WIND AND ENERGY STORAGE

Optimal Control of Power Systems in Context of Wind Energy Generation and Storage

Shuyang Li

Overview

- **Motivation**
- Model
- Algorithms
- Methodology
- Data
- Results
- Further Exploration

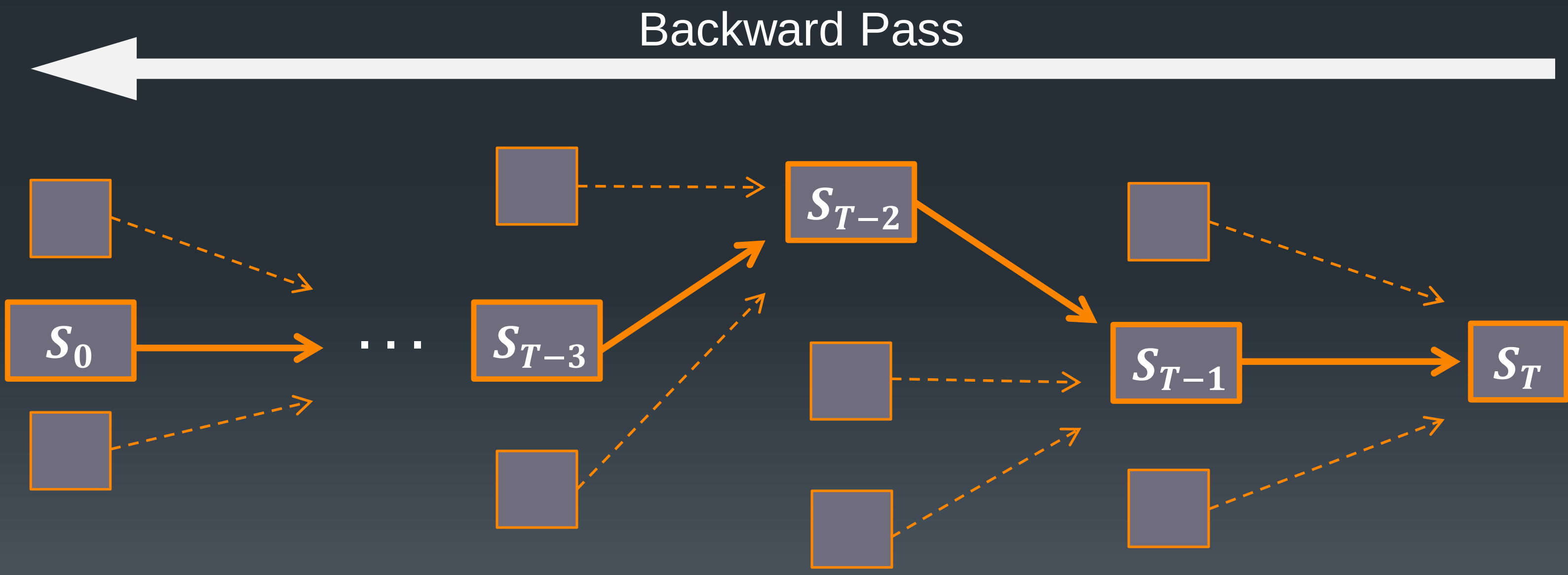
MOTIVATION

- Increasing reliance on wind energy
- Wind Energy Variance
  - Avoid waste
- Storing Wind
  - Storage/Unit Allocation

MOTIVATION – Dynamic Programming

- Solving Bellman's Equation:

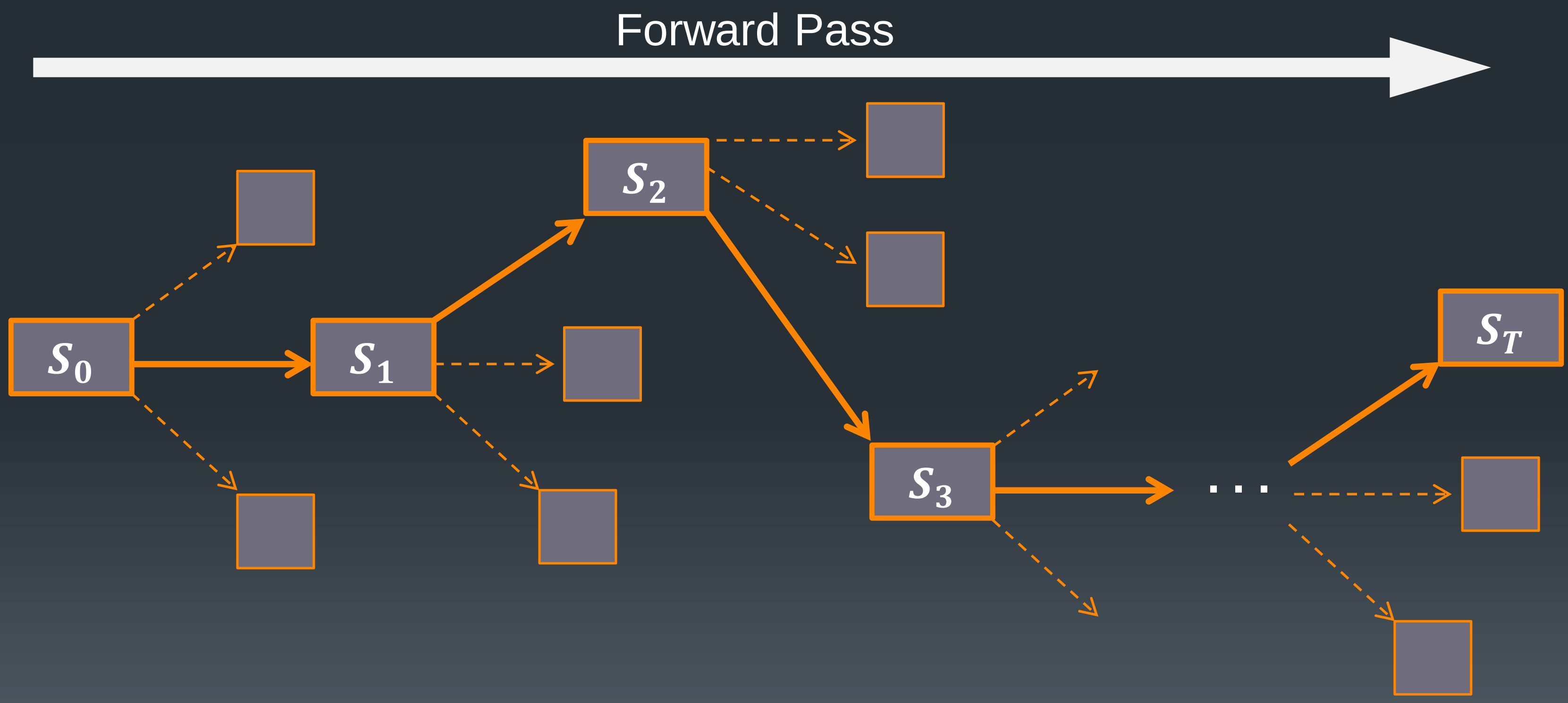
$$V(S^n) = \max_a (C(S^n, a) + \gamma \mathbb{E}\{V(S^{n+1}) | S^n\})$$



From *Optimal Learning* (Powell & Ryzhov 2012)

MOTIVATION – ADP/RL

- *Monte Carlo Simulation*



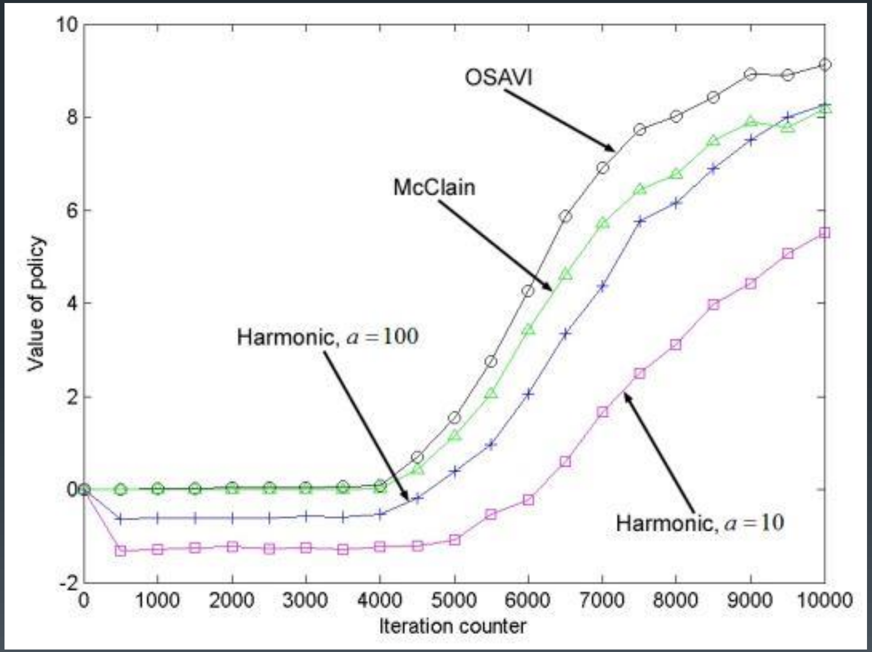
MOTIVATION - Literature

- *"A Comparison of Approximate Dynamic Programming Techniques on Benchmark Energy Storage Problems: Does Anything Work?"*
  - Jiang, Pham & Powell, 2014

- *"Benchmarking a Scalable Approximate Dynamic Programming Algorithm for Stochastic Control of Multidimensional Energy Storage Problems"*
  - Salas & Powell, 2014

PROBLEM

- Algorithmic Performance
  - Q-Learning
  - SARSA(λ)
  - Step Sizes

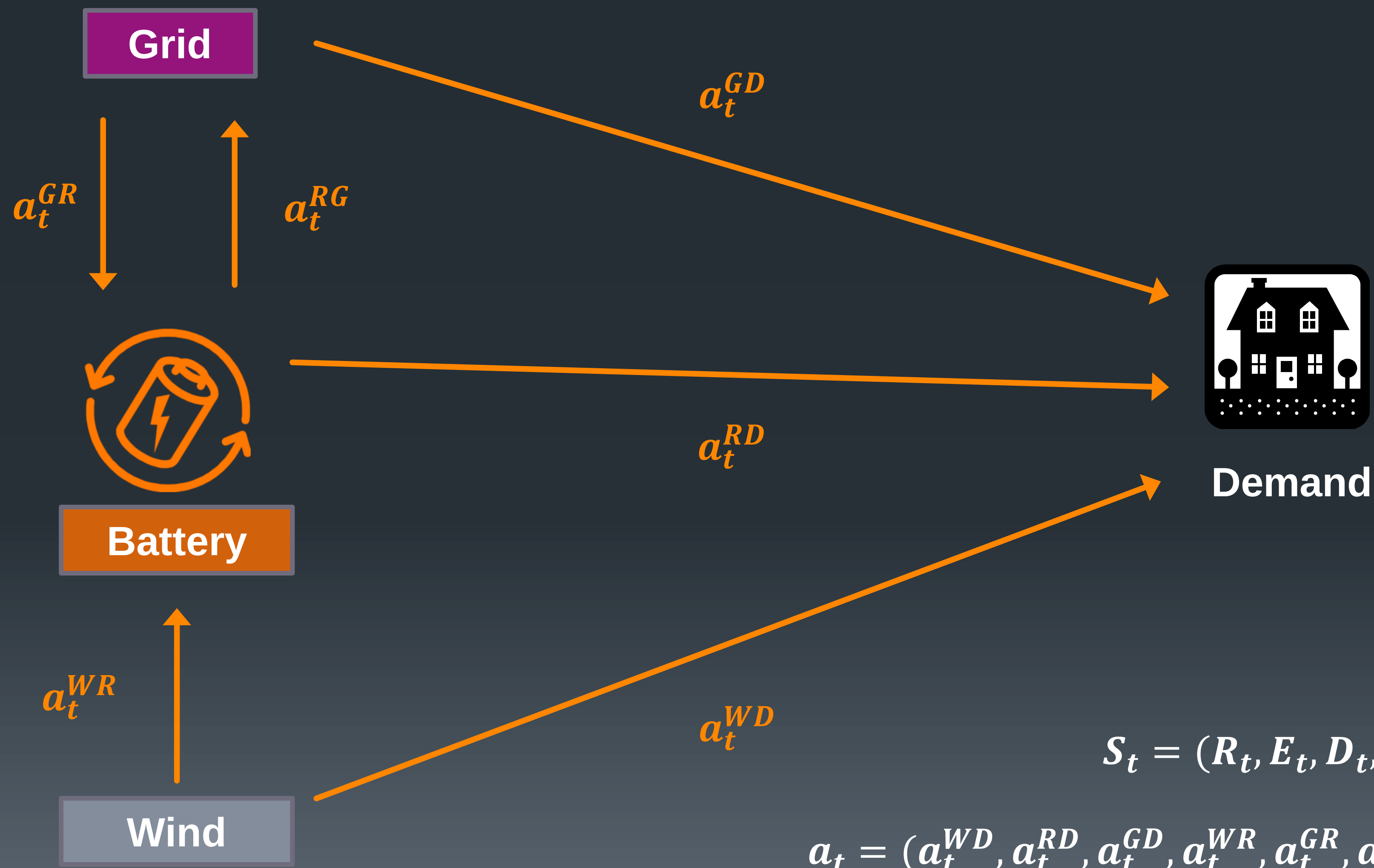


Ryzhov, Frazier & Powell (2014)

Overview

- Motivation
- **Model**
- Algorithms
- Methodology
- Data
- Results
- Further Exploration

