# TD Learning

Sean Meyn

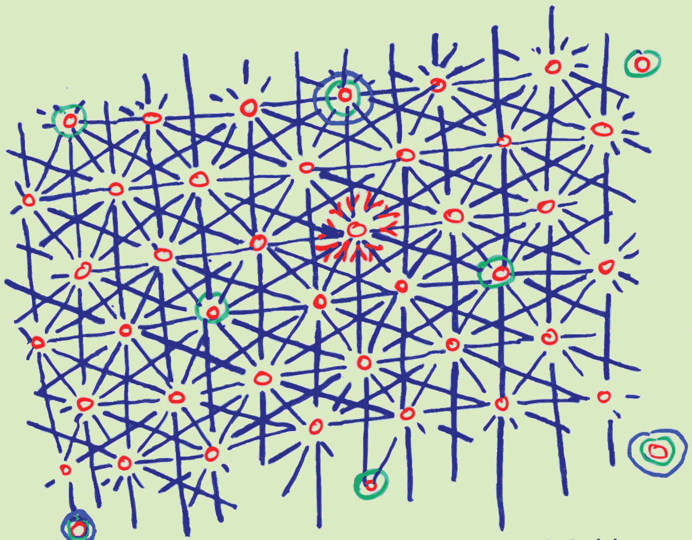
Department of Electrical and Computer Engineering
University of Illinois & the Coordinated Science Laboratory

Joint work with

- I.-K. Cho, Illinois
- S. Mannor, McGill
- V. Tadic, Sheffield
- S. Henderson, Cornell

NSF support: ECS 05-23620 and DARPA ITMANET

# Art



- A.Rybko, 2006

# Value Function Solidarity




Relative value function

$$h(x) = \int_0^\infty \mathsf{E}[c(Q(t;x)) - \eta]\, dt$$

$$\eta = \int c(x)\, \pi(dx)$$

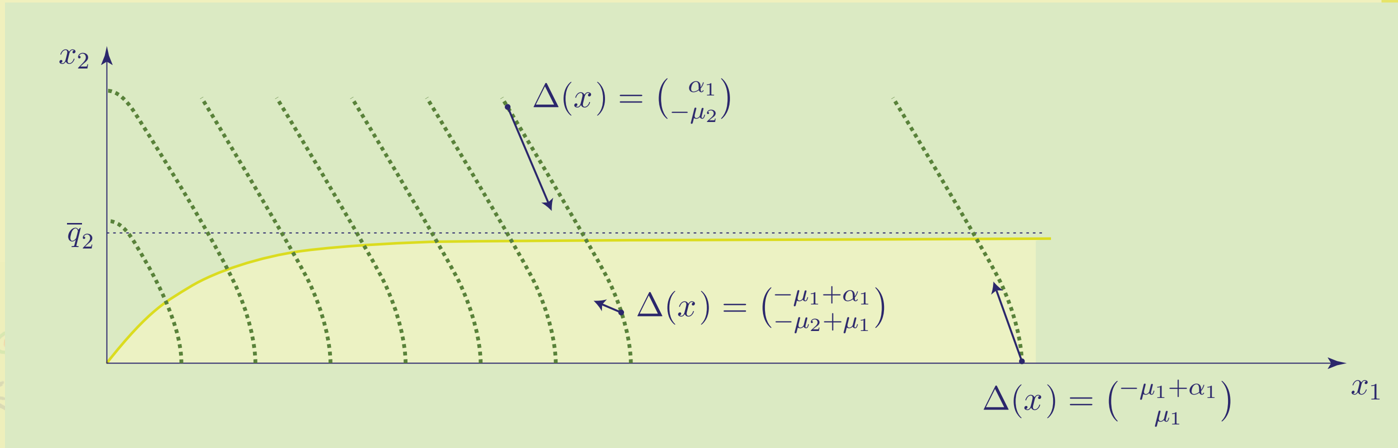
Fluid value function

$$J(x) = \int_0^\infty c(q(t;x))\, dt$$

$$\lim_{\|x\|\to\infty} \left[\frac{J(x)}{h(x)}\right] = 1$$

Recollec

# Lyapunov Functions, and Policy Translation



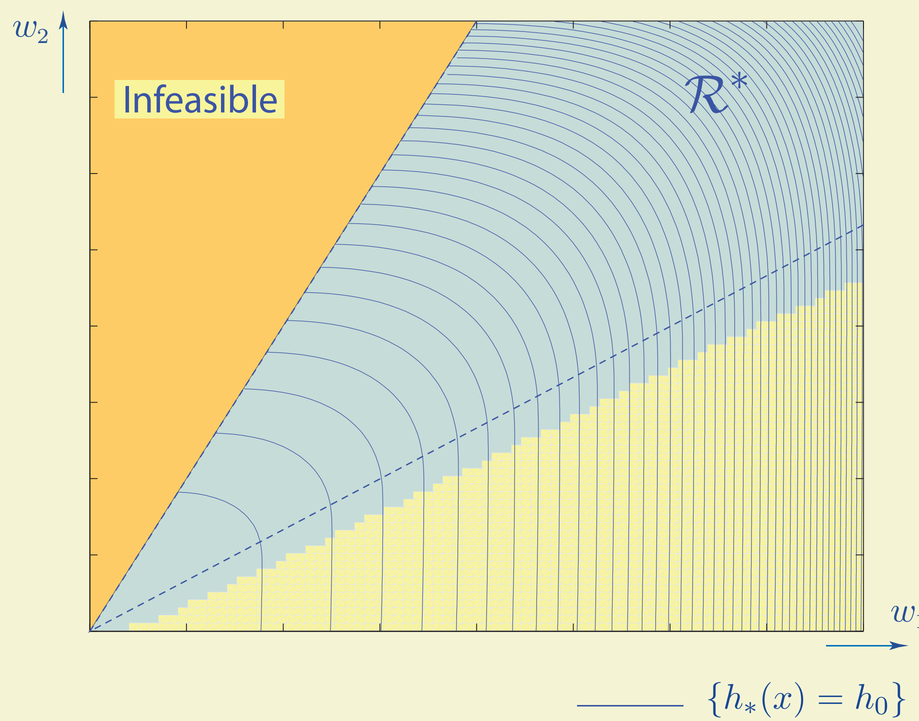



Art

$$V(x) = J(\tilde{x})$$

$$PV(x) := \mathsf{E}[V(Q(k+1))|Q(k)=x]$$
$$\leq V(x) - \varepsilon\|x\| + \bar{\eta}$$

# Outline



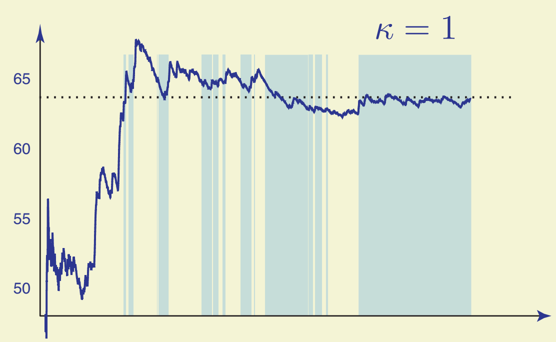

Value function approximation

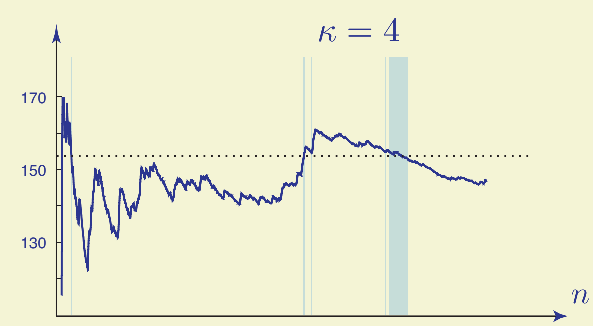
- Performance evaluation
- Perfromance improvement
- Simulation

TD learning

Numerics

Conclusions



# Outline

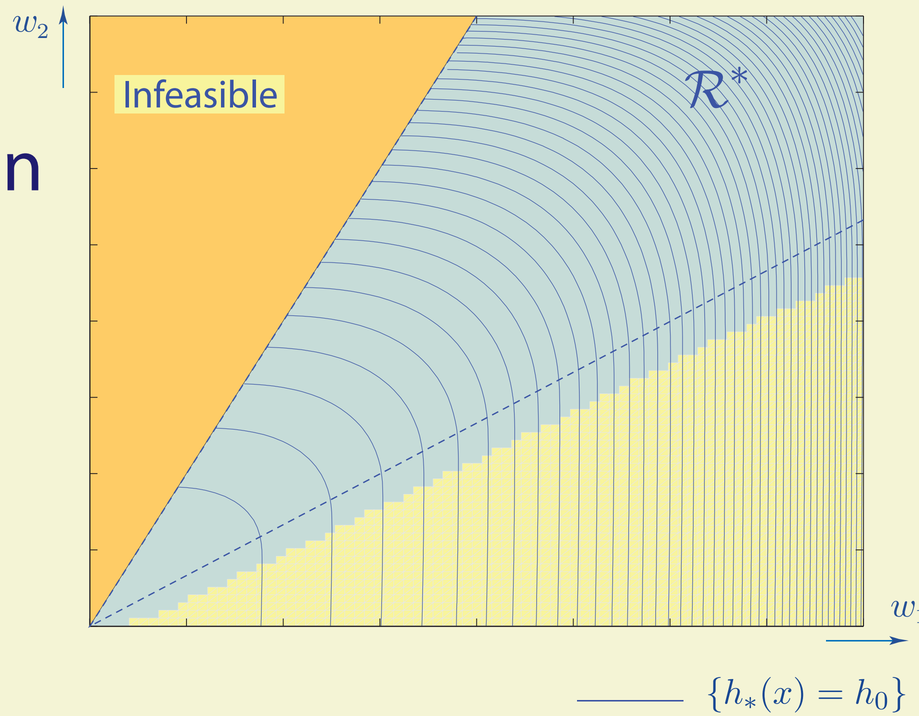
## Value function approximation

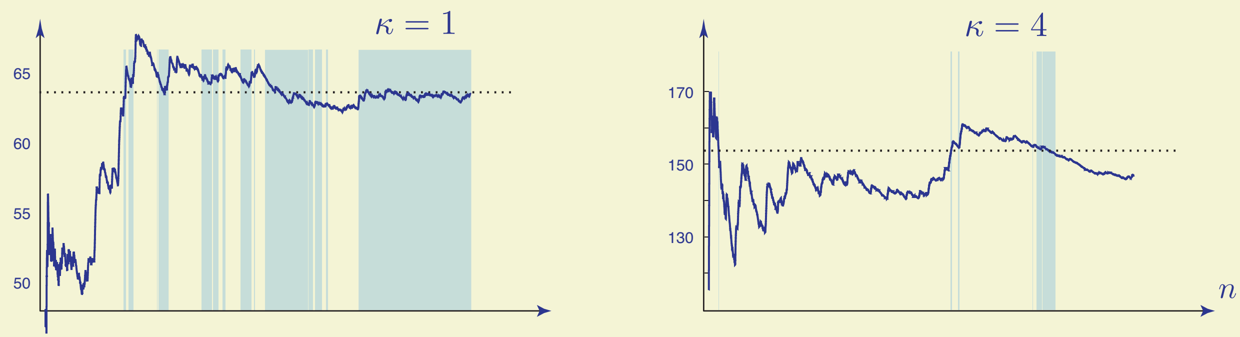
- Performance evaluation
- Perfromance improvement
- Simulation

