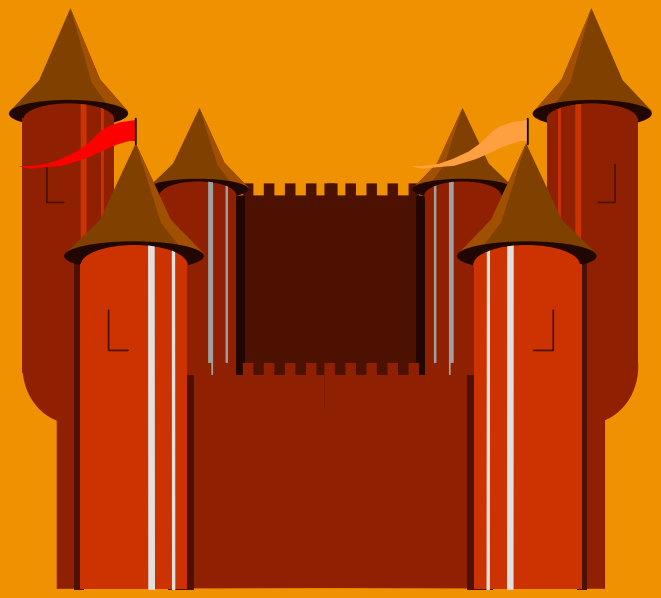
ORF 411

Sequential decision analytics and modeling

Fall, 2018



Warren Powell
Princeton University
http://www.castlelab.princeton.edu

Fall break

Week 7 – Monday

SMART-ISO lecture

Hugo Simao

Week 7 – Wednesday

Uncertainty modeling

Modeling uncertainty

Types of uncertainty

Modeling uncertainty

- Observational uncertainty
- Prognostic uncertainty (forecasting)
- Experimental noise/variability
- Transitional uncertainty
- Inferential uncertainty
- Model uncertainty
- Systemic exogenous uncertainty
- Control/implementation uncertainty
- Algorithmic noise
- Goal uncertainty

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

Modeling uncertainty

- Types of uncertainty
  - » Observational uncertainty – Errors in our observations of the state of the system:
    - What is the CO2 content of the atmosphere?
    - What is inventory of oil in the U.S.?
  - » Prognostic uncertainty – Uncertainty in the forecast of a future event.
    - Forecasting demands
    - Forecasting the weather

Modeling uncertainty

- Types of uncertainty
  - » Experimental noise – This is the variability that arises when running repeated experiments (either in a lab or in the field)
    - Testing the impact of a new flu drug.
    - Testing the effect of a new material on battery lifetimes
  - » Transitional uncertainty – We have a model of how a (presumably) deterministic system evolves, but there is still noise:

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

    - Modeling the location of an aircraft moving at a certain speed from a known location.
    - Predicting the time of arrival of a car at a downstream node

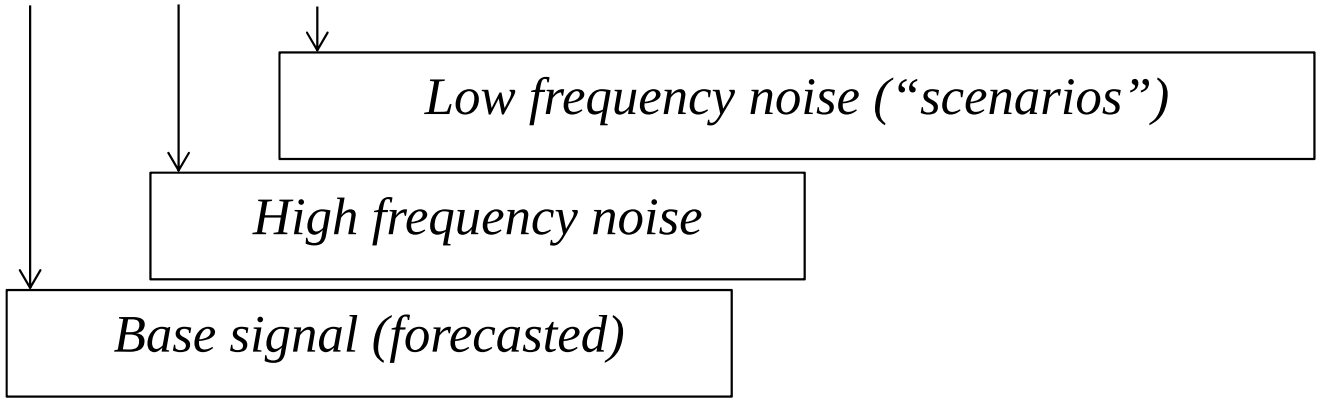
Modeling uncertainty

- Types of uncertainty
  - » Inferential uncertainty
    - Uncertainty in parameters estimated from observational data
    - Sometimes known as *diagnostic uncertainty* which might arise in the context of estimating a condition such as disease or the reason for a malfunction (in an engine). Such an assessment would an inference based on indirect observations.
  - » Model uncertainty – This is uncertainty about the model itself, which comes in two forms:
    - Uncertainty about the structure of the model:
      - – Linear approximation of a nonlinear model
      - – Different sets of equations describing the climate
    - Parameters characterizing the model

Modeling uncertainty

- Types of uncertainty
  - » Systemic exogenous uncertainty - Errors in the model of exogenous information that occur on long time scales:
    - Modeling the effect of long-term drops in oil consumption due to conservation
    - Modeling the effect of increased cloud cover due to climate change

$$W_t = \mu_t + \varepsilon_t + \psi_t$$



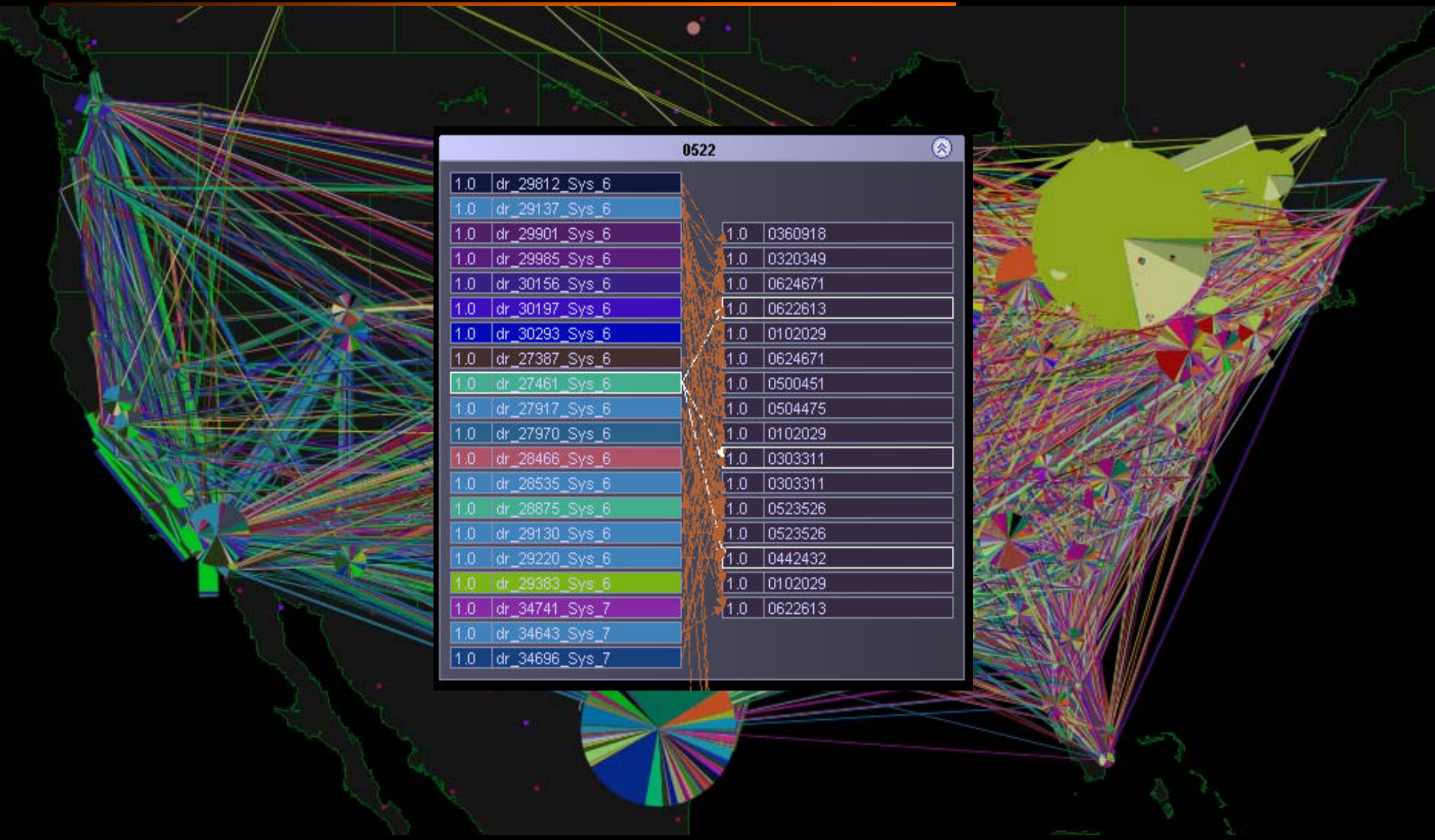
Modeling uncertainty

- Types of uncertainty
  - » Control uncertainty
    - You ask for x_t but you get $x_t + \varepsilon_t$
    - Wiley sets a wholesale price of $80, but Amazon sells at some random price above that (limits Wiley's ability to set prices).
    - You order 10 items, but only get 8 due to a stockout.
    - You try to drive at 70 mph, but there is variability due to your inability to hold a speed perfectly, along with the effect of traffic.
    - Uber would like 50 drivers on duty, but can only influence their behavior through surge pricing.
  - » Algorithmic uncertainty
    - Run the same algorithm twice, and you may get different answers (depends on the algorithm and the nature of the compute environment).

Modeling uncertainty

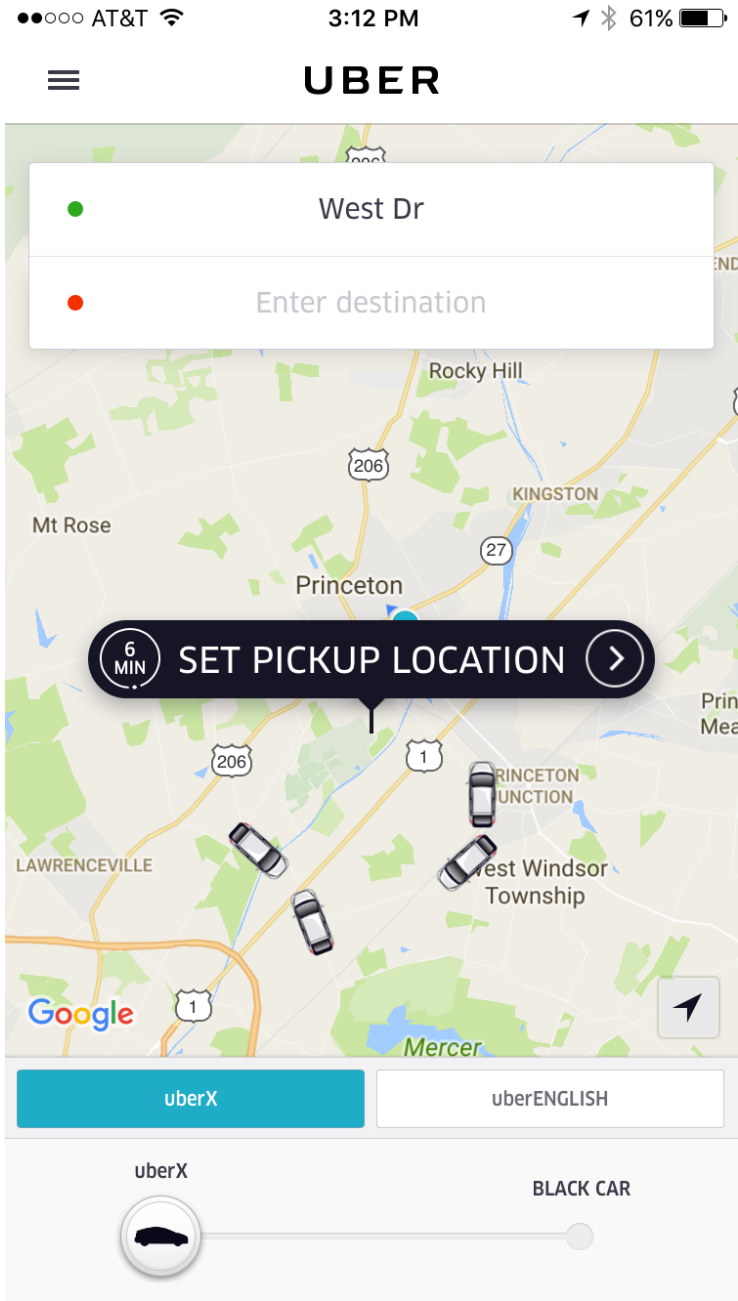
- Types of uncertainty
  - » Goal uncertainty
    - There are many settings where decisions are being made by a group of people:
      - – Dispatchers for a trucking company
    - What are we trying to achieve?
    - Maximize revenue? Minimize costs? Maximize profits?
    - Best customer service?
    - Lowest risk?

Schneider National

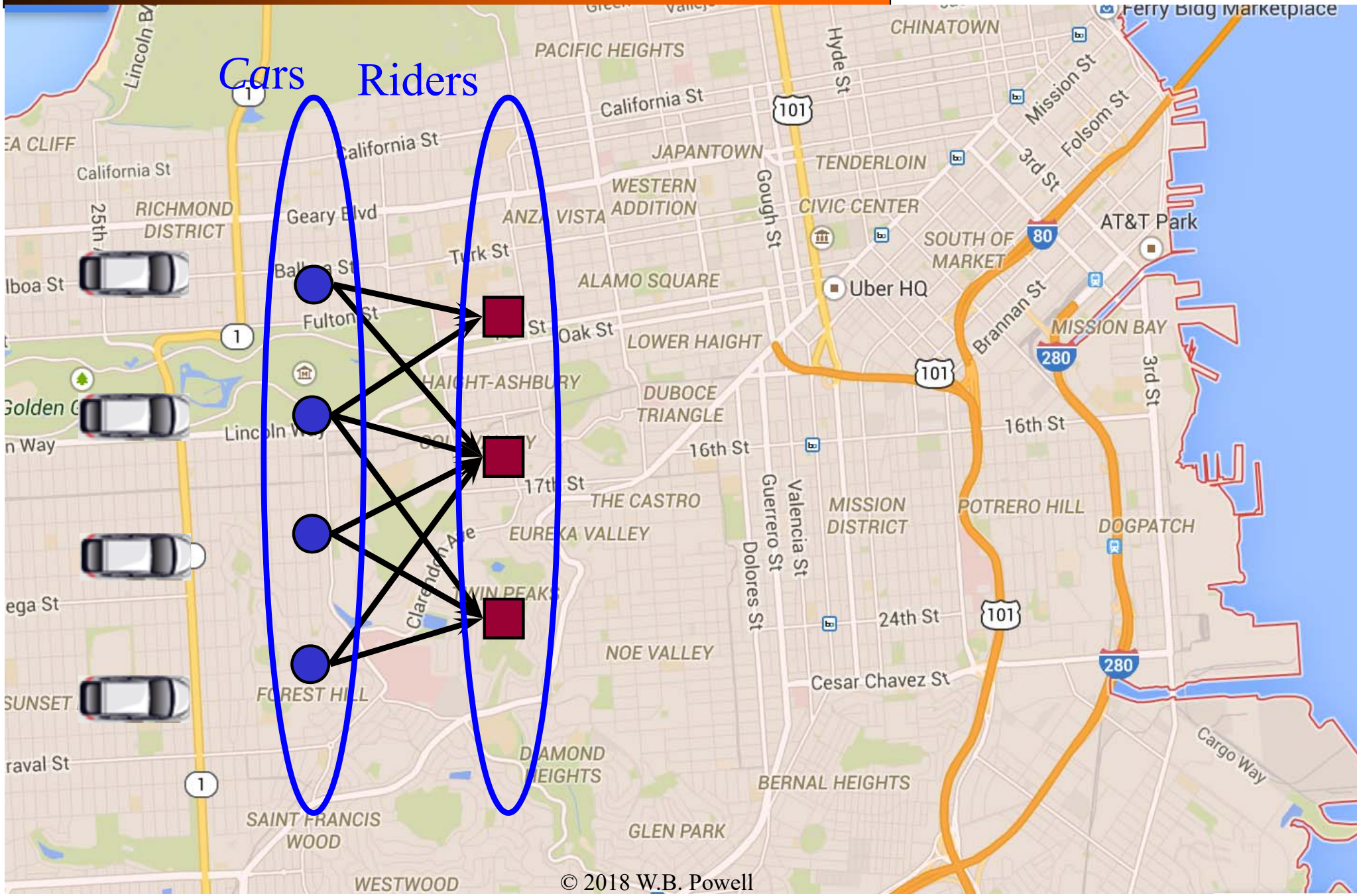


Goal uncertainty

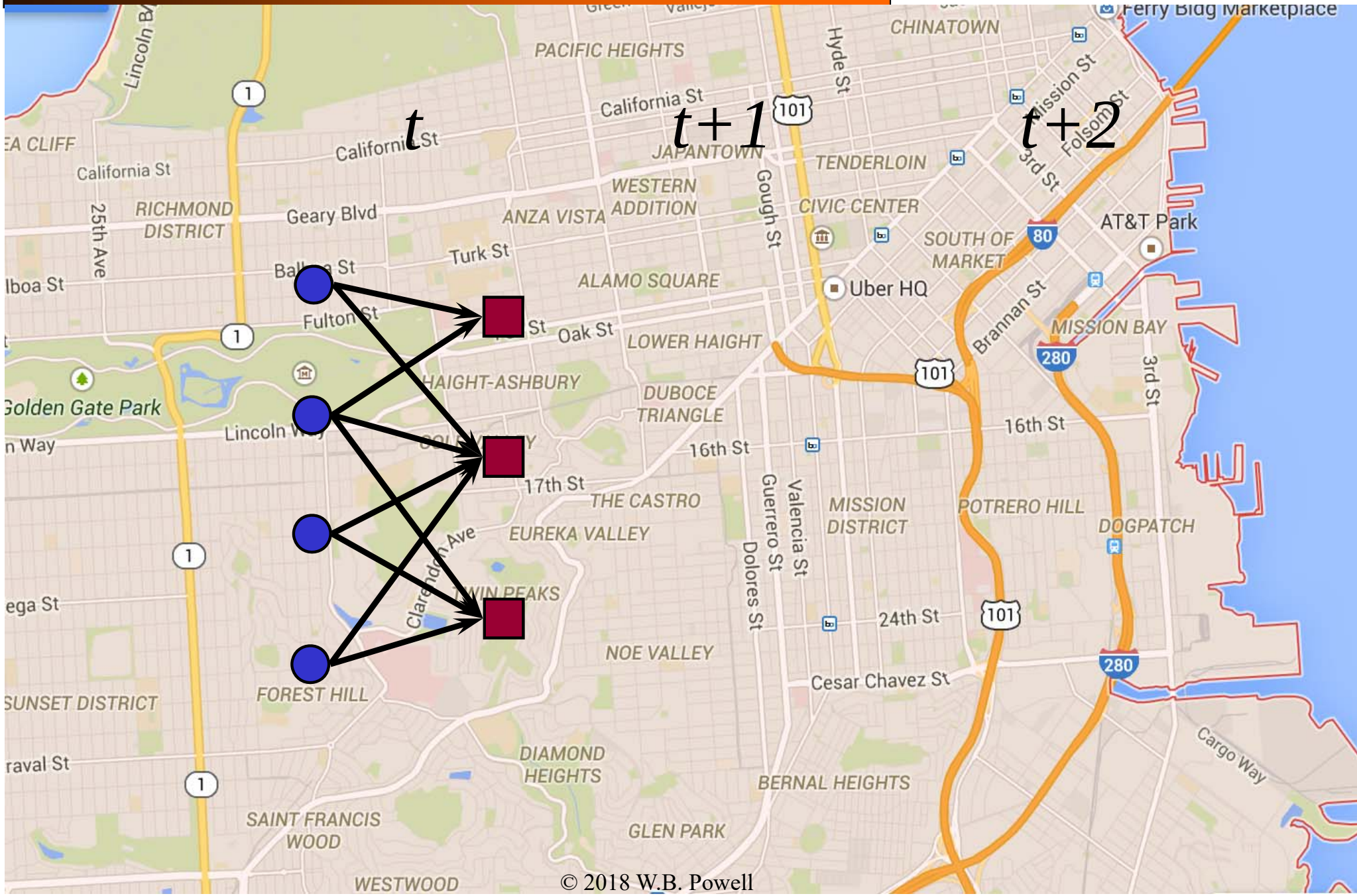
- Uber
  - » Provides real-time, on-demand transportation.
  - » Drivers are encouraged to enter or leave the system using pricing signals and informational guidance.
- Decisions:
  - » How to price to get the right balance of drivers relative to customers.
  - » Assigning and routing drivers to manage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



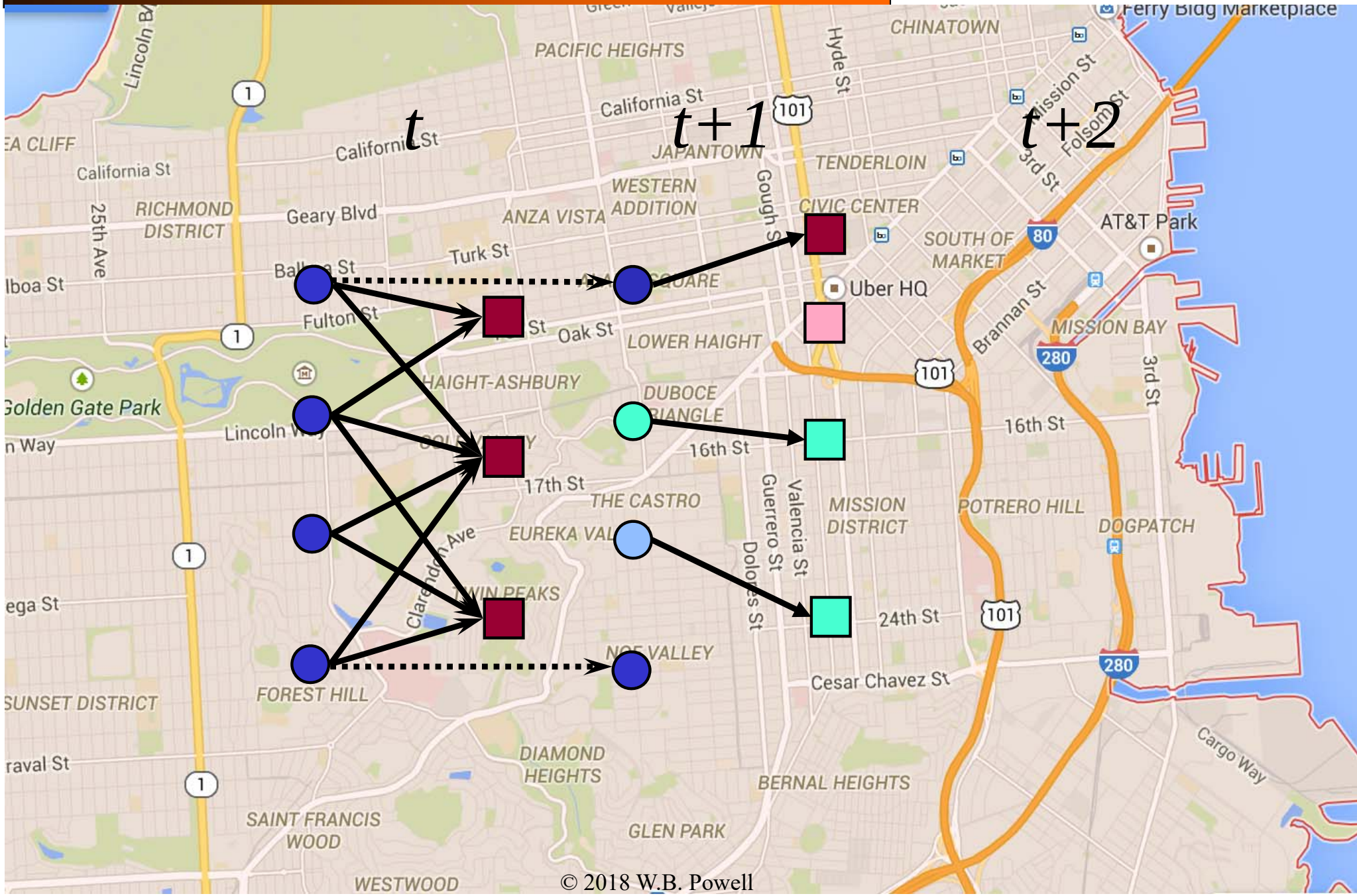
Goal uncertainty



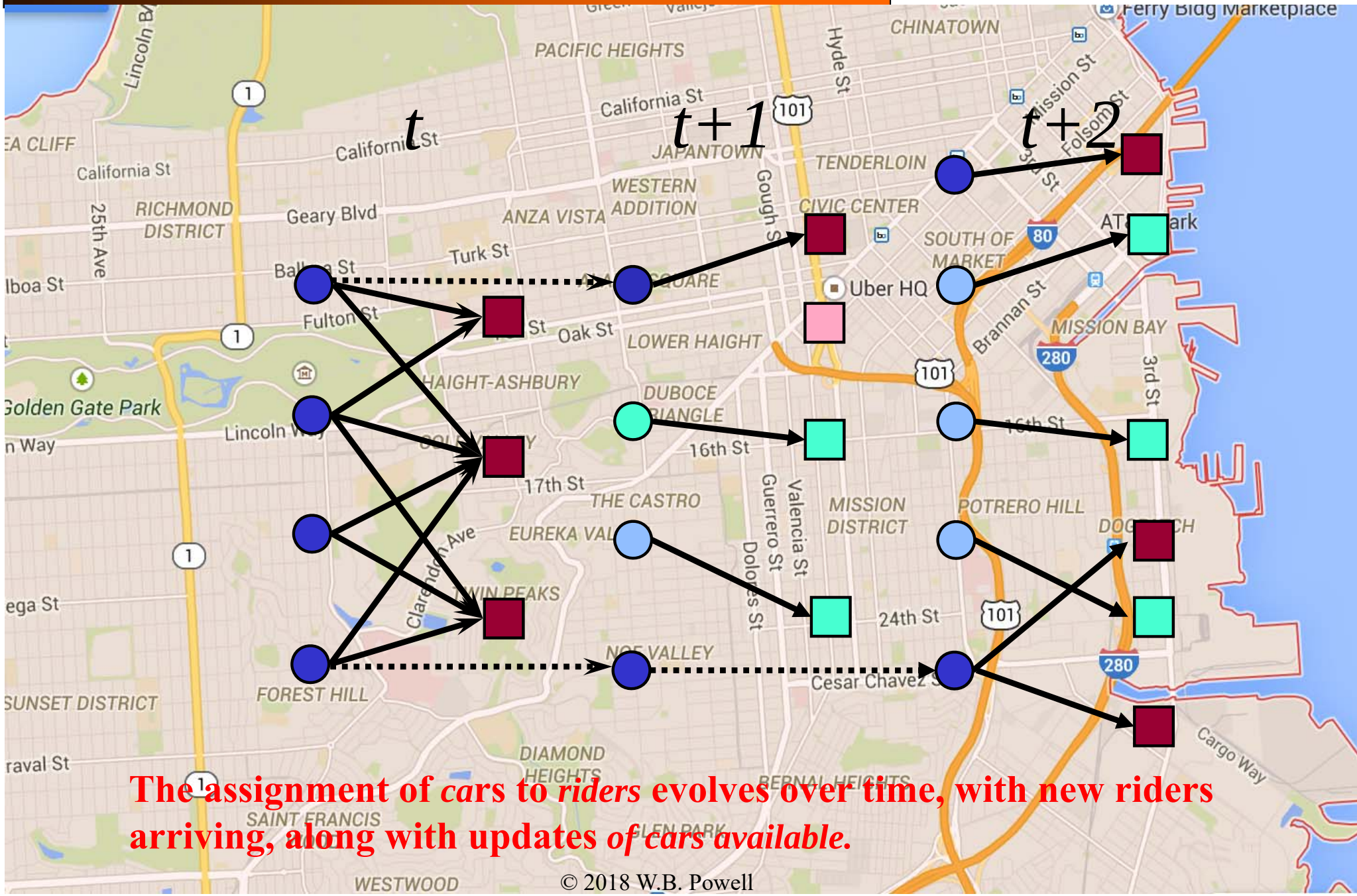
Goal uncertainty



Goal uncertainty



Goal uncertainty

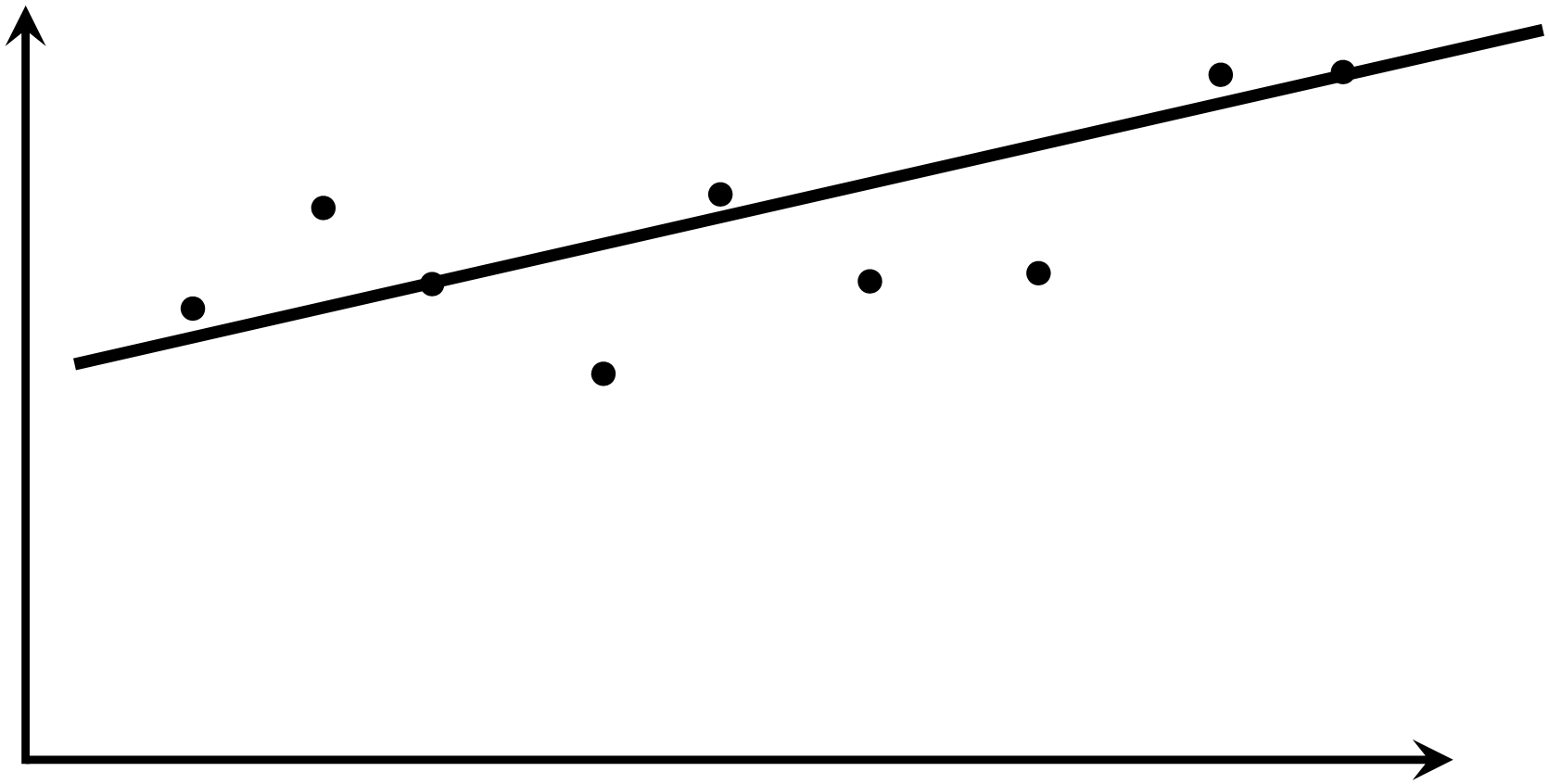


The assignment of *cars* to *riders* evolves over time, with new riders arriving, along with updates *of cars available.*


SCHNEIDER
Week 45
1 Way Rank
Dedic Rank
Trkfll Rank
Brok. Rank
Miles
18
4
1
3
1
Working units
18
1
By industry
Auto Asmby
Auto Parts
Retail
Paper
Miles
8
27
1
9
33
Nov YOY Growth
1 Way
Dedic
Trkfll
-5%
-6%
28%
SCHNEIDER

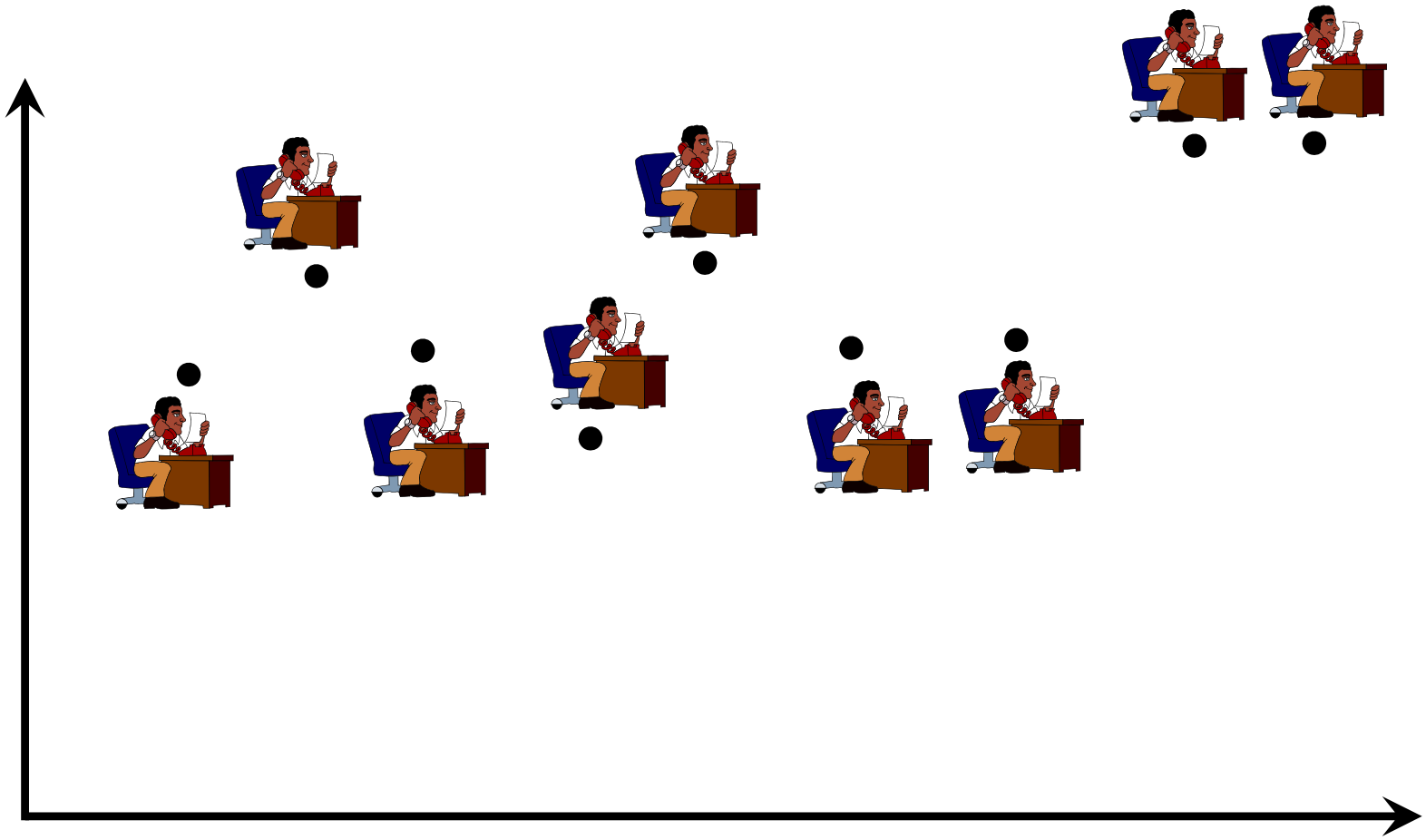
Goal uncertainty

- We can find one model that balances everyone's wishes (badly):