MODEL



$$S_t = (R_t, E_t, D_t, P_t)$$

$$a_t = (a_t^{WD}, a_t^{RD}, a_t^{GD}, a_t^{WR}, a_t^{GR}, a_t^{RG})$$

MODEL – Action Constraints

Storage capacity:

- $a_t^{WR} + a_t^{GR} \leq R^{max} - R_t$
- $a_t^{RD} + a_t^{RG} \leq R_t$

Charging / Discharging

- $a_t^{WR} + a_t^{GR} \leq \gamma^c$
- $a_t^{RD} + a_t^{RG} \leq \gamma^d$

Demand satisfied:

- $a_t^{WD} + \eta^d a_t^{RD} + a_t^{GD} = D_t$

Flow Conservation

- $a_t^{WR} + a_t^{WD} \leq E_t$

Maximal Wind Usage:

- $a_t^{WD} = \min(D_t, E_t)$
- $a_t^{WR} = \min(R^{max} - R_t, E_t - a_t^{WD})$

From *Benchmarking a Scalable Approximate Dynamic Programming Algorithm for Stochastic Control of Multidimensional Energy Storage Problems* (Salas, Powell) [*Except Maximal Wind Usage*]

MODEL – Transition Functions

Storage:

$$R_{t+1} = R_t + \phi^T a_t \; ; \; \phi = (0, -1, 0, \eta^c, \eta^c, -1)$$

Simplified in the stochastic model:

$$R_{t+1} = R_t - a_t^{RD} + a_t^{WR} + a_t^{GR} - a_t^{RG}$$

From *Benchmarking a Scalable Approximate Dynamic Programming Algorithm for Stochastic Control of Multidimensional Energy Storage Problems* (Salas, Powell)

MODEL – Transition Functions

Wind:

- $E_{t+1} = E_t + \hat{E}_{t+1}$
- $\hat{E}_t \sim \mathcal{N}(\mu_E, \sigma_E^2)$

Price:

- $P_{t+1} = P_t + \hat{P}_{0,t+1} + 1_{\{u_{t+1} \leq 0.031\}} \hat{P}_{1,t+1}$
- $\hat{P}_{0,t} \sim \mathcal{N}(\mu_P, \sigma_P^2)$
- $\hat{P}_{1,t} \sim \mathcal{N}(0, 50^2)$
- $u_t \sim \mathcal{U}(0,1)$

Demand:

- $D_t = \left\lfloor \max\left[0, 3 - 4\sin\left(\frac{2\pi t}{T}\right)\right] \right\rfloor$

From *Benchmarking a Scalable Approximate Dynamic Programming Algorithm for Stochastic Control of Multidimensional Energy Storage Problems* (Salas, Powell)

MODEL – Objective Function

Reward Function:

$$C(S_t, a_t) = P_t D_t - P_t\left(a_t^{GR} - \eta^d a_t^{RG} + a_t^{GD}\right) - c^h R_{t+1}$$

Objective Function:

$$F^{\pi^*} = \max_{\pi \in \Pi} \mathbb{E}\left[\sum_{t \in T} C(S_t, A_t^{\pi}(S_t))\right]$$

From *Benchmarking a Scalable Approximate Dynamic Programming Algorithm for Stochastic Control of Multidimensional Energy Storage Problems* (Salas, Powell)

Overview

- Motivation
- Model
- **Algorithms**
- Methodology
- Data
- Results
- Further Exploration

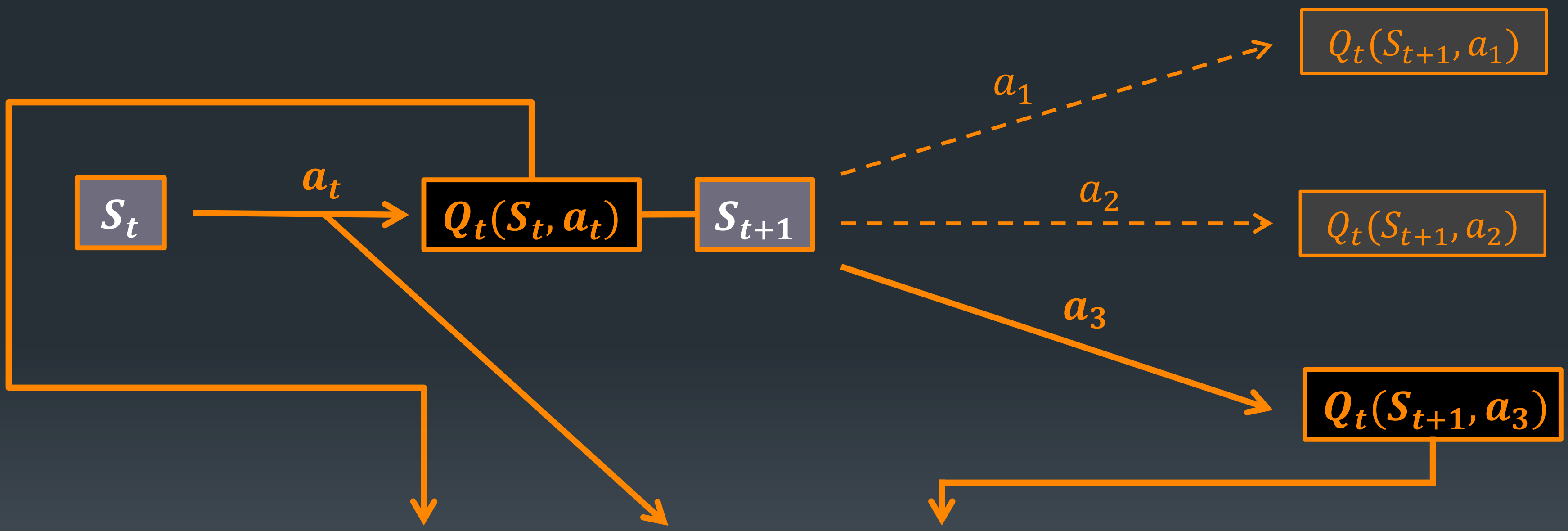
ALGORITHMS – Q-Learning

- Action-value function Q

$$Q_{t+1}(S_t, a_t) = Q_t(S_t, a_t) + \alpha[R_{t+1} + \gamma \max_a Q_t(S_{t+1}, a) - Q_t(S_t, a_t)]$$

- *Off-policy*
  - Selection policy: ε-greedy
  - Evaluation policy: pure greedy

ALGORITHMS – Q-Learning



$$Q_{t+1}(S_t, a_t) = Q_t(S_t, a_t) + \alpha[R_{t+1} + \gamma \max_a Q_t(S_{t+1}, a) - Q_t(S_t, a_t)]$$

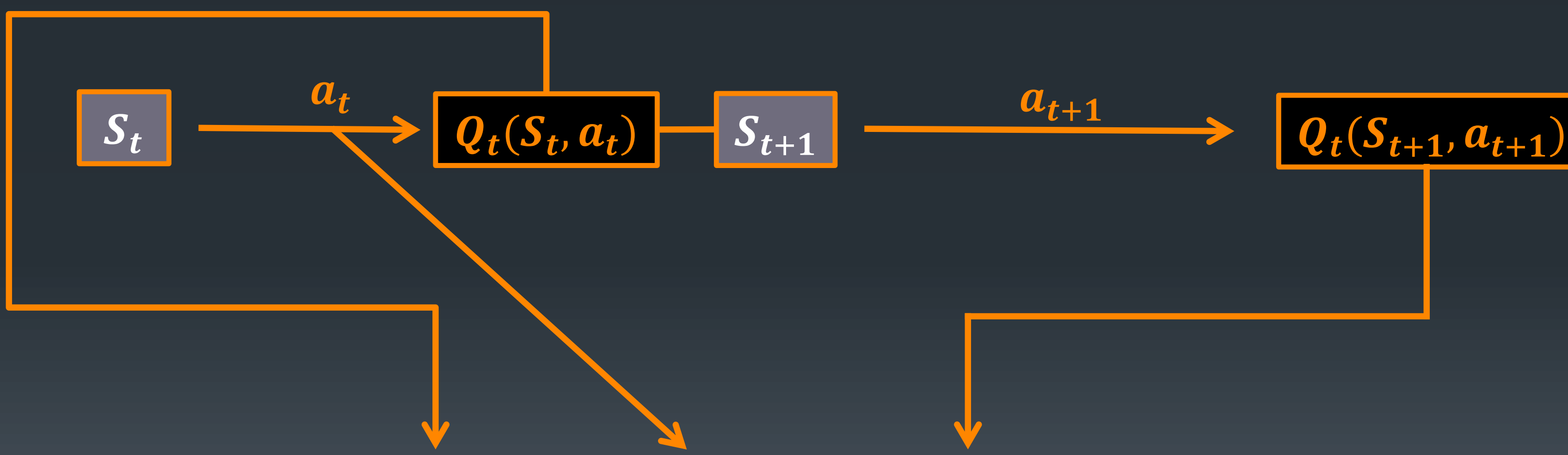
ALGORITHMS - SARSA

- SARSA $[\boldsymbol{S}_t\ \boldsymbol{a}_t\ \boldsymbol{R}_{t+1}\ \boldsymbol{S}_{t+1}\ \boldsymbol{a}_{t+1}]$

$$Q_{t+1}(S_t, a_t) = Q_t(S_t, a_t) + \alpha[R_{t+1} + \gamma \boldsymbol{Q}_t(\boldsymbol{S}_{t+1}, \boldsymbol{a}_{t+1}) - Q_t(S_t, a_t)]$$

- *On-policy*
  - Evaluation policy **is** selection policy

ALGORITHMS - SARSA



$$Q_{t+1}(S_t, a_t) = Q_t(S_t, a_t) + \alpha[R_{t+1} + \gamma Q_t(S_{t+1}, a_{t+1}) - Q_t(S_t, a_t)]$$

ALGORITHMS – SARSA(λ)

- SARSA(λ) includes a backward pass along a path (eligibility trace)

$$Q_{t+1}(S, a) = Q_t(S, a) + \alpha \delta_t Z_t(s, a)$$

$$\delta_t = R_{t+1} + \gamma Q_t(S_{t+1}, a_{t+1}) - Q_t(S_t, a_t)$$

$$Z_t = \begin{cases} \gamma \lambda Z_{t-1} + 1, & S = S_t, a = a_t \\ \gamma \lambda Z_{t-1}, & otherwise \end{cases}$$

ALGORITHMS – SARSA(λ)

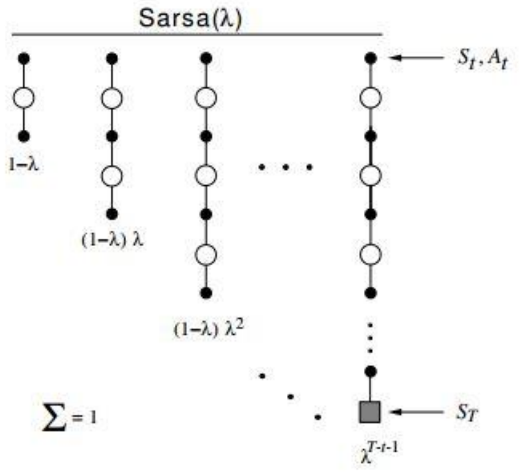


Figure 7.10: Sarsa(λ)'s backup diagram.

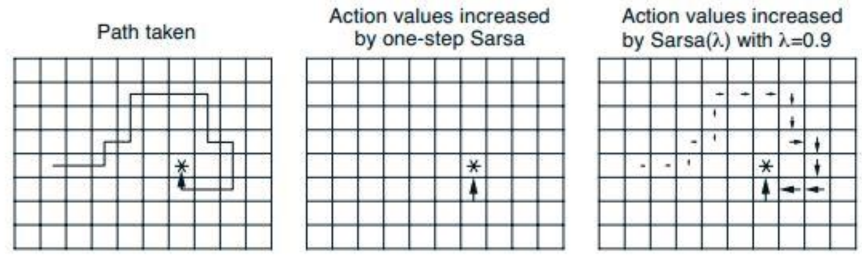


Figure 7.12: Gridworld example of the speedup of policy learning due to the use of eligibility traces.

