Stochastic Learning and Optimization

- A Sensitivity-Based Approach

Plenary Presentation

2008 IFAC World Congress

July 8, 2008

Xi-Ren Cao

The Hong Kong Uni. of Science & Tech.

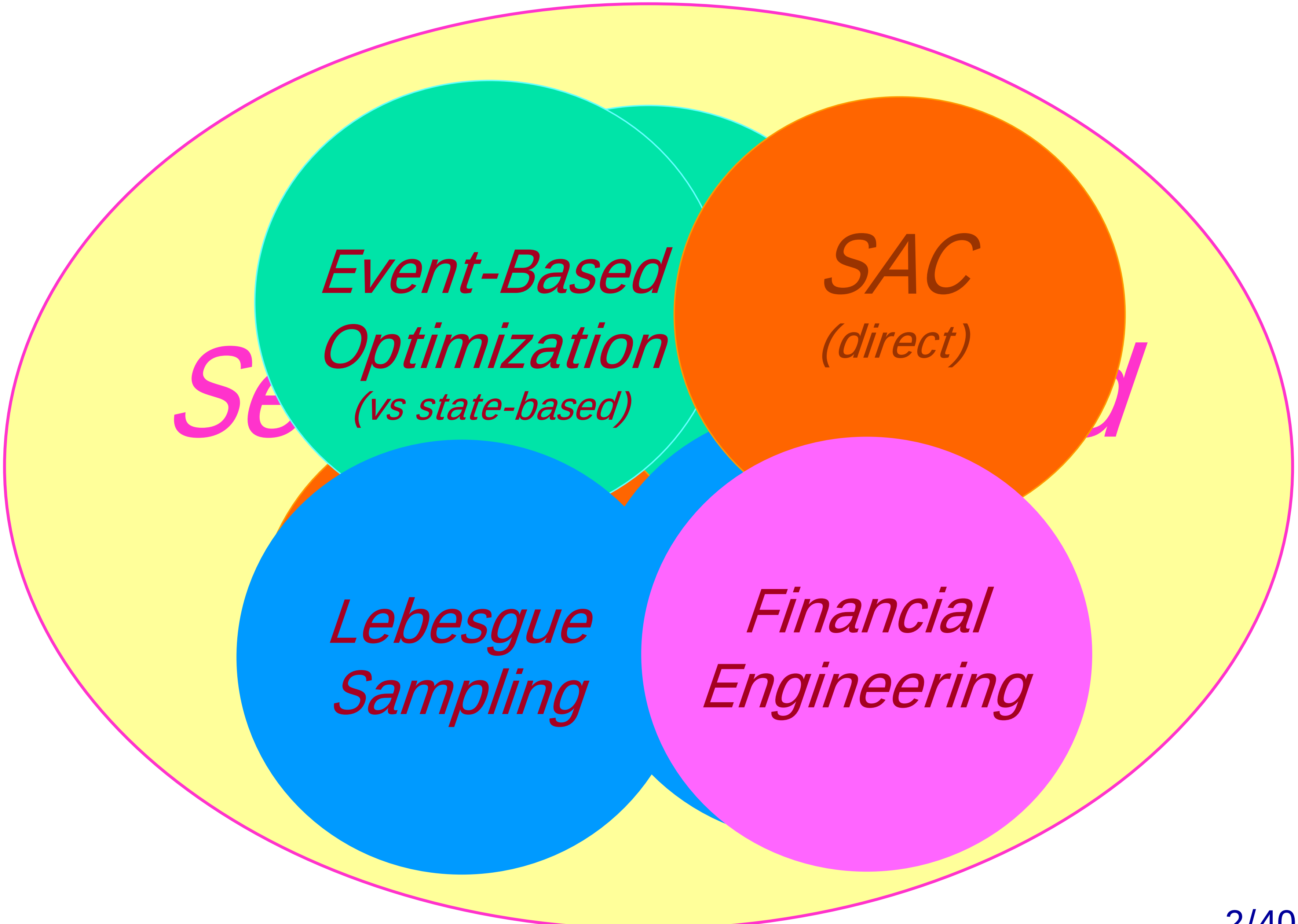
A Unified Framework for
Stochastic Learning and Optimization
(with a sensitivity-based view)

a. Perturbation analysis (PA):
a counterpart of MDPs

b. Markov decision processes (MDPs)
a new and simple approach

c. Overview of reinforcement learning (RL)

d. Event-based Optimization and others



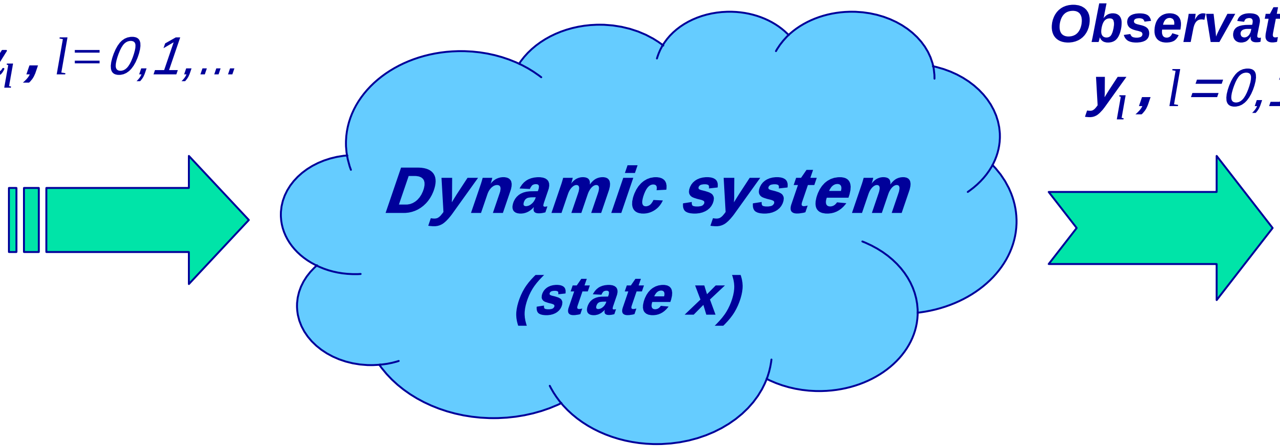
Event-Based
Optimization
(vs state-based)
SAC
(direct)
Lebesgue
Sampling
Financial
Engineering

Optimization Problems

System Performance
η

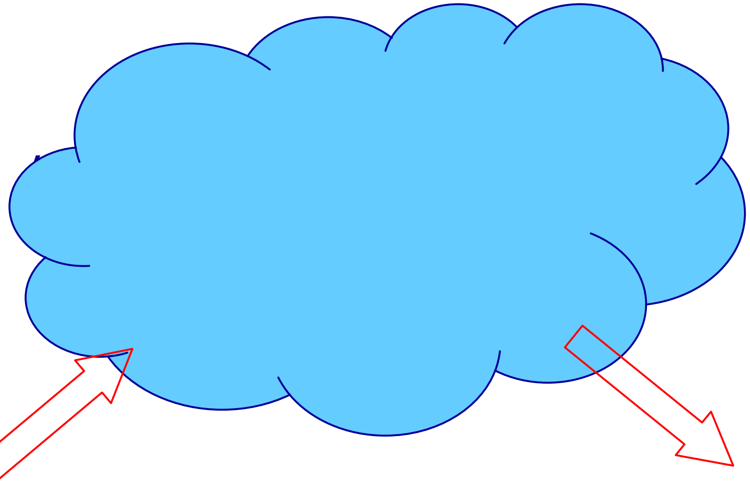
Actions α_l **,** $l=0,1,\ldots$

Observations
$\boldsymbol{y_l}$ **,** $l=0,1,\ldots$



Policy: action= d(information) , $\alpha = d(y)$

Goal – to find a policy that has the best performance



Actions:
service rate μ_n
(state dependent)

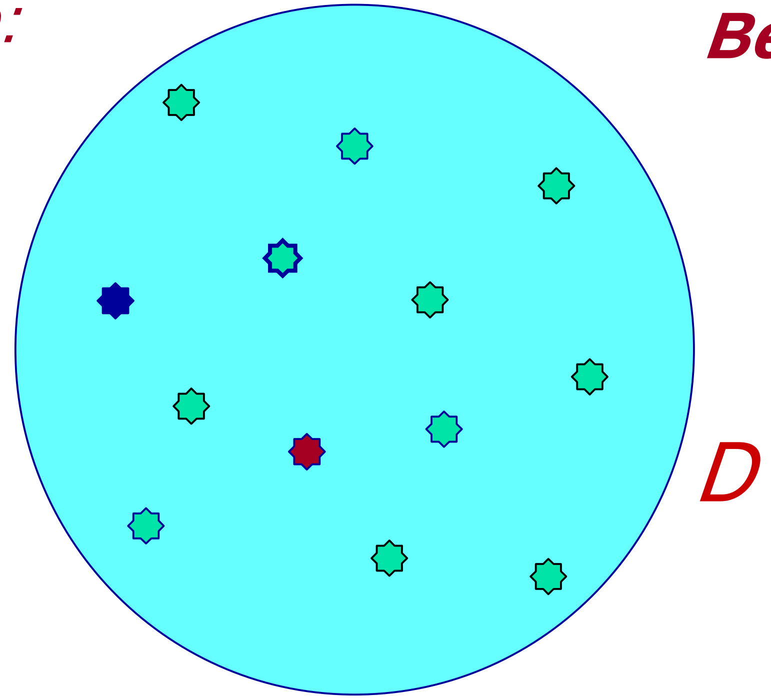
Observations:
number of packets (state)
$n=0,1,\ldots N$

Policy $\mu_n = d(n)$

Performance: average # served/sec - costs

Policy Space:

Best Policy?

