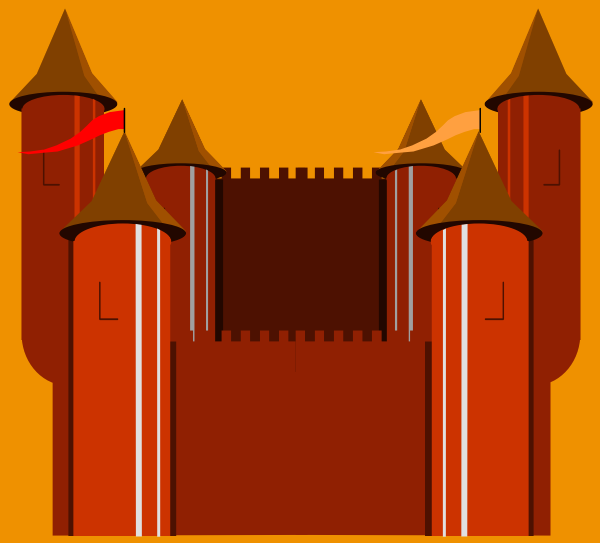
Tutorial: A Unified Framework for Optimization under Uncertainty

Enterprisewide Optimization Group
Carnegie Mellon University

September 11, 2018

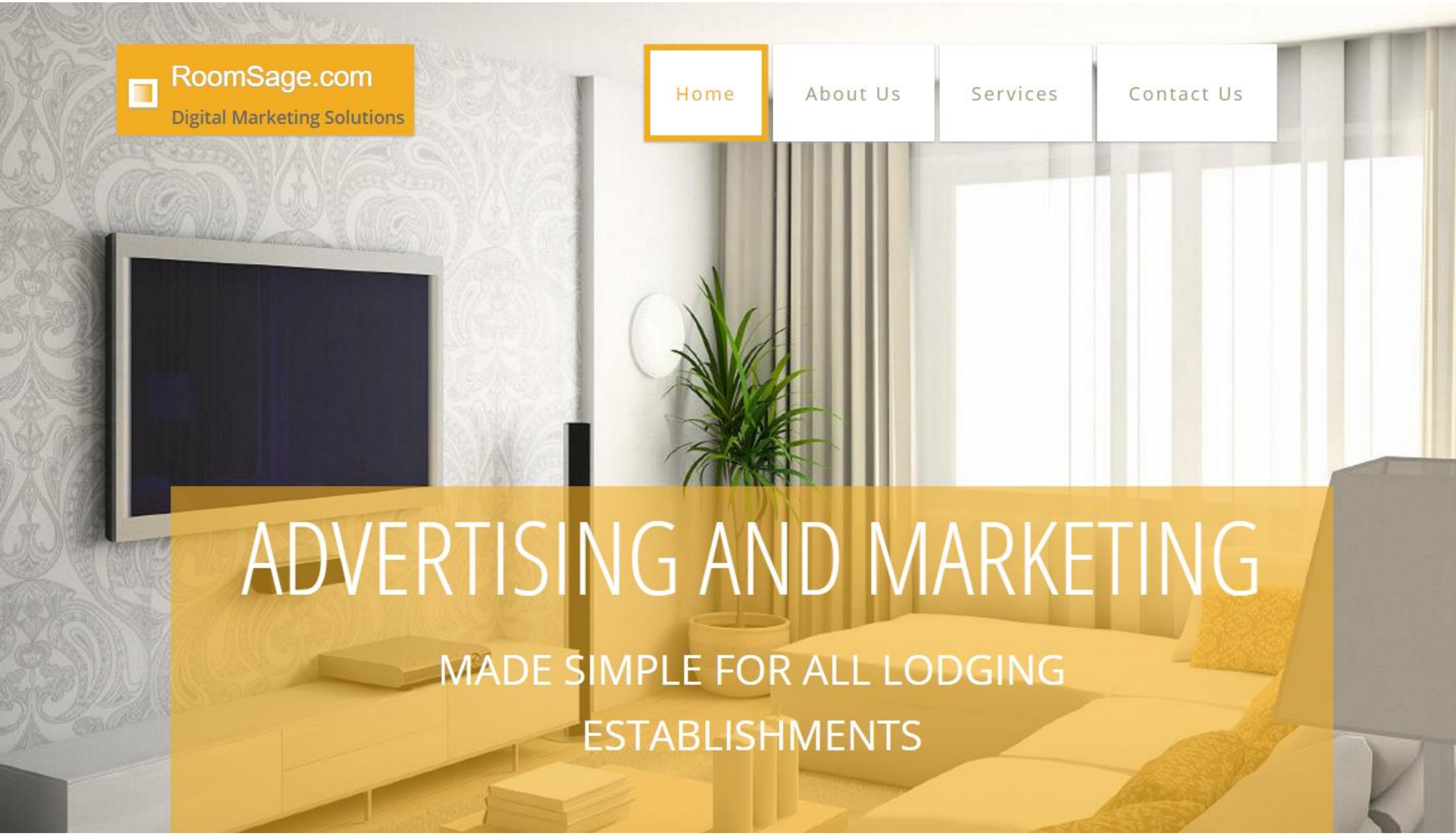


Warren B. Powell

Princeton University
Department of Operations Research and Financial Engineering

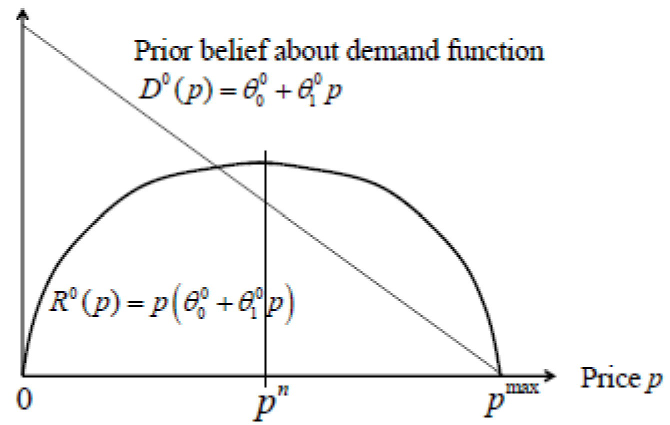
Ad-click bidding

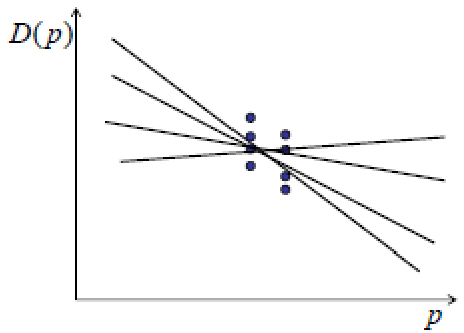
- Roomsage.com



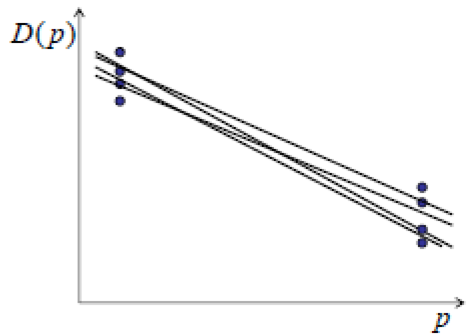
Revenue management

- Earning vs. learning
  - » You want to maximize revenues, but you do not know how demand responds to price.





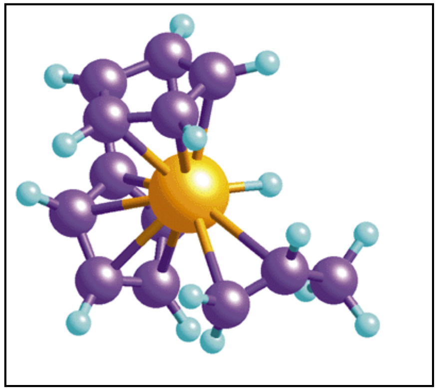
You earn the most with prices near the middle, but you do not learn anything.



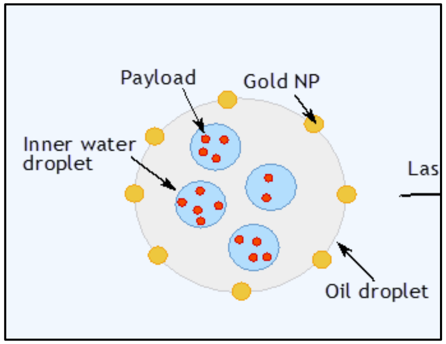
You learn the most by sampling endpoints, but then you do not earn anything.

Learning problems

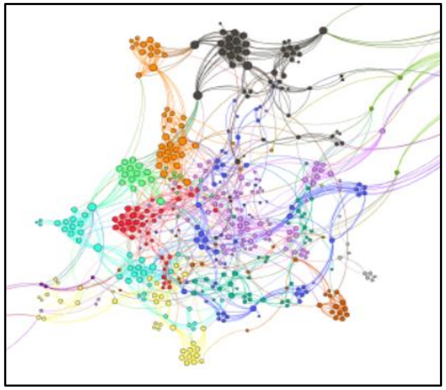
- Health sciences
  - » Sequential design of experiments for drug discovery



  - » Drug delivery – Optimizing the design of protective membranes to control drug release

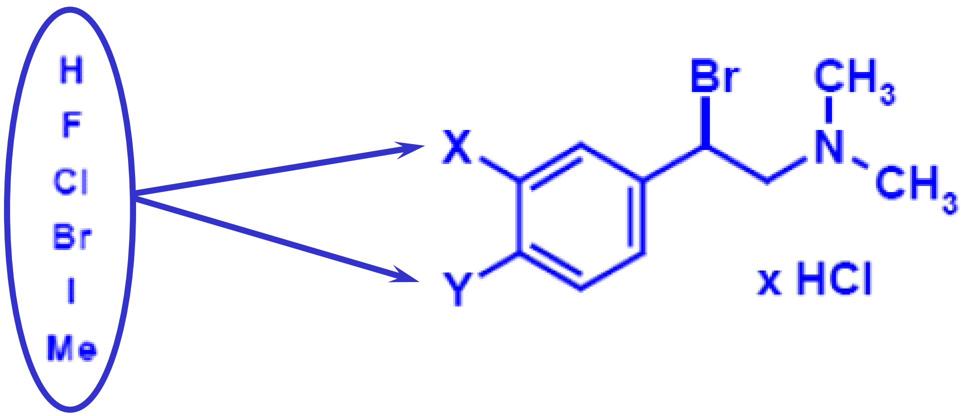


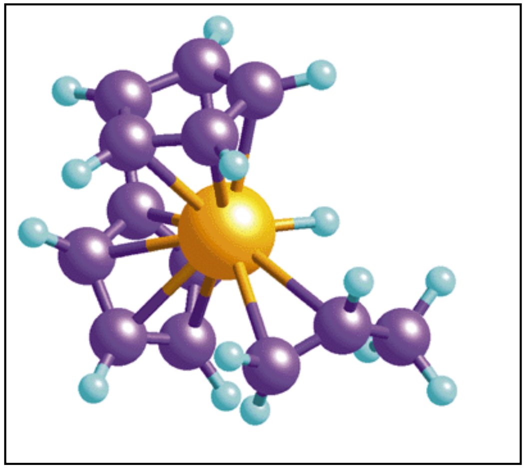
  - » Medical decision making – Optimal learning for medical treatments.



Drug discovery

- Designing molecules

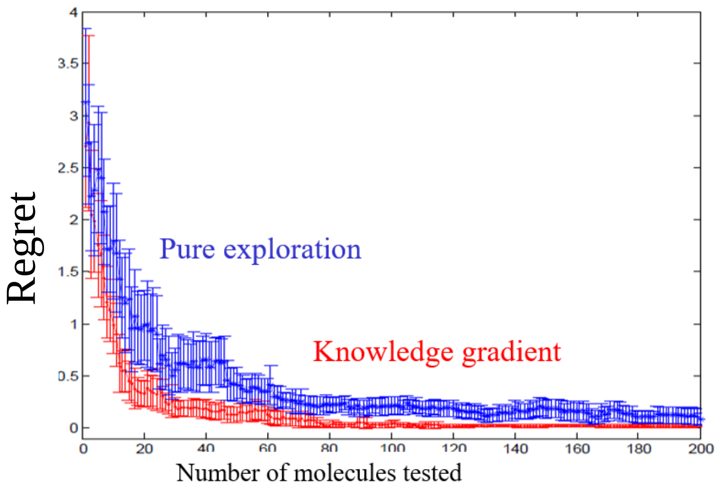




» X and Y are *sites* where we can hang *substituents* to change the behavior of the molecule. We approximate the performance using a linear belief model:

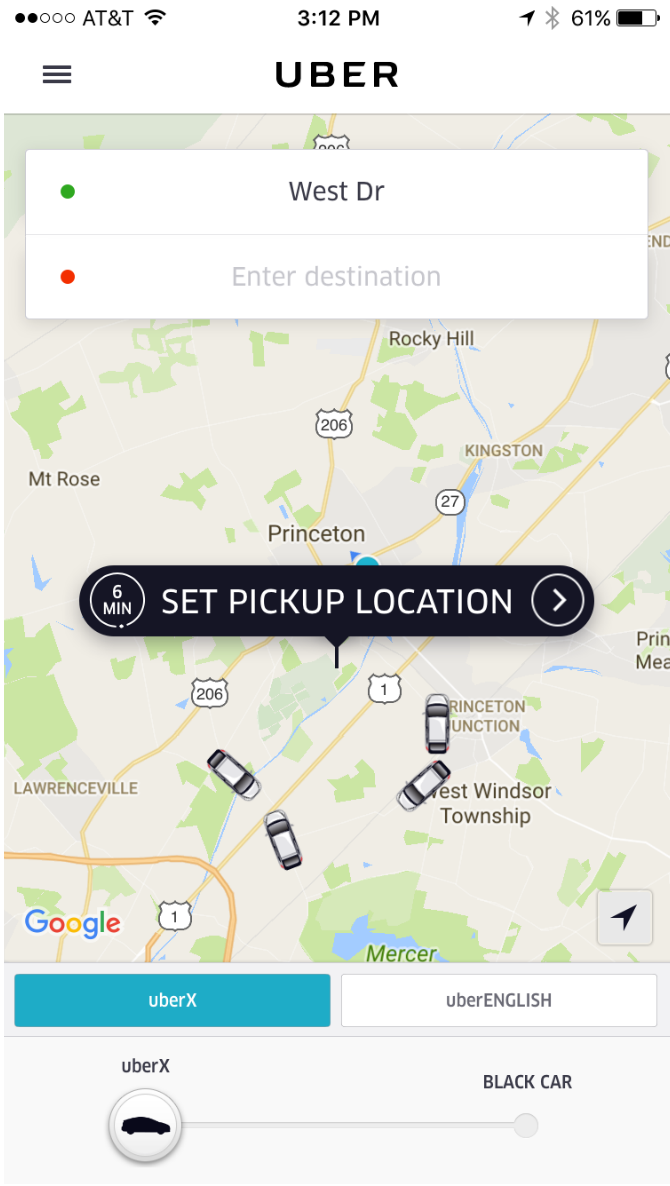
$$Y = \theta_0 + \sum_{sites\ i} \sum_{substituents\ j} \theta_{ij} X_{ij}$$

» How to sequence experiments to learn the best molecule as quickly as possible?



Ride sharing

- Uber/Lyft
  - » Provides real-time, on-demand transportation.
  - » Drivers are encouraged to enter or leave the system using pricing signals and informational guidance.
- Decisions:
  - » How to price to get the right balance of drivers relative to customers.
  - » Real-time management of drivers.
  - » Policies (rules for managing drivers, customers, …)



Matching buyers with sellers

- Now we have a logistic curve for each origin-destination pair (i,j)

$$P^Y(p,a\,|\,\theta) = \frac{e^{\theta_{ij}^0+\theta_{ij}p+\theta_{ij}^a a}}{1+e^{\theta_{ij}^0+\theta_{ij}p+\theta_{ij}^a a}}$$

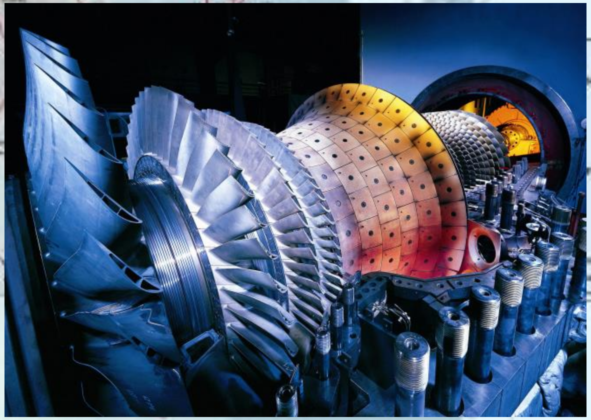
- Number of offers for each (i,j) pair is relatively small.
- Need to generalize the learning across hundreds to thousands of markets.

Emergency storm response

- Hurricane Sandy
  - » Once in 100 years?
  - » Rare convergence of events
  - » But, meteorologists did an amazing job of forecasting the storm.

- The power grid
  - » Loss of power creates cascading failures (lack of fuel, inability to pump water)
  - » How to plan?
  - » How to react?

Meeting variability with *portfolios* of generation with mixtures of *dispatchability*

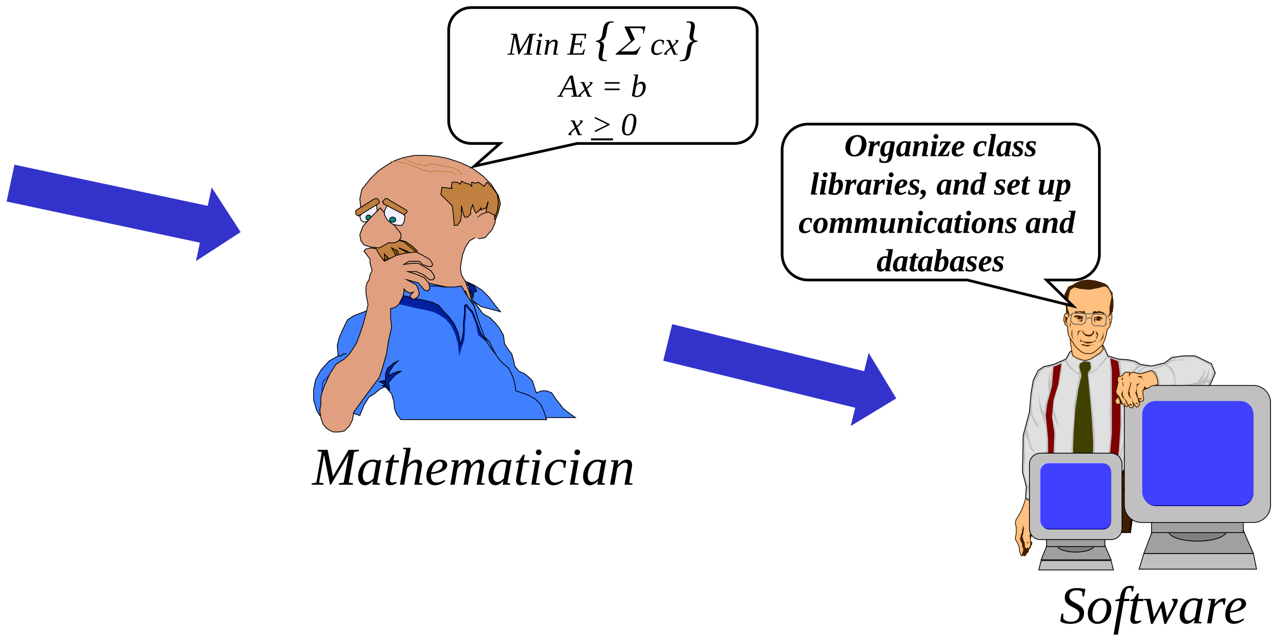


Storage applications

- How much energy to store in a battery to handle the volatility of wind and spot prices to meet demands?

Modeling

- Before we can *solve* complex problems, we have to know how to *think* about them.



- The biggest challenge when making decisions under uncertainty is *modeling*.

Modeling

- For deterministic problems, we speak the language of mathematical programming
  - » Linear programming:

$$\begin{aligned} \min_x \; & cx \\ Ax &= b \\ x &\geq 0 \end{aligned}$$

  - » For time-staged problems

$$\begin{aligned} \min_{x_0,\ldots,x_T} \; & \sum_{t=0}^{T} c_t x_t \\ A_t x_t - B_{t-1} x_{t-1} &= b_t \\ D_t x_t &\leq u_t \\ x_t &\geq 0 \end{aligned}$$

Arguably Dantzig's biggest contribution, more so than the simplex algorithm, was his articulation of optimization problems in a standard format, which has given algorithmic researchers a common language.


Stochastic programming
Robust optimization
Decision analysis
Approximate dynamic programming
Simulation optimization
Optimal learning
Model predictive control
Dynamic Programming and control
Stochastic search
Bandit problems
Optimal control
Reinforcement learning
Online computation
Markov decision processes
Stochastic control
Simulation optimization


John R. Birge
François Louveaux
Introduction to Stochastic Programming
Second Edition
Michael C. Fu Editor
Handbook of Simulation Optimization
Princeton Series in APPLIED MATHEMATICS
Robust Optimization
Introduction to Decision Analysis
A Practitioner's Guide to Improving Decision Quality
SECOND EDITION
Approximate Dynamic Programming
Solving the Curses of Dimensionality
Warren B. Powell
Wiley Series in Probability and Statistics
Optimal Learning
WILEY
SECOND EDITION
Model Predictive Control
VOLUME 2 • 4th EDITION
Dynamic Programming and Optimal Control
APPROXIMATE DYNAMIC PROGRAMMING
Dimitri P. Bertsekas
Athena Scientific
WILEY
INTRODUCTION TO STOCHASTIC SEARCH AND OPTIMIZATION
Estimation, Simulation, and Control
JAMES C. SPALL
MULTI-ARMED BANDIT ALLOCATION INDICES
SECOND EDITION
John Gittins, Kevin Glazebrook and Richard Weber
OPTIMAL CONTROL
Reinforcement Learning
Introduction
Richard S. Sutton and Andrew G. Barto
43
Jiongmin Yong
Xun Yu Zhou
Stochastic Controls
Hamiltonian Systems and HJB Equations
Modelling and Applied Probability
WILEY
Markov Decision Processes
Discrete Stochastic Dynamic Programming
MARTIN L. PUTERMAN
Online Computation and Competitive Analysis
Allan Borodin
Ran El-Yaniv
Vol. 1
STOCHASTIC SIMULATION OPTIMIZATION
An Optimal Computing Budget Allocation
Chun-Hung Chen • Loo Hay Lee

Outline

- Elements of a dynamic model
- Modeling uncertainty
- Designing policies
- The four classes of policies
- From deterministic to stochastic optimization

Outline

- Elements of a dynamic model
- Modeling uncertainty
- Designing policies
- The four classes of policies
- From deterministic to stochastic optimization

Modeling dynamic systems

- All sequential decision problems can be modeled using five core components:
  - » State variables
    - What do we need to know at time t?
  - » Decision variables
    - What are our decisions?
  - » Exogenous information
    - What do we learn for the first time between t and t+1?
  - » Transition function
    - How do the state variables evolve over time?.
  - » Objective function
    - What are our performance metrics?

Modeling dynamic problems

- The state variable:

Controls community

$$x_t = \text{"Information state"}$$

Operations research/MDP/Computer science

$$S_t = \left(R_t, I_t, B_t\right) = \text{System state, where:}$$

$R_t =$ Resource state (physical state)

Location/status of truck/train/plane

Energy in storage

$I_t =$ Information state

Prices

Weather

$B_t =$ Belief state ("state of knowledge")

Belief about performance of a drug or catalyst

Belief about the status of equipment

The state variable

- My definition of a state variable:

Definition 9.3.1 ***A state variable** is:*

a) Policy-dependent version *A function of history that, combined with the exogenous information (and a policy), is necessary and sufficient to compute the cost/contribution function, the decision function (the policy), and any information required to model the evolution of information needed in the cost/contribution and decision functions.*

b) Optimization version *A function of history that is necessary and sufficient to compute the cost/contribution function, the constraints, and any information required to model the evolution of information needed in the cost/contribution function and the constraints.*

  - » The first depends on a policy. The second depends only on the problem (and includes the constraints).
  - » Using either definition, ***all properly modeled problems are Markovian!***

Modeling dynamic problems

- Decisions:

Markov decision processes/Computer science

a_t = Discrete action

Control theory

u_t = Low-dimensional continuous vector

Operations research

x_t = Usually a discrete or continuous but high-dimensional vector of decisions.

At this point, we do not specify *how* to make a decision. Instead, we define the function $X^\pi(s)$ (or $A^\pi(s)$ or $U^\pi(s)$), where π specifies the type of policy. "π" carries information about the type of function f, and any tunable parameters $\theta \in \Theta^f$.

The decision variables

- Styles of decisions
  - » Binary

    $$x \in X = \{0,1\}$$
  - » Finite

    $$x \in X = \{1, 2, \ldots, M\}$$
  - » Continuous scalar

    $$x \in X = [a, b]$$
  - » Continuous vector

    $$x = (x_1, \ldots, x_K), \quad x_k \in \mathbb{R}$$
  - » Discrete vector

    $$x = (x_1, \ldots, x_K), \quad x_k \in \mathbb{Z}$$
  - » Categorical

    $$x = (a_1, \ldots, a_I), \quad a_i \text{ is a category (e.g. red/green/blue)}$$

Modeling dynamic problems

- Exogenous information:

W_t = New information that first became known at time t

$$= \left(\hat{R}_t, \hat{D}_t, \hat{p}_t, \hat{E}_t\right)$$

$\hat{R}_t$ = Equipment failures, delays, new arrivals

New drivers being hired to the network

$\hat{D}_t$ = New customer demands

$\hat{p}_t$ = Changes in prices

$\hat{E}_t$ = Information about the environment (temperature, ...)