Reinforcement Learning: An Introduction 2nd Ed. (Richard Sutton & Andrew Barto; 2014)

ALGORITHMS – Step Sizes

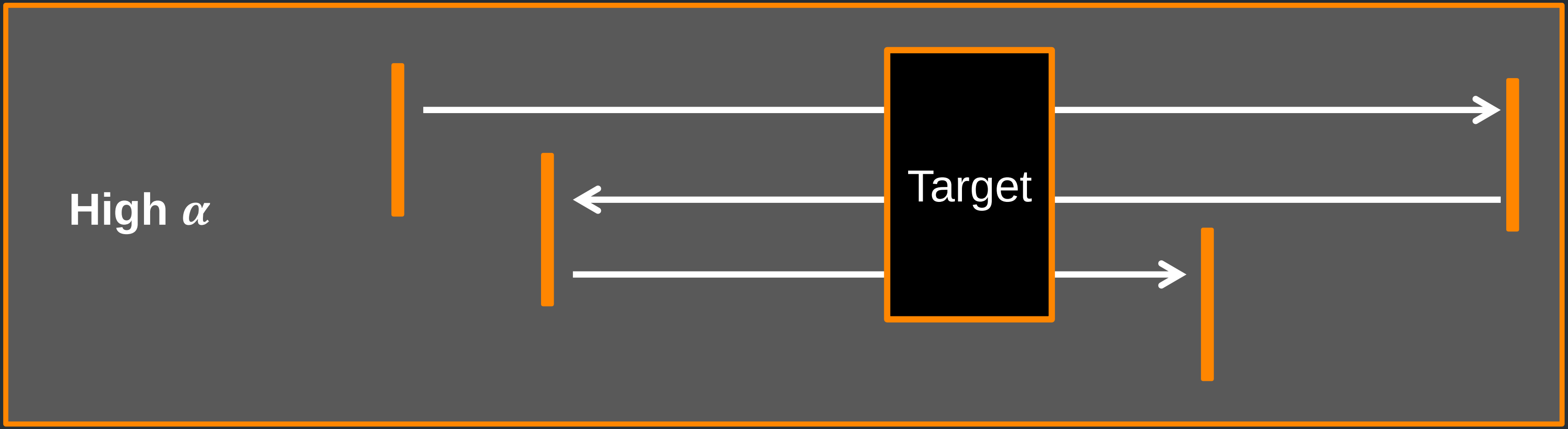
- Updating equation:

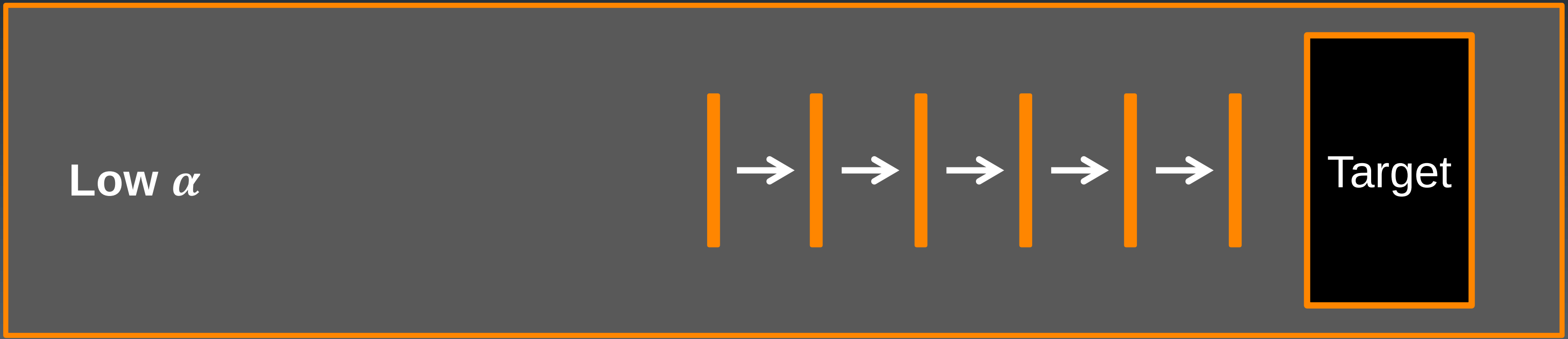
$$Q_{t+1}(S_t, a_t) = Q_t(S_t, a_t) + \alpha[R_{t+1} + \gamma \max_a Q_t(S_{t+1}, a) - Q_t(S_t, a_t)]$$

- Rearranged:

$$Q_{t+1}(S_t, a_t) = (1 - \alpha)Q_t(S_t, a_t) + \alpha[R_{t+1} + \gamma \max_a Q_t(S_{t+1}, a)]$$

ALGORITHMS – Step Sizes





ALGORITHMS – Step Sizes

- Constant : $\alpha = k$
- Harmonic : $\alpha = \frac{a}{a+n}$
- 1/n : $\alpha = \frac{1}{n}$

- Ryzhov Formula
  - Ryzhov, Frazier & Powell (2014)

Overview

- Motivation
- Model
- Algorithms
- **Methodology**
- Data
- Results
- Further Exploration

METHODOLOGY - Simulation

- 1 Deterministic Problem
- 17 Stochastic Problems

- 256 Sample Paths per stochastic problem

- Training iterations: transitions by Monte Carlo simulation
  - ε-greedy action selection

- Evaluation: Averaged over all sample paths

Overview

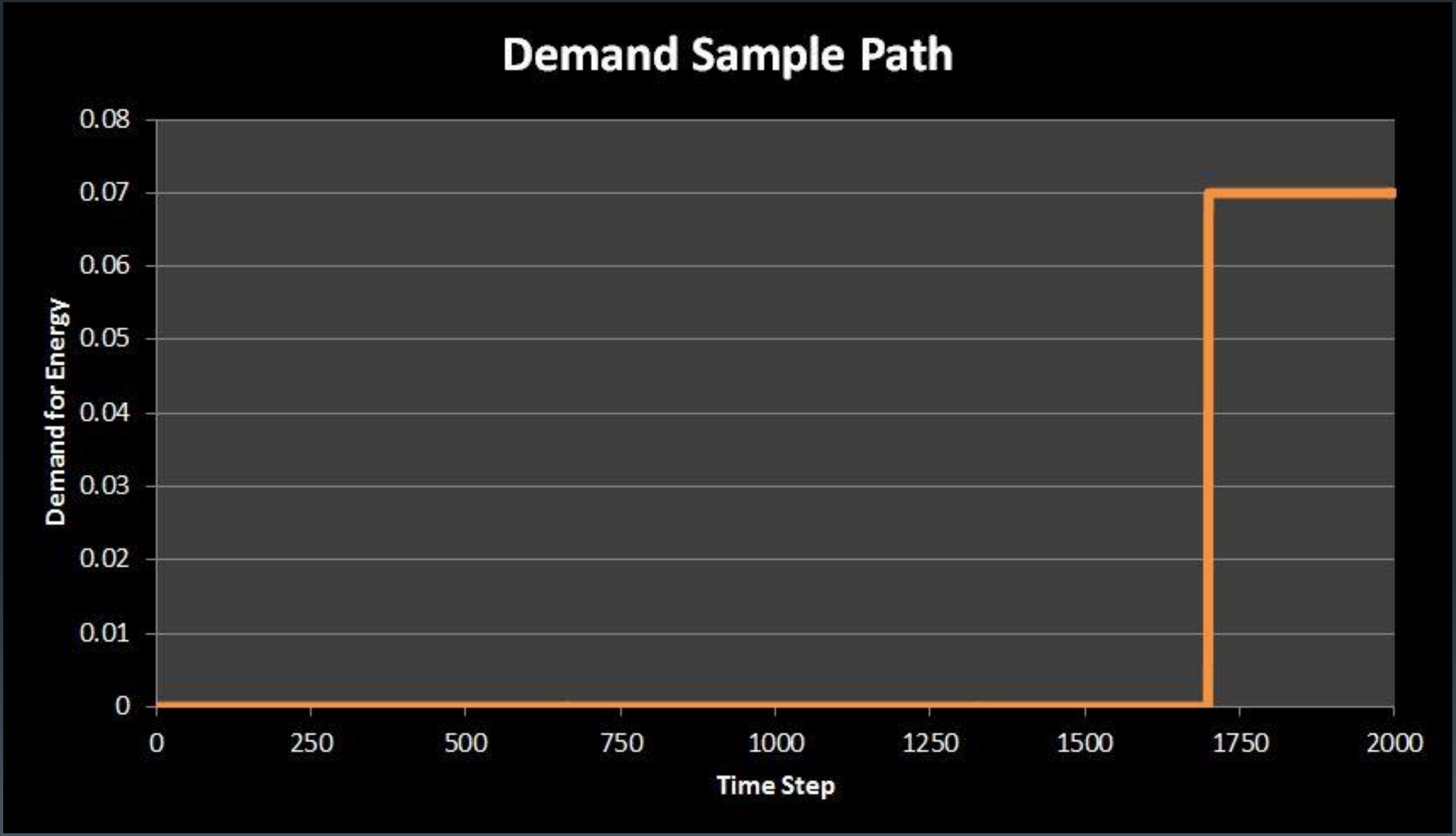
- Motivation
- Model
- Algorithms
- Methodology
- **Data**
- Results
- Further Exploration

DATA – Deterministic Parameters

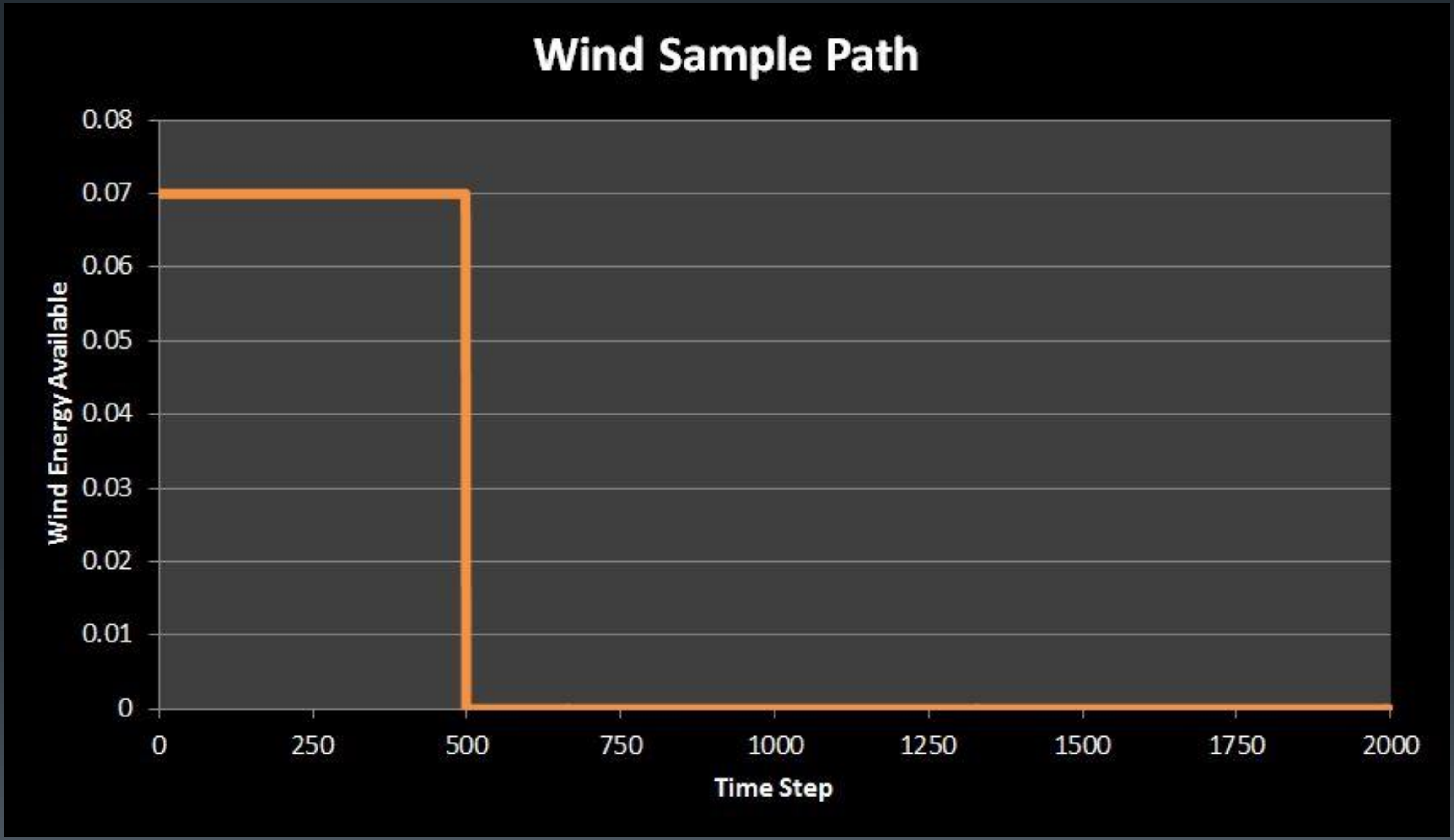
- $R^{max} = 100$
- $R^{min} = 0$
- $R_0 = 0$
- $\eta^c = \eta^d = 0.90$
- $\gamma^c = \gamma^d = 0.10$

