 1 (High Initial Propensity)

Average Total Profits for G5 (Variant Roth-Erev RL Algorithm)



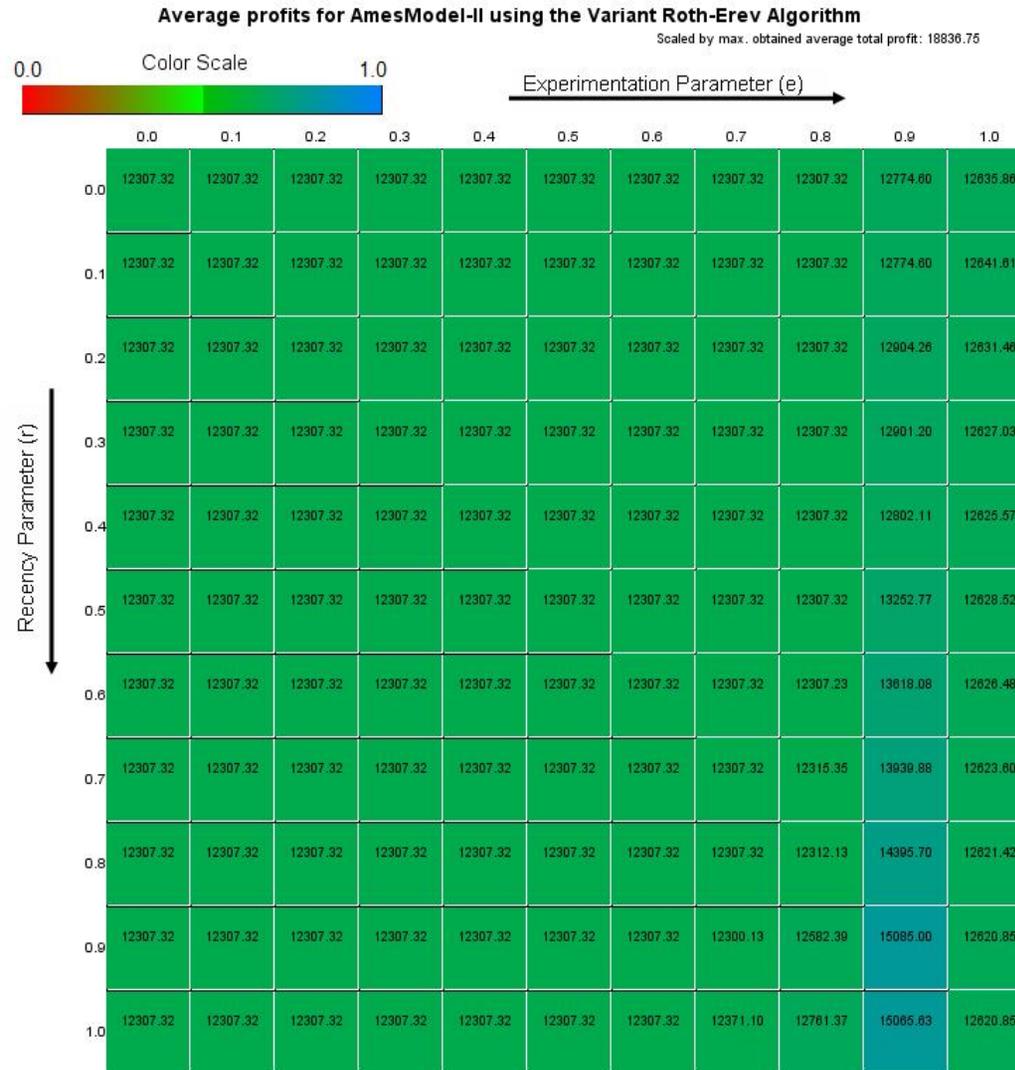
AMESModel-II: Experiment 2 (Low Initial Propensity)

Average Total Profits for G5 (Roth-Erev RL Algorithm)

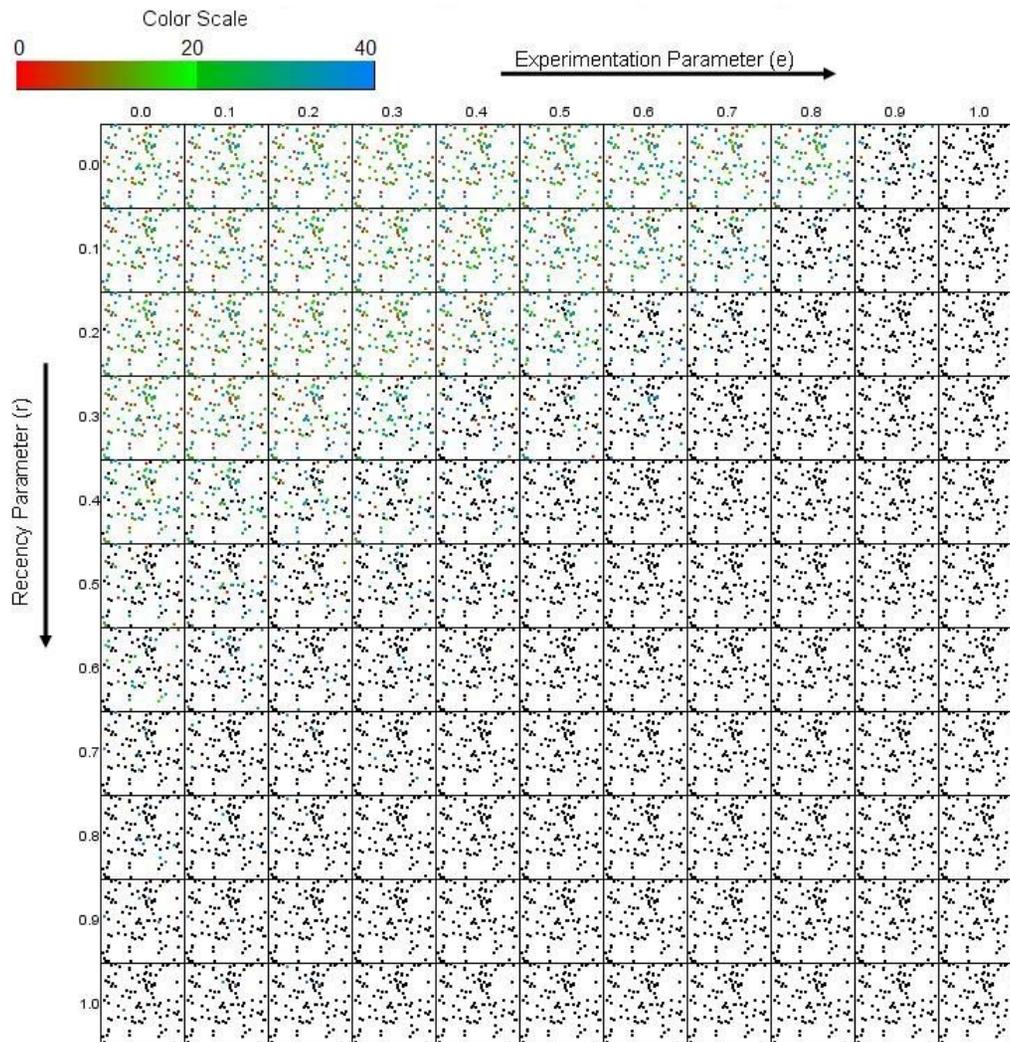


AMESModel-II: Experiment 2 (Low Initial Propensity)

Average Total Profits for G5 (Variant Roth-Erev RL Algorithm)



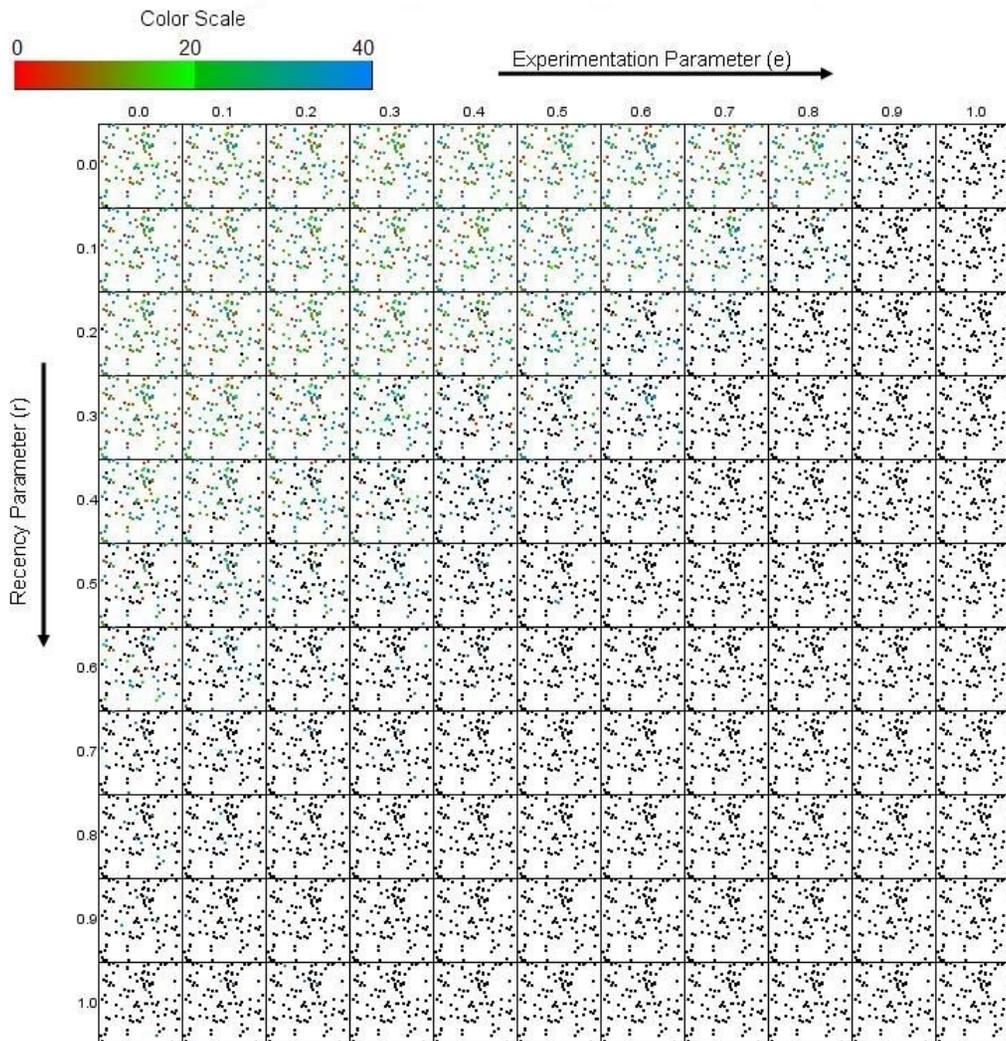
AMESModel-II: Experiment 1 (High Initial Propensity) G5's Convergent Actions (Roth – Erev RL Algorithm)



- A run for which none of G5's action choice probabilities exceed .999 at 100th tick is marked with black.
- In all other runs, the color indicates which of G5's 40 actions has a choice probability $> .999$ at the 100th tick.