Note: Any variable indexed by t is known at time t. This convention, which is not standard in control theory, dramatically simplifies the modeling of information.

Below, we let ω represent a sequence of actual observations $W_1, W_2, ...$

$W_t(\omega)$ refers to a sample realization of the random variable W_t.

Modeling dynamic problems

- The transition function

$$S_{t+1} = S^M(S_t, \ x_t, \ W_{t+1})$$

$$R_{t+1} = R_t + x_t + \hat{R}_{t+1}$$ Inventories

$$p_{t+1} = p_t + \hat{p}_{t+1}$$ Spot prices

$$D_{t+1} = D_t + \hat{D}_{t+1}$$ Market demands

Also known as the:

"System model"	"State equation"
"State transition model"	"Transfer function"
"Plant model"	"Transformation function"
"Plant equation"	"Law of motion"
"Transition law"	"Model"

For many applications, these equations are unknown. This is known as "model-free" dynamic programming.

Modeling stochastic, dynamic problems

- Objective functions
  - » Cumulative reward ("online learning")

$$\max_\pi \mathrm{E}\left\{\sum_{t=0}^{T} C_t\left(S_t, X_t^\pi(S_t), W_{t+1}\right) | S_0\right\}$$

    - Policies have to work well *over time.*
  - » Final reward ("offline learning")

$$\max_\pi \mathrm{E}\left\{F(x^{\pi,N},\hat{W}) | S_0\right\}$$

    - We only care about how well the final decision $x^{\pi,N}$ works.
  - » Risk

$$\max_\pi \rho\left\{C(S_0, X_0^\pi(S_0)), C(S_1, X_1^\pi(S_1)), \ldots, C(S_T, X_T^\pi(S_T)) | S_0\right\}$$

Modeling stochastic, dynamic problems

- The complete model:
  - » Objective function
    - Cumulative reward ("online learning")

$$\max_\pi \mathrm{E}\left\{\sum_{t=0}^{T} C_t\left(S_t, X_t^\pi(S_t), W_{t+1}\right) | S_0\right\}$$

    - Final reward ("offline learning")

$$\max_\pi \mathrm{E}\left\{F(x^{\pi,N}, \hat{W}) | S_0\right\}$$

    - Risk:

$$\max_\pi \rho\left\{C(S_0, X_0^\pi(S_0)), C(S_1, X_1^\pi(S_1)), \ldots, C(S_T, X_T^\pi(S_T)) | S_0\right\}$$

  - » Transition function:

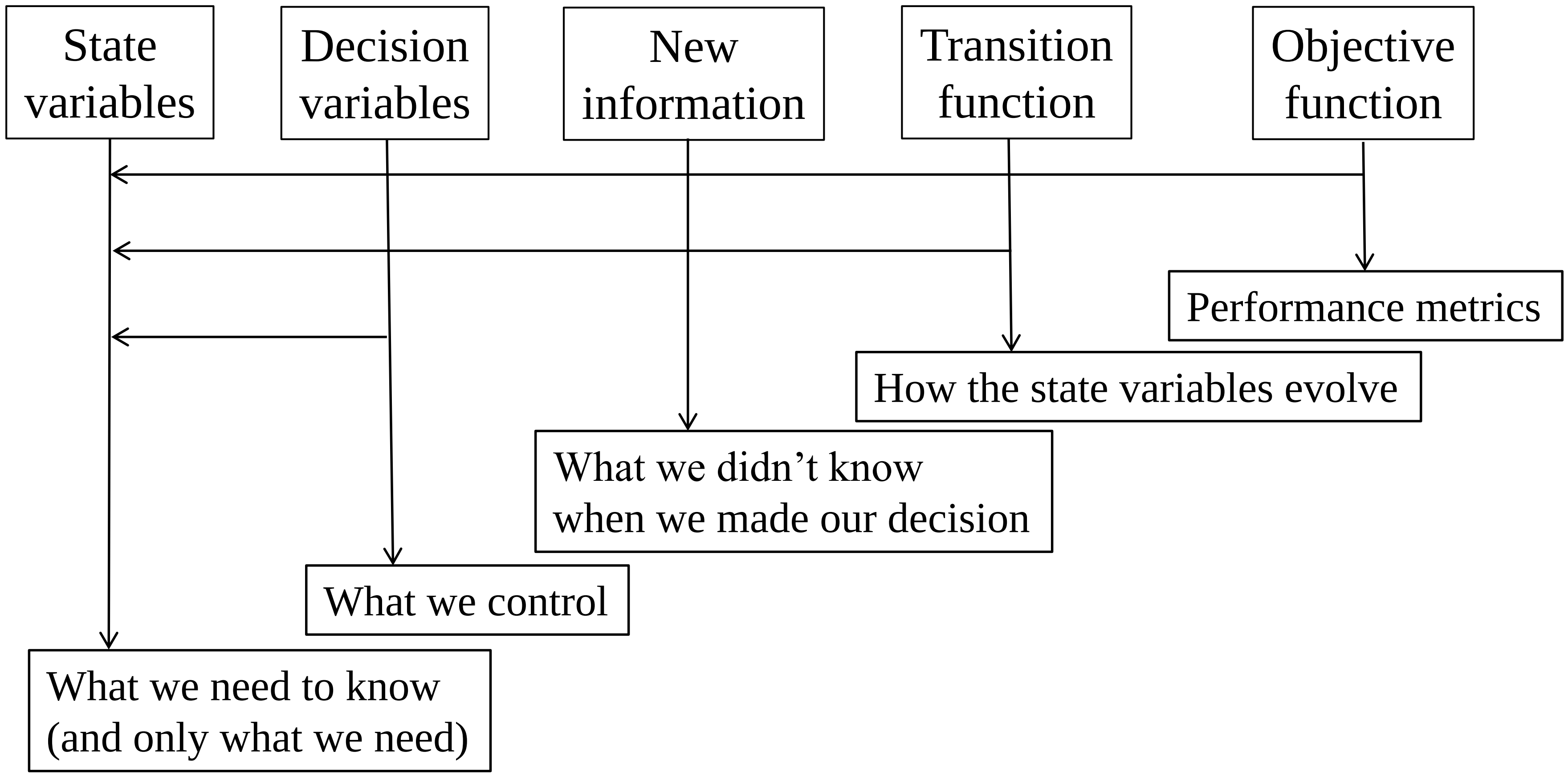
$$S_{t+1} = S^M\left(S_t, x_t, W_{t+1}(\omega)\right)$$

  - » Exogenous information:

$$\left(S_0, W_1, W_2, \ldots, W_T\right)$$

The modeling process

- Modeling real applications
  - » I conduct a conversation with a domain expert to fill in the elements of a problem:



Outline

- Elements of a dynamic model
- Modeling uncertainty
- Designing policies
- The four classes of policies
- From deterministic to stochastic optimization

Modeling uncertainty

- Classes of uncertainty
  - » Observational uncertainty
  - » Prognostic uncertainty (forecasting)
  - » Experimental noise/variability
  - » Transitional uncertainty
  - » Inferential uncertainty
  - » Model uncertainty
  - » Systematic exogenous uncertainty
  - » Control/implementation uncertainty
  - » Algorithmic noise
  - » Goal uncertainty

Modeling uncertainty in the context of stochastic optimization is a relatively untapped area of research.

Outline

- Elements of a dynamic model
- Modeling uncertainty
- Designing policies
- The four classes of policies
- From deterministic to stochastic optimization

Designing policies

- We have to start by describing what we mean by a policy.
  - » Definition:

    A policy is a mapping from a state to an action.

    ... any mapping.

- How do we search over an arbitrary space of policies?

Designing policies

- "Policies" and the English language

Behavior	Habit	Procedure
Belief	Laws/bylaws	Process
Bias	Manner	Protocols
Commandment	Method	Recipe
Conduct	Mode	Ritual
Convention	Mores	Rule
Culture	Patterns	Style
Customs	Plans	Technique
Dogma	Policies	Tenet
Etiquette	Practice	Tradition
Fashion	Prejudice	Way of life
Formula	Principle	

Designing policies

- Two fundamental strategies for finding policies:

1) Policy search – Search over a class of functions for making decisions to optimize some metric.

$$\max_{\pi=(f\in F,\theta^f\in\Theta^f)} E\left\{\sum_{t=0}^{T} C_t\left(S_t, X_t^{\pi}(S_t\mid\theta)\right)\mid S_0\right\}$$

2) Lookahead approximations – Approximate the impact of a decision now on the future.

$$X_t^*(S_t) = \arg\max_{x_t}\left(C(S_t,x_t) + \mathrm{E}\left\{\max_{\pi\in\Pi}\left\{\mathrm{E}\sum_{t'=t+1}^{T} C_{t'}(S_{t'}, X_{t'}^{\pi}(S_{t'}))\mid S_{t+1}\right\}\mid S_t, x_t\right\}\right)$$

Designing policies

- Policy search:

  1a) Policy function approximations (PFAs) $x_t = X^{PFA}(S_t \mid \theta)$

  - Lookup tables
    - "when in this state, take this action"
  - Parametric functions
    - Order-up-to policies: if inventory is less than s, order up to S.
    - Affine policies - $x_t = X^{PFA}(S_t \mid \theta) = \sum_{f \in F} \theta_f \phi_f(S_t)$
    - Neural networks
  - Locally/semi/non parametric
    - Requires optimizing over local regions

  1b) Cost function approximations (CFAs)

  - Optimizing a deterministic model modified to handle uncertainty (buffer stocks, schedule slack)

$$X^{CFA}(S_t \mid \theta) = \arg\max_{x_t \in \bar{X}_t^{\pi}(\theta)} \bar{C}^{\pi}(S_t, x_t \mid \theta)$$

Designing policies

- Lookahead approximations – Approximate the impact of a decision now on the future:

  2a) Approximating the value of being in a state (VFA):

$$X_t^*(S_t) = \arg\max_{x_t} \left(C(S_t, x_t) + \mathrm{E}\left\{ V_{t+1}(S_{t+1}) \mid S_t, x_t \right\} \right)$$

$$X_t^{VFA}(S_t) = \arg\max_{x_t} \left(C(S_t, x_t) + \mathrm{E}\left\{ \bar{V}_{t+1}(S_{t+1}) \mid S_t, x_t \right\} \right)$$

$$= \arg\max_{x_t} \left(C(S_t, x_t) + \bar{V}_t^x(S_t^x) \right)$$

  2b) Direct lookahead (DLA)

  Optimal policy:

$$X_t^*(S_t) = \arg\max_{x_t} \left(C(S_t, x_t) + \mathrm{E}\left\{ \max_{\pi \in \Pi} \left\{ \mathrm{E} \sum_{t'=t+1}^{T} C(S_{t'}, X_{t'}^{\pi}(S_{t'})) \mid S_{t+1} \right\} \mid S_t, x_t \right\} \right)$$

  Approximate policy – solve an approximate *lookahead model*:

$$X_t^{DLA}(S_t) = \arg\max_{x_t} \left(C(S_t, x_t) + \tilde{\mathrm{E}}\left\{ \max_{\tilde{\pi} \in \tilde{\Pi}} \left\{ \tilde{\mathrm{E}} \sum_{t'=t+1}^{t+H} C(\tilde{S}_{tt'}, \tilde{X}_{t'}^{\pi}(\tilde{S}_{tt'})) \mid \tilde{S}_{t,t+1} \right\} \mid \tilde{S}_{tt}, x_t \right\} \right)$$

Four (meta)classes of policies

Policy search

1) **Policy function approximations (PFAs)**
   » Lookup tables, rules, parametric/nonparametric functions
2) **Cost function approximation (CFAs)**
   » $X^{CFA}(S_t \mid \theta) = \arg\max_{x_t \in \bar{\mathcal{X}}_t^{\pi}(\theta)} \bar{C}^{\pi}(S_t, x_t \mid \theta)$

Lookahead approximations

3) **Policies based on value function approximations (VFAs)**
   » $X_t^{VFA}(S_t) = \arg\max_{x_t} \left(C(S_t, x_t) + \bar{V}_t^{x}\left(S_t^{x}(S_t, x_t) \right) \right)$
4) **Direct lookahead policies (DLAs)**
   » *Deterministic lookahead/rolling horizon proc./model predictive control*
   $$X_t^{LA-D}(S_t) = \arg \max_{\tilde{x}_{tt},\ldots,\tilde{x}_{t,t+H}} C(\tilde{S}_{tt}, \tilde{x}_{tt}) + \sum_{t'=t+1}^{T} C(\tilde{S}_{tt'}, \tilde{x}_{tt'})$$
   » *Chance constrained programming*
   $$P[A_t x_t \leq f(W)] \leq 1 - \delta$$
   » *Stochastic lookahead /stochastic prog/Monte Carlo tree search*
   $$X_t^{LA-S}(S_t) = \arg\max_{\tilde{x}_{tt}, \tilde{x}_{t,t+1}, \ldots, \tilde{x}_{t,t+T}} C(\tilde{S}_{tt}, \tilde{x}_{tt}) + \sum_{\tilde{\omega} \in \tilde{\Omega}_t} p(\tilde{\omega}) \sum_{t'=t+1}^{T} C(\tilde{S}_{tt'}(\tilde{\omega}), \tilde{x}_{tt'}(\tilde{\omega}))$$
   » *"Robust optimization"*
   $$X_t^{LA-RO}(S_t) = \arg \max_{\tilde{x}_{tt},\ldots,\tilde{x}_{t,t+H}} \min_{w \in W_t(\theta)} C(\tilde{S}_{tt}, \tilde{x}_{tt}) + \sum_{t'=t+1}^{T} C(\tilde{S}_{tt'}(w), \tilde{x}_{tt'}(w))$$

Four (meta)classes of policies

Function approx.

1) **Policy function approximations (PFAs)**

» Lookup tables, rules, parametric/nonparametric functions

2) **Cost function approximation (CFAs)**

» $X^{CFA}(S_t \mid \theta) = \arg\max_{x_t \in \bar{\mathcal{X}}_t^{\pi}(\theta)} \bar{C}^{\pi}(S_t, x_t \mid \theta)$

3) **Policies based on value function approximations (VFAs)**

» $X_t^{VFA}(S_t) = \arg\max_{x_t} \left(C(S_t, x_t) + \bar{V}_t^{x}\left(S_t^{x}(S_t, x_t) \right) \right)$

4) **Direct lookahead policies (DLAs)**

» *Deterministic lookahead/rolling horizon proc./model predictive control*

$$X_t^{LA-D}(S_t) = \arg \max_{\tilde{x}_{tt},\ldots,\tilde{x}_{t,t+H}} C(\tilde{S}_{tt}, \tilde{x}_{tt}) + \sum_{t'=t+1}^{T} C(\tilde{S}_{tt'}, \tilde{x}_{tt'})$$

» *Chance constrained programming*

$$P[A_t x_t \le f(W)] \le 1 - \delta$$

» *Stochastic lookahead /stochastic prog/Monte Carlo tree search*

$$X_t^{LA-S}(S_t) = \arg\max_{\tilde{x}_{tt},\tilde{x}_{t,t+1},\ldots,\tilde{x}_{t,t+T}} C(\tilde{S}_{tt}, \tilde{x}_{tt}) + \sum_{\tilde{\omega} \in \tilde{\Omega}_t} p(\tilde{\omega}) \sum_{t'=t+1}^{T} C(\tilde{S}_{tt'}(\tilde{\omega}), \tilde{x}_{tt'}(\tilde{\omega}))$$

» *"Robust optimization"*

$$X_t^{LA-RO}(S_t) = \arg \max_{\tilde{x}_{tt},\ldots,\tilde{x}_{t,t+H}} \min_{w \in W_t(\theta)} C(\tilde{S}_{tt}, \tilde{x}_{tt}) + \sum_{t'=t+1}^{T} C(\tilde{S}_{tt'}(w), \tilde{x}_{tt'}(w))$$

Four (meta)classes of policies

1) **Policy function approximations (PFAs)**

» Lookup tables, rules, parametric/nonparametric functions

Imbedded optimization

2) **Cost function approximation (CFAs)**

» $$X^{CFA}(S_t \mid \theta) = \arg\max_{x_t \in \bar{\mathcal{X}}_t^{\pi}(\theta)} \bar{C}^{\pi}(S_t, x_t \mid \theta)$$

3) **Policies based on value function approximations (VFAs)**

» $$X_t^{VFA}(S_t) = \arg\max_{x_t} \left(C(S_t, x_t) + \bar{V}_t^{x}\left(S_t^{x}(S_t, x_t)\right)\right)$$

4) **Direct lookahead policies (DLAs)**

» *Deterministic lookahead/rolling horizon proc./model predictive control*

$$X_t^{LA-D}(S_t) = \arg \max_{\tilde{x}_{tt},\ldots,\tilde{x}_{t,t+H}} C(\tilde{S}_{tt}, \tilde{x}_{tt}) + \sum_{t'=t+1}^{T} C(\tilde{S}_{tt'}, \tilde{x}_{tt'})$$

» *Chance constrained programming*

$$P[A_t x_t \leq f(W)] \leq 1-\delta$$

» *Stochastic lookahead /stochastic prog/Monte Carlo tree search*

$$X_t^{LA-S}(S_t) = \arg\max_{\tilde{x}_{tt},\tilde{x}_{t,t+1},\ldots,\tilde{x}_{t,t+T}} C(\tilde{S}_{tt}, \tilde{x}_{tt}) + \sum_{\tilde{\omega}\in\tilde{\Omega}_t} p(\tilde{\omega}) \sum_{t'=t+1}^{T} C(\tilde{S}_{tt'}(\tilde{\omega}), \tilde{x}_{tt'}(\tilde{\omega}))$$

» *"Robust optimization"*

$$X_t^{LA-RO}(S_t) = \arg \max_{\tilde{x}_{tt},\ldots,\tilde{x}_{t,t+H}} \min_{w\in W_t(\theta)} C(\tilde{S}_{tt}, \tilde{x}_{tt}) + \sum_{t'=t+1}^{T} C(\tilde{S}_{tt'}(w), \tilde{x}_{tt'}(w))$$

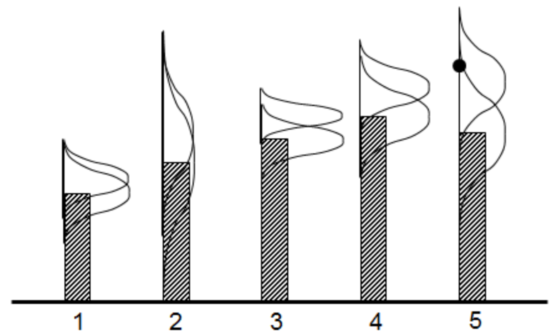
Learning problems

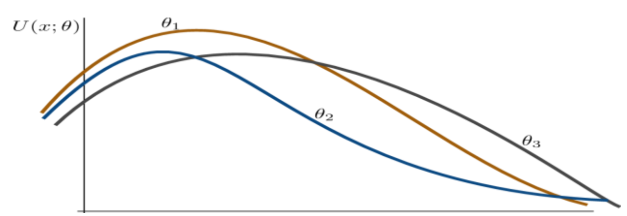
- Classes of learning problems in stochastic optimization

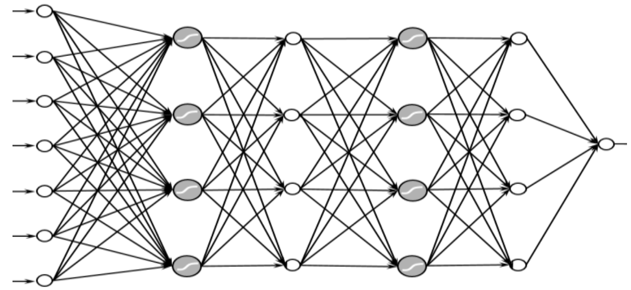
  1) Approximating the objective
  $$\bar{F}(x|\theta) \approx \mathbb{E}F(x, W).$$
  2) Designing a policy $X^{\pi}(S|\theta)$.
  3) A value function approximation
  $$\bar{V}_t(S_t|\theta) \approx V_t(S_t).$$
  4) Designing a cost function approximation:
     - The objective function $\bar{C}^{\pi}(S_t, x_t|\theta)$.
     - The constraints $X^{\pi}(S_t|\theta)$
  5) Approximating the transition function
  $$\bar{S}^M(S_t, x_t, W_{t+1}|\theta) \approx S^M(S_t, x_t, W_{t+1})$$

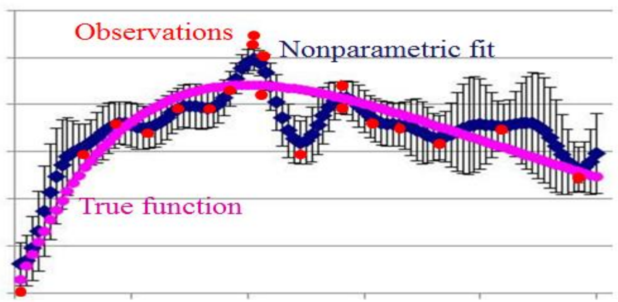
Approximation strategies

- Approximation strategies
  - » Lookup tables
    - Independent beliefs
    - Correlated beliefs
  - » Linear parametric models
    - Linear models
    - Sparse-linear
    - Tree regression
  - » Nonlinear parametric models
    - Logistic regression
    - Neural networks
  - » Nonparametric models
    - Gaussian process regression
    - Kernel regression
    - Support vector machines
    - Deep neural networks





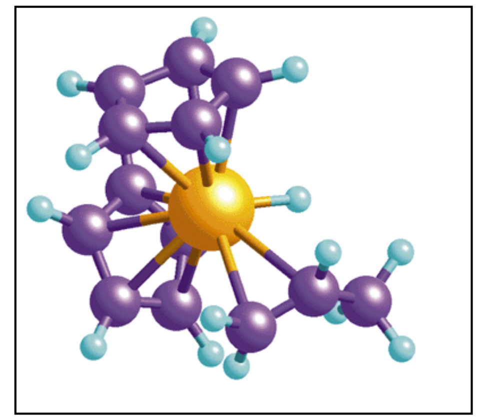


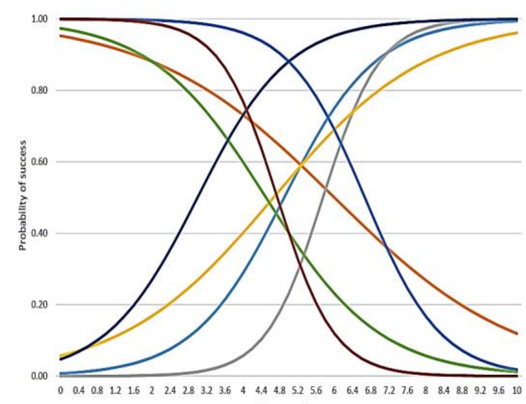


Learning challenges

- The learning challenge

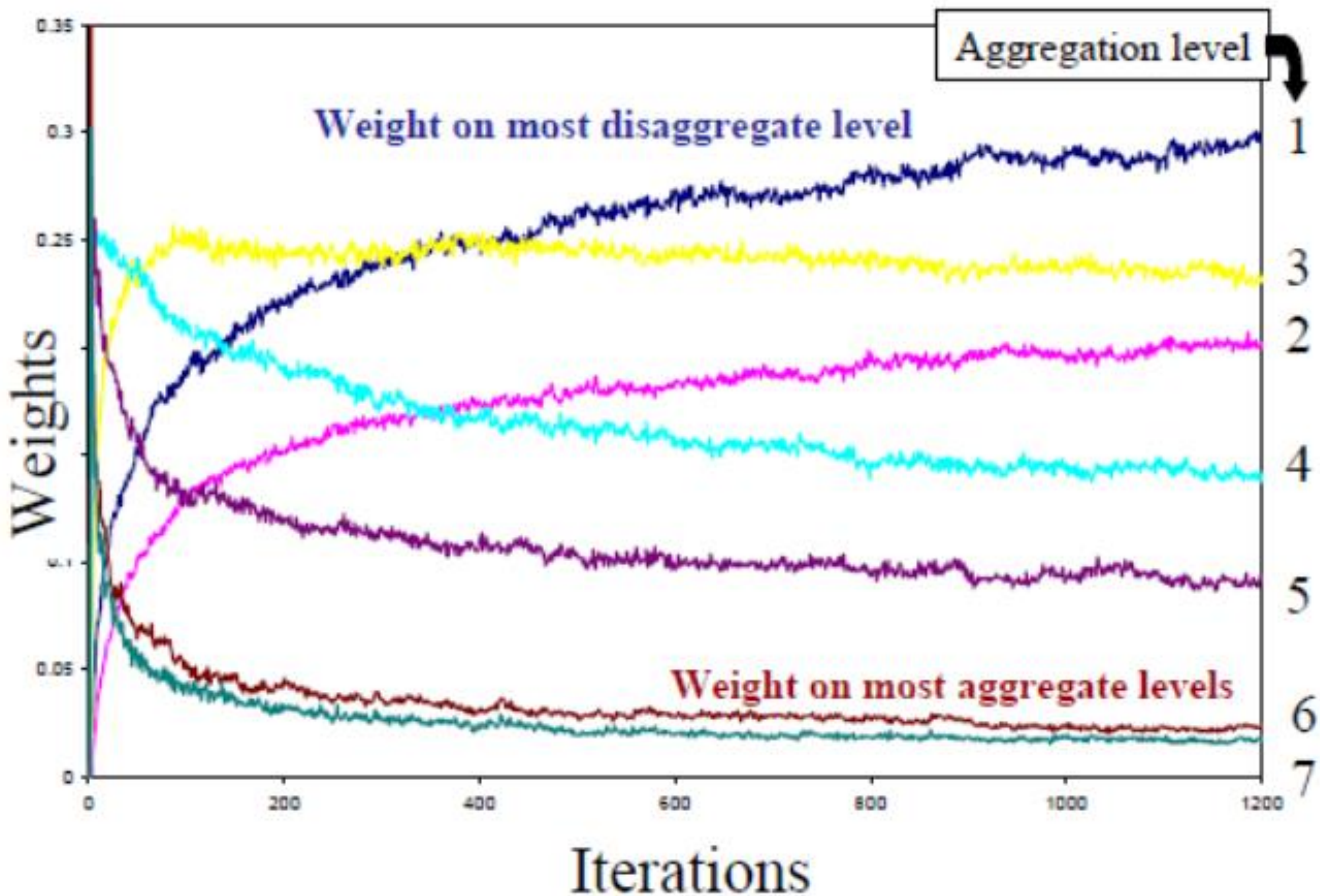
From big (batch) data… … to recursive learning