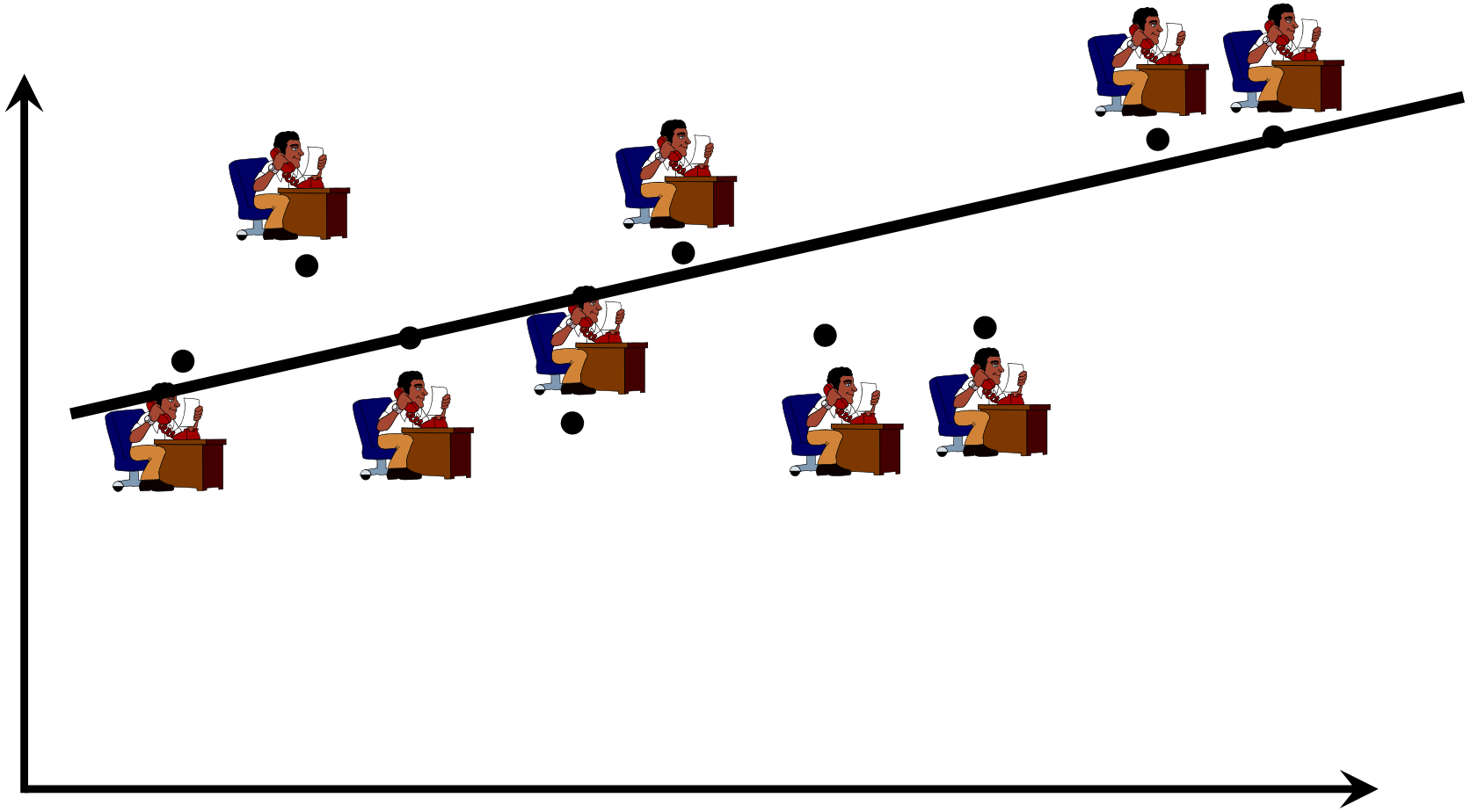
Goal uncertainty

- Formulating a model is like fitting a line through a set of users:



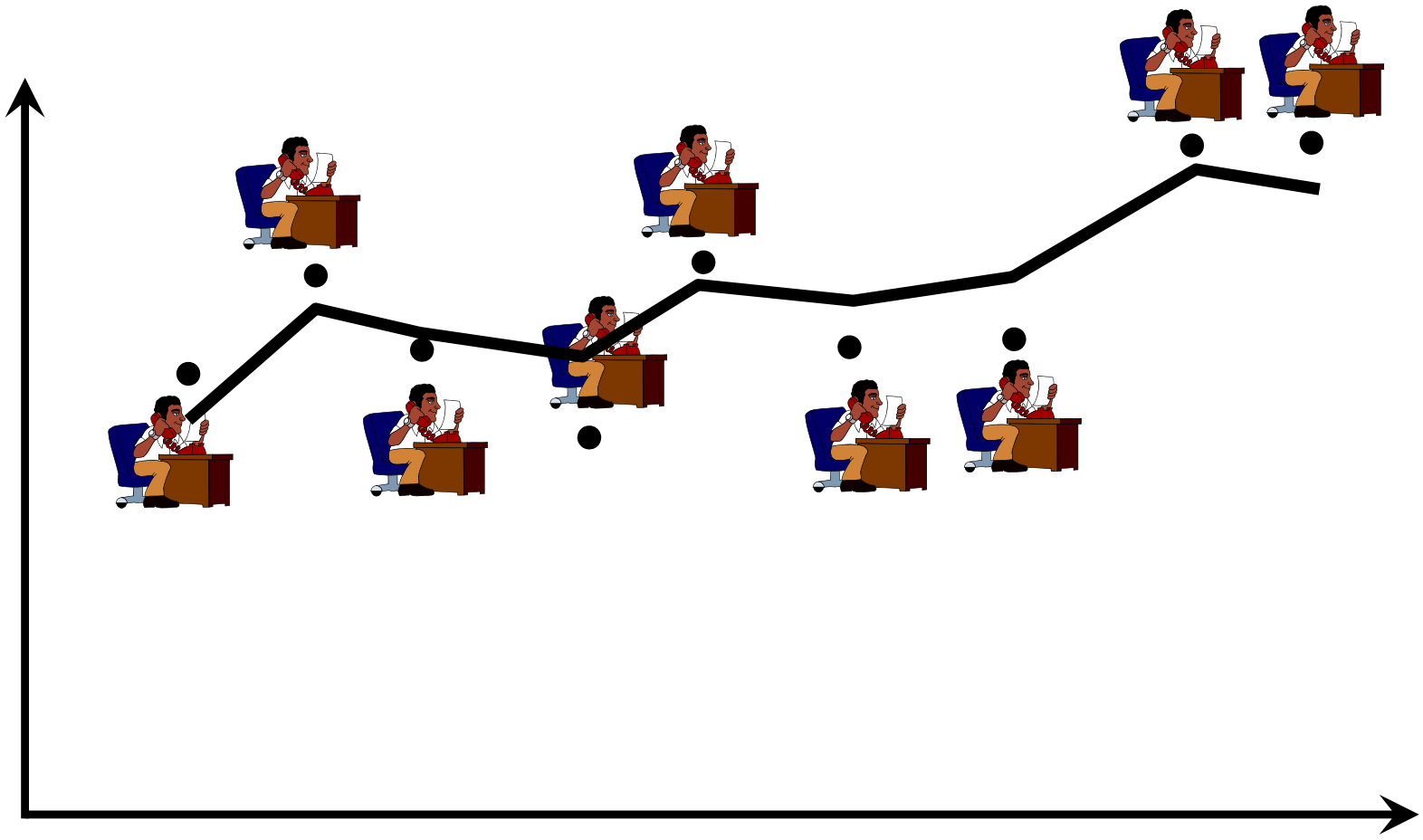
Goal uncertainty

- Formulating a model is like fitting a line through a set of users:



Goal uncertainty

- Should we customize the model (somewhat) for each user?



Modeling uncertainty

Modeling uncertainty

- There are two information processes that drive the system:
  - » Decisions x_t – This is the *endogenously* controllable information process.
  - » Exogenous information - This comes from the initial state S_0, and the exogenous information process W_t.
- To figure out how to make good decisions, you need:
  - » The system model $S_{t+1} = S^M(S_t, x_t, W_{t+1})$
  - » The initial state S_0 and the exogenous information process W_t

Modeling uncertainty

- The initial state S_0. This contains:
  - » All deterministic parameters needed by the system. This is static data, so it is not modeled as part of the dynamic state S_t.
  - » "State of knowledge" – probabilistic information about uncertain parameters. This information is always represented as a probability distribution of some form:
    - Normally distributed uncertain parameters – This might be:
      - Estimated age of a power transformer
      - Estimated growth rate of a stock
      - The blood sugar of a patient
    - Discrete distributions – Examples include
      - Whether a patient has an infection
      - Probabilities of a discretized distribution of demand

Modeling uncertainty

- The exogenous information process W_t which might include:
  - » Passive information – This is information that arrives regardless of any actions we may take. Examples:
    - Purely exogenous – Information that is not influenced by the state of the system or any actions we take. Examples:
      - – Rainfall, traffic
      - – Traffic congestion
      - – Stock prices (if we are a small player)
    - Exogenous distributions are influenced by states and/or actions.
  - » Active information – This is information we choose to collect
    - Running a laboratory experiment
    - Purchasing a report

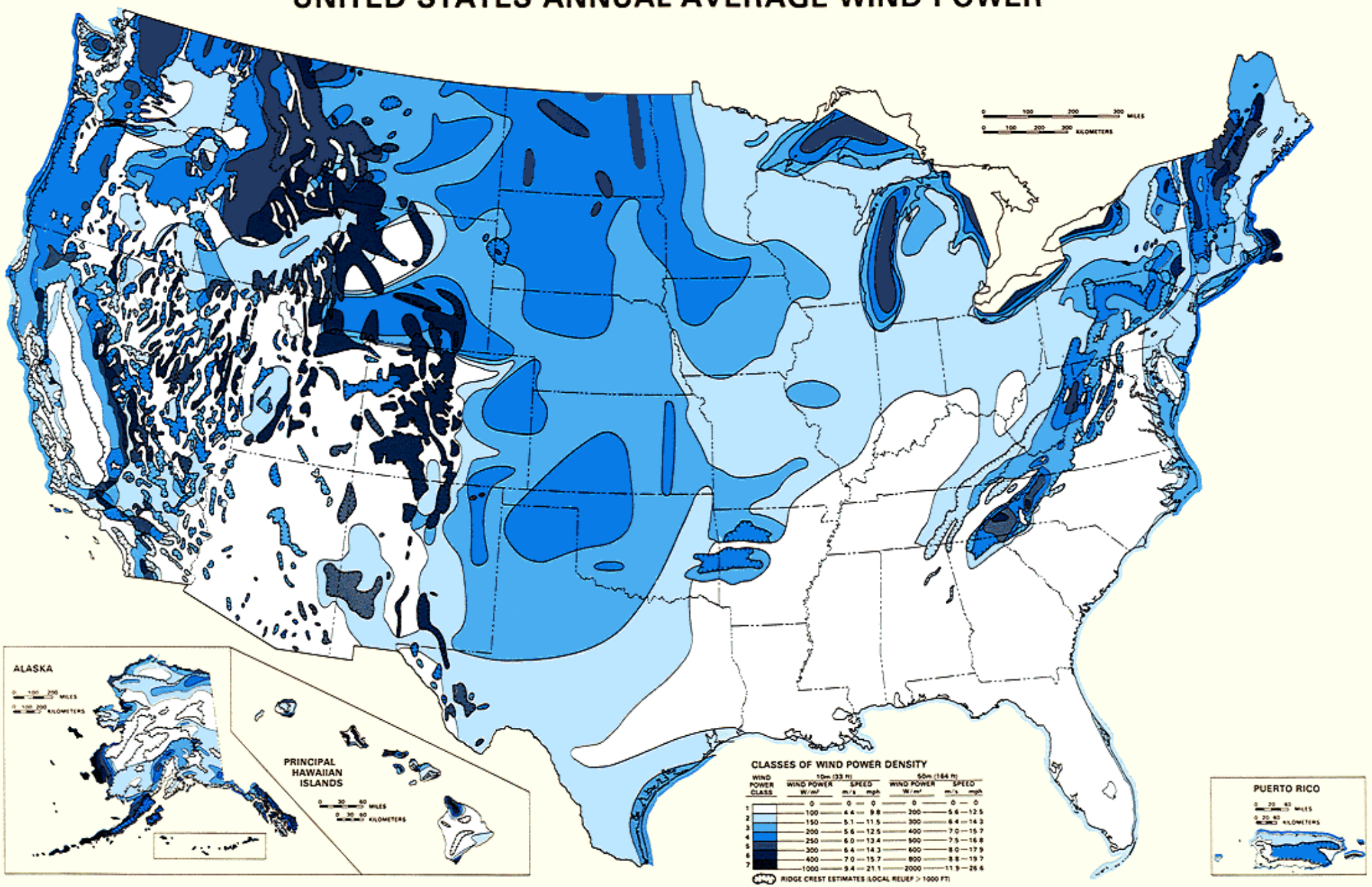
Modeling uncertainty

- Bayesian vs. frequentist uncertainty
  - » Bayesian uncertainty is captured by a distribution of belief derived from prior information:
    - Expert judgment
    - Information collected from different settings
    - Past experience
    - Bayesian uncertainty is always communicated through S_0
  - » Frequentist uncertainty
    - This is uncertainty derived from statistical analysis of the variability inherent in the exogenous information W_t

Modeling uncertainty

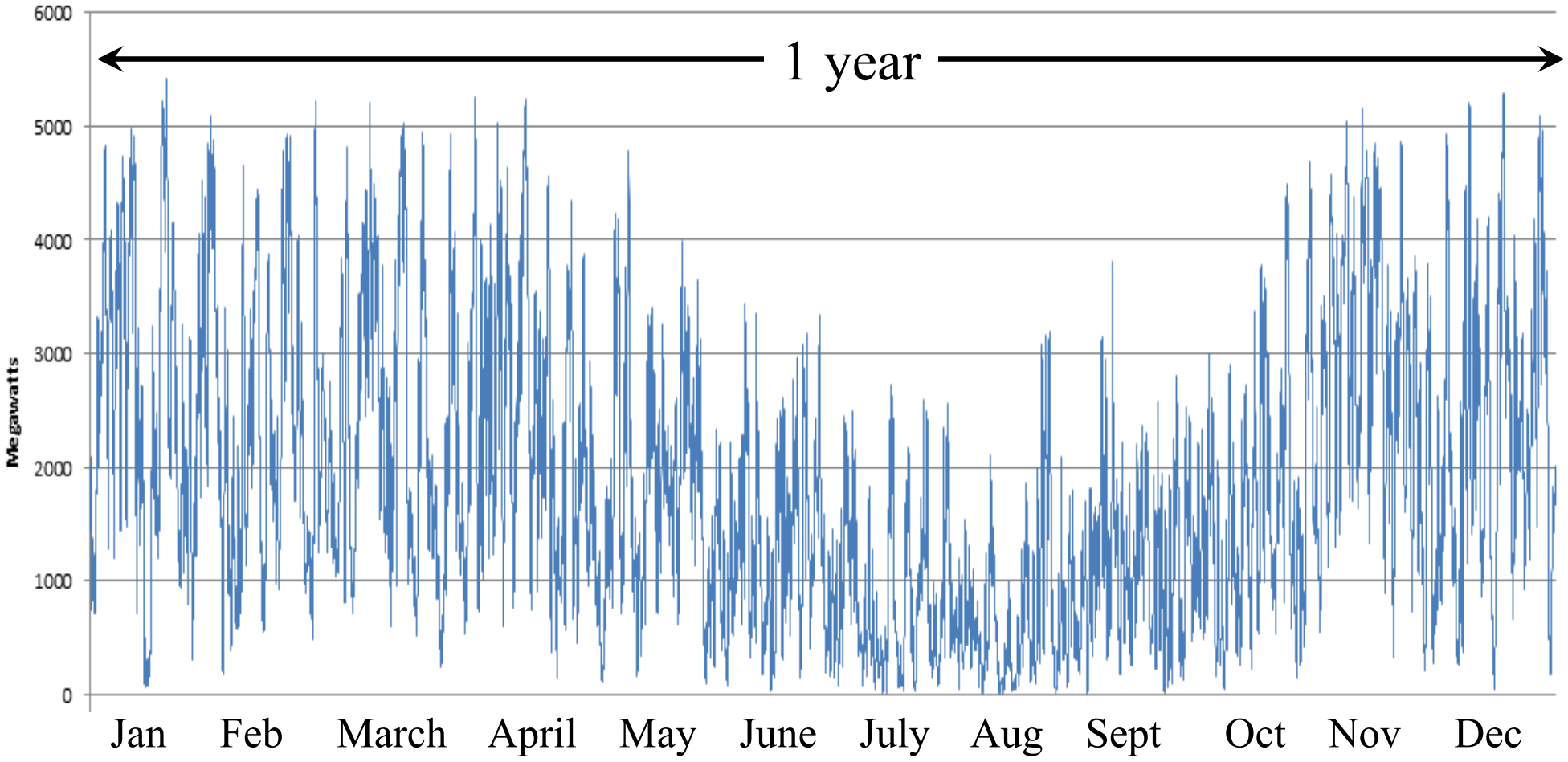
Examples of stochastic processes

UNITED STATES ANNUAL AVERAGE WIND POWER



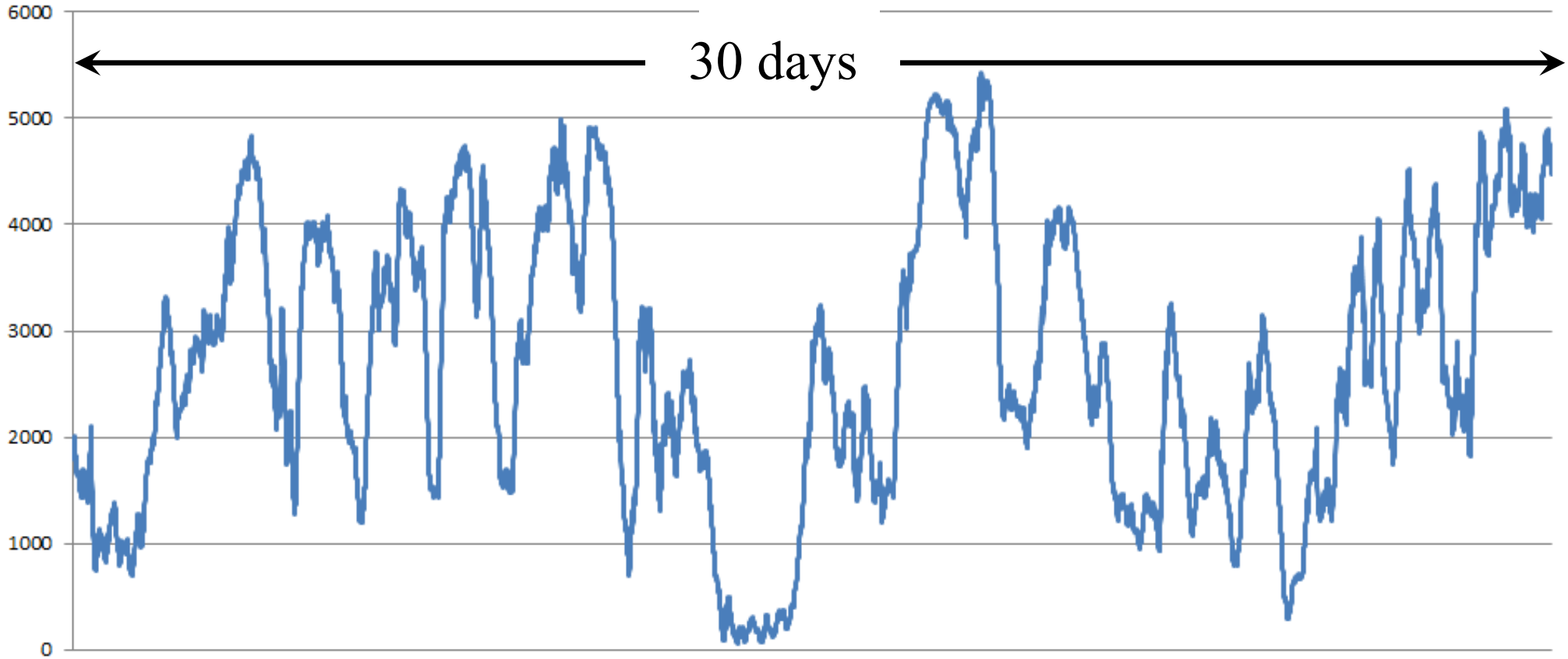
Energy from wind

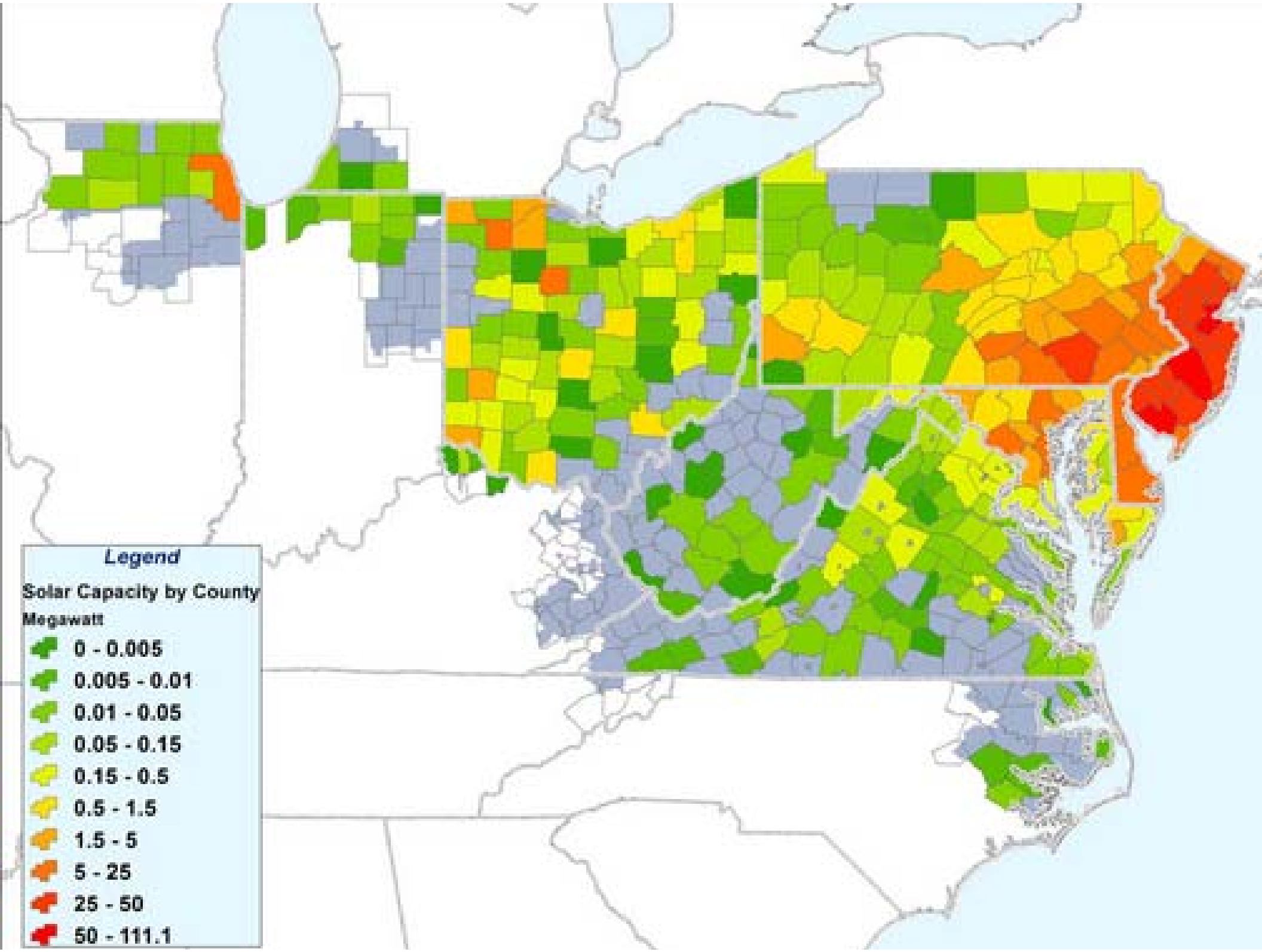
- Wind power from all PJM wind farms



Energy from wind

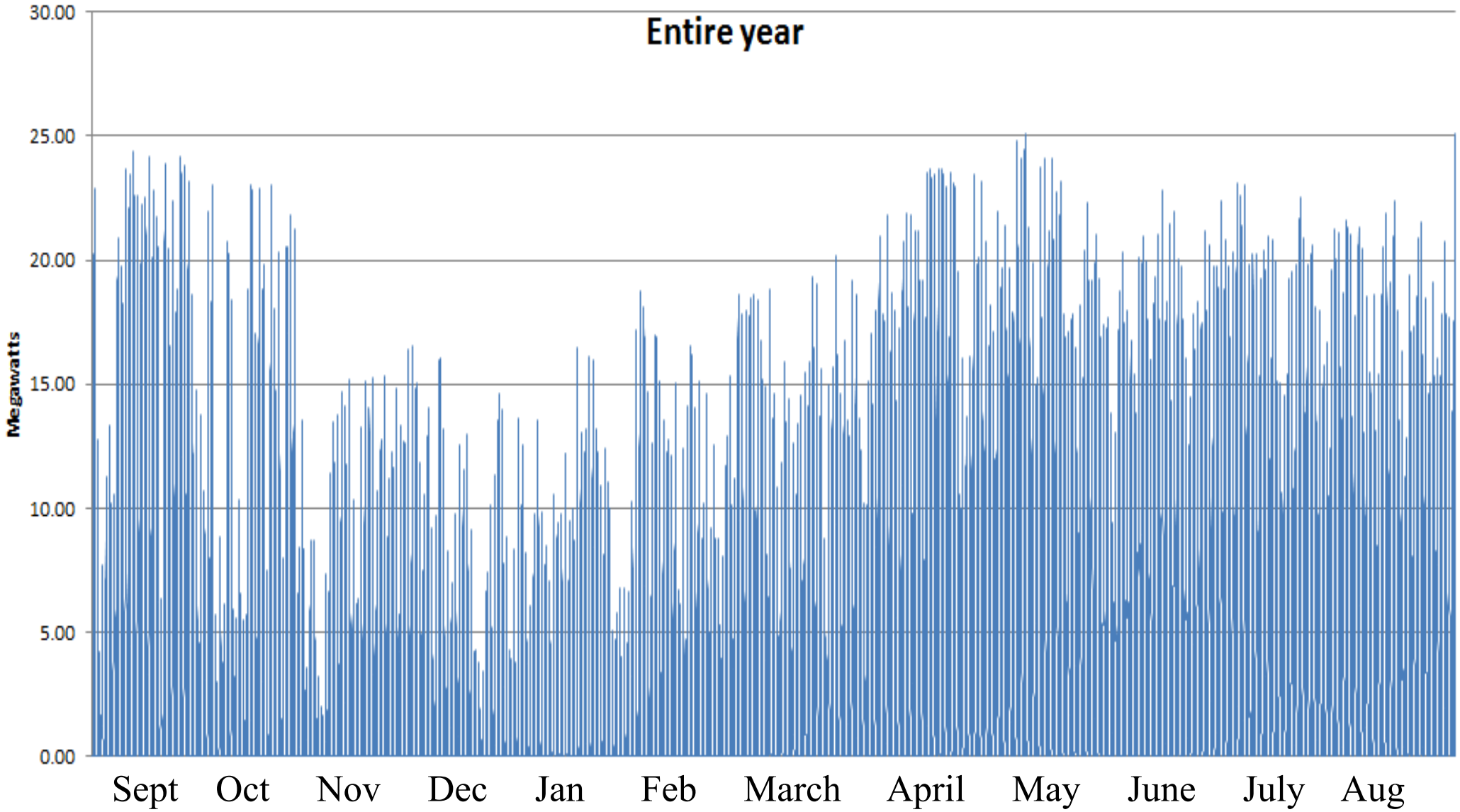
- Wind from all PJM wind farms



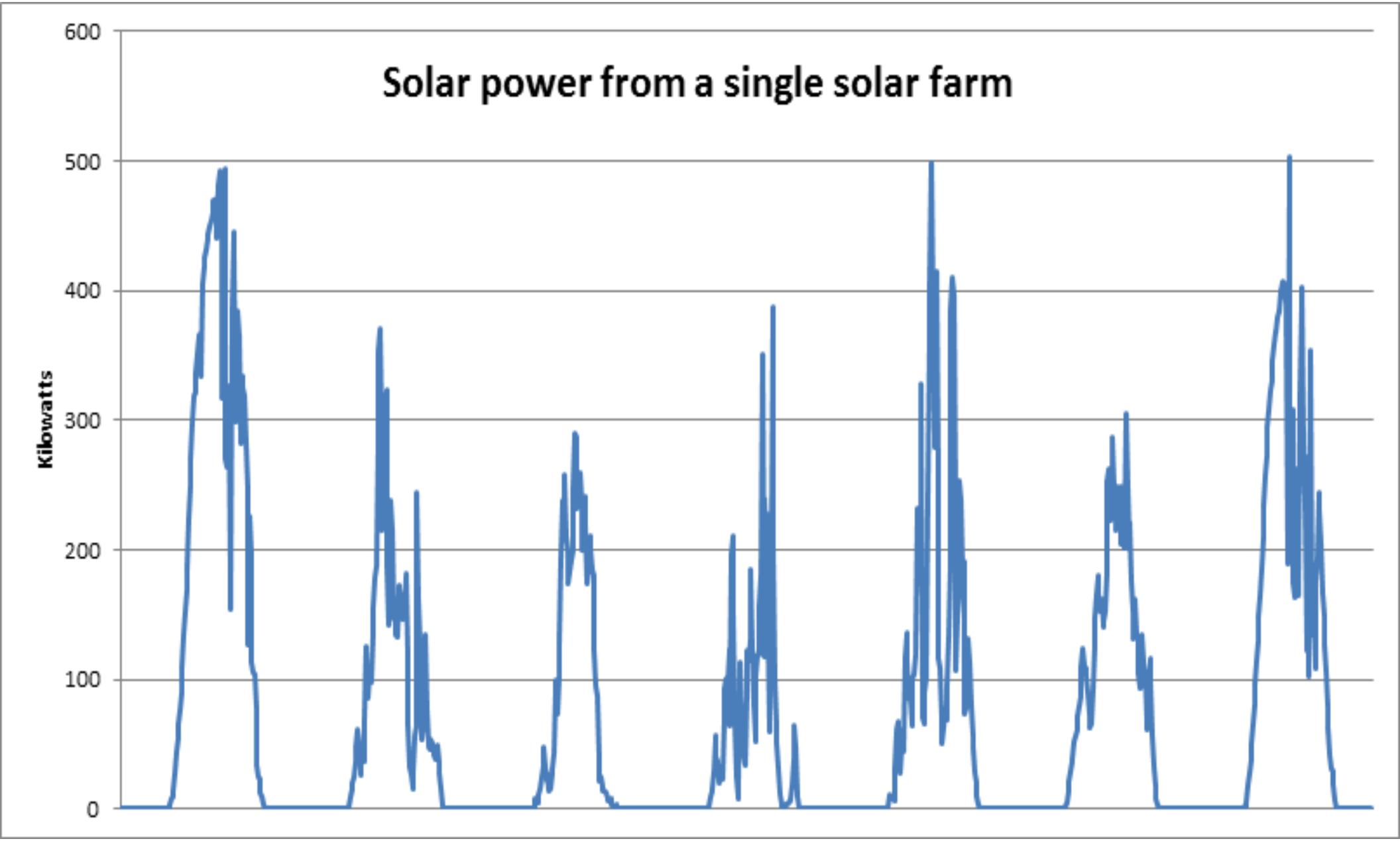

Legend
Solar Capacity by County
Megawatt
0 - 0.005
0.005 - 0.01
0.01 - 0.05
0.05 - 0.15
0.15 - 0.5
0.5 - 1.5
1.5 - 5
5 - 25
25 - 50
50 - 111.1

Energy from solar

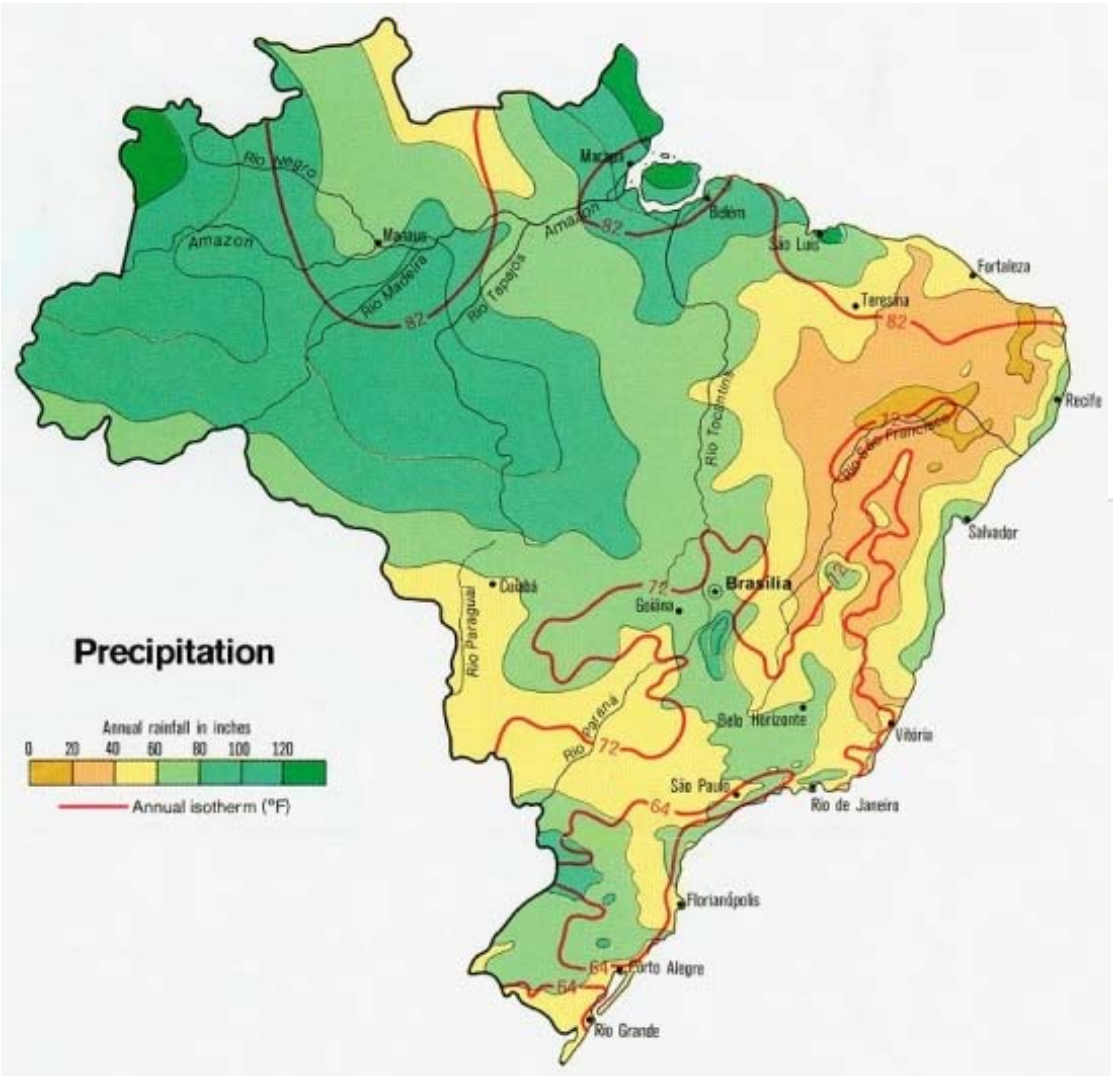
- Solar output over entire year (all farms)