AMESModel-II: Experiment 1 (High Initial Propensity)

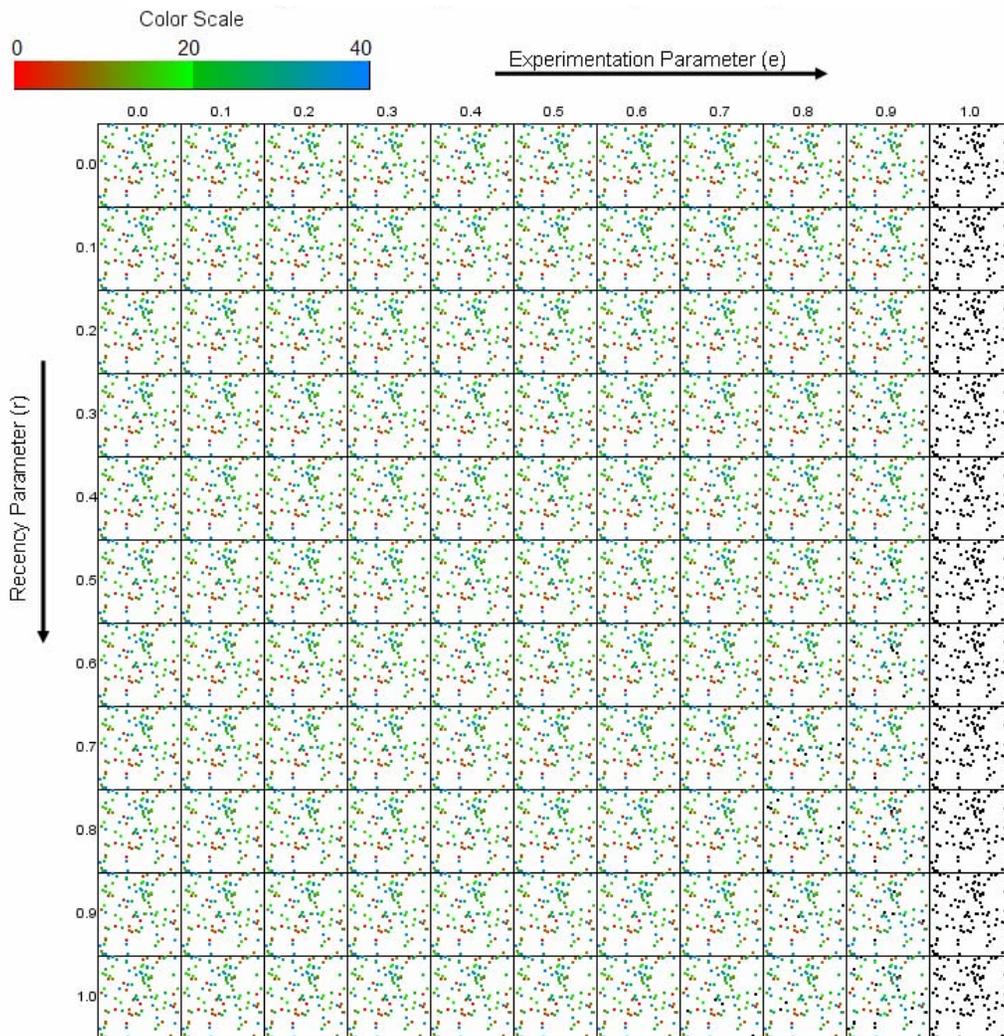
G5's Convergent Actions (Variant Roth – Erev RL Algorithm)



- A run for which none of G5's action choice probabilities exceed .999 at 100th tick is marked with black.
- In all other runs, the color indicates which of G5's 40 actions has a choice probability $> .999$ at the 100th tick.

AMESModel-II: Experiment 2 (Low Initial Propensity)

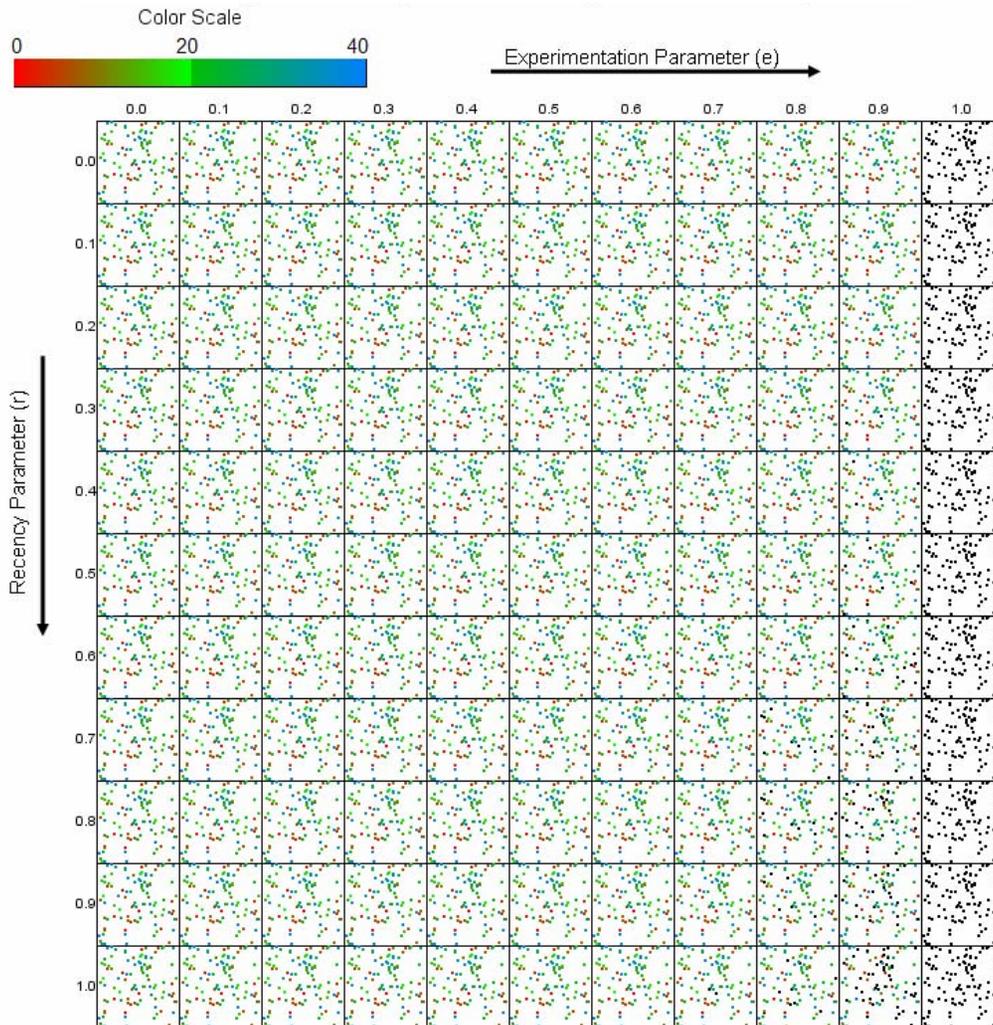
G5's Convergent Actions (Roth – Erev RL Algorithm)



- A run for which none of G5's action choice probabilities exceed .999 at 100th tick is marked with black.
- In all other runs, the color indicates which of G5's 40 actions has a choice probability $> .999$ at the 100th tick.

AMESModel-II: Experiment 2 (Low Initial Propensity)

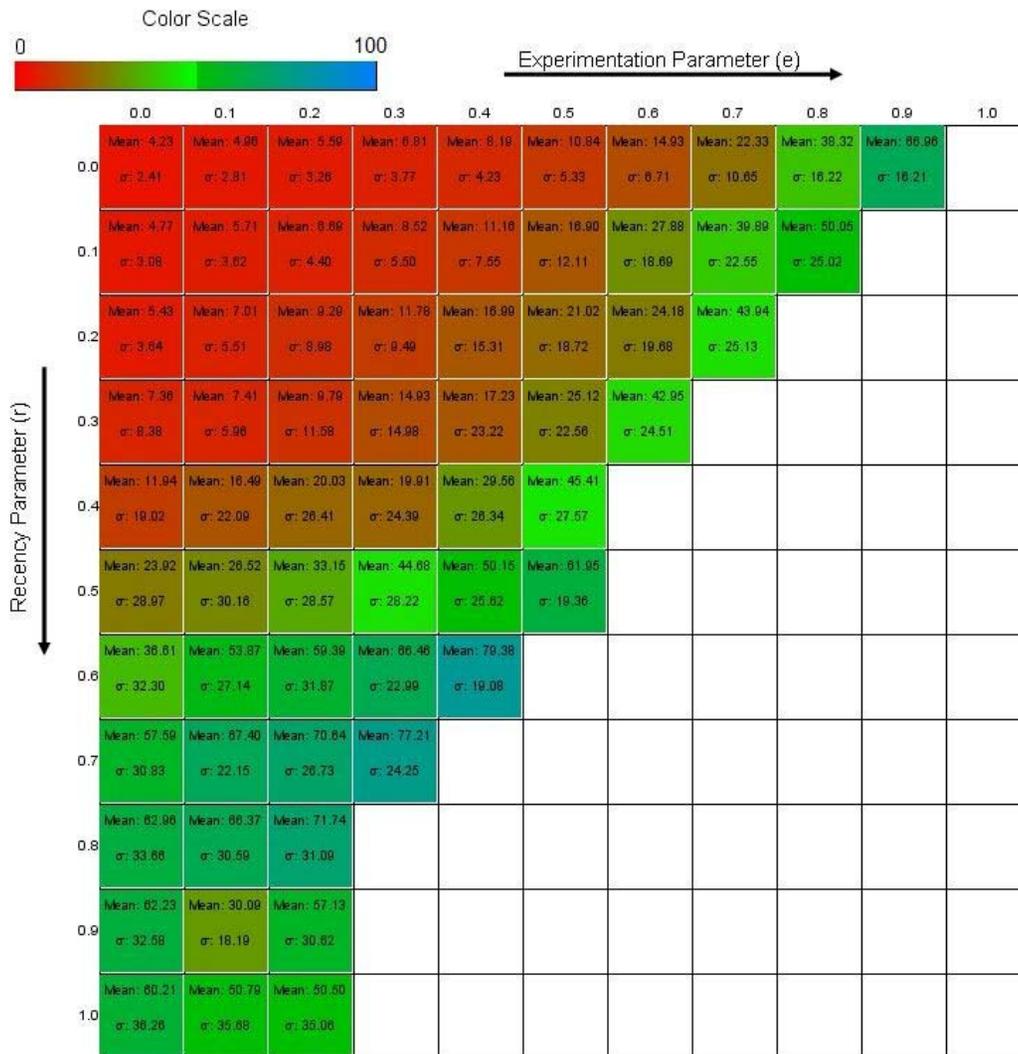
G5's Convergent Actions (Variant Roth – Erev RL Algorithm)



- A run for which none of G5's action choice probabilities exceed .999 at 100th tick is marked with black.
- In all other runs, the color indicates which of G5's 40 actions has a choice probability $> .999$ at the 100th tick.