Learning challenges

- Variable-dimensional learning



Outline

- Elements of a dynamic model
- Modeling uncertainty
- Designing policies
- The four classes of policies
- From deterministic to stochastic optimization

Outline

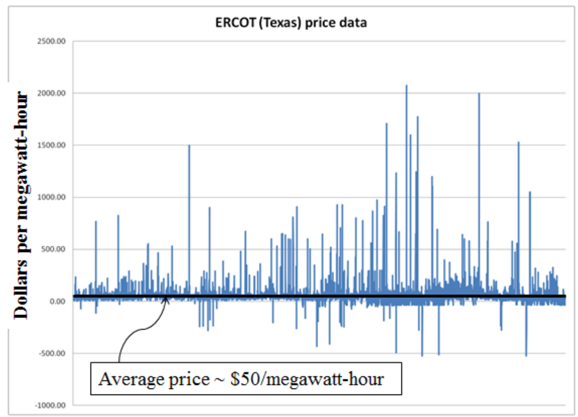
- The four classes of policies
  - » Policy function approximations (PFAs)
  - » Cost function approximations (CFAs)
  - » Value function approximations (VFAs)
  - » Direct lookahead policies (DLAs)
  - » A hybrid lookahead/CFA

Outline

- The four classes of policies
  - » Policy function approximations (PFAs)
  - » Cost function approximations (CFAs)
  - » Value function approximations (VFAs)
  - » Direct lookahead policies (DLAs)
  - » A hybrid lookahead/CFA

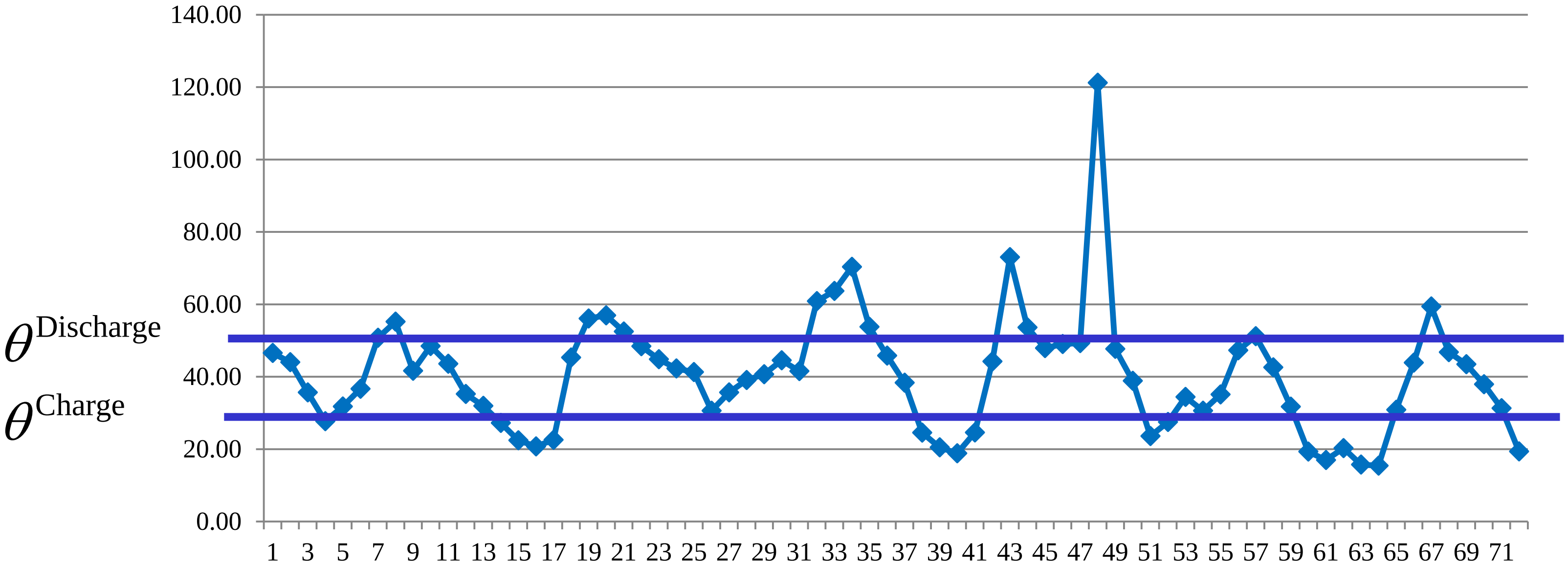
Policy function approximations

- Battery arbitrage – When to charge, when to discharge, given volatile LMPs



Policy function approximations

- Grid operators require that batteries bid charge and discharge prices, an hour in advance.

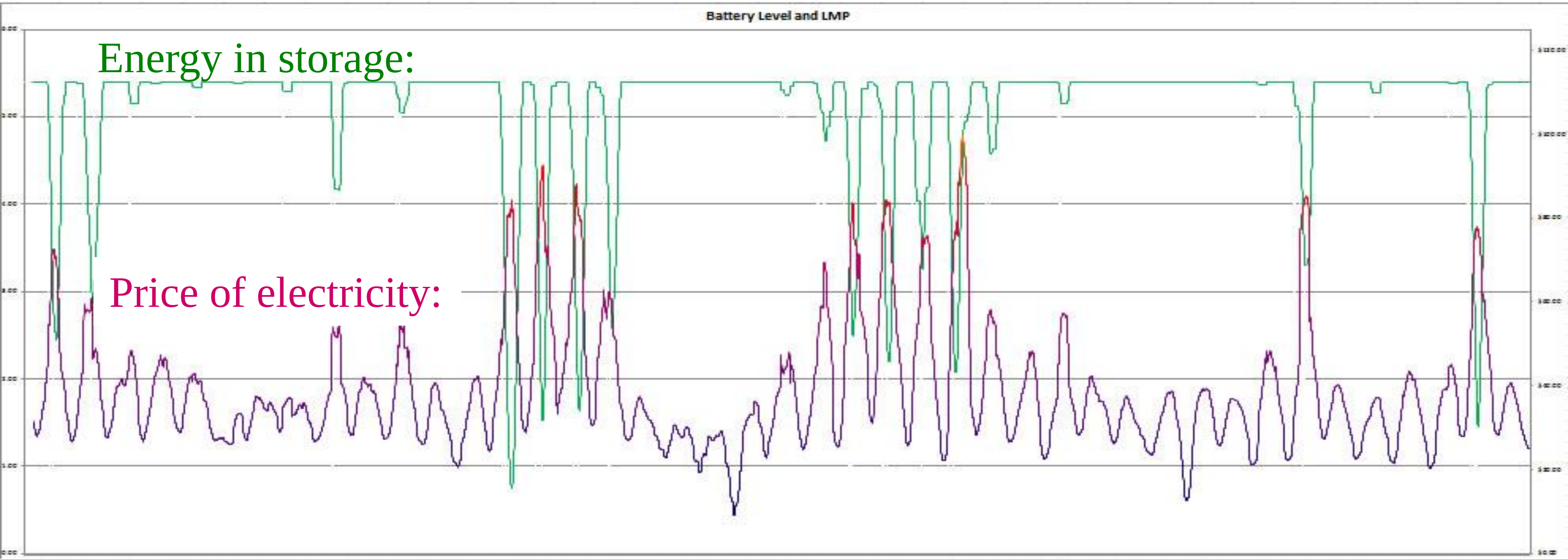


- We have to search for the best values for the policy parameters θ^{Charge} and $\theta^{\text{Discharge}}$.

Policy function approximations

- Our policy function might be the parametric model (this is nonlinear in the parameters):

$$X^{\pi}(S_t \mid \theta) = \begin{cases} +1 & \text{if } p_t < \theta^{\text{charge}} \\ 0 & \text{if } \theta^{\text{charge}} < p_t < \theta^{\text{discharge}} \\ -1 & \text{if } p_t > \theta^{\text{charge}} \end{cases}$$

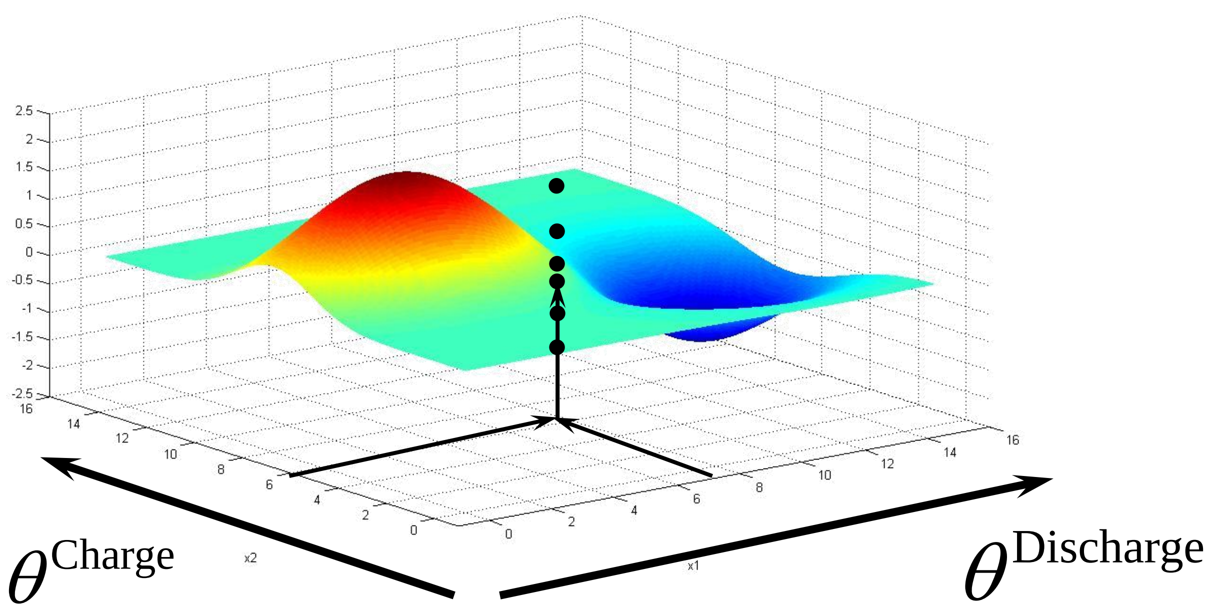


Policy function approximations

- Finding the best policy
  - » We need to maximize

$$\max_\theta F(\theta) = \mathrm{E}\sum_{t=0}^{T}\gamma^t C\Big(S_t, X_t^\pi(S_t \mid \theta)\Big)$$

  - » We cannot compute the expectation, so we run simulations:



Outline

- The four classes of policies
  - » Policy function approximations (PFAs)
  - » Cost function approximations (CFAs)
  - » Value function approximations (VFAs)
  - » Direct lookahead policies (DLAs)
  - » A hybrid lookahead/CFA

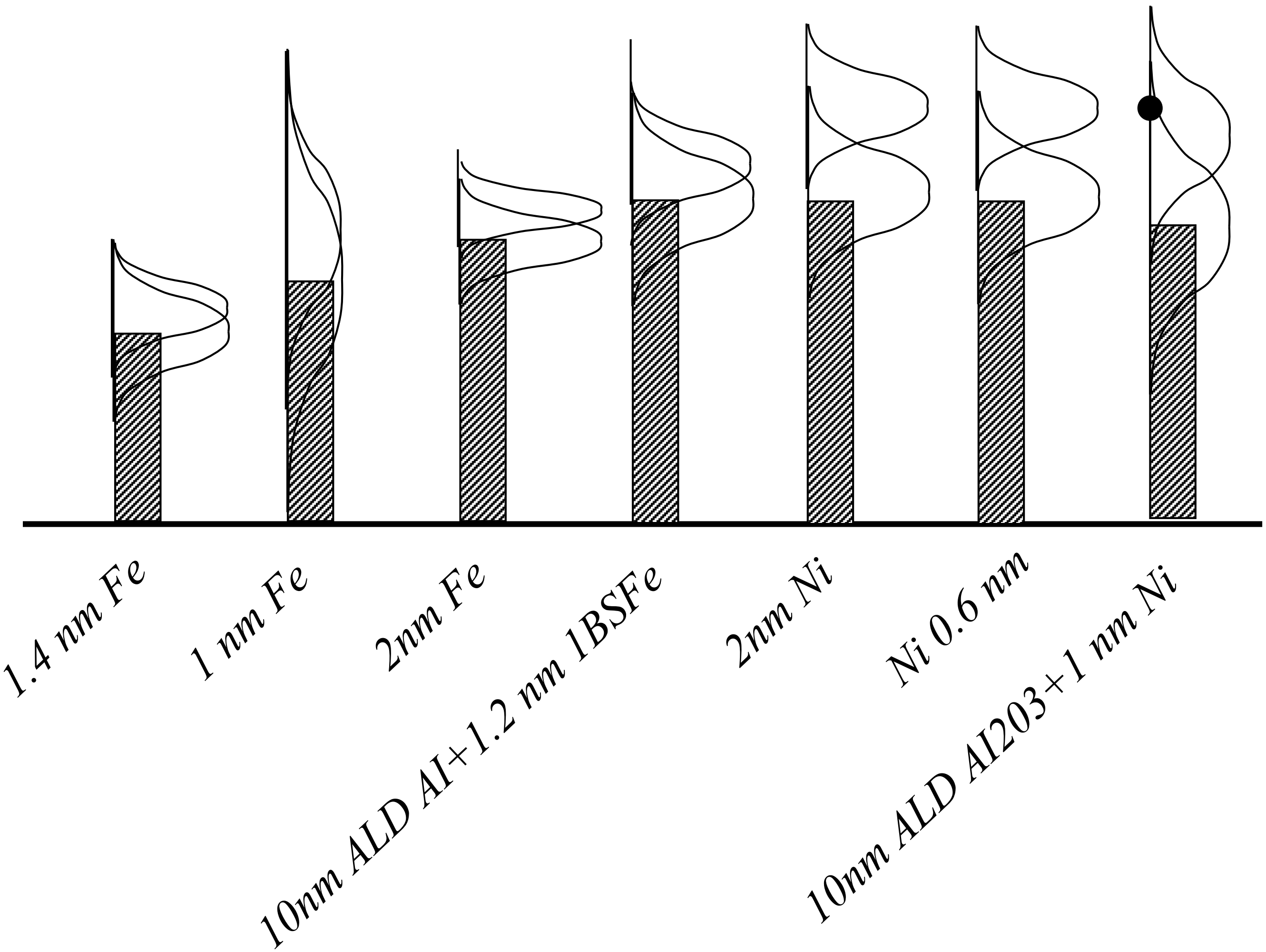
Cost function approximations

- Lookup table
  - » We can organize potential catalysts into groups
  - » Scientists using domain knowledge can estimate correlations in experiments between similar catalysts.

	1.4 nm Fe	1 nm Fe	2nm Fe	10nm ALD Al2O3+1.2 nm IBS Fe	2 nm Ni	Ni 0.6 nm	10nm ALD Al2O3+1 nm Ni
1.4 nm Fe	1	0.7	0.7	0.6	0.4	0.4	0.2
1 nm Fe	0.7	1	0.7	0.6	0.4	0.4	0.2
2nm Fe	0.7	0.7	1	0.6	0.4	0.4	0.2
10nm ALD Al2O3+1.2 nm IBS Fe	0.6	0.6	0.6	1	1	0.3	0
2 nm Ni	0.4	0.4	0.4	1	1	0.7	0.6
Ni 0.6 nm	0.4	0.4	0.4	0.3	0.7	1	0.6
10nm ALD Al2O3+1 nm Ni	0.2	0.2	0.2	0	0.6	0.6	1

Cost function approximations

- Correlated beliefs: Testing one material teaches us about other materials

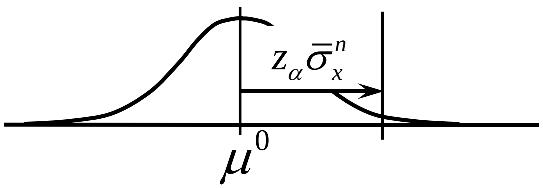


Cost function approximations

- Cost function approximations (CFA)
  » Upper confidence bounding

$$X^{UCB}(S^n \mid \theta^{UCB}) = \arg\max_x \left(\bar{\mu}_x^n + \theta^{UCB} \sqrt{\frac{\log n}{N_x^n}} \right)$$

  » Interval estimation



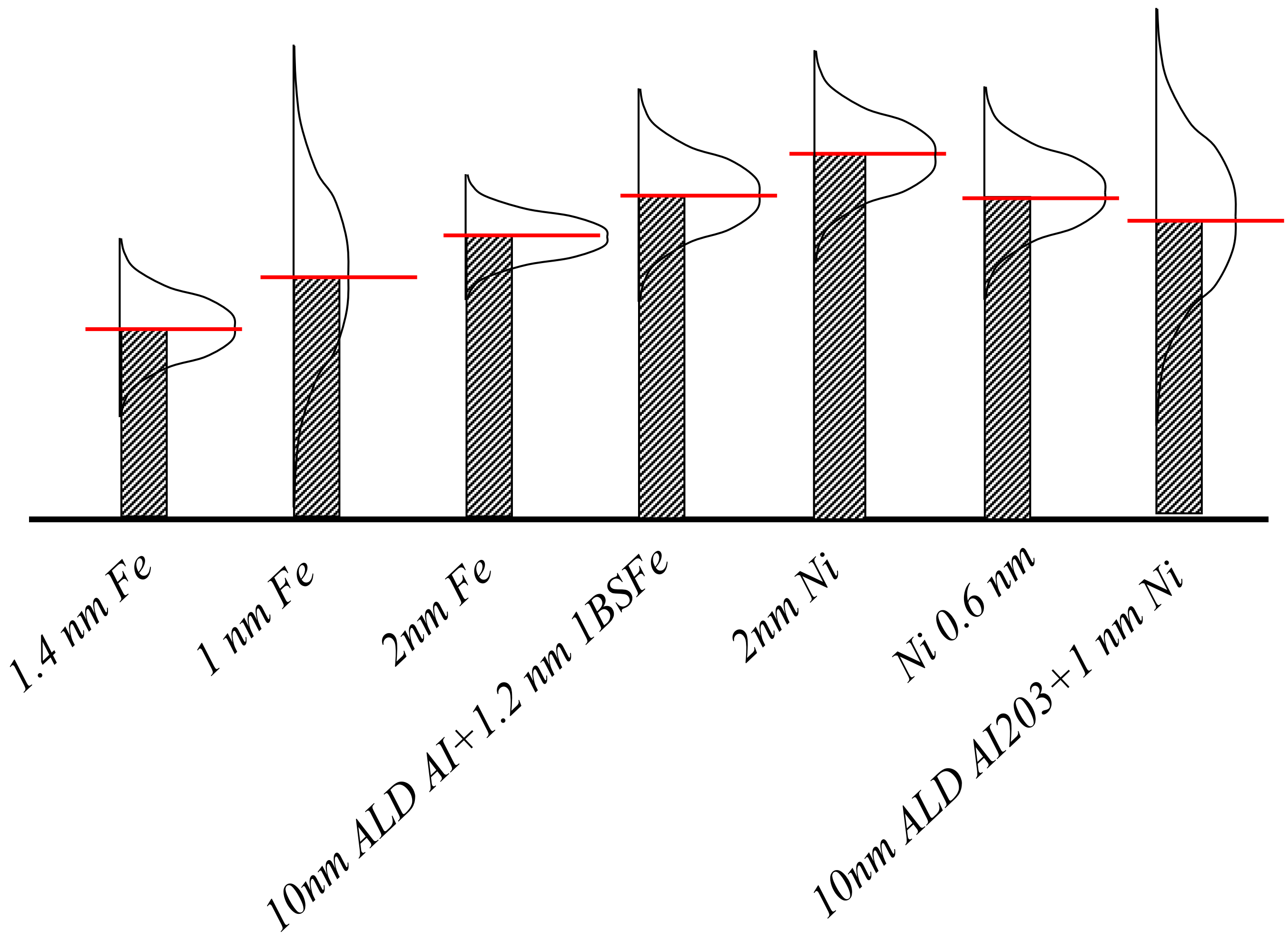
$$X^{IE}(S^n \mid \theta^{IE}) = \arg\max_x \left(\bar{\mu}_x^n + \theta^{IE} \bar{\sigma}_x^n \right)$$

  » Boltzmann exploration ("soft max")
    - Choose x with probability: $P_x^n(\theta) = \dfrac{e^{\theta \bar{\mu}_x^n}}{\sum_{x'} e^{\theta \bar{\mu}_{x'}^n}}$

$$X^{Boltz}(S^n|\theta) = \arg\max_x \{x | P_x^n(\theta) \leq U\}.$$

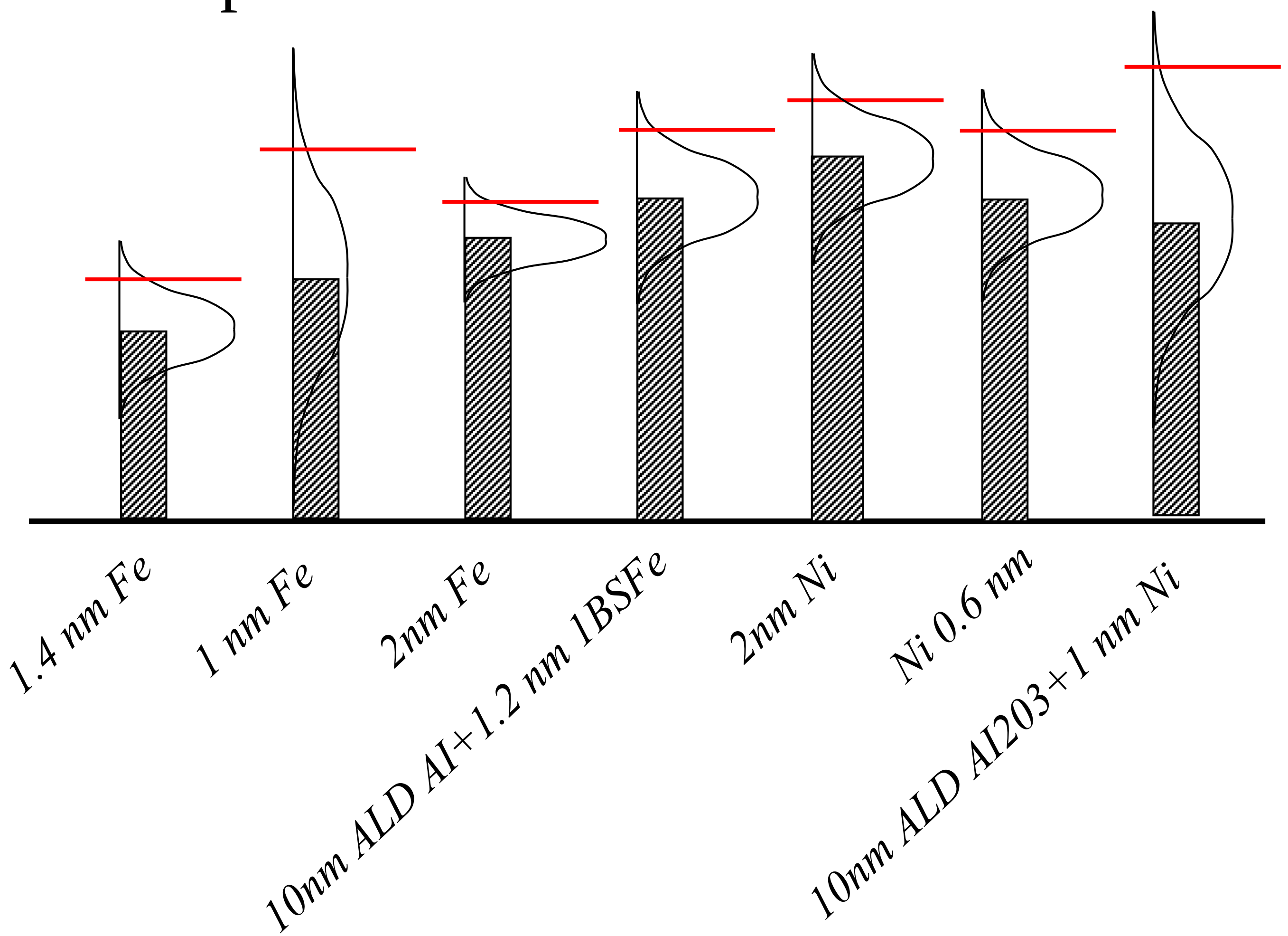
Cost function approximations

- Picking $\theta^{IE} = 0$ means we are evaluating each choice at the mean.