## TD learning

## Numerics

## Conclusions



# Average cost

$\{ X(t) : t = 0, 1, 2, \dots \}$ Ergodic Markov chain on $\mathsf{X}$ (countable)

$P(x, y)$ Transition matrix

$\pi P = \pi$ Invariant probability measure

$c \colon \mathsf{X} \to \mathbb{R}_+$ Cost function (near monotone)

$\eta = \pi(c)$ Finite average cost

$$\eta = \pi(c) = \lim_{n\to\infty} \frac{1}{n} \sum_{t=0}^{n} c(X(t))$$

# Value functions

Value functions:

Discounted

$$h(x) := \mathsf{E}\Big[\sum_{t=0}^{\infty}(1+\gamma)^{-t-1}c(X(t))\Big]$$

Relative

$$h(x) := \mathsf{E}\Big[\sum_{t=0}^{\tau_0}(c(X(t)) - \eta)\Big]$$

$$X(0) = x \in \mathsf{X}$$

# Value functions

## Value functions:

Discounted

$$h(x) := \mathsf{E}\Big[\sum_{t=0}^{\infty}(1+\gamma)^{-t-1}c(X(t))\Big]$$

Relative

$$h(x) := \mathsf{E}\Big[\sum_{t=0}^{\tau_0}(c(X(t)) - \eta)\Big]$$

$$X(0) = x \in \mathsf{X}$$

## Dynamic programming equations: $\mathcal{D} = P - I$

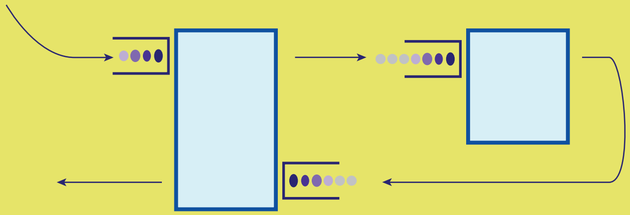
Discounted

$$[\gamma I - \mathcal{D}]h = c$$

Relative

$$\mathcal{D}h = c - \eta$$

# Value function approximation



Discounted-cost value function:

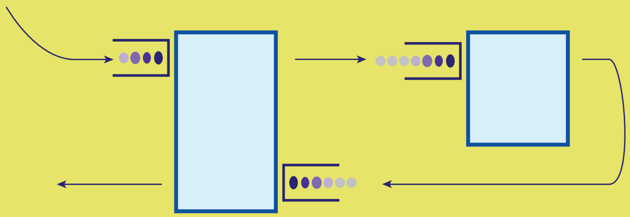
$$h(x) = \mathsf{E}\Big[\sum_{t=0}^{\infty}(1+\gamma)^{-t-1}c(X(t))\Big]$$

*Network model with* $c(x)$ *a norm on X*

Under general conditions, approximated by scaled cost:

$$\lim_{r\to\infty}\frac{1}{r}h(rx) = \frac{1}{\gamma}c(x)$$

# Value function approximation



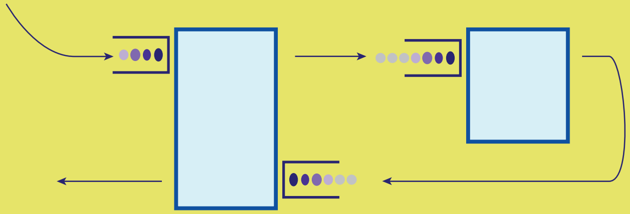
Relative value function,

$$h(x) = \mathsf{E}\Big[\sum_{t=0}^{\tau_0} (c(X(t)) - \eta)\Big]$$

$h$ solves Poisson's equation,

$$\mathcal{D}\, h = -c + \eta$$

# Value function approximation



Value function approximated by fluid model:

$$h(x) = \mathsf{E}\Big[\sum_{t=0}^{\tau_0} (c(X(t)) - \eta)\Big] \qquad J(x) = \int_0^\infty c(q(t))\,dt$$

Relative value function

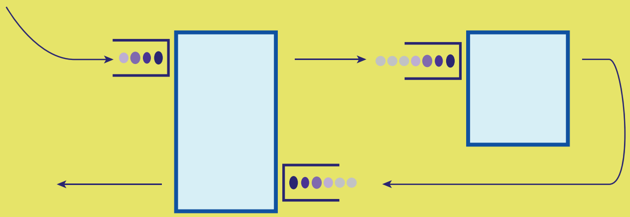
Fluid value function

Under general conditions,

$$\lim_{r\to\infty} \frac{1}{r^2} h(rx) = J(x)$$

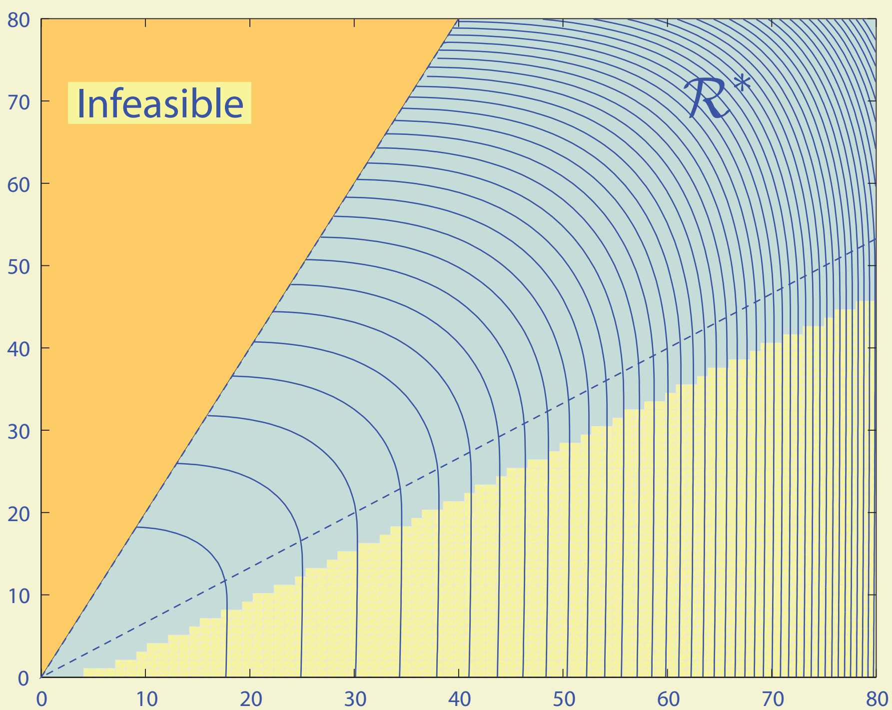
M' 97, 01, 05

# Value function approximation



Value function approximated by fluid model:

$$h(x) = \mathsf{E}\Big[\sum_{t=0}^{\tau_0}(c(X(t)) - \eta)\Big] \qquad J(x) = \int_0^\infty c(q(t))\,dt$$