AMESModel-II: Experiment 1 (High Initial Propensity)

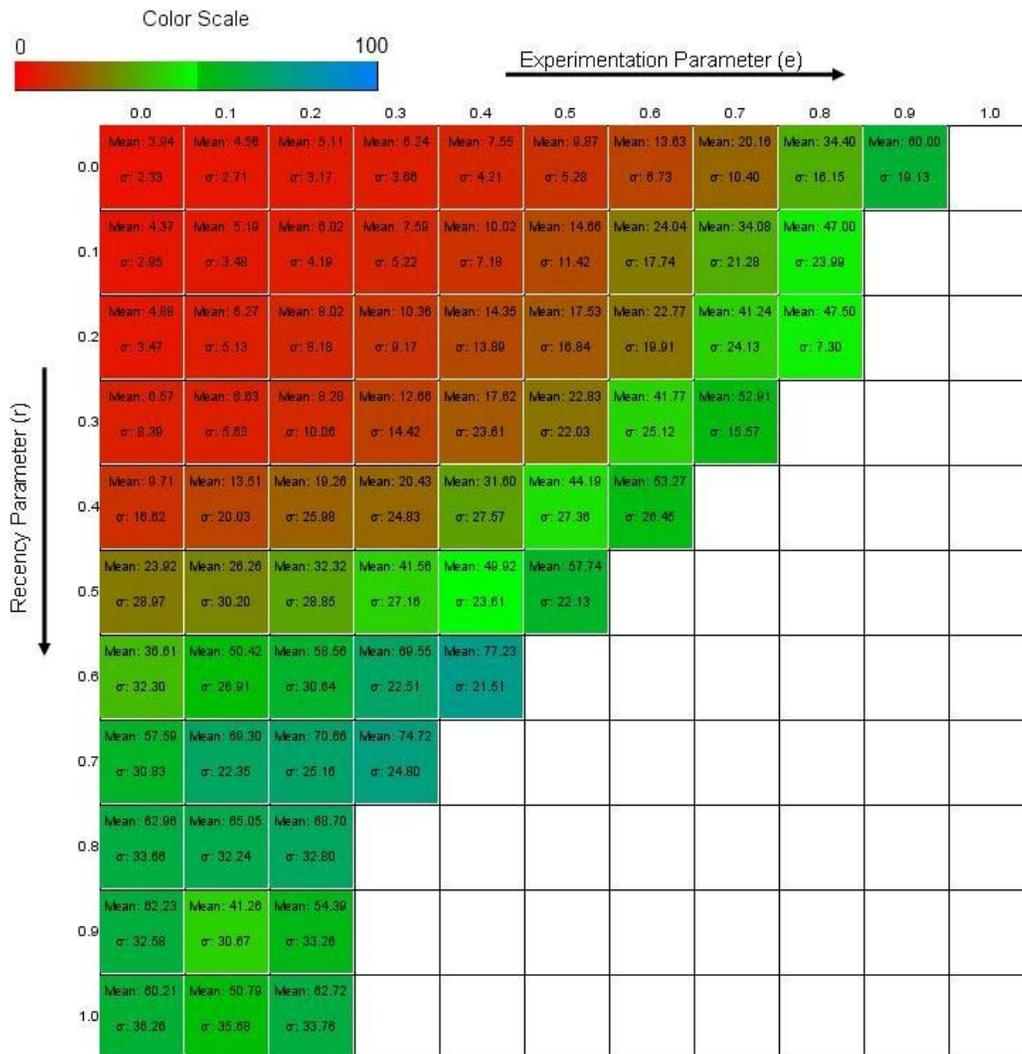
G5's Convergent Action Reaching Probability 0.999 (Roth – Erev RL Algorithm)



- The average time and std. deviation are reported for the Convergent Action of G5 (if any) to reach the threshold probability of 0.999, conditional on given settings $\{r, e\}$ for the recency parameter r and the experimentation parameter e .

AMESModel-II: Experiment 1 (High Initial Propensity)

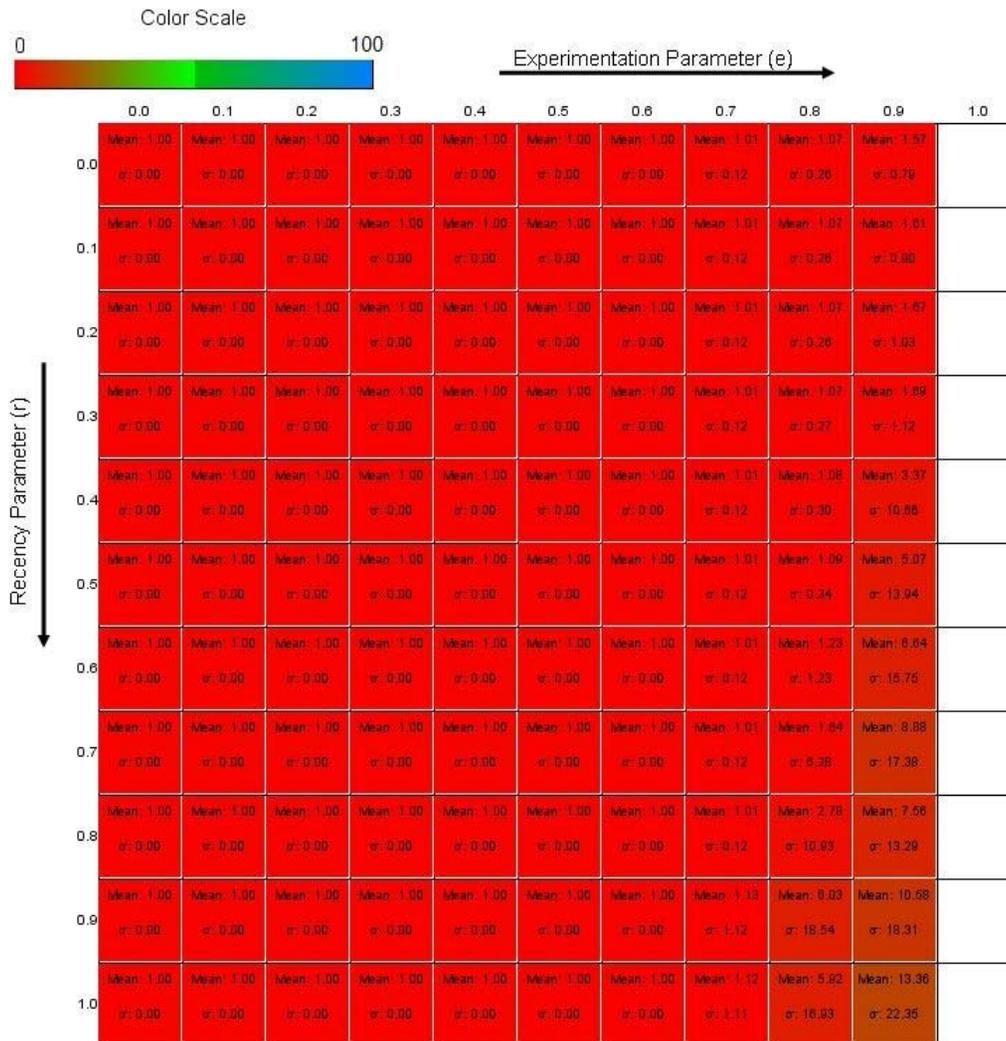
G5's Convergent Action Reaching Probability 0.999 (Variant Roth – Erev RL Algorithm)



- The average time and std. deviation are reported for the Convergent Action of G5 (if any) to reach the threshold probability of 0.999, conditional on given settings $\{r, e\}$ for the recency parameter r and the experimentation parameter e .

AMESModel-II: Experiment 2 (Low Initial Propensity)

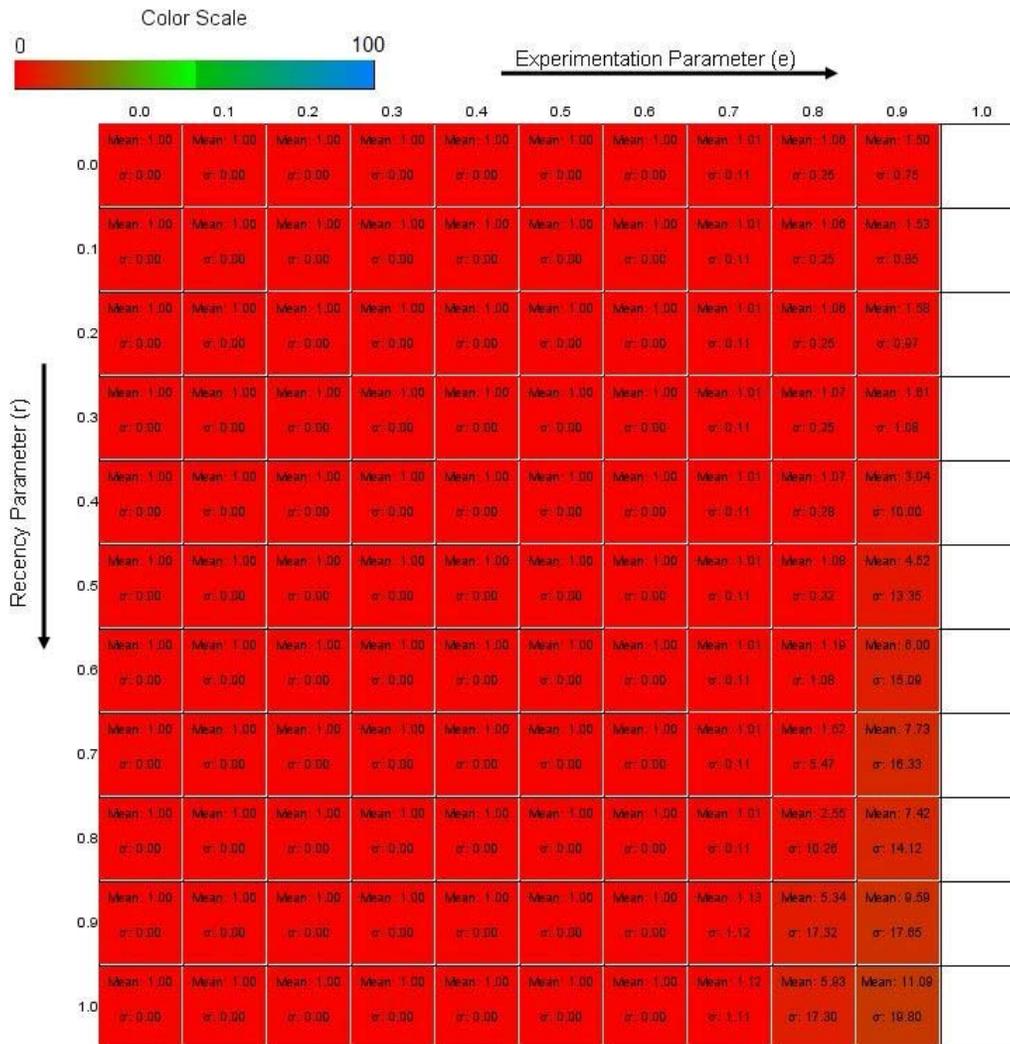
G5's Convergent Action Reaching Probability 0.999 (Roth – Erev RL Algorithm)