Cost function approximations

- Picking $\theta^{IE} = 2$ means we are evaluating each choice at the 95th percentile.



Cost function approximations

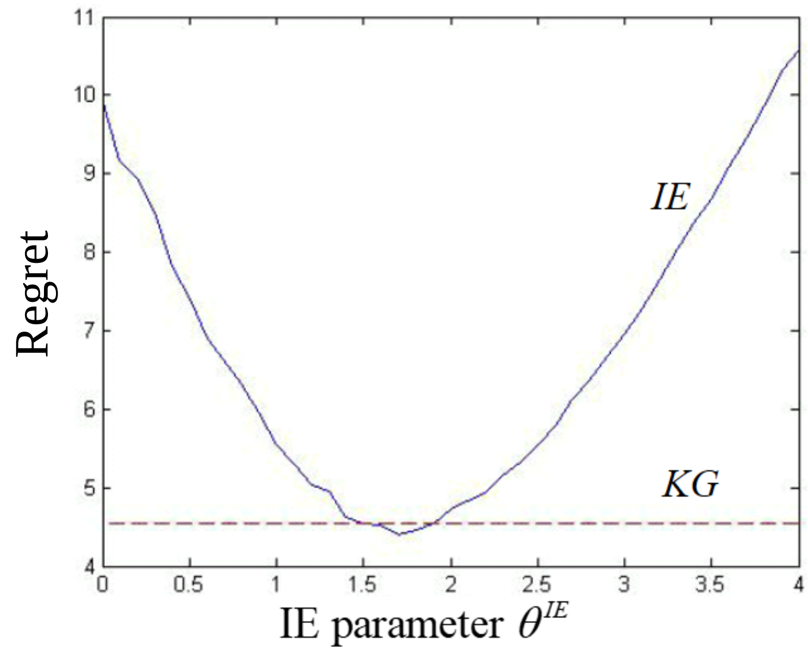
- Optimizing the policy
  - » We optimize θ^{IE} to maximize:

$$\max_{\theta^{IE}} F(\theta^{IE}) = \mathbb{E}F\left(x^{\pi,N}, W\right)$$

where

$$x^n = X^{IE}(S^n \mid \theta^{IE}) = \arg\max_x \left(\bar{\mu}_x^n + \theta^{IE}\bar{\sigma}_x^n\right) \qquad S^n = (\bar{\mu}_x^n, \bar{\sigma}_x^n)$$

- Notes:
  - » This can handle any belief model, including correlated beliefs, nonlinear belief models.
  - » All we require is that we be able to simulate a policy.



Cost function approximations

- Other applications
  - » Airlines optimizing schedules with schedule slack to handle weather uncertainty.
  - » Manufacturers using buffer stocks to hedge against production delays and quality problems.
  - » Grid operators scheduling extra generation capacity in case of outages.
  - » Adding time to a trip planned by Google maps to account for uncertain congestion.

Outline

- The four classes of policies
  - Policy function approximations (PFAs)
  - Cost function approximations (CFAs)
  - Value function approximations (VFAs)
  - Direct lookahead policies (DLAs)
  - A hybrid lookahead/CFA

Value function approximations

- Q-learning (for discrete actions)

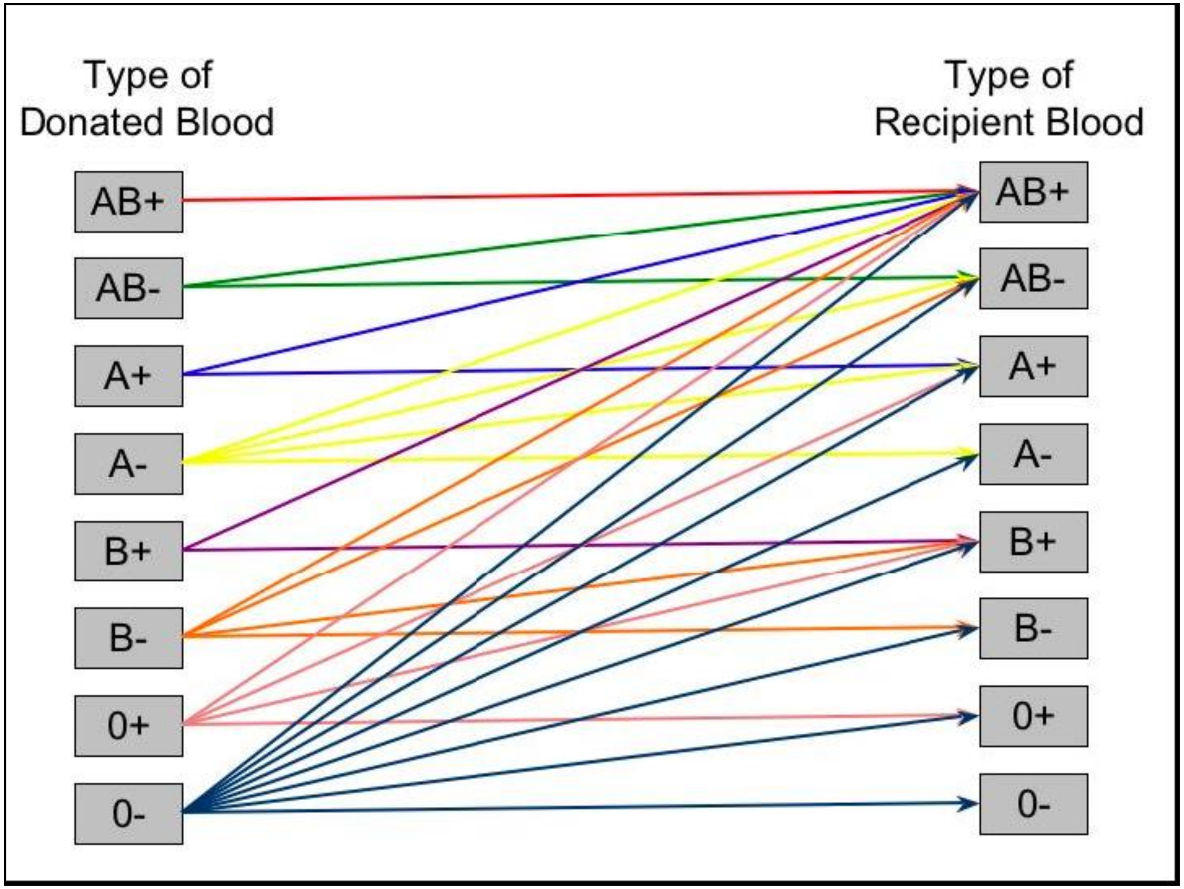
$$\hat{q}^n(s^n, a^n) = r(s^n, a^n) + g \max_{a'} \bar{Q}^{n-1}(s', a')$$

$$\bar{Q}^n(s^n, a^n) = (1 - a_{n-1})\bar{Q}^{n-1}(s^n, a^n) + a_{n-1}\hat{q}^n(s^n, a^n)$$

» But what if the action *a* is a vector?

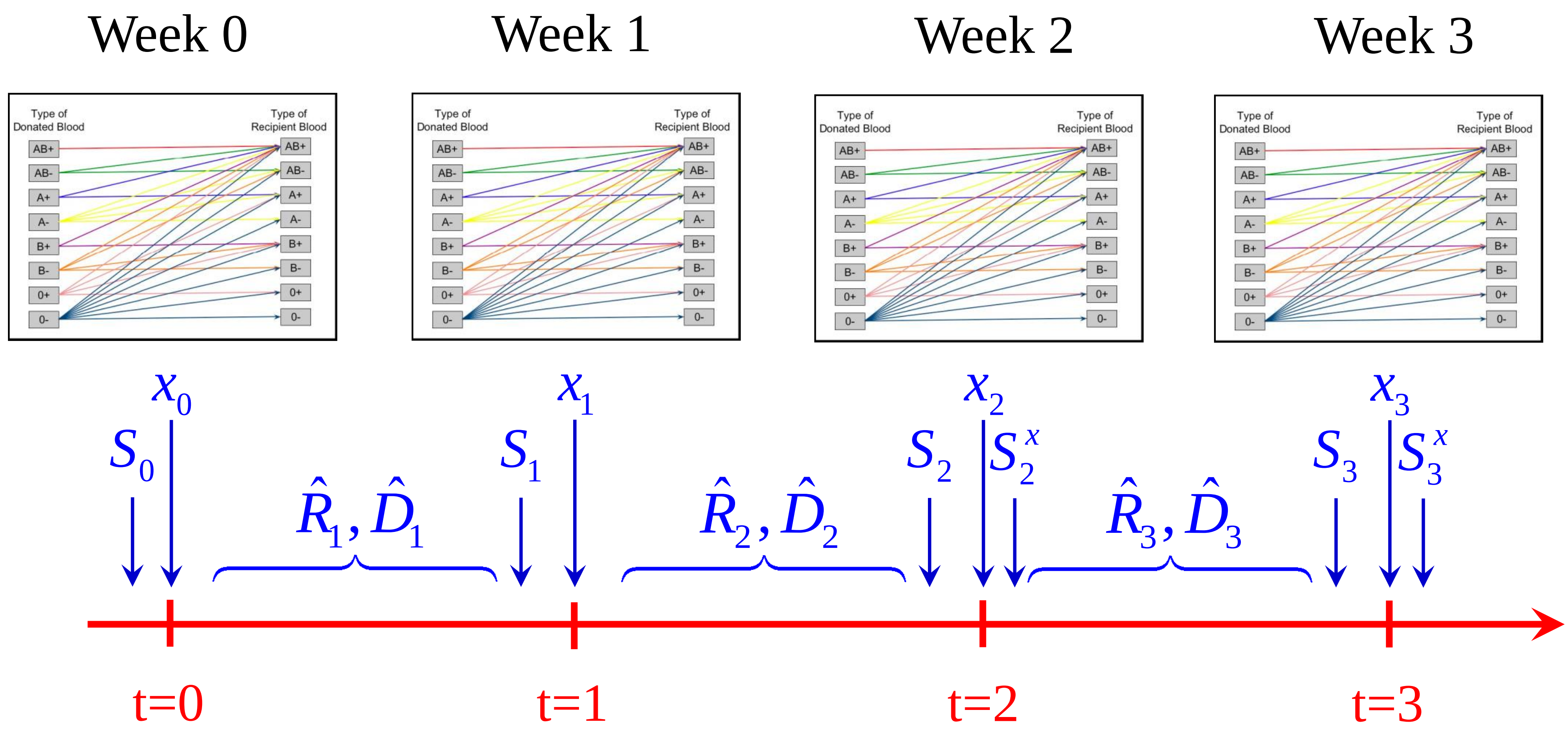
Blood management

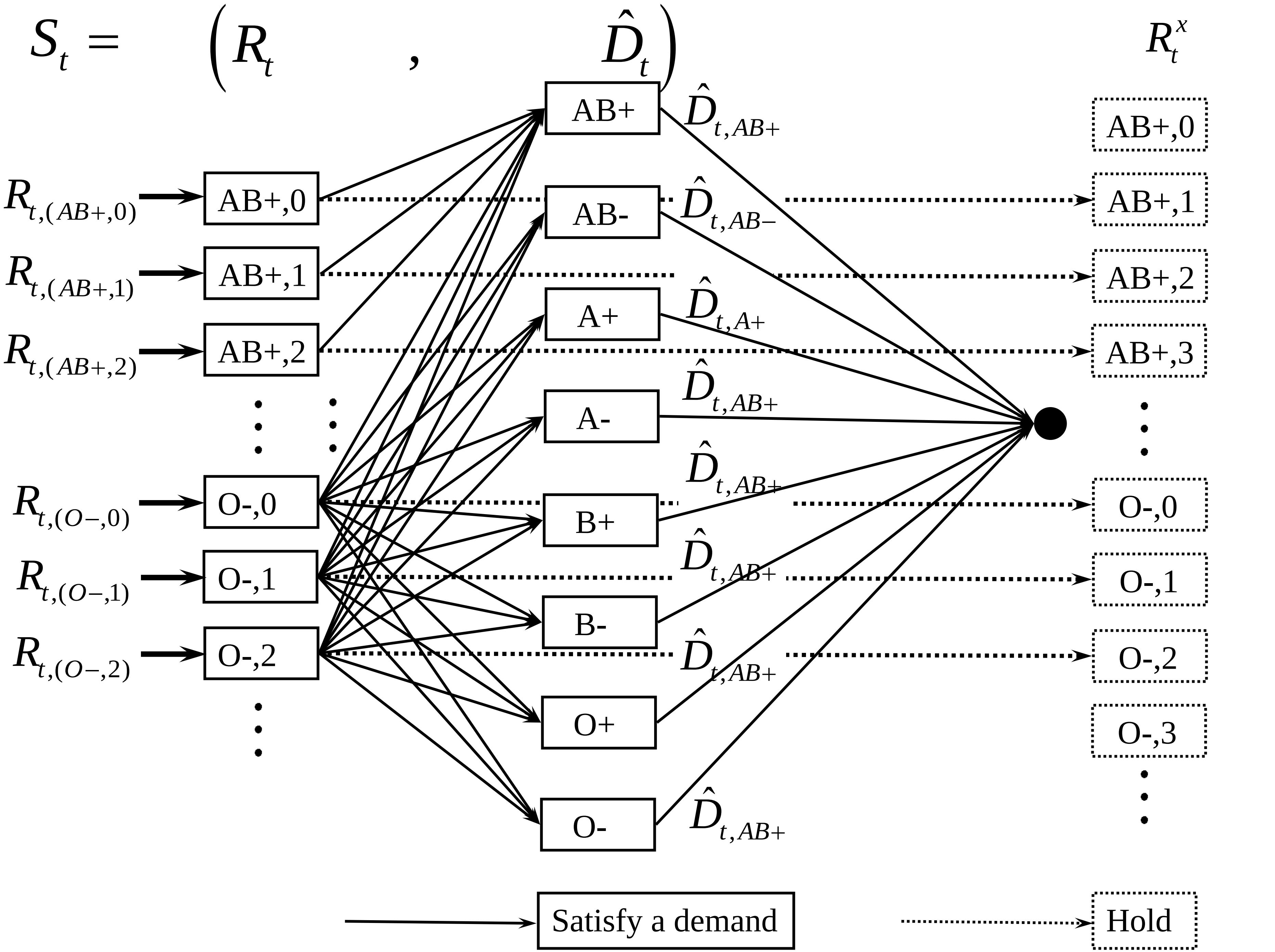
- Managing blood inventories

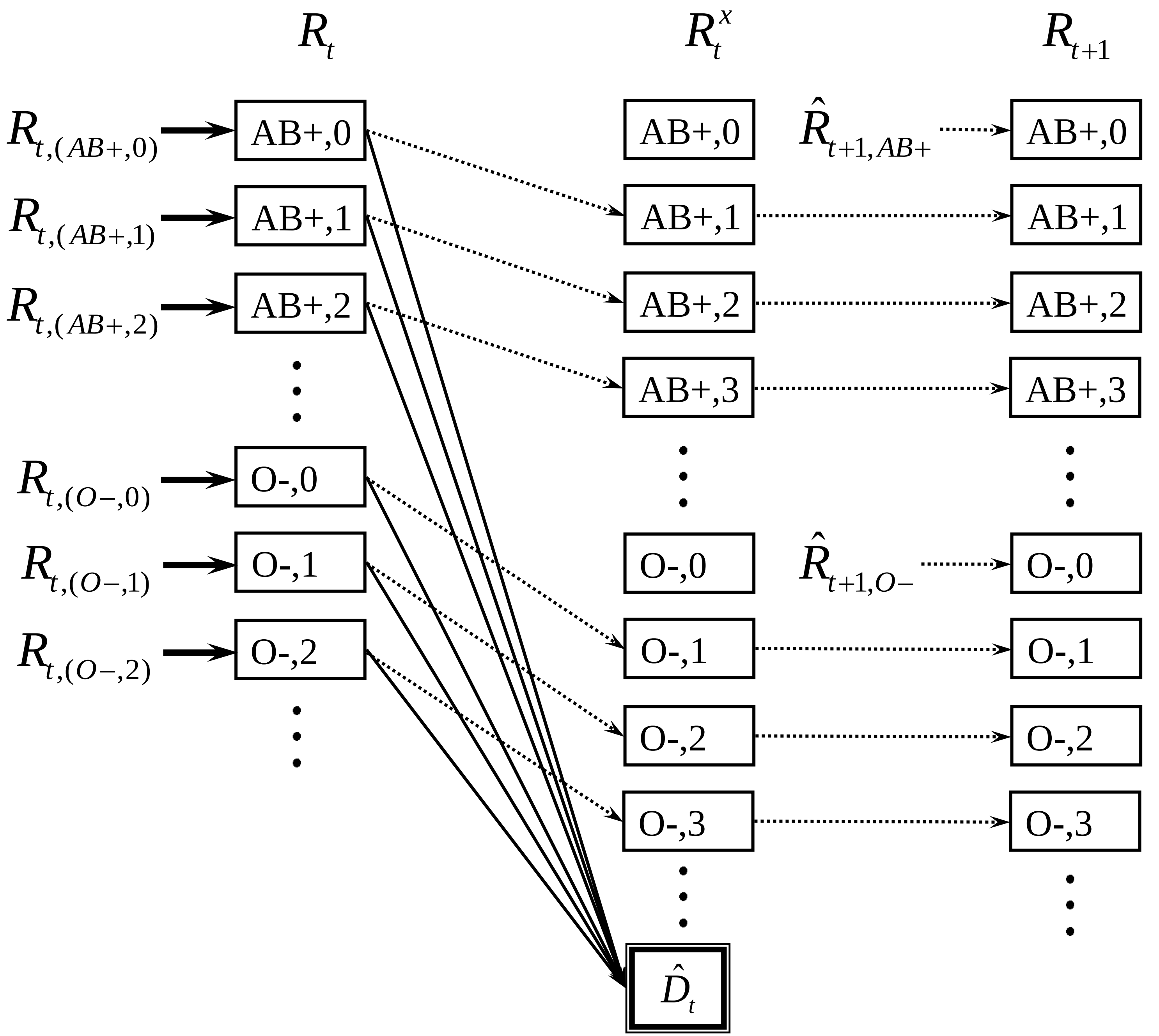


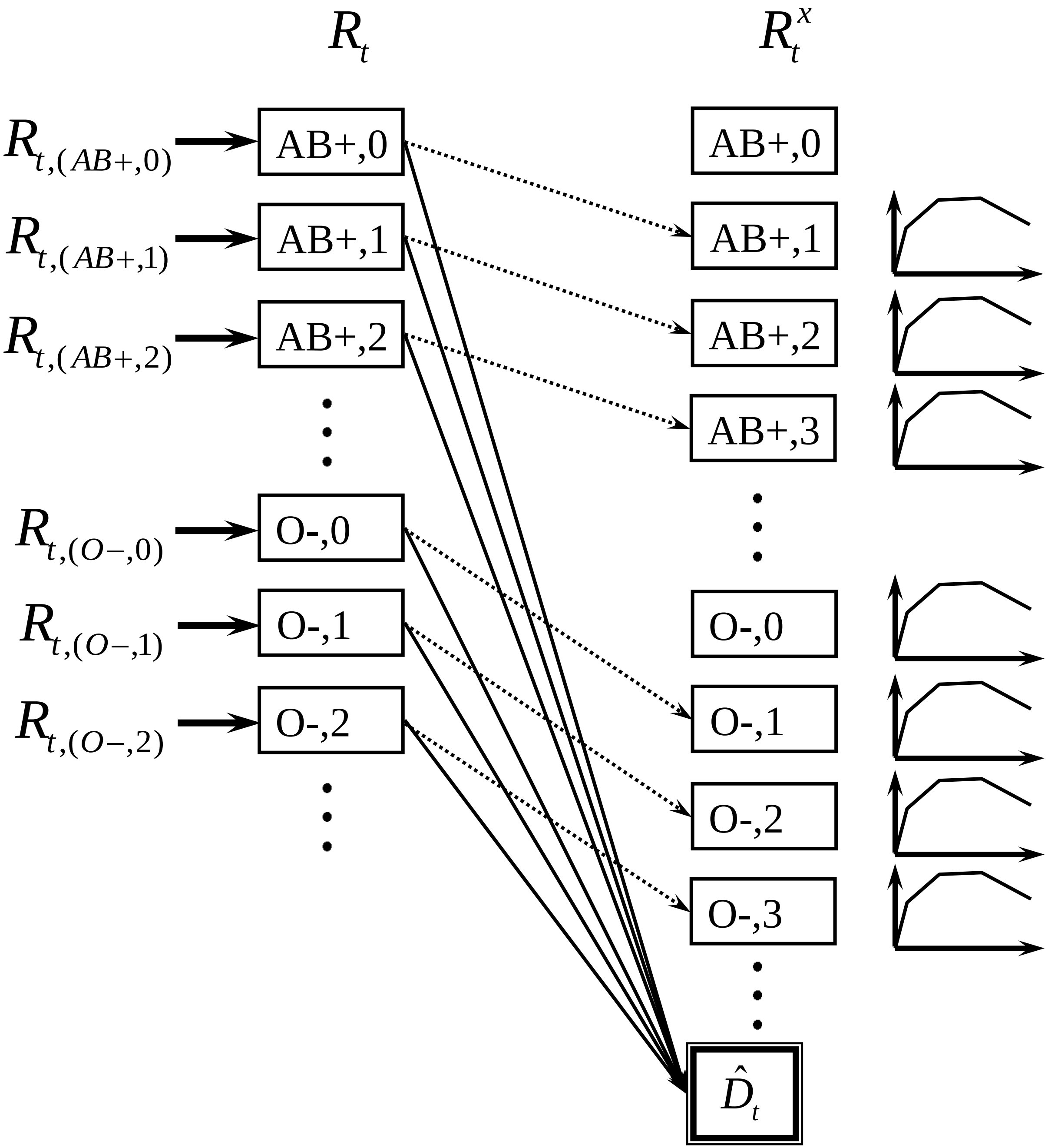
Blood management

- Managing blood inventories over time

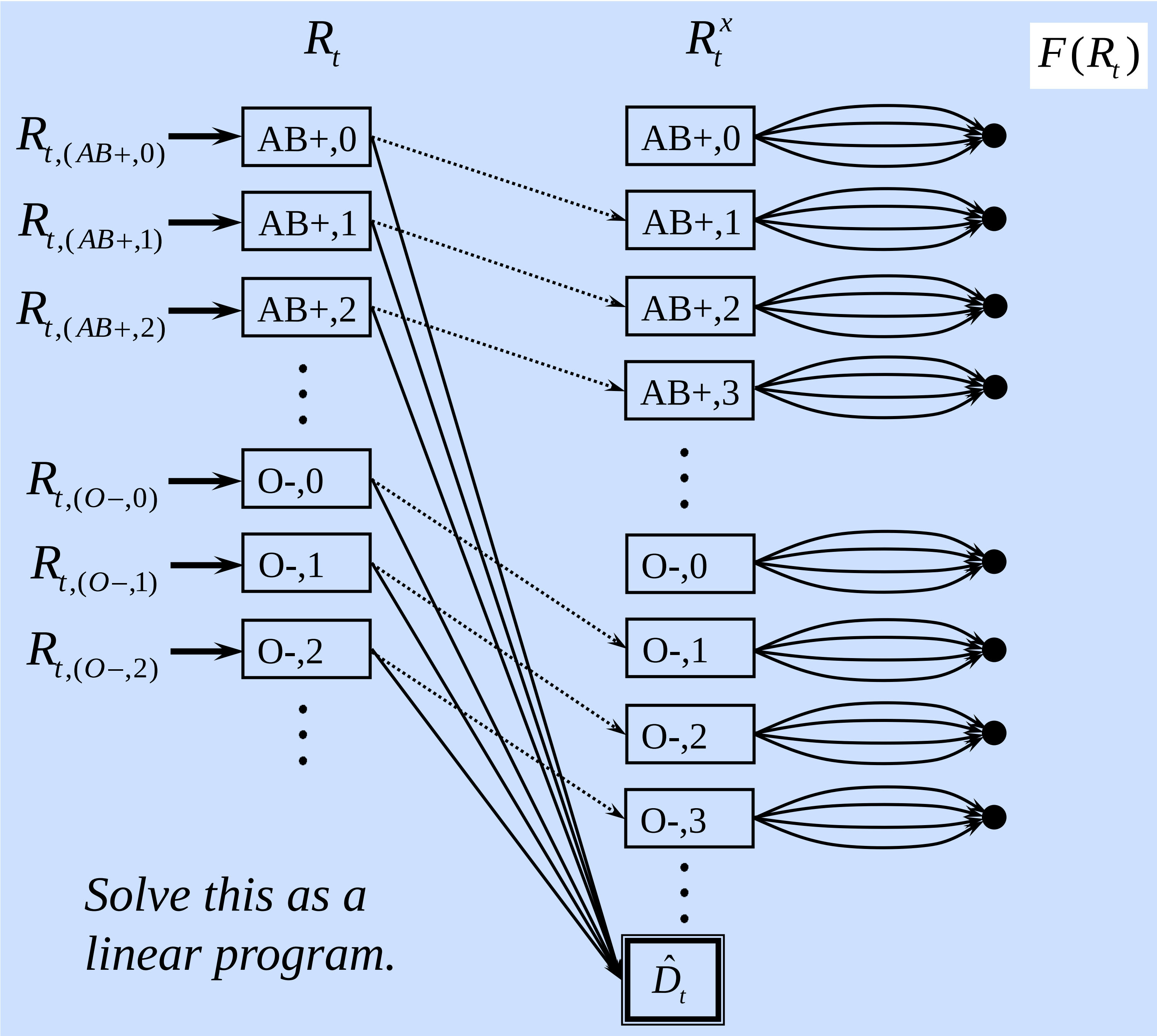



$S_t = (R_t , \hat{D}_t)$
R_t^x
$R_{t,(AB+,0)}$
$R_{t,(AB+,1)}$
$R_{t,(AB+,2)}$
$R_{t,(O-,0)}$
$R_{t,(O-,1)}$
$R_{t,(O-,2)}$
AB+,0
AB+,1
AB+,2
O-,0
O-,1
O-,2
AB+
AB-
A+
A-
B+
B-
O+
O-
$\hat{D}_{t,AB+}$
$\hat{D}_{t,AB-}$
$\hat{D}_{t,A+}$
$\hat{D}_{t,AB+}$
$\hat{D}_{t,AB+}$
$\hat{D}_{t,AB+}$
$\hat{D}_{t,AB+}$
$\hat{D}_{t,AB+}$
AB+,0
AB+,1
AB+,2
AB+,3
O-,0
O-,1
O-,2
O-,3
Satisfy a demand
Hold