- $\boldsymbol{T = 2000}$

DATA – Deterministic Problems



DATA – Deterministic Problems



DATA – Deterministic Problems

DATA – Stochastic Parameters

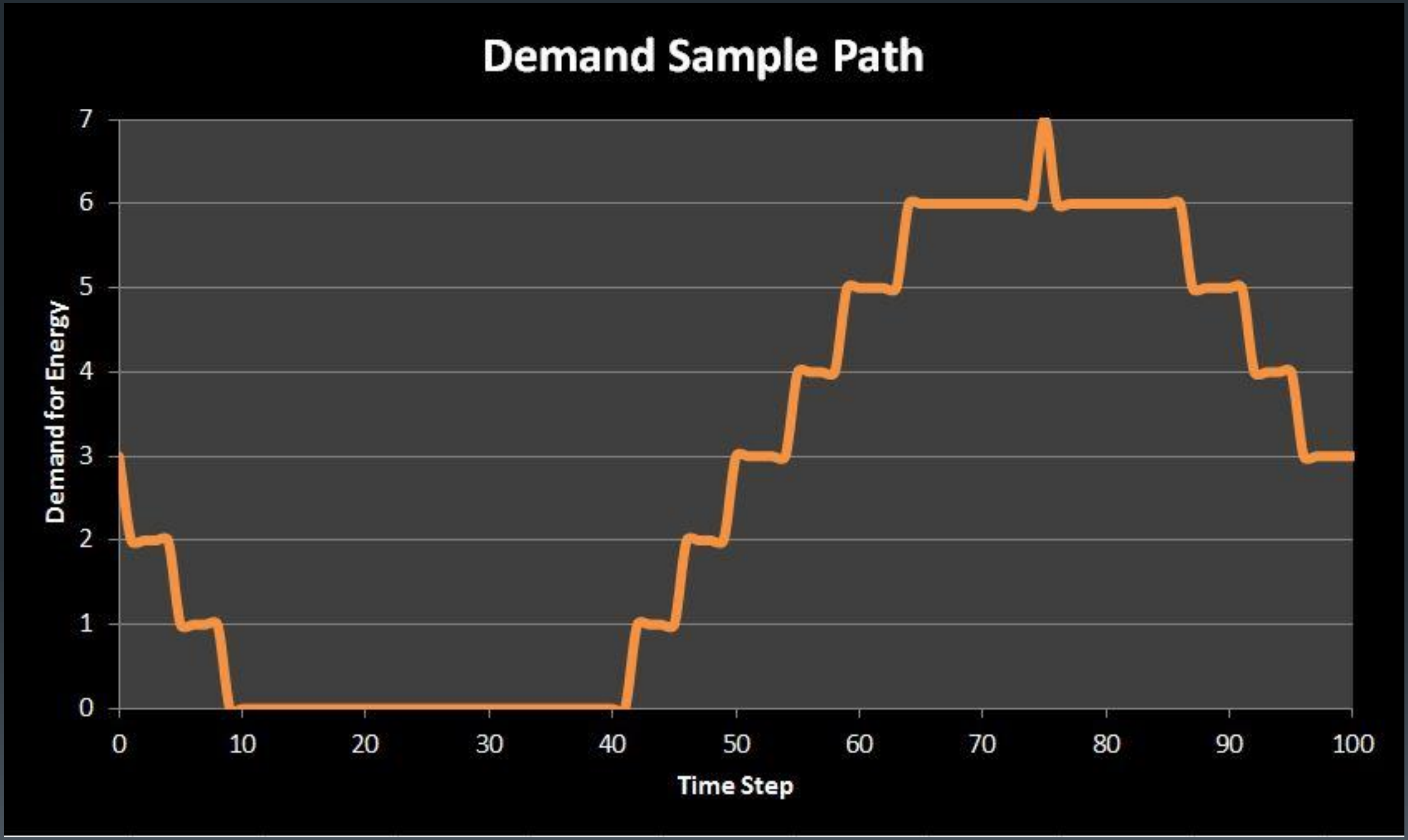
- $R^{max} = 30$
- $R^{min} = 0$
- $R_0 = 25$
- $\eta^c = \eta^d = 1.00$
- $\gamma^c = \gamma^d = 5$

- $\boldsymbol{T = 100}$

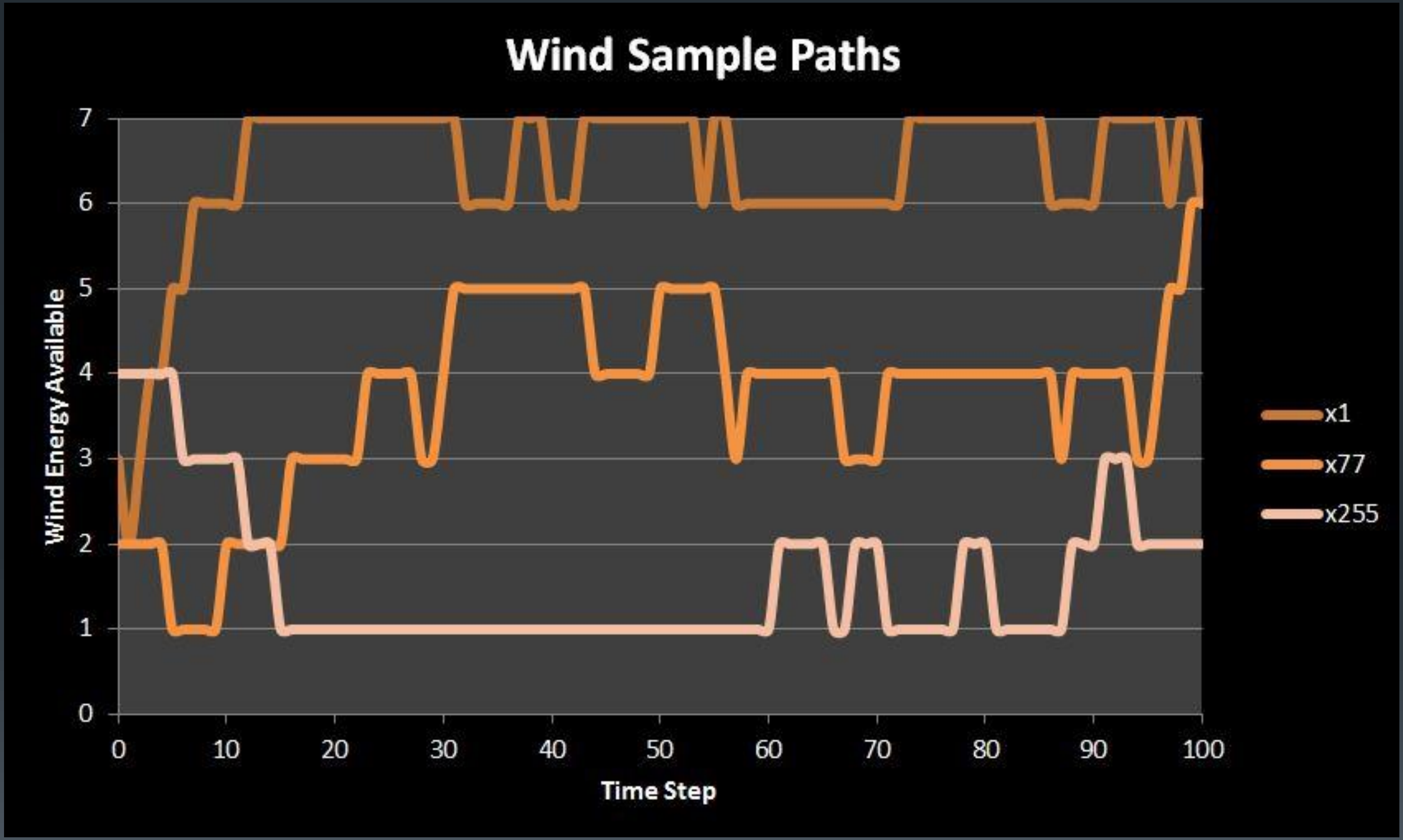
- $P^{max} = 70$
- $P^{min} = 30$

- $E^{max} = 7.00$
- $E^{min} = 1.00$

DATA – Stochastic Problems



DATA – Stochastic Problems



DATA – Stochastic Problems

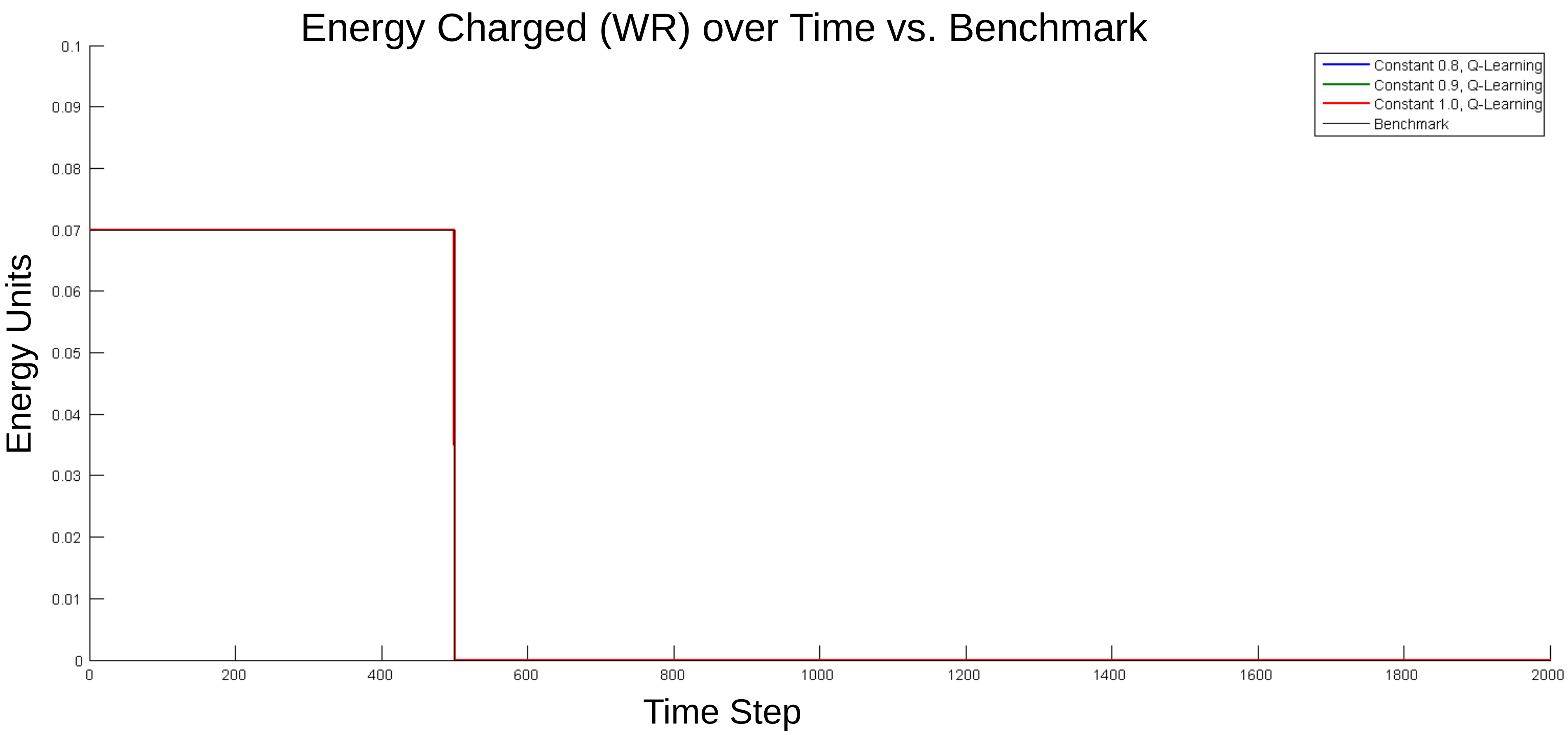
Overview

- Motivation
- Model
- Algorithms
- Methodology
- Data
- **Results**
- Further Exploration

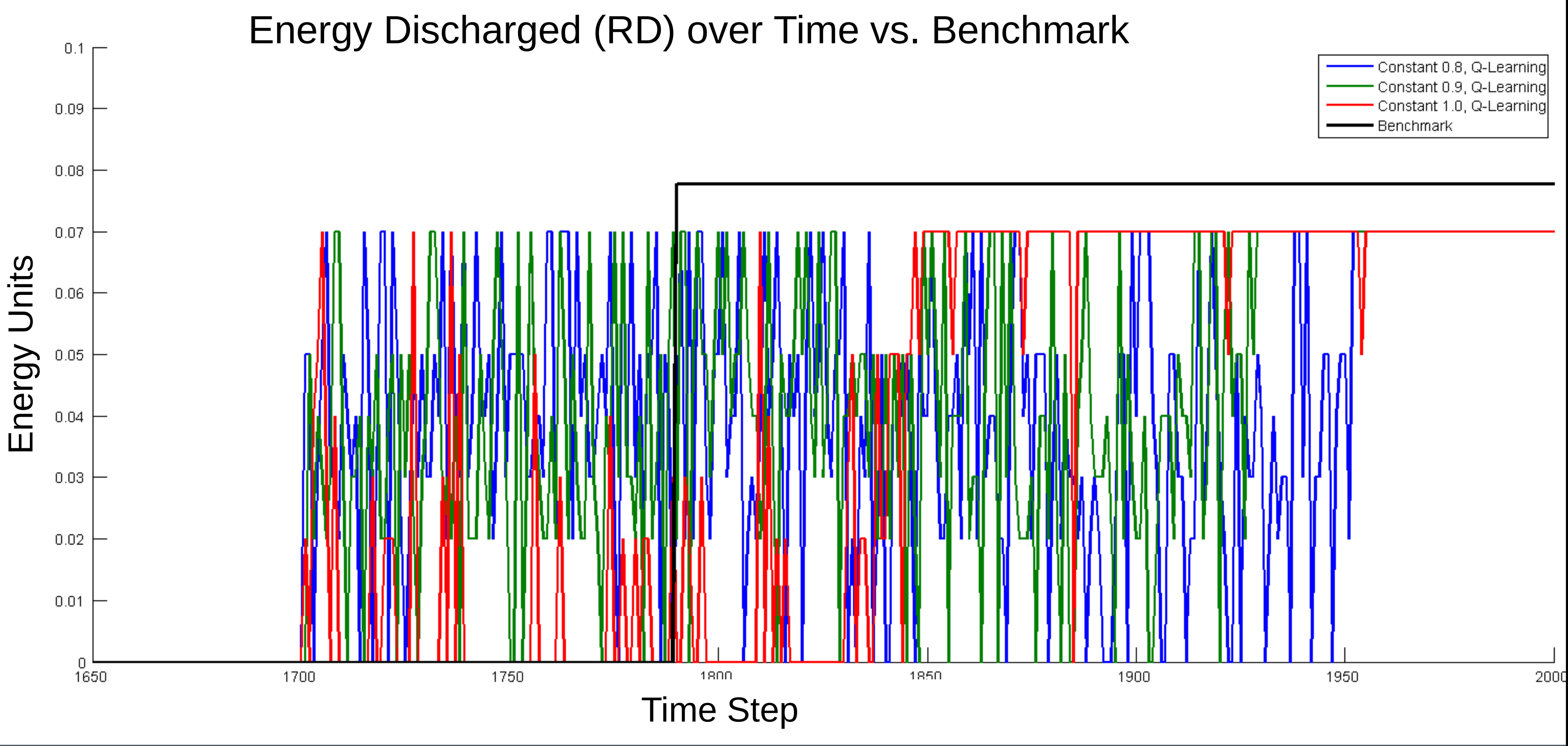
RESULTS – METRICS

- Action Traces (Deterministic)
- Step Size over Updates
- Stored Energy over Time vs. Benchmark
- Performance over # of Training Iterations