Workload model: level sets of $h$

$\widehat{\iota}_2(k) = 0$

$\widehat{\iota}_2(k) > 0$

$\{h_*(x) = h_0\}$

# Value function approximation: Simulation

Simulation: Standard Monte Carlo,

$$\eta_k = k^{-1} \sum_{t=0}^{k-1} c(X(t))$$

CLT variance: $\sigma^2 = \pi(h^2 - (Ph)^2)$

where $h$ solves Poisson's equation $\mathcal{D}\, h = -c + \eta$

# Value function approximation: Simulation

Simulation: Standard Monte Carlo,

$$\eta_k = k^{-1} \sum_{t=0}^{k-1} c(X(t))$$

CLT variance: $\sigma^2 = \pi(h^2 - (Ph)^2)$

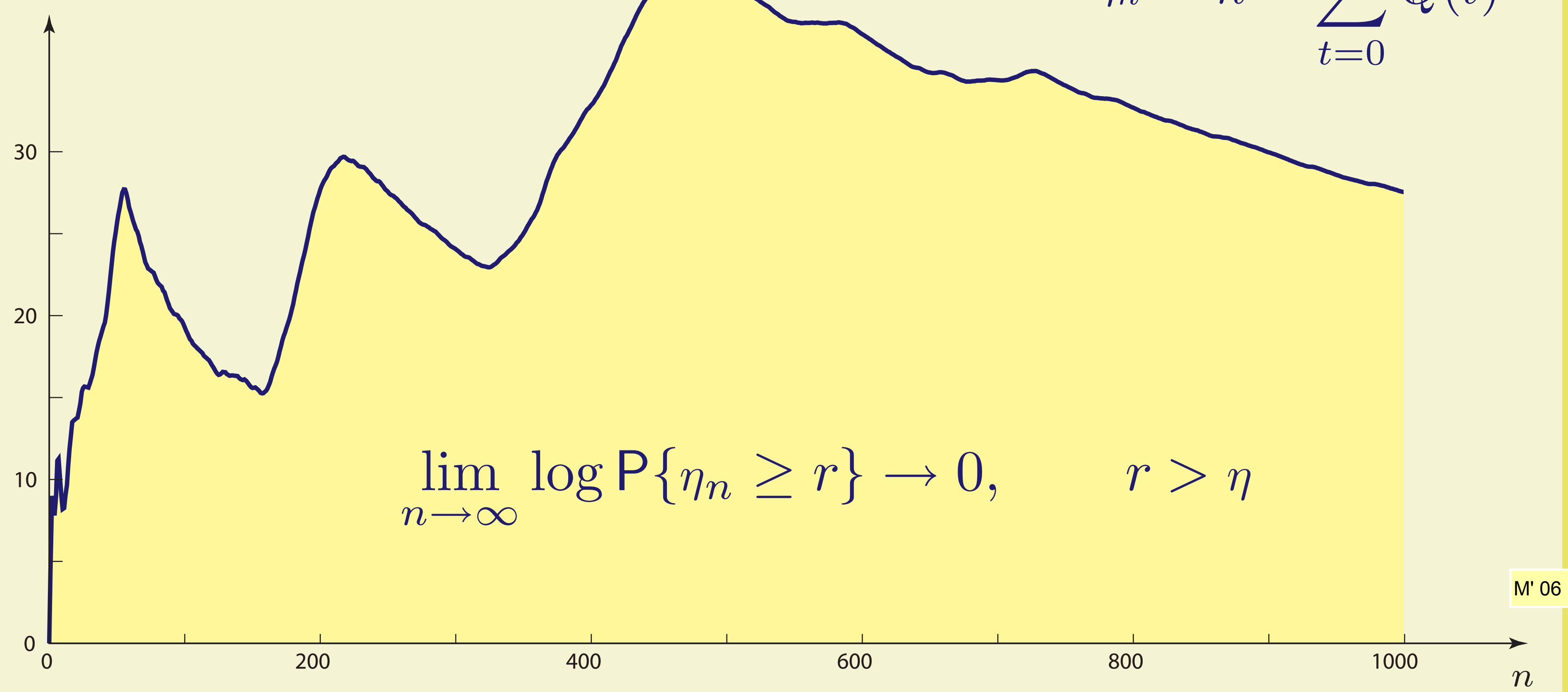
where $h$ solves Poisson's equation $\mathcal{D}\, h = -c + \eta$

Example: Simulating the single queue: CLT variance = $O\Big(\frac{1}{(1-\rho)^4}\Big)$

Henderson & Glynn '02, Whitt 94

# Value function approximation: Simulation

$$\eta_n = n^{-1} \sum_{t=0}^{n-1} Q(t)$$




$$\lim_{n\to\infty} \log \mathsf{P}\{\eta_n \geq r\} \to 0, \qquad r > \eta$$

Example: Simulating the single queue: CLT variance = $O\Big(\frac{1}{(1-\rho)^4}\Big)$

# Value function approximation: Simulation

Control variates:

$$g : \mathsf{X} \to \mathbb{R} \qquad \textit{has known mean} \qquad \pi(g) = 0,$$

Smoothed estimator:

$$\eta_k(\vartheta) = k^{-1} \sum_{t=0}^{k-1} \Bigl( c(X(t)) - \vartheta g(X(t)) \Bigr), \qquad k \ge 1$$

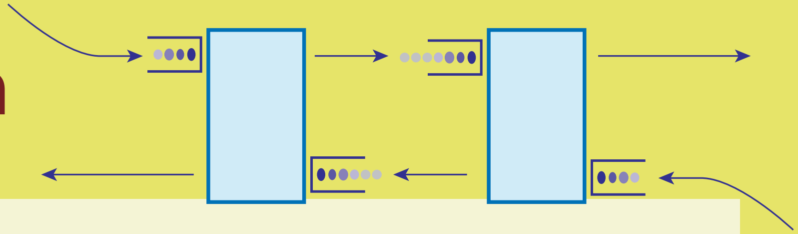
Perfect estimator:

$$g = h - Ph = -\mathcal{D}h = c - \eta$$

Zero CLT variance

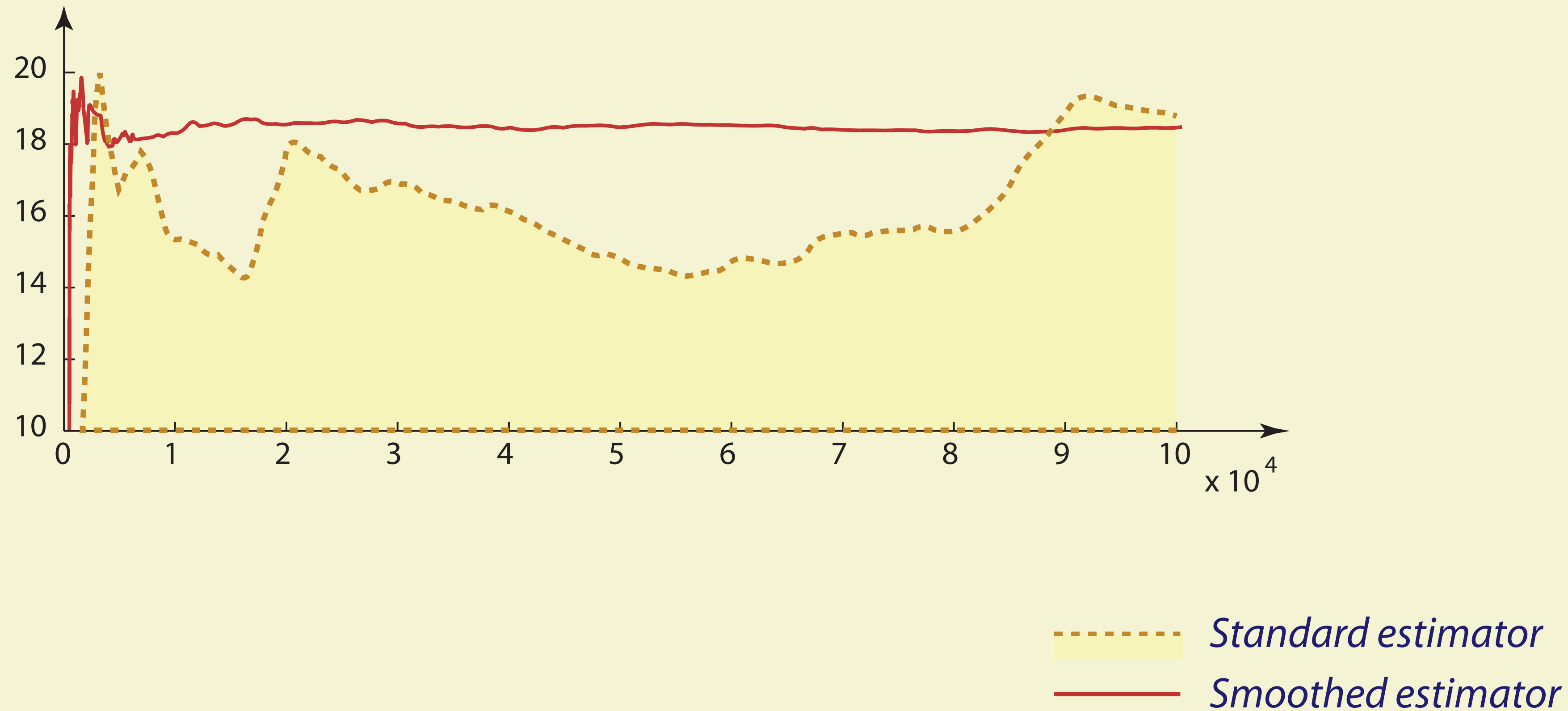
Nelson '90,Henderson & Glynn '02, Glynn & Szechtman '02, M & Henderson '02

# Steady state customer population



## Smoothed estimator using fluid value function:

$$g = h - Ph = -\mathcal{D}h, \qquad h = J$$




HMT '03

# Outline

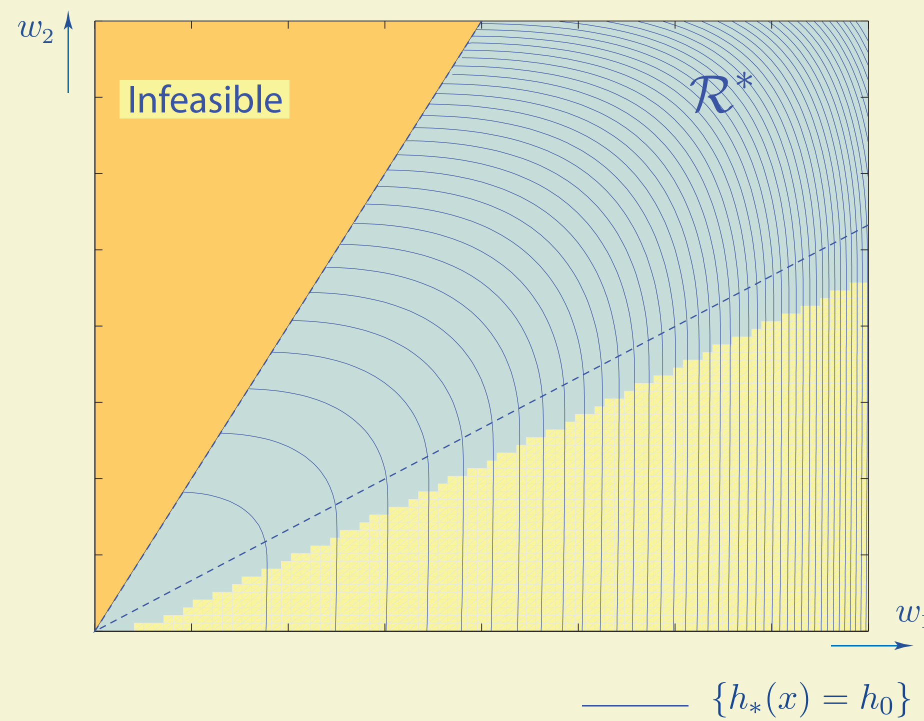
Value function approximation

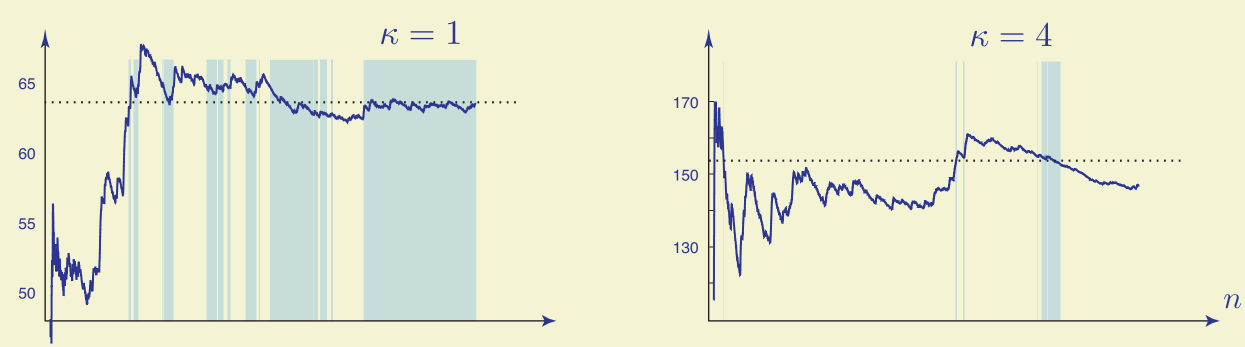
- Performance evaluation
- Perfromance improvement
- Simulation