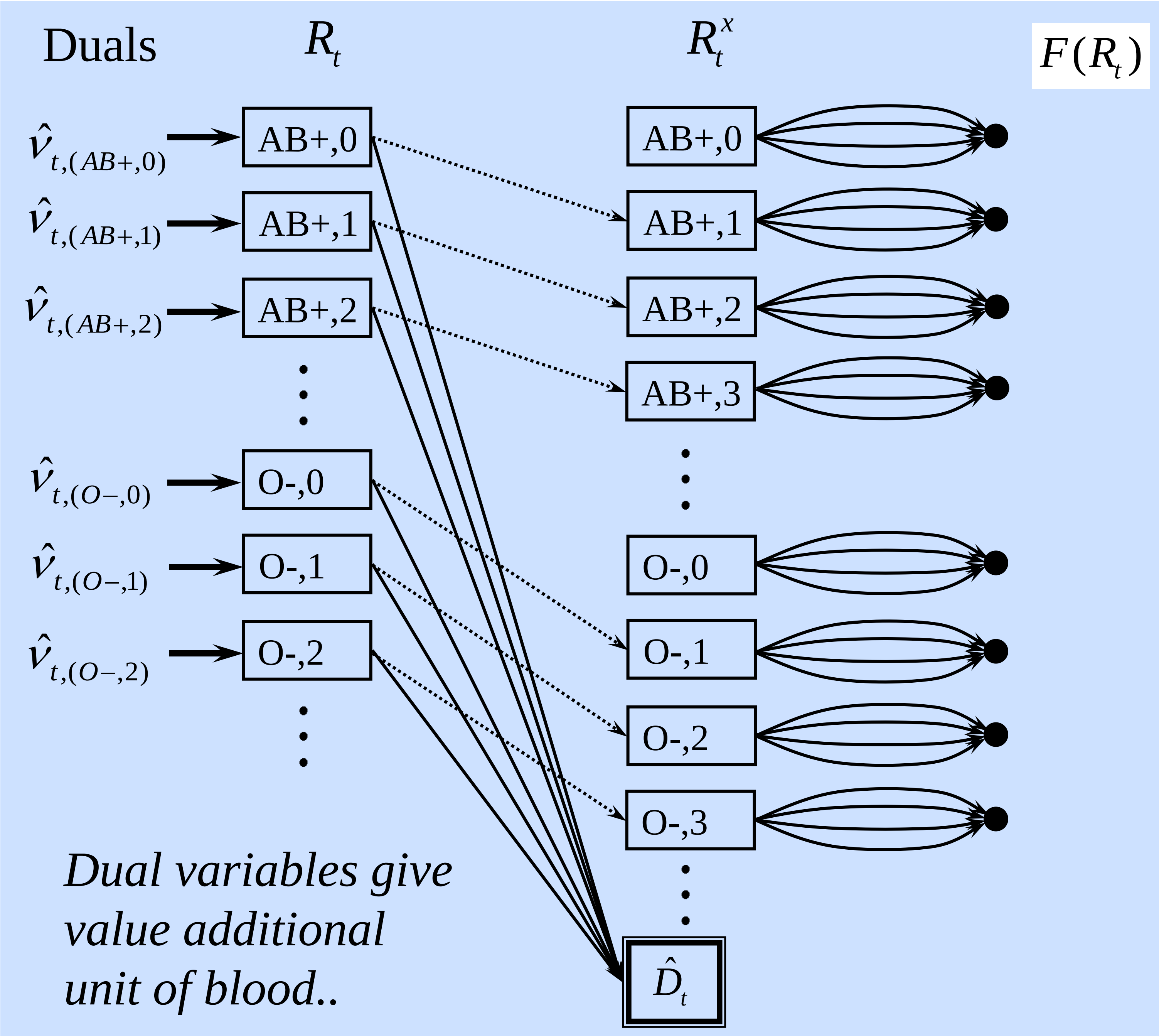

R_t
R_t^x
R_{t+1}
$R_{t,(AB+,0)}$
$R_{t,(AB+,1)}$
$R_{t,(AB+,2)}$
$R_{t,(O-,0)}$
$R_{t,(O-,1)}$
$R_{t,(O-,2)}$
AB+,0
AB+,1
AB+,2
AB+,3
O-,0
O-,1
O-,2
O-,3
$\hat{R}_{t+1,AB+}$
$\hat{R}_{t+1,O-}$
$\hat{D}_t$



R_t
R_t^x
$R_{t,(AB+,0)}$
$R_{t,(AB+,1)}$
$R_{t,(AB+,2)}$
$R_{t,(O-,0)}$
$R_{t,(O-,1)}$
$R_{t,(O-,2)}$
AB+,0
AB+,1
AB+,2
AB+,3
O-,0
O-,1
O-,2
O-,3
$\hat{D}_t$



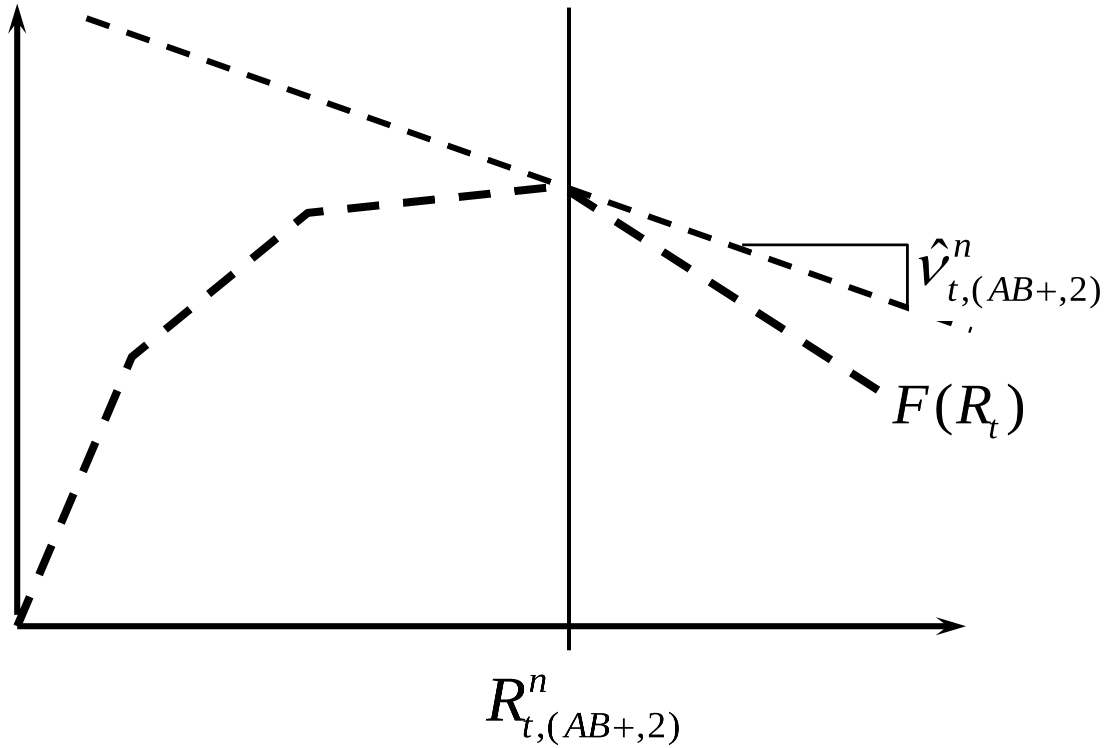
R_t
R_t^x
$F(R_t)$
$R_{t,(AB+,0)}$
$R_{t,(AB+,1)}$
$R_{t,(AB+,2)}$
$R_{t,(O-,0)}$
$R_{t,(O-,1)}$
$R_{t,(O-,2)}$
AB+,0
AB+,1
AB+,2
O-,0
O-,1
O-,2
AB+,0
AB+,1
AB+,2
AB+,3
O-,0
O-,1
O-,2
O-,3
$\hat{D}_t$
Solve this as a linear program.



Duals
R_t
R_t^x
$F(R_t)$
$\hat{v}_{t,(AB+,0)}$
$\hat{v}_{t,(AB+,1)}$
$\hat{v}_{t,(AB+,2)}$
$\hat{v}_{t,(O-,0)}$
$\hat{v}_{t,(O-,1)}$
$\hat{v}_{t,(O-,2)}$
AB+,0
AB+,1
AB+,2
O-,0
O-,1
O-,2
AB+,0
AB+,1
AB+,2
AB+,3
O-,0
O-,1
O-,2
O-,3
$\hat{D}_t$
Dual variables give value additional unit of blood..

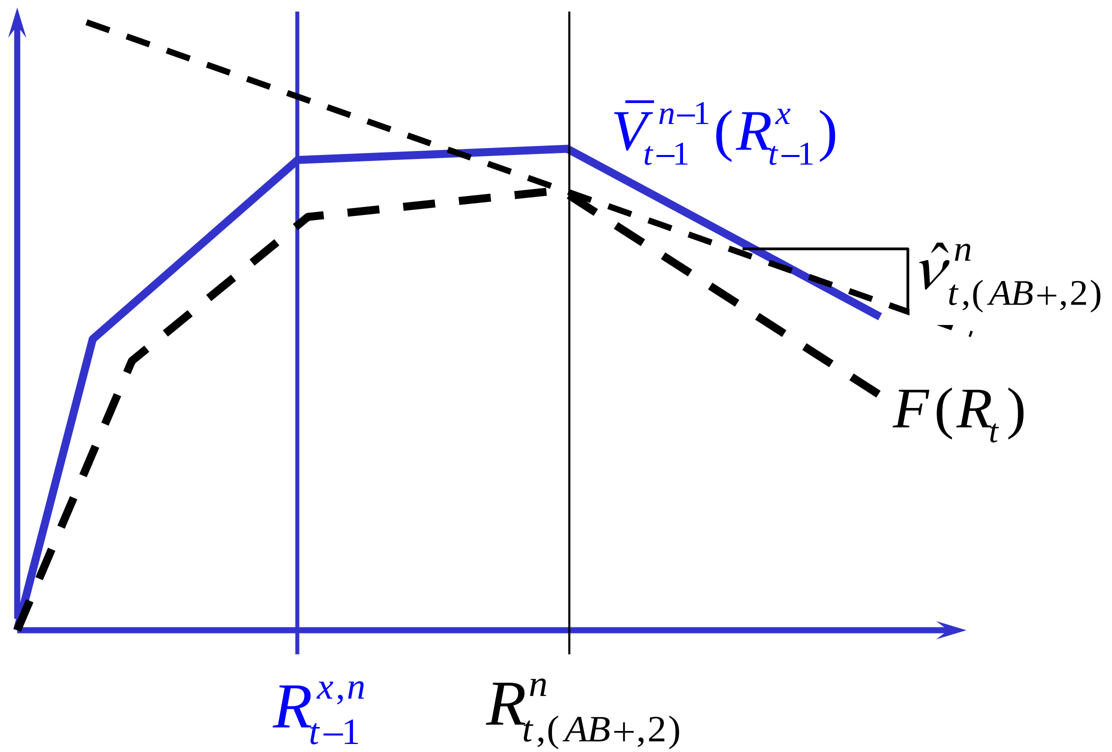
Updating the value function approximation

- Estimate the gradient at R_t^n



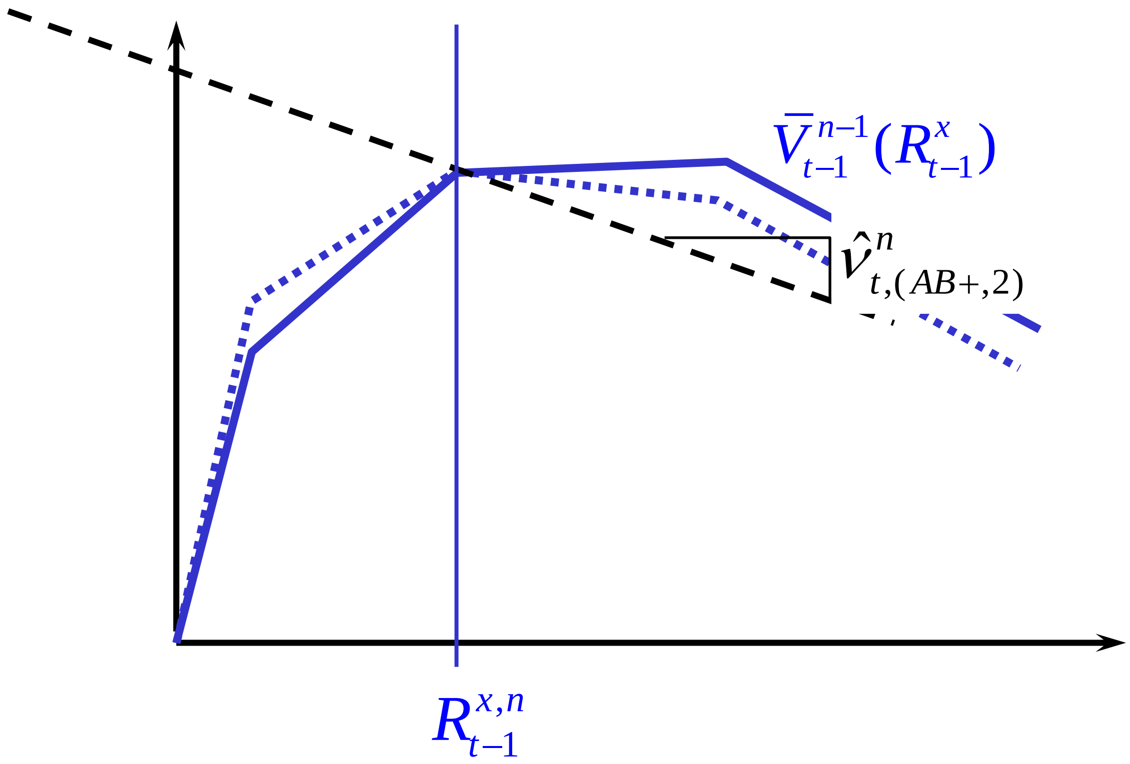
Updating the value function approximation

- Update the value function at $R_{t-1}^{x,n}$



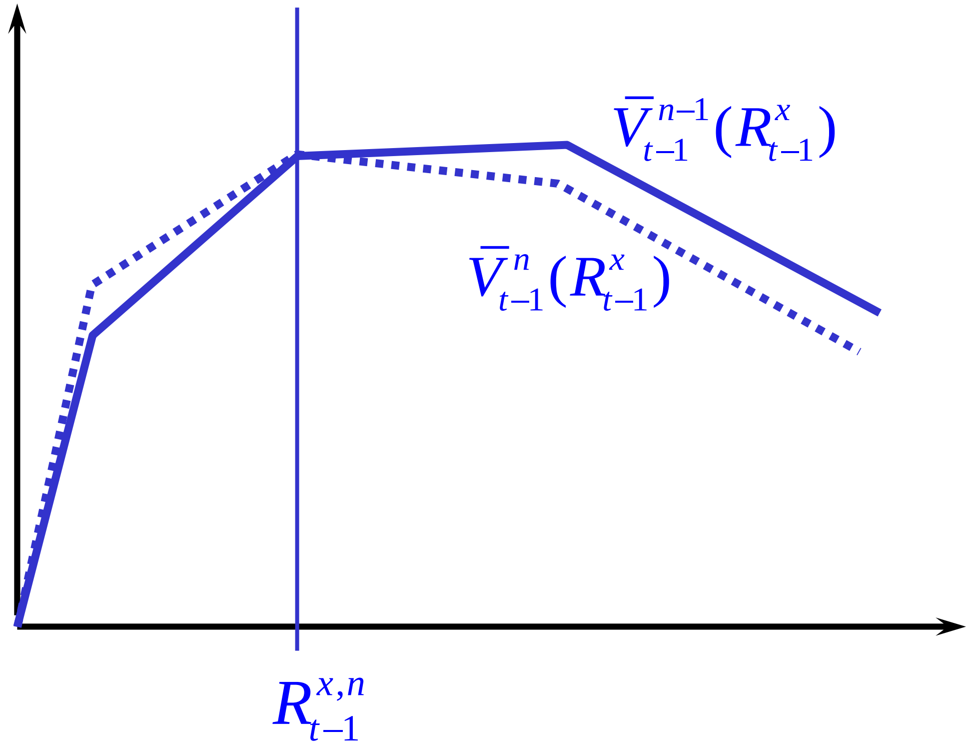
Updating the value function approximation

- Update the value function at $R_{t-1}^{x,n}$



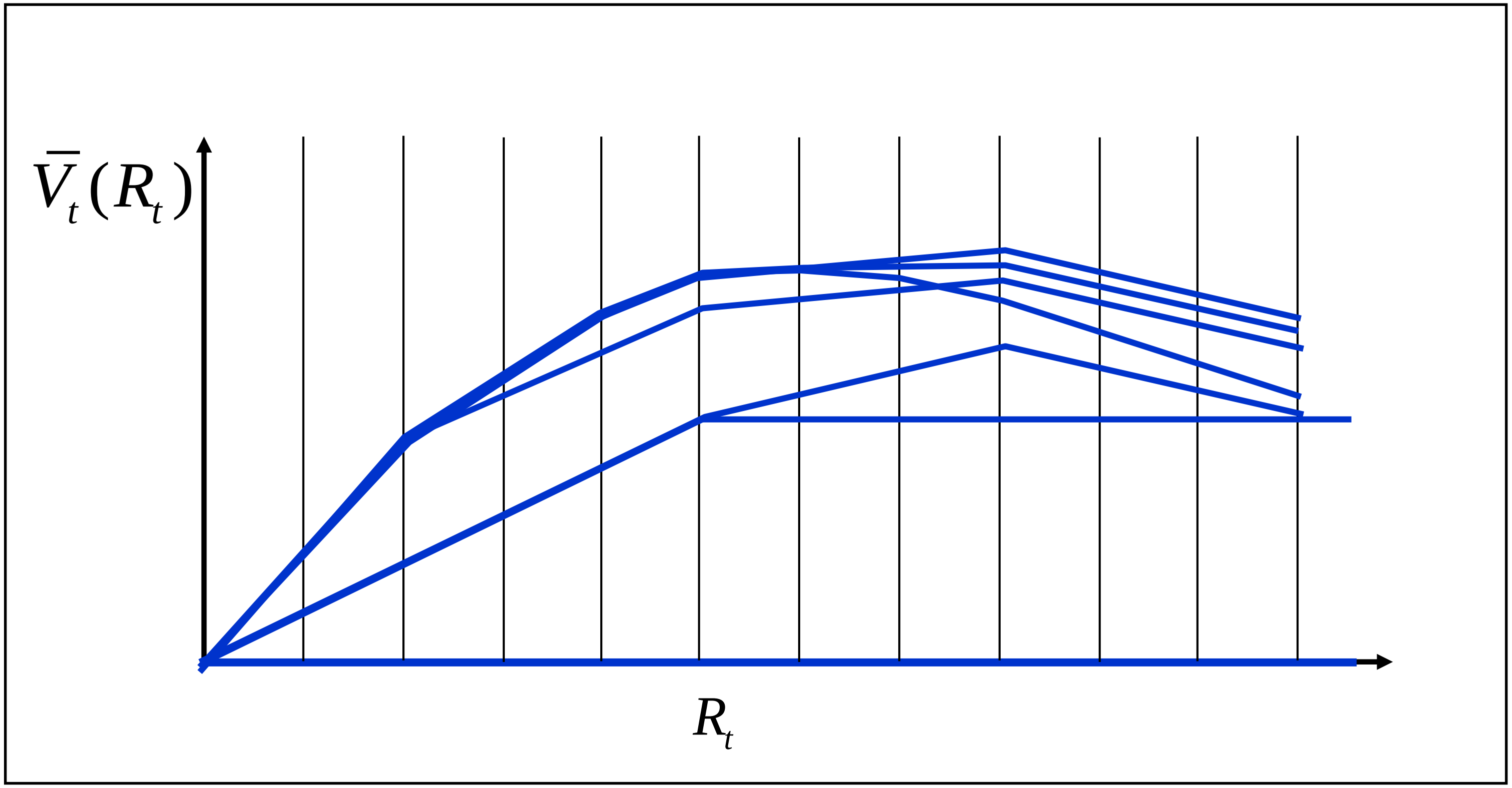
Updating the value function approximation