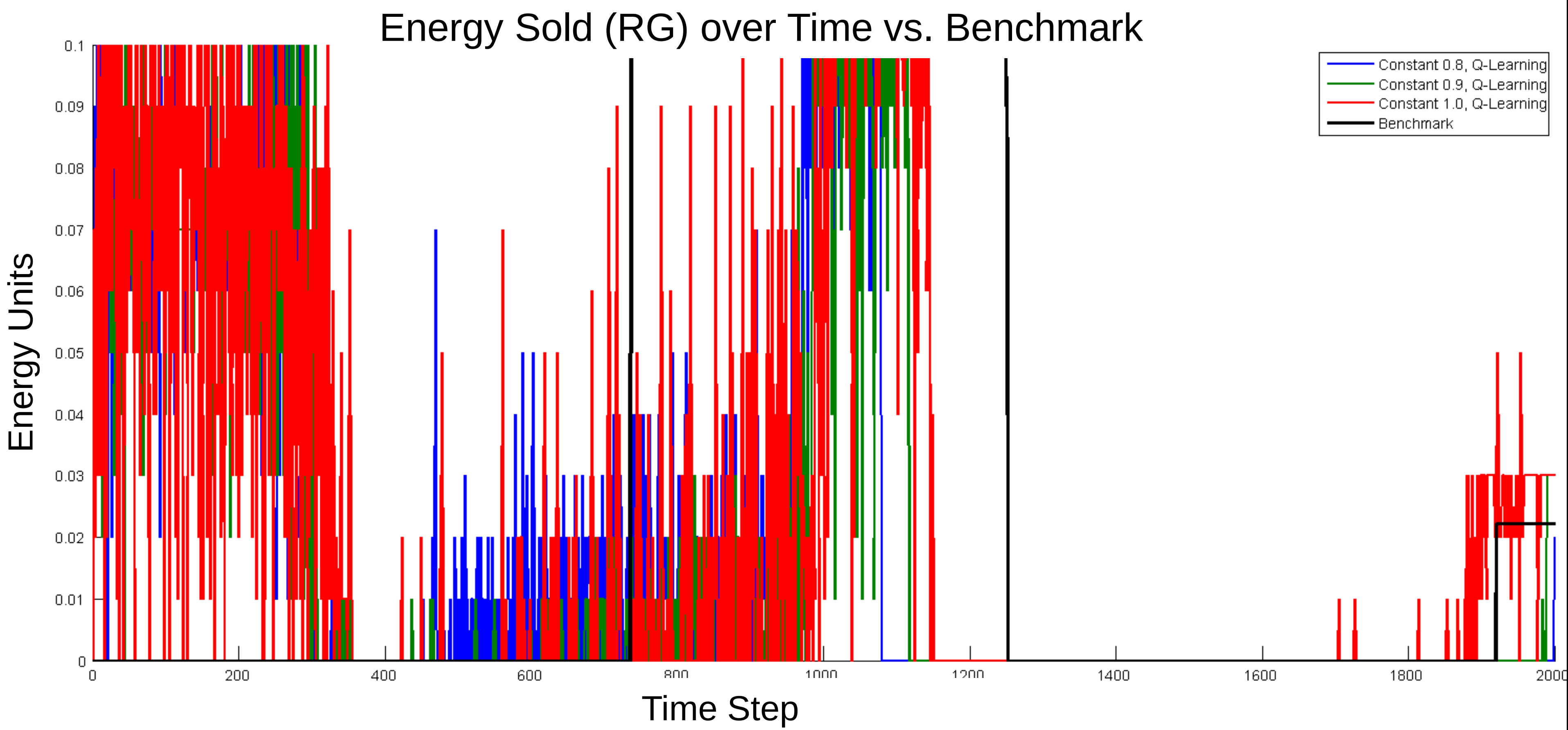
DETERMINISTIC ACTIONS



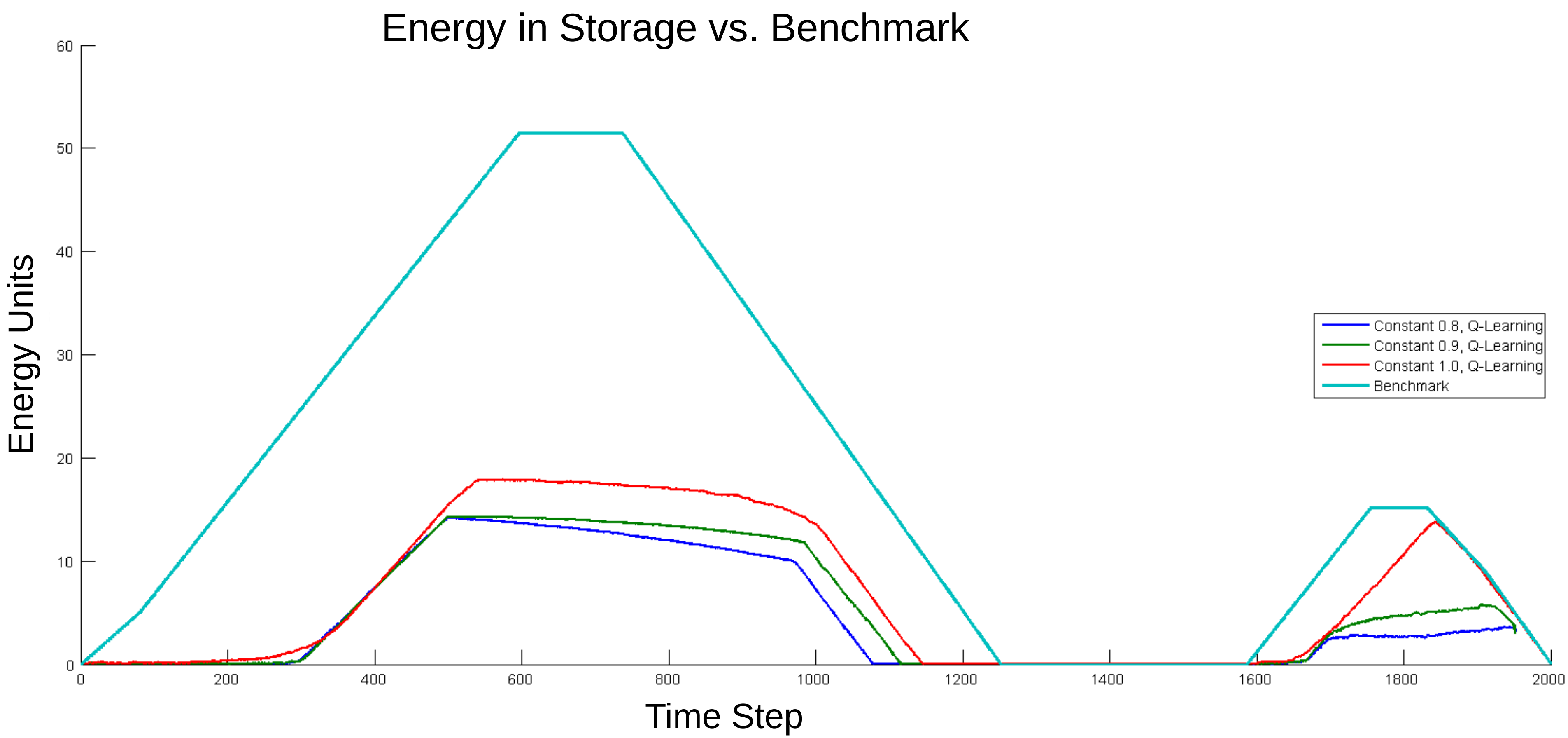
DETERMINISTIC ACTIONS



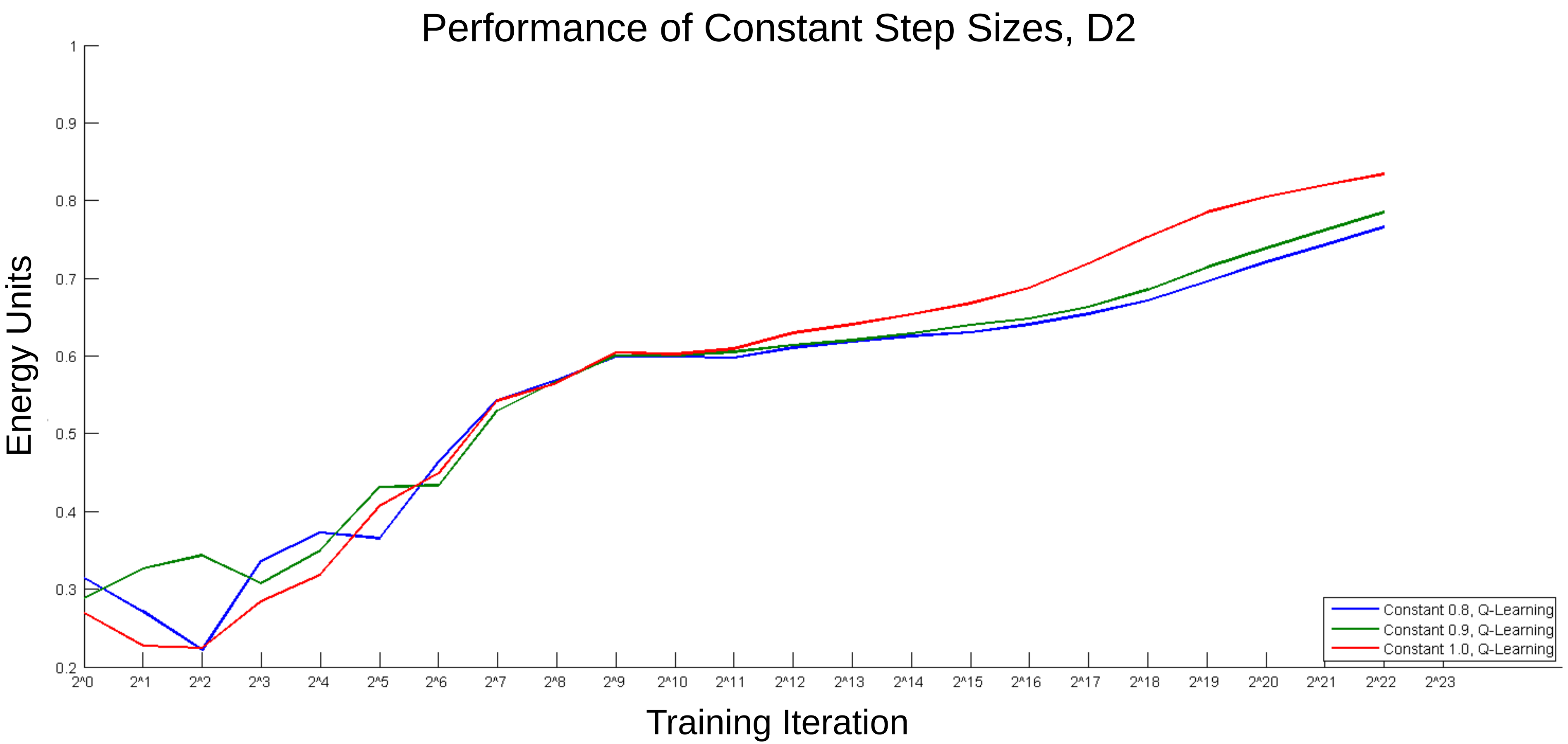
DETERMINISTIC ACTIONS



DETERMINISTIC STORAGE



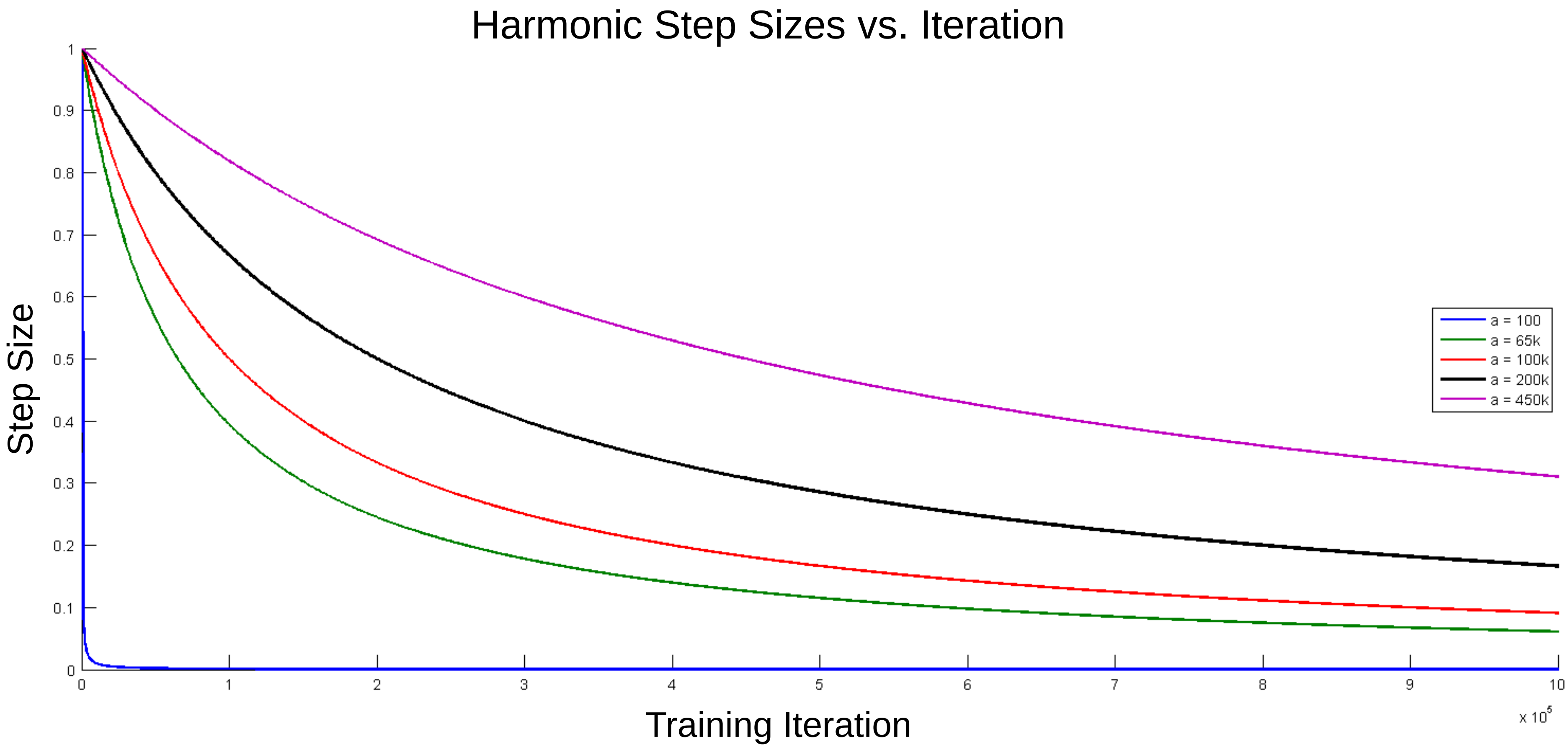
DETERMINISTIC PERFORMANCE



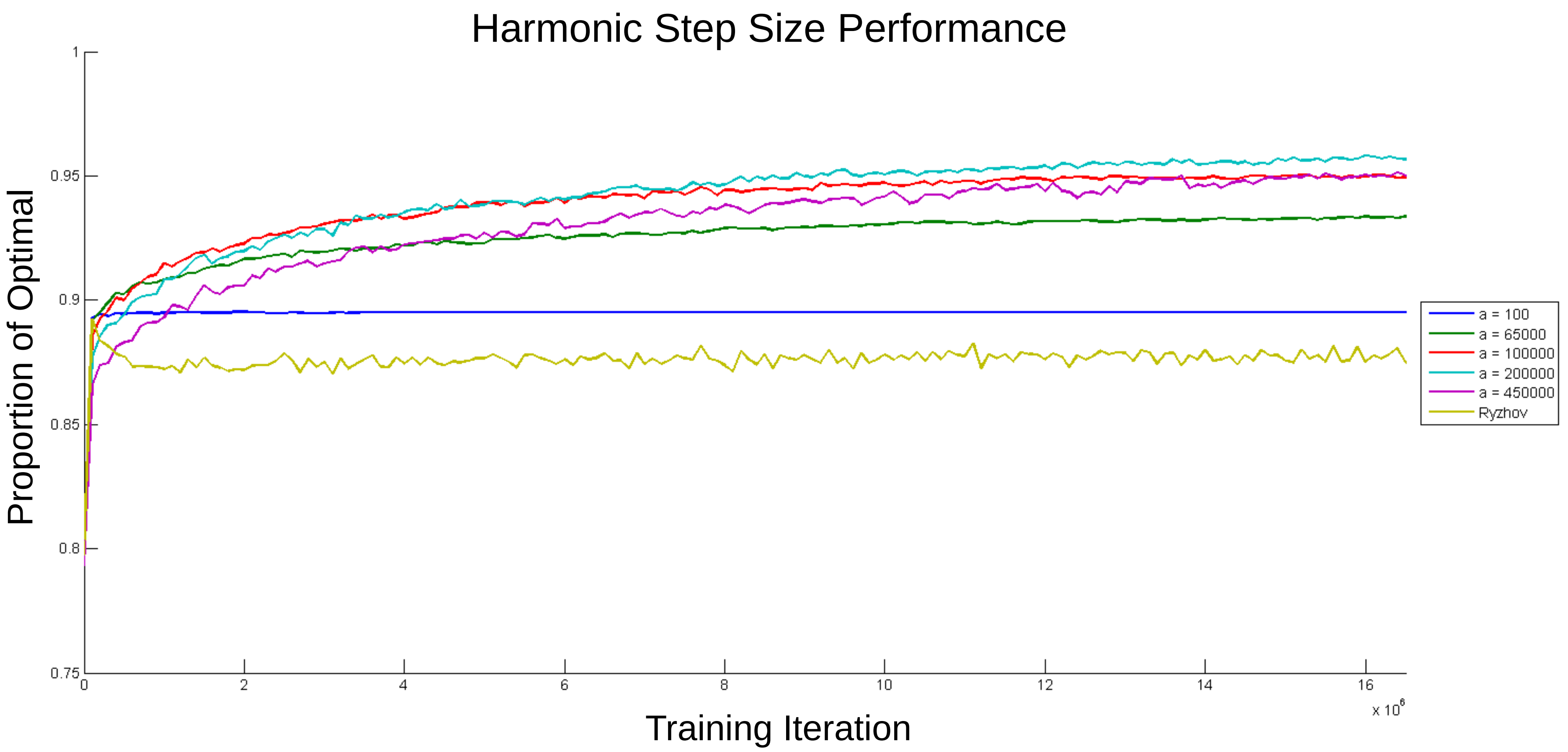
STEP SIZE TUNING

- Declining Step Sizes
- Tunable parameters
  - Harmonic: $\frac{a}{a+n}$
  - Ryzhov: Estimator update factor ν

HARMONIC STEP SIZES



HARMONIC PERFORMANCE



RYZHOV STEP SIZE

- No-discount formula:

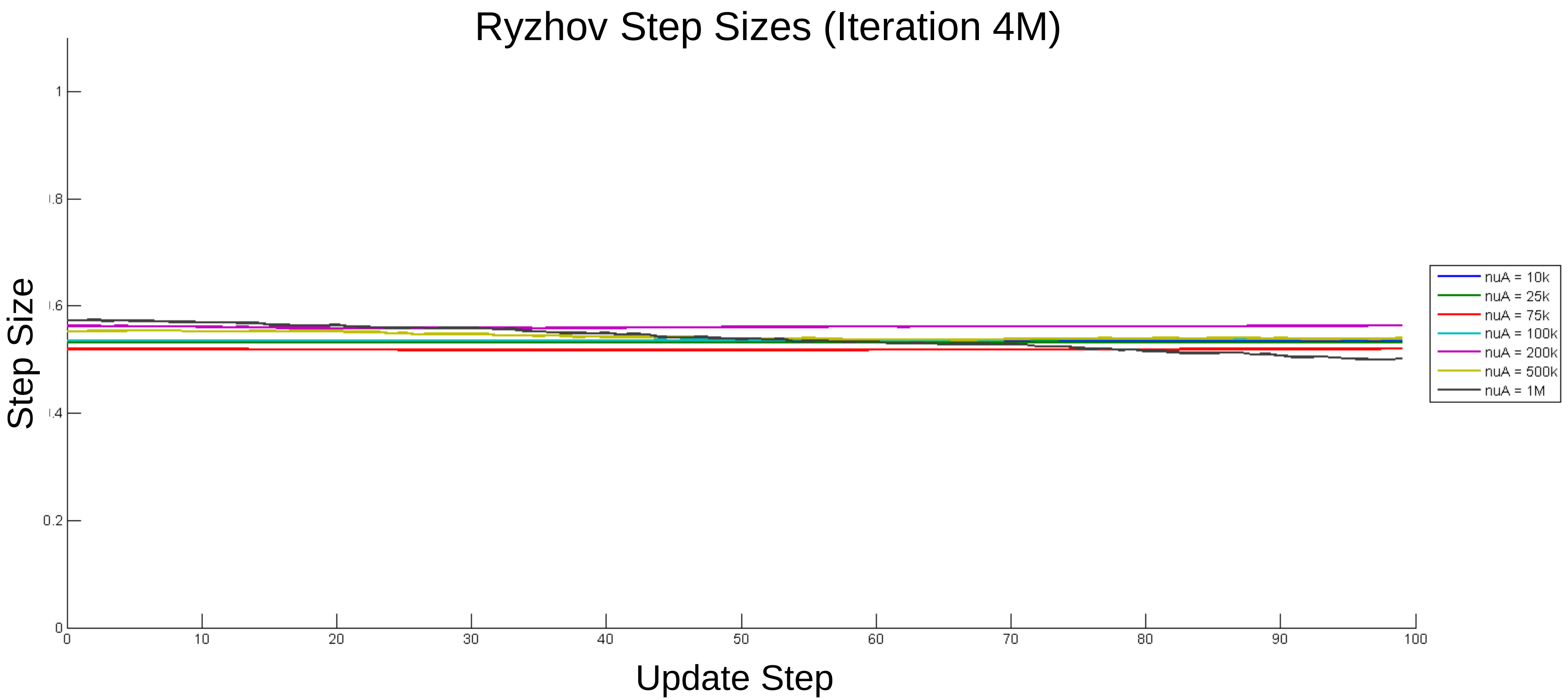