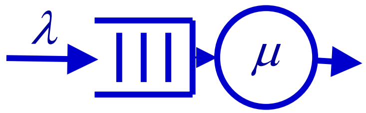


Continuous (with parameters θ) or discrete

- Policy space too large

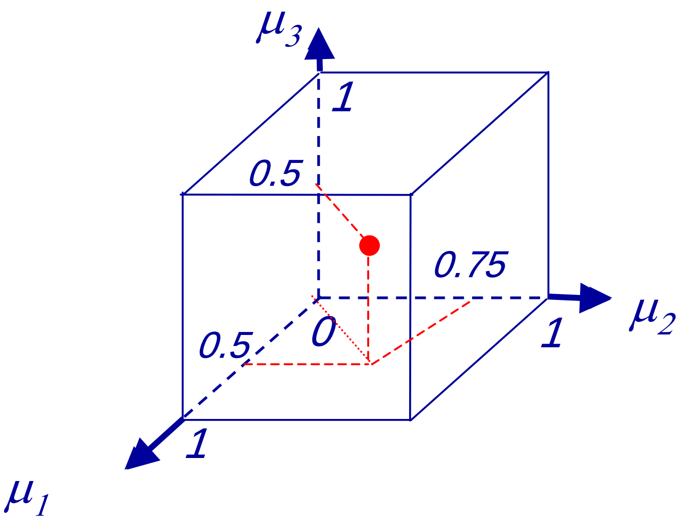
(100 states, 2 actions ➔ $2^{100} = 10^{30}$ policies, 10Gh -> 10^{12} yrs to count)

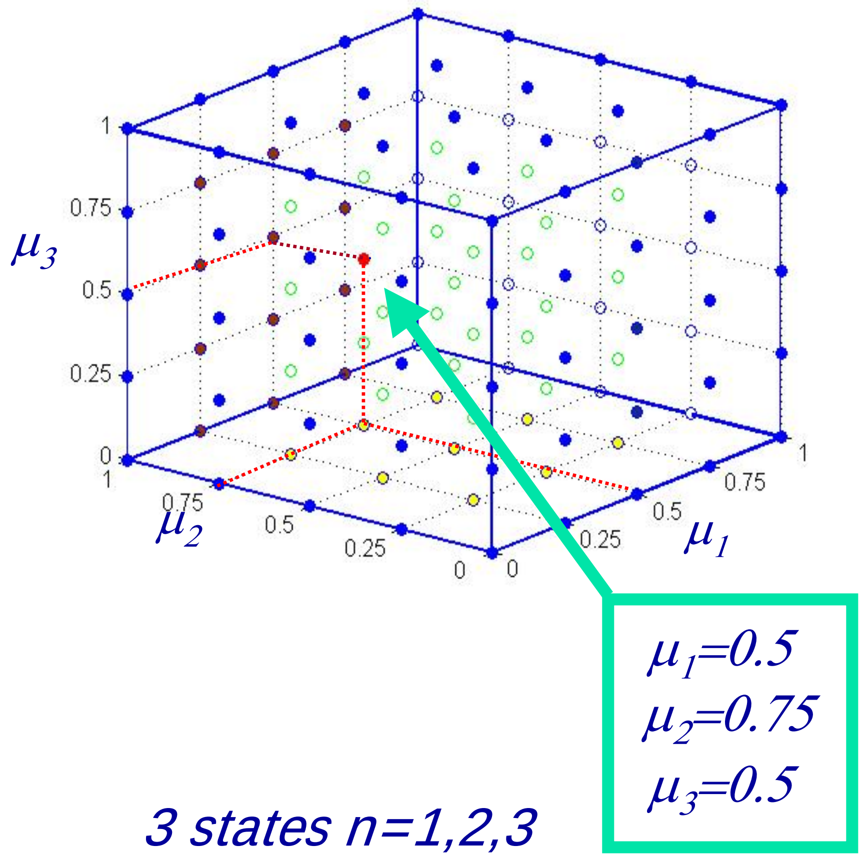
- State space too large and structure unknown



Policy space D
Discrete: grid (5^3)

Policy $\mu_n = d(n)$





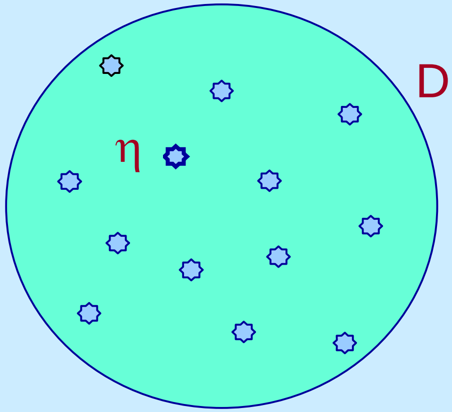
Continuous: $D=[0,1]^3$

3 states n=1,2,3

With no structural information of the system

Search Methods

➔ Evaluate each policy



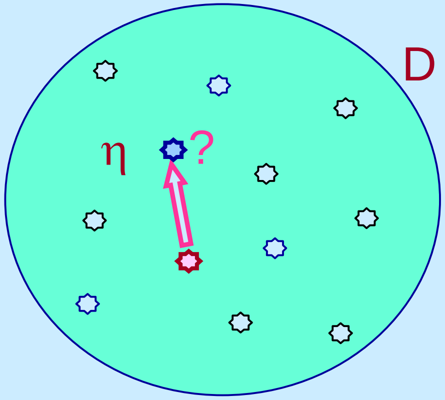
Blind random search

Ordinal optimization

Exploring geometric properties of distribution of η over D

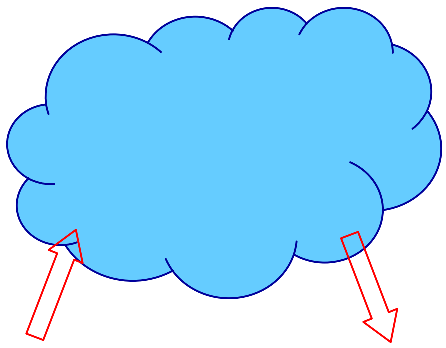
With structural information of the system

Analyzing behavior of one policy
➔ Interpret performance of others

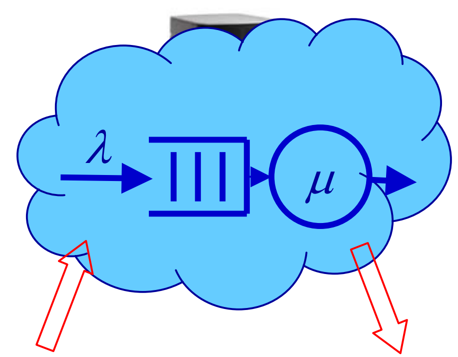


How to obtain as much perf. inf. of other policies as possible?

Black Box



Structure known



Simplicity is Beauty

$$F = ma \qquad E = mc^2 \qquad f \propto \frac{m_1 m_2}{r^2}$$

How about Stochastic Learning & Optimization?

$$\frac{PV}{T} = const \qquad m = \frac{m_0}{\sqrt{1 - v^2 / c^2}}$$

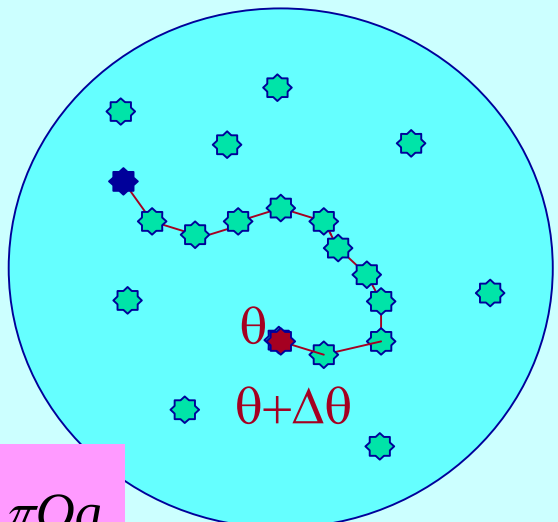
$$\frac{d\eta}{d\theta} = \pi Q g \qquad \eta' - \eta = \pi' Q g$$

$$\nabla \bullet D = \rho_f \qquad \nabla \times E = -\frac{\partial B}{\partial t} \qquad \nabla \bullet B = 0$$

......