- Update the value function at $R_{t-1}^{x,n}$

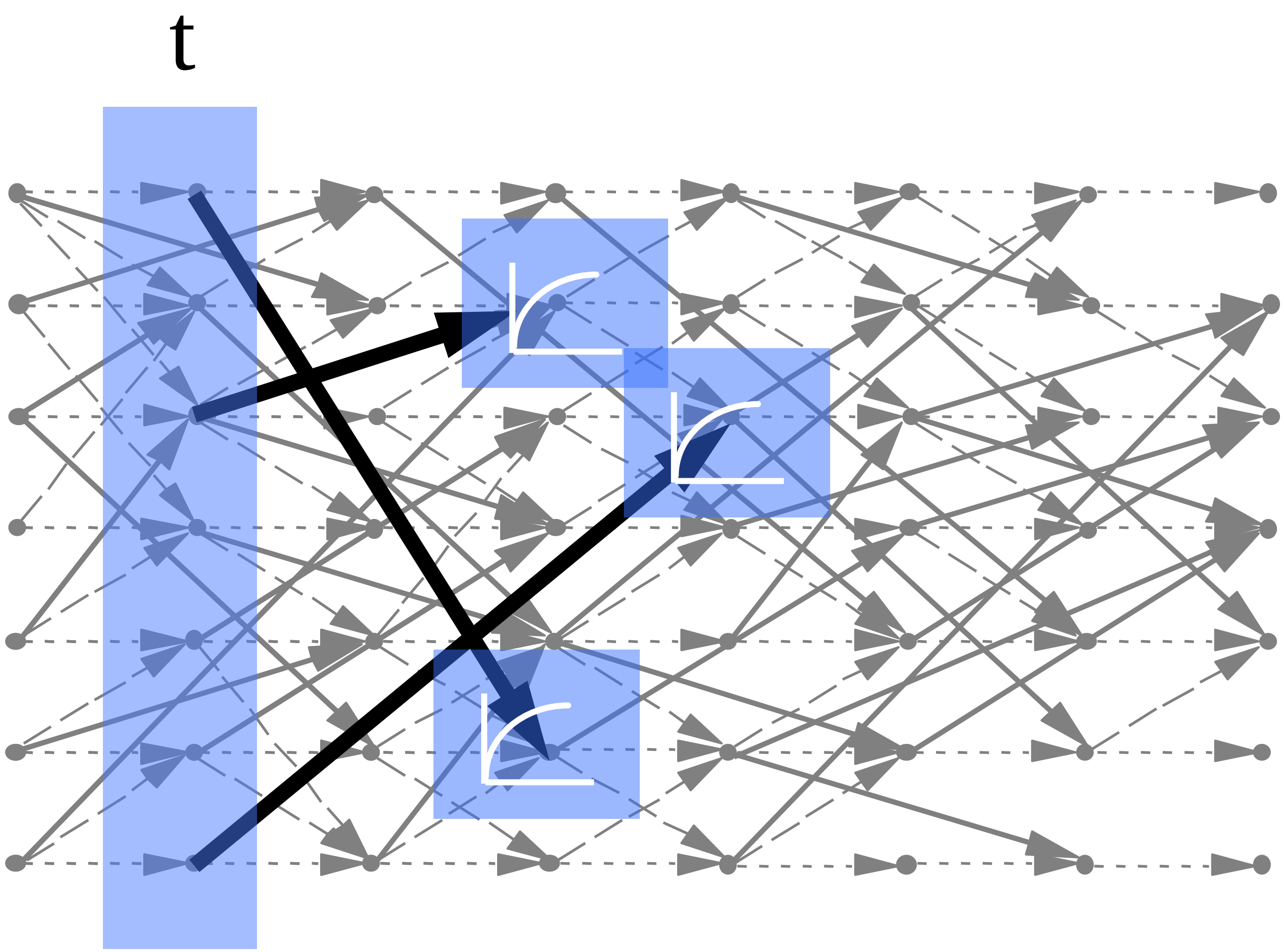


Exploiting concavity

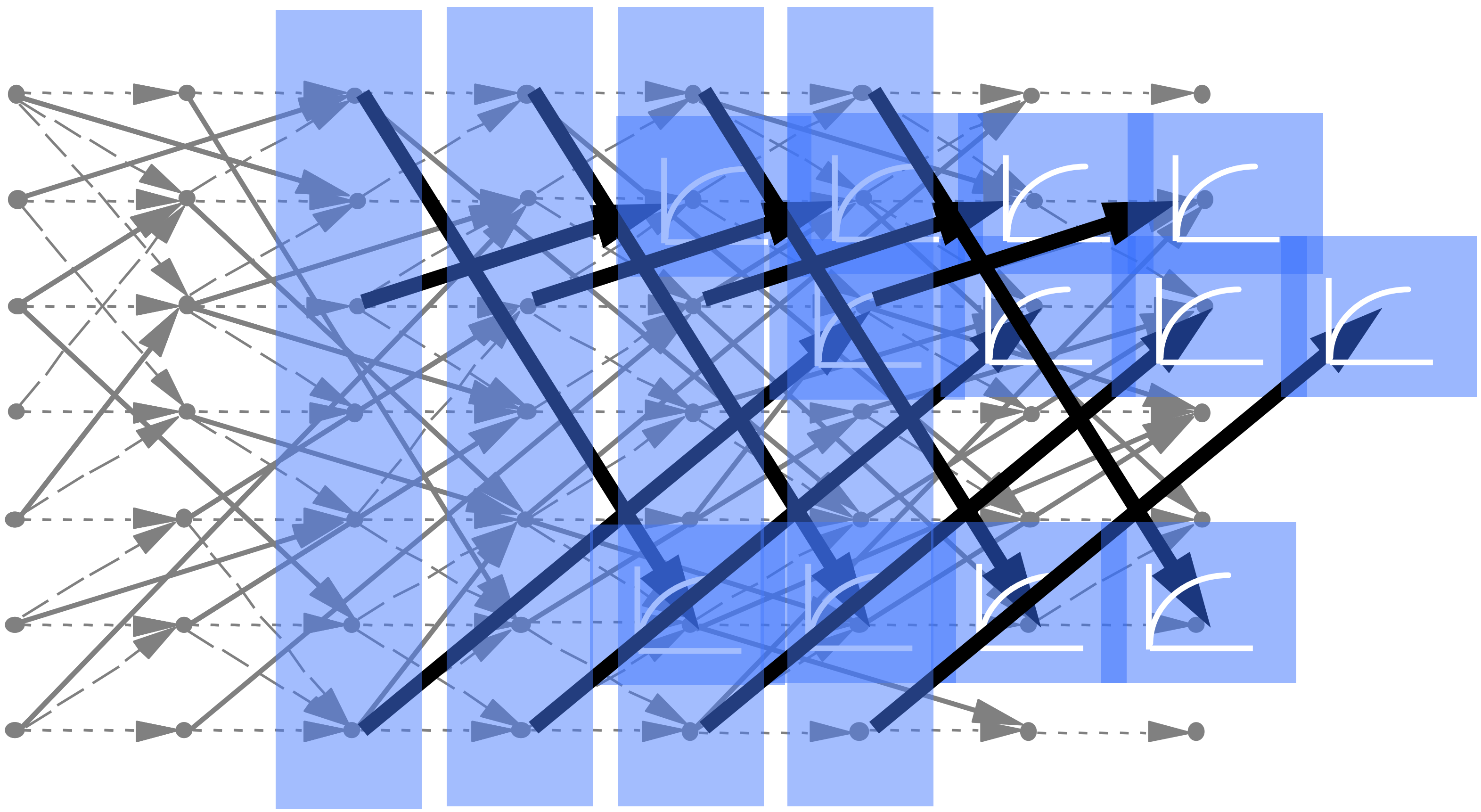
- Derivatives are used to estimate a piecewise linear approximation



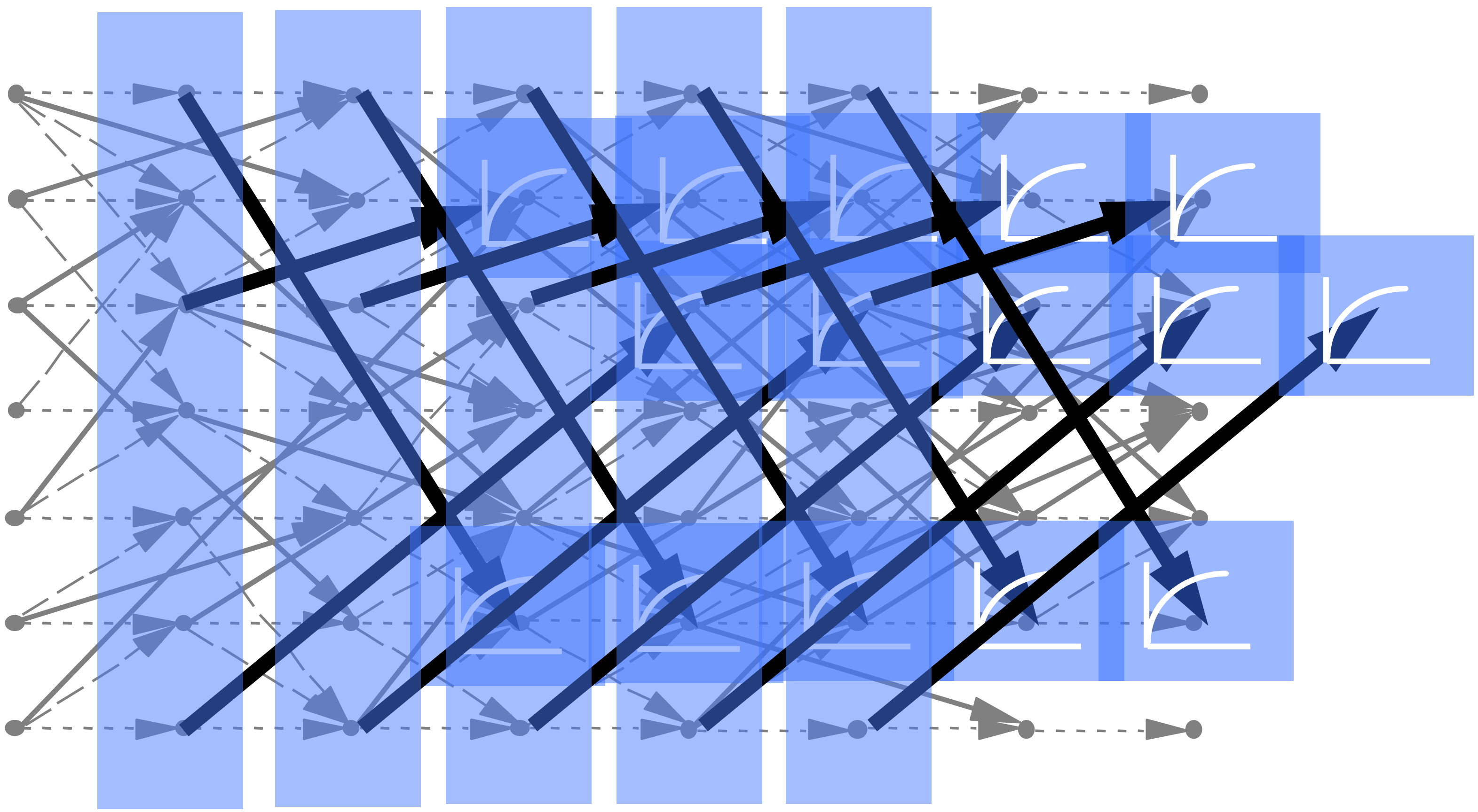
Iterative learning



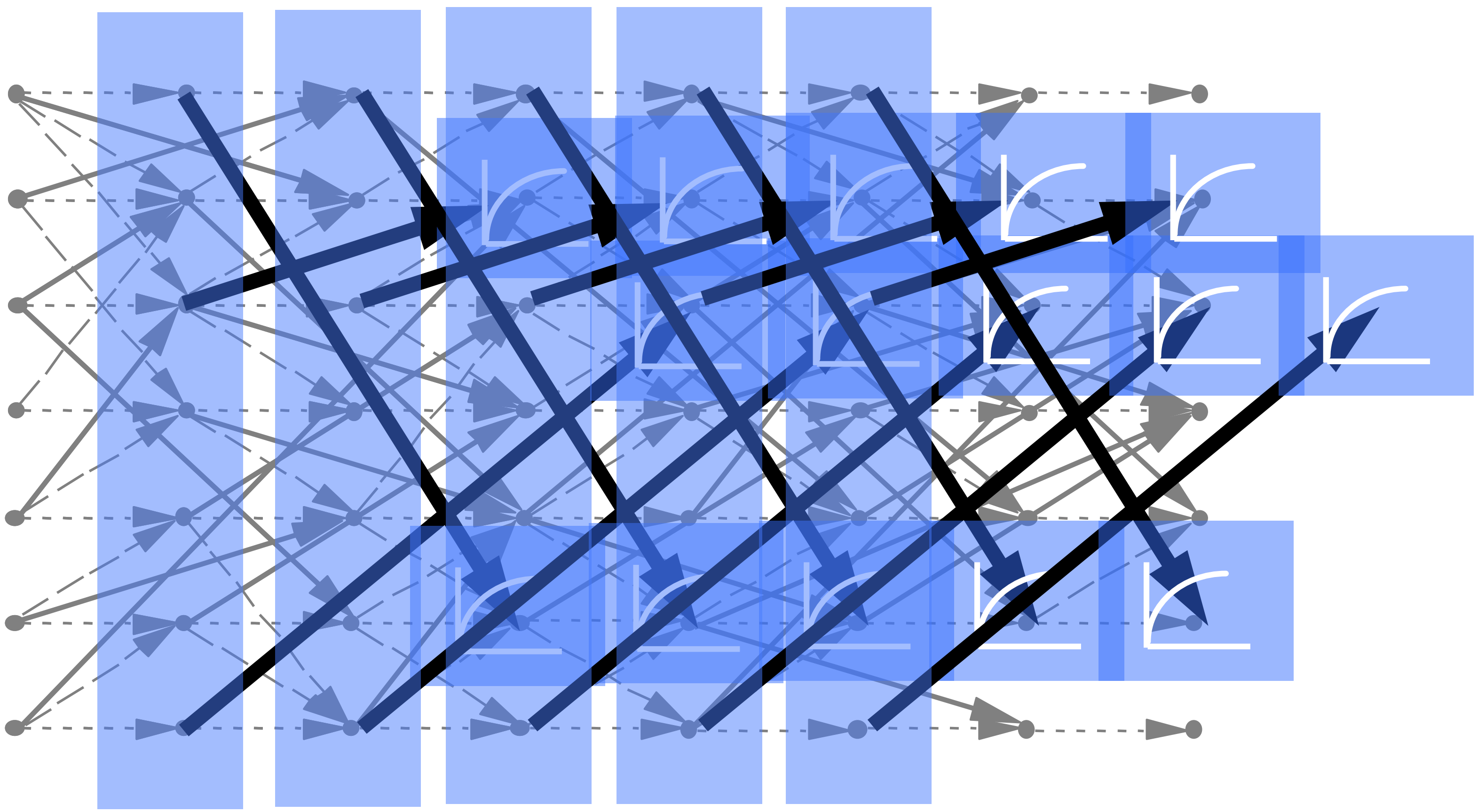
Iterative learning



Iterative learning

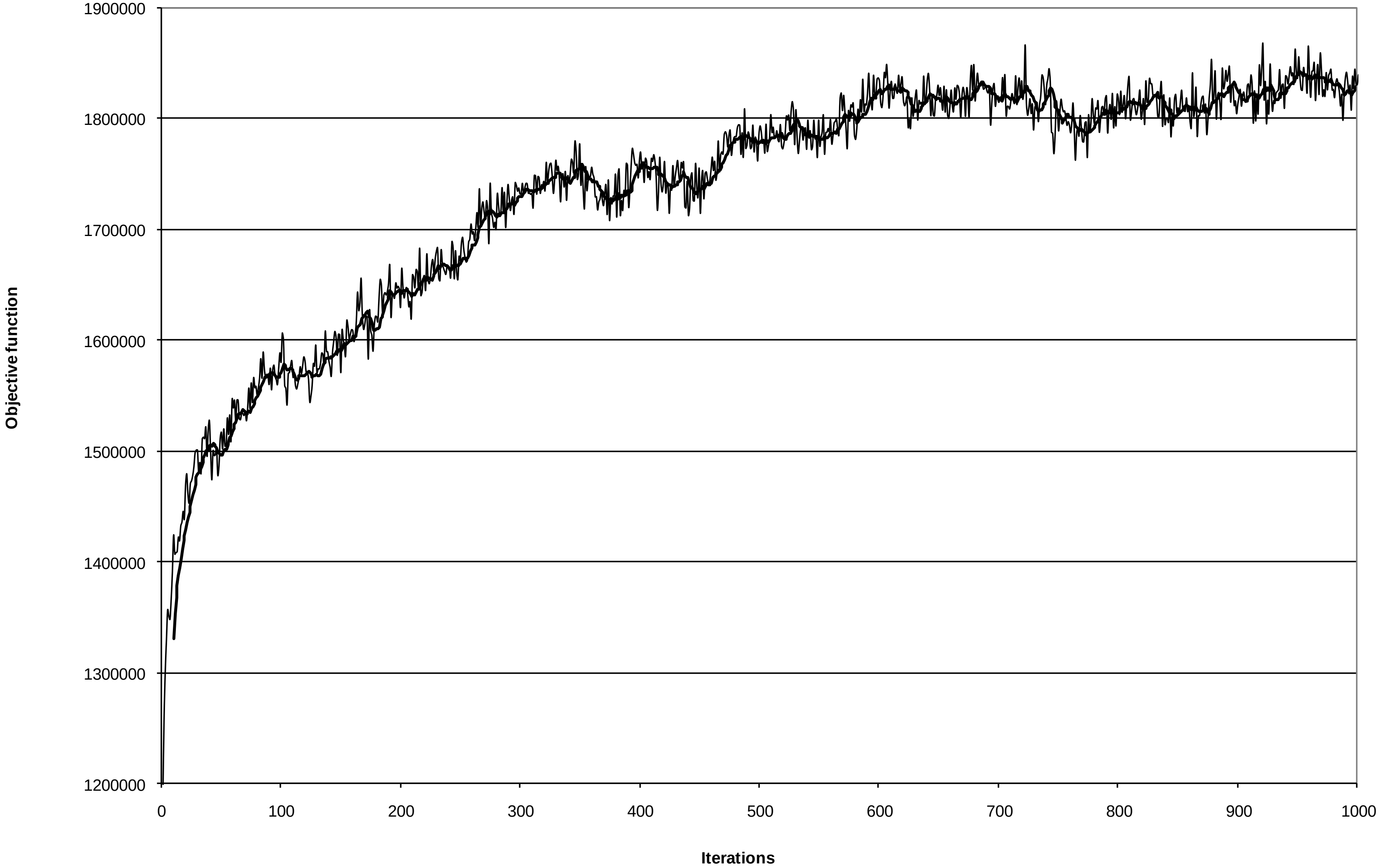


Iterative learning



Approximate dynamic programming

- … a typical performance graph.

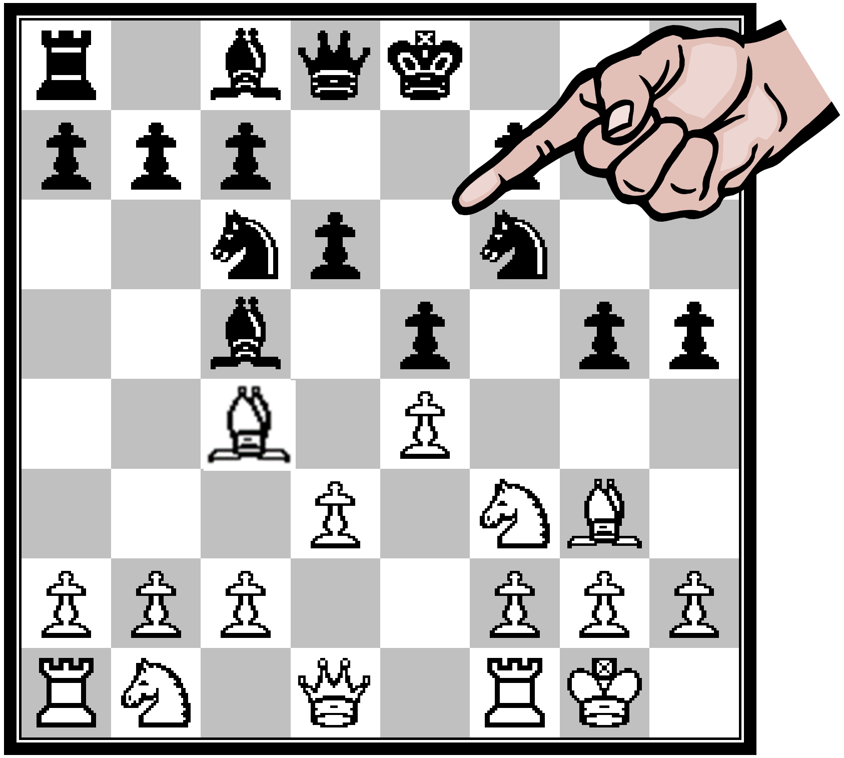


Outline

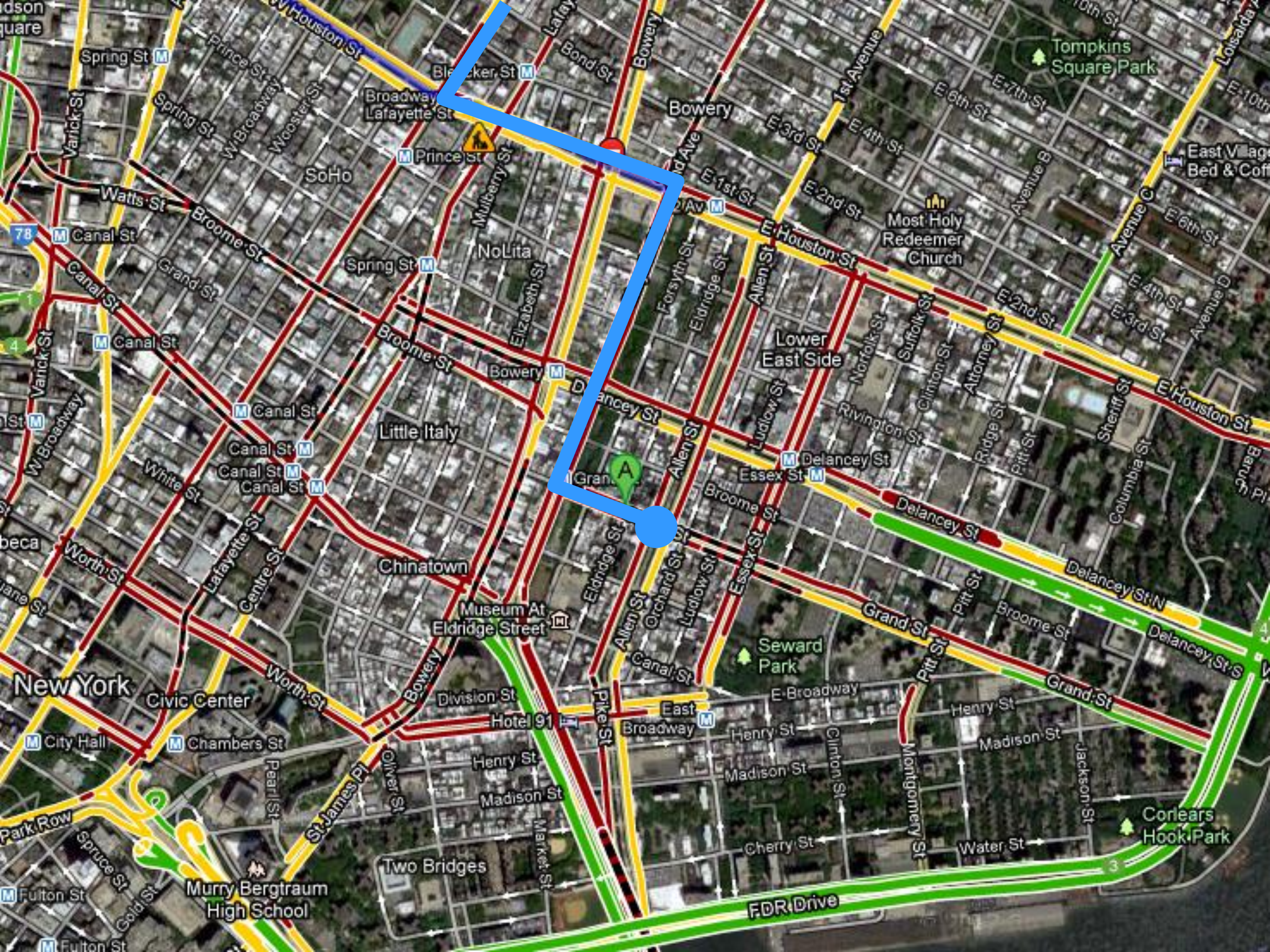
- The four classes of policies
  - » Policy function approximations (PFAs)
  - » Cost function approximations (CFAs)
  - » Value function approximations (VFAs)
  - » Direct lookahead policies (DLAs)
  - » A hybrid lookahead/CFA

Lookahead policies

- Planning your next chess move:



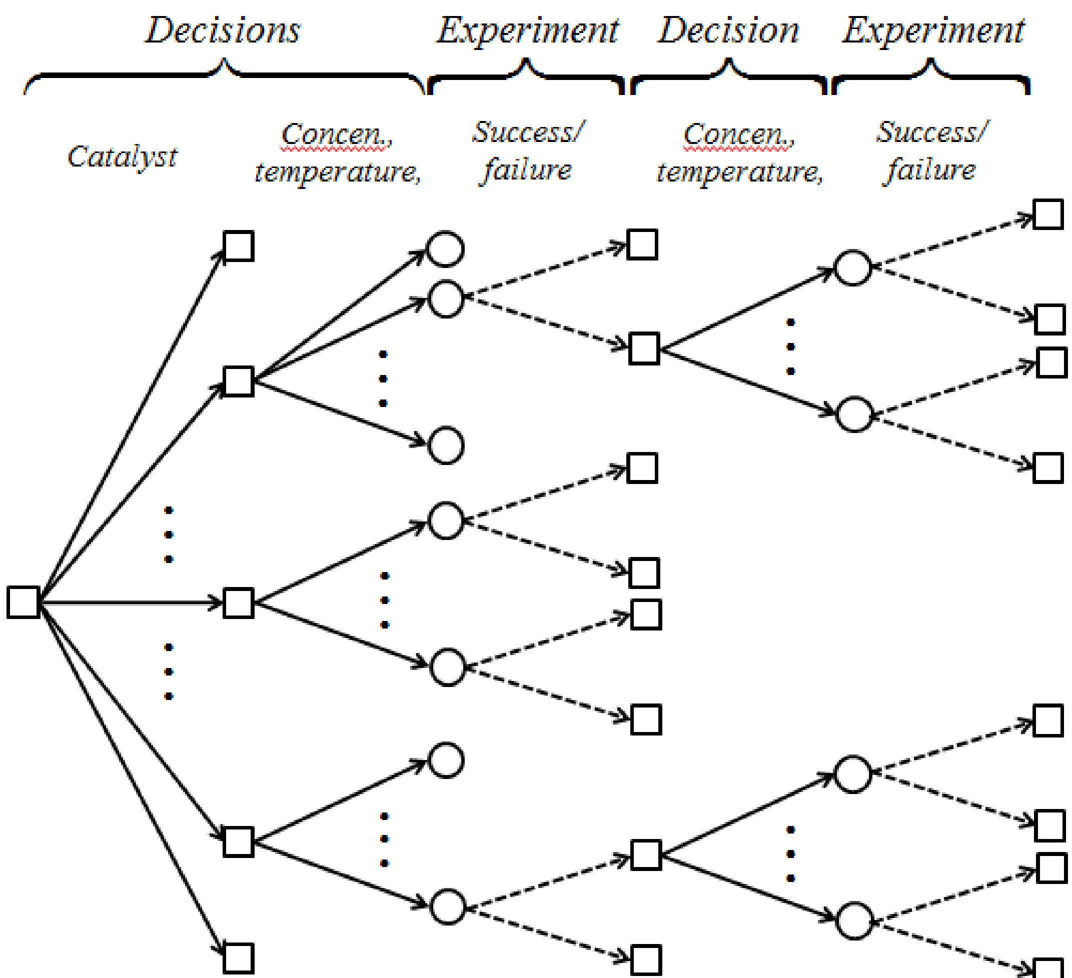
  » You put your finger on the piece while you think about moves into the future. This is a lookahead policy, illustrated for a problem with discrete actions.