- The average time and std. deviation are reported for the Convergent Action of G5 (if any) to reach the threshold probability of 0.999, conditional on given settings $\{r, e\}$ for the recency parameter r and the experimentation parameter e .

AMESModel-II: Experiment 2 (Low Initial Propensity)

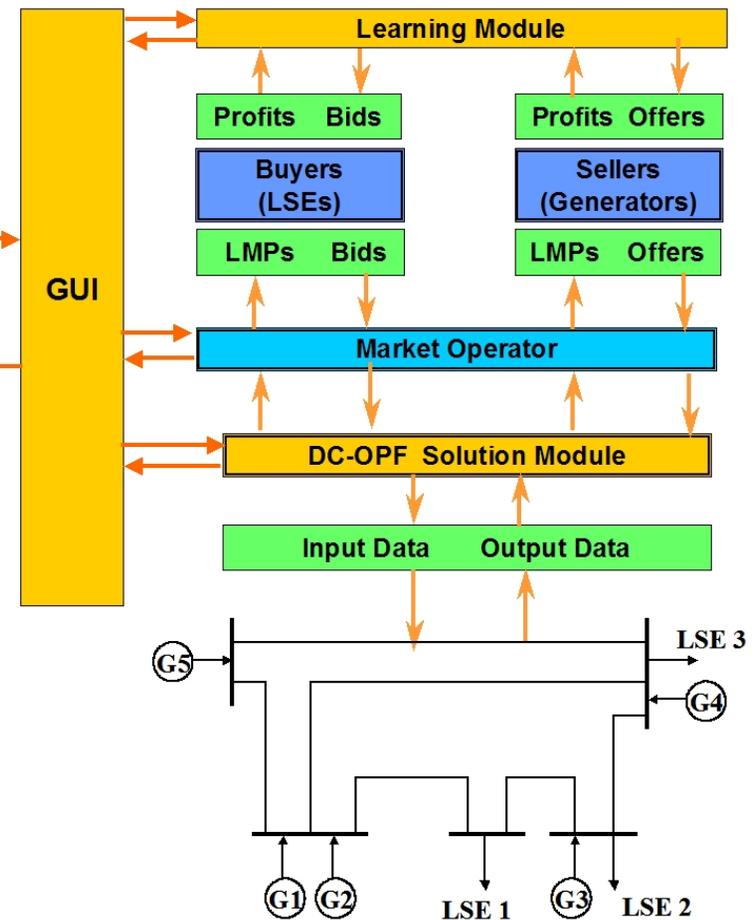
G5's Convergent Action Reaching Probability 0.999 (Variant Roth – Erev RL Algorithm)



- The average time and std. deviation are reported for the Convergent Action of G5 (if any) to reach the threshold probability of 0.999, conditional on given settings $\{r, e\}$ for the recency parameter r and the experimentation parameter e .

AMESModel-III: All Five Learning Sellers G1 – G5

- AMESModel-III makes use of a 5-node test case conducted with the AMES Market Package (Li, Sun, Tesfatsion, 2008).
- Generator G5 is a learning seller with 40 action choices (i.e., 40 possible supply offers).
- Generators G1-G4 are now learning sellers with 40 action choices, with a VRE RL component. In all simulation runs, for G1-G4, recency = 0.04, experimentation = 0.97, and $q_j(0)$ = initial propensity of G5 for all actions j .
- Hence, there are **five learning generators**, and **zero non-learning generators**.
- LSEs (buyers) report fixed demand curves to the market operator each day.
- Each generator reports a supply curve to the market operator each day.
- The Market Operator uses daily reported demand/supply curves to solve for daily prices/quantities
- Each gen/LSE uses posted solution to compute its profits for each day.



AMES Market Package

Experimental Design for AMESModel-III

- Two experiments are carried out
 - Experiment 1 (High Initial Propensity for G1-G5): Initial propensity values ($q_j(0)$) = 140,000.0 and a cooling parameter value $T = 35,000$
 - Experiment 2 (Low Initial Propensity for G1-G5): Initial propensity values = 6,000.0 and $T = 1,000.0$
- Experimentation (e) value varied from 0.0 to 1.0 in increments of 0.1 for G5.
- Recency (r) value varied from 0.0 to 1.0 in increments of 0.1 for G5.
- 100 runs for each $\{r, e\}$ setting with a different initial random seed for each run.
- Each run consists of 100 market rounds, with the G5's profit (π) calculated for each round
- For each run, at the end of the 100th round the Total Profits obtained by G5 over the run are calculated, and a “Convergent Action” (if any) for G5 is recorded.

Experimental Design (continued)

Experimentation parameter (e) →

		0.0	0.1	0.2	...	1.0
Recency parameter (r)	0.0	100 runs	100 runs	100 runs	• • •	100 runs
	0.1	100 runs				• • •
	.	• • •				• • •
	.					
	1.0	100 runs		• • •	• • •	100 runs

Initial propensity has settings of i) 140,000 with $T = 35,000$ or ii) 6000 with $T = 1000$

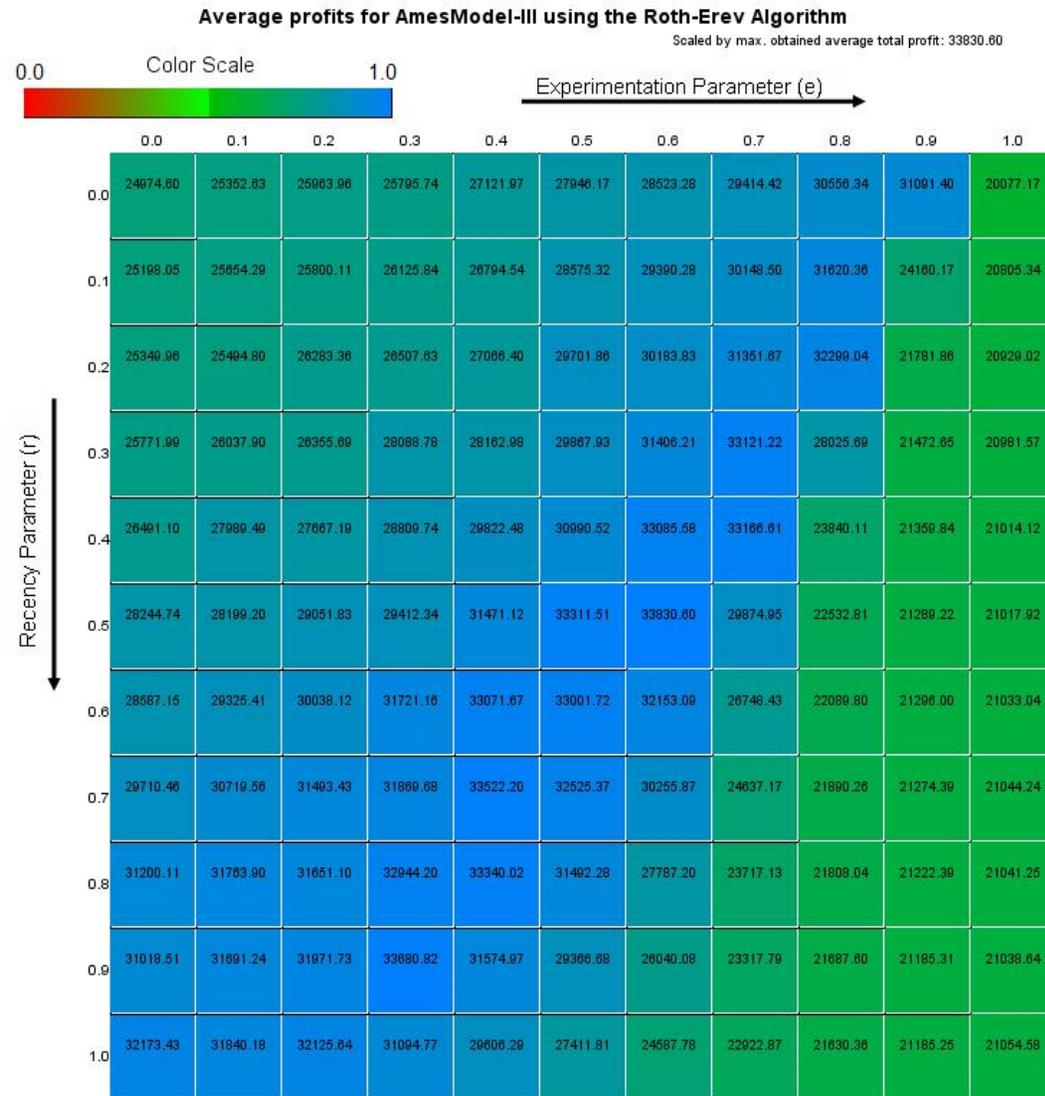
Profit of G5 for a run = Sum of profits for G5 obtained across all 100 rounds.