Energy from solar



Brazil

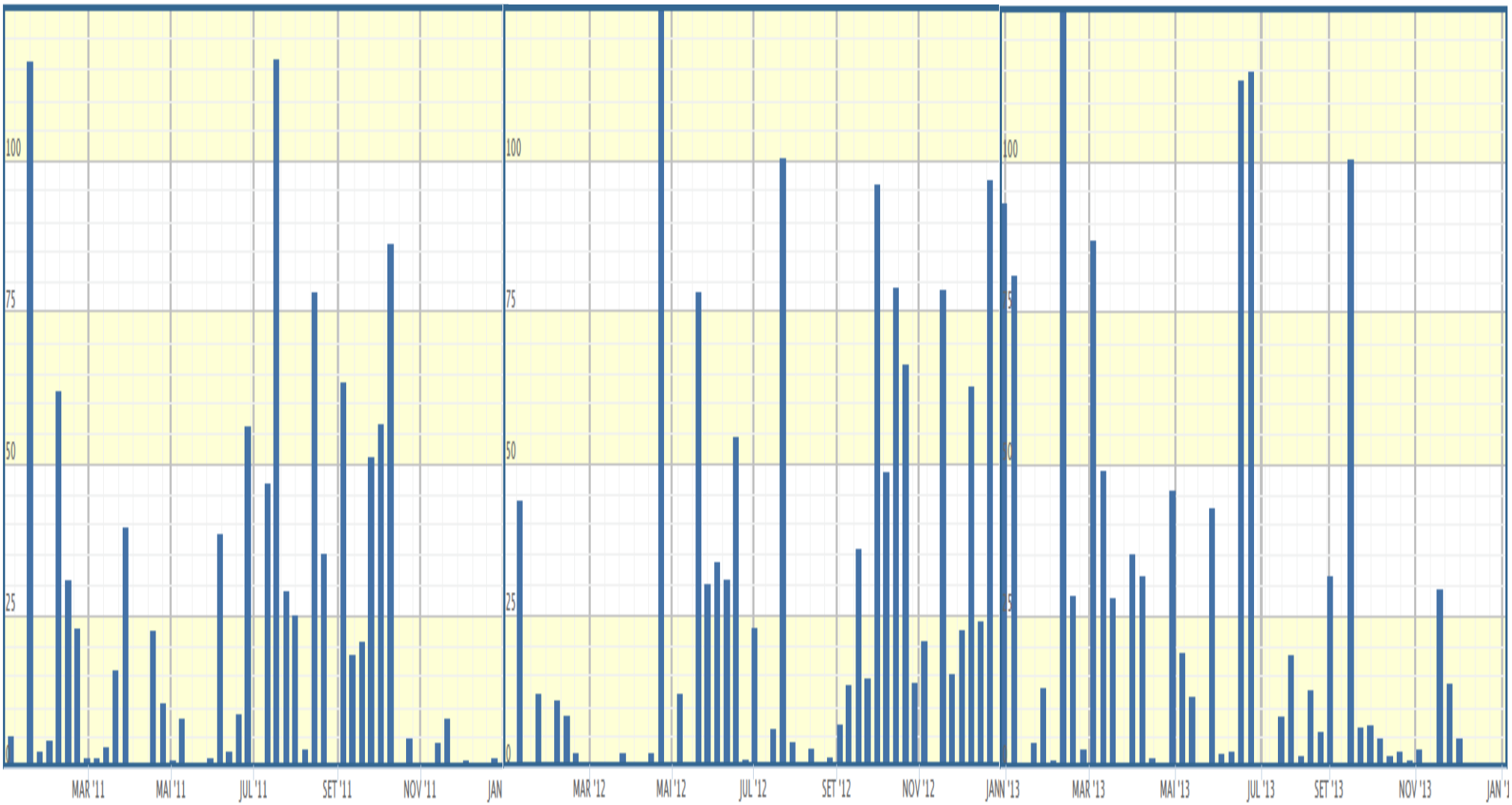


Brazil drought: It's a really dry January in the South American country, with rainfall is at its lowest level since 1930

Brazil Drought: Worst Water Crisis In 80 Years Affecting Four Million People

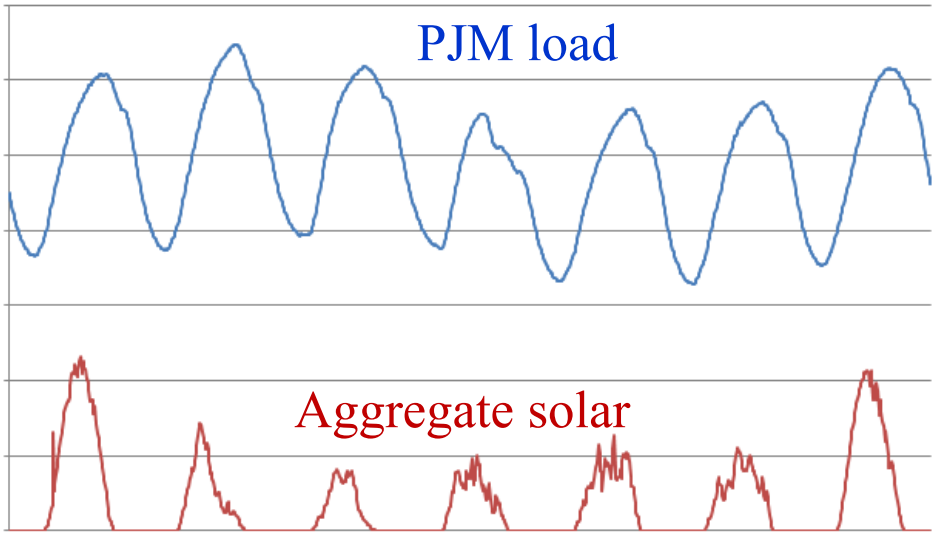
Rainfall

- Rainfall over 3 years in Brazil

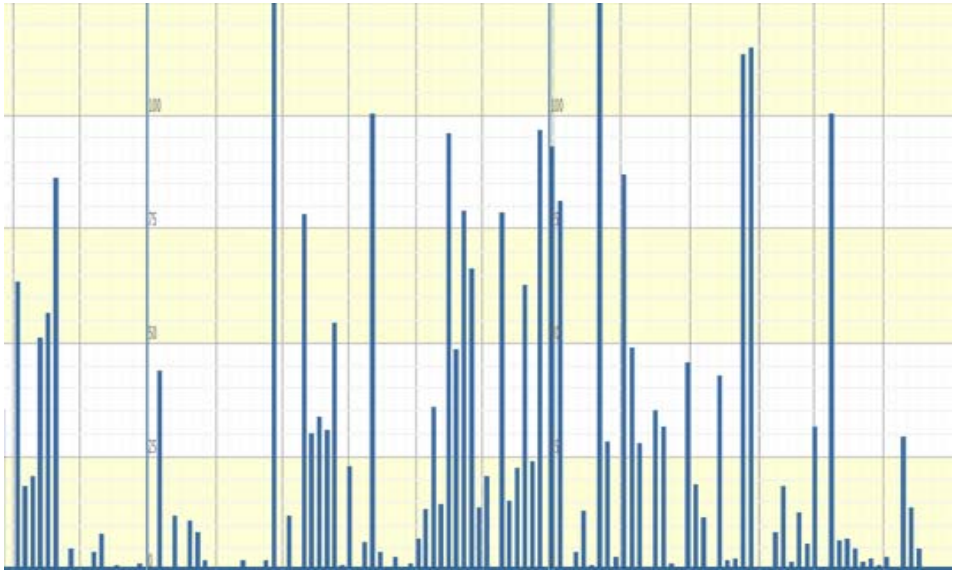


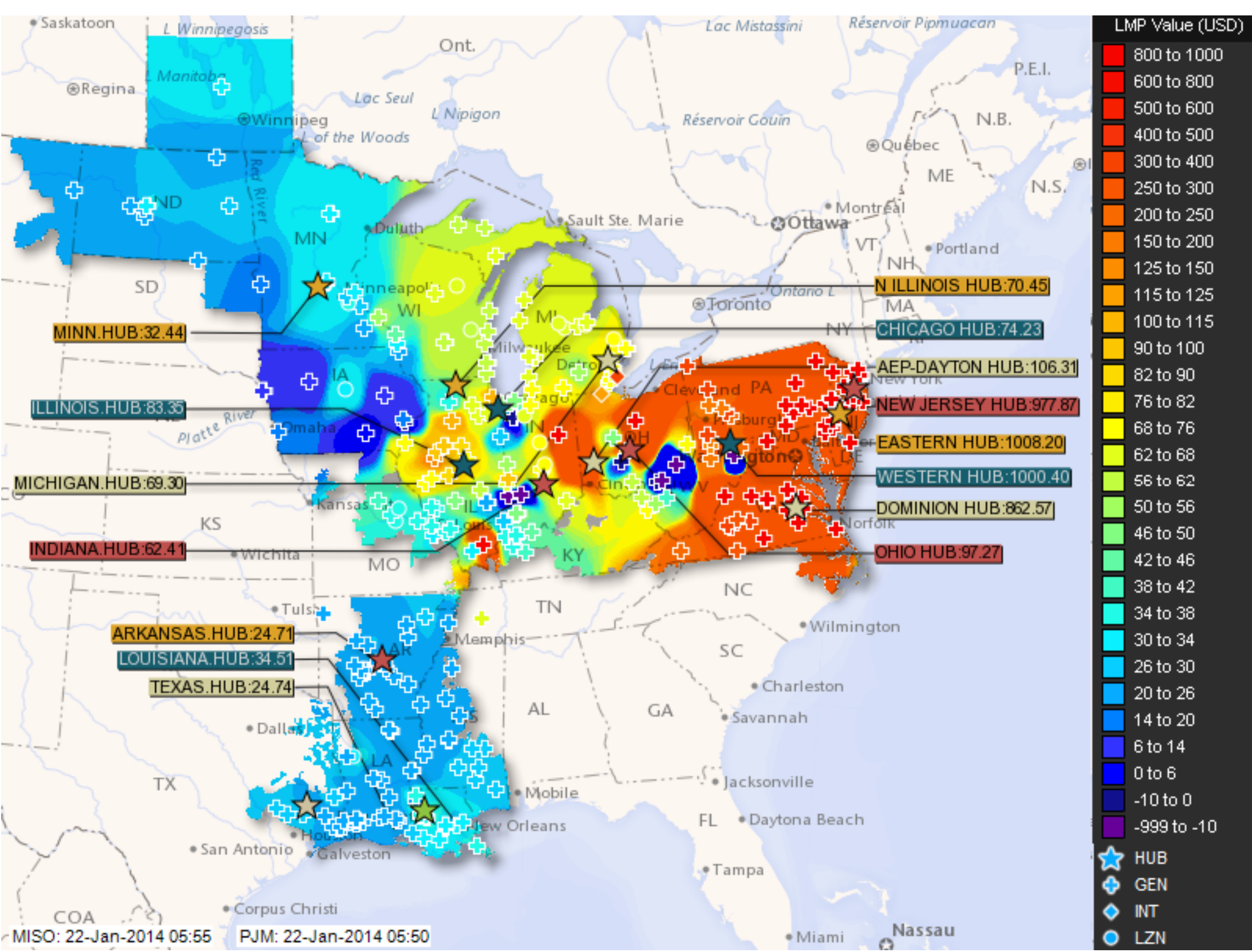
Uncertainty

- It is important to separate:
  - » Predictable variability
    - Diurnal cycles
    - Large weather patterns
    - Major human events (Super bowl)

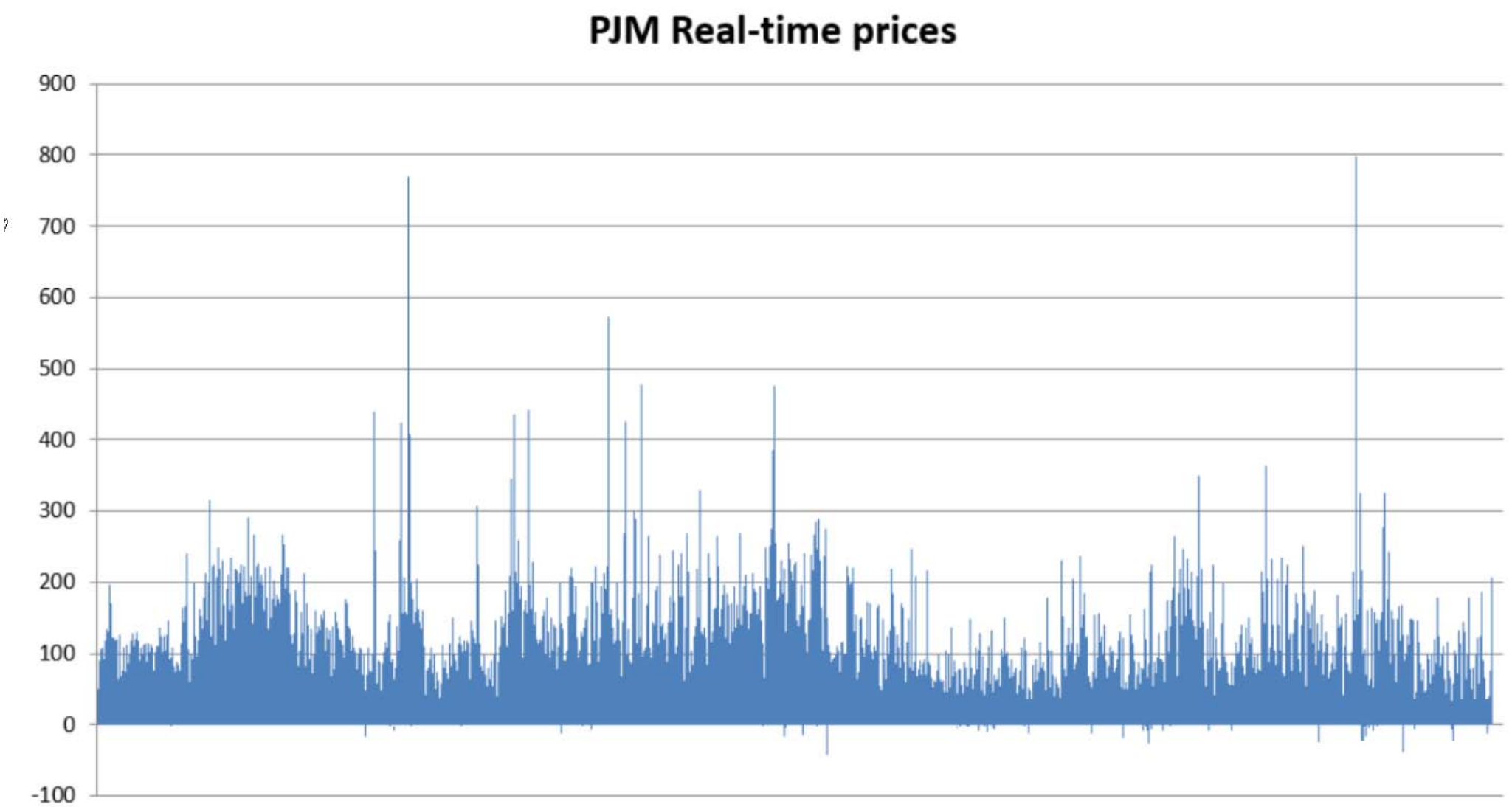


  - » Stochastic uncertainty
    - Temperature deviations from forecast
    - Late/early arrival of a storm
    - Generator failures
    - Wind shifts

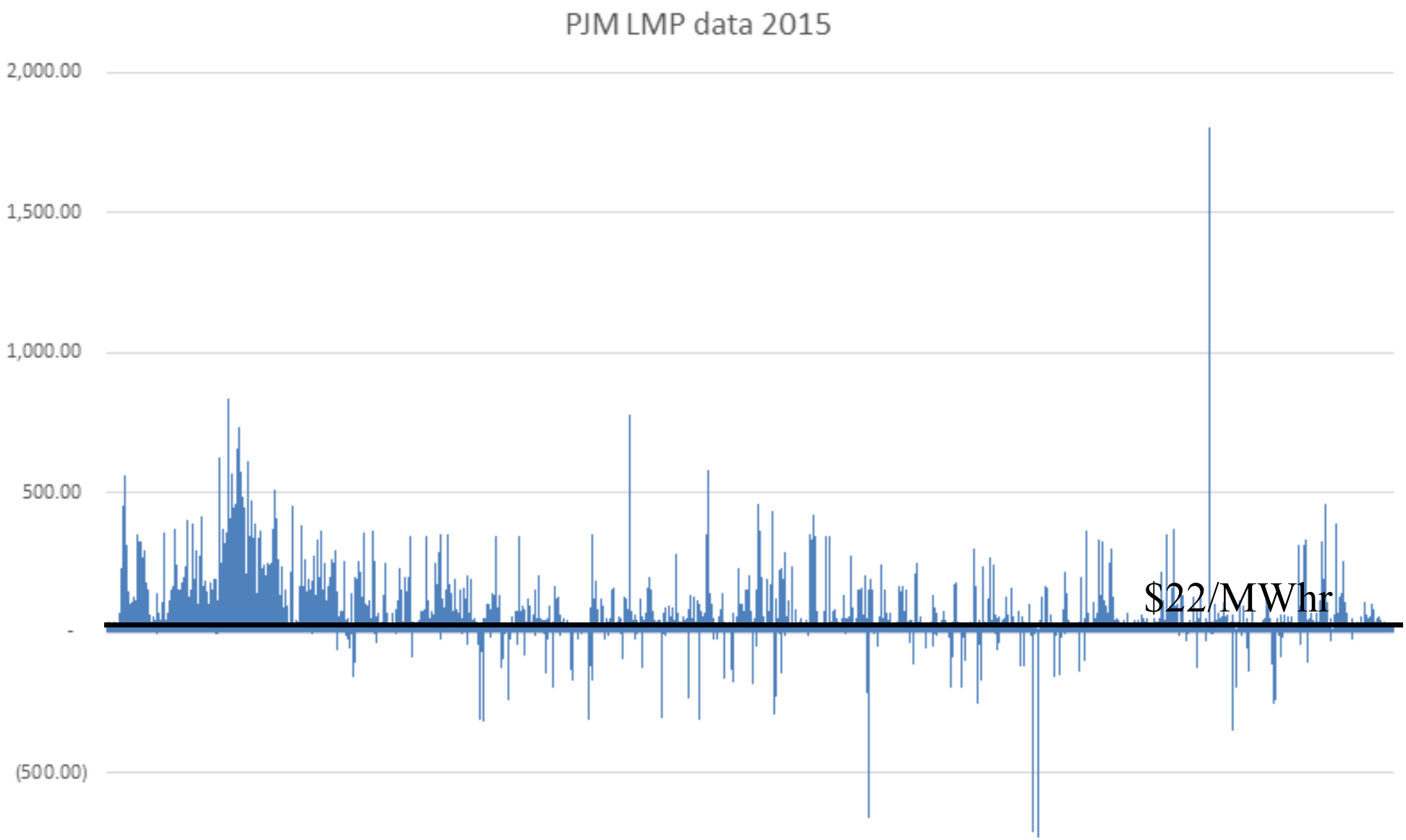



LMP Value (USD)
800 to 1000
600 to 800
500 to 600
400 to 500
300 to 400
250 to 300
200 to 250
150 to 200
125 to 150
115 to 125
100 to 115
90 to 100
82 to 90
76 to 82
68 to 76
62 to 68
56 to 62
50 to 56
46 to 50
42 to 46
38 to 42
34 to 38
30 to 34
26 to 30
20 to 26
14 to 20
6 to 14
0 to 6
-10 to 0
-999 to -10
HUB
GEN
INT
LZN
N ILLINOIS HUB:70.45
CHICAGO HUB:74.23
AEP-DAYTON HUB:106.31
NEW JERSEY HUB:977.87
EASTERN HUB:1008.20
WESTERN HUB:1000.40
DOMINION HUB:862.57
OHIO HUB:97.27
MINN.HUB:32.44
ILLINOIS.HUB:83.35
MICHIGAN.HUB:69.30
INDIANA.HUB:62.41
ARKANSAS.HUB:24.71
LOUISIANA.HUB:34.51
TEXAS.HUB:24.74
MISO: 22-Jan-2014 05:55
PJM: 22-Jan-2014 05:50

Locational marginal prices on the grid



Locational marginal prices on the grid



Time series modeling

8.3.1 Time series models

The time series literature is quite rich, so we are just going to illustrate a basic model which represents the price p_{t+1} as a function of the recent history of prices. For illustration, we are going to use the last three time periods, which means that we would write our model as

$$\begin{aligned} p_{t+1} &= \bar{\theta}_{t0} p_t + \bar{\theta}_{t1} p_{t-1} + \bar{\theta}_{t2} p_{t-2} + \varepsilon_{t+1}, \\ &= \bar{\theta}_t^T \phi_t + \varepsilon_{t+1}, \end{aligned} \tag{8.3}$$

where

$$\phi_t = \begin{pmatrix} p_t \\ p_{t-1} \\ p_{t-2} \end{pmatrix}$$

is our vector of prices. We assume that the noise $\varepsilon \sim N(0, \sigma_\varepsilon^2)$ for a given σ_ε^2.

The vector of coefficients $\bar{\theta}_t = (\bar{\theta}_{t0}, \bar{\theta}_{t1}, \bar{\theta}_{t2})^T$ can be estimated recursively using the methods presented in chapter 3 of StOaL. Assume that we start with an initial estimate $\bar{\theta}_0$ of the vector of coefficients. We are also going to need a three-by-three matrix B^0 that for now we can assume is a scaled identity matrix (we provide a better idea below).

The vector of coefficients $\bar{\theta}_t = (\bar{\theta}_{t0}, \bar{\theta}_{t1}, \bar{\theta}_{t2})^T$ can be estimated recursively using the methods presented in chapter 3 of StOaL. Assume that we start with an initial estimate $\bar{\theta}_0$ of the vector of coefficients. We are also going to need a three-by-three matrix B^0 that for now we can assume is a scaled identity matrix (we provide a better idea below).

The basic updating equation for $\bar{\theta}_t$ is given by

$$\bar{\theta}_{t+1} = \bar{\theta}_t - H_t \phi_t \varepsilon_{t+1}, \tag{8.4}$$

The error $\hat{\varepsilon}_t$ is computed using

$$\varepsilon_{t+1} = \bar{\theta}_t^T \phi_t - p_{t+1}. \tag{8.5}$$

The three-by-three matrix H_t is computed using

$$H_t = \frac{1}{\gamma_t} B_{t-1}, \tag{8.6}$$

where the matrix B_n is computed recursively using

$$B_t = B_{t-1} - \frac{1}{\gamma_t}(B_{t-1}\phi_t(\phi_t)^T B_{t-1}). \tag{8.7}$$

The variable γ_t is a scalar computed using

$$\gamma_t = 1 + (\phi_t)^T B_{t-1} \phi_t. \tag{8.8}$$

These equations need initial estimate for $\bar{\theta}_0$ and B_0. One way to do this is to collect some initial data and then solve a static estimation problem. Assume you observe K prices. Let Y_0 be a K-element column factor of the observed prices $p_3, p_4, \ldots, p_{K+3-1}$ (we have to start with the third price because of our need for the trailing three prices in our model).

Then let X_0 be a matrix with K rows, where each row consists of p_k, p_{k-1}, p_{k-2}. Our best estimate of $\bar{\theta}$ is given by the normal equations

$$\bar{\theta}_0 = [(X_0)^T X_0]^{-1} (X_0)^T Y_0. \tag{8.9}$$

Finally let $B_0 = [(X_0)^T X_0]^{-1}$, which shows that the matrix B_t is the time t estimate of $[(X_t)^T X_t]^{-1}$.

There are entire families of time series models that capture the relationship of variables over time. If we were to just apply these methods directly to price data, the results would be quite poor. First, the prices are not normally distributed. Second, the prices cannot be negative, but a direct application of this model would be very likely to produce negative prices if the variance σ_ϵ^2 was calibrated to the high-noise of this type of data. Finally, the behavior of the jumps in prices over time would not be realistic.

- Mean reverting models
  - » Basic mean reverting model (Ornstein-Uhlenbeck process)

$$p_{t+1} = p_t + \beta\,(\bar{\mu}_t - p_t) + \varepsilon_{t+1}$$

  - » where

$$\bar{\mu}_{t+1} = (1-\alpha)\bar{\mu}_t + \alpha p_{t+1}$$

β is the rate of mean reversion. Note that if $p_t > \bar{\mu}_t$, then p_{t+1} will trend lower (and vice versa). This is what is meant by "mean reversion"

- Mean reverting with jump diffusion

$$p_{t+1} = p_t + \beta\,(\bar{\mu}_t - p_t) + \varepsilon_{t+1} + J_{t+1}\varepsilon^J_{t+1}$$

  - where

$$\varepsilon_t \sim N(0,\sigma^2_\varepsilon)$$

$$J_t = \begin{cases} 1 & w.p.\ \eta \\ 0 & w.p.\ 1-\eta \end{cases}$$

$$\varepsilon^J_t \sim \lambda e^{-\lambda x}$$

  - To estimate η, pass through data 2-3 times, discarding data that is more than $3\sigma_\varepsilon$ from the mean. Then η is the fraction of data we have discarded.
  - Fit λ so that $\frac{1}{\lambda}$ = variance of the discarded data.

8.3.2 Jump diffusion

A major criticism of the linear model above is that it does a poor job of capturing the large spikes that are familiar in the study of electricity prices. A simple idea for overcoming this limitation is to use what is known as a *jump diffusion model*, where we add another noise term to equation (8.3) giving us

$$p_{t+1} = \bar{\theta}_{t0}p_t + \bar{\theta}_{t1}p_{t-1} + \bar{\theta}_{t2}p_{t-2} + \varepsilon_{t+1} + \mathbb{I}_t\varepsilon^J_{t+1}. \tag{8.10}$$

Here, the indicator variable $\mathbb{I}_t = 1$ with some probability p^{jump}, and the noise ε^J_{t+1} is normally distributed with mean μ^{jump} (which is typically much larger than zero) and variance $(\sigma^{jump})^2$ which is quite large.

We have to estimate the jump probability p^J, and the mean and variance $(\mu^J, (\sigma^{jump})^2)$. This is done by starting with a basic model where $p^{jump} = 0$. We use this basic model to estimate σ^2_ϵ. We then choose some tolerance such as three standard deviations (that is, $3\sigma_\epsilon$), and assume any observations outside of this range should be attributed to a different source of noise. Let p^{jump} be the fraction of time periods where these observations occur. Then, compute the mean and standard deviation of these observations to get $(\mu^J, (\sigma^{jump})^2)$.

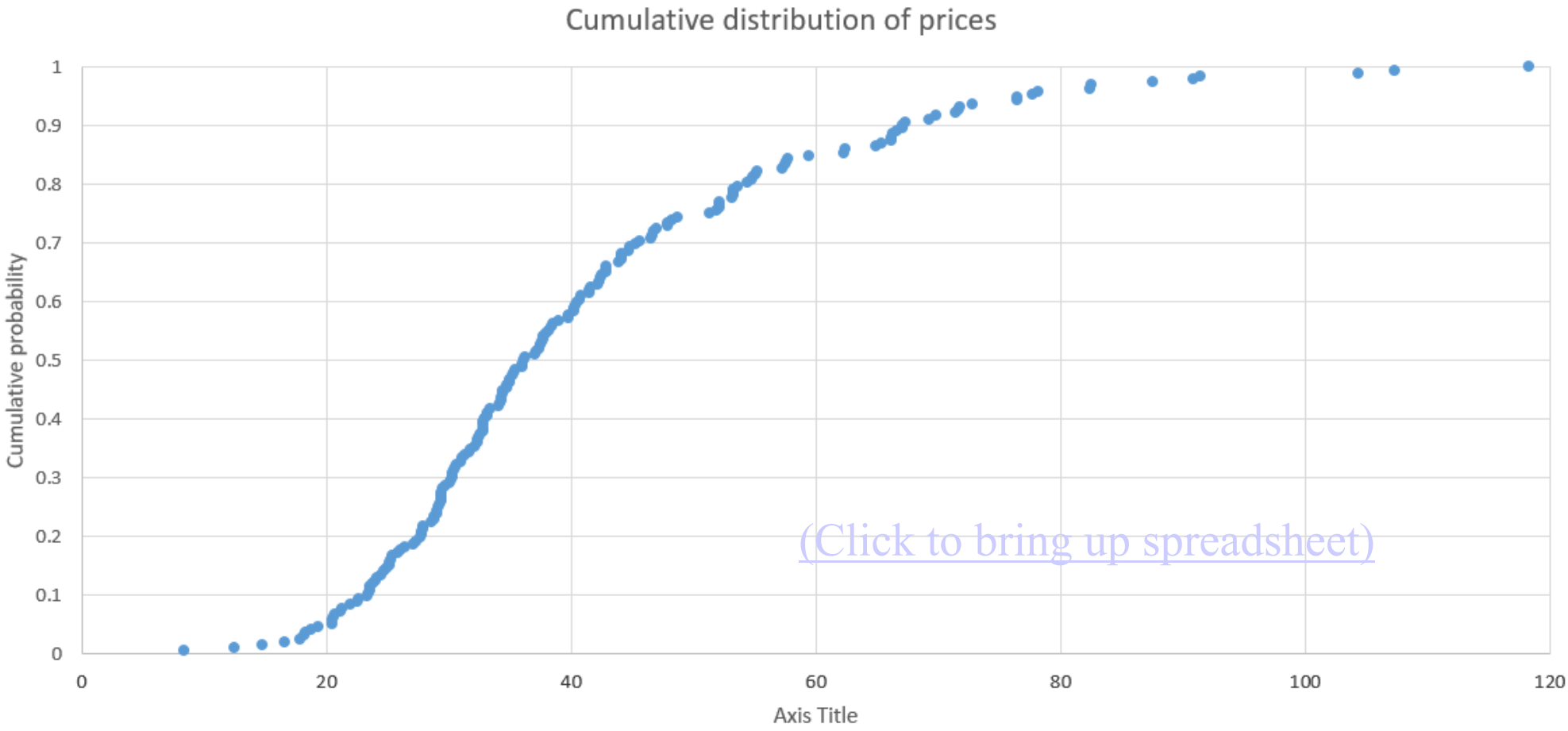
We do not stop here. After taking these extreme variations from the data, we should re-fit our linear model without these observations. Standard practice is to repeat this process several times until these estimates stop changing.

8.3.3 Empirical distribution

While it may be possible to fit other parametric distributions, a powerful strategy is to numerically compute the cumulative distribution from the data, creating what is often called an *empirical distribution*. To compute this, we simply sort the prices from smallest to largest. Denote this ordered sequence by $\tilde{p}_t$, where $\tilde{p}_{t-1} \leq \tilde{p}_t$. Let $T = 105,210$ which is the number of 5-minute time periods in a year, which means that the percentage of time periods with a price less than $\tilde{p}_t$ is t/T. We can then create a cumulative distribution using

$$F_P(\tilde{p}_t) = \frac{t}{T},$$

which is illustrated in figure 8.4.