**TD learning**

Numerics

Conclusions



# Value function approximation

Linear parametrization: $h^\theta(x) := \theta^T \psi(x) = \sum_{i=1}^{\ell_h} \theta_i \psi_i(x), \qquad x \in \mathsf{X}$

How to find the best parameter $\theta$ ?

How to define *best*?

# Value function approximation: Hilbert space setting

Linear parametrization: $h^\theta(x) := \theta^T \psi(x) = \sum_{i=1}^{\ell_h} \theta_i \psi_i(x), \qquad x \in \mathsf{X}$

Minimize the mean square error $\|h^\theta - h\|_\pi^2$

# Value function approximation: Hilbert space setting

Linear parametrization: $h^\theta(x) := \theta^T \psi(x) = \sum_{i=1}^{\ell_h} \theta_i \psi_i(x), \qquad x \in \mathsf{X}$

Minimize the mean square error $\|h^\theta - h\|_\pi^2$

$$= \mathsf{E}_\pi[(h^\theta(X(k)) - h(X(t)))^2]$$

$$\nabla_\theta \|h^\theta - h\|_\pi^2 = 2\mathsf{E}_\pi[(h^\theta(X(k)) - h(X(t)))\,\psi(X(t))]$$

Setting the gradient to zero gives optimal parameter

# Value function approximation: Hilbert space setting

Linear parametrization: $h^\theta(x) := \theta^T \psi(x) = \sum_{i=1}^{\ell_h} \theta_i \psi_i(x), \qquad x \in \mathsf{X}$

Minimize the mean square error $\|h^\theta - h\|_\pi^2$

$$= \mathsf{E}_\pi[(h^\theta(X(k)) - h(X(t)))^2]$$

$$\nabla_\theta \|h^\theta - h\|_\pi^2 = 2\mathsf{E}_\pi[(h^\theta(X(k)) - h(X(t)))\ \psi(X(t))]$$

Setting the gradient to zero gives optimal parameter

$$\theta^* = \Sigma^{-1} b \qquad b = \mathsf{E}_\pi[h(X)\psi(X)]\,,\ \Sigma = \mathsf{E}_\pi[\psi(X)\psi(X)^T]$$

# Value function approximation: Hilbert space setting

Define for two functions $f,\ g$ on X, $$\langle f, g\rangle_\pi = \pi(fg)$$

# Value function approximation: Hilbert space setting

Define for two functions $f, \ g$ on X, $\langle f, g \rangle_\pi = \pi(fg)$

$L_2$ norm,

$$\|f - g\|_\pi^2 := \langle f - g, f - g \rangle_\pi = \mathsf{E}_\pi[(f(X(0)) - g(X(0)))^2]$$

# Value function approximation: Hilbert space setting

Define for two functions $f,\ g$ on X, $\langle f, g\rangle_\pi = \pi(fg)$

$L_2$ norm,

$$\|f-g\|_\pi^2 := \langle f-g, f-g\rangle_\pi = \mathsf{E}_\pi[(f(X(0)) - g(X(0)))^2]$$

Resolvent, $$R_\gamma = \sum_{t=0}^{\infty} (1+\gamma)^{-(t+1)} P^t$$

# Value function approximation: Hilbert space setting

Define for two functions $f,\ g$ on X, $\langle f, g\rangle_\pi = \pi(fg)$

$L_2$ norm,

$$\|f-g\|_\pi^2 := \langle f-g, f-g\rangle_\pi = \mathsf{E}_\pi[(f(X(0)) - g(X(0)))^2]$$

Resolvent, $$R_\gamma = \sum_{t=0}^{\infty} (1+\gamma)^{-(t+1)} P^t$$

$$b = \mathsf{E}_\pi[h(X)\psi(X)] \ = \langle R_\gamma c, \psi\rangle_\pi$$

# Value function approximation: Hilbert space setting

Define for two functions $f,\ g$ on X, $\langle f, g\rangle_\pi = \pi(fg)$

$L_2$ norm,

$$\|f - g\|_\pi^2 := \langle f-g, f-g\rangle_\pi = \mathsf{E}_\pi[(f(X(0)) - g(X(0)))^2]$$

Resolvent, $$R_\gamma = \sum_{t=0}^{\infty} (1+\gamma)^{-(t+1)} P^t$$

Adjoint,

$$\langle R_\gamma f, g\rangle_\pi = \langle f, R_\gamma^\dagger g\rangle_\pi\,, \qquad f, g \in L_2(\pi)$$

# Representing the adjoint

For the stationary process,

$$\langle R_\gamma f, g\rangle_\pi = \mathsf{E}\Big[\Big(\sum_{t=0}^{\infty}(1+\gamma)^{-t-1}P^t f\,(X(0))\Big)g(X(0))\Big]$$

(discounted cost)

# Representing the adjoint

For the stationary process,

$$\langle R_\gamma f, g\rangle_\pi = \mathsf{E}\Big[\Big(\sum_{t=0}^{\infty}(1+\gamma)^{-t-1}P^t f\,(X(0))\Big)g(X(0))\Big]$$

Smoothing,

$$\mathsf{E}[P^t f\,(X(0))g(X(0))] = \mathsf{E}[f(X(t))g(X(0))] \;=\; \mathsf{E}[f(X(0))g(X(-t))]$$

# Representing the adjoint

For the stationary process,

$$\langle R_\gamma f, g\rangle_\pi = \mathsf{E}\Big[\Big(\sum_{t=0}^{\infty}(1+\gamma)^{-t-1}P^t f\,(X(0))\Big)g(X(0))\Big]$$

Smoothing,

$$\mathsf{E}[P^t f\,(X(0))g(X(0))] = \mathsf{E}[f(X(t))g(X(0))] \;=\; \mathsf{E}[f(X(0))g(X(-t))]$$

$$\langle R_\gamma f, g\rangle_\pi = \sum_{t=0}^{\infty}(1+\gamma)^{-t-1}\mathsf{E}[f(X(0))g(X(-t))]$$

# Representing the adjoint

For the stationary process,

$$\langle R_\gamma f, g\rangle_\pi = \mathsf{E}\Big[\Big(\sum_{t=0}^{\infty}(1+\gamma)^{-t-1}P^t f\,(X(0))\Big)g(X(0))\Big]$$

Smoothing,

$$\mathsf{E}[P^t f\,(X(0))g(X(0))] = \mathsf{E}[f(X(t))g(X(0))] \;=\; \mathsf{E}[f(X(0))g(X(-t))]$$

$$\langle R_\gamma f, g\rangle_\pi = \sum_{t=0}^{\infty}(1+\gamma)^{-t-1}\mathsf{E}[f(X(0))g(X(-t))] = \langle f, R_\gamma^\dagger g\rangle_\pi$$

# Representing the adjoint

Adjoint is defined in terms of the time-reversed process

$$R_\gamma^\dagger g\ (x) \ = \ \sum_{t=0}^{\infty} (1+\gamma)^{-t-1} \mathsf{E}[\, g(X(-t)) \mid X(0) = x\,]$$

Causal $\longrightarrow$ computable!

# TD Learning for discounted cost

Causal ⟶ computable!

$$\nabla_\theta \, \tfrac{1}{2} \|h^\theta - h\|_\pi^2 \;=\; \mathsf{E}_\pi[(h^\theta(X(k)) - h(X(t)))\;\psi(X(t))]$$

$$= \langle (h^\theta - R_\gamma c), \psi \rangle_\pi$$

$$= \langle R_\gamma (R_\gamma^{-1} h^\theta - c), \psi \rangle_\pi$$

$$= \langle R_\gamma^{-1} h^\theta - c, R_\gamma^\dagger \psi \rangle_\pi$$

# TD Learning for discounted cost

Causal ⟶ computable!

$$\nabla_\theta \tfrac{1}{2}\|h^\theta - h\|_\pi^2 = \mathsf{E}_\pi[(h^\theta(X(k)) - h(X(t)))\,\psi(X(t))]$$

$$= \langle (h^\theta - R_\gamma c), \psi\rangle_\pi$$

$$= \langle R_\gamma(R_\gamma^{-1} h^\theta - c), \psi\rangle_\pi$$

$$= \langle R_\gamma^{-1} h^\theta - c, R_\gamma^\dagger \psi\rangle_\pi \qquad = -\langle d, P\varphi\rangle_\pi$$

Notation

# TD Learning for discounted cost

Causal $\longrightarrow$ computable!

$$\mathsf{E}[d(k)\varphi(k+1)] \approx -\tfrac{1}{2}\nabla_\theta \|h^\theta - h_\gamma\|_\pi^2, \qquad \theta(i) \equiv \theta$$

$$\begin{aligned}
\nabla_\theta\ \tfrac{1}{2}\|h^\theta - h\|_\pi^2 &= \mathsf{E}_\pi[(h^\theta(X(k)) - h(X(t)))\ \psi(X(t))] \\
&= \langle (h^\theta - R_\gamma c), \psi\rangle_\pi \\
&= \langle R_\gamma (R_\gamma^{-1} h^\theta - c), \psi\rangle_\pi \\
&= \langle R_\gamma^{-1} h^\theta - c, R_\gamma^\dagger \psi\rangle_\pi \quad = -\langle\, d, P\varphi\rangle_\pi
\end{aligned}$$