$$\alpha_{n-1} = \frac{(\bar{c}^n)^2}{(\bar{c}^n)^2 + (\bar{\sigma}^n)^2}$$

- Tunable parameter:

$$\bar{c}^n = (1 - \boldsymbol{\nu_{n-1}})\bar{c}^{n-1} + \boldsymbol{\nu_{n-1}}\hat{c}^n$$

$$(\bar{\sigma}^n)^2 = (1 - \boldsymbol{\nu_{n-1}})(\bar{\sigma}^{n-1})^2 + \boldsymbol{\nu_{n-1}}(\hat{c}^n - \bar{c}^{n-1})^2$$

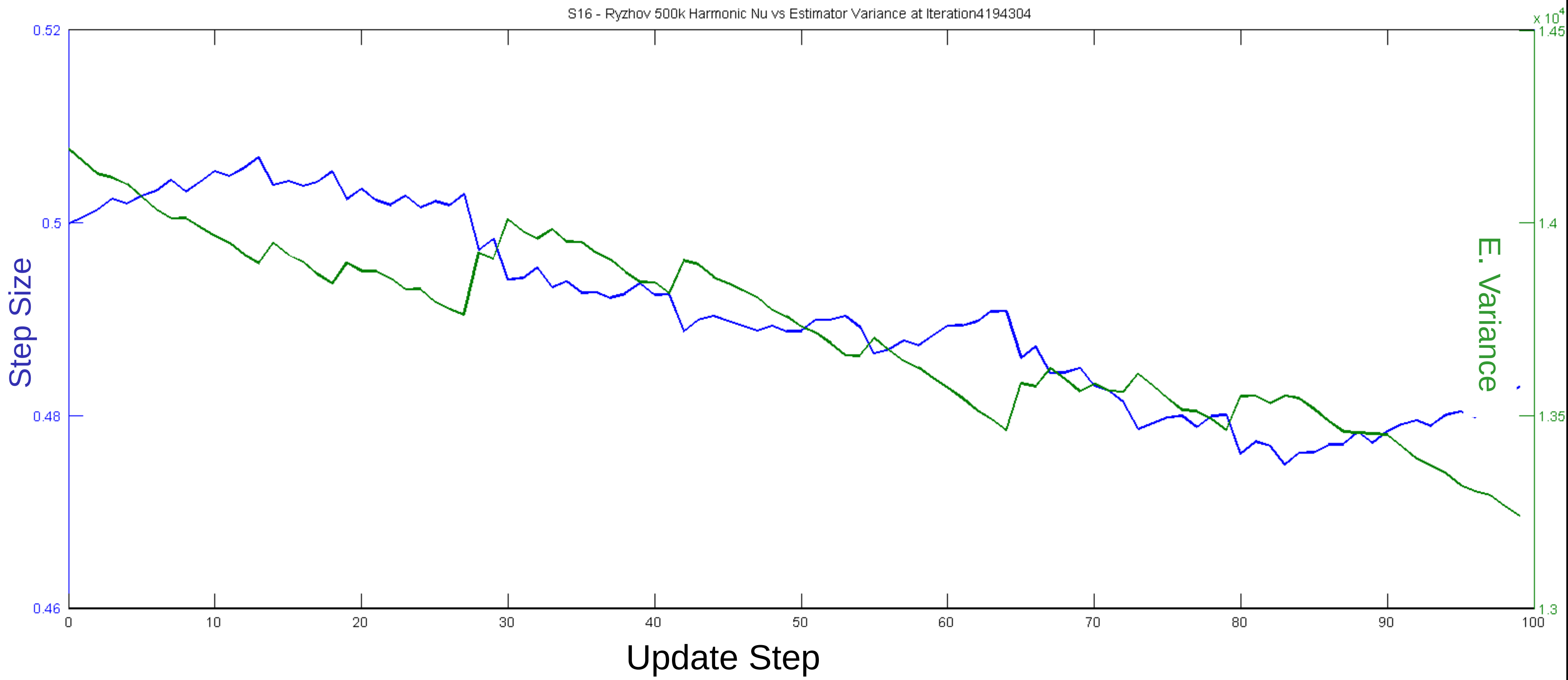
RYZHOV STEP SIZES



Ryzhov explanation

- It should even out at a certain constant depending on the variance in the rewards
- For this problem, the rewards are deterministic, so the variance measured is the variance in the different states that we reach/are in