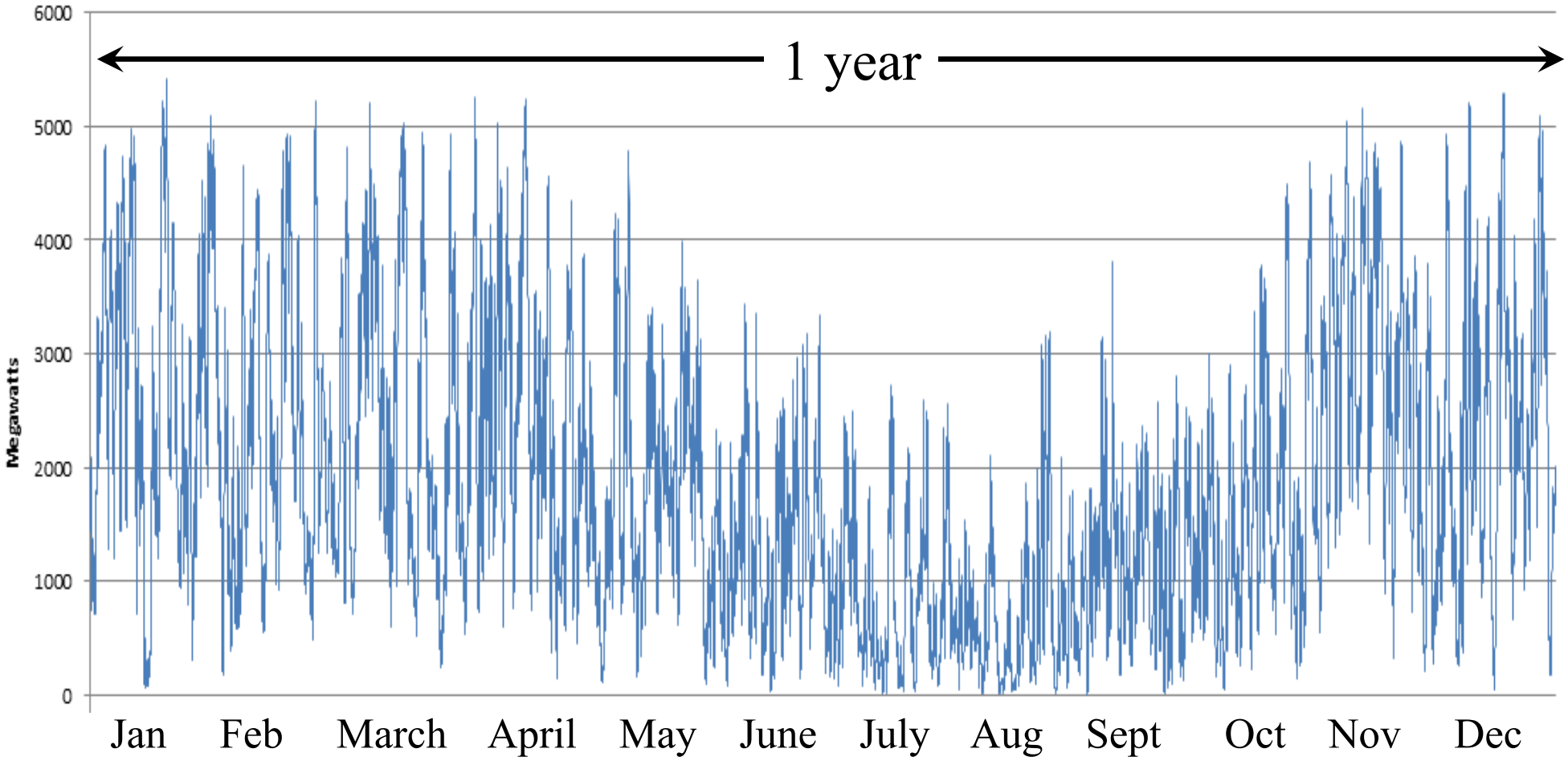


NORTA

Normal to anything

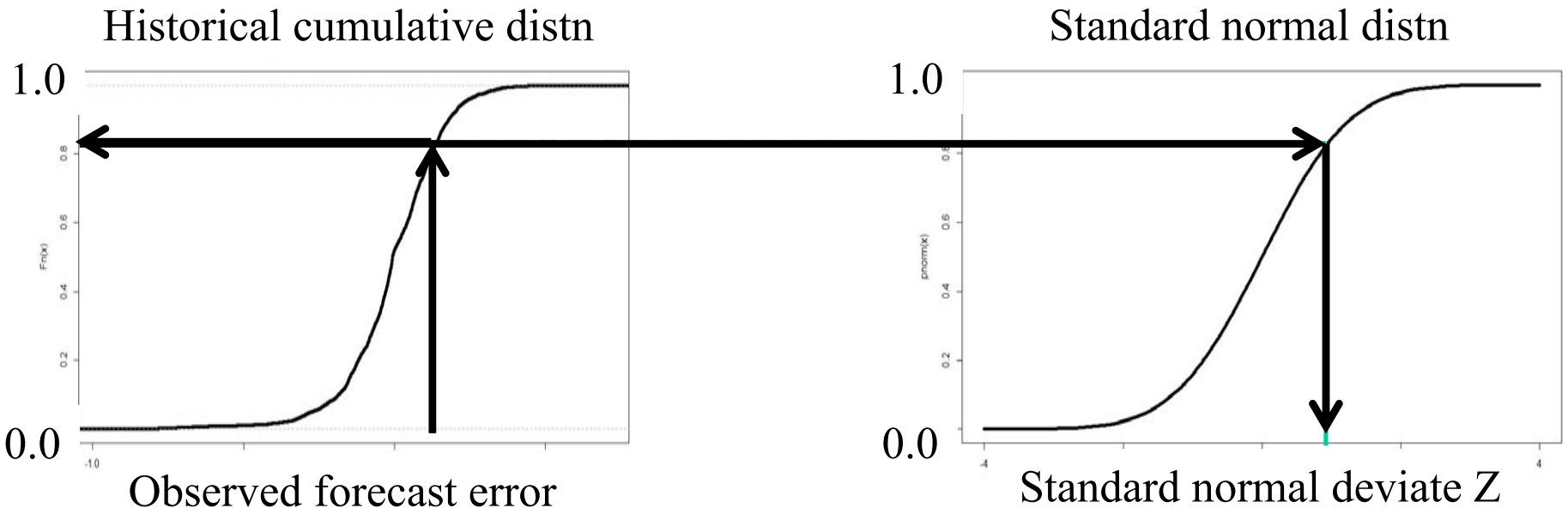
Modeling wind forecasting errors

- Wind power from all PJM wind farms



Modeling wind forecasting errors

- Data transformation
  - » Observed errors are transformed to normally distributed errors using quantile mapping:

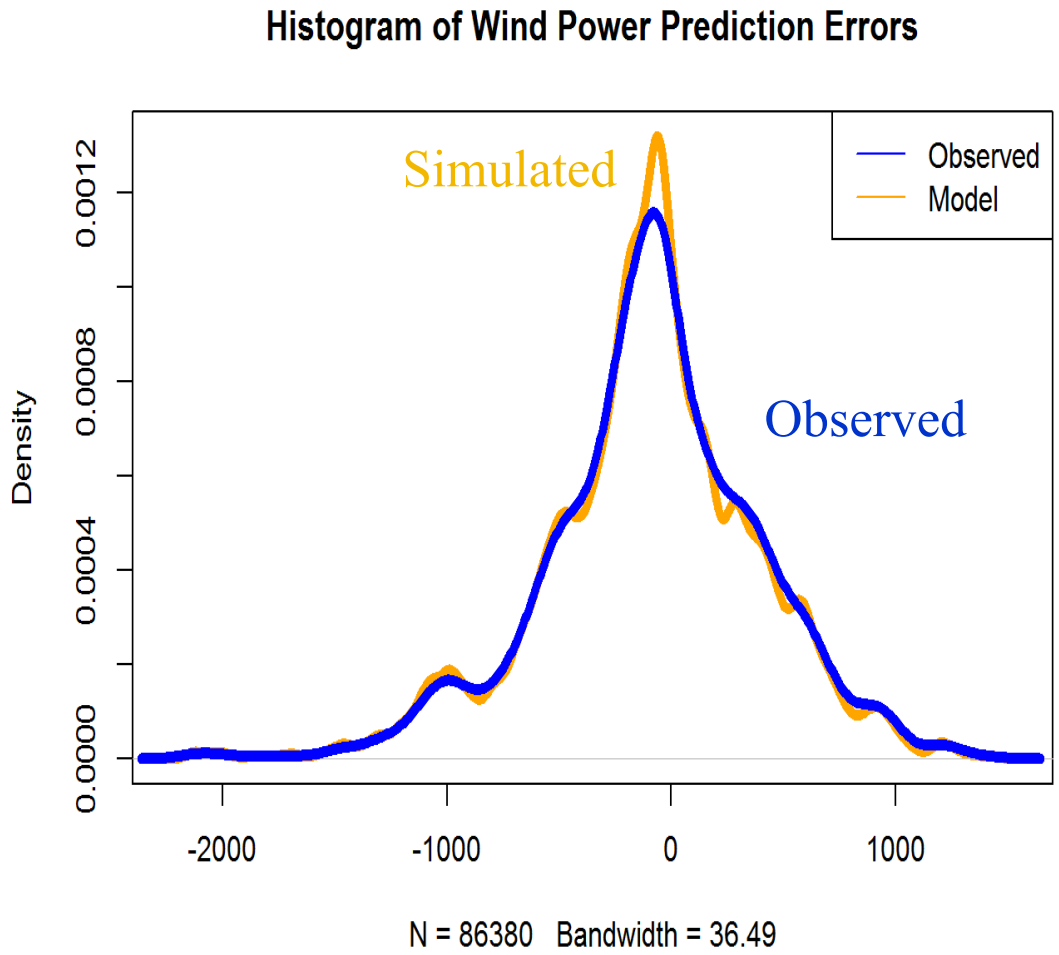


- Talk through problem set

Simulating onshore wind

- Modeling errors

Error distribution



Modeling uncertainty

Probability distributions

Modeling uncertainty

- Types of distributions
  - » Probability distributions come in different forms:
    - Classical "thin tailed" distributions –
      - – Exponential family
        - » Normal, exponential, gamma
        - » Uniform
      - – Discrete variants
    - Heavy-tailed distributions
      - – Cauchy distribution (may have infinite variance)
      - – "Jump diffusion" – Sum of low-variance normally distributed error, plus a high-variance error that occurs with low probability
    - Spikes
    - Bursts
    - Rare events

- Capturing distributions
  - » Data driven
    - Just use observations rather than a mathematical model
    - Implies on-line process
  - » Parametric models
    - Exponential family – Fitting parameters
    - Capturing uncertainty in parameters
  - » Sampled representations
  - » Ranges ("robust optimization")
    - 95% confidence ranges for random variables
    - How to handle simultaneous distributions for multiple random variables?

- Data driven

- Parametric models – known parameters

Parametric distribution

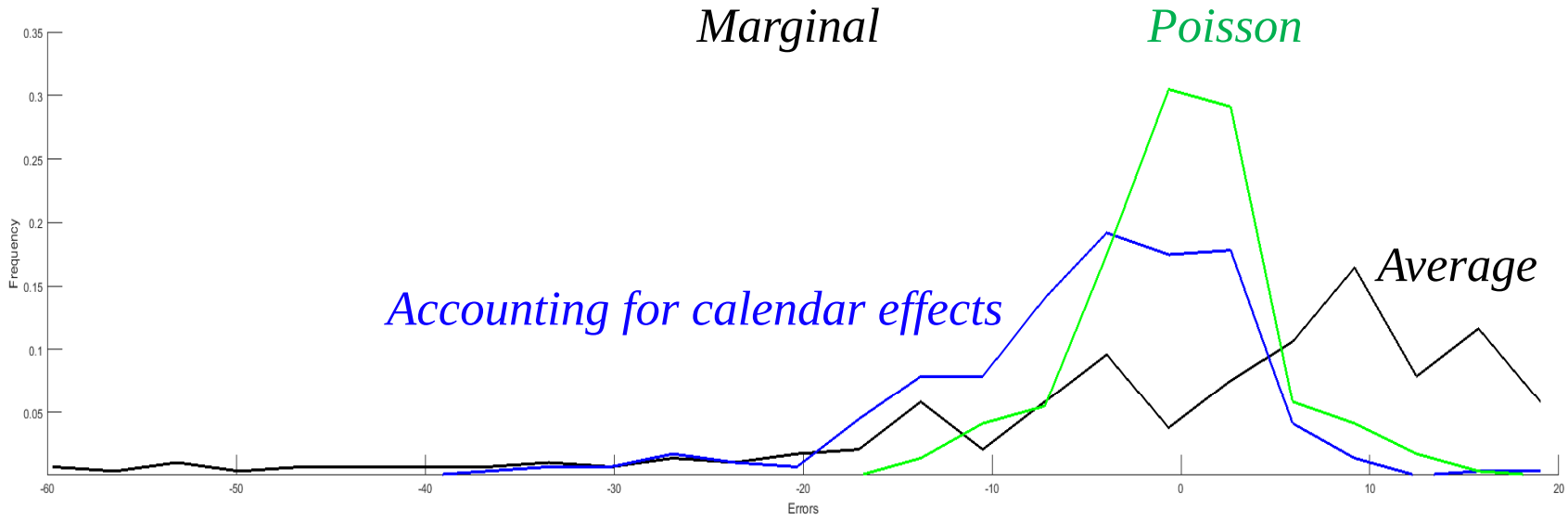
Uncertain parameters – fitting Poisson to actual data

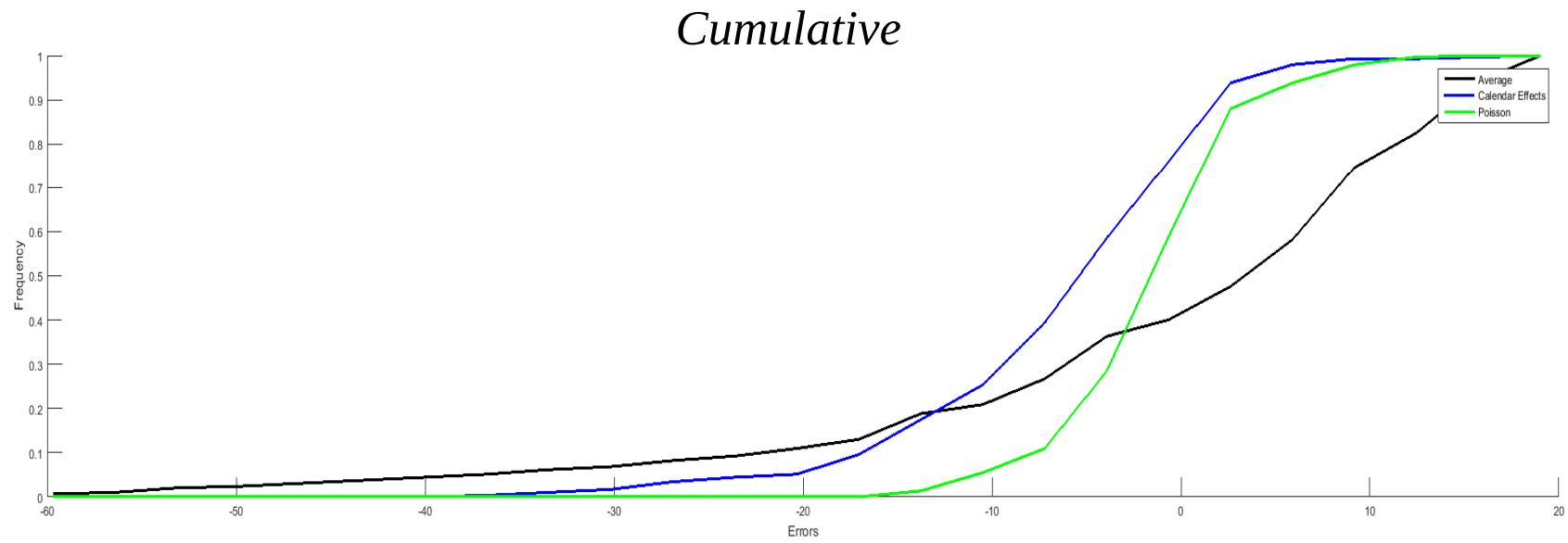
Booking process

- Error distributions:
  - » Ways of estimating the error distribution:
    - Relative to the forecast (presumably made before any bookings have been made, but we can define a forecast at any time up to the stay date).
    - Relative to a base average (which loses the benefit of seasonal adjustments (this would be the worst case).
  - » Best case, with a perfect forecast, is that we should observe an error distribution that follows a Poisson.
    - This assumes we can perfectly predict the *booking rate*.
    - Of course, we cannot perfectly predict the booking rate, so the observed error distribution will show a variance higher than that predicted by a Poisson.

Booking process

- Error distributions





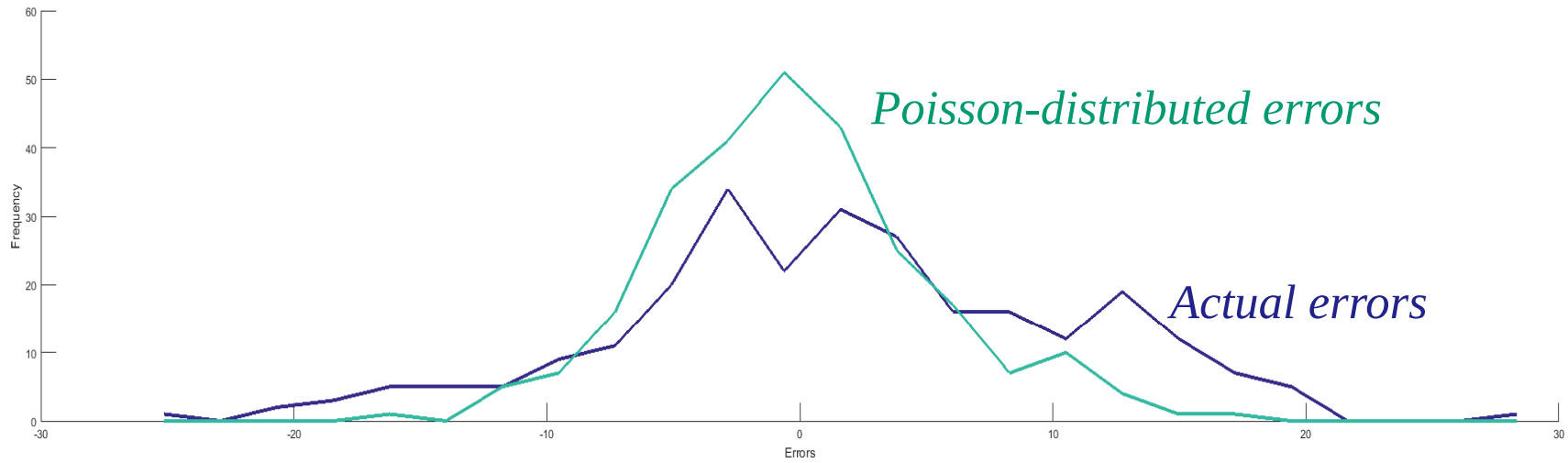
Booking process

- Fixing the variability
  - » We need a booking model that produces the same degree of variability as we observe in real data.
  - » Assuming a Poisson distribution does not introduce enough variability (but we still have to assume Poisson arrivals for our Bayesian updating formula).
  - » We are going to introduce additional variability by randomizing the arrival rate λ_t.
  - » You will work this out on your problem set!

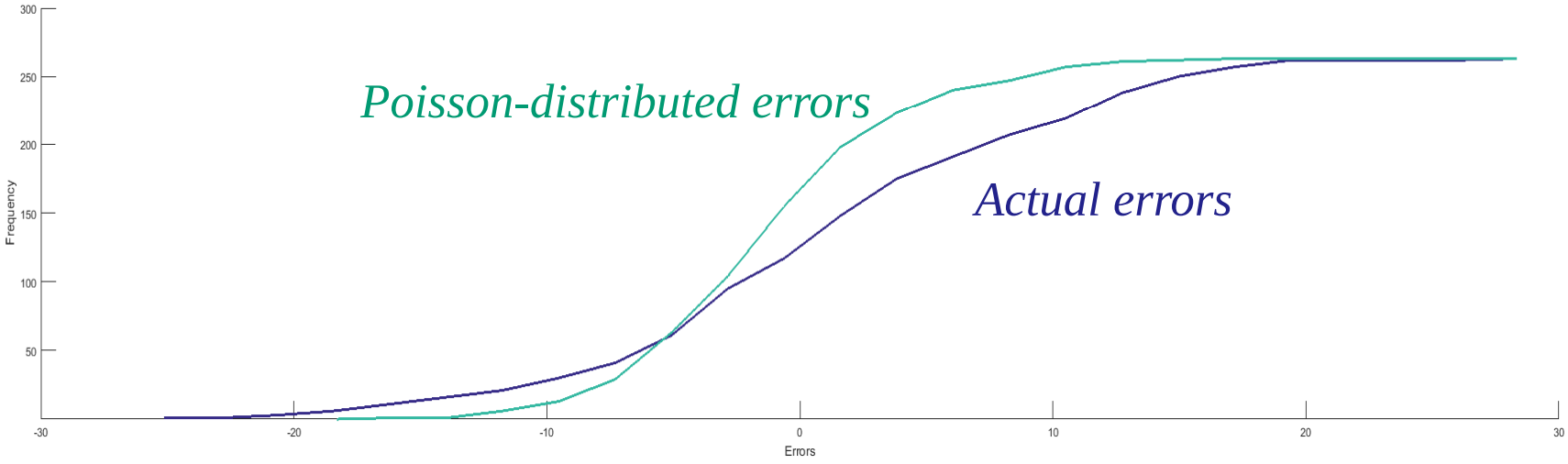
Booking process

- Actual error compared to Poisson error

Marginal

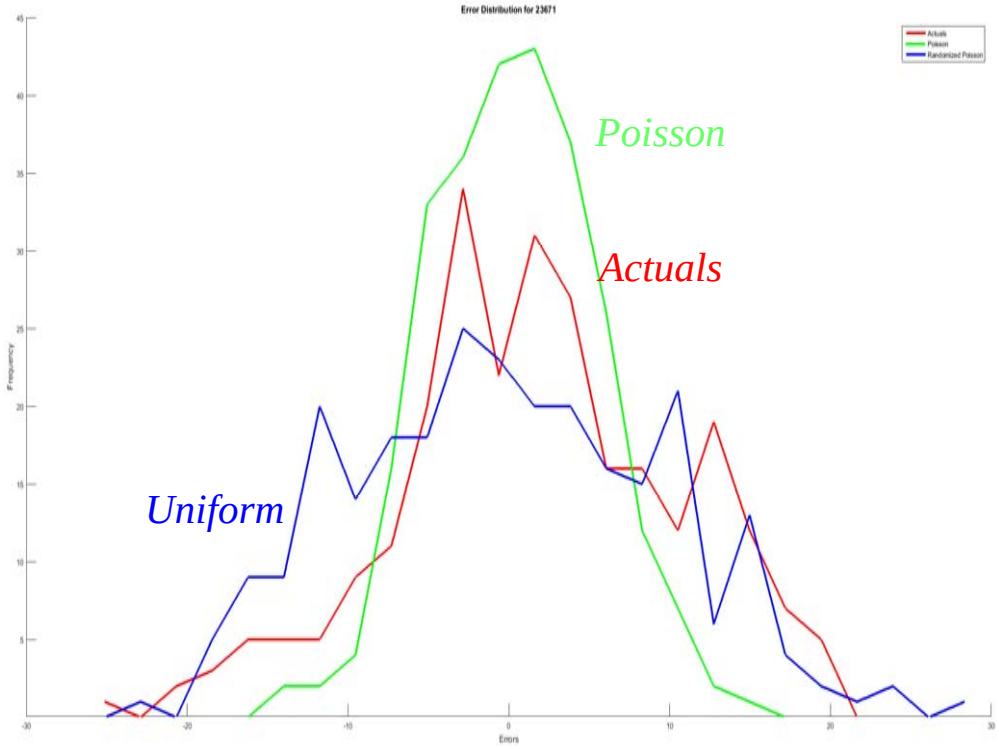


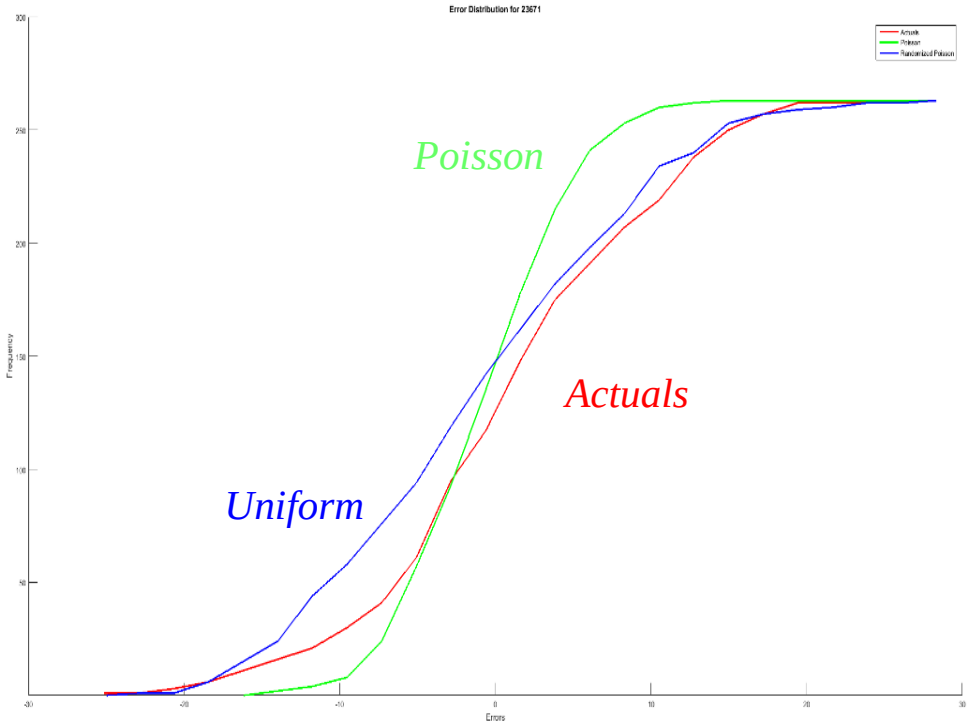
Cumulative



Booking process

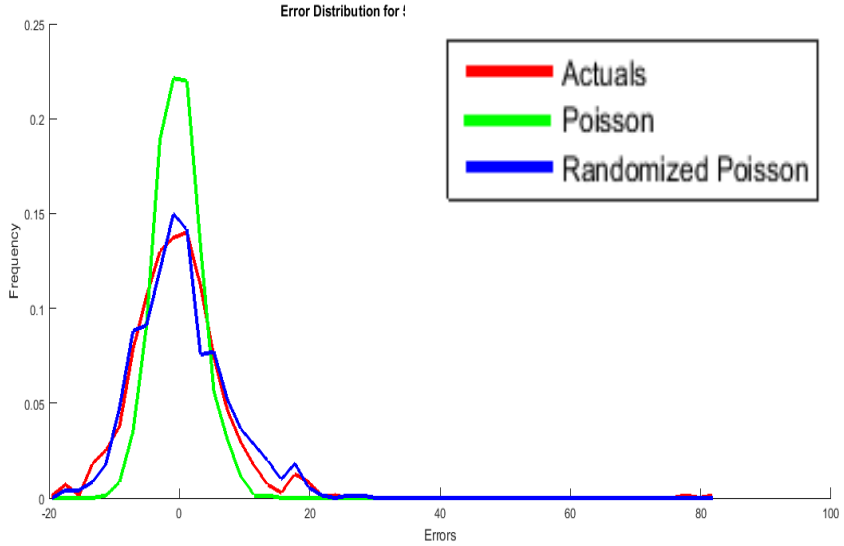
- Fix using uniformly distributed λ

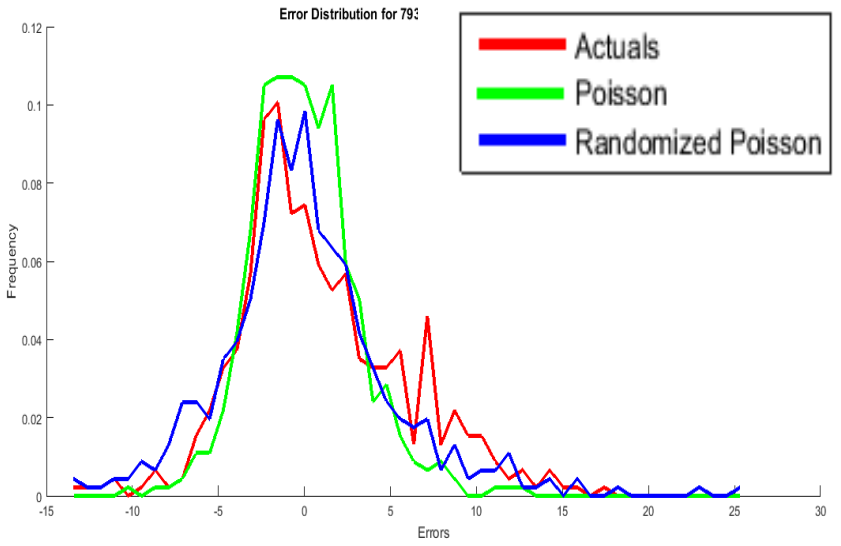


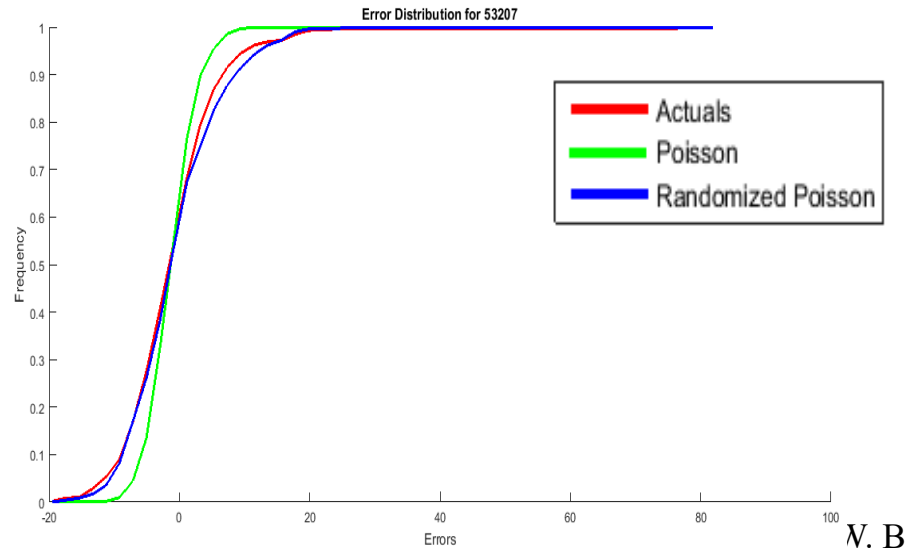


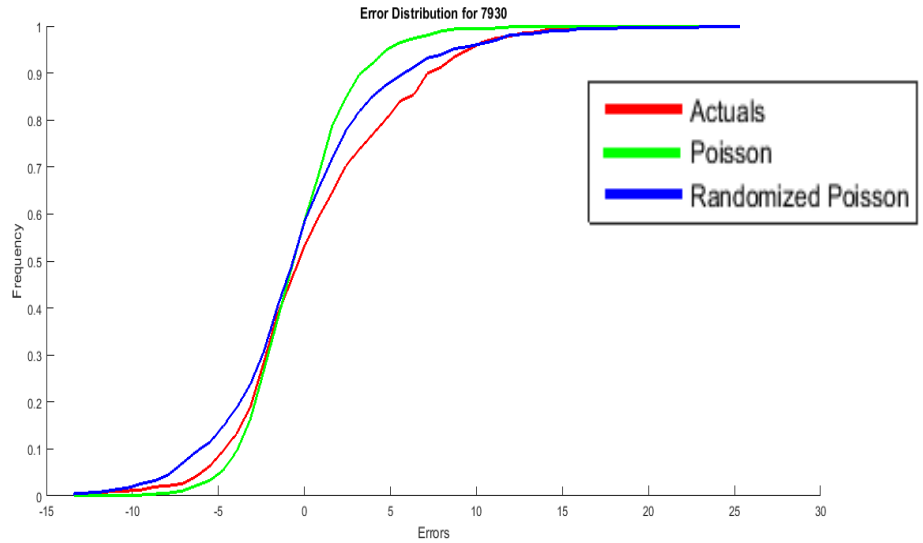
Booking process

- Fix using beta-distributed λ









N. B

Sampled distributions

- Sampled distributions

- Ranges
  - » Example:
    - Design a building (where x are design parameters) to withstand the worst wind (w captures wind speed and direction).
  - » Robust optimization

$$\min_{x \in X} \max_{w \in W(\theta)} F(x, w)$$

  - » We construct the set $W(\theta)$ to capture reasonable ranges of what w might be.
    - Boxes

$$w^{min} \leq w_i \leq w^{max}$$

    - Ellipse

Gaussian process regression (GPR)

Gaussian process regression

Cholesky decomposition

We could assume that the noise terms $\varepsilon_{t+1,i}$ are independent across the assets $i \in \mathcal{I}$, but a more realistic model would be to assume that the prices of different assets are correlated. Let

$$\sigma_{ij} = Cov_t(p_{t+1,i}, p_{t+1,j})$$

be the covariance of the random prices $p_{t+1,i}$ and $p_{t+1,j}$ for assets i and j given what we know at time t. Assume for the moment that we know the covariance matrix Σ, perhaps by using a historical datset to estimate it (but holding it fixed once it is estimated).

We can use the covariance matrix to generate sample realizations of correlated prices using a technique called Cholesky decomposition. It proceeds by creating what we call the "square root" of the covariance matrix Σ which we store in a lower triangular matrix L. In python, using the NumPy package, we would use the python command

$$L = \texttt{scipy.linalg.cholesky}(\Sigma^X, lower = True)$$

The matrix L allows us to obtain the matrix Σ using

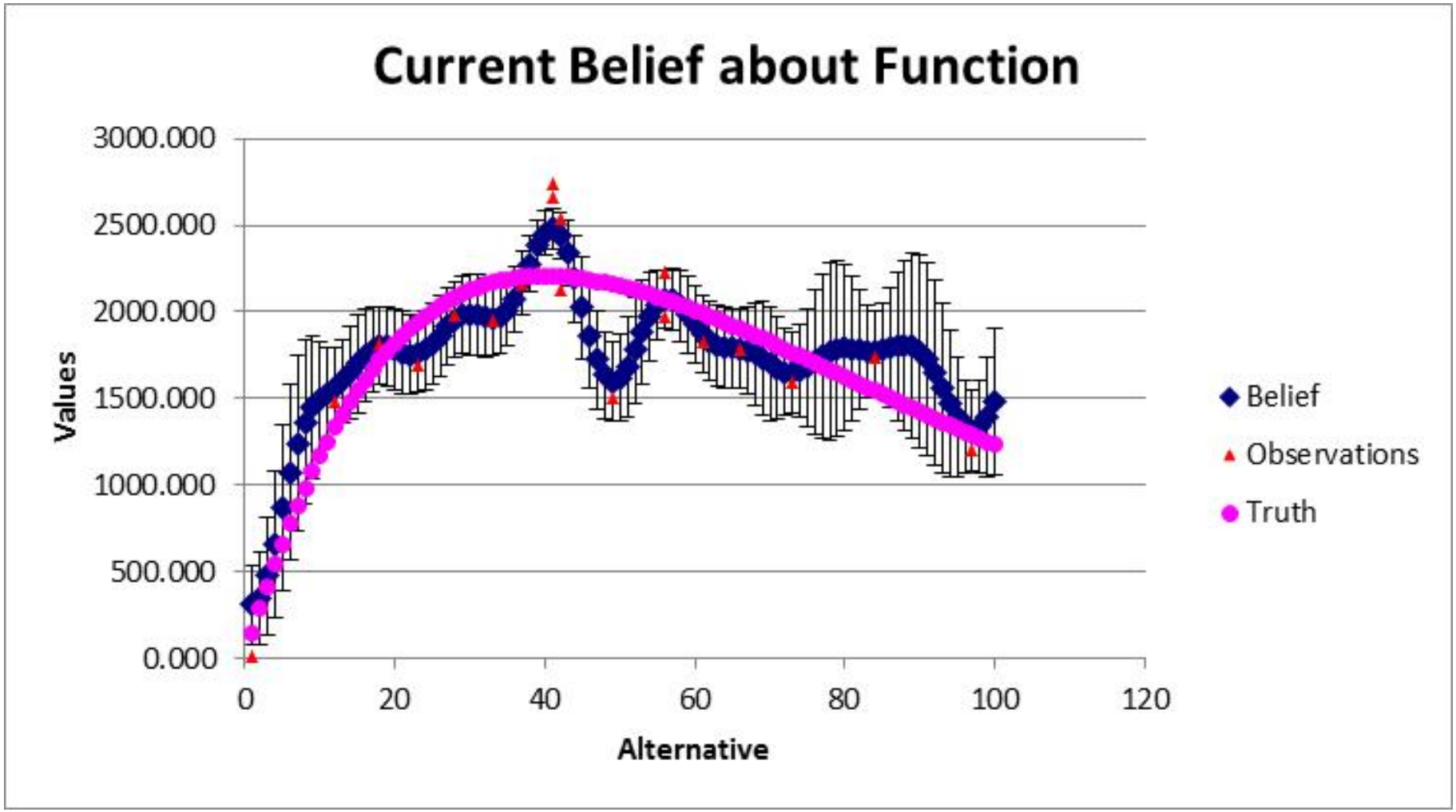
$$\Sigma = L^T L.$$

Now let Z be a vector of random variables, one for each asset, where $Z_i \sim N(0,1)$ (virtually every computer language has routines for creating random samples from normal distributions with mean 0, variance 1). Let p_t, p_{t+1} and Z be column vectors (dimensioned by the number of assets). We first create a sample $\hat{Z}$ by sampling from $N(0,1)$ $|\mathcal{I}|$ times. Our sample of prices p_{t+1} is then given by

$$p_{t+1} = p_t + L\hat{Z}.$$

Gaussian process regression

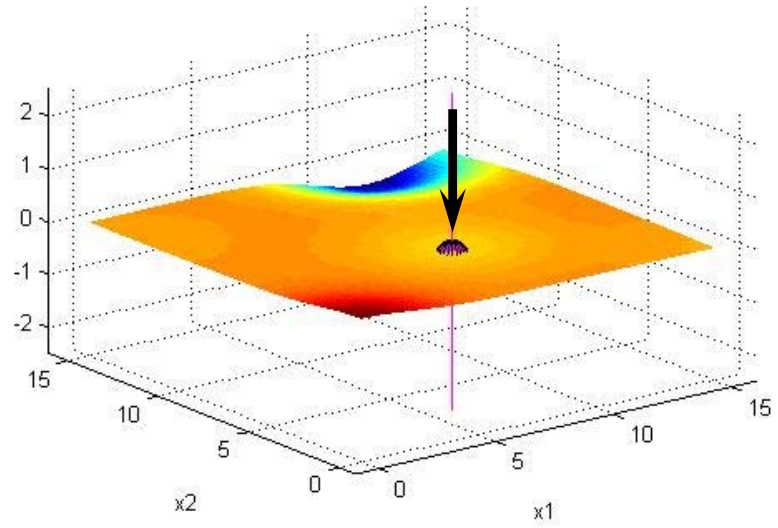
- Cholesky decomposition
  - » Useful for simulating smooth surfaces



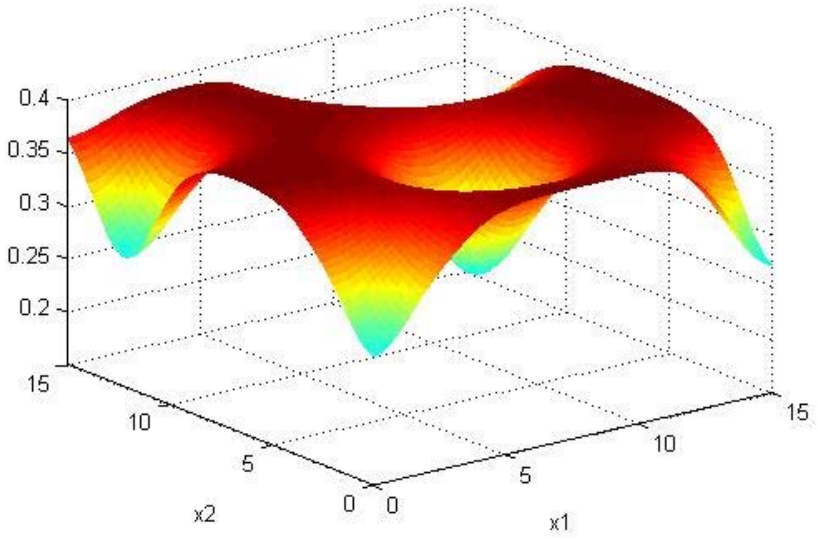
Gaussian process regression

- After five measurements:

Estimated concentration



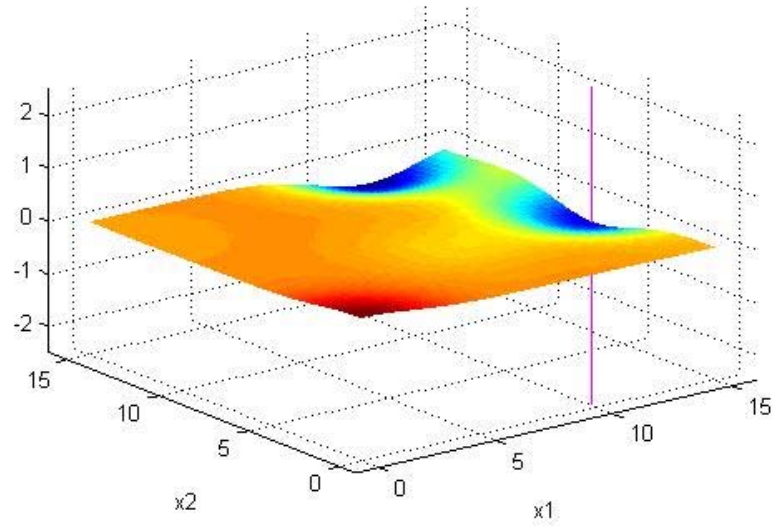
Knowledge gradient



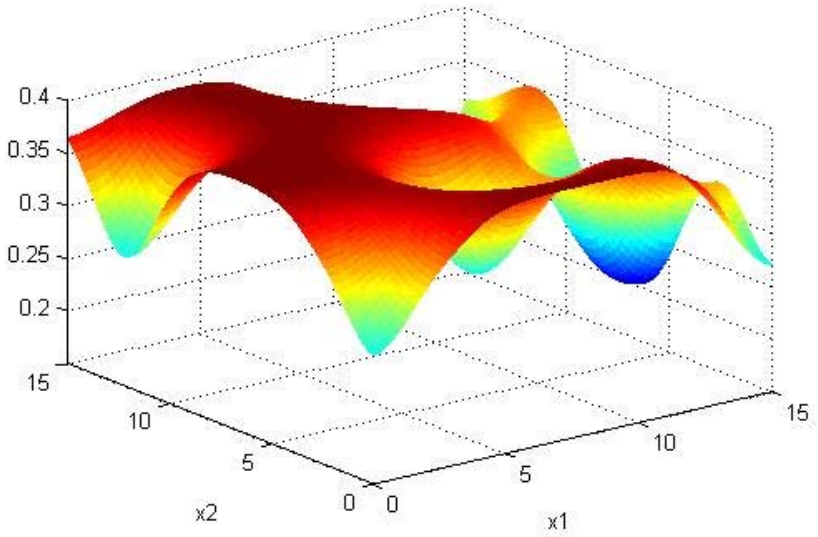
Gaussian process regression

- After six samples

Estimated concentration



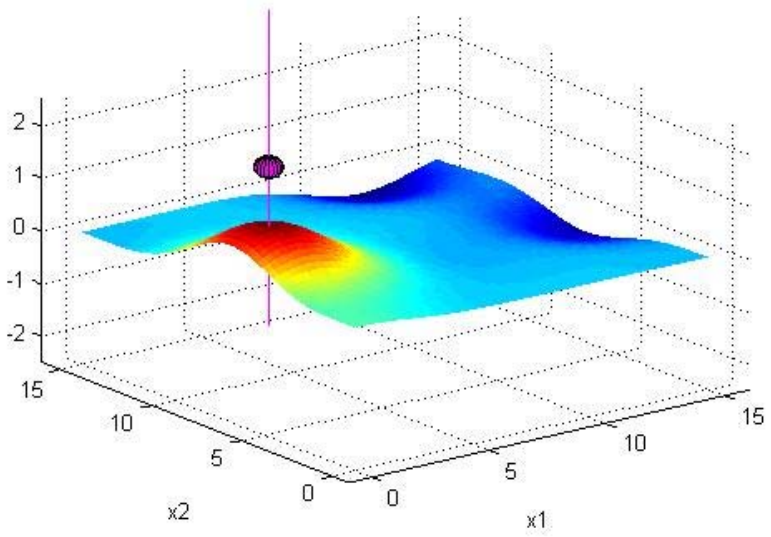
Knowledge gradient



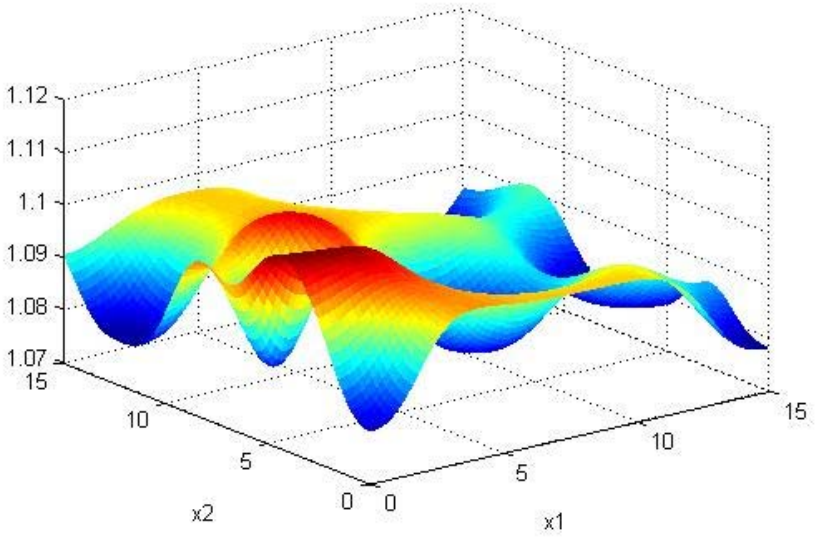
Gaussian process regression

- After seven samples

Estimated concentration



Knowledge gradient

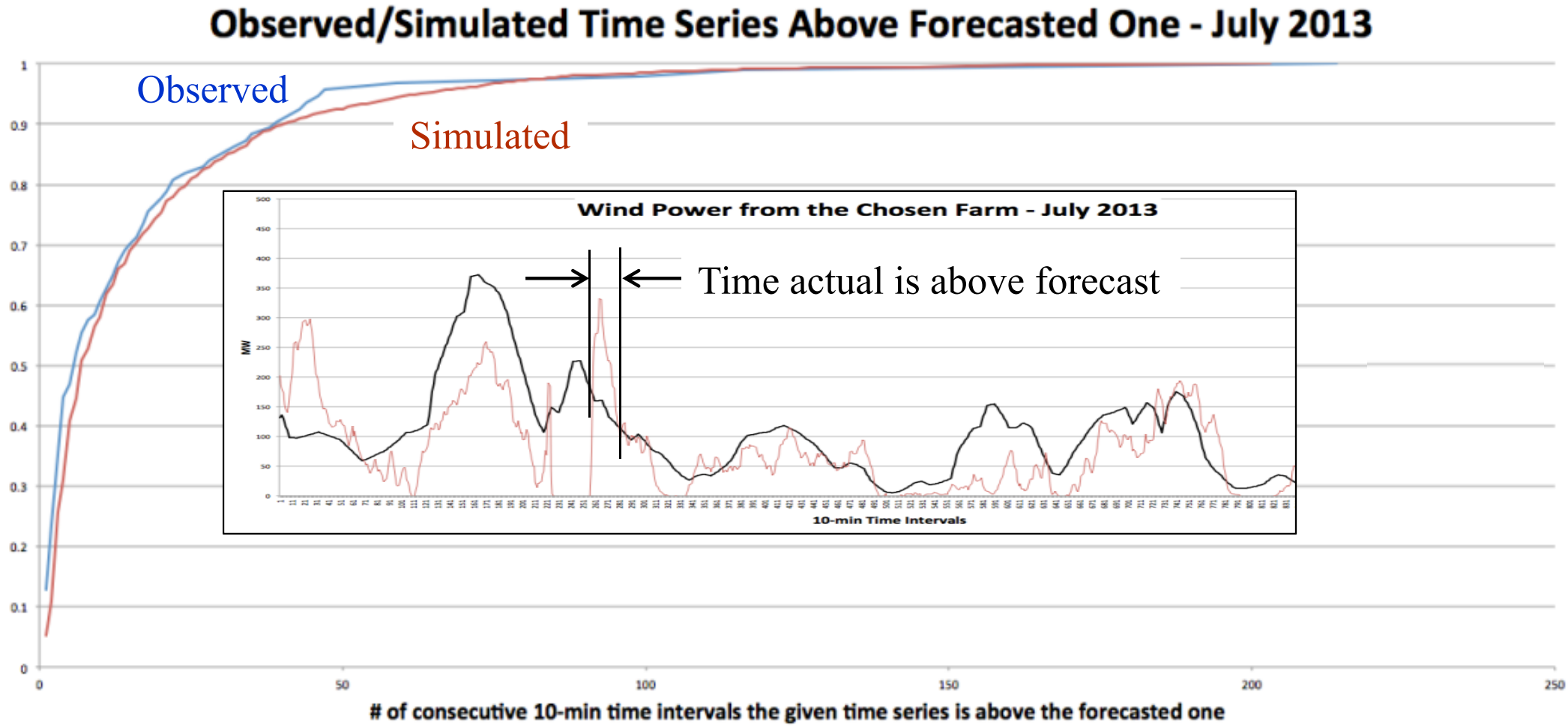


Hidden semi-Markov

Crossing times

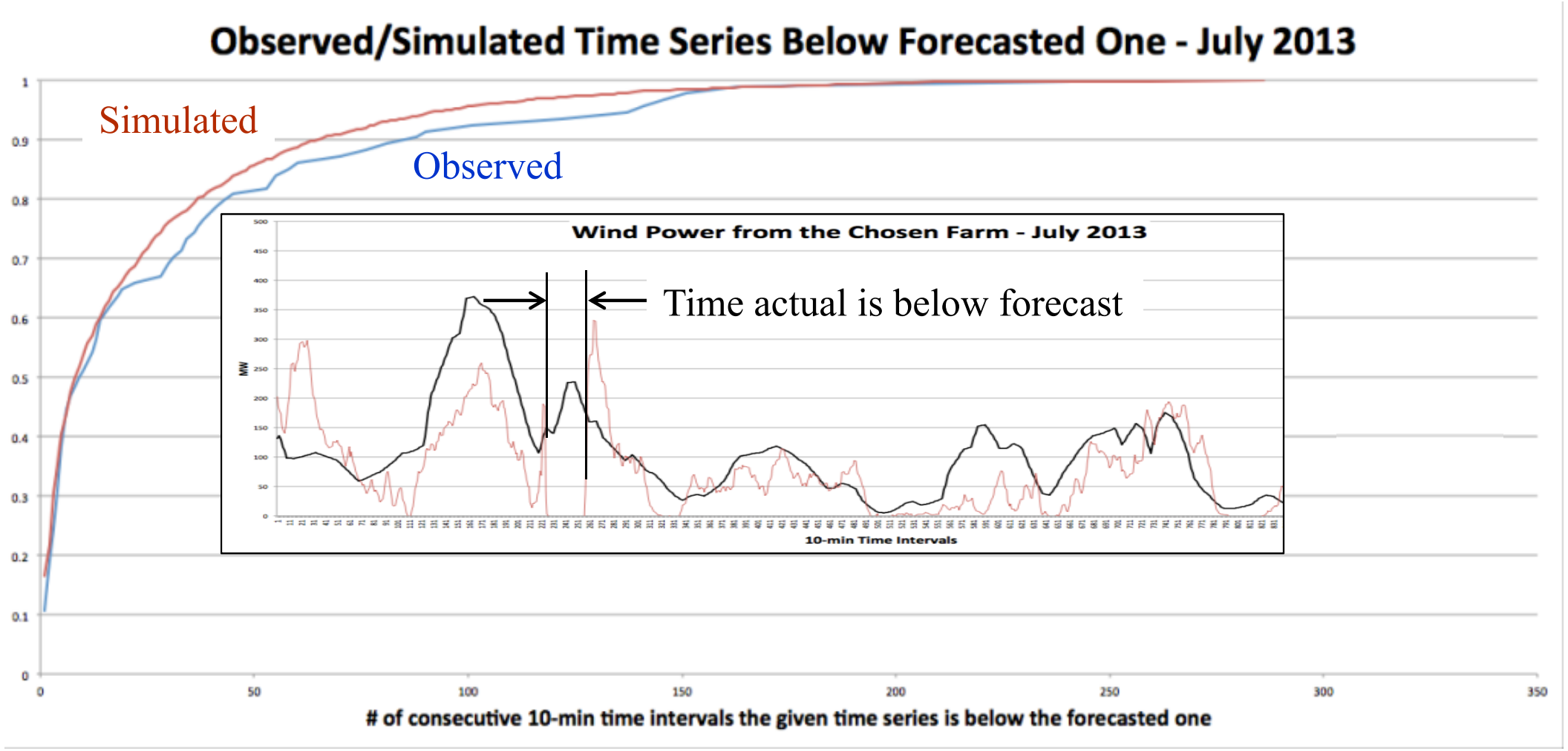
Modeling wind forecasting errors

- Cumulative histogram of the # of consecutive time intervals the observed/simulated time series is above the forecasted one (chosen farm only):



Modeling wind forecasting errors

- Cumulative histogram of the # of consecutive time intervals the observed/simulated time series is below the forecasted one (chosen farm only):



Modeling wind forecasting errors

- Challenges
  - » We would like to replicate:
    - Error distribution
    - Upcrossing distribution
    - Downcrossing distribution
  - » We would like to replicate performance at two levels:
    - Each wind farm
    - All wind farms when aggregated together
- What has *not* worked:
  - » ARMA, ARIMA, GARCH, even when using quantile transformation to handle non-normality of wind errors
    - Times series models struggle to capture the longer-term behavior

10.3.2 Hidden semi-Markov model

A challenge when developing stochastic models in energy is capturing a property known as a *crossing time*. This is the time that an actual process (e.g. price or wind speed) is above or below some benchmark such as a forecast. Figure 10.3 illustrates an up-crossing time for a wind process.

Replicating crossing times using standard time series modeling proved unsuccessful. What did work was the development of Markov model with a hidden state variable S_t^C which is calibrated to capture the dynamics of the process moving above or below the benchmark. The process uses the following steps:

Step 1 Comparing the actual process to the benchmark, find the times at which the actual process moves above or below the benchmark, and output a dataset that captures whether the process was above (A) or below (B) and for how long. Aggregate these periods into three buckets (S/M/L) for short/medium/long, and label each segment with A/B-S/M/L, creating six states. These are called "hidden states" because, while we will know at time t if the actual process is above or below the benchmark, we will not know if the length is short, medium or long until after the process crosses the benchmark.

Step 2 Using the historical sequence of S_t^C, compute a one-step transition matrix

$$P^C[S_{t+1}^C|S_t^C] = \text{The probability that the crossing process takes on value } S_{t+1}^C \text{ given that it is currently in state } S_t^C.$$

Step 3 Aggregate the actual process (e.g. wind speed) into, say, five buckets based on the empirical cumulative distribution. Let W_t^g be the aggregated wind speed (a number from 1 to 5).

Step 4 From history, compute the conditional distribution of wind speed given W_t^g and S_t^C,

$$F^W[W_{t+1}|W_t^g, S_t^C] = \text{Empirical cumulative distribution of the wind speed } W_{t+1} \text{ given } W_t^g \text{ and } S_t^C.$$

- Hidden semi-Markov model (HSMM)
  - » We use a hybrid Markov chain model with two stage variables:
    - The crossing state S_t^C:

$$S_t^C = \begin{cases} A\,|\,(\text{S/M/L}) & \text{If we are in an "above the forecast" state } |\text{ S/M/L} \\ B\,|\,(\text{S/M/L}) & \text{If we are in a "below the forecast" state } |\text{ S/M/L} \end{cases}$$

A/B means above or below
S/M/L means a short, medium or long crossing distribution
$\mathbb{P}(S_{t+1}^C \,|\, S_t^C)$ is estimated from historical data.

    - The wind speed W_t given the crossing state:

W_t = Wind speed at time t.

W_t^g = Wind speed aggregated into 5 ranges.

$\mathbb{P}(W_{t+1} \,|\, W_t^g, S_t^C)$ = Density of W_{t+1} given W_t^g and S_t^C

- Numerical example
  - » Imagine sequence of A/B (above/below) intervals (which alternate), combined with S/M/L (short/medium/long):

(A,S), (B,M), (A,S), (B,L), (A,L), (B,S), (A,M), (B,M), (A,S) …

  - » Now set up matrix and count how many times each transition happens:

	(A,S)	(A,M)	(A,L)	(B,S)	(B,M)	(B,L)
(A,S)	0	0	0	0	1	0
(A,M)	0	0	0	0	1	0
(A,L)	0	0	0	1	0	0
(B,S)	0	1	0	0	0	0
(B,M)	2	0	0	0	0	0
(B,L)	0	0	1	0	0	1

  - » Now normalize rows so they sum to 1.

- Notes:
  - » After computing $P(S_{t+1}^C|S_t^C)$, next need to get conditional distribution of *change* in wind given the aggregated wind speed W_t^g.
    - For each (W_t, W_{t+1}) pair, get the change in wind speed $W_{t+1} - W_t$, and then compute the distribution conditioned on W_t^g.
    - This gives us

$$W_t = \text{Wind speed at time } t.$$

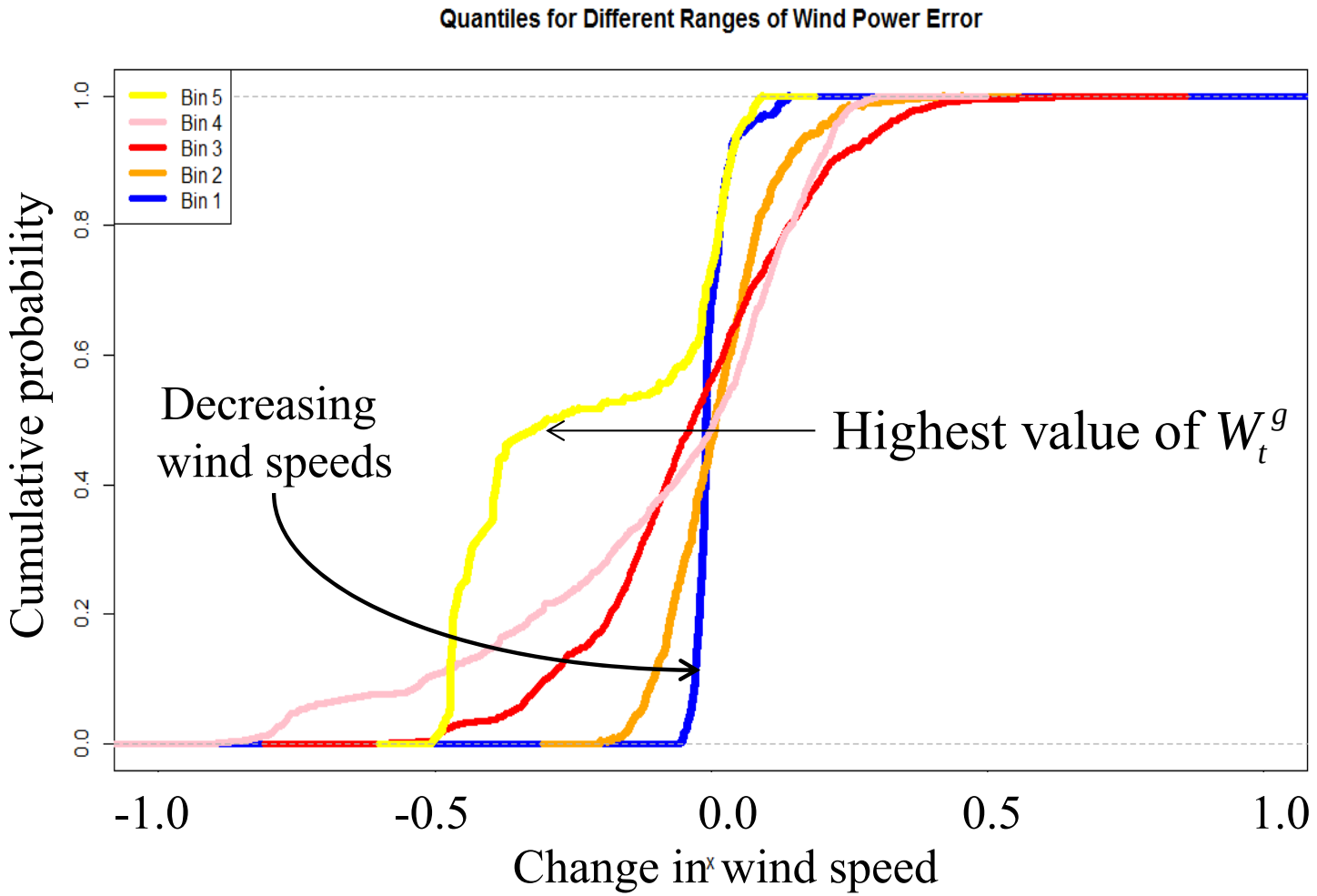
$$W_t^g = \text{Wind speed aggregated into 5 ranges.}$$

$$\mathbb{P}(W_{t+1} \,|\, W_t^g, S_t^C) = \text{Density of } W_{t+1} \text{ given } W_t^g \text{ and } S_t^C$$

    - We can now simulate the process moving forward in time. Not that we are simulating whether an interval is S/M/L, but we would not know this in advance. This is why it is a *hidden state*.

Simulating onshore wind

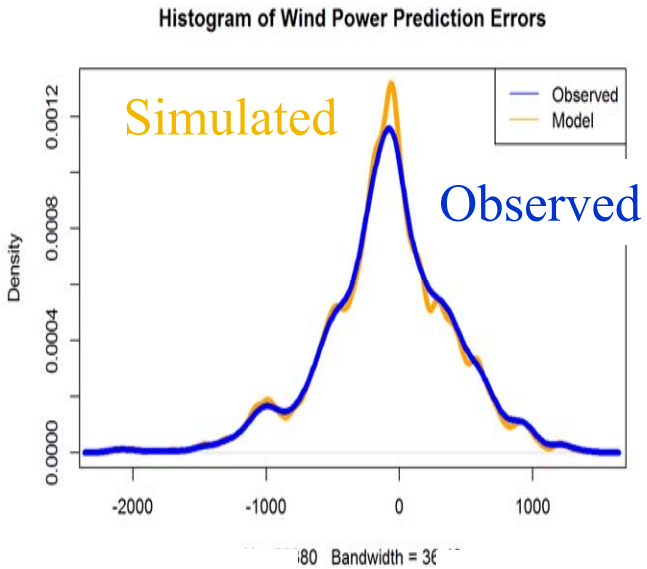
- Conditional cumulative distribution of wind error
  - » Conditioned on the aggregated wind state W_t^g



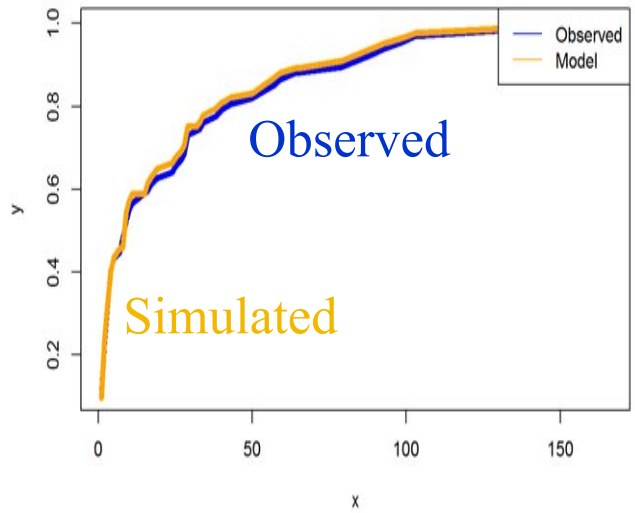
Modeling wind forecasting errors

- Error distribution and crossing times

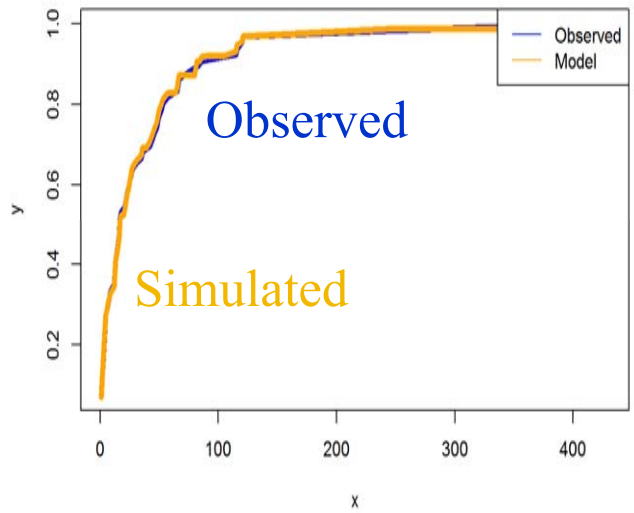
Error distribution



Upcrossing distribution



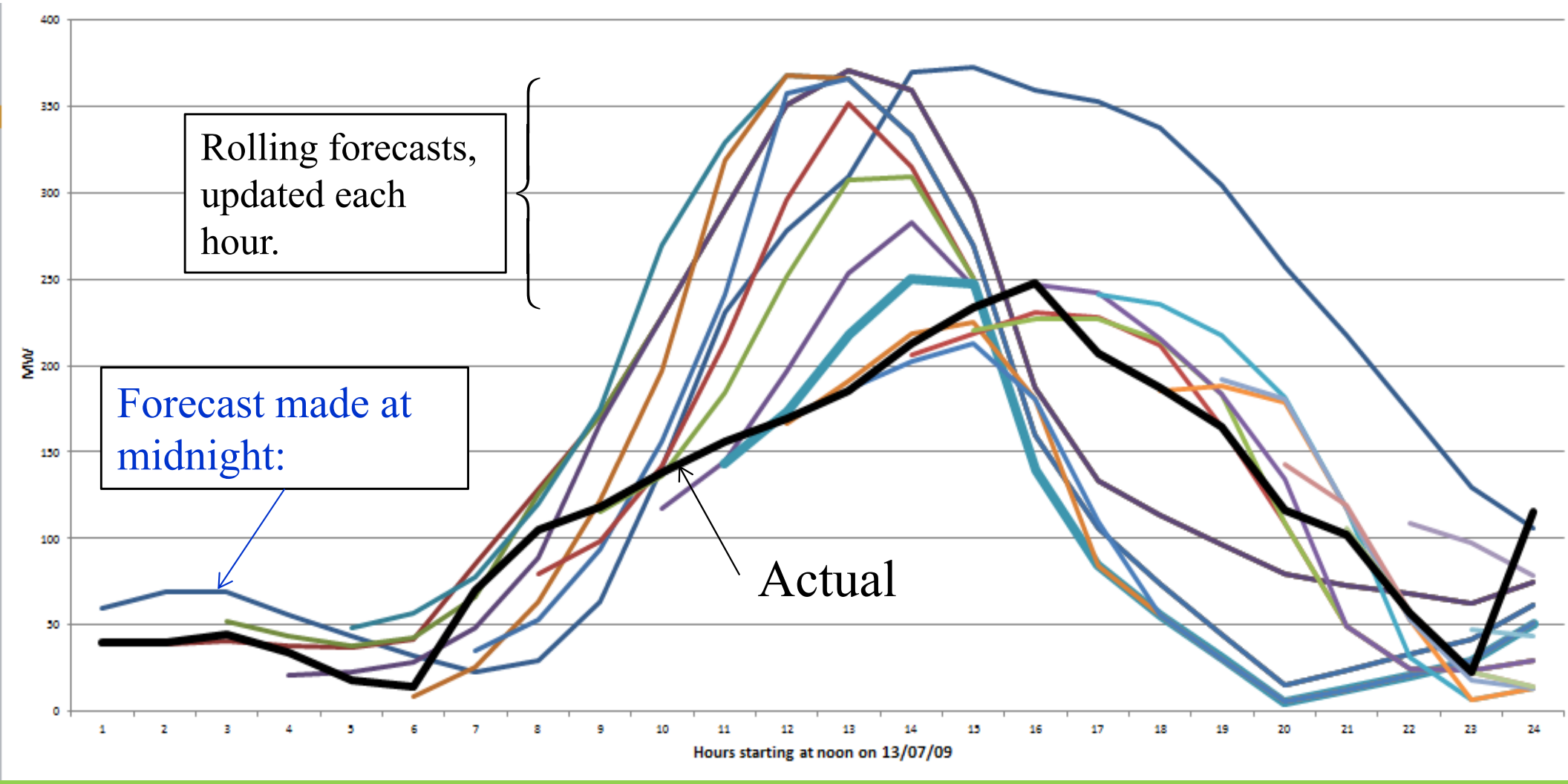
Downcrossing distribution



Modeling forecasting errors

Parametric cost function approximation

- Forecasts evolve over time as new information arrives:



The state variables

- Forecasting

  Imagine that we are forecasting demand using the model:

  $$f_{tt'} = \theta_{t0} + \theta_{t1}(t' - t)$$

  Our forecast of demand for time $t+1$ is

  $$f_{t,t+1} = \theta_{t0} + \theta_{t1}(t+1-t) = \theta_{t0} + \theta_{t1}$$

  Assume we observe $\hat{D}_{t+1}$ at time $t+1$, and we believe that

  $$\hat{D}_{t+1} = \hat{D}_t + \varepsilon_{t+1} \quad \text{where } \varepsilon_{t+1} \sim N(0, \sigma^2)$$

  We can update our parameters using

  $$\theta_{t+1,0} = (1-\alpha)\theta_{t0} + \alpha\hat{D}_{t+1}$$

  $$\theta_{t+1,1} = (1-\alpha)\theta_{t1} + \alpha\left(\hat{D}_{t+1} - \hat{D}_t\right)$$

  The state of our system would be

  $$S_t = \left(\hat{D}_t, \theta_{t0}, \theta_{t1}\right)$$

- Saeed's logic for modeling evolving forecasts

Week 8 - Monday

Two-agent newsvendor

Logistics!

Supply chain management

- Amazon-UPS meltdown, Christmas, 2013
  - » Amazon promised 2-day service (based on UPS service guarantees). UPS was overwhelmed.

NEW REPUBLIC

amazon.com

DECEMBER 26, 2013

If the Private Sector Is So Great, Why Did UPS Botch Christmas?

A corrective for market triumphalists

By Alec MacGillis @AlecMacGillis

Photo: Lionel Bonaventure/AFP/Getty

At the heart of the great big pile-on of ridicule for the flawed healthcare.gov rollout the past few months was a large helping of private-sector triumphalism. Just imagine, the chorus went, if tech giants like Amazon or Google had been in charge of the Web site instead of those clueless, fusty bureaucrats – first, the problems would not have

UPS Shipping Delays Show Perils of Stores Overpromising

Mary Schlangenstein, Leslie Patton and Alex Barinka
December 26, 2013 — 5:19 PM EST

(Corrects 22nd paragraph to show UPS did make most of its deliveries.)

Dec. 26 (Bloomberg) -- The failure of United Parcel Service Inc. and FedEx Corp. to deliver packages in time for Christmas has exposed the perils of retailers promising to get last-minute gifts to customers.

Chains from Kohl's Corp. to Amazon.com Inc. to 1-800-Flowers.com Inc. offered gift cards and refunds after angry shoppers took to social media to vent their frustrations at the missed shipments. On its website, UPS said the volume of last-minute air packages exceeded its capacity to process them.

Supply chain management

THE WALL STREET JOURNAL.

http://www.wsj.com/articles/wal-mart-reins-back-inventory-in-a-revamped-supply-chain-1439933834

BUSINESS | LOGISTICS REPORT

Wal-Mart Reins Back Inventory in a Revamped Supply Chain

The retailer is holding goods longer at distribution centers, increasing flexibility and trying to meet e-commerce competition and the changing consumer expectations

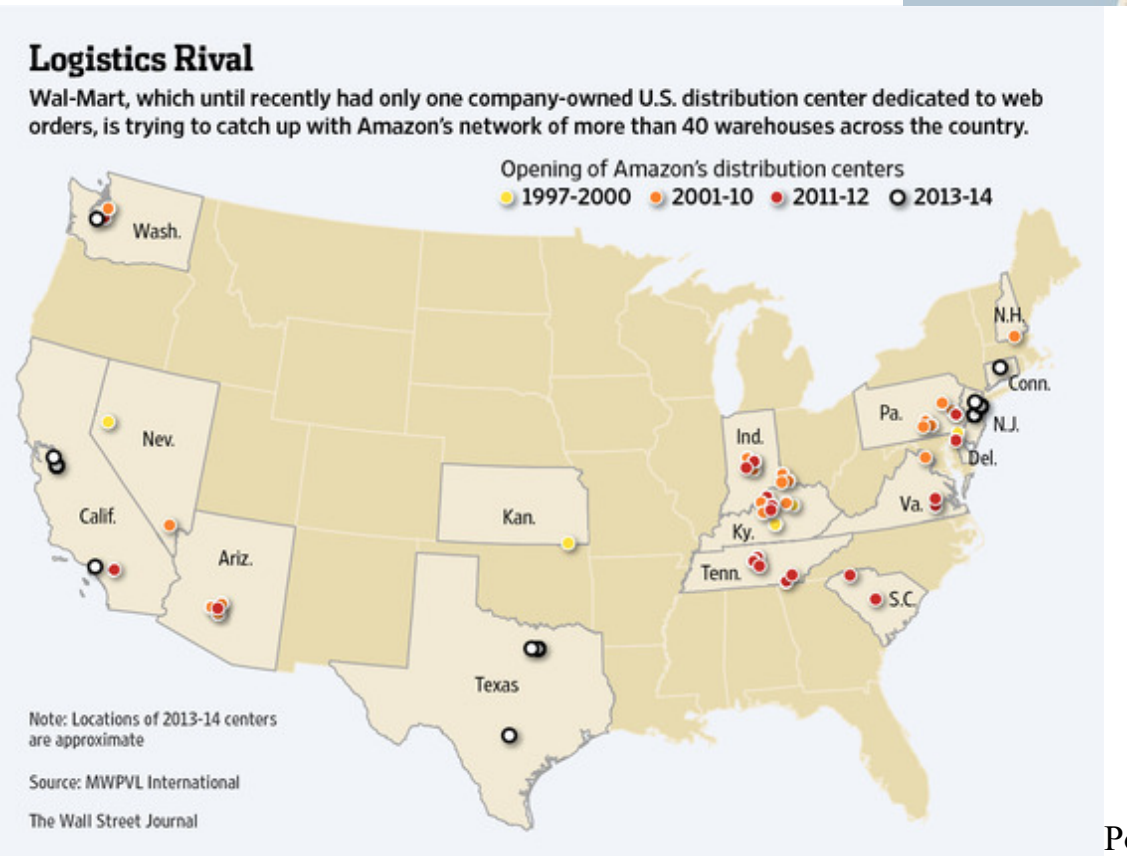
Why?