Notation

# TD Learning for discounted cost

Causal ⟶ computable!

$$\mathsf{E}[d(k)\varphi(k+1)] \approx -\tfrac{1}{2}\nabla_\theta \|h^\theta - h_\gamma\|_\pi^2, \qquad \theta(i) \equiv \theta$$

(i) *Temporal differences* defined by,

$$d(k) = -[(1+\gamma)h^{\theta(k)}(X(k)) - h^{\theta(k)}(X(k+1)) - c(X(k))]$$

(ii) *Eligibility vectors*: the sequence of $d$-dimensional vectors,

$$\varphi(k) = \sum_{t=0}^{k} (1+\gamma)^{-t-1}\psi(X(k-t)), \qquad k \geq 1$$

# Algorithms

TD(1) algorithm

$$\theta(k+1) - \theta(k) = a_k d(k) \varphi(X(k+1)), \qquad k \geq 0$$

$a_k$ stochastic approx. gain

# Algorithms

## TD(1) algorithm

$$\theta(k+1) - \theta(k) = a_k d(k) \varphi(X(k+1)), \qquad k \geq 0$$

$a_k$ stochastic approx. gain

Consistent under mild conditions,

$$\theta^* = \Sigma^{-1} b \qquad b = \mathsf{E}_\pi[h(X)\psi(X)]\,,\ \Sigma = \mathsf{E}_\pi[\psi(X)\psi(X)^T]$$

Bertsekas & Tsitsiklis '96, Sutton & Barto '98, Tsitsiklis & Van Roy, '97

# Algorithms

Newton algorithm

$$\theta(k+1) = \Big(\frac{1}{k}\sum_{i=1}^{k}\psi(X(i+1))\psi^T(X(i+1))\Big)^{-1}\Big(\frac{1}{k}\sum_{i=1}^{k}c(i)\varphi(X(i+1))\Big)$$

Consistent under mild conditions,

$$\theta^* = \Sigma^{-1}b \qquad b = \mathsf{E}_\pi[h(X)\psi(X)]\,,\ \Sigma = \mathsf{E}_\pi[\psi(X)\psi(X)^T]$$

# TD Learning for average cost

Potential matrix

$$R_0 g\,(x) = \mathsf{E}_x\Big[\sum_{t=0}^{\sigma_\alpha} g(X(t))\Big], \qquad x \in \mathsf{X}$$

$\sigma_\alpha$ hitting time to $\boldsymbol{\alpha} \in \mathsf{X}$

# TD Learning for average cost

Potential matrix

$$R_0 g\,(x) = \mathsf{E}_x\Big[\sum_{t=0}^{\sigma_\alpha} g(X(t))\Big], \qquad x \in \mathsf{X}$$

$\sigma_\alpha$ hitting time to $\boldsymbol{\alpha} \in \mathsf{X}$

Solves Poisson's equation when $g = c - \pi(c)$

# TD Learning for average cost

## Potential matrix

$$R_0 g\,(x) = \mathsf{E}_x\Big[\sum_{t=0}^{\sigma_\alpha} g(X(t))\Big], \qquad x \in \mathsf{X}$$

$\sigma_\alpha$ hitting time to $\boldsymbol{\alpha} \in \mathsf{X}$

## Adjoint

$$R_0^\dagger g\,(x) = \mathsf{E}\Big[\sum_{\tilde\sigma_\alpha^{[0]} \le t \le 0} g(X(t)) \;\Big|\; X(0) = x\Big] \ , \qquad g \in L_2(\pi)$$

## Backward hitting time

$$\tilde\sigma_\alpha^{[k]} = \max\{t \le k : X(t) = \boldsymbol{\alpha}\}$$

# Value function approximation: Other metrics

**Bellman error:** $\mathsf{E}_\pi[(Ph^\theta(X) - (1+\gamma)h^\theta(X) + c(X))^2]$

(discounted cost)

# Value function approximation: Other metrics

Bellman error: $\mathsf{E}_\pi[(Ph^\theta(X) - (1+\gamma)h^\theta(X) + c(X))^2]$

(discounted cost)

CLT error: $\sigma^2(\theta) = \mathsf{E}_\pi[(h - h^\theta)^2 - (Ph - Ph^\theta)^2]$

($h$ solves Poisson's eqn)

Tadic & M.'03, Kim & Henderson '03, CTCN '07

# Value function approximation: Other metrics

Bellman error: $\mathsf{E}_\pi[(Ph^\theta(X) - (1+\gamma)h^\theta(X) + c(X))^2]$

(discounted cost)

CLT error: $\sigma^2(\theta) = \mathsf{E}_\pi[(h - h^\theta)^2 - (Ph - Ph^\theta)^2]$

($h$ solves Poisson's eqn)

In each case we can construct an adjoint equation to solve for optimal parameter

# Outline

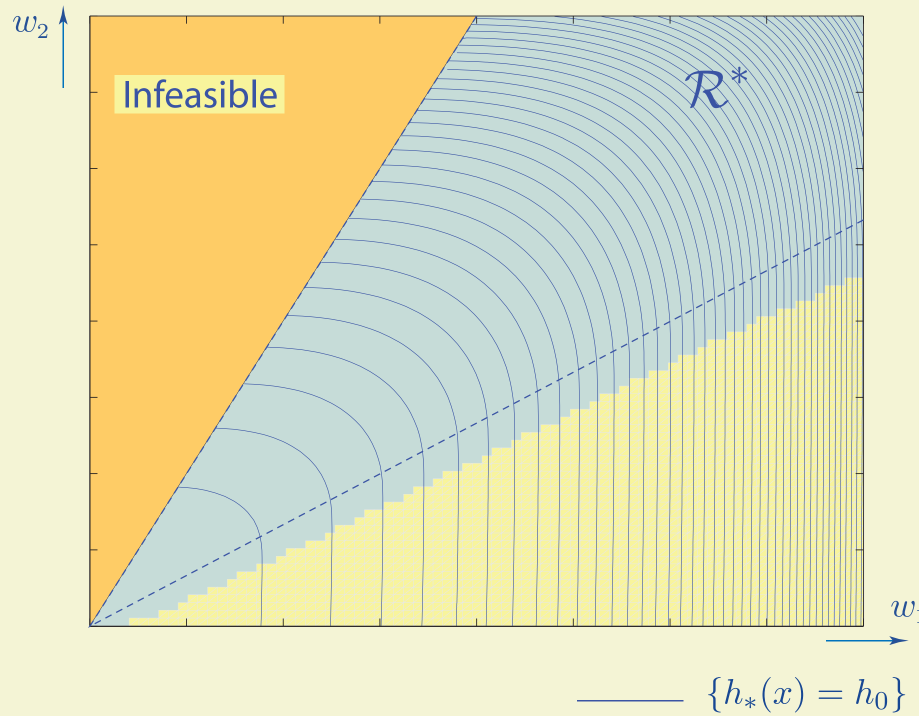
Value function approximation

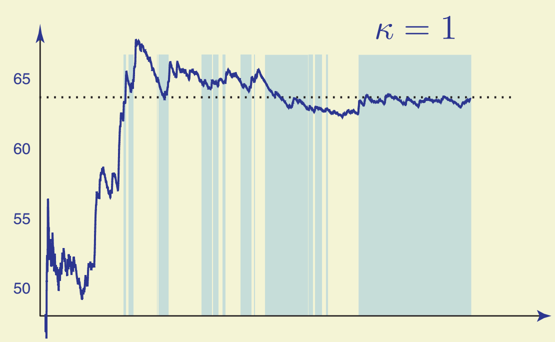
- Performance evaluation
- Perfromance improvement
- Simulation