Total Profits of G5, given a $\{r, e\}$ setting = Sum of all profits for G5 in all runs with this $\{r, e\}$ setting

Average Total Profits = Total Profits divided by number of runs (for G5 for given $\{r, e\}$ setting)

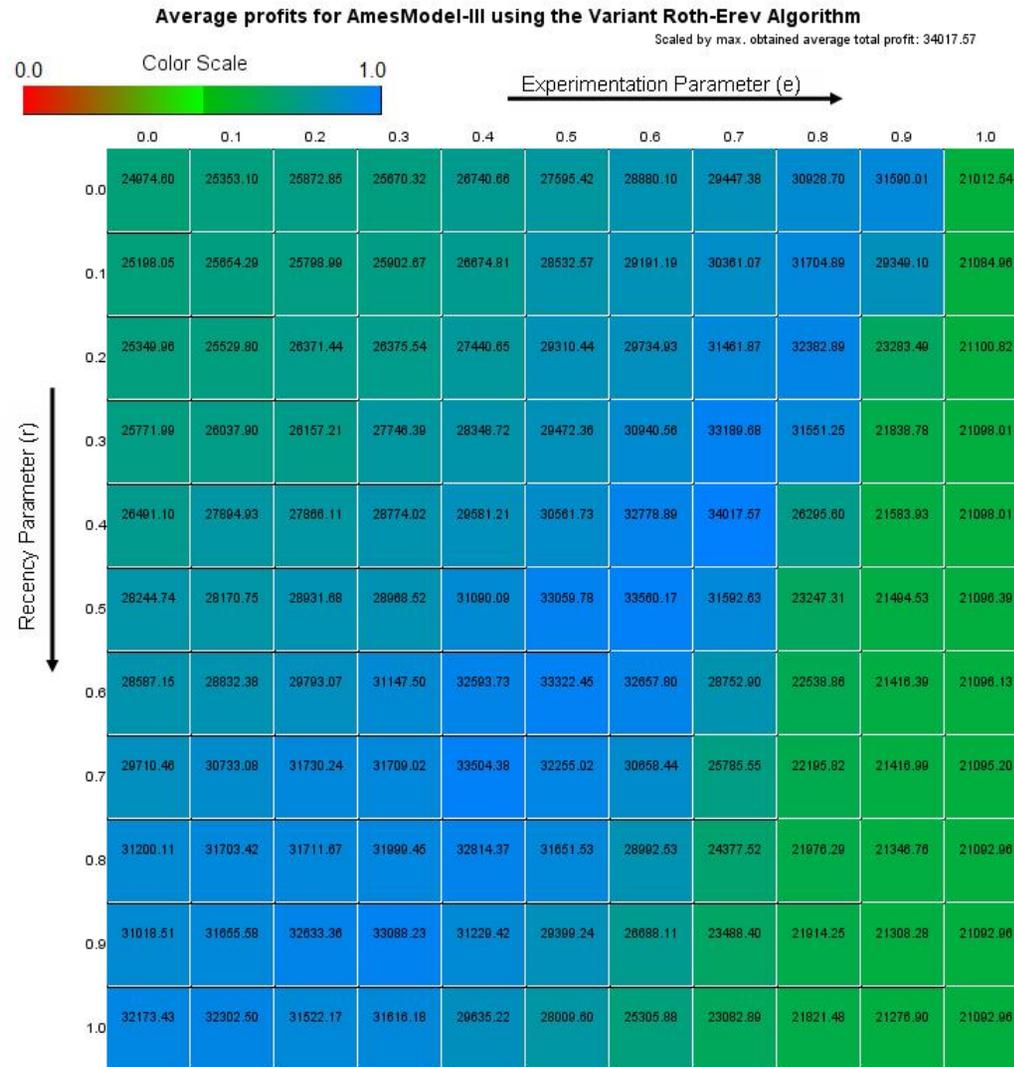
AMESModel-III: Experiment 1 (High Initial Propensity)

Average Total Profits for G5 (Roth-Erev RL Algorithm)



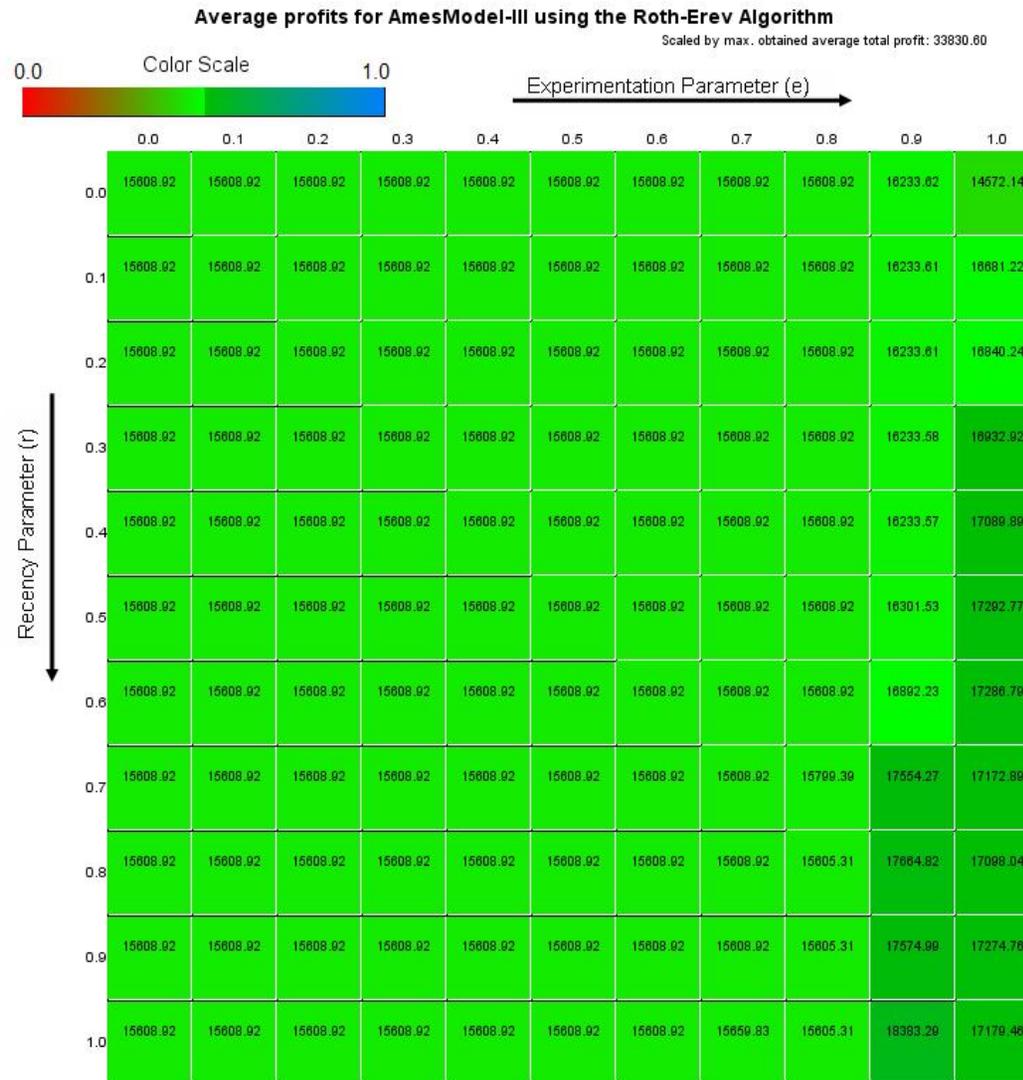
AMESModel-III: Experiment 1 (High Initial Propensity)

Average Total Profits for G5 (Variant Roth-Erev RL Algorithm)



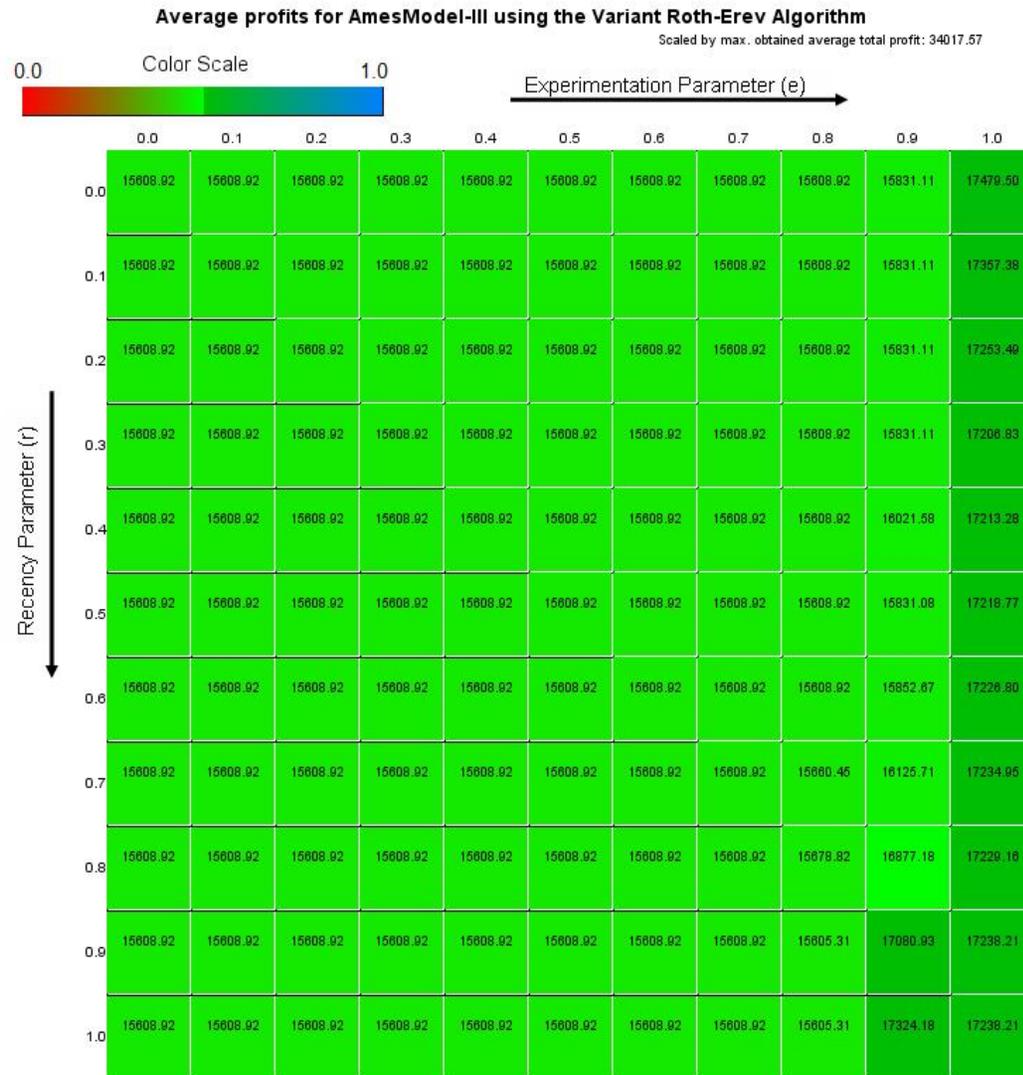
AMESModel-III: Experiment 2 (Low Initial Propensity)