Wal-Mart Wal-Mart says part of its inventory management effort has been aimed at keeping workers on the sales floor rather than in the stock room. *PHOTO: BLOOMBERG NEWS*

By **LORETTA CHAO**

Aug. 18, 2015 5:37 p.m. ET

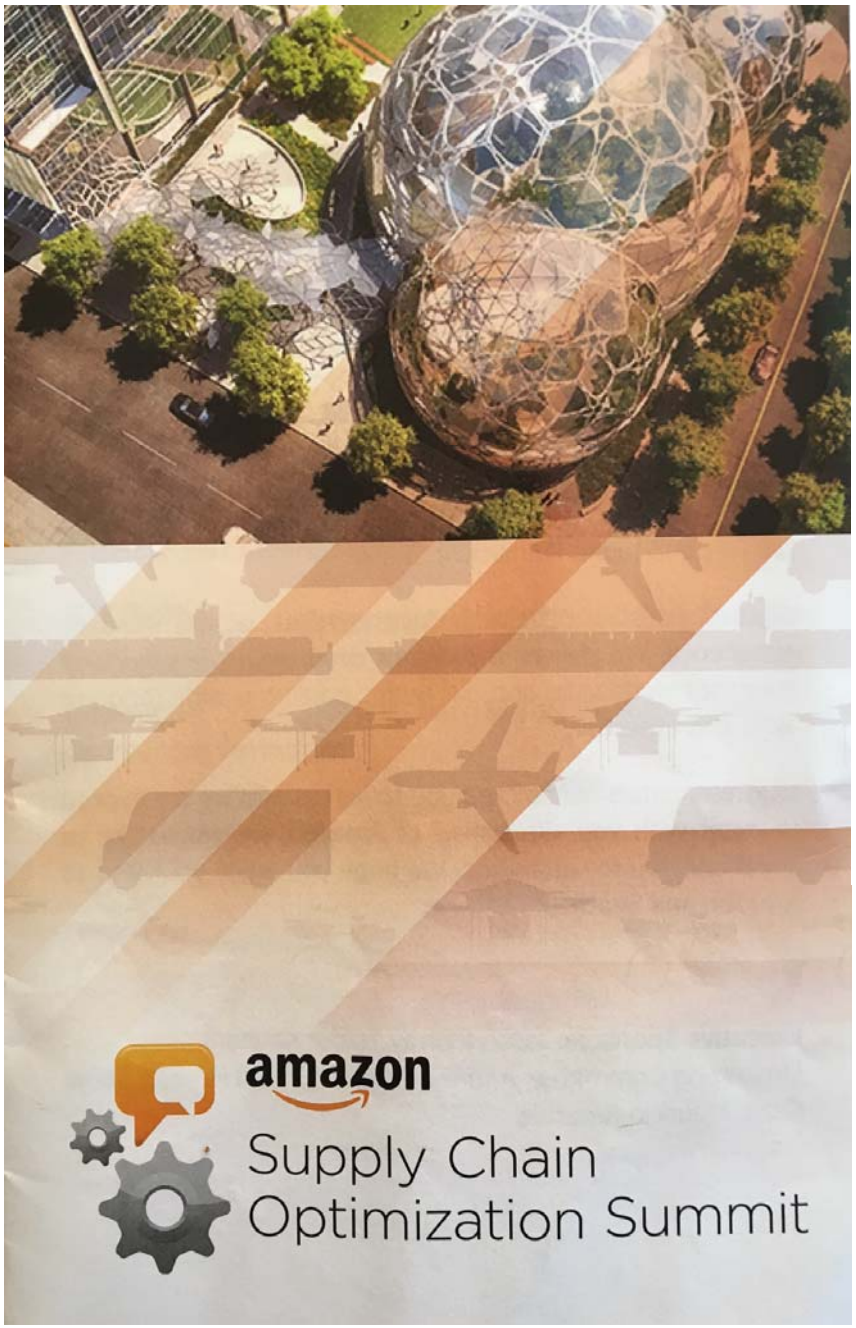
- Amazon distribution network
  - » Small number of DC's
  - » Uses UPS, FedEx, for final distribution



- *Walmart distribution network*
  - » *Extensive retail network*
  - » *Limited need for distribution centers*
  - » *Increasing role of internet sales*

Amazon

- First supply chain logistics conference in 2015
- Still running off of spreadsheets developed by Jeff Bezos
- Numerous instances where uncertainty is an important element.
- Exclusively using deterministic optimization tools at that time.



Global logistics

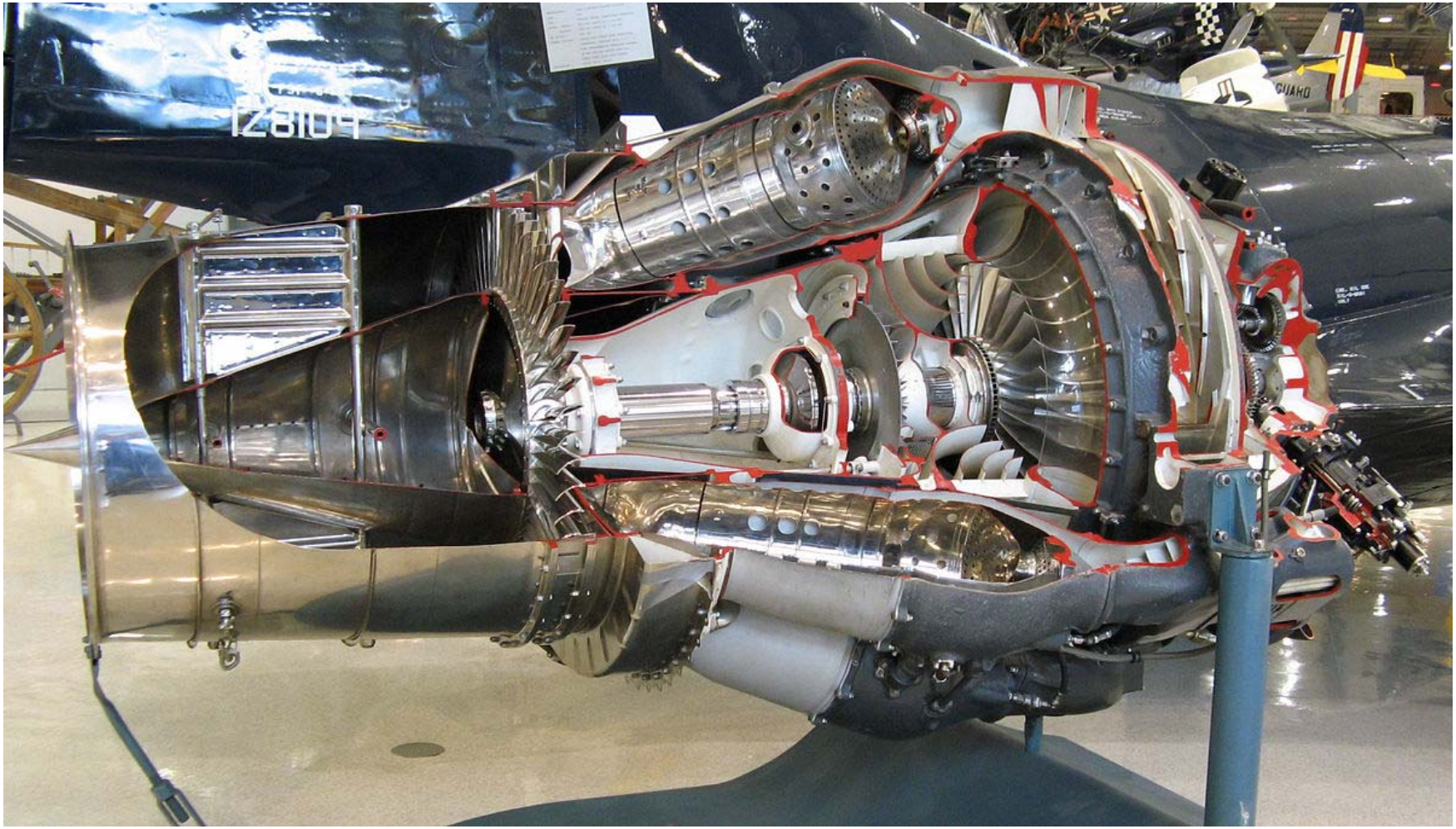
- Apple, Inc.

That's Logistics: Without Tim Cook, iPad would cost $5,000


PHOTO: CHRIS HONDROS/GETTY IMAGES

Before becoming the CEO of Apple last year, Tim Cook was known as the "logistics king" of the company. MacTrast.com pointed to an article on Business Insider that may demonstrate just how much: *"If it weren't for Tim Cook, the iPad would cost $5,000"*.

- Pratt&Whitney jet engines



- Pratt&Whitney jet engines

- Pratt&Whitney jet engines

Information exchange problems

Sample problems

» Pricing IPO's:

- Situation:
  - Bank wishes to bring an IPO to market.
- Decision:
  - How should the bank price the IPO?
- Behavior:
  - Bank solicits information from investment banks, who indicate a price lower than they are willing to pay.
- Outcome:
  - If the bank goes to market with too low of a price, it leaves money on the table.
  - If bank goes to market with too high of a price, it will not clear the market and is left with shares on hand.

Sample problems

- IT department:
  - Situation:
    - Request for programming
  - Decision:
    - IT department requests 3000 hours to complete a programming assignment.
  - Behavior:
    - Big penalty for not finishing on time, so estimate is inflated.
  - Outcome:
    - Project takes only 2000 hours.
      - If IT brings the project early, business unit may learn that IT inflates estimates by 50 percent.
      - IT pads project by checking, documentation, adding features, cleaning code, etc. etc.

Sample problems

» Spending a budget:

- Situation:
  - You have a $1.2 million budget for the year. After three months, you have spent $200,000.
- Decision:
  - You receive a request to spend $500,000.
- Behavior:
  - You have the money in the budget, but have to think about future expenses.
- Outcome:
  - You only allow $400,000 of the request. You did not satisfy the demand, but you have money left over.

Sample problems

- Military exercise:
  - Situation:
    - Civilian leadership initiates a military action.
  - Decision:
    - Armed forces evaluate situation and ask for resources:
      - Soldiers
      - Military equipment
      - Supporting people and equipment
      - Airlift capability
  - Outcome:
    - If too few resources are provided:
      - Conflict is prolonged
    - If too many resources are provided:
      - Resources are spread around activities.

write former Clinton national-security staffers Daniel Benjamin and Steven Simon in their book, "The Age of Sacred Terror," Clinton approached Joint Chiefs Chairman Gen. Hugh Shelton and said, "You know, it would scare the shit out of Al Qaeda if suddenly a bunch of black ninjas rappelled out of helicopters into the middle of their camp. It would get us enormous deterrence and show those guys we're not afraid." Shelton, "a huge, powerfully built man, blanched," write the authors. Nothing came of Clinton's somewhat whimsical suggestion.

The military has a way of testing the seriousness of the civilian leadership. Asked to do something difficult and dangerous, like putting combat troops into a far-off country like Afghanistan, the top brass will make impossible manpower and logistical requirements: whole divisions, massive airlift and backup, everything including "a bowling alley and a PX," says one White House cynic. After 9/11, Rumsfeld says, he was "impatient" to get troops into Afghanistan. Bush administration higher-ups made it clear they wanted bin Laden, as the president put it, "dead or alive." But that doesn't mean the bureaucracy smartly saluted and set about trying to kill the Qaeda leadership.

Indeed, within hours of Bush's statement, lawyers at the National Security Council, the State Department and the Pentagon launched a flurry of e-mails and calls warning that Bush's macho rhetoric could be viewed as a violation of the Geneva Conventions. The notion that Bush could be prosecuted as a war criminal was rejected by their bosses as absurd. Still, while a top administration official told NEWSWEEK that there was very little philosophical discussion at the top about "targeted killings" of terrorist leaders, there continued to be legal qualms—and a reluctance to act—down in the ranks.

The CIA and the Air Force had recently developed the perfect execution machine, the Predator, a remote-control unmanned vehicle able to loiter over a target and launch Hellfire missiles with deadly accuracy. On one of the first nights of the Afghanistan conflict, a Predator spotted a convoy believed to be carrying Taliban leader Mullah Omar. The passengers got out and entered a building. The CIA was almost—but not entirely—sure that Mullah Omar was inside. Should the Predator take a shot? At CENTCOM headquarters, General Franks's top military lawyer, a female Navy captain, posed tough questions. What if innocent civilians were killed? And there was a mosque right next door. What if the mosque were damaged?

The strike was aborted; Mullah Omar got away (and is still at large somewhere in Afghanistan). After Rumsfeld and others expressed their dissatisfaction, the rules of engagement were tweaked, and the Predator was used to kill a senior bin Laden lieutenant near Kabul, among others. But once the Afghanistan conflict was over, the debate resumed. Killing a "leadership target" in wartime is not assassination. But is the war on terror open-ended? Under the laws of war, can nations strike pre-emptively at an "imminent" threat? Just how imminent does the threat have to be?

Since Afghanistan, a senior intelligence official says, the Predator has been used only once to eliminate a Qaeda leader, blowing up a car containing Abu Ali—a former bin Laden security guard suspected of plotting the attack against the USS Cole—in the Yemeni desert last winter. Even that attack made the lawyers nervous: one of the passengers in the car was an American citizen. The man was a suspected terrorist; even so, the lawyers asked, was it lawful to execute him without a trial? If so, did that mean

MIDEAST

Abbas resigns in frustration, leaving Bush's dreams of reform in tatters

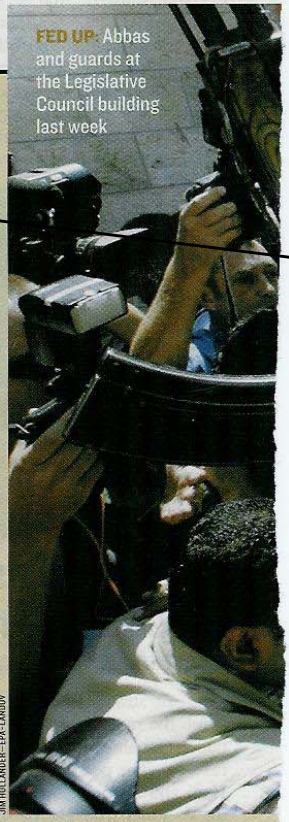
Running Off the Roadmap

BY JOSHUA HAMMER

Last Thursday morning, Palestinian Prime Minister Mahmoud Abbas called a meeting of the Legislative Council in Ramallah to report on the achievements of his government after 100 days in office. As the prime minister got up to speak, a gang loyal to Yasir Arafat, including several men wearing black masks, burst into the room. The intruders called Abbas a collaborator, smashed windows, then spray-painted his car parked out front with the word TRAITOR. For Abbas, it was the worst in a series of humiliations that had made his job increasingly intolerable. Two days later, in a closed-door session, the 68-year-old Abbas informed the council that he was stepping down. Arafat's interference, he said, had simply become too much to bear. "This experience really scarred him," says Qadoura Faris, a council member from Ramallah.

Abbas might as well have ripped up the U.S.-backed Roadmap, the three-phase peace plan that was supposed to lead to the creation of a Palestinian state in two years. Although Abbas could return to form a new government at Arafat's behest, Palestinian officials close to him say the chances are very slim. His resignation leaves the Bush administration groping to put its peace policy back on track, even as it contends with an Israeli government determined to intensify "all-out war" against militant groups. Just four hours after Abbas stepped down, an Israeli F-16 fired three 500-pound missiles at an office building in Gaza City, where Hamas spiritual leader Sheik Ahmed Yassin was meeting militia commanders. The raid lightly wounded the quadriplegic cleric in the hand and injured 15 others. A senior Israeli military official said the timing of the attack was coincidental and called the meeting of the Hamas leadership "a rare operational opportunity."

Only a short while ago, the Bush administration was betting that war in Iraq would change the whole Mideast calculus for the better—that "the road to Jerusalem ran through Baghdad." Even more recently, State Department



FED UP: Abbas and guards at the Legislative Council building last week

JIM HOLLANDER—EPA-LANDOV

The military has a way of testing the seriousness of the civilian leadership. Asked to do something difficult and dangerous, like putting combat troops into a far-off country like Afghanistan, the top brass will make impossible manpower and logistical requirements: whole divisions, massive airlift and backup, everything including "a bowling alley and a PX," says one White House cynic.

Sample problems

- IMF makes loans to countries:
  - Situation:
    - A country has a financial crisis and requests a loan from the IMF.
  - Decision:
    - IMF evaluates the need and (potentially) makes a loan.
  - Behavior:
    - There is an international incentive to make sure that all economies get past problem periods.
  - Outcome:
    - IMF loans too much:
      - Country becomes dependent on external funding and/or money may be used for other purposes.
    - IMF loans too little:
      - Country remains in crisis and/or has to ask for more.

Narrative

11.1 NARRATIVE

Imagine that a logistics manager for Amazon has to provide trailers to move freight out of Chicago on a weekly basis. The logistics manager, working in the field, has access to information that allows him to estimate how many trailers he will need that week, but the actual might be higher or lower. The field manager then makes a request to a central manager for trailers, who then makes a judgment on her own how many trailers to provide, and makes the final decision on the number of trailers that will be provided.

The two managers both work for the same company, but the field agent is far more worried about running out, since he has to do short-term rentals if he runs out of trailers. The central manager, on the other hand, also does not want the field agent to run out, but she also does not want him to have excess trailers, since she has to pay for those trailers.

We assume the process unfolds as follows:

Step 1: The field agent observes the initial estimate of how many trailers are needed. This information is private to the field agent.

Step 2: The field agent then requests trailers from the central agent.

Step 3: The central agent then decides how many trailers to give to agent q.

Step 4: The field agent receives the number of trailers granted by the central agent, and then observes the actual need for trailers.

Step 5: The field and central agents compute their own costs.

The tension in this problem arises first because the initial estimate of trailers needed is just an estimate, which we may assumed is unbiased (that is, it is true on average). The problem is that the field agent has a large cost if he runs out, so his strategy is to overestimate his needs (recall the newsvendor problem in chapter 3). The central agent, on the other hand, probably has balanced costs for being over and under, and does not want to order too many or too few.

What complicates the problem is the initial estimate given to the field agent. While not perfect, it has value information since it will indicate if a day is going to have high or low demand. This means that the central agent has to pay attention to the request made by the field agent, while recognizing that the field agent will make request that are biased upward. Knowing this, the central agent would tend to use the field agent's request as a starting point, but then reduce this for the final allocation. Not surprisingly, the field agent knows that the central agent will do this, and compensates accordingly.

Two-agent resource allocation

- Original newsvendor problem:
  - » Single “agent” requests resources, and gets whatever is requested.
  - » Has to live with the outcome.

- Two-agent newsvendor problem:
  - » “Field agent” requests resources from a “central command”.
  - » Request represents information to the central command. Central command may decide to a different amount from what was requested.

Two-agent resource allocation

- Problem characteristics:
  - » Central command's cost of underage is typically lower than that for the field.
  - » Field agent has better (potentially perfect) information about actual requirements.
  - » After the event, the field agent may know what was really required.
  - » What does the central command learn?
    - Case A: Exactly what was required.
    - Case B: An unbiased but noisy estimate of what was required.
    - Case C: A biased (and possibly noisy) estimate of what was required.

Two-agent resource allocation

- Rules
  - » Field agent gets to see information that provides an estimate of what is required.
  - » Field agent requests resources from central command.
  - » Central command thinks about it and decides how much to give the field.
  - » Central command and field get to see what was really required.

Two-agent newsvendor game

Distribute sheets

Two-agent resource allocation

- Problem characteristics:
  - » Central command's cost of underage is typically lower than that for the field.
  - » Field agent has better (potentially perfect) information about actual requirements.
  - » After the event, the field agent may know what was really required.
  - » What does the central command learn?
    - Case A: Exactly what was required.
    - Case B: An unbiased but noisy estimate of what was required.
    - Case C: A biased (and possibly noisy) estimate of what was required.

Two-agent resource allocation

- Rules
  - » Field agent gets to see information that provides an estimate of what is required.
  - » Field agent requests resources from central command.
  - » Central command thinks about it and decides how much to give the field.
  - » Central command and field get to see what was really required.

Two-agent resource allocation

- The two-agent resource game:

	Field		Central	Actual	Costs	
	Initial	Request	Give to	exogenous	Central	Field
	estimate	from central	Field	demand	c^o = 5 c^u=5	c^o=2 c^u=10
1	14					
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						

The resource allocation game

- Now you do it:
  - » Divide into pairs. Each pair will play the game twice.
  - » For the first round, one takes the role of central command and the second is the field. The field takes the sheet marked "field – round 1."
  - » The field looks at the "estimate" (which is confidential) and makes a request to the central command. The central command then makes an allocation. The central command faces equal costs for underage and overage (5). The field faces a cost of 10 for being under vs. 2 for being over.
  - » After making decisions, wait for instructor to reveal the actual.
  - » Record decisions, actual, and overage/underage.
  - » Maintain a running total of costs.
  - » For round 2, switch roles, and use "field/central – round 2" sheet.

Basic model

- Notes:
  - » Set up board with five dimensions of a sequential decision problem.
  - » Present basic multi-agent notation:
    - q for field agent, q' for central.
    - qq' means information owned (or initiated) by q sent to q'.
    - $\delta_{tq}^{central}$ would refer to an estimate owned by agent q (field) about the central agent. Note that q is an index (there may be many of these agents), while "central" is a label (only applies to two-agent newsvendor problems).

State variables

The initial information available to the field agent is the estimate of the number of trailers that will be needed, which we represent using

$$R_{tq}^{est} = \text{The initial estimate of how many trailers are needed.}$$

This initial estimate may be biased, so we are going to let our initial estimate of the bias be

$$\bar{\delta}_{tq}^{est} = \text{Initial estimate of the difference between } R_{tq}^{est} \text{ and the true demand.}$$

We will also have to estimate how much the central agent reduces the request of the field agent, which we represent using

$$\bar{\delta}_{tq}^{central} = \text{Estimate of how much the central agent will reduce the request of the field agent, which we define (below) as } x_{tqq'}.$$

Similarly, the central agent will learn the difference between the request made by the field agent, and what the field agent eventually needs, which we represent by

$$\bar{\delta}_{tq'}^{field} = \text{Estimate of the difference between what the field agent requests } (x_{tqq'}), \text{ and what the field eventually needs.}$$

The state variable for each agent is the information they have before they make a decision. For the field agent, the state variable is

$$S_{tq} = (R_{tq}^{est}, \bar{\delta}_{tq}^{est}, \bar{\delta}_{tq}^{central}).$$

The state variable for the central agent is

$$S_{tq'} = (x_{tqq'}, \bar{\delta}_{tq'}^{field}).$$

Decision variables

The decisions for each agent are given by

$$\begin{aligned} x_{tqq'} &= \text{The number of trailers that agent } q \text{ asks for from agent } q', \\ x_{tq'q} &= \text{The number of trailers that agent } q' \text{ gives to agent } q, \text{ which is what is implemented in the field.} \end{aligned}$$

Exogenous information

The exogenous information for the field agent can be thought of as the initial estimate of the trailers needed (although we put that in the state variable):

$$R_{tq}^{est} = \text{The initial estimate of how many trailers are needed.}$$

After making the decision $x_{tqq'}$, we then receive two types of information: what the central agent grants us, and then the actual required demand:

$$\begin{aligned} x_{tq'q} &= \text{The decision made by the central agent in response to the request of the field agent,} \\ \hat{R}_{t+1} &= \text{The actual number of trailers that field agent } q \text{ ends up needing (this information is available to the central agent as well).} \end{aligned}$$

The exogenous information for agent q is then

$$W_{t+1,q} = (x_{tq'q}, \hat{R}_{t+1}).$$

We note in passing that while this information is indexed at time $t+1$, the request from the field, $x_{tqq'}$, is indexed by t (since it depends on information available up through time t).

The central agent receives the initial request $x_{tqq'}$ which arrives as exogenous information, but because this is received before she makes her decision, then enters through the state variable for the central agent. The only exogenous information for the central agent is the final demand, so

$$W_{t+1,q'} = (\hat{R}_{t+1}).$$

Transition function

For the field manager, there are three state variables: R_{tq}^{est}, the bias $\bar{\delta}_{tq}^{est}$ between the estimate R_{tq}^{est} and the actual $\hat{R}_t$, and the bias $\bar{\delta}_{tq}^{central}$ introduced by the central manager when the field makes a request. The first state variable, R_{tq}^{est}, arrives directly as exogenous information. The biases $\bar{\delta}_{tq}^{est}$ and $\bar{\delta}_{t,q}^{central}$ are updated using

$$\begin{aligned}
\bar{\delta}_{t+1,q}^{est} &= (1-\alpha)\bar{\delta}_{tq}^{est} + \alpha(\hat{R}_t - R_t^{est}) \\
\bar{\delta}_{t+1,q}^{central} &= (1-\alpha)\bar{\delta}_{tq}^{central} + \alpha(x_{tqq'} - x_{tq'q}).
\end{aligned}$$

The transition function for the central manager is similar. Again, the decision of the field manager, $x_{tqq'}$ arrives to the state variable exogenously. Then, we update the bias that the central manager estimates in the request of the field manager using

$$\bar{\delta}_{t+1,q'}^{field} = (1-\alpha)\bar{\delta}_{t+1,q'}^{field} + \alpha(x_{tqq'} - \hat{R}_t).$$

Objective function

We begin by defining:

$$
\begin{aligned}
c_q^o &= \text{The unit cost incurred by the field agent for each excess trailer (what the field pays per day for each trailer), also known as the overage cost,} \\
c_q^u &= \text{The unit cost incurred by the field agent for each trailer that has to be rented to make up for lack of capacity, also known as the underage cost,} \\
c_{q'}^o, c_{q'}^u &= \text{The cost of overage and underage for the central agent.}
\end{aligned}
$$

The costs for each agent are given by

$$
\begin{aligned}
C_{tq}(S_{tq}, x_{tq'q}) &= \text{Overage/underage costs for agent } q, \\
&= c_q^o \max\{x_{tq'q} - \hat{R}_t, 0\} + c_{q'}^o \max\{\hat{R}_t - x_{tq'q}, 0\}, \\
C_{tq'}(S_{tq'}, x_{tq'q}) &= \text{Overage/underage costs for agent } q', \\
&= c_{q'}^o \max\{x_{tq'q} - \hat{R}_t, 0\} + c_q^o \max\{\hat{R}_t - x_{tq'q}, 0\}.
\end{aligned}
$$

The performance of both the field and central agents depend on the number of trailers $x_{tq'q}$ that the central agent gives to the field. This decision, however, depends on the decision made by the field agent.

Designing policies

11.4.1 Field manager

The field manager starts with an estimate R_t^{est}, but has to account for three factors:

1) The estimate R_t^{est} may have a bias β^{est} (we cannot be sure about the source of the estimate R_t^{est}). The bias is given by

$$\beta^{est} = \mathbb{E}\hat{R}_t - R_t^{est}.$$

So, if $\beta^{est} > 0$ then this means that R_t^{est} is upwardly biased.

2) The true number of trailers needed, $\hat{R}_t$ is random even once you have factored in the bias. The field manager has a higher cost of having too few trailers than too many, so he will want to introduce an upward bias to reflect the higher cost of being caught short.

3) The central manager has a balanced attitude toward having too many or too few, and knows about the bias of the field manager. As a result, the central manager will typically use the request of the field manager, $x_{tqq'}$, just as the field manager may be adjusting for a possible bias between the estimate R^{est} and the actual $\hat{R}_t$. The field manager knows that the central manager will be making this adjustment, and as a result has to try to estimate it, and counteract it. Since the field manager knows both his request $x_{tqq'}$ and then sees what the central manager provides, the time t observation of the bias is given by

$$\beta_{tq}^{central} = x_{tq'q} - x_{tqq'}.$$

We need to use our estimates of the differences between R_t^{est} and $\hat{R}_t$, the difference between $x_{tqq'}$ and $x_{tq'q}$, and the difference between $x_{tqq'}$ and $\hat{R}_t$. We propose a policy for the field manager given by

$$X_{tqq'}^{field}(S_t|\theta_q^{field}) = R_t^{est} - \delta_{t-1,q}^{est} - \delta_{t-1,q}^{central} + \theta_q^{field}. \tag{11.3}$$

This policy starts with the initial estimate R_t^{est}, corrects for the bias in this initial estimate using $\delta_{t-1,q}^{est}$, then corrects for the bias from the central manager $\delta_{t-1,q}^{central}$, and then finally introduces a shift that can capture the different costs of over and under for the field manager. The parameter θ^{field} has to be tuned.

11.4.2 Central manager

Our policy for the central manager is given by

$$X^{central}_{tq'q}(S_t|\theta^{central}_{q'}) = x_{tqq'} - \delta^{field}_{t-1,q'} + \delta^{central}_{q'}. \quad (11.4)$$

Here, we start with the request made by the field manager, subtract our best estimate of the difference between the field manager's request and what was eventually needed, $\delta^{field}_{tq'}$, and then add in $\theta^{central}_{q'}$ which is a tunable parameter for the central manager.

Extensions

- Extensions:
  - » Introduce estimate of standard deviation in the policy (requires changes to state variable and transition function).
  - » Introduce estimate made by field (central) about what central (field) thinks that field (central) will do.
  - » What if the central agent is not allowed to see $\hat{R}_t$?

Week 8 - Wednesday

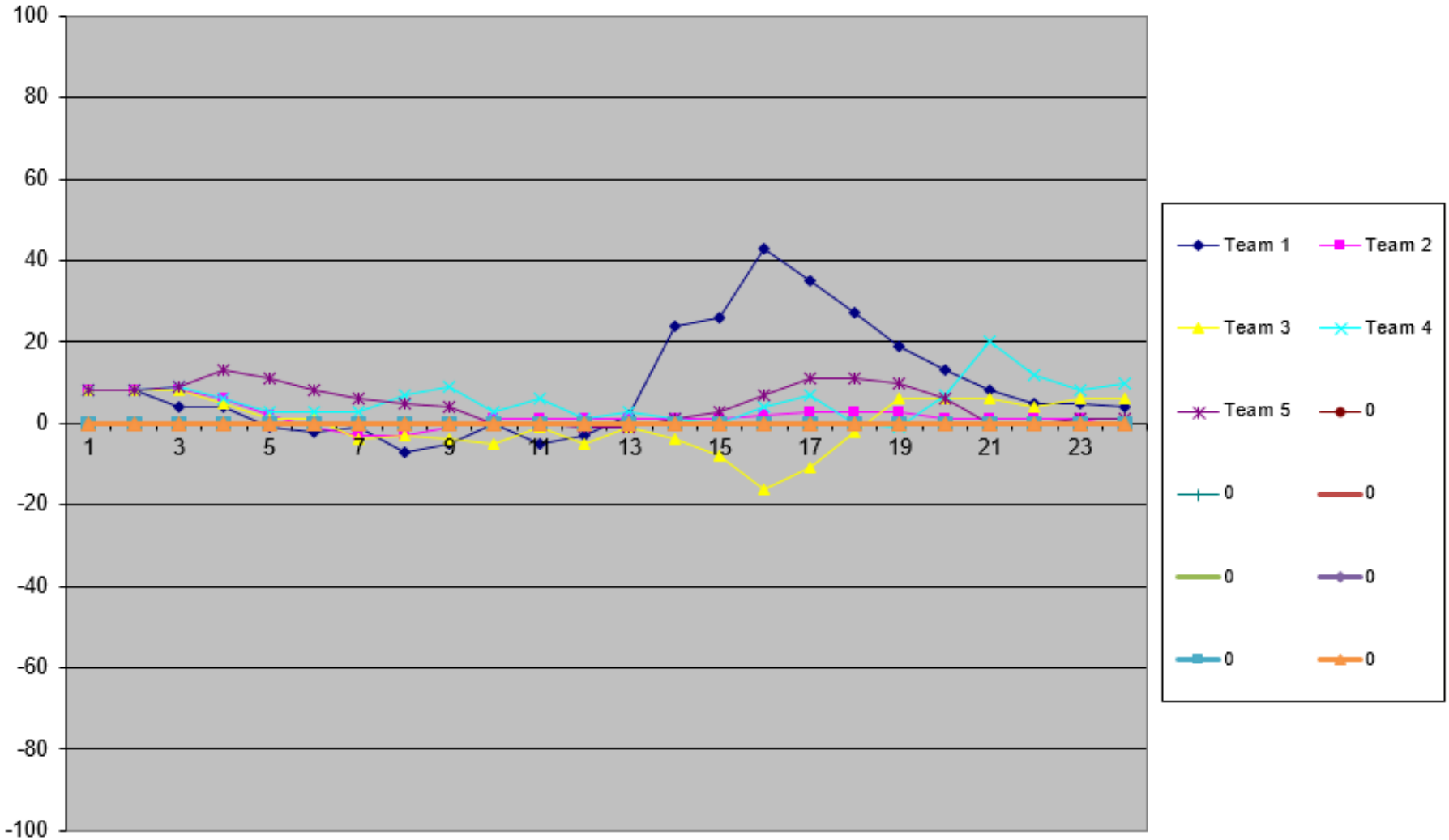
Beer game!