RYZHOV STEP SIZES vs. ESTIMATOR VARIANCE



RYZHOV PERFORMANCE

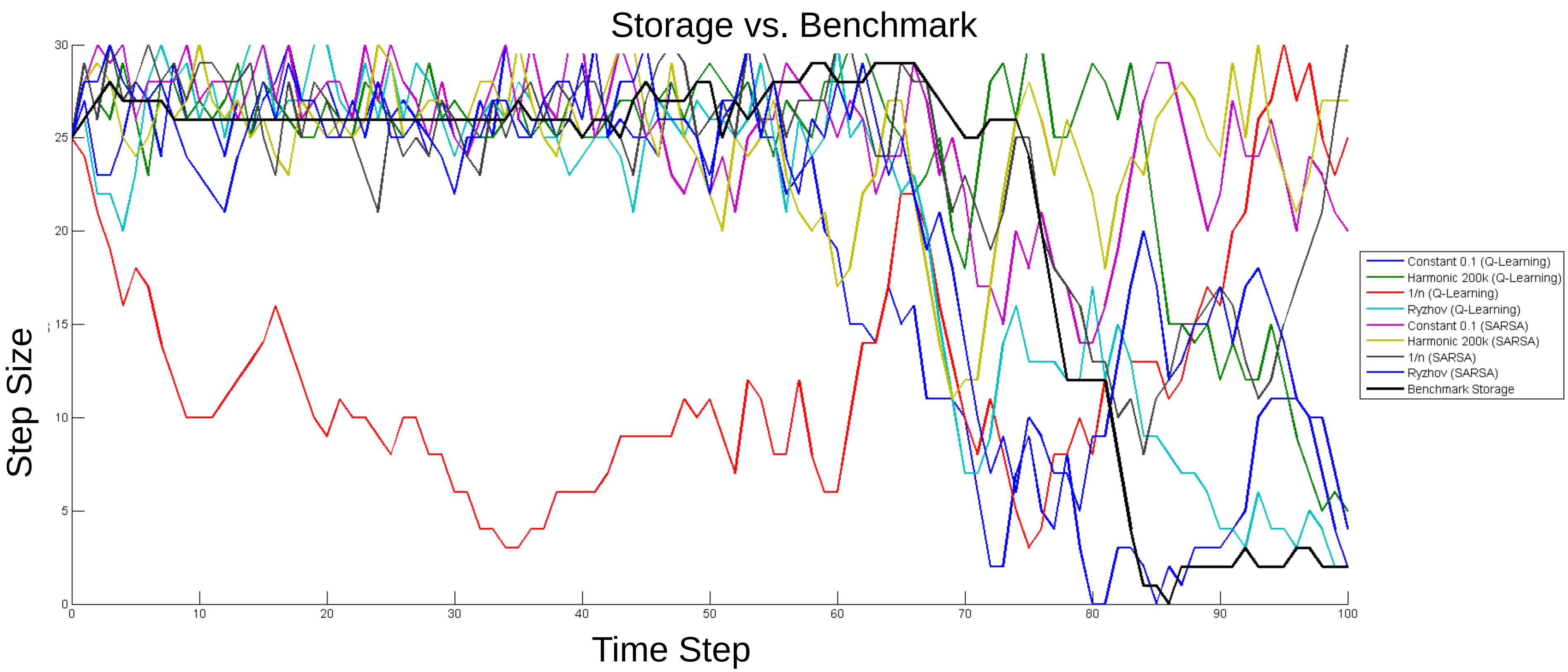
AGGREGATE PARAMETERS

	Resource, R_t		Wind, E_t			Price, P_t		
Label	Levels	ΔR	Levels	ΔE	$\hat{E}_t$	Levels	Process	$\hat{P}_{0,t}$
S1	61	0.50	13	0.50	$\mathcal{U}(-1,1)$	7	Sinusoidal	$\mathcal{N}(0,25^2)$
S2	61	0.50	13	0.50	$\mathcal{N}(0,0.5^2)$	7	Sinusoidal	$\mathcal{N}(0,25^2)$
S3	61	0.50	13	0.50	$\mathcal{N}(0,1.0^2)$	7	Sinusoidal	$\mathcal{N}(0,25^2)$
S4	61	0.50	13	0.50	$\mathcal{N}(0,1.5^2)$	7	Sinusoidal	$\mathcal{N}(0,25^2)$
S5	31	1.00	7	1.00	$\mathcal{U}(-1,1)$	41	1st-order + jump	$\mathcal{N}(0,0.5^2)$
S6	31	1.00	7	1.00	$\mathcal{U}(-1,1)$	41	1st-order + jump	$\mathcal{N}(0,1.0^2)$
S7	31	1.00	7	1.00	$\mathcal{U}(-1,1)$	41	1st-order + jump	$\mathcal{N}(0,2.5^2)$
S8	31	1.00	7	1.00	$\mathcal{U}(-1,1)$	41	1st-order + jump	$\mathcal{N}(0,5.0^2)$
S9	31	1.00	7	1.00	$\mathcal{N}(0,0.5^2)$	41	1st-order + jump	$\mathcal{N}(0,5.0^2)$
S10	31	1.00	7	1.00	$\mathcal{N}(0,1.0^2)$	41	1st-order + jump	$\mathcal{N}(0,5.0^2)$
S11	31	1.00	7	1.00	$\mathcal{N}(0,1.5^2)$	41	1st-order + jump	$\mathcal{N}(0,5.0^2)$
S12	31	1.00	7	1.00	$\mathcal{N}(0,2.0^2)$	41	1st-order + jump	$\mathcal{N}(0,5.0^2)$
S13	31	1.00	7	1.00	$\mathcal{N}(0,0.5^2)$	41	1st-order + jump	$\mathcal{N}(0,1.0^2)$
S14	31	1.00	7	1.00	$\mathcal{N}(0,1.0^2)$	41	1st-order + jump	$\mathcal{N}(0,1.0^2)$
S15	31	1.00	7	1.00	$\mathcal{N}(0,1.5^2)$	41	1st-order + jump	$\mathcal{N}(0,1.0^2)$
S16	31	1.00	7	1.00	$\mathcal{N}(0,0.5^2)$	41	1st-order	$\mathcal{N}(0,1.0^2)$
S17	31	1.00	7	1.00	$\mathcal{N}(0,1.0^2)$	41	1st-order	$\mathcal{N}(0,1.0^2)$
S18	31	1.00	7	1.00	$\mathcal{N}(0,1.5^2)$	41	1st-order	$\mathcal{N}(0,1.0^2)$
S19	31	1.00	7	1.00	$\mathcal{N}(0,0.5^2)$	41	1st-order	$\mathcal{N}(0,5.0^2)$
S20	31	1.00	7	1.00	$\mathcal{N}(0,1.0^2)$	41	1st-order	$\mathcal{N}(0,5.0^2)$
S21	31	1.00	7	1.00	$\mathcal{N}(0,1.5^2)$	41	1st-order	$\mathcal{N}(0,5.0^2)$

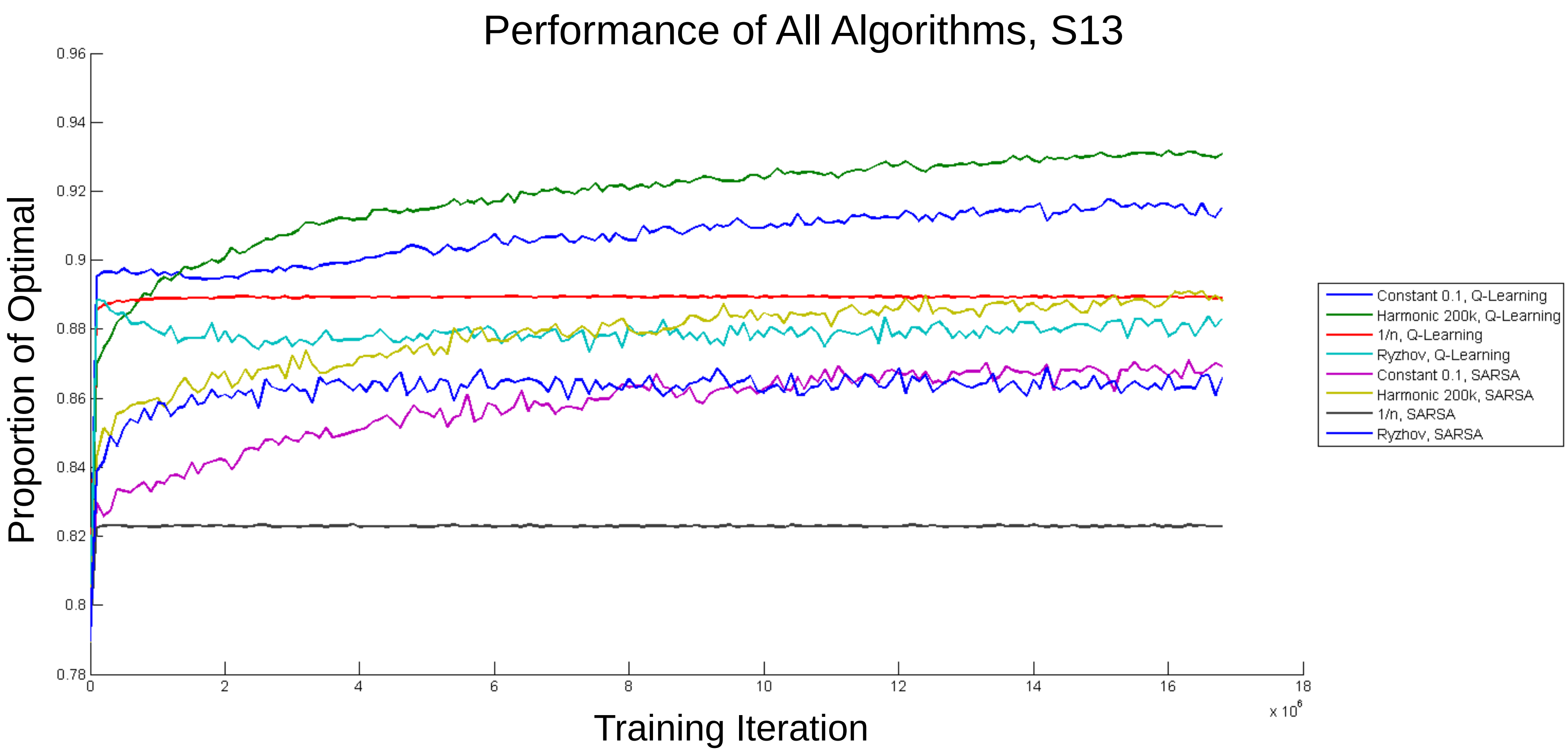
Table 2: Stochastic test problems.

From *Benchmarking a Scalable Approximate Dynamic Programming Algorithm for Stochastic Control of Multidimensional Energy Storage Problems* (Salas, Powell)