Average Total Profits for G5 (Roth-Erev RL Algorithm)



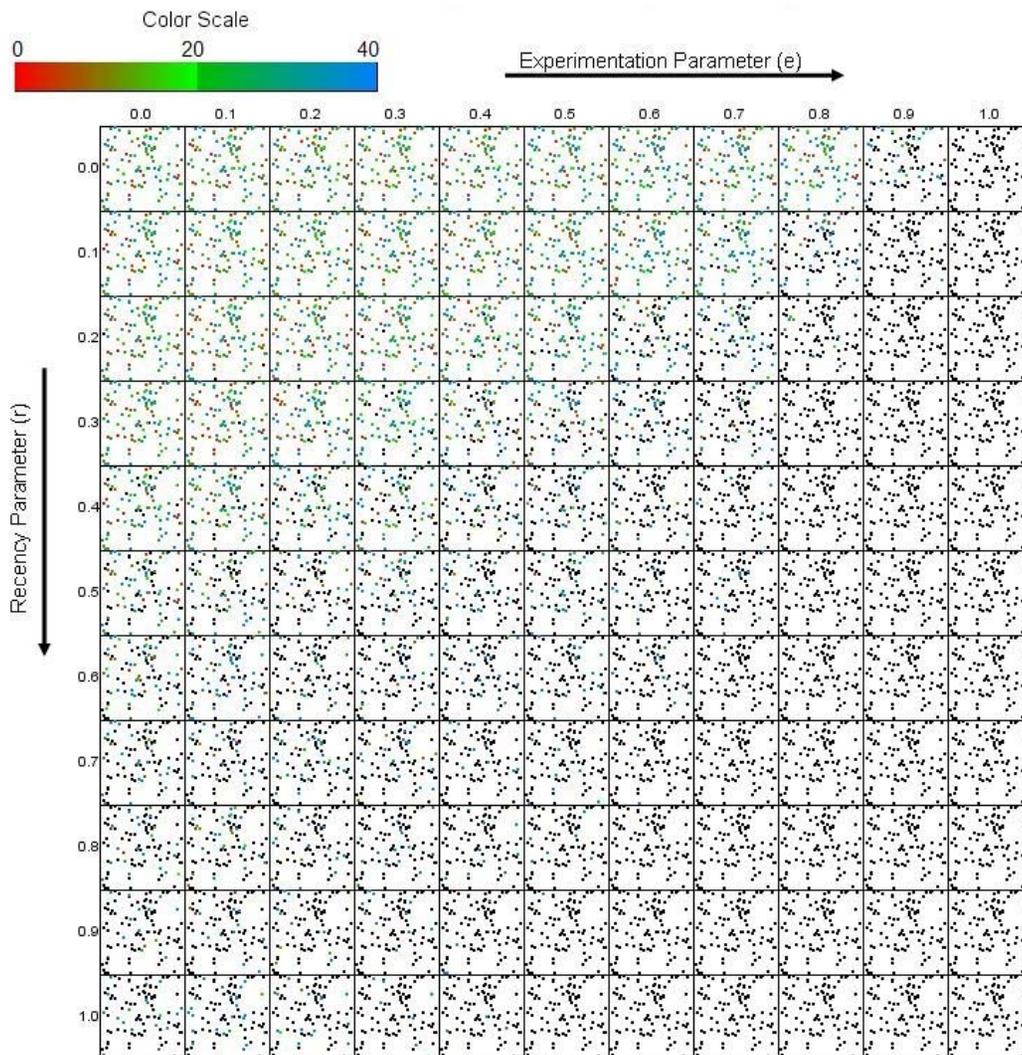
AMESModel-III: Experiment 2 (Low Initial Propensity)

Average Total Profits for G5 (Variant Roth-Erev RL Algorithm)



AMESModel-III: Experiment 1 (High Initial Propensity)

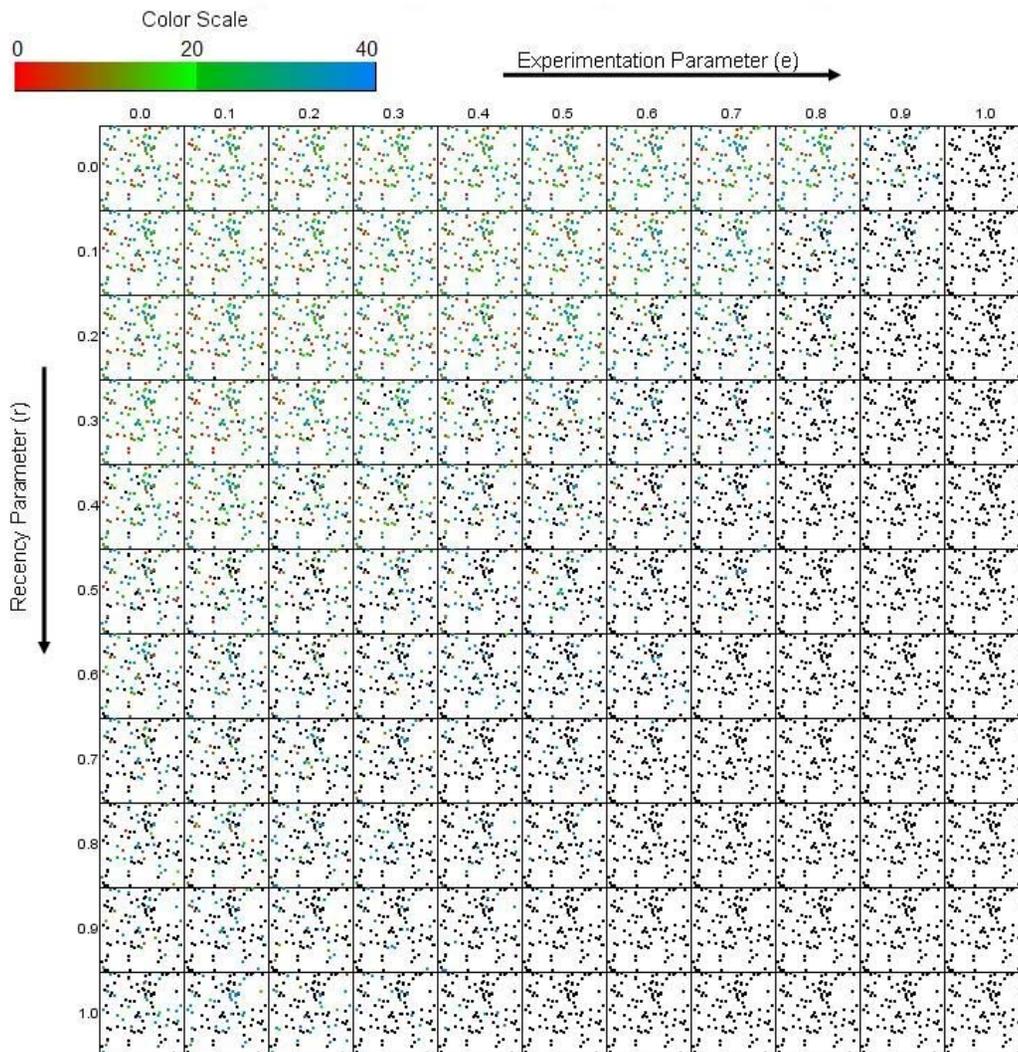
G5's Convergent Actions (Roth – Erev RL Algorithm)



- A run for which none of G5's action choice probabilities exceed .999 at 100th tick is marked with black.
- In all other runs, the color indicates which of G5's 40 actions has a choice probability > .999 at the 100th tick.

AMESModel-III: Experiment 1 (High Initial Propensity)

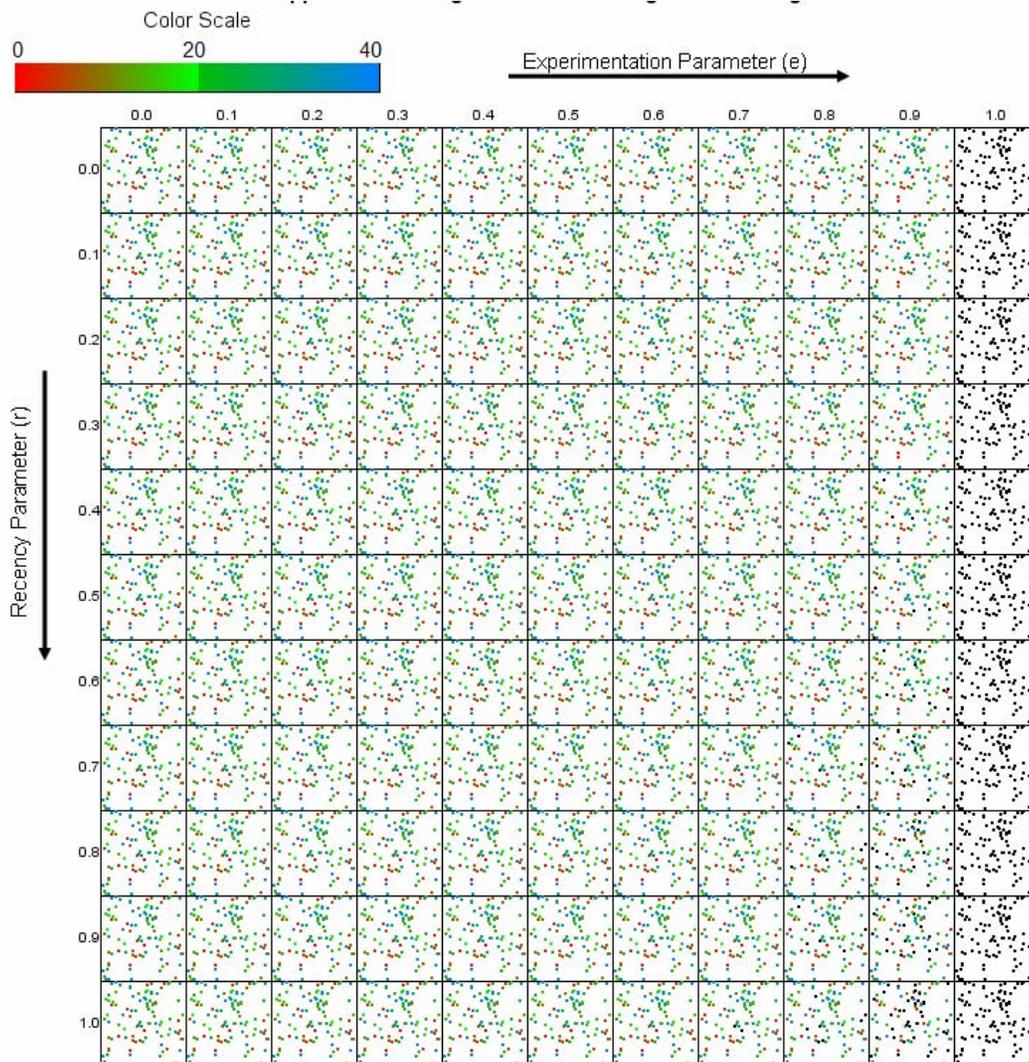
G5's Convergent Actions (Variant Roth – Erev RL Algorithm)



- A run for which none of G5's action choice probabilities exceed .999 at 100th tick is marked with black.
- In all other runs, the color indicates which of G5's 40 actions has a choice probability $> .999$ at the 100th tick.