Week 9 - Monday

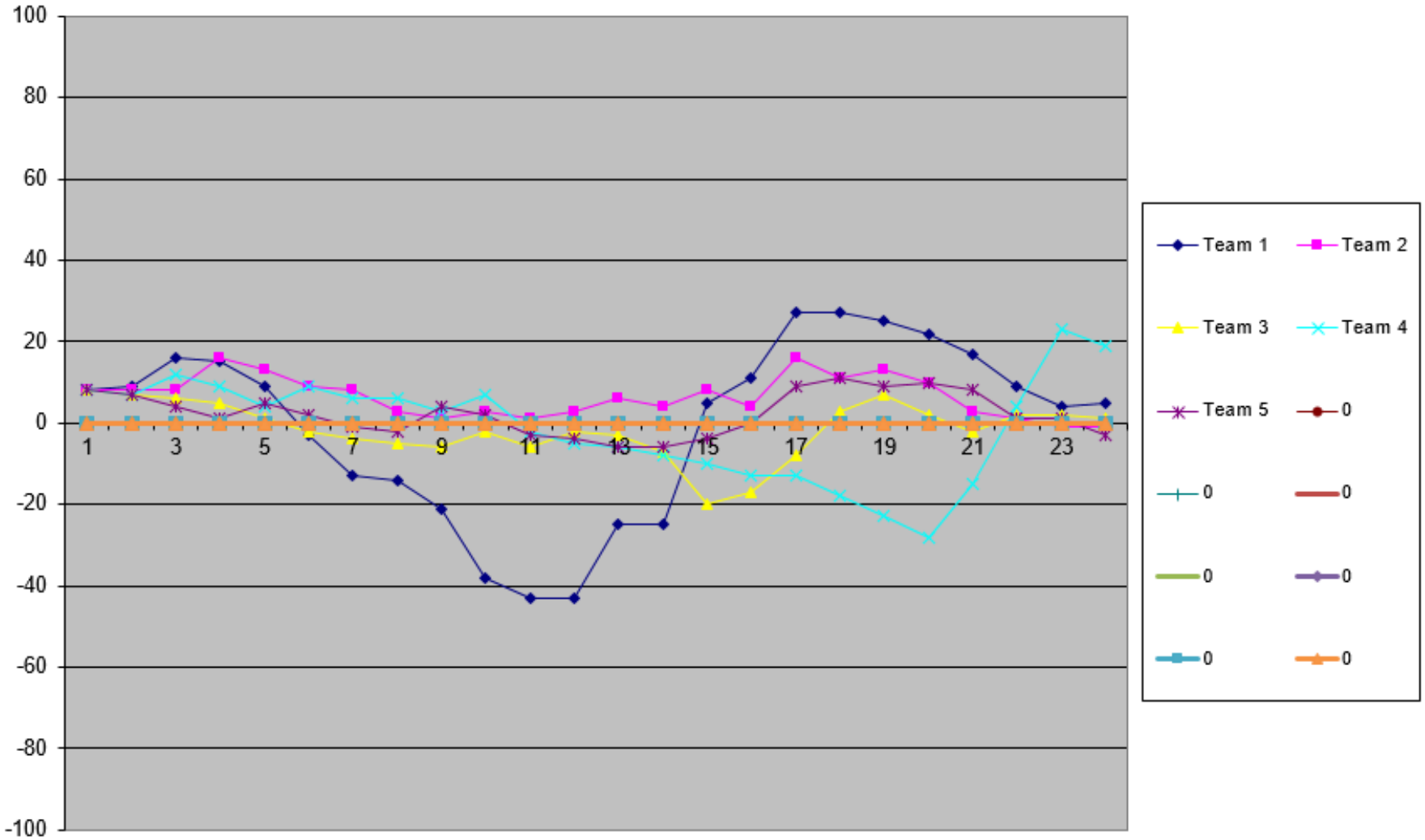
Beer game discussion

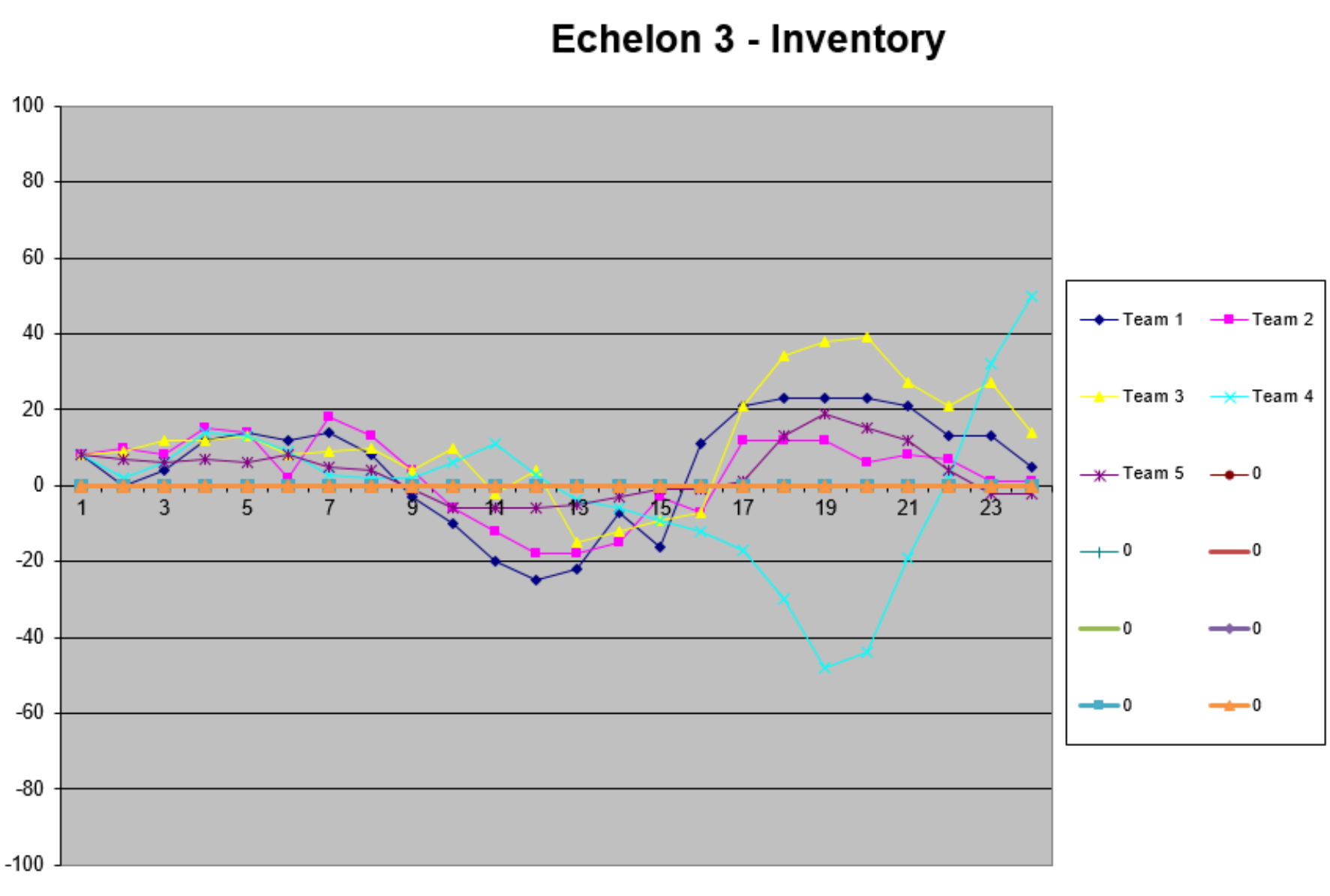
Results of the beer game

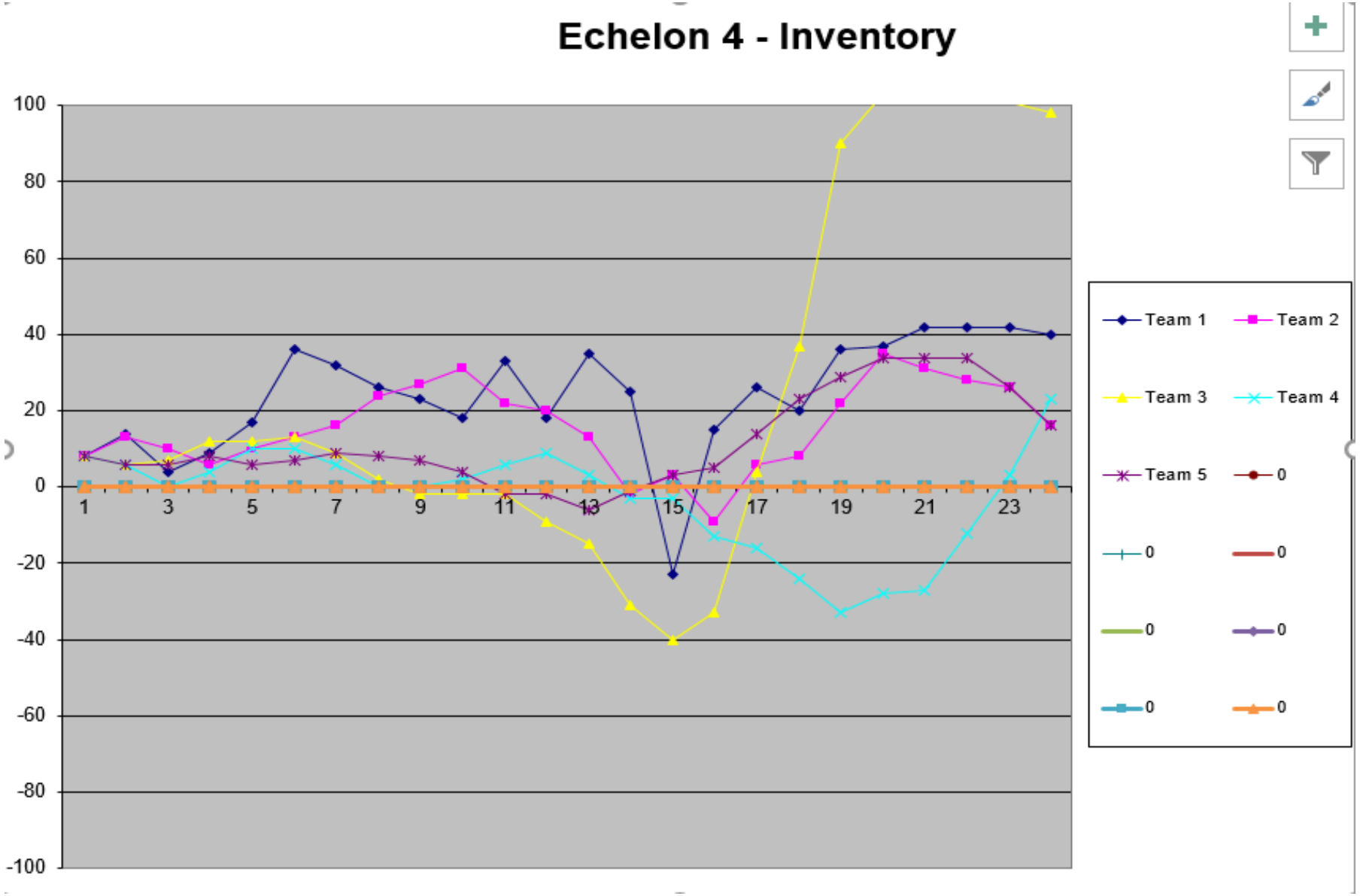
Echelon 1 - Inventory



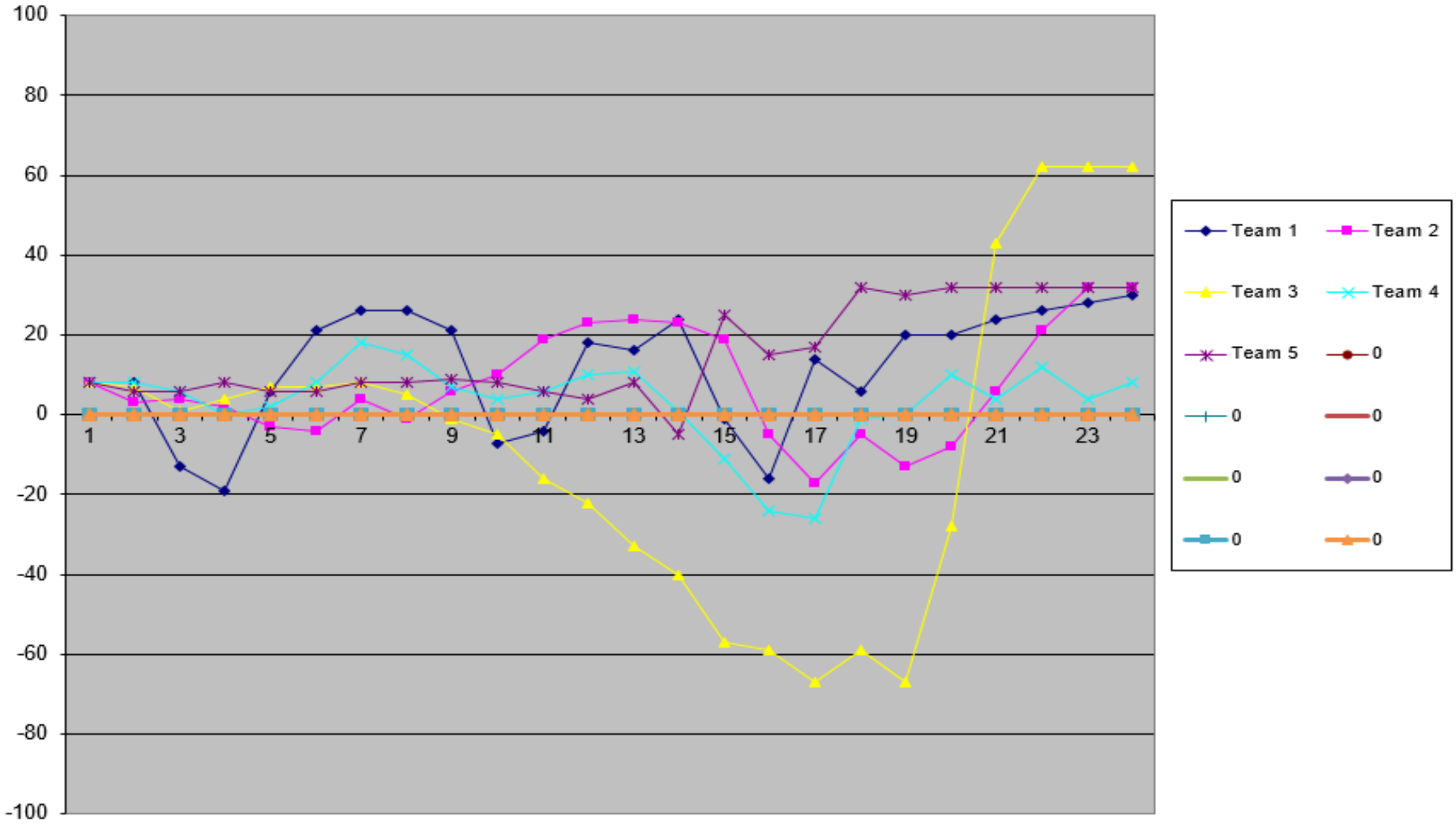
Echelon 2 - Inventory

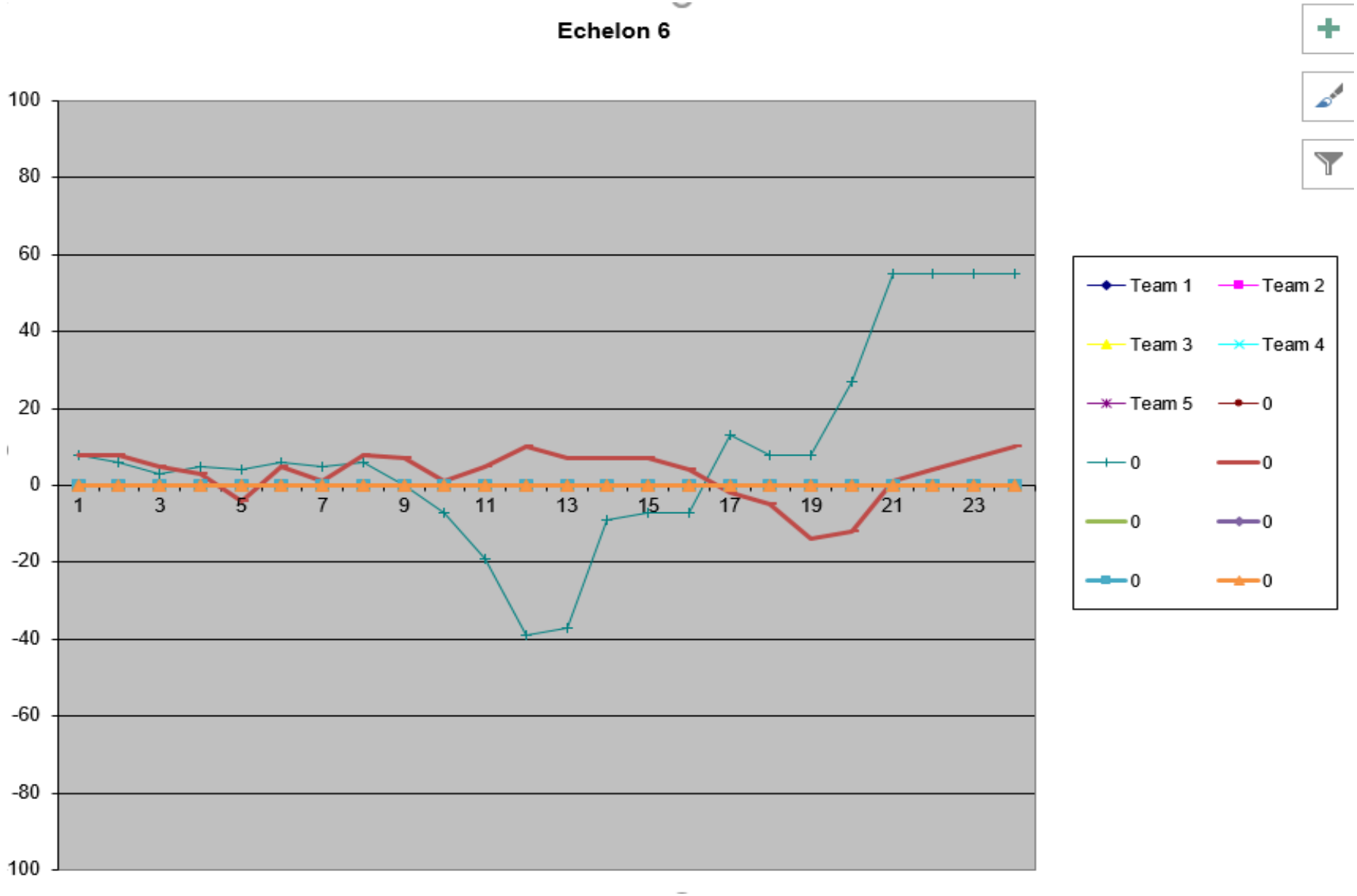


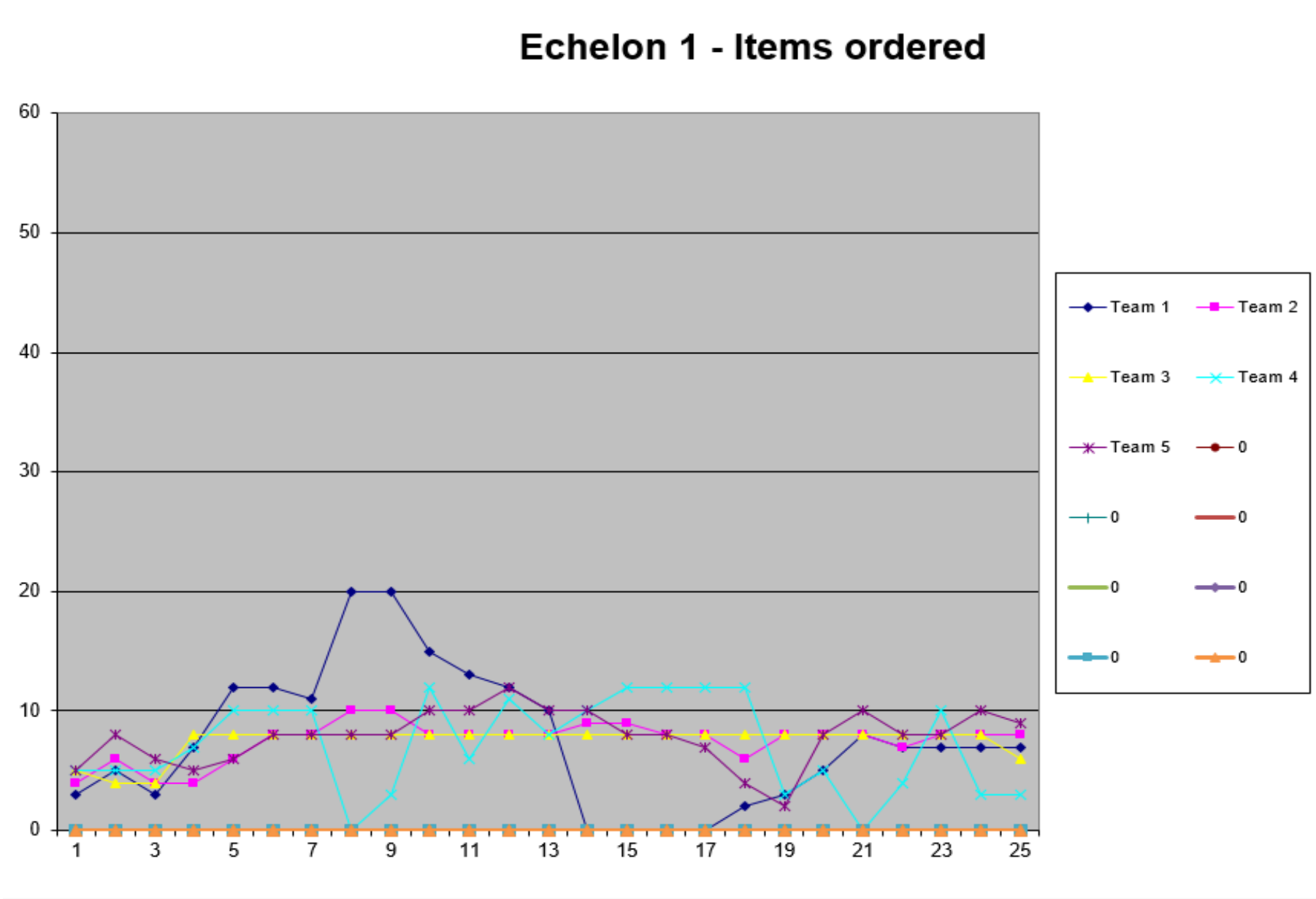


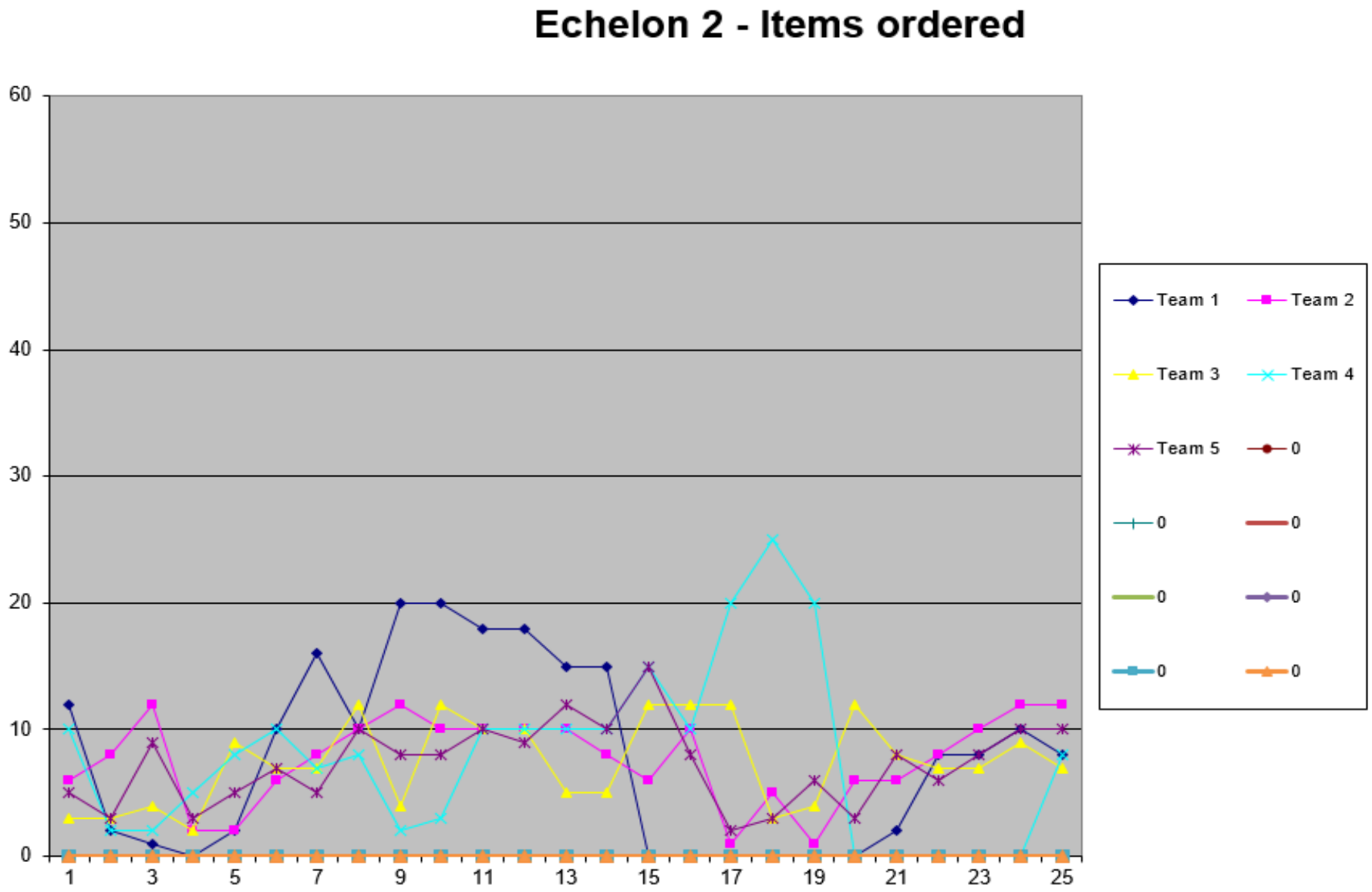


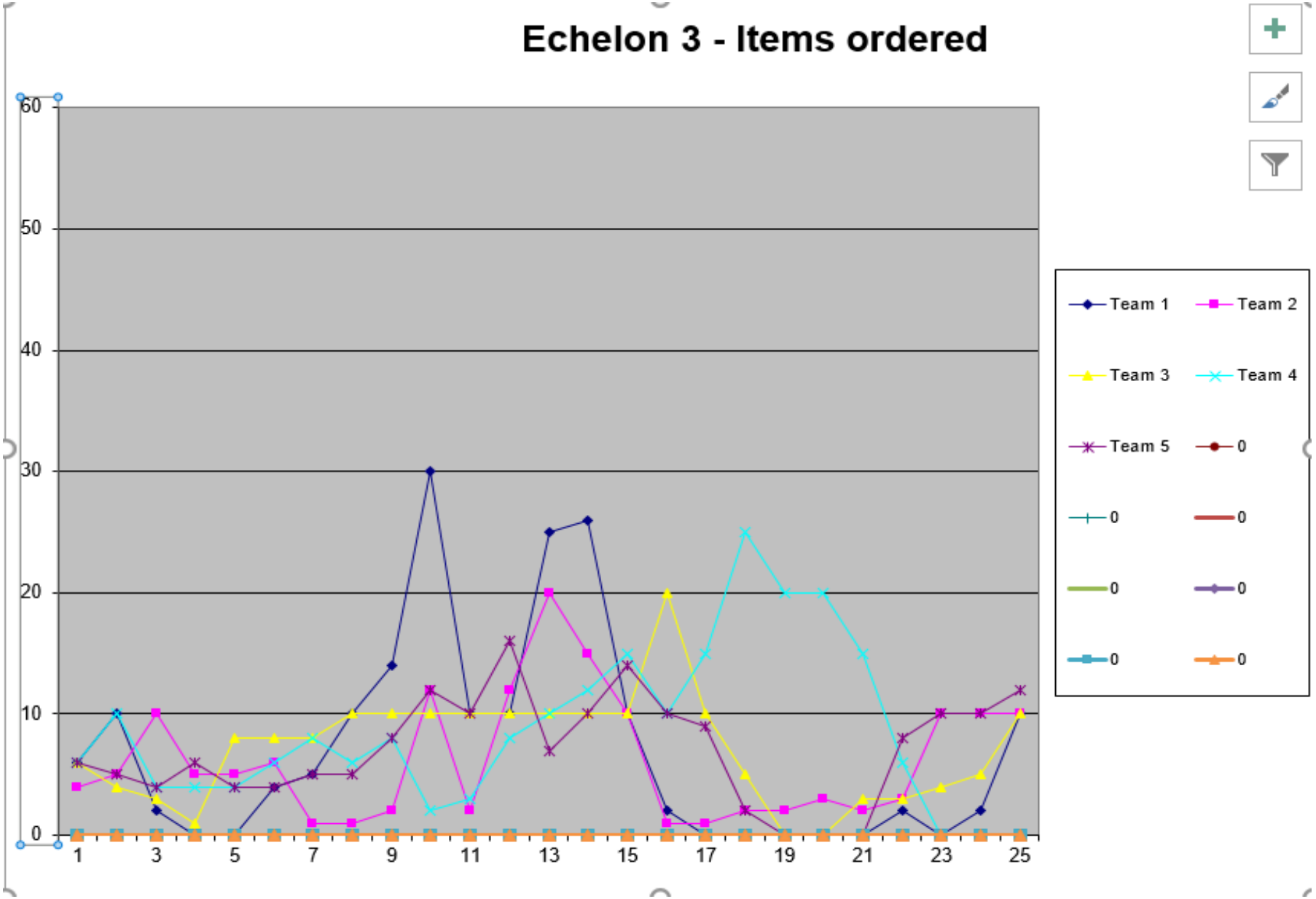
Echelon 5

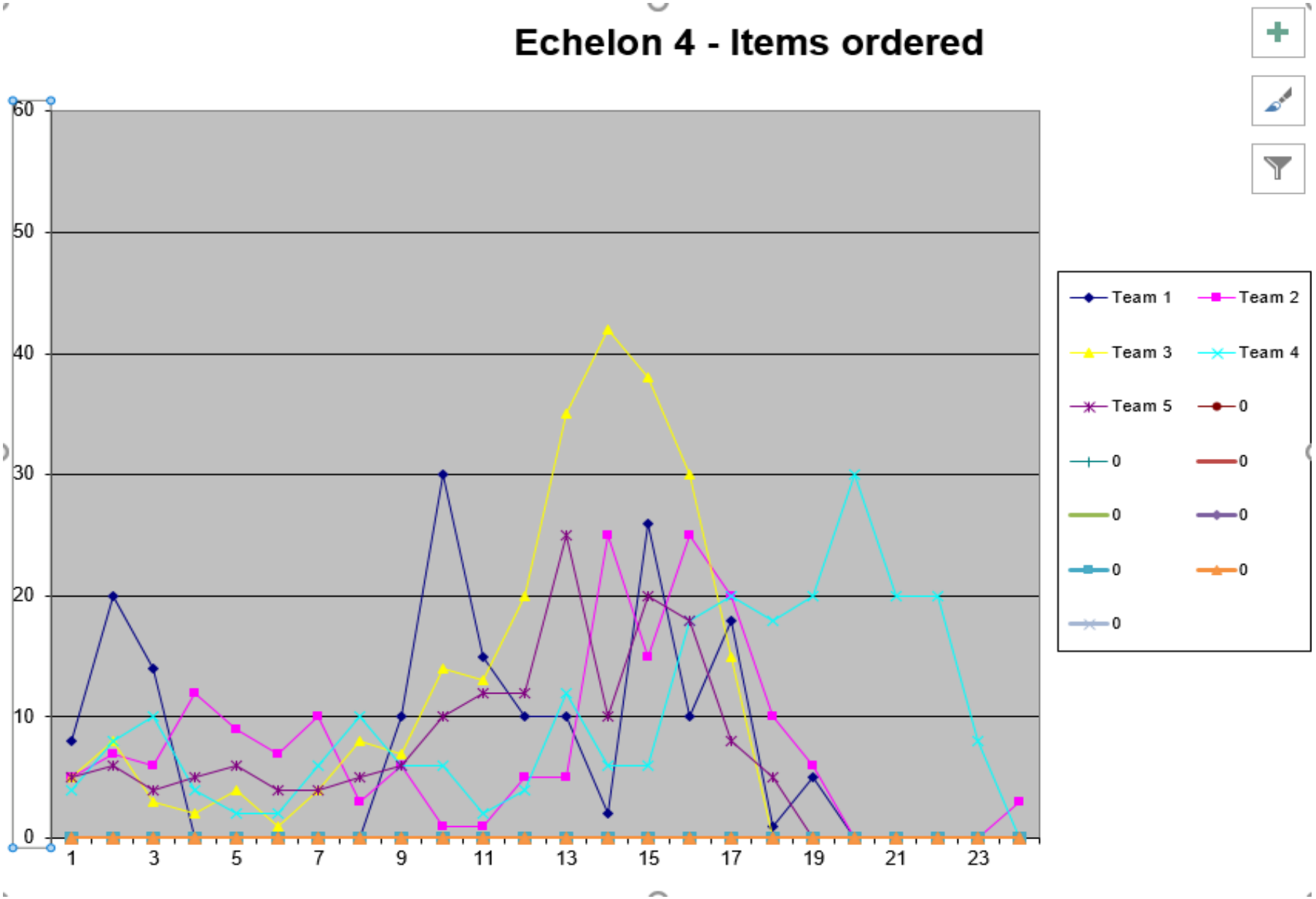


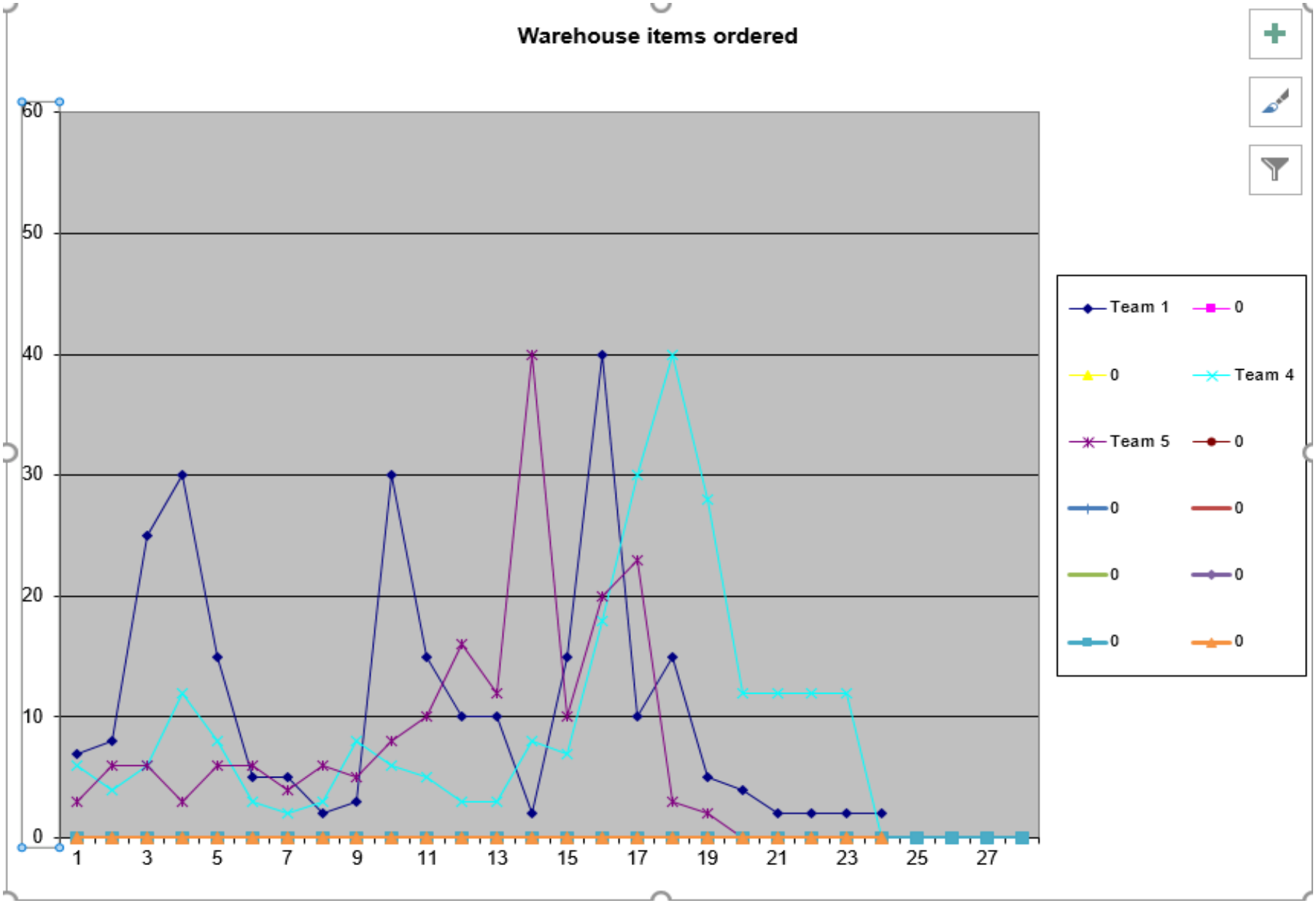


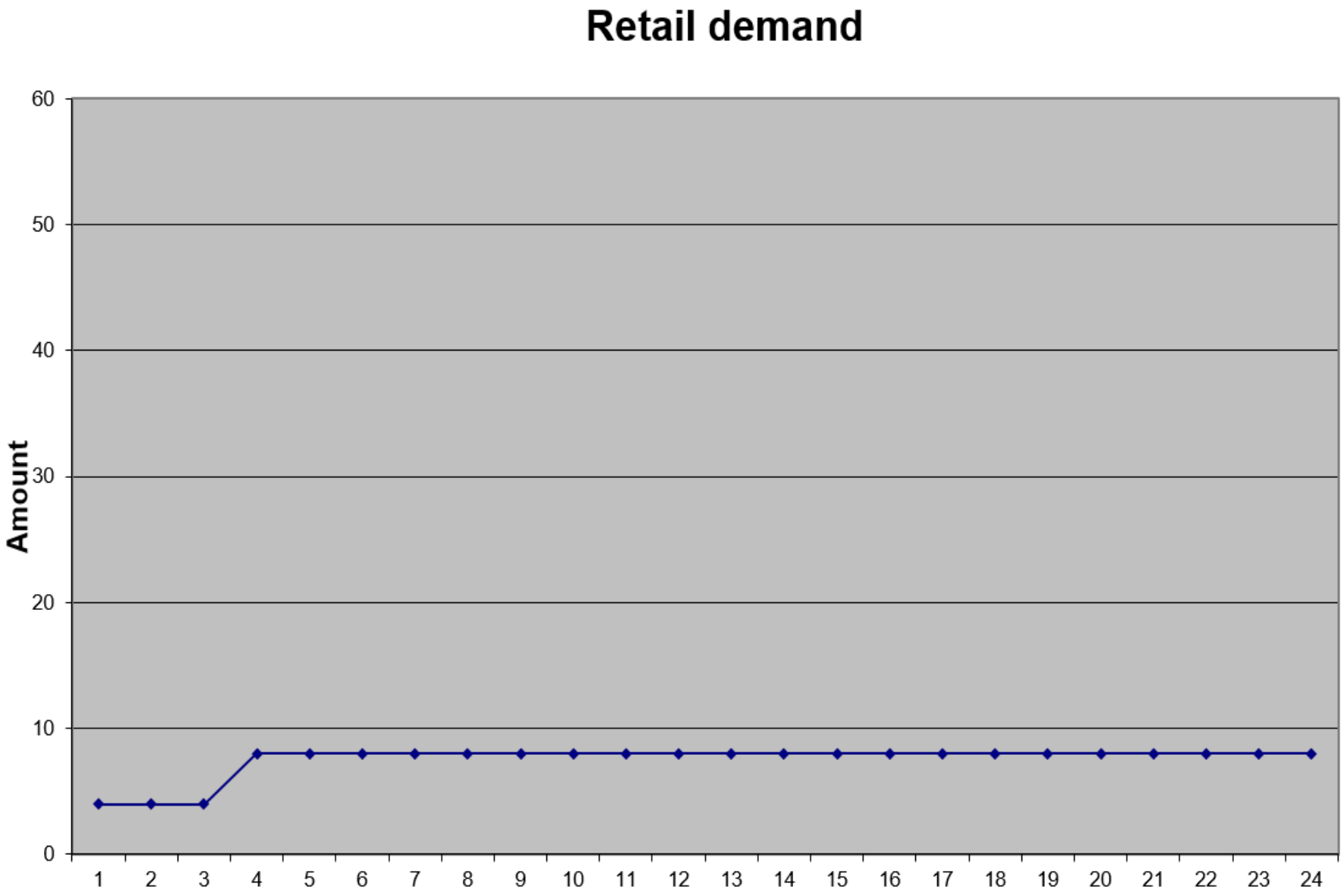










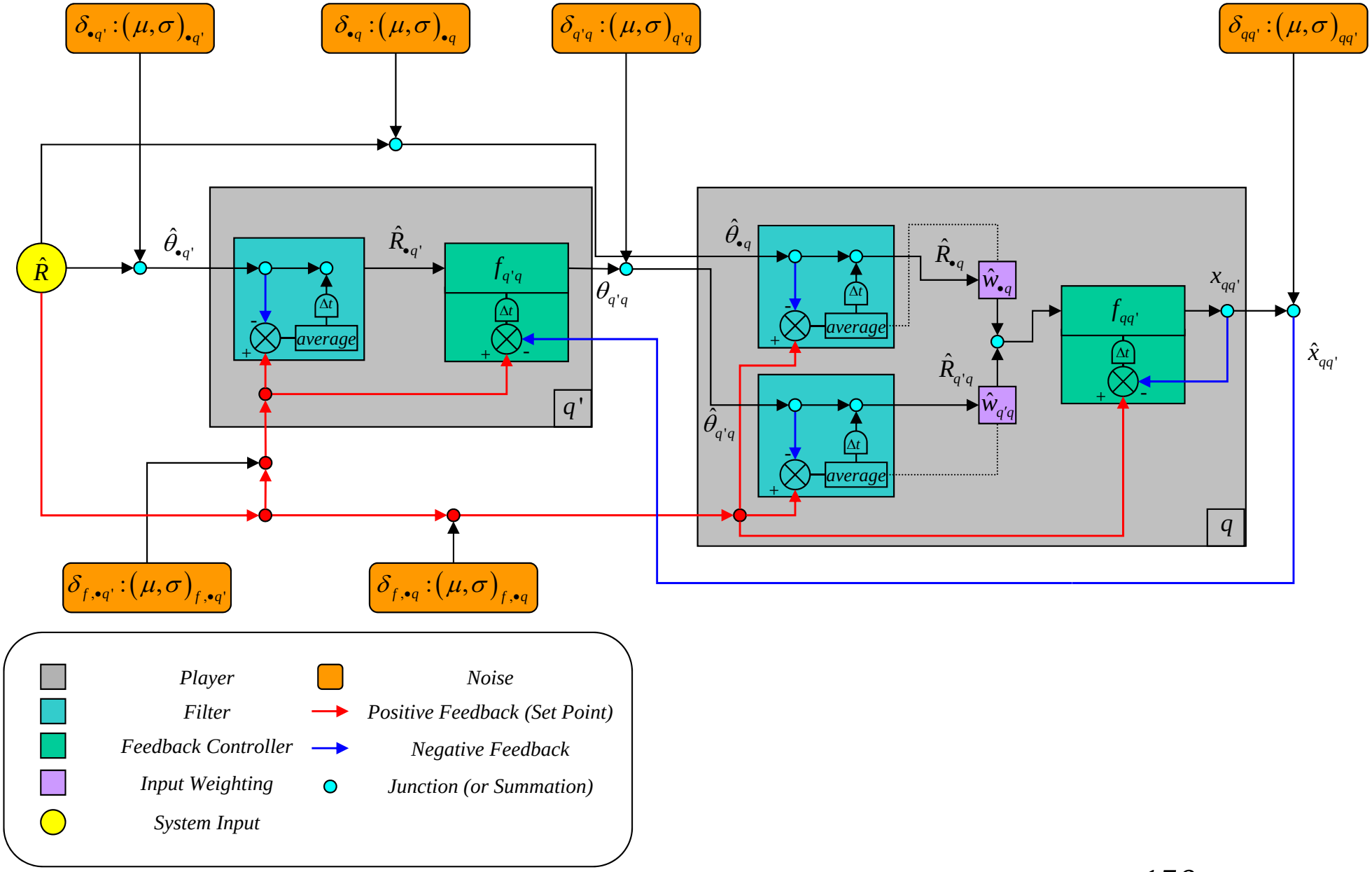


Information exchange problems

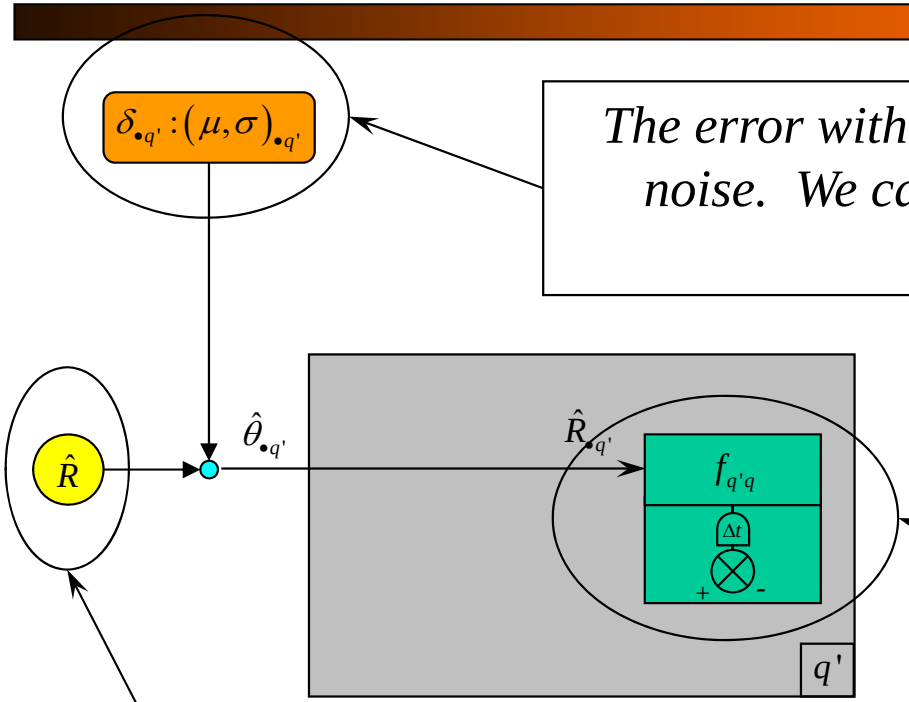
Communication network

Draw on board – handout copy of full diagram to students

Two-agent newsvendor schematic



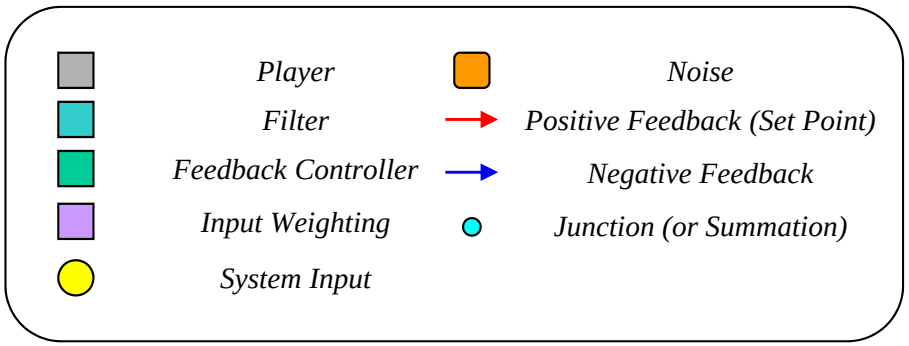
Two-agent newsvendor schematic



The error with which we perceive the need, with bias and noise. We can reduce these by making investments in information technology.

The "agent" that has to think and decide how much to order.

Exogenous requirement. $\hat{R}$ represents the real need which we cannot measure perfectly.

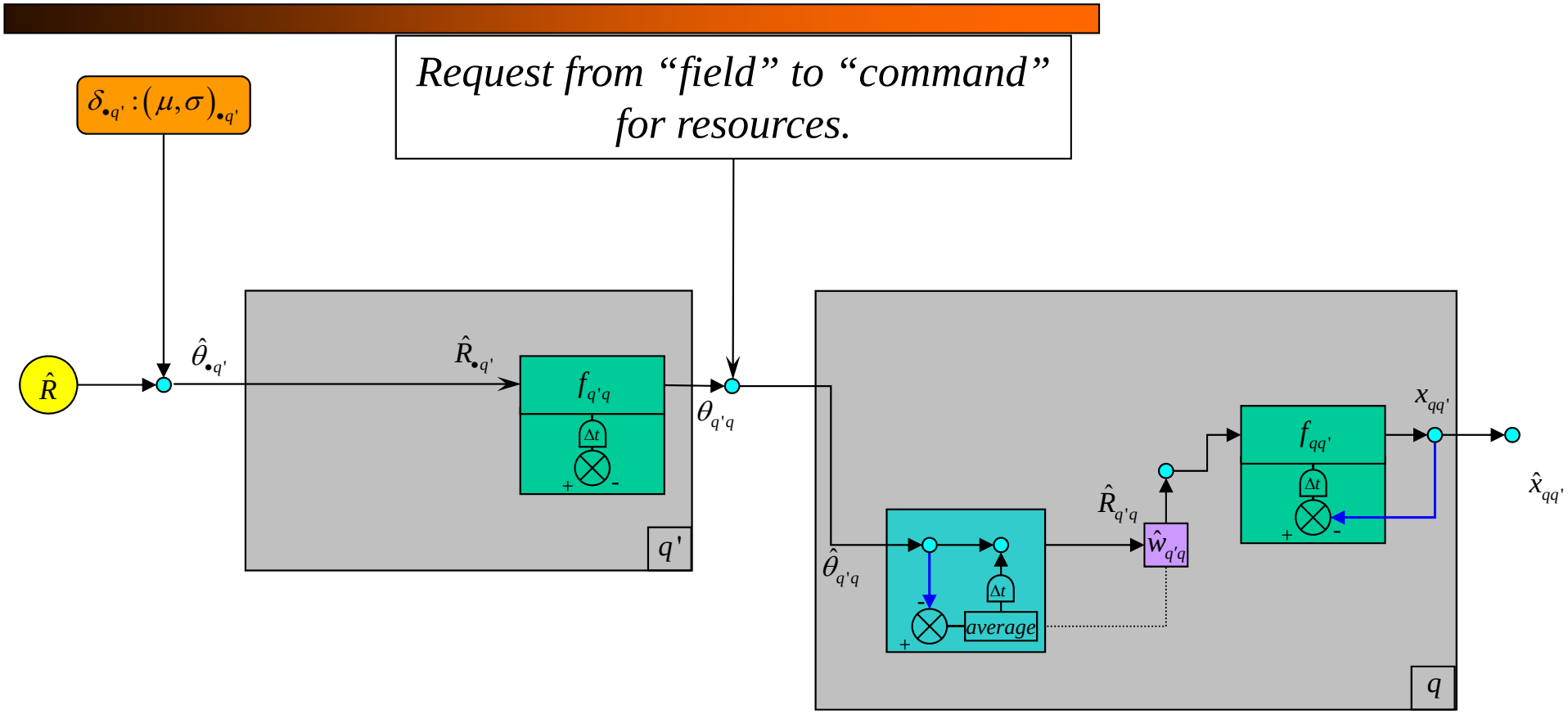


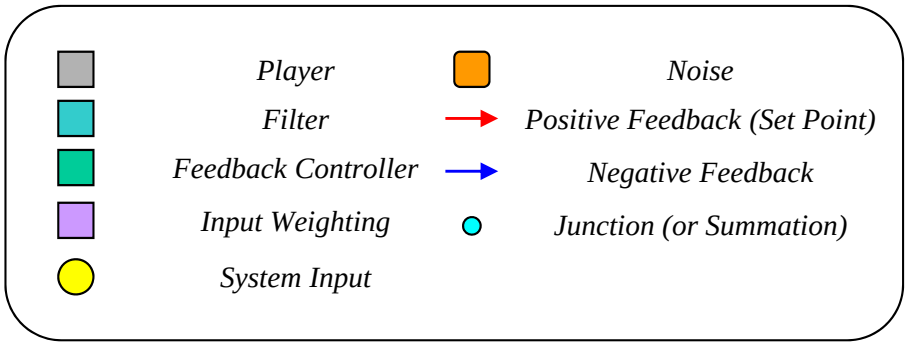
Our agent wants to solve:

$$\min_x \left\{ c^o_{q'} \left(x - \hat{R} \right)^+ + c^u_{q'} \left(\hat{R} - x \right)^+ \right\}$$

As in the past, he does not know exactly what is required, but does know if he requested too much or too little.

Two-agent newsvendor schematic





Two-agent newsvendor schematic

The field agent q' wants to solve:

$$\min_x \left\{ c^o_{q'} \left(x - \hat{R} \right)^+ + c^u_{q'} \left(\hat{R} - x \right)^+ \right\}$$

The central command q wants to solve:

$$\min_x \left\{ c^o_{q} \left(x - \hat{R} \right)^+ + c^u_{q} \left(\hat{R} - x \right)^+ \right\}$$

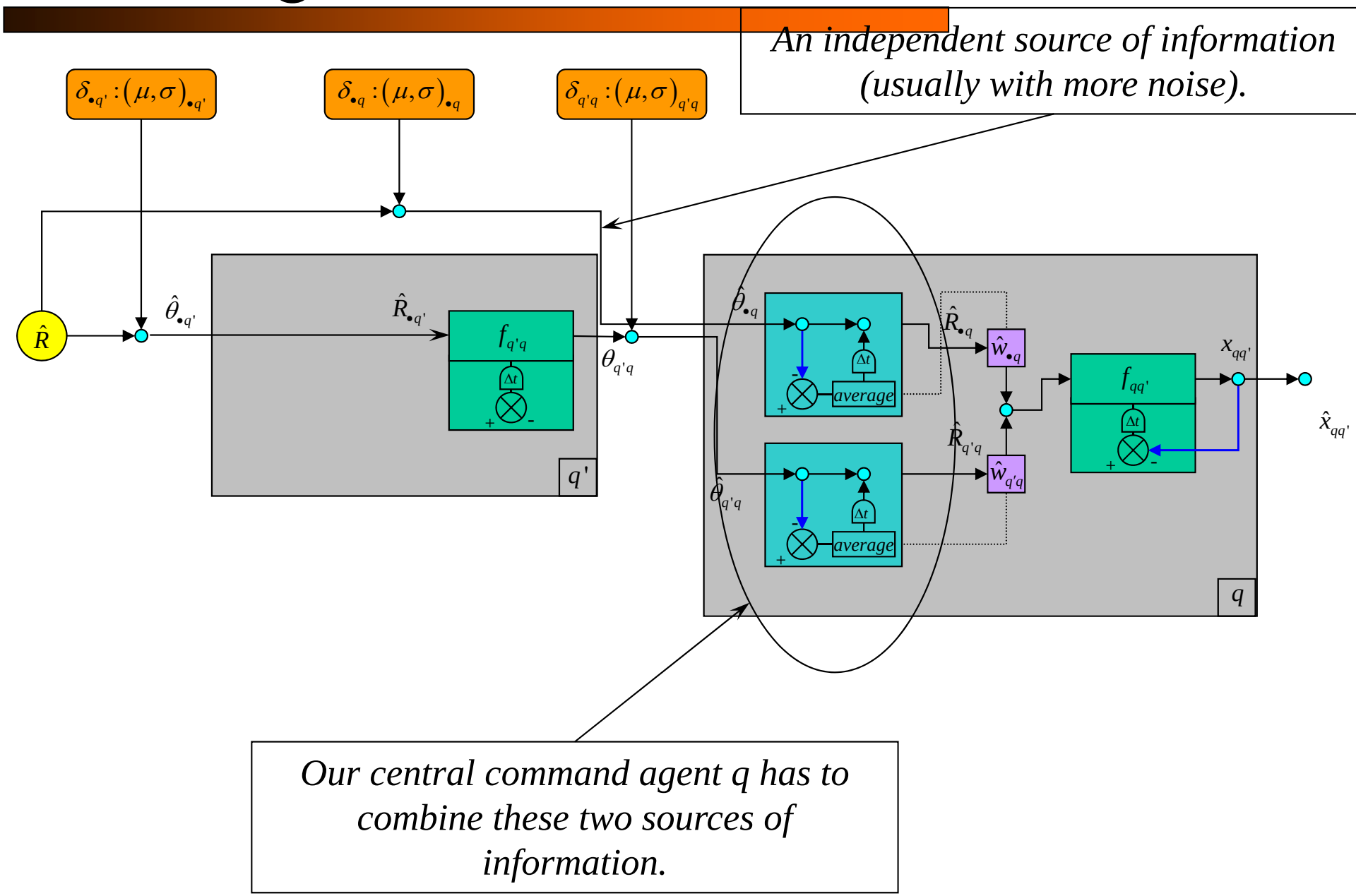
Typically:

$$c^u_q < c^u_{q'}$$

because the field is usually much more risk averse.

How does this affect the exchange of information between the agents?

Two-agent newsvendor schematic



Two-agent newsvendor schematic

Let:

$\hat{R}_{q'q}$ = Information in the form of a request for resources from q' to q.

$\hat{R}_{\circ q}$ = Information about the requirement from an independent source.

Each of these flows of information come with error with mean μ and variance σ^2.

The 'central' combines these requests using a weighted average:

$$\hat{R}_q = \hat{w}_{q'q}\hat{R}_{q'q} + \hat{w}_{\circ q}\hat{R}_{\circ q}$$