With Structural Information

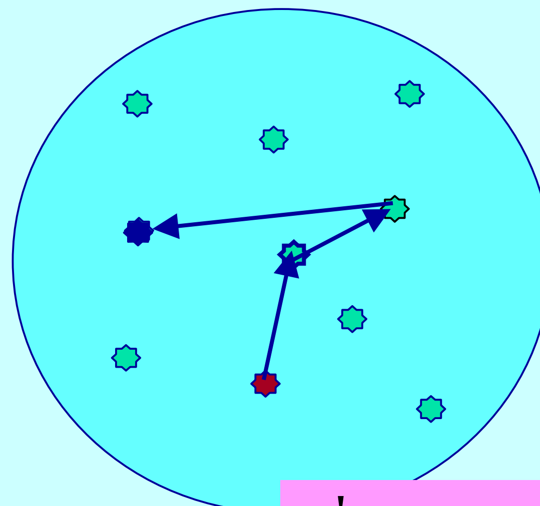
With some knowledge,
studying one policy
➔ neighborhood perf.



$$\frac{d\eta}{d\theta} = \pi Q g$$

Continuous policy spaces

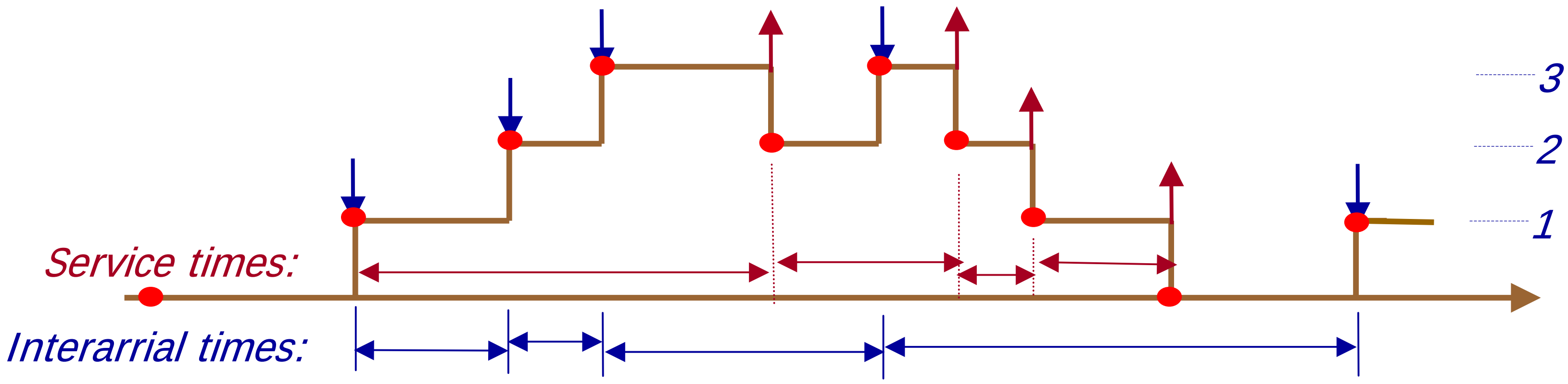
With some knowledge,
studying one policy
➔ find a better policy



$$\eta' - \eta = \pi' Q g$$

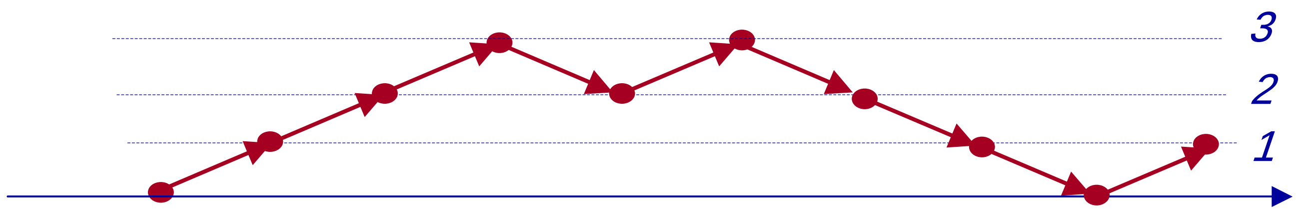
Discrete policy spaces

A Sample Path

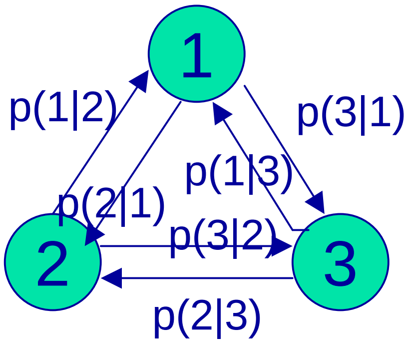


- The dynamic behavior of a system under a policy can be represented by a sample path
- Analyzing a sample path ➔ performance under the policy

  ? ➔ ? Other policies ?
- Discrete time model (embedded Markov chain):



The Markov Model



System dynamics:

- $X = \{X_n, n=1,2,...\}$, X_n in $S = \{1,2,...,M\}$
- Transition Prob. Matrix $P=[p(j|i)]_{i,j=1,..,M}$

System performance:

- Reward function $f=(f(1),\ldots,f(M))^T$
  Performance measure:

$$\eta = \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} f(X_t) = \pi f = \sum_{i\in S}\pi(i) f(i)$$

Steady-state probability:

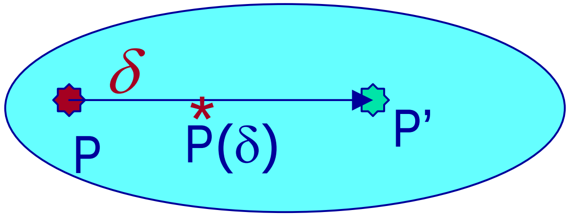
- Steady-state probability:
  $\pi=(\pi(1), \pi(2),..,\pi(M))$.
  $\pi(I-P)=0, \quad \pi e=1$
  I:identity matrix, $e=(1,\ldots,1)^T$

Perturbation Analysis

Perturbation Analysis (PA)

For two Markov chains

P=[p(j|i)], η, π and P'=[p'(j|i)], η', π', (Q=P'-P)



$$P(\delta) = (1-\delta)P + \delta P' \qquad \delta \in [0,1]$$

Performance gradient:

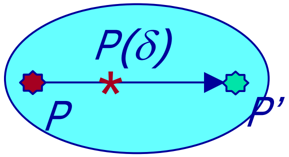
$$\frac{d\eta(\delta)}{d\delta} = \pi Q g = \pi P' g - \pi P g$$

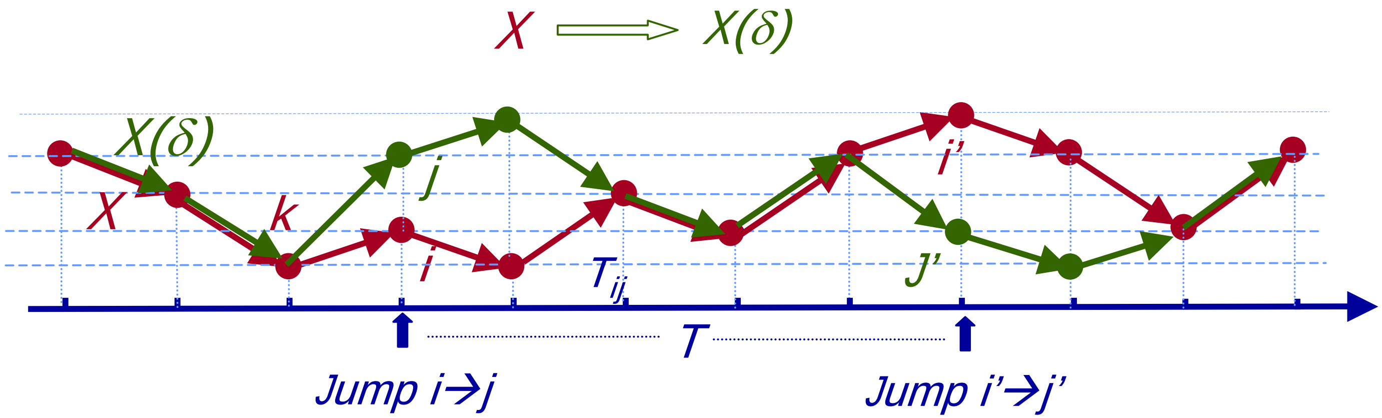
Poisson equation:

$$(I-P)g + \eta e = f$$

X: *sample path with P and performance η*

$X(\delta)$: *sample path with $P(\delta) = P+\delta Q$, $Q=P'-P$ and $\eta(\delta)$*