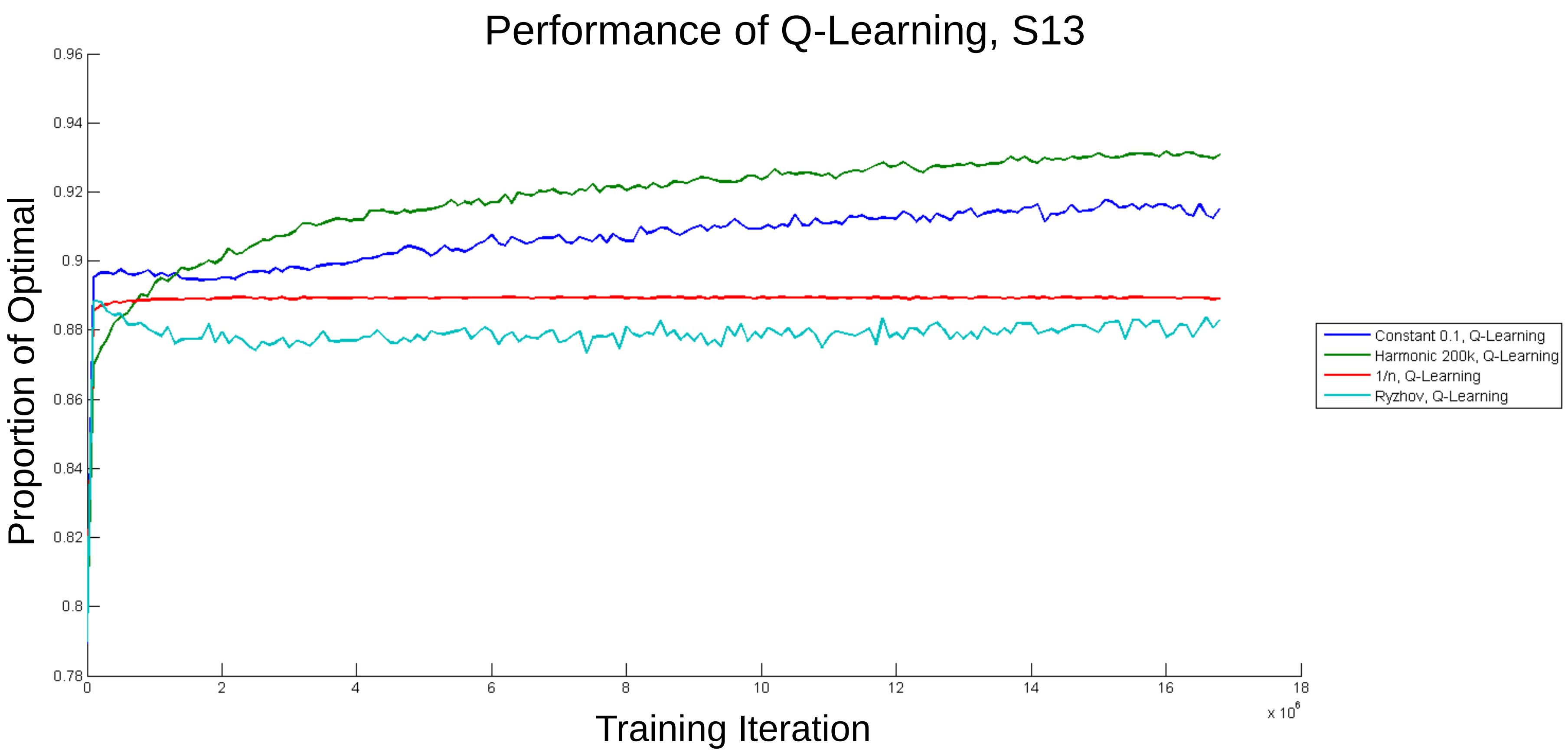
AGGREGATE STORAGE



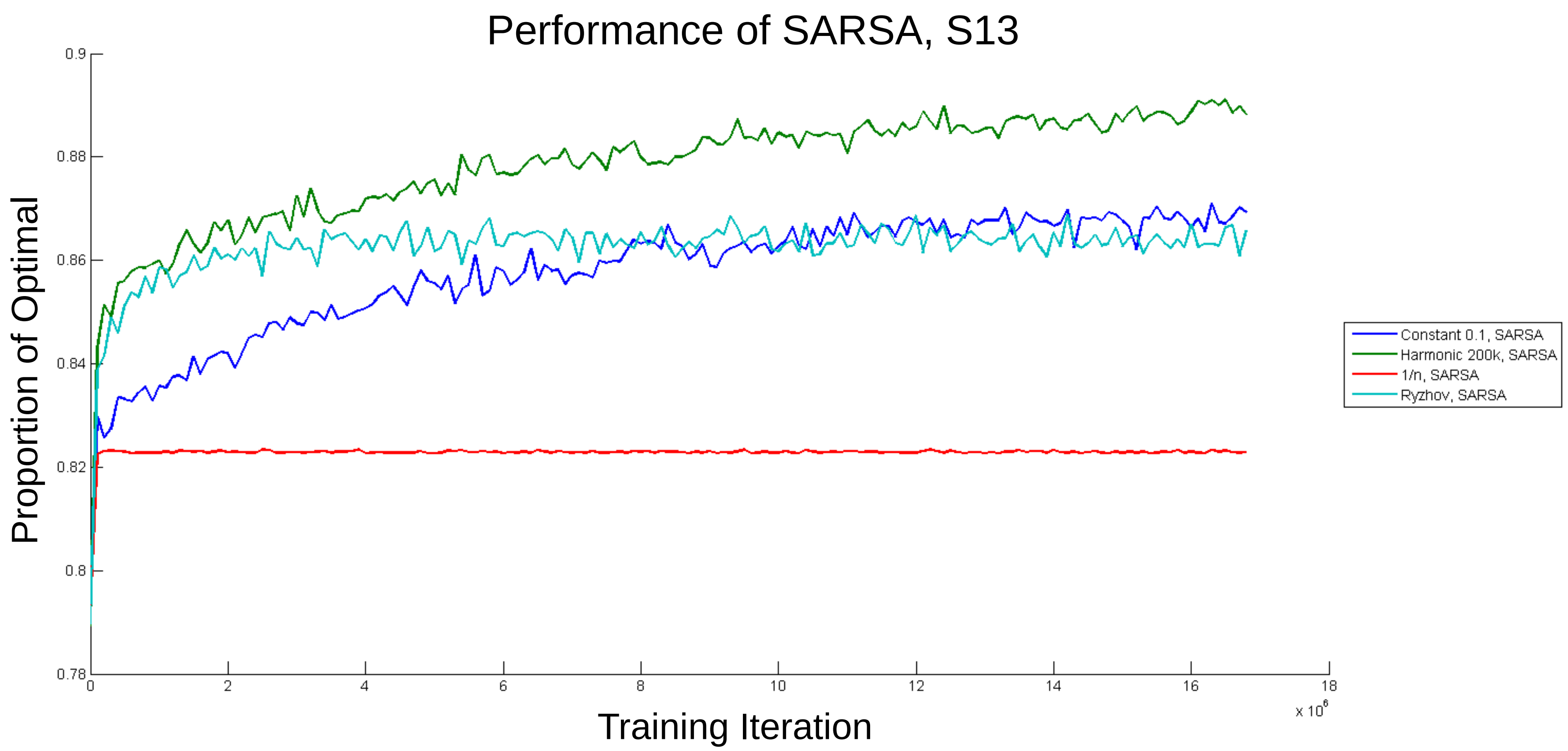
AGGREGATE PERFORMANCE



AGGREGATE PERFORMANCE



AGGREGATE PERFORMANCE

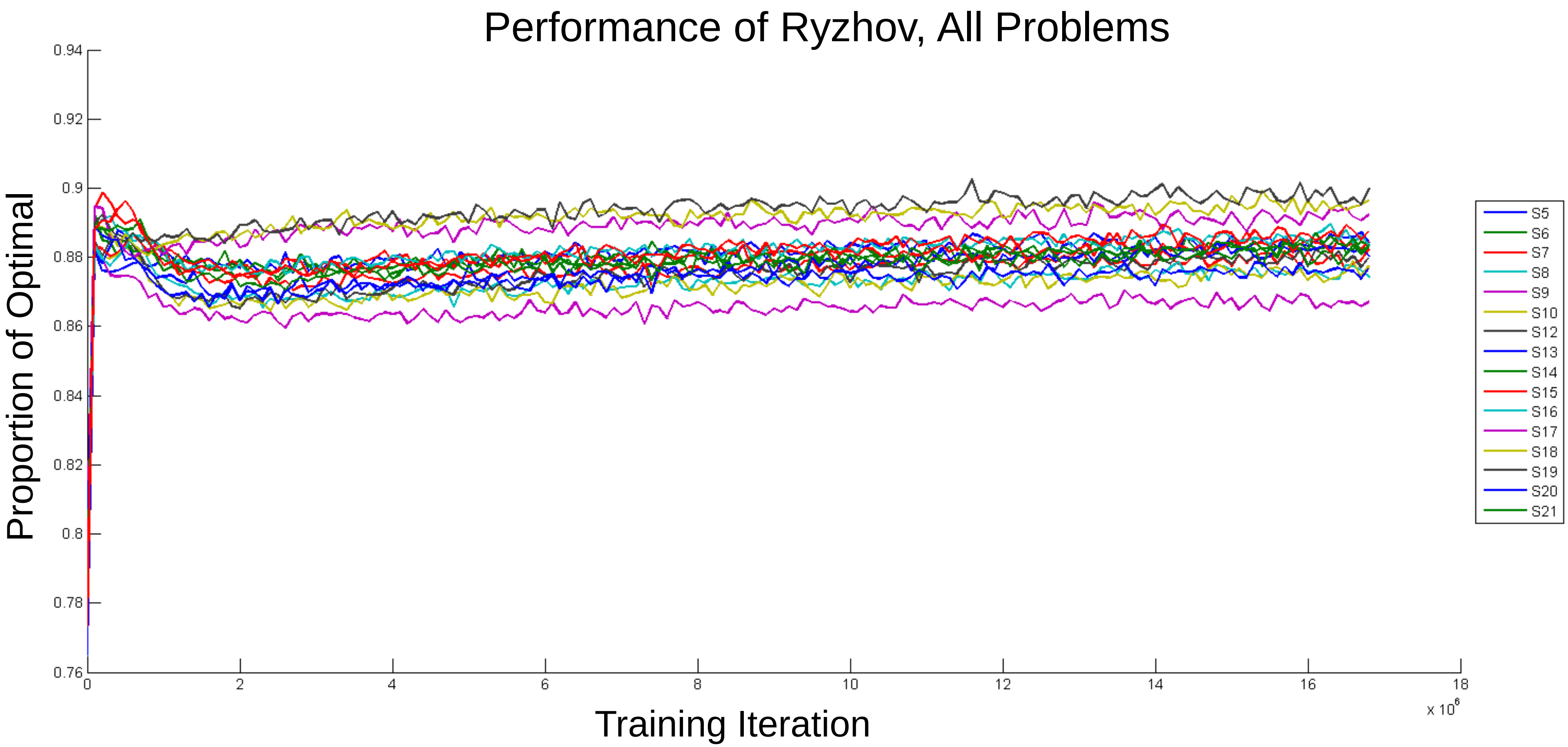


Explanations

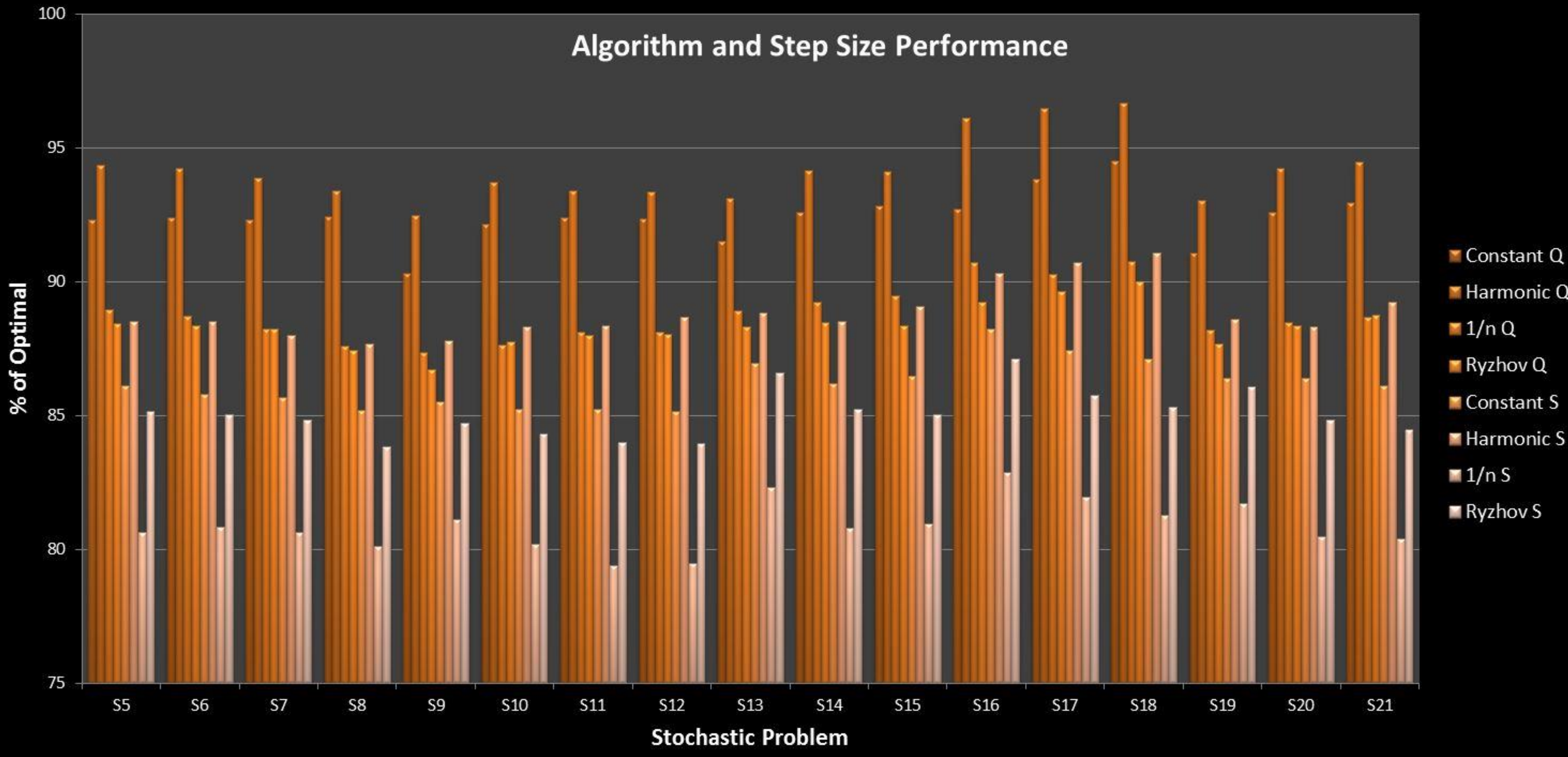
- Ryzhov flattens out very noticeably because the step size is too high, it repeatedly overcompensates and bounces around.

- Harmonic looks like it's still getting better! Need to have it decline more slowly / not go to 0 so quickly. Ryzhov harmonic constant maybe?

STEP SIZE PERFORMANCE



AGGREGATE PERFORMANCE



Overview

- Motivation
- Model
- Algorithms
- Methodology
- Data
- Results
- **Further Exploration**

FURTHER EXPLORATION

- Additional renewable energy sources / storage devices
- Finer levels of discretization
- Remove wind usage restriction
- Additional algorithms
  - Actor-Critic
  - Gradient Descent (Linear/Non-Linear VFAs)

Questions?