AMESModel-III: Experiment 2 (Low Initial Propensity)

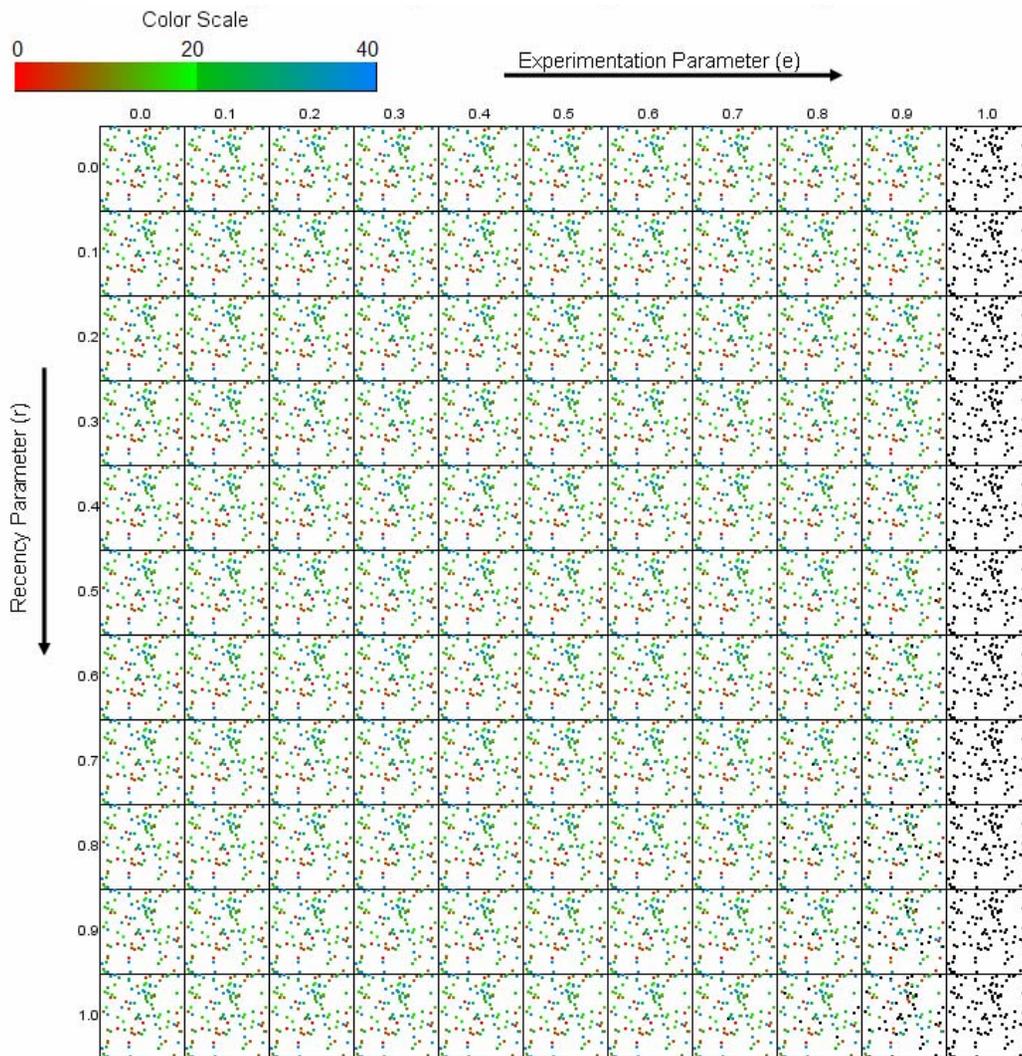
G5's Convergent Actions (Roth – Erev RL Algorithm)



- A run for which none of G5's action choice probabilities exceed .999 at 100th tick is marked with black.
- In all other runs, the color indicates which of G5's 40 actions has a choice probability $> .999$ at the 100th tick.

AMESModel-III: Experiment 2 (Low Initial Propensity)

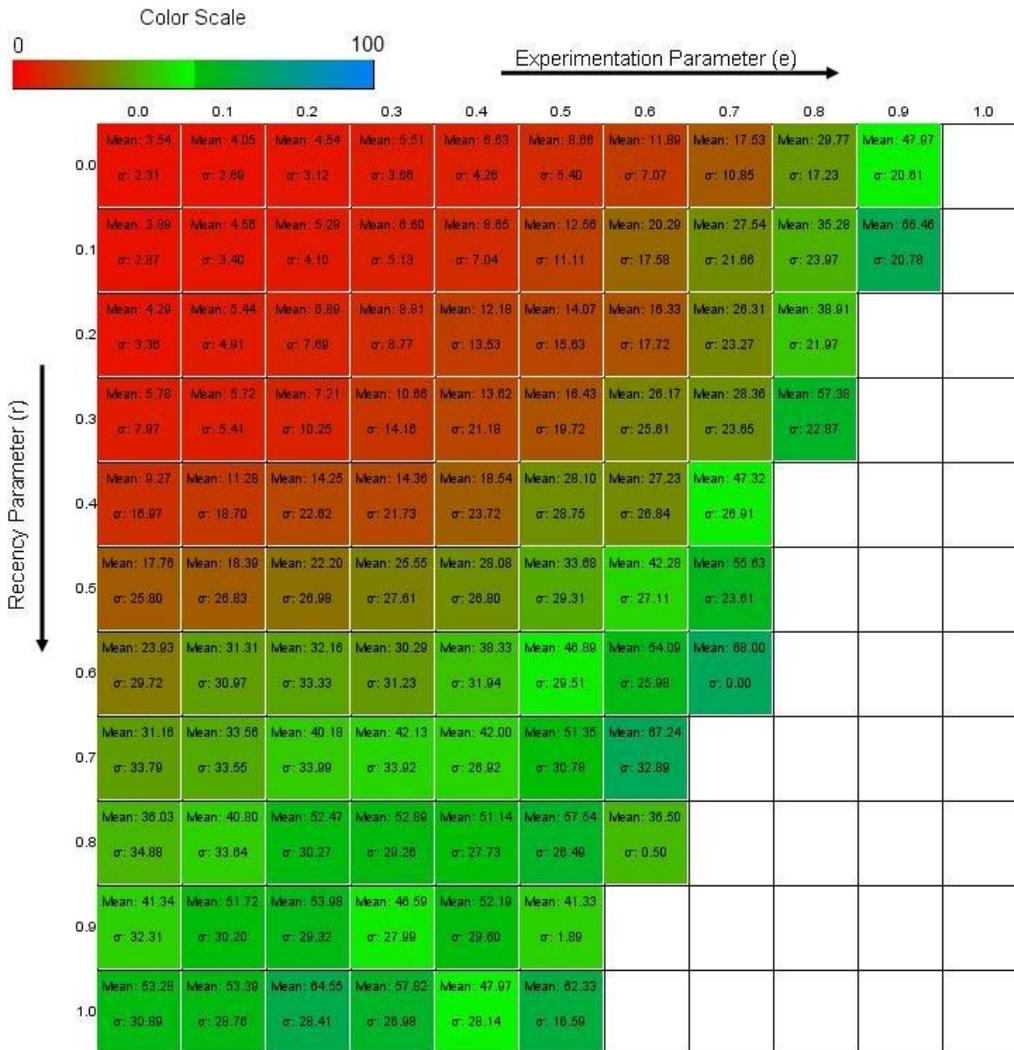
G5's Convergent Actions (Variant Roth – Erev RL Algorithm)



- A run for which none of G5's action choice probabilities exceed .999 at 100th tick is marked with black.
- In all other runs, the color indicates which of G5's 40 actions has a choice probability $> .999$ at the 100th tick.

AMESModel-III: Experiment 1 (High Initial Propensity)

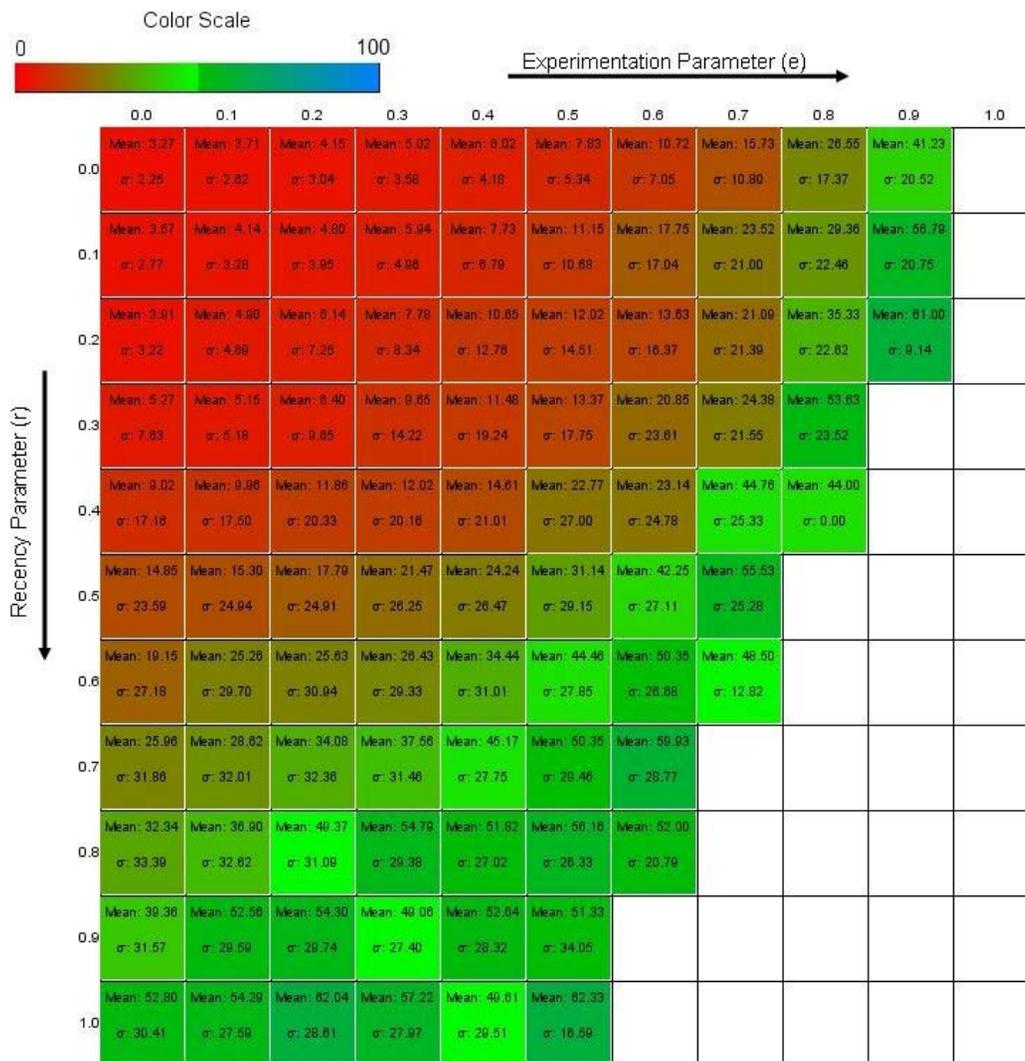
G5's Convergent Action Reaching Probability 0.999 (Roth – Erev RL Algorithm)