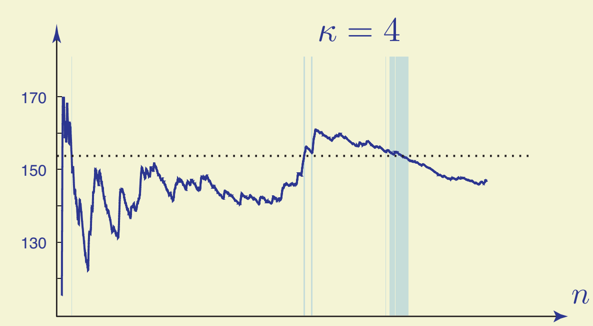
TD learning

**Numerics**

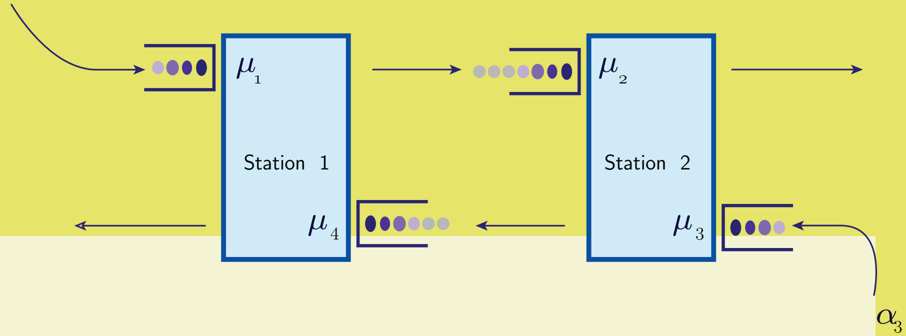
Conclusions



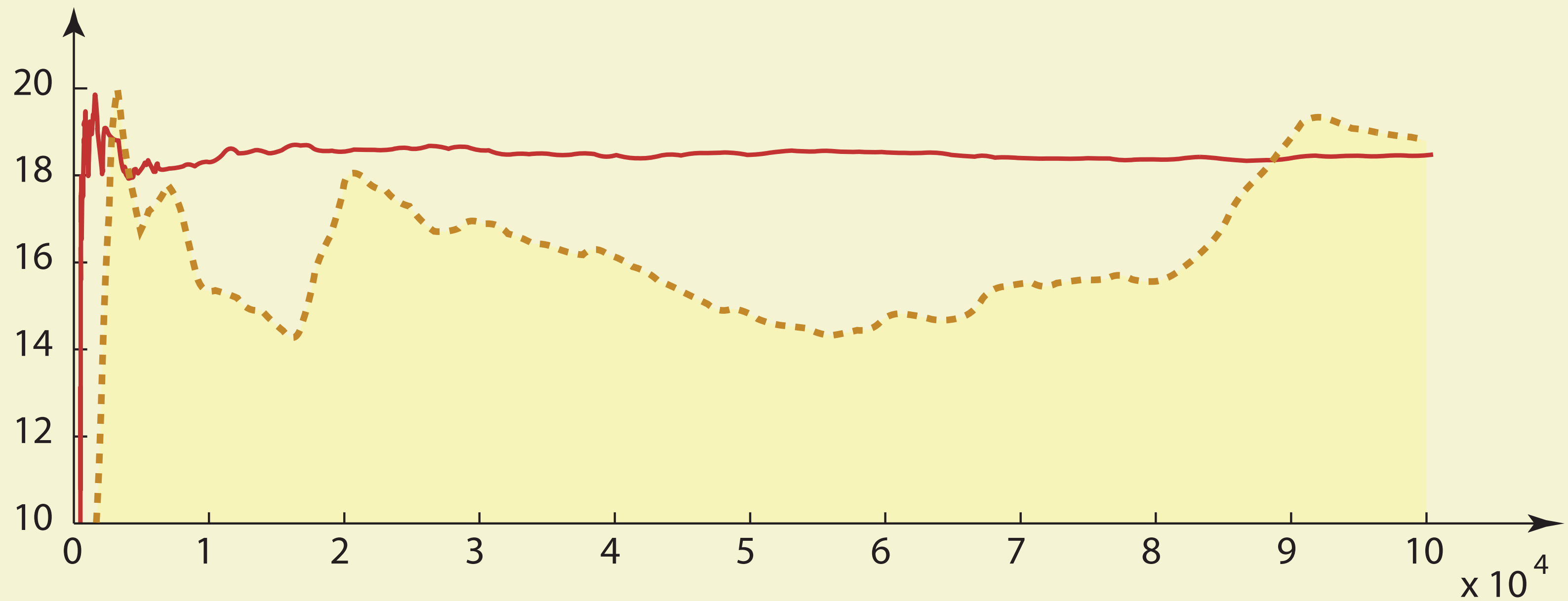



# Once optimal CV is identified:



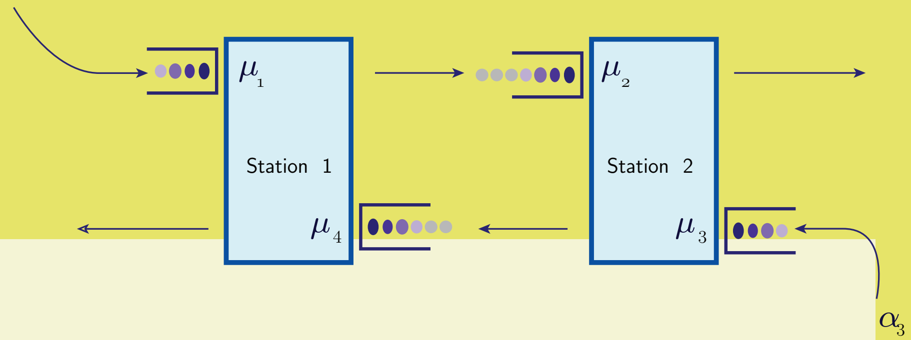

## Smoothed estimator using fluid value function:

$$g = h - Ph = -\mathcal{D}h, \qquad h = J$$




HMT '03

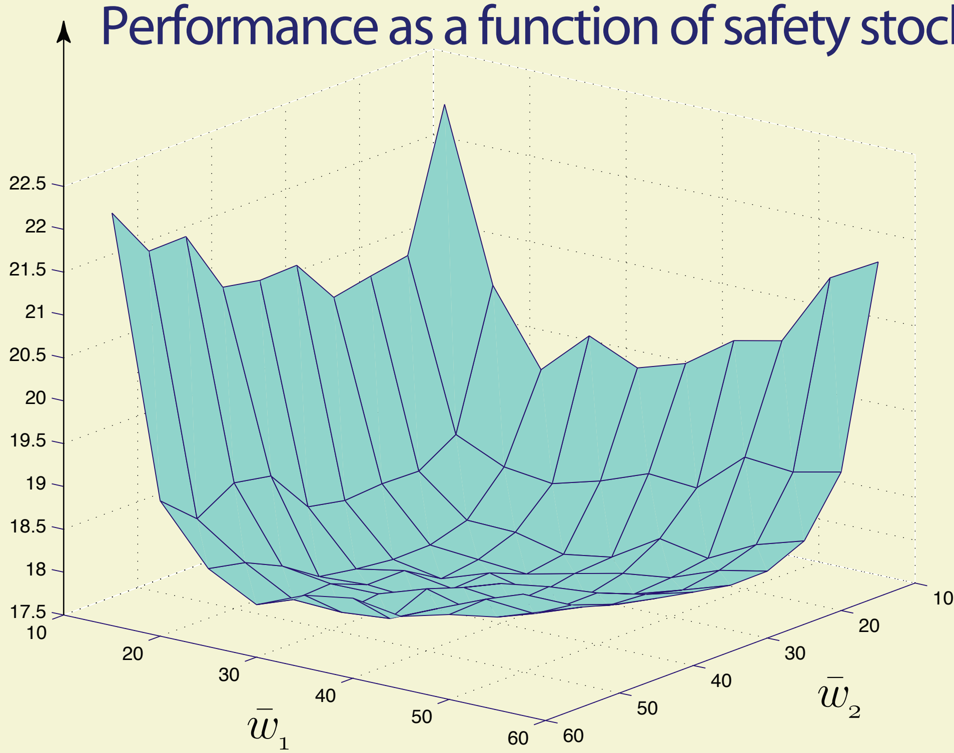
# Policy Selection



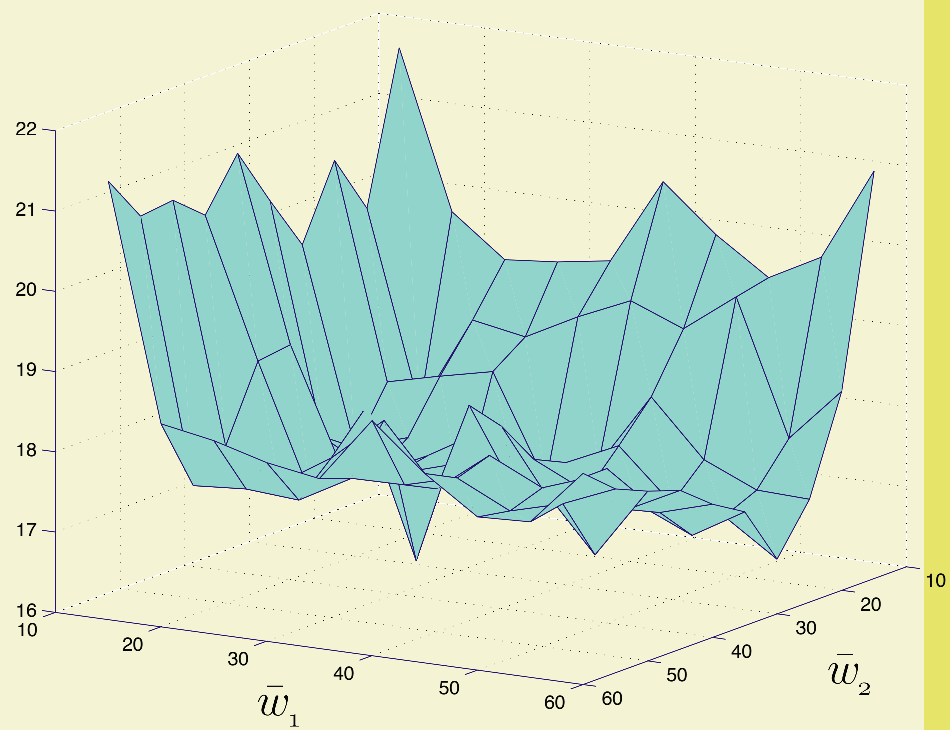

Serve buffer 1 if buffer 4 is empty, or

$$W_2 \geq s_i^* W_1 - \beta_1^* \quad \text{and} \quad m_2 Q_2 + m_3 Q_3 \leq \bar{w}_2$$

Performance as a function of safety stocks




(i) Simulation using smoothed estimator




(i) Simulation using standard estimator

# Optimal speed scaling

Joint work with Adam Wierman and the students of ECE 555

Intel's huge bet turns iffy.
John Markoff and Steve Lohr.
New York Times, September 29 2002.

*What matters most to the computer designers at Google is not speed, but power, low power, because data centers can consume as much electricity as a city.*

—Dr. Eric Schmidt, CEO of Google


[1] O. S. Unsal andI. Koren, "System-level power-aware deisgn techniques in real-time systems," Proc. IEEE, vol. 91, no. 7, pp. 1055–1069, 2003.