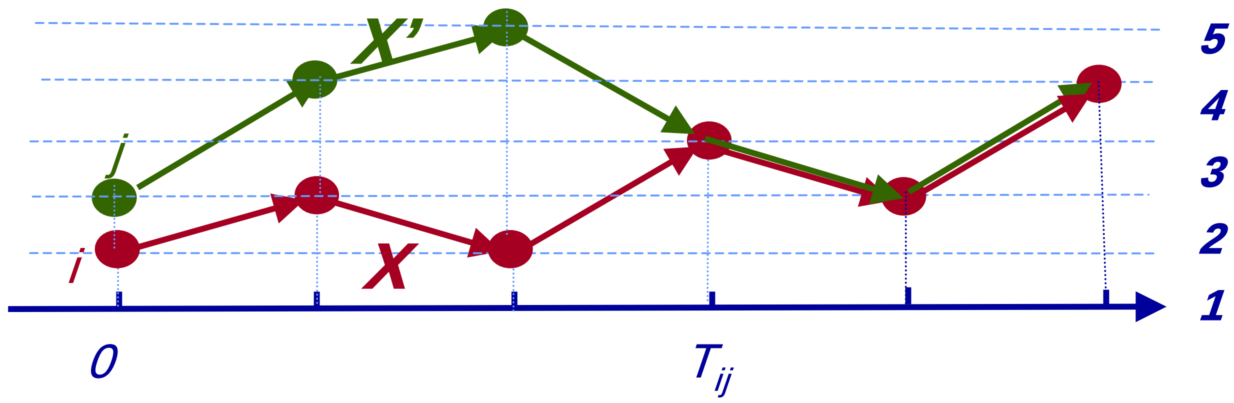


δ is very small ⟹ changes in sample path are also very small

Changes are represented by many jumps

Performance

$$\eta = \pi f = \lim_{N \to \infty} \frac{1}{N} \sum_{n=0}^{N-1} f(X_n)$$



Define performance potential of state i:

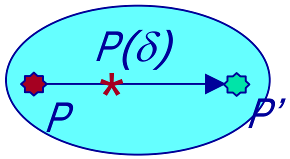
$$g(i) = \lim_{N\to\infty} E\{\sum_{n=0}^{N}[f(X_n)-\eta] \mid X_0 = i\}.$$

➔ *Potential contribution of state i to the performance* $\eta = \lim_{N\to\infty}\frac{1}{N}E\{\sum_{n=0}^{N-1} f(X_n)\}$

⟹ *Poisson equation:* $(I-P)g + \eta e = f.$ $g = (g(1),\ldots,g(M))^T$

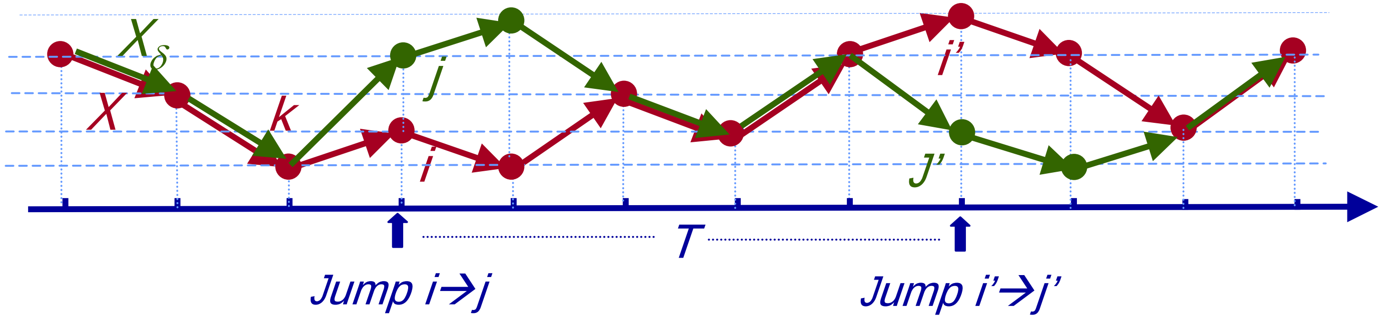
Effect of a jump from i to j on performance:

$$\gamma(i,j) = g(j) - g(i)$$



$P \longrightarrow P(\delta)$

$X \longrightarrow X(\delta)$



Adding the effects of all the jumps we obtain $\eta(\delta)-\eta$

⟶ *Performance gradient:*

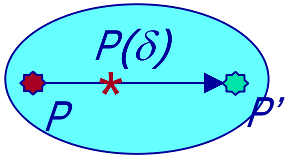
$$\frac{d\eta(\delta)}{d\delta} = \pi Q g, \qquad Q = P' - P.$$

Markov Decision Processes
- Policy Iteration

Two Sensitivity Formulas

Two Markov chains P, η, π
P', η', π';

with $Q=P'-P$



Continuous policy space

Performance gradient formula:

$$\frac{d\eta(\delta)}{d\delta} = \pi Q g, \qquad Q = P' - P.$$

Gradient-based optimization

Discrete policy space

Similarly, we can construct

Performance difference formula:

$$\eta' - \eta = \pi' Q g. \qquad Q = P' - P.$$

Policy iteration

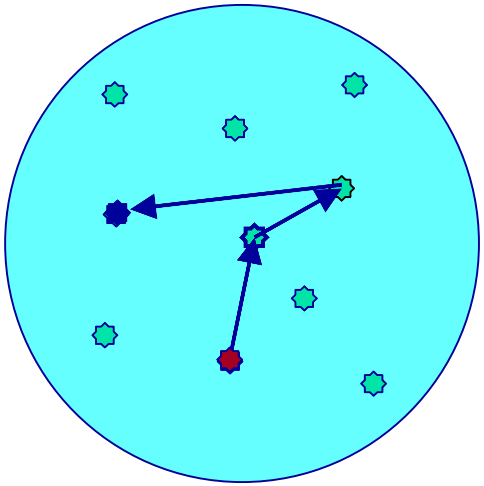
Policy Iteration

Perf. diff. $$\eta' - \eta = \pi' Qg = \pi'(P' - P)g$$

1. $\pi' > 0$ ➔ $\boxed{\eta' > \eta \text{ if } P'g > Pg}$

2. Policy iteration:
   At any state find a policy P' with
   P'g>Pg

3. Improve performance iteratively,
   Stop when no improvement can
   be made



More on Policy Iteration

Performance criteria:

- Average performance $\eta = \pi f$
- Discounted performance

$$\eta_i = E\{\sum_{k=0}^{\infty} \beta^n f(X_k) \mid X_0 = i\}$$

- Bias g:

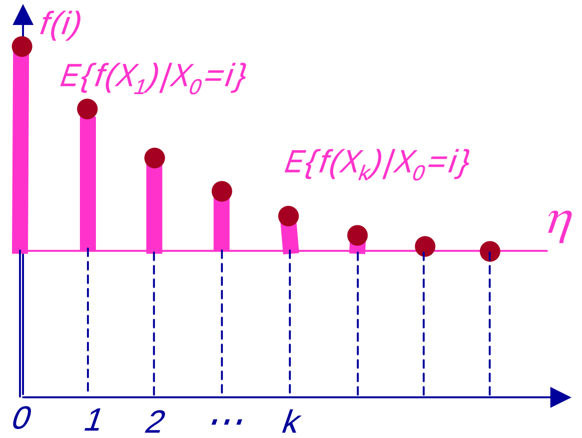
$$g(i) = E\{\sum_{k=0}^{\infty} [f(X_k) - \eta] \mid X_0 = i\}$$

- Bias of bias (2nd order), g_2:

$$g_2(i) = E\{\sum_{k=0}^{\infty} [g(X_k) \mid X_0 = i\} \qquad \pi g = 0$$

- Bias of (n-1)th bias (nth order), g_n:

$$g_n(i) = E\{\sum_{k=0}^{\infty} [g_{n-1}(X_k) \mid X_0 = i\} \qquad \pi g_{n-1} = 0$$



Bias measures transient behavior

Perf./Bias Difference Formulas

Policy Iteration

Two policies P' : π' , η', g', g_2'... and P : π, η, g, g_2,...

$$\eta' - \eta = P'^{*}[(f' + P'g) - (f + Pg)] + [P'^{*} - I]\eta, \qquad P^{*} = \lim_{N\to\infty}\frac{1}{N}\sum_{n=0}^{N-1}P^{n}.$$