- The average time and std. deviation are reported for the Convergent Action of G5 (if any) to reach the threshold probability of 0.999, conditional on given settings $\{r, e\}$ for the recency parameter r and the experimentation parameter e .

AMESModel-III: Experiment 1 (High Initial Propensity)

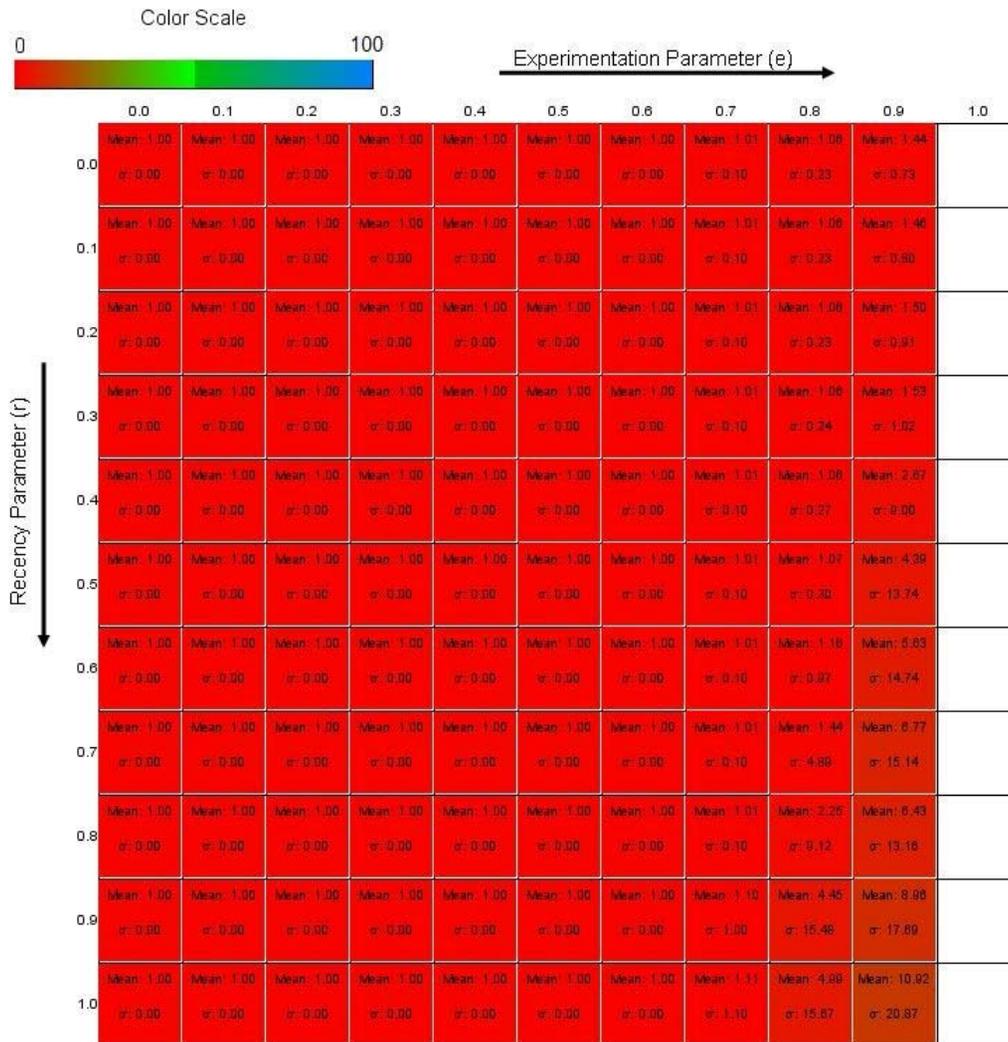
G5's Convergent Action Reaching Probability 0.999 (Variant Roth – Erev RL Algorithm)



- The average time and std. deviation are reported for the Convergent Action of G5 (if any) to reach the threshold probability of 0.999, conditional on given settings $\{r, e\}$ for the recency parameter r and the experimentation parameter e .

AMESModel-III: Experiment 2 (Low Initial Propensity)

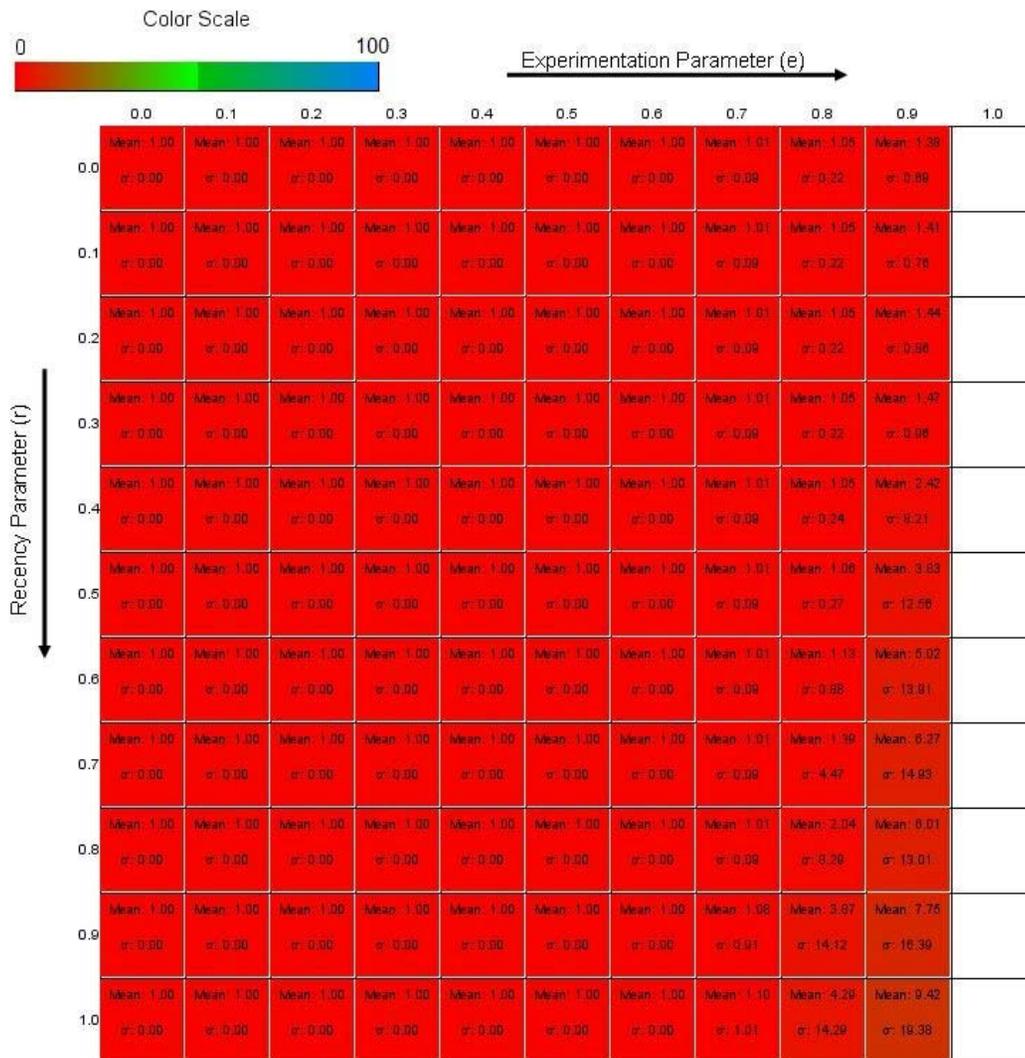
G5's Convergent Action Reaching Probability 0.999 (Roth – Erev RL Algorithm)



- The average time and std. deviation are reported for the Convergent Action of G5 (if any) to reach the threshold probability of 0.999, conditional on given settings $\{r, e\}$ for the recency parameter r and the experimentation parameter e .

AMESModel-III: Experiment 2 (Low Initial Propensity)

G5's Convergent Action Reaching Probability 0.999 (Variant Roth – Erev RL Algorithm)



- The average time and std. deviation are reported for the Convergent Action of G5 (if any) to reach the threshold probability of 0.999, conditional on given settings $\{r, e\}$ for the recency parameter r and the experimentation parameter e .

Conclusions

- This M.S. thesis demonstrates that computational experiments are a powerful tool for studying the behavior of learning algorithms in multi-agent contexts.
- ‘Heat maps’ are used to visualize experimental outcomes resulting from intensive parameter sweeps.
- These heat-map visualizations are used to identify systematic multi-dimensional patterns in the way changes in learning parameter values affect outcomes.
- It is shown how these computationally-determined patterns can point to mathematical theorems.

Future Work

- The cooling parameter has a significant effect on the Variant Roth-Erev algorithm. A mechanism for adapting the cooling parameter may help the Variant Roth-Erev algorithm to learn over a wide range of parameter settings.
- Different reinforcement learning algorithms can be compared in these experiments. One possible reinforcement learning algorithm combines evolutionary learning with reinforcement learning. Q-learning is currently being incorporated into the AMES market framework, and is an ideal candidate for these computational experiments.
- The next release of the AMES framework (v2.0) permits hourly demands to be price sensitive as well as fixed, rather than all fixed hourly demands as present in the current version (v1.31). These experiments can be continued in this more sophisticated environment.

Questions?