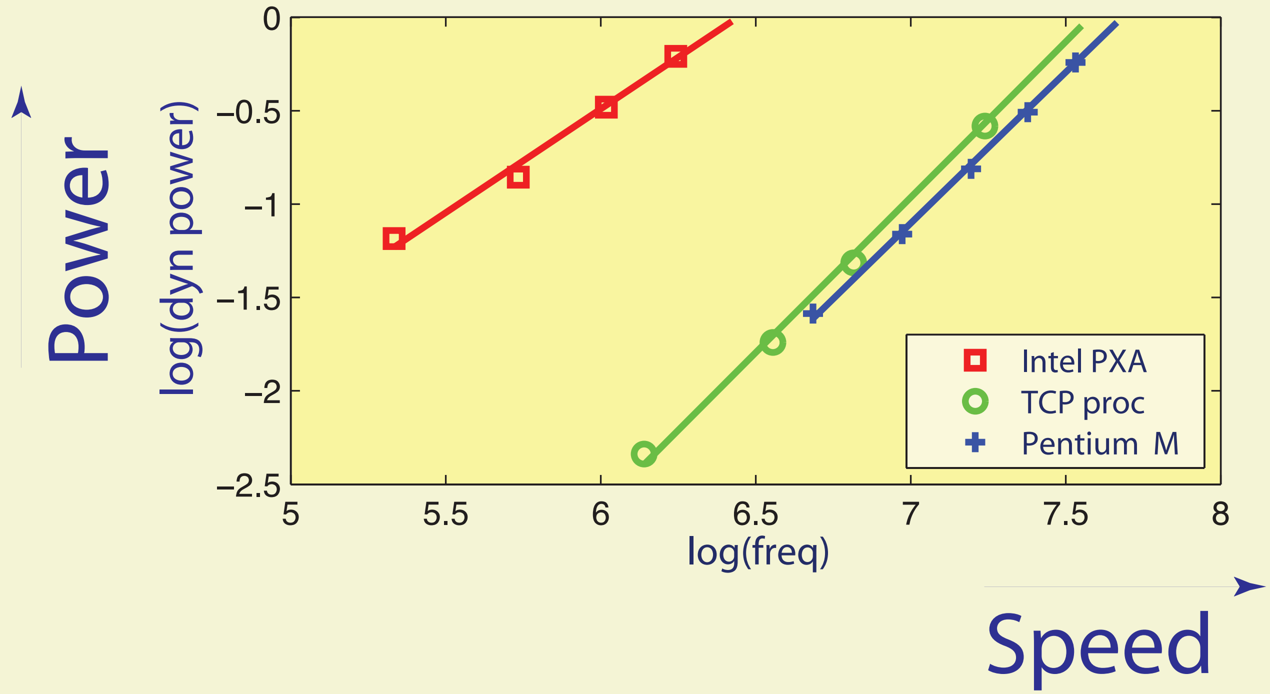
[2] S. Irani andK. R. Pruhs, "Algorithmic problems in power management, "SIGACT News, vol. 36, no. 2, pp. 63–76, 2005.

[3] S. Kaxirasand M. Martonosi, Computer Architecture Techniques for Power-Efficiency. MorganandClaypool, 2008.

# Optimal speed scaling

Joint work with Adam Wierman and the students of ECE 555

Can we adjust power to computer processor
to optimize delay/power utilization tradeoff?




[1] O. S. Unsal andI. Koren, "System-level power-aware deisgn techniques in real-time systems," Proc. IEEE, vol. 91, no. 7, pp. 1055–1069, 2003.

[2] S. Irani andK. R. Pruhs, "Algorithmic problems in power management, "SIGACT News, vol. 36, no. 2, pp. 63–76, 2005.

[3] S. Kaxirasand M. Martonosi, Computer Architecture Techniques for Power-Efficiency. MorganandClaypool, 2008.

# Optimal speed scaling - queueing model

Abstraction: A controlled queue

$$Q(k+1) - Q(k) = -U(k) + A(k+1)$$

*Processing rate* → $U(k)$ *Job arrivals* → $A(k+1)$

Minimize the average cost:

$$\eta = \lim_{t\to\infty} \mathsf{E}[c(Q(t), U(t))]$$

Quadratic processing cost

$$c(x,u) = x + \tfrac{1}{2}u^2 \qquad \textit{for sake of example}$$

# Optimal speed scaling - fluid model

Abstraction: A controlled fluid model

$$\frac{d}{dt} q(t) = -\zeta + \alpha$$

*Processing rate* → $\zeta$ *Job arrivals* → $\alpha$

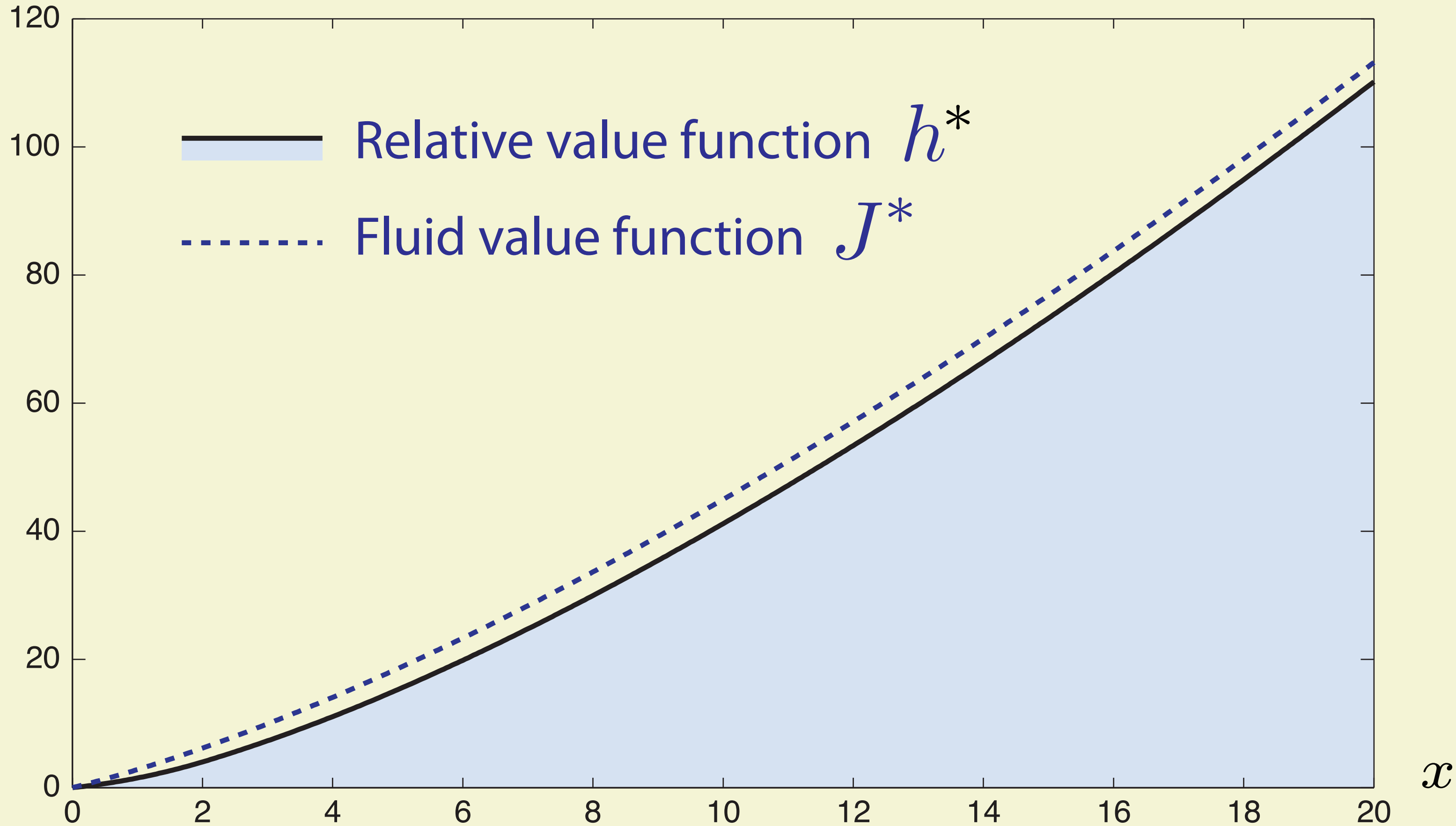
Minimize the total cost (up to draining time):

$$J^*(x) = \min \int_0^{T_0} c(q(t), \zeta(t))\, dt, \qquad q(0) = x$$

$$= \alpha x + \frac{1}{3}\big((2x + \alpha^2)^{3/2} - \alpha^3\big)$$

$$c(x,u) = x + \tfrac{1}{2} u^2$$

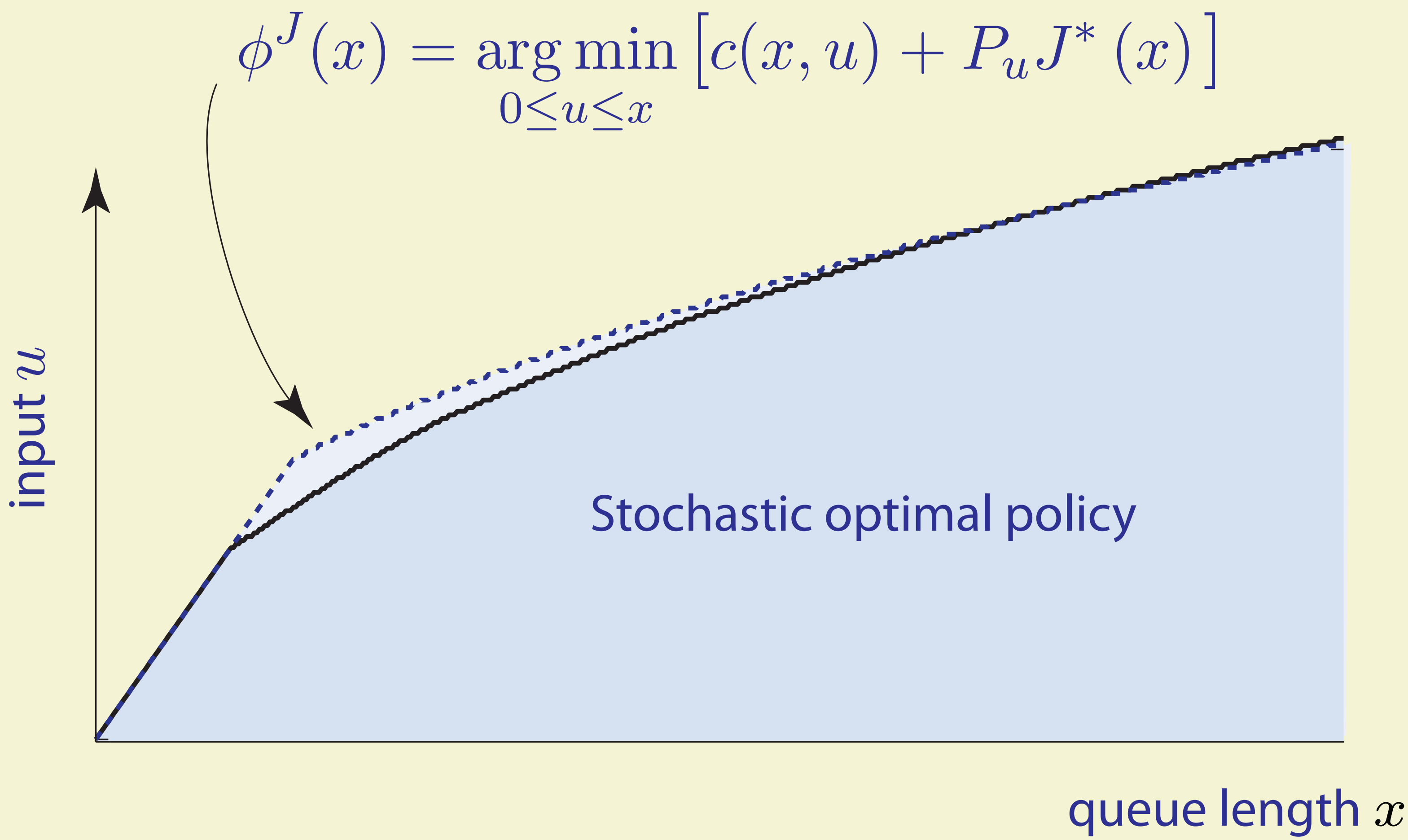
# Optimal speed scaling - value functions