If $\eta' = \eta$ *then*

$$g' - g = P'^{*}[P' - P]g_2 + [I - P' + P'^{*}]^{-1}[(f' + P'g) - (f + Pg)].$$

If $g_n' = g_n \quad n = 1,2,\ldots$ *then*

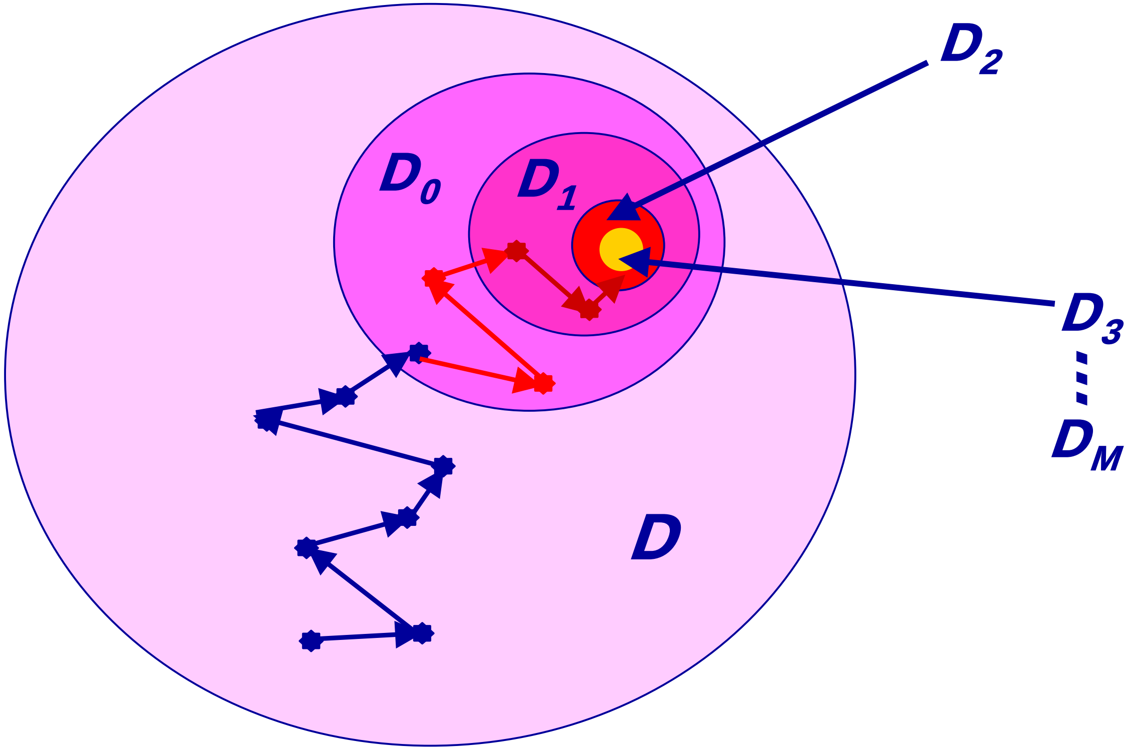
$$g_{n+1}' - g_{n+1} = P'^{*}[P' - P]g_{n+2} + [I - P' + P'^{*}]^{-1}(P' - P)g_{n+1}.$$

- *Policy iteration for optimal n-bias*
- *Optimality equations for n-bias optimization.*

Mutli-Chain MDPs

Perf./ Bias/ Blackwell Optimization

With perf. difference formulas, we can derive a simple, intuitive approach without discounting



D: *Policy space*

D_0: *Perf. optimal policies*

D_1: *(1st) Bias optimal policies*

D_2: *2nd Bias optimal policies*

...... **D_M:** *Blackwell optimal policies*

PA

MDP

With some knowledge

With some knowledge,

Observations:

- *Do not need to evaluate every policy*
  ~~*(large policy space)*~~
- *State space is too large ➔*
  difficult to evaluate each policy
  ➔ estimate g, Pg, or πQg

$$\frac{d\eta}{d\theta} = \pi Q g$$

$$\eta' - \eta = \pi' Q g$$

Continuous policy spaces

Discrete policy spaces

Reinforcement Learning

- *P too large, or not completely known*
- *Learning: estimate from sample path*

PA: $\frac{d\eta(\delta)}{d\delta} = \pi Q g = \pi P' g - \pi P g$ MDPs: $\eta' - \eta = \pi' Q g = \pi'(P' - P) g$

- *Estimating g:*

$$g(i) = E\{\sum_{k=0}^{\infty}[f(X_k) - \eta] \mid X_0 = i\} = E\{[f(i) - \eta] + g(X_1) \mid X_0 = i\}.$$

Monte Carlo: *Average of $\Sigma[f(X_k) - \eta]$*

Stochastic approximation

$$g(X_k) := g(X_k) + \alpha_k\{f(X_k) - \eta + g(X_{k+1}) - g(X_k)\},$$

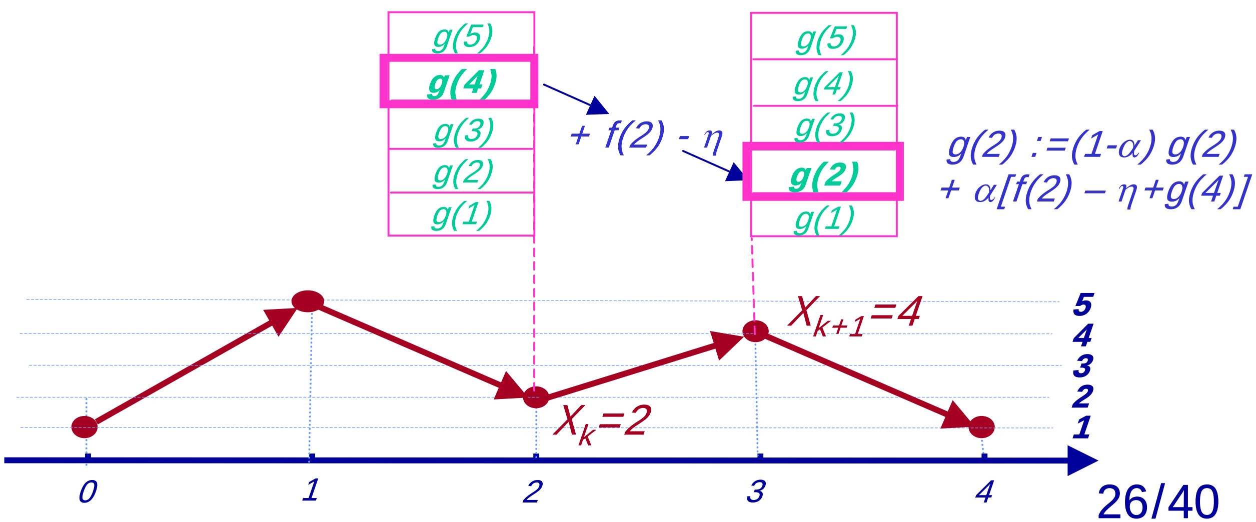
$$\delta_k = f(X_k) - \eta + g(X_{k+1}) - g(X_k)$$ - *Temporal difference (TD)*

- *Stepsize* α_k

$$\alpha_k > 0,$$

$$\sum_{k=0}^{\infty} \alpha_k = \infty,$$

$$\sum_{k=0}^{\infty} \alpha_k^2 < \infty.$$



PA: $\frac{d\eta(\delta)}{d\delta} = \pi Qg = \pi P' g - \pi Pg$ MDPs: $\eta' - \eta = \pi' Qg = \pi'(P' - P)g$

- *Estimating Pg, (Q-factors)*

$$Q(i,\alpha) = \sum_{j=1}^{M} p^{\alpha}(j\,|\,i)g(i) + f(i,\alpha) - \eta.$$

Similar Temporal Dfference (TD) algorithms can be developed

- *Estimating πQg directly* $Q = P' - P = \Delta P$

$$\frac{d\eta(\delta)}{d\delta} = \pi(\Delta P)g$$

$$= E\{\frac{\Delta p(X_{k+1}\,|\,X_k)}{p(X_{k+1}\,|\,X_k)} g(X_{k+1})\}.$$

$$\Rightarrow \quad \frac{d\eta(\delta)}{d\delta} = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N} \{\frac{\Delta p(X_{n+1}\,|\,X_n)}{p(X_{n+1}\,|\,X_n)} \hat{g}(X_{n+1}, X_{n+2}, \ldots)\},$$