Tompkins Square Park
W Houston St
Bleecker St
Broadway Lafayette St
Prince St
Spring St
SoHo
NoLita
Bowery
Lower East Side
Most Holy Redeemer Church
E Houston St
Delancey St
Essex St
Little Italy
Chinatown
Canal St
Museum At Eldridge Street
Seward Park
East Broadway
Hotel 91
New York
Civic Center
City Hall
Chambers St
Two Bridges
Murry Bergtraum High School
Corlears Hook Park
FDR Drive
East Village Bed & Coffee
Grand St
Broome St
Worth St
Division St
Henry St
Madison St
Cherry St
Water St
Fulton St
Park Row

Lookahead policies

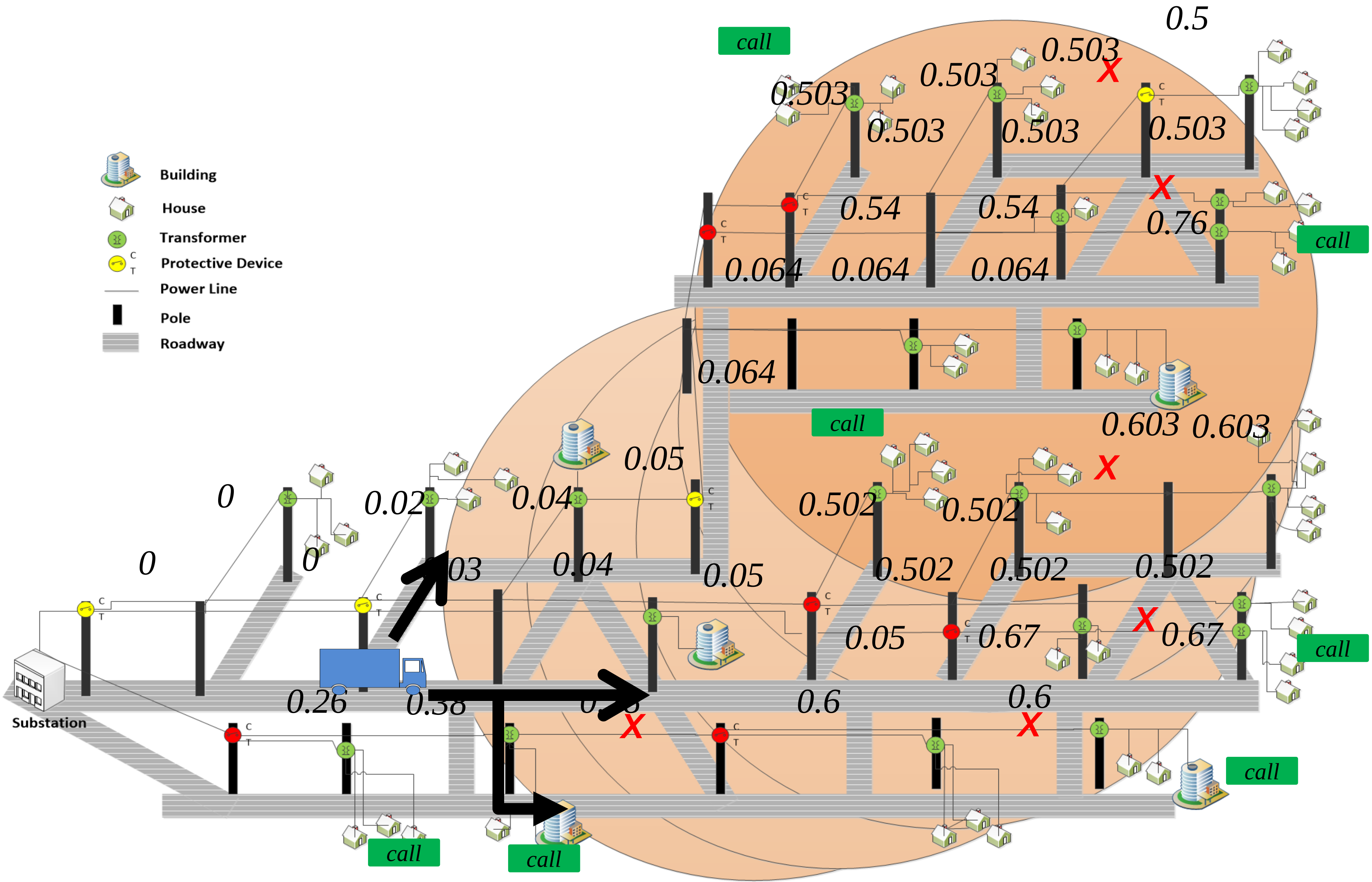
- Decision trees:



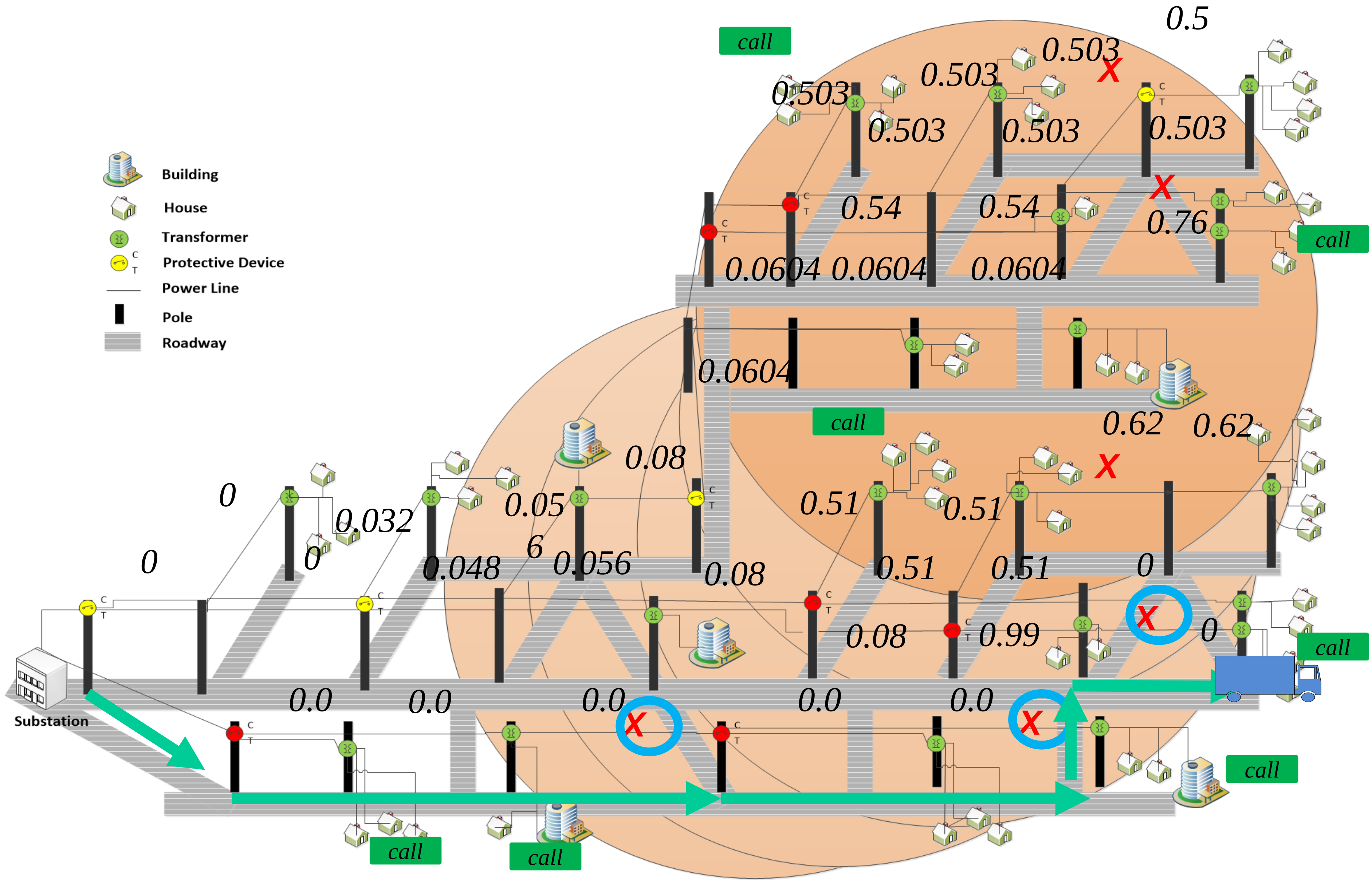
Lookahead policies

- Modeling lookahead policies
  » Lookahead policies solve a *lookahead model*, which is an approximation of the future.
  » It is important to understand the difference between the:
    • Base model – this is the model we are trying to solve by finding the best policy. This is usually some form of simulator.
    • The lookahead model, which is our approximation of the future to help us make better decisions now.
  » The base model is typically a simulator, or it might be the real world.

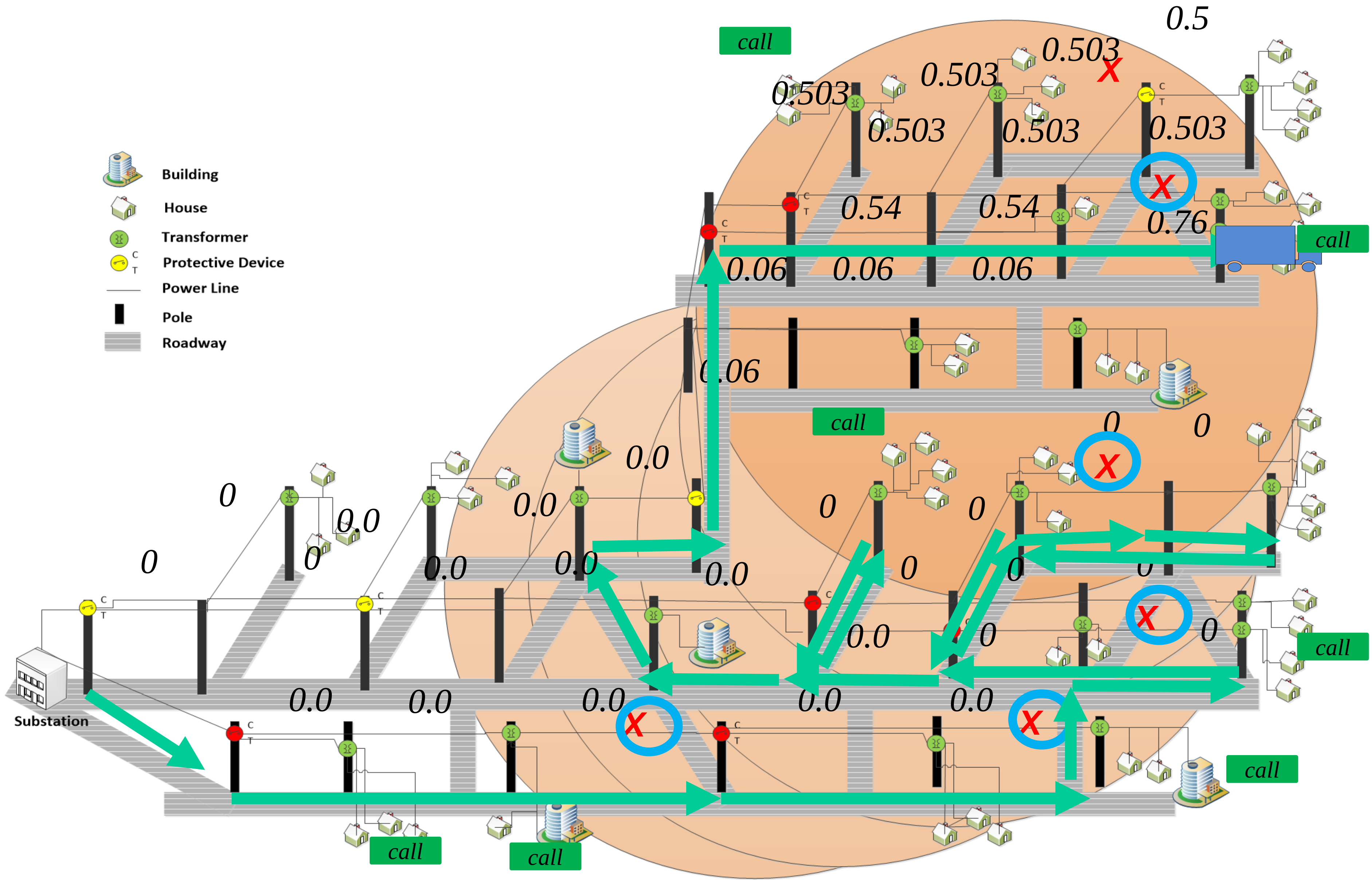
Emergency storm response

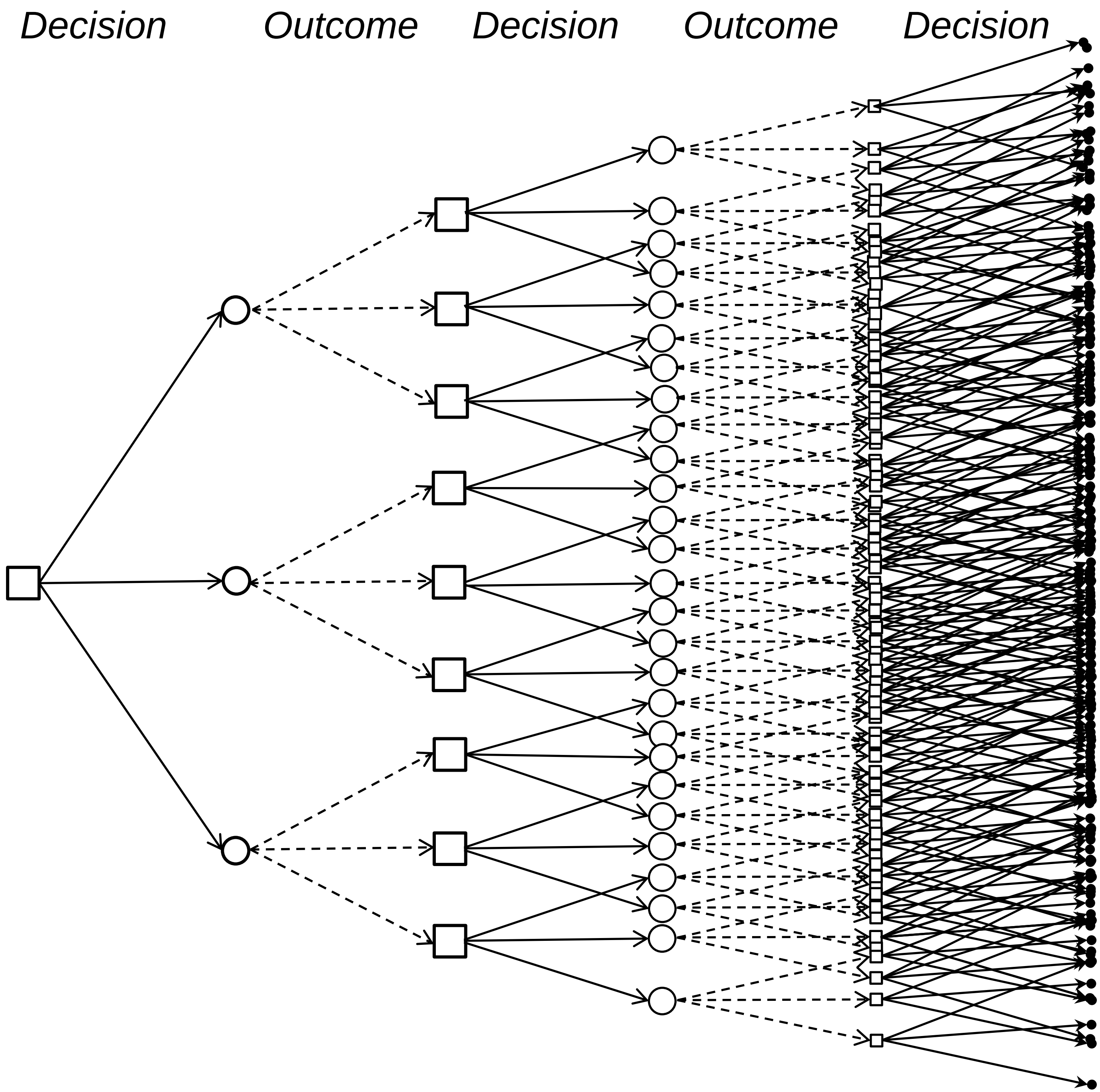


Emergency storm response



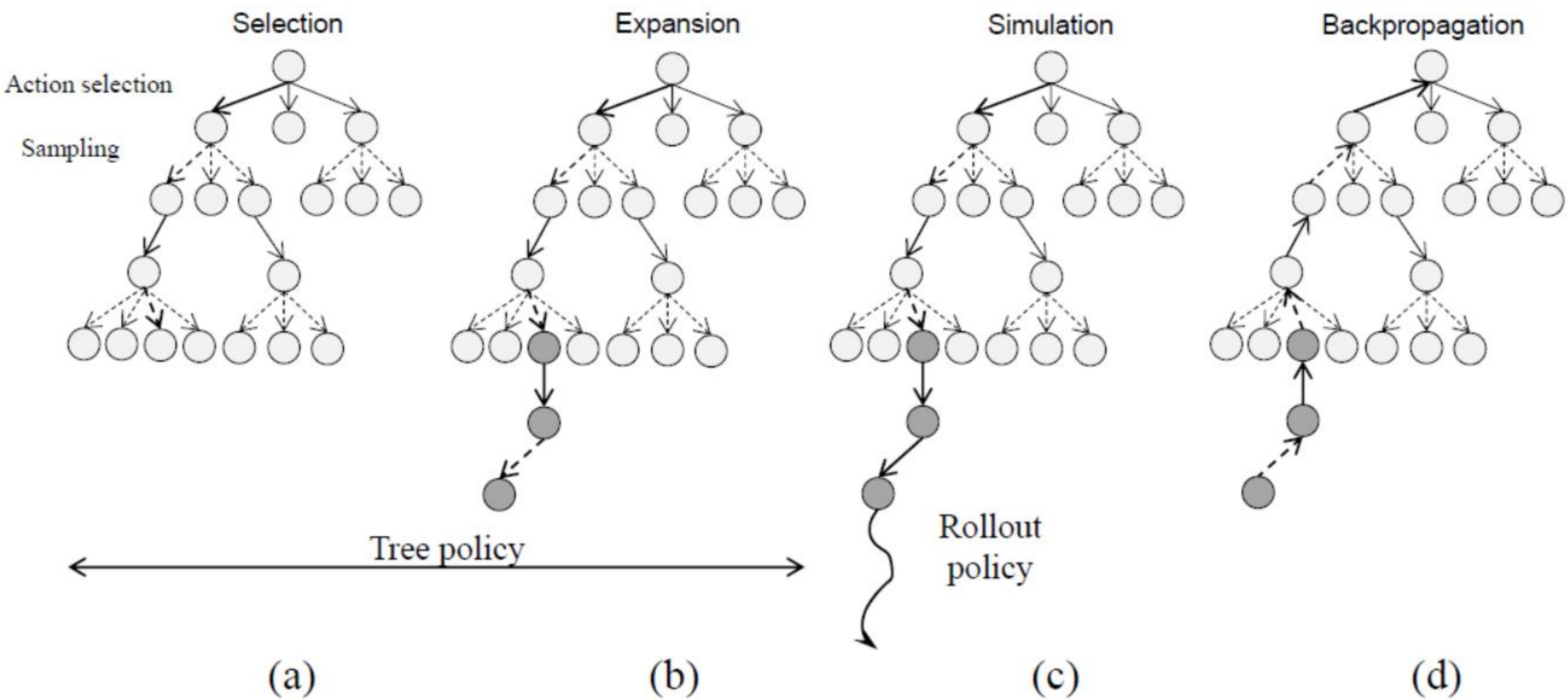
Emergency storm response



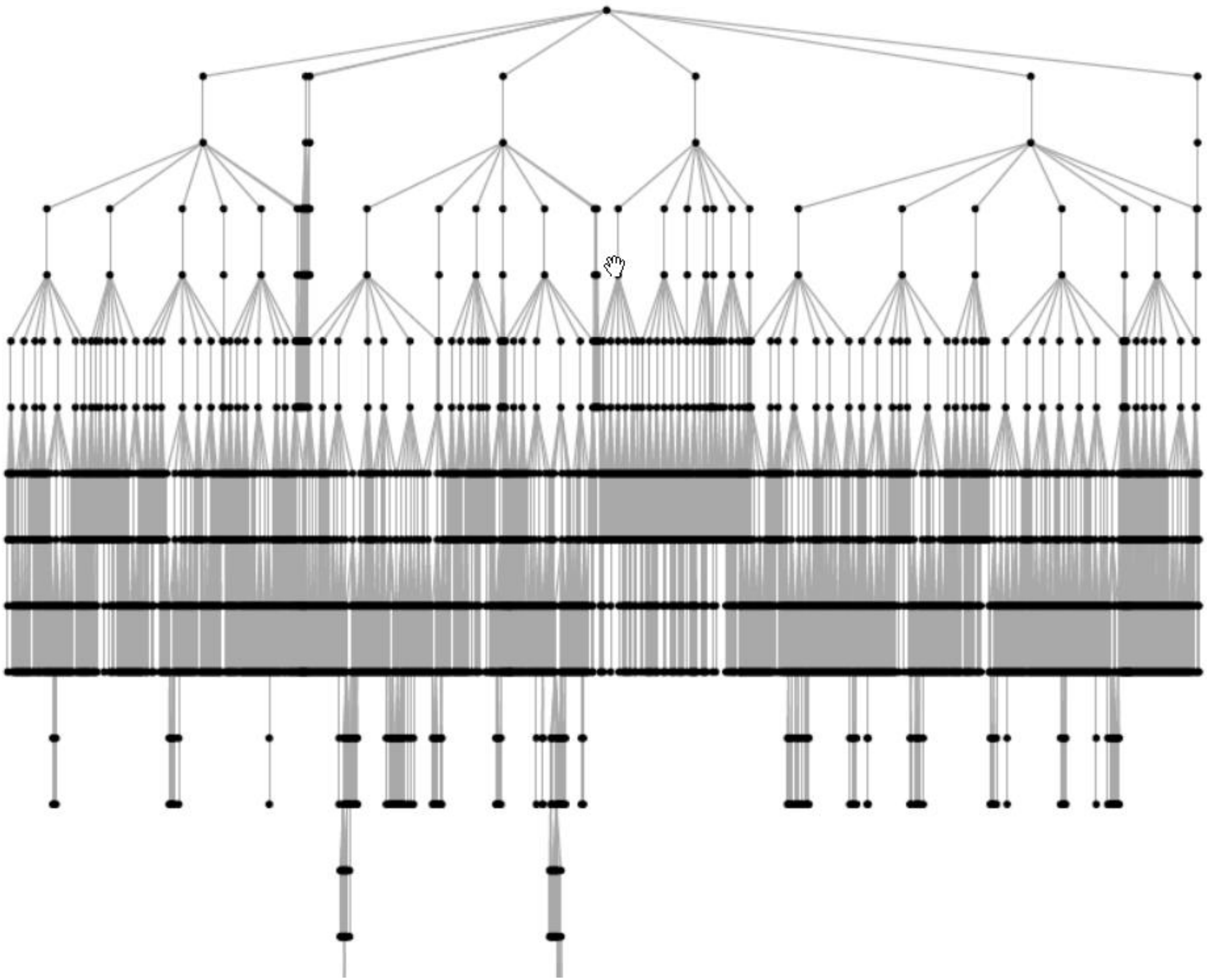

Decision
Outcome
Decision
Outcome
Decision

Lookahead policies

- Monte Carlo tree search:



C. Browne, E. Powley, D. Whitehouse, S. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis and S. Colton, "A survey of Monte Carlo tree search methods," IEEE Transactions on Computational Intelligence and AI in Games, vol. 4, no. 1, pp. 1–49, March 2012.