$$J^*(x) = \min \int_0^{T_0} c(q(t), \zeta(t))\, dt, \qquad q(0) = x$$

$$c(x,u) = x + \tfrac{1}{2}u^2$$

# Optimal speed scaling - policies

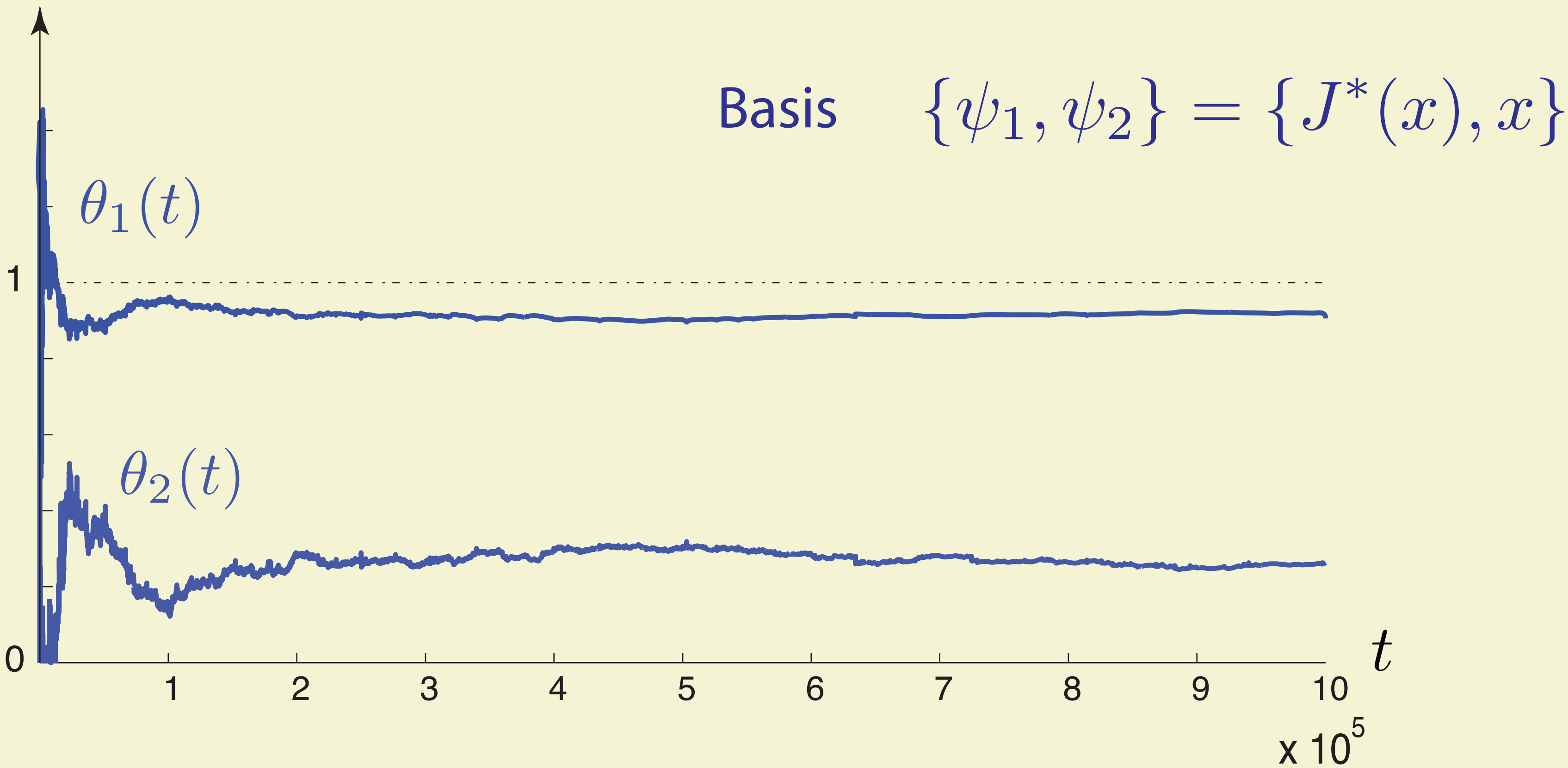



$$J^*(x) = \min \int_0^{T_0} c(q(t), \zeta(t))\, dt, \qquad q(0) = x$$

$$c(x,u) = x + \tfrac{1}{2} u^2$$

# Optimal speed scaling - TD Learning

$$h_\theta(x) = \theta_1 \psi_1(x) + \theta_2 \psi_2(x)$$



# Outline

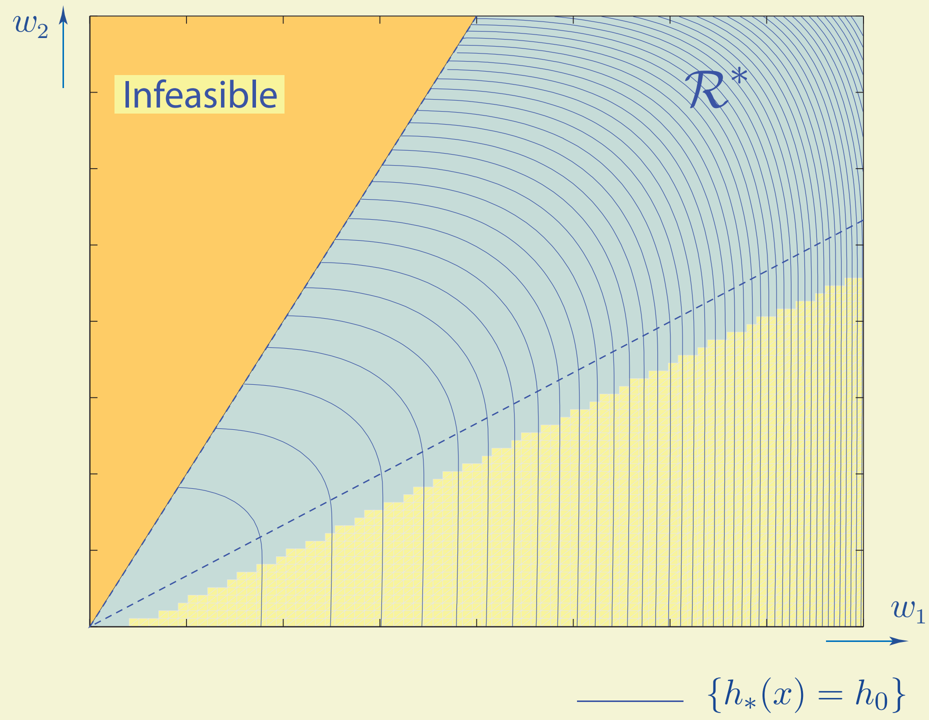
Value function approximation

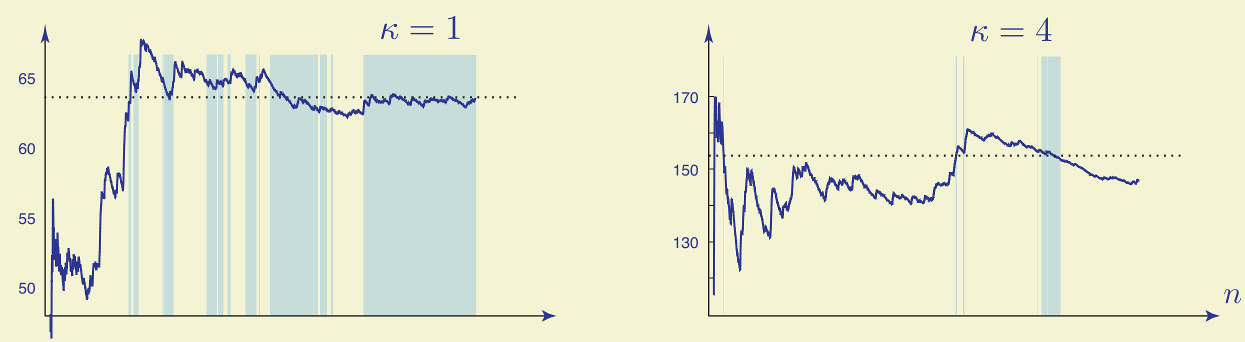
- Performance evaluation
- Perfromance improvement
- Simulation

TD learning

Numerics

**Conclusions**



# Conclusions

Possible to apply adjoint for algorithm synthesis in many settings

Many questions remaining!

- → Variance of Newton and SA recursions
- → Parametizations in networks
- → Complexity: workload relaxations (see CTCN and Henderson and M. 2003)
- → Policy improvement, approximate DP

Veatch 2004
Mannor, Menache, and Shimkin, 2005
Moallemi, Kumar, and Van Roy 2007

# Conclusions

Possible to apply adjoint for algorithm synthesis in many settings

Many questions remaining!

- ⟶ Variance of Newton and SA recursions
- ⟶ Parametizations in networks
- ⟶ Complexity: workload relaxations
- ⟶ Policy improvement, approximate DP