with $E\{\hat{g}(X_{n+1}, X_{n+2}, \ldots)\} = g(X_n).$

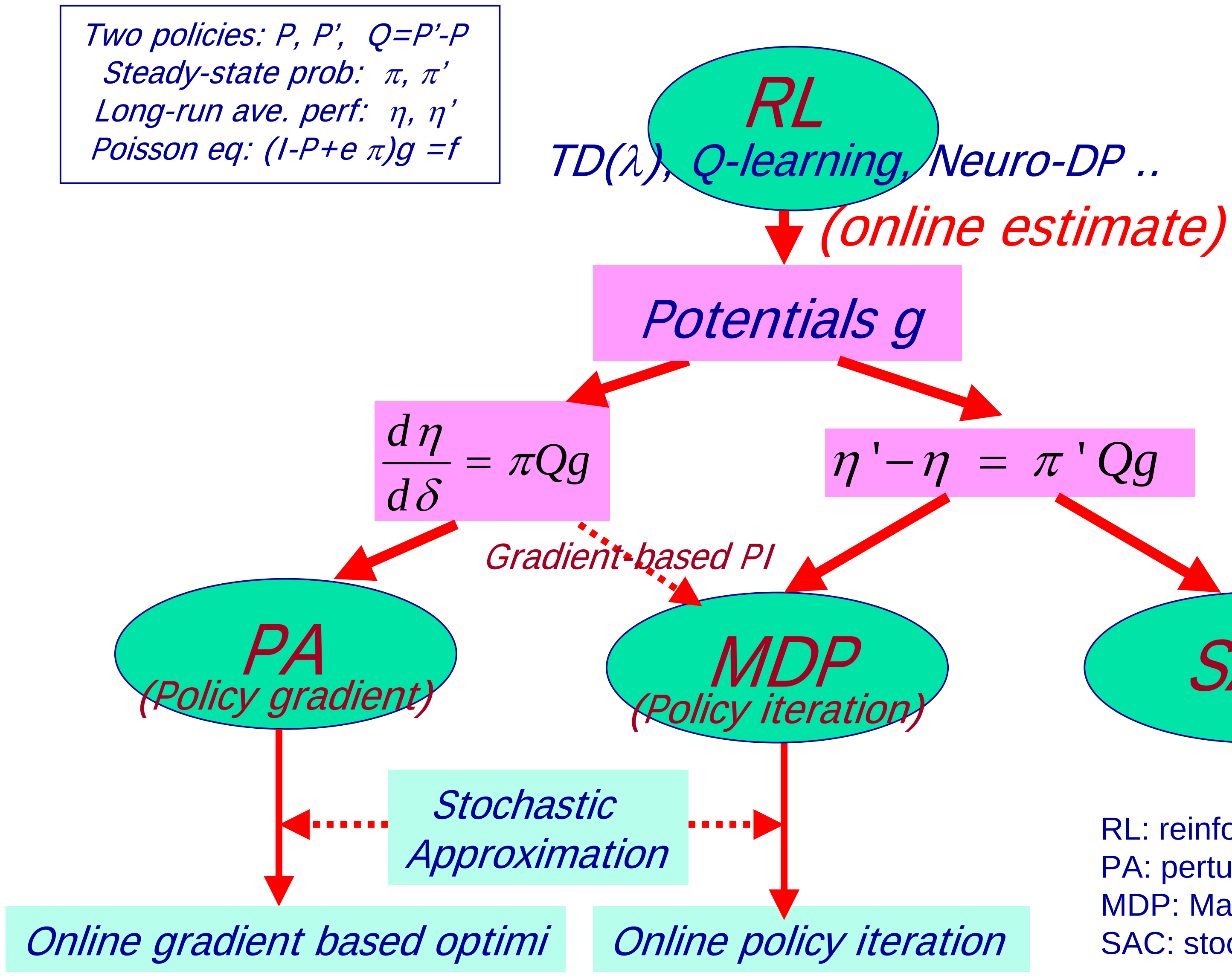
Policy Iteration Based Learning and Optimization

Analytical (P,f known)	Learn g(i) (No matrix inversion, etc)		Learn Q(i,α) (P completely unknown)	
Policy Iteration	Monte Carlo		Monte Carlo	
	Long run accurate est. + PI	Short run noised est. + SA + GPI	Long run accurate est. + PI	Short run noised est. + SA + GPI (to be done)
Solving Poisson Eq. or by numerical methods for g	Temporal Difference		Temporal Difference	
	Long run accurate est. + PI	Short run noised est. + SA + GPI	Long run accurate est. + PI	Short run noised est. + SA + GPI (SARSA)

PA-Gradient Based Learning and Optimization

<table>
<tr><td>Analytical
(P,f known)</td><td>Learn g(i)</td><td>Learn $\frac{d\eta}{d\theta}$ directly</td><td>Find a zero of $\frac{d\eta}{d\theta}$</td></tr>
<tr><td rowspan="5">Perf.
Derivative
Formula
(PDF)
+
Gradient
Methods
(GM)</td><td colspan="2">Monte Carlo</td><td rowspan="3">Updates every regenerative period:

Updates every transition:</td></tr>
<tr><td>Long run
accurate est.
+ PDF+GM</td><td>Long run
accurate est.
+ GM</td></tr>
<tr><td colspan="2">Temporal Difference</td></tr>
<tr><td>Long run
accurate est.
+ PDF+GM</td><td>Long run
accurate est
+ GM</td><td>Short run
noised est.
+TD</td></tr>
</table>



RL: reinforcement learning
PA: perturbation analysis
MDP: Markov decision proc.
SAC: stochastic adaptive cont.

A Map of the L&O World

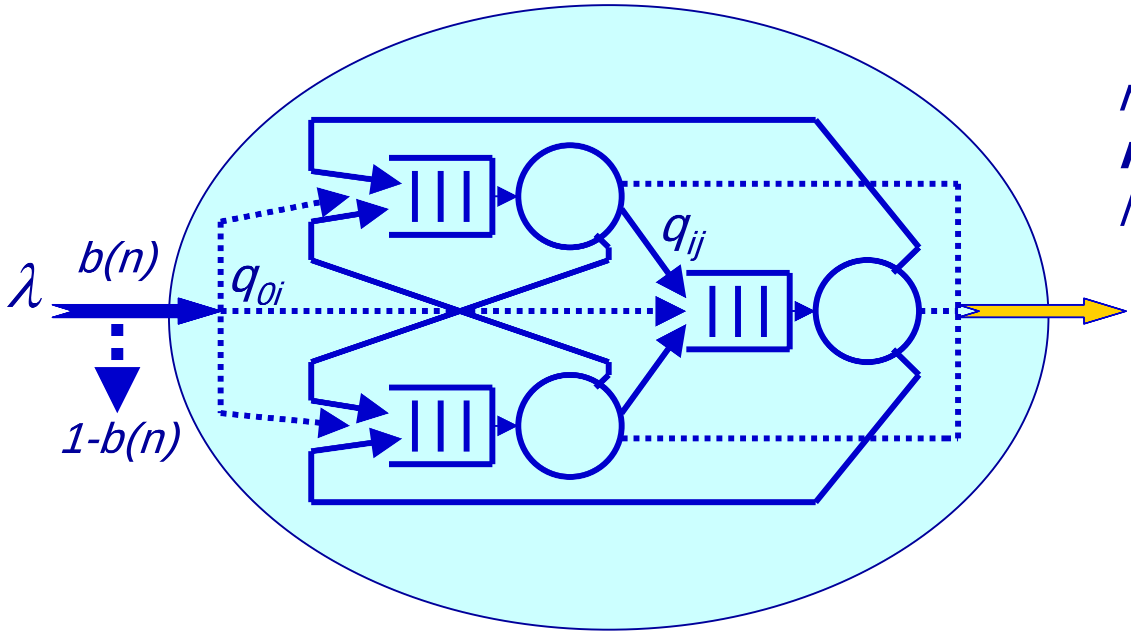
Event-Based Optimization
- New directions

Limitations of State-Based Model

1. *Curse of dimensionality*
2. *State based policies may not be the best*
3. *Special features not captured*