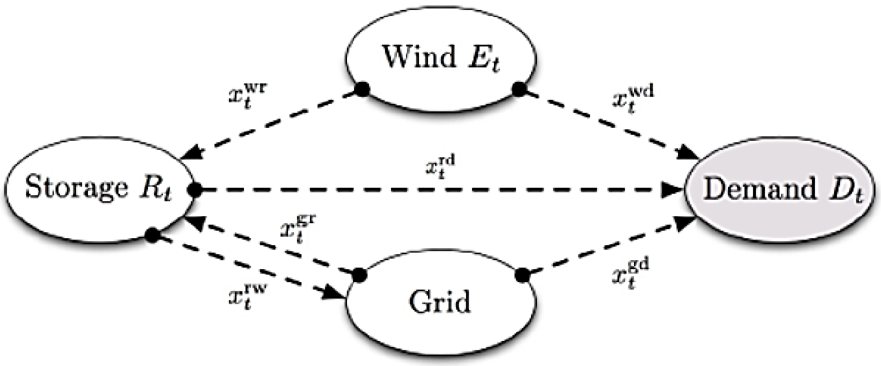


Outline

- The four classes of policies
  - » Policy function approximations (PFAs)
  - » Cost function approximations (CFAs)
  - » Value function approximations (VFAs)
  - » Direct lookahead policies (DLAs)
  - » A hybrid lookahead/CFA

Parametric cost function approximation

- An energy storage problem:



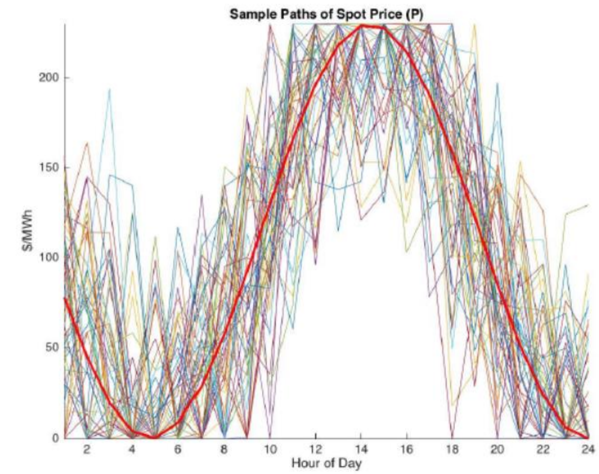


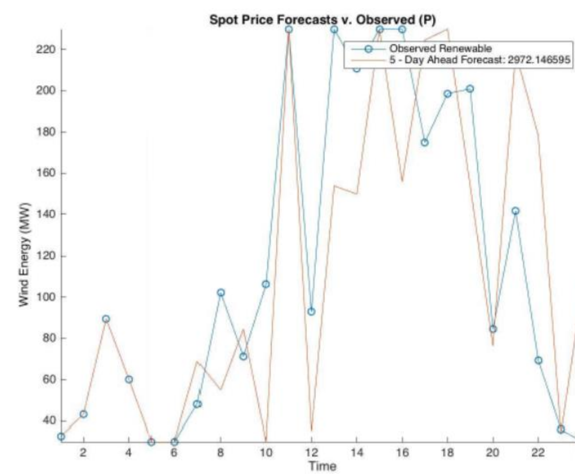
Figure: Sample paths of spot prices (P_t)

The state of the system can be represented by the following five dimensional vector,

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

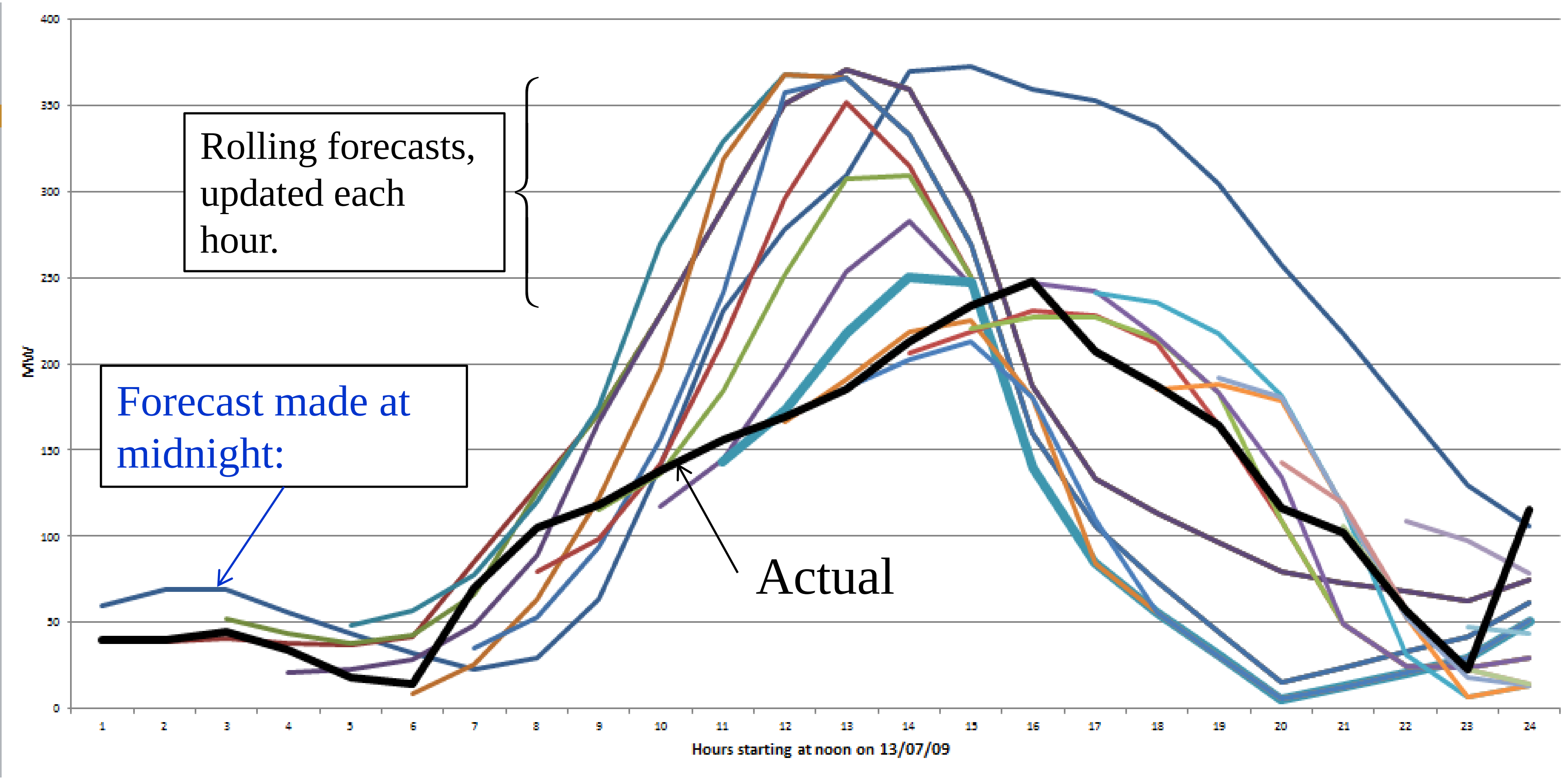
where

- $R_t \in [0, R_{\max}]$ is the level of energy in storage at time t
- E_t is the amount of energy available from wind
- P_t is the spot price of electricity
- D_t is the power demand
- G_t is the energy available from the grid



Parametric cost function approximation

- Forecasts evolve over time as new information arrives:



Parametric cost function approximation

- Benchmark policy – Deterministic lookahead

$$X_t^{\text{D-LA}}(S_t) = \underset{x_t,(\tilde{x}_{tt'},t'=t+1,\dots,t+H)}{\operatorname{argmin}} \left(C(S_t,x_t) + \left[\sum_{t'=t+1}^{t+H} \tilde{c}_{tt'}\tilde{x}_{tt'} \right] \right)$$

$$\tilde{x}_{tt'}^{wd} + \beta \tilde{x}_{tt'}^{rd} + \tilde{x}_{tt'}^{gd} \le f_{tt'}^{D}$$

$$\tilde{x}_{tt'}^{rd} + \tilde{x}_{tt'}^{rg} \le \tilde{R}_{tt'}$$

$$\tilde{x}_{tt'}^{wr} + \tilde{x}_{tt'}^{gr} \le R^{\max} - \tilde{R}_{tt'}$$

$$\boxed{\tilde{x}_{tt'}^{wr} + \tilde{x}_{tt'}^{wd} \le f_{tt'}^{E}}$$

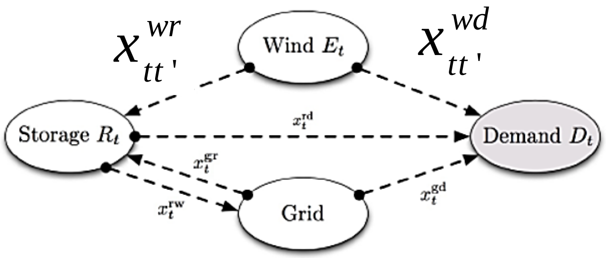
$$\tilde{x}_{tt'}^{wr} + \tilde{x}_{tt'}^{gr} \le \gamma^{charge}$$

$$\tilde{x}_{tt'}^{rd} + \tilde{x}_{tt'}^{rg} \le \gamma^{discharge}$$

Parametric cost function approximation

- Parametric cost function approximations
  - » Replace the constraint

$$\boxed{\tilde{x}^{wr}_{tt'} + \tilde{x}^{wd}_{tt'} \le f^E_{tt'}}$$



with:

  - » Lookup table modified forecasts (one adjustment term for each time $\tau = t' - t$ in the future):

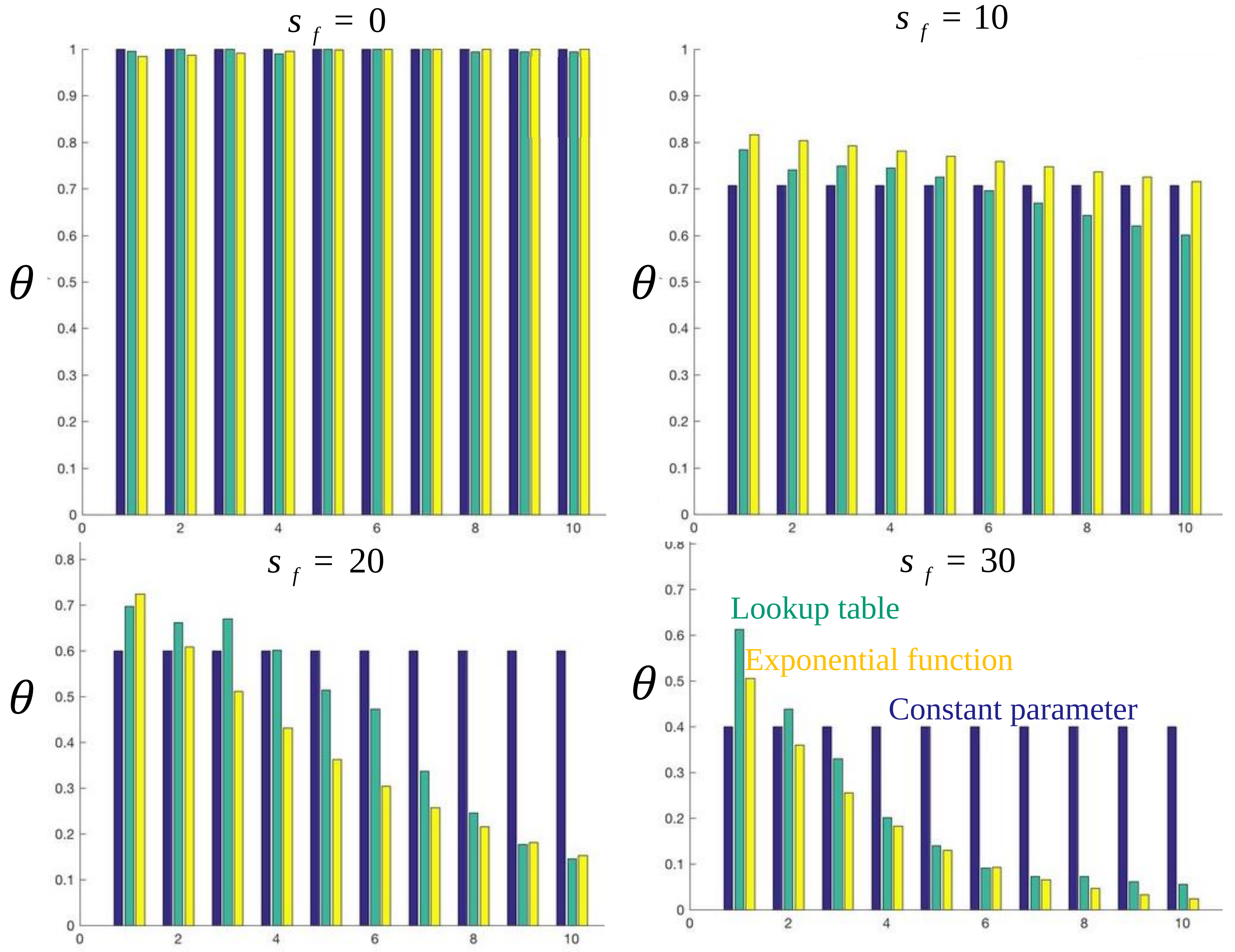
$$x^{wr}_{tt'} + x^{wd}_{tt'} \le \theta_{t'-t} f^E_{tt'}$$

  - » Exponential function for adjustments (just two parameters)

$$x^{wr}_{tt'} + x^{wd}_{tt'} \le \theta_1 e^{\theta_2 (t'-t)} f^E_{tt'}$$

  - » Constant adjustment (one parameter)

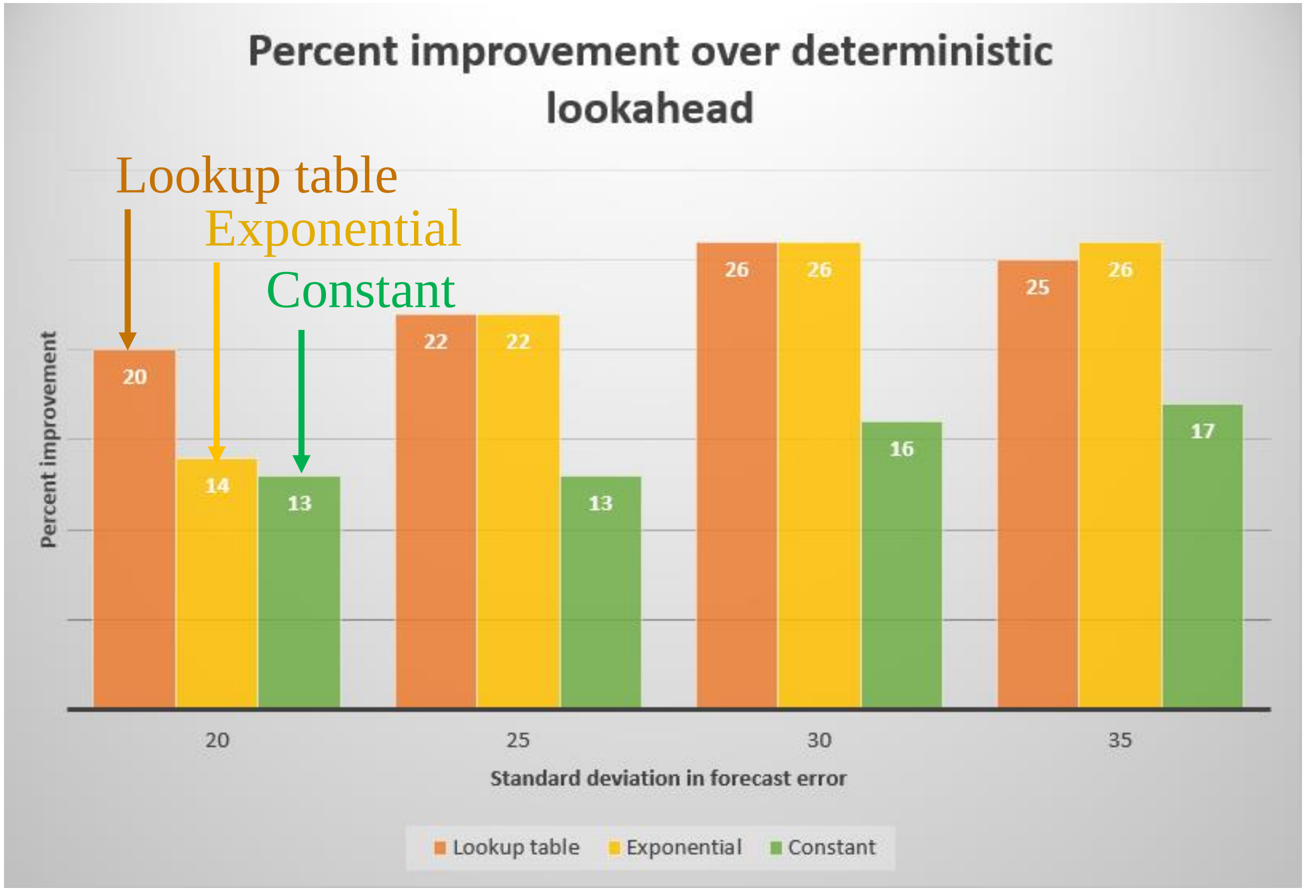
$$x^{wr}_{tt'} + x^{wd}_{tt'} \le \theta f^E_{tt'}$$



$s_f = 0$
$s_f = 10$
$s_f = 20$
$s_f = 30$
θ
Lookup table
Exponential function
Constant parameter

Parametric cost function approximation

- Improvement over deterministic benchmark:



An energy storage problem

- Consider a basic energy storage problem:

» We are going to show that with minor variations in the characteristics of this problem, we can make *each* class of policy work best.

An energy storage problem

- We can create distinct flavors of this problem:
  - » Problem class 1 – Best for PFAs
    - Highly stochastic (heavy tailed) electricity prices
    - Stationary data
  - » Problem class 2 – Best for CFAs
    - Stochastic prices and wind (but not heavy tailed)
    - Stationary data
  - » Problem class 3 - Best for VFAs
    - Stochastic wind and prices (but not too random)
    - Time varying loads, but inaccurate wind forecasts
  - » Problem class 4 – Best for deterministic lookaheads
    - Relatively low noise problem with accurate forecasts
  - » Problem class 5 – A hybrid policy worked best here
    - Stochastic prices and wind, nonstationary data, noisy forecasts.

An energy storage problem

- The policies
  - » The PFA:
    - Charge battery when price is below p1
    - Discharge when price is above p2
  - » The CFA
    - Optimize over a horizon H; maintain upper and lower bounds (u, l) for every time period except the first (note that this is a hybrid with a lookahead).
  - » The VFA
    - Piecewise linear, concave value function in terms of energy, indexed by time.
  - » The lookahead (deterministic)
    - Optimize over a horizon H (only tunable parameter) using forecasts of demand, prices and wind energy
  - » The lookahead CFA
    - Use a lookahead policy (deterministic), but with a tunable parameter that improves robustness.

An energy storage problem

- Each policy is best on certain problems
  - » Results are percent of *posterior* optimal solution

Problem:	Problem description	PFA	CFA Error correction	VFA	Determ. Lookahead	CFA Lookahead
A	A stationary problem with heavy-tailed prices, relatively low noise, moderately accurate forecasts.	**0.959**	**0.839**	**0.936**	**0.887**	**0.887**
B	A time-dependent problem with daily load patterns, no seasonalities in energy and price, relatively low noise, less accurate forecasts.	**0.714**	**0.752**	**0.712**	**0.746**	**0.746**
C	A time-dependent problem with daily load, energy and price patterns, relatively high noise, forecast errors increase over horizon.	**0.865**	**0.590**	**0.914**	**0.886**	**0.886**
D	A time-dependent problem, relatively low noise, very accurate forecasts.	**0.962**	**0.749**	**0.971**	**0.997**	**0.997**
E	Same as (C), but the forecast errors are stationary over the planning horizon.	**0.865**	**0.590**	**0.914**	**0.922**	**0.934**

  - » … any policy might be best depending on the data.

Joint research with Prof. Stephan Meisel, University of Muenster, Germany.

Outline

- Elements of a dynamic model
- Modeling uncertainty
- Designing policies
- The four classes of policies
- From deterministic to stochastic optimization

From deterministic to stochastic

- Imagine that you would like to solve the time-dependent linear program:

$$\min_{x_0,\ldots,x_T} \sum_{t=0}^{T} c_t x_t$$

  - » subject to

$$A_0 x_0 = b_0$$

$$A_t x_t - B_{t-1} x_{t-1} = b_t, \quad t \geq 1.$$

- We can convert this to a proper stochastic model by replacing x_t with $X_t^\pi(S_t)$ and taking an expectation:

$$\min_\pi \mathrm{E} \sum_{t=0}^{T} c_t X_t^\pi(S_t)$$

The policy $X_t^\pi(S_t)$ has to satisfy $A_t x_t = R_t$ with transition function:

$$S_{t+1} = S^M\left(S_t, x_t, W_{t+1}\right)$$

Modeling

- Deterministic
  - » Objective function

$$\min_{x_0,\dots,x_T} \sum_{t=0}^{T} c_t x_t$$

  - » Decision variables:

$$(x_0,\dots,x_T)$$

  - » Constraints:
    - at time t

$$\left.\begin{aligned} A_t x_t &= R_t \\ x_t &\geq 0 \end{aligned}\right\} \mathcal{X}_t$$

    - Transition function

$$R_{t+1} = b_{t+1} + B_t x_t$$

- Stochastic
  - » Objective function

$$\max_\pi E^\pi \left\{ \sum_{t=0}^{T} C_t\left(S_t, X_t^\pi(S_t), W_{t+1}\right) \mid S_0 \right\}$$

  - » Policy

$$X^\pi : S \mapsto \mathcal{X}$$

  - » Constraints at time t

$$x_t = X_t^\pi(S_t) \in \mathcal{X}_t$$

  - » Transition function

$$S_{t+1} = S^M\left(S_t, x_t, W_{t+1}\right)$$

  - » Exogenous information

$$(S_0, W_1, W_2, \dots, W_T)$$

From deterministic to stochastic

- Stochastic problems
  - » Modeling is the most important, and hardest, aspect of stochastic optimization
  - » Searching for policies is important, but less critical.
  - » Modeling uncertainty is often overlooked, but is of central importance.
  - » Evaluating a policy is important, and difficult. In a simulator? In the field?
- Deterministic problems
  - » Modeling is important, but not central.
  - » Algorithms are the most important, and hardest part.
  - » Huh?
  - » Just add up the costs!!

Modeling stochastic, dynamic problems

- The universal objective function

$$\max_\pi E^\pi \left\{ \sum_{t=0}^{T} C_t \left(S_t, X_t^\pi(S_t), W_{t+1} \right) | S_0 \right\}$$

with $S_{t+1} = S^M \left(S_t, x_t, W_{t+1}(\omega) \right)$

- You next need to develop a stochastic model:
  - » Model uncertainty about parameters in S_0
  - » Model the stochastic process $W_1, W_2, \ldots, W_N$ (for training)
  - » Model the random variable $\widehat{W}$ (for testing, if necessary)

- Then search for policies:
  - » Policy search:
    - PFAs, CFAs
  - » Lookahead policies:
    - VFAs, DLAs

Thank you!

For more information, go to

http://www.castlelab.princeton.edu/jungle/

Scroll to "Educational materials"

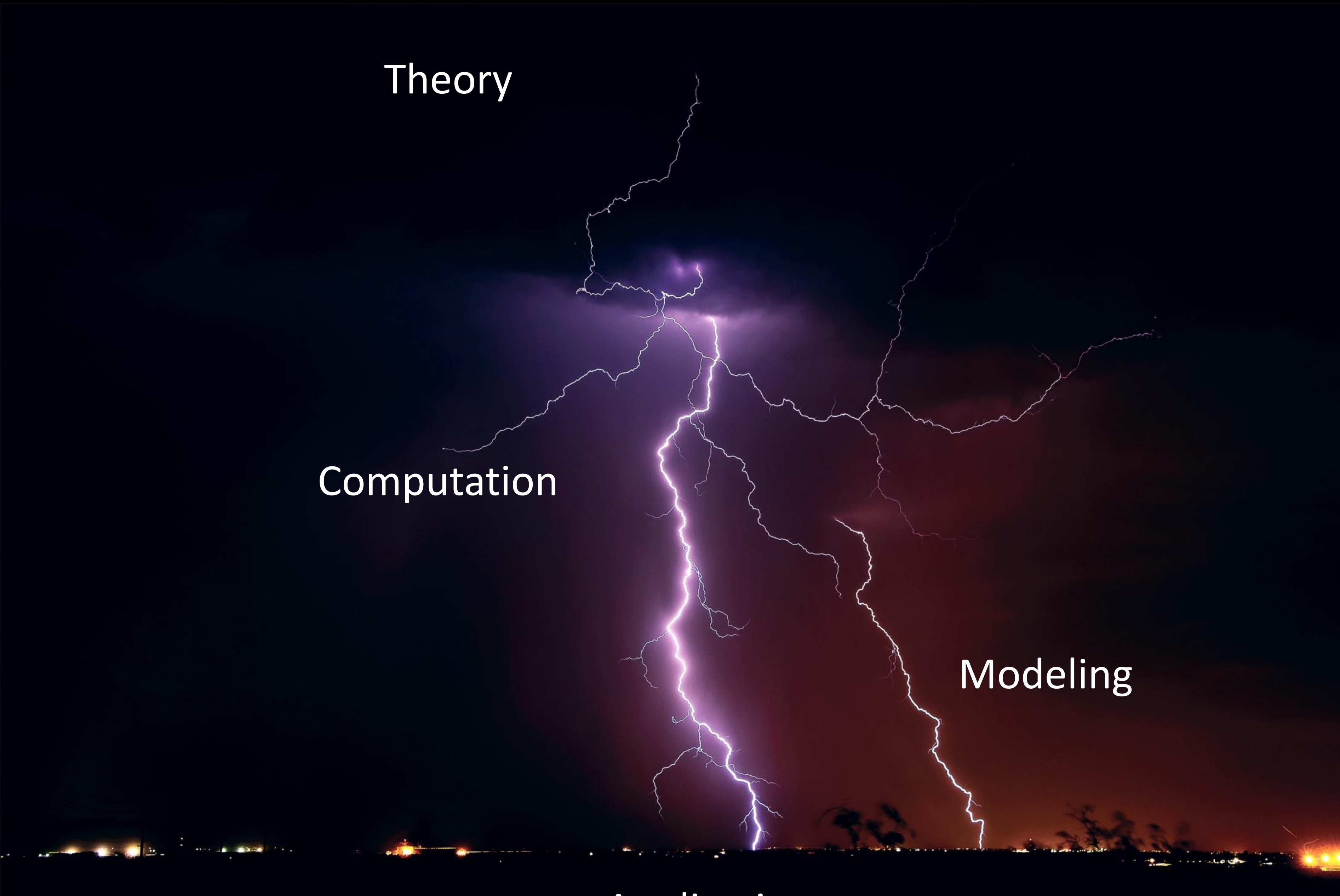
SEQUENTIAL DECISION ANALYTICS AND MODELING:

Modeling exercises with python

Warren B. Powell

September 10, 2018

CONTENTS



Applications