→ *Event-Based Formulation*

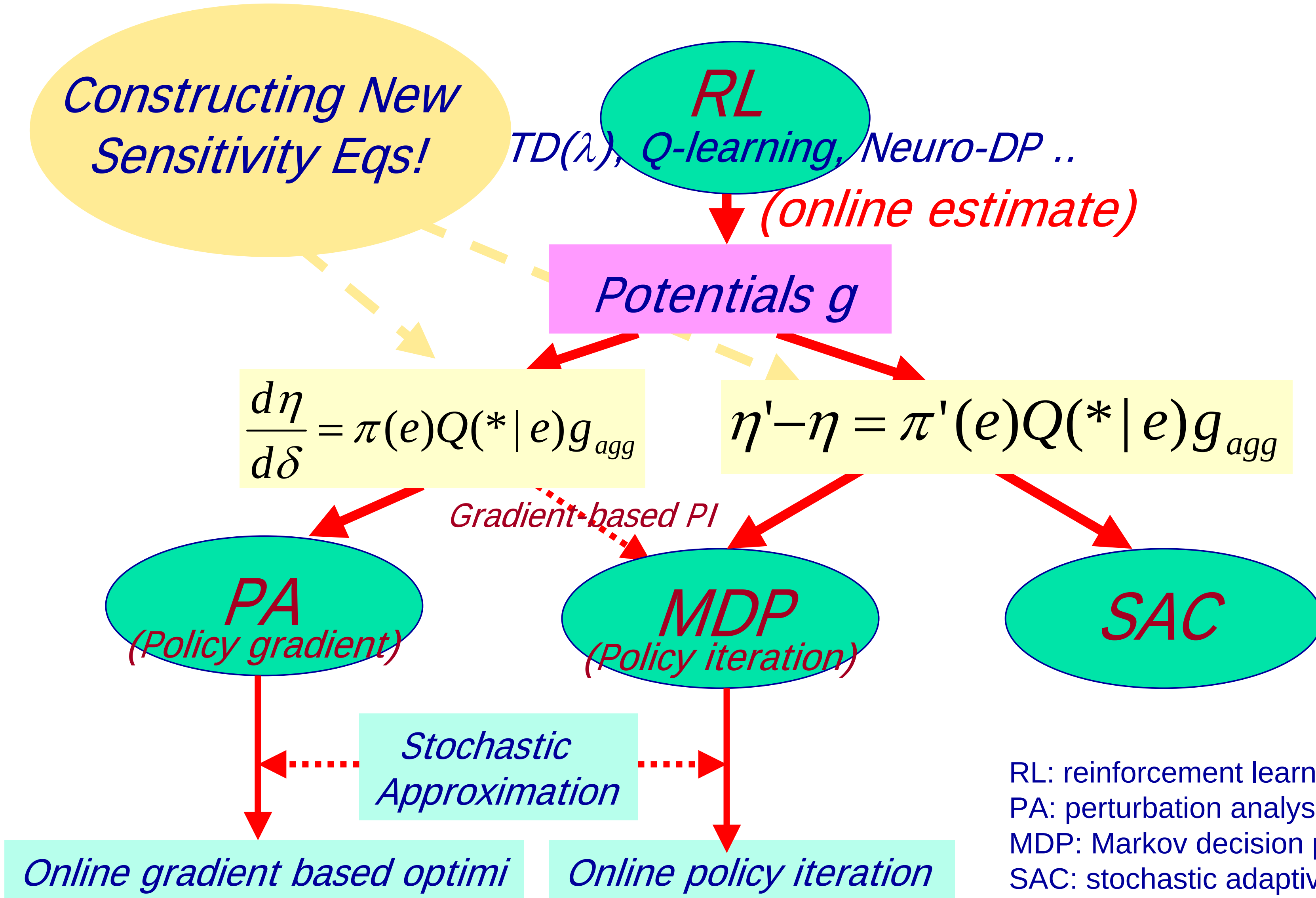
Admission Control in Communication



n: population
No. of all cust. in net
n_i: No. of cust. at svr i
$\mathbf{n}=(n_1,\ldots,n_M)$: state
N: Capacity

How do we choose the admission probability *b(n)?*

Event: A customer arrives finding a population n



RL: reinforcement learning
PA: perturbation analysis
MDP: Markov decision proc.
SAC: stochastic adaptive cont.

Advantages of the Event-Based Approach

1. *# of aggregated potentials d(n): N*

   may be linear in system

2. *Actions at different states are correlated*

   standard MDPs do not apply

3. *Special features captured by events*

   action depends on future information

4. *May have better performance*

5. *Opens up a new direction*

   to many engineering problems

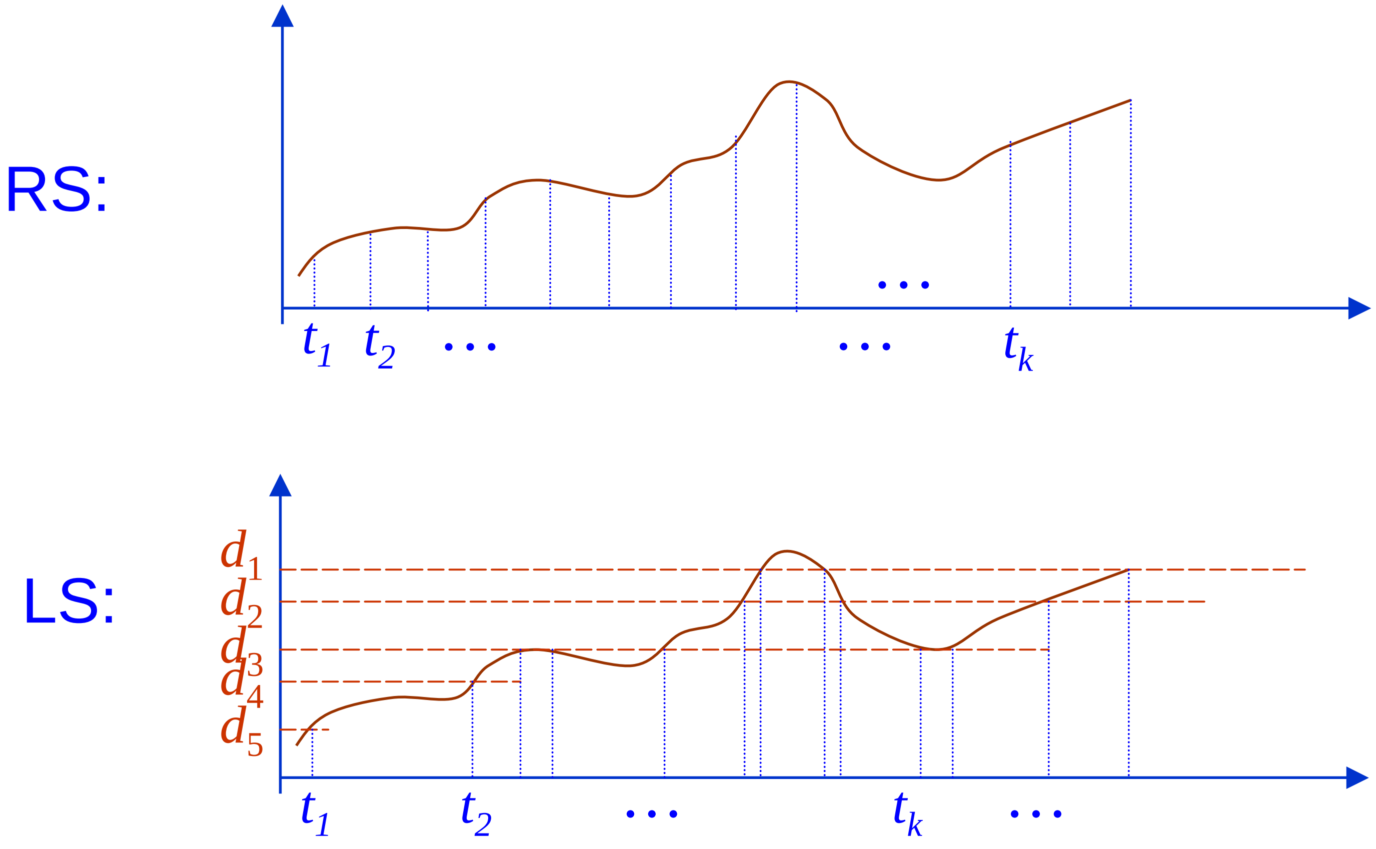
   POMDPs: observation y as event

   hierarchical control: mode change as event

   network of networks: transitions among subnets as events

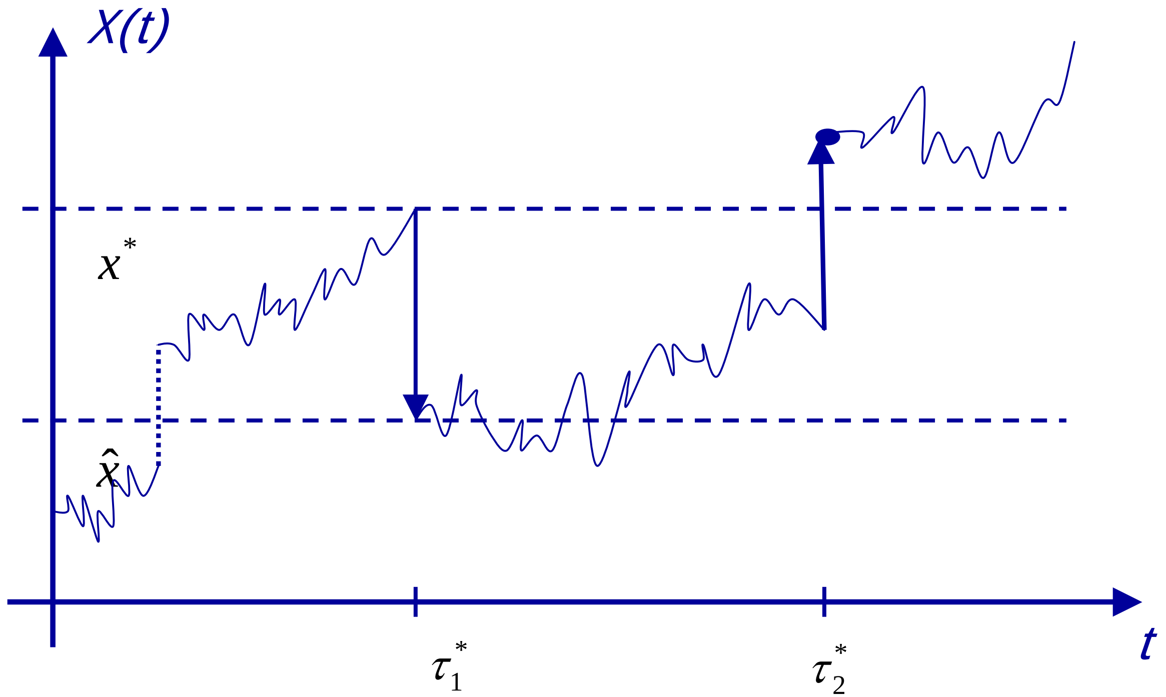
   Lebesgue Sampling

Riemann Sampling vs. Lebesgue Sampling



Sample the system whenever the signal reaches a certain prespecified level, and control is added then.

A Model for Stock Price or Financial Assess

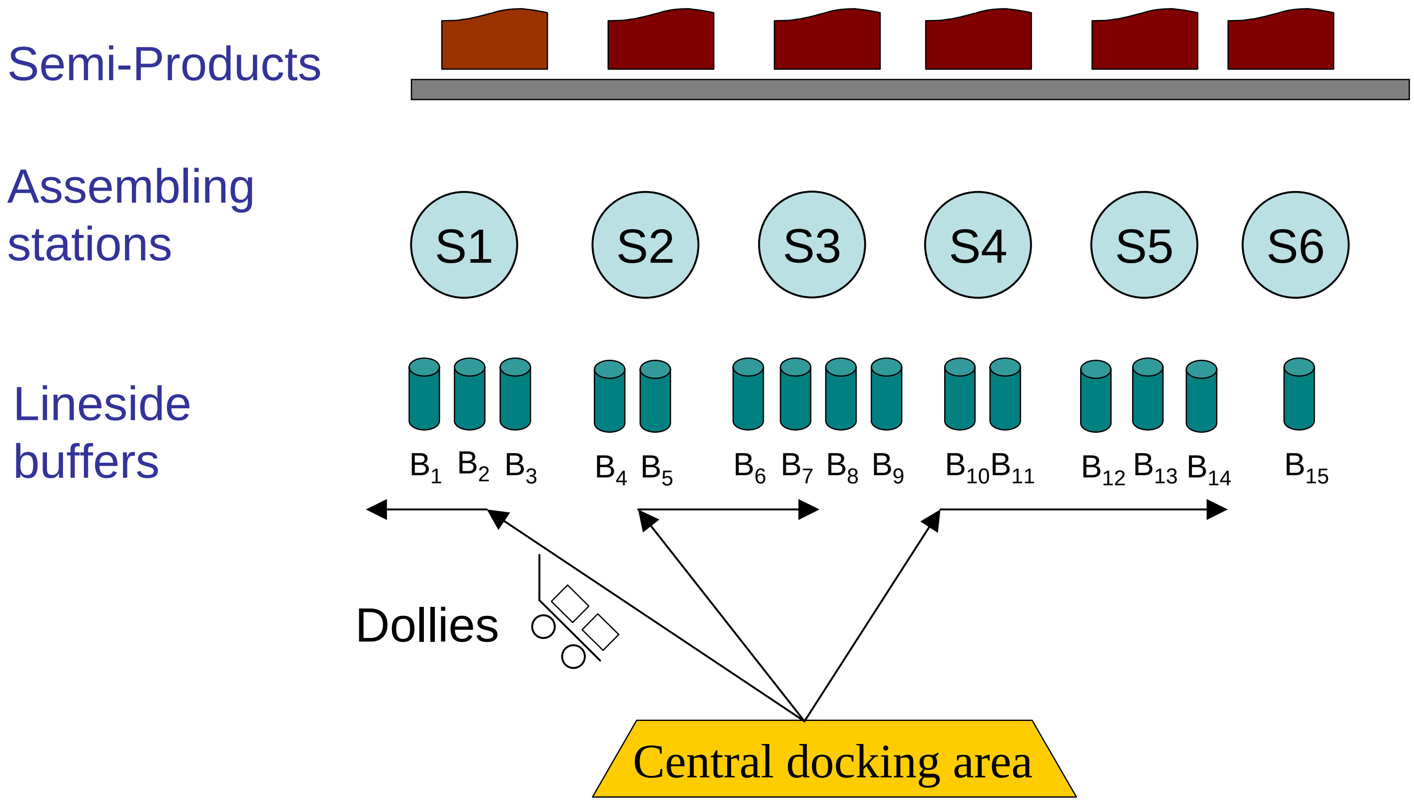


$$dX(t) = b(t, X(t))dt + \sigma(t, X(t))dw(t) + \int \gamma(t, X(t-), z)N(dt, dz).$$

w(t): Brownian motion;   N(dt,dz): Poisson random measure

X(t): Ito-Levy process

A Material Handling System for an Assembly Line



Event-based approach leads to 6-10% performance improvement

Sensitivity-Based View of Optimization

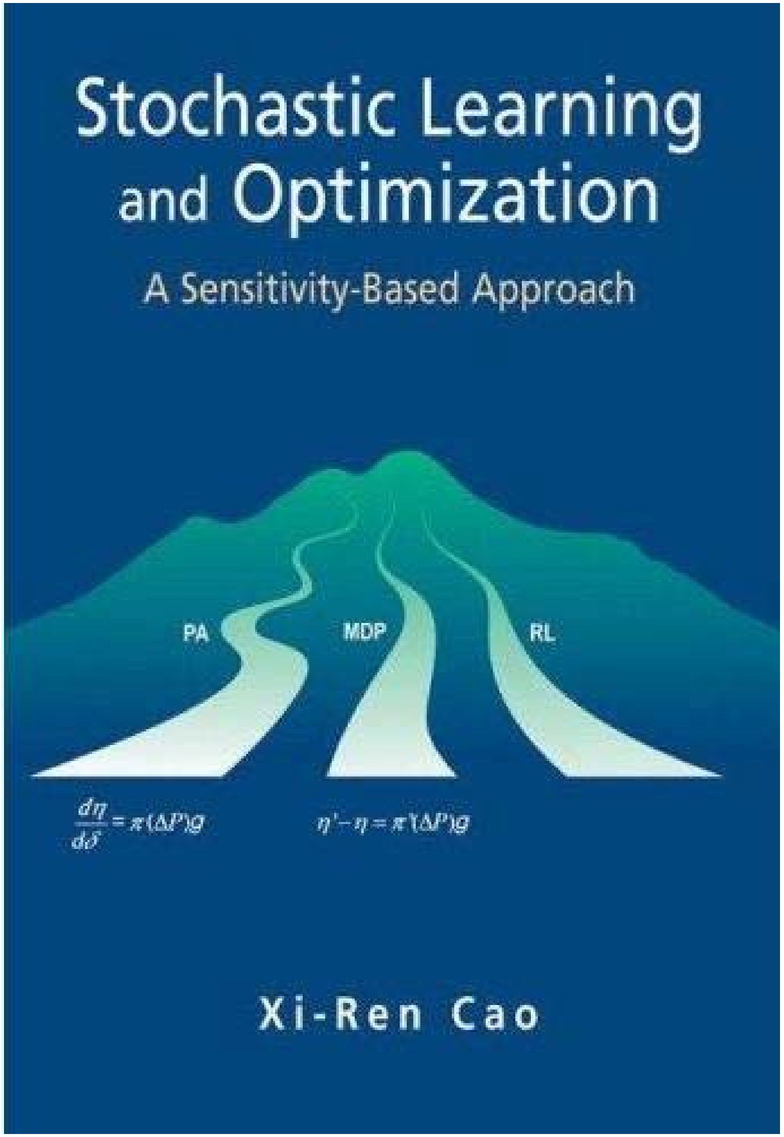
1. *A map of the learning and optimization world:*
   - *Results in Different areas can be obtained / explained from two sensitivity equations*
   - *Simple and complete derivation for MDPs*

2. *Extension to event-based optimization*
   - *Policy iteration, perturbation analysis reinforcement learning, time aggregation….*
   - *Lebesgue sampling, sensor networks, POMDPs, hierarchical control*

……

THANKS !

Questions?



Xi-Ren Cao:

Stochastic Learning and Optimization - A Sensitivity Based Approach

9 Chapters, 566 pages
119 Figures, 27 Tables,
212 homework problems

Springer
October 2007