The best weights are proportional to the inverse estimates of the variance of the quality of these two sources of information. Let:

$s^2_{q'q}$ = Estimate of the variance of the noise coming from q'.

$s^2_{\circ q}$ = Estimate of the variance of the noise coming from the exogenous source.

Now set the weights to:

$$\hat{w}_{q'q} = \frac{\frac{1}{s^2_{q'q}}}{\frac{1}{s^2_{q'q}} + \frac{1}{s^2_{\circ q}}} = \frac{s^2_{\circ q}}{s^2_{q'q} + s^2_{\circ q}} \qquad \hat{w}_{\circ q} = \frac{\frac{1}{s^2_{\circ q}}}{\frac{1}{s^2_{q'q}} + \frac{1}{s^2_{\circ q}}} = \frac{s^2_{q'q}}{s^2_{q'q} + s^2_{\circ q}}$$

Two-agent newsvendor schematic

Take a close look at our information weights:

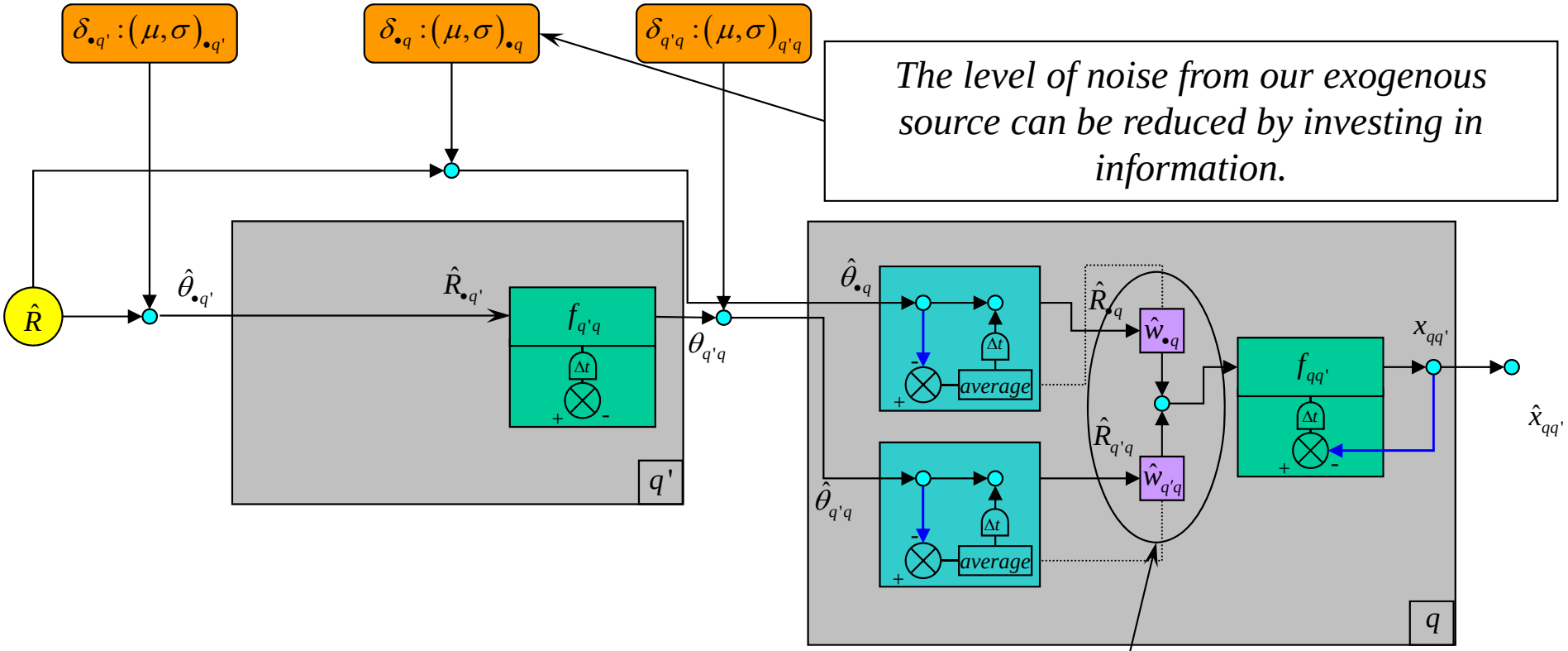
$$\hat{w}_{q'q} = \frac{s^2_{\circ q}}{s^2_{q'q} + s^2_{\circ q}} \qquad \hat{w}_{\circ q} = \frac{s^2_{q'q}}{s^2_{q'q} + s^2_{\circ q}}$$

The weight we put on the information from q' increases with the variance of the information from the exogenous source.

Similarly, the weight we put on the information from the exogenous source increases with the variance of the information from agent q'.

This weighting makes intuitive sense. It is also the optimal weighting, producing a combined estimate with the lowest variance.

Two-agent newsvendor schematic

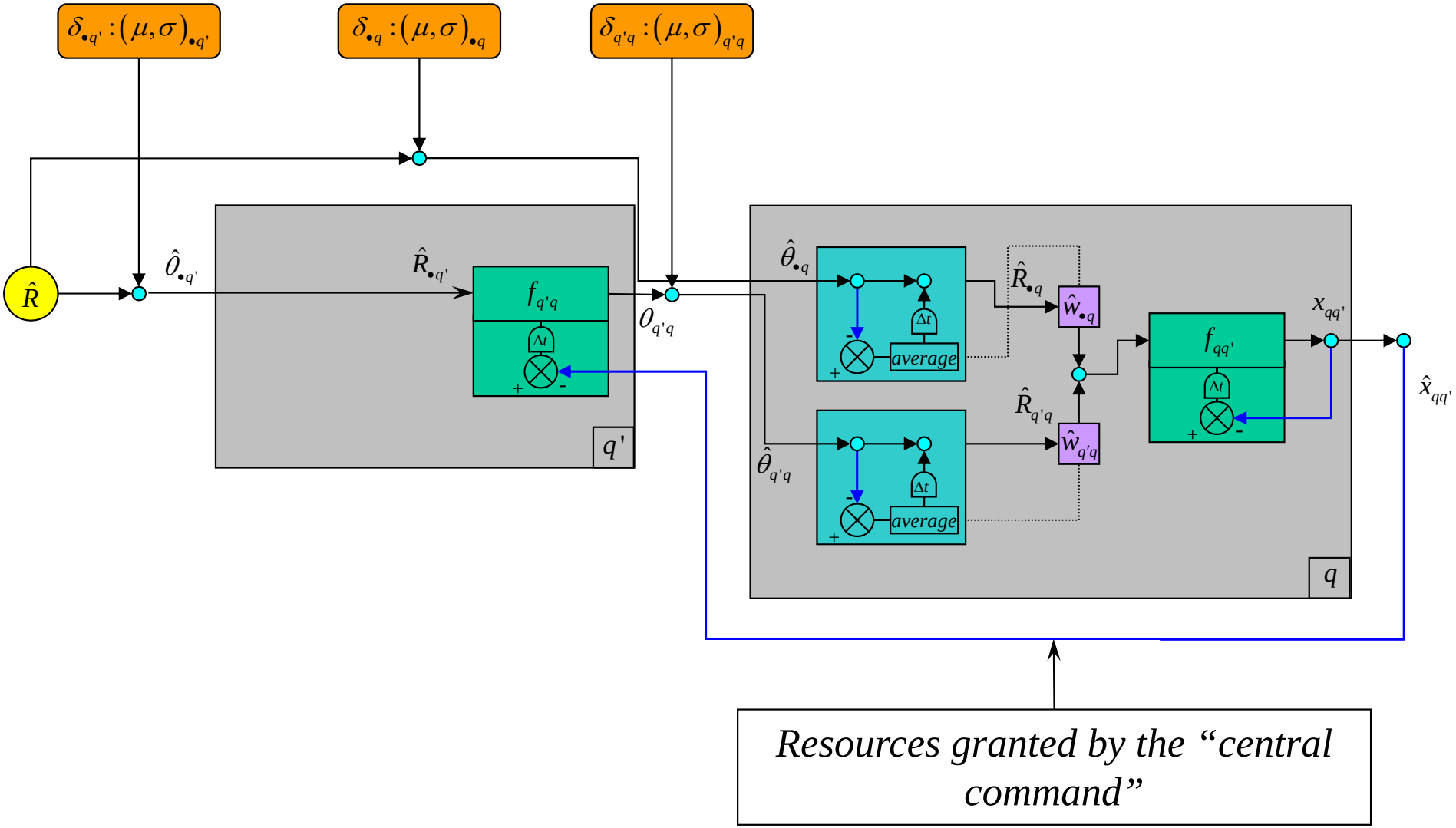


The level of noise from our exogenous source can be reduced by investing in information.

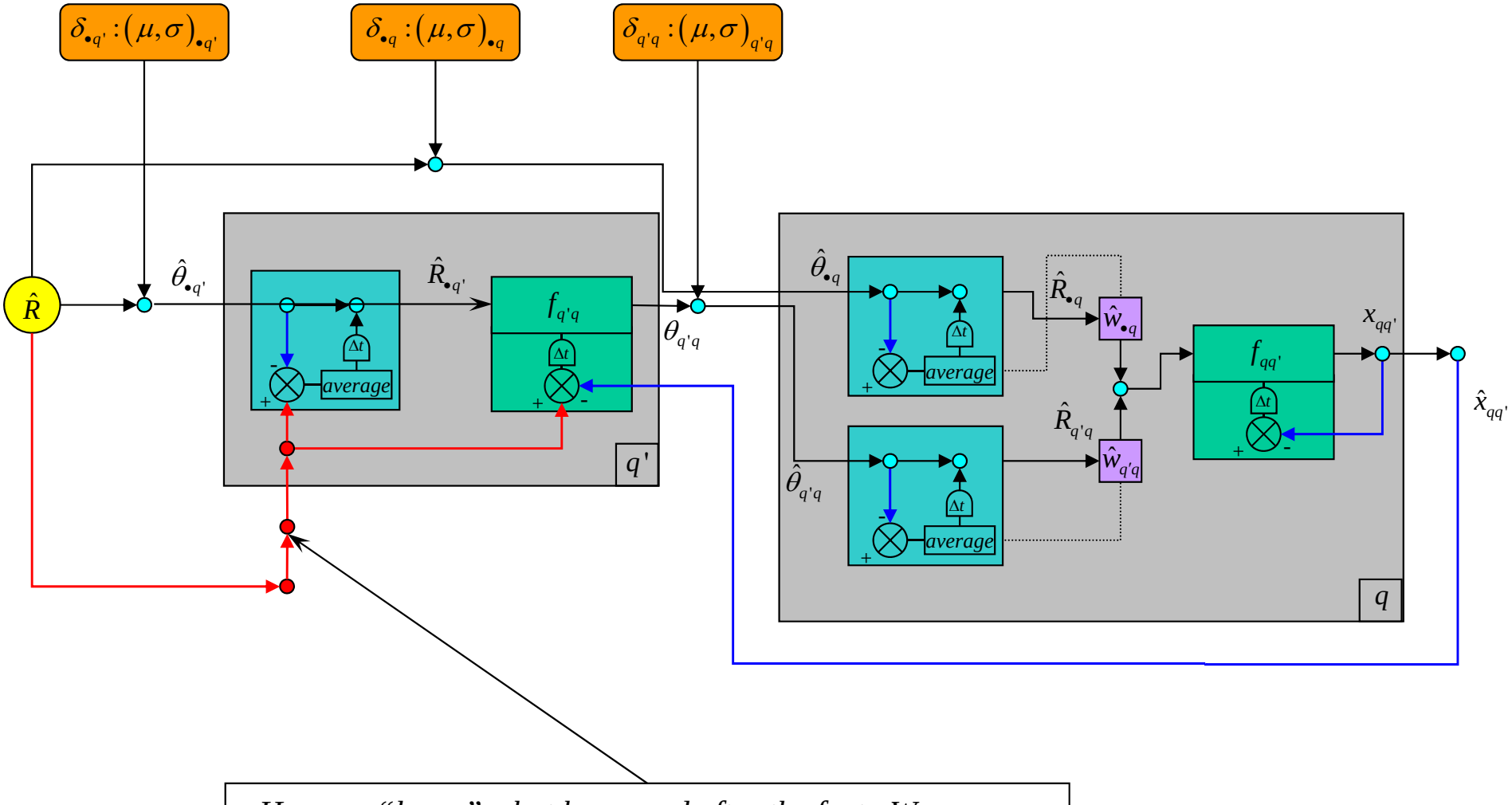
Weighting the two inputs provides the best estimate of the actual need.

But how do we get the estimates of the variances?

Two-agent newsvendor schematic



Two-agent newsvendor schematic



Here we "learn" what happened after the fact. We can use this information to learn about biases and variances, and use this for future calculations.

Two-agent newsvendor schematic

Estimating biases:

$$\hat{\mu}_{\bullet q'}^{k+1} = \hat{\alpha}_{\bullet q'}(\hat{\theta}_{\bullet q'}^{k} - \hat{R}^{k}) + (1-\hat{\alpha}_{\bullet q'})\hat{\mu}_{\bullet q'}^{k}, \qquad \hat{\mu}_{\bullet q'}^{0} = 0$$

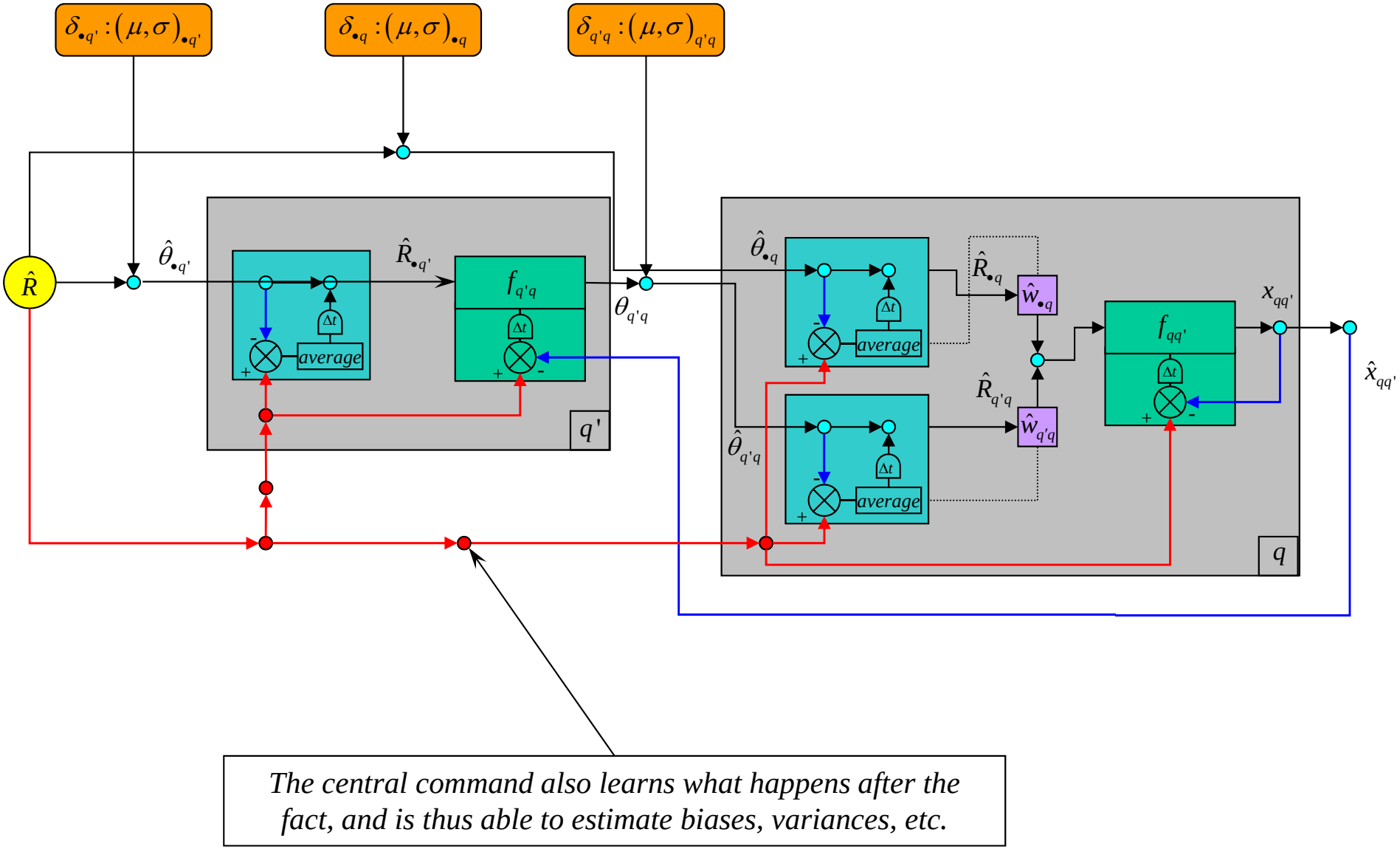
Estimating variances (wherever applicable):

$$\hat{\delta}_{\bullet q'}^{2^{k+1}} = \hat{\alpha}_{\bullet q'}(\hat{\theta}_{\bullet q'}^{k} - \hat{R}^{k})^2 + (1-\hat{\alpha}_{\bullet q'})\hat{\delta}_{\bullet q'}^{2^{k}}, \qquad \hat{\delta}_{\bullet q'}^{2^{0}} = 0$$

$$\hat{\sigma}_{\bullet q'}^{2^{k+1}} = \hat{\delta}_{\bullet q'}^{2^{k+1}} - (\hat{\mu}_{\bullet q'}^{k+1})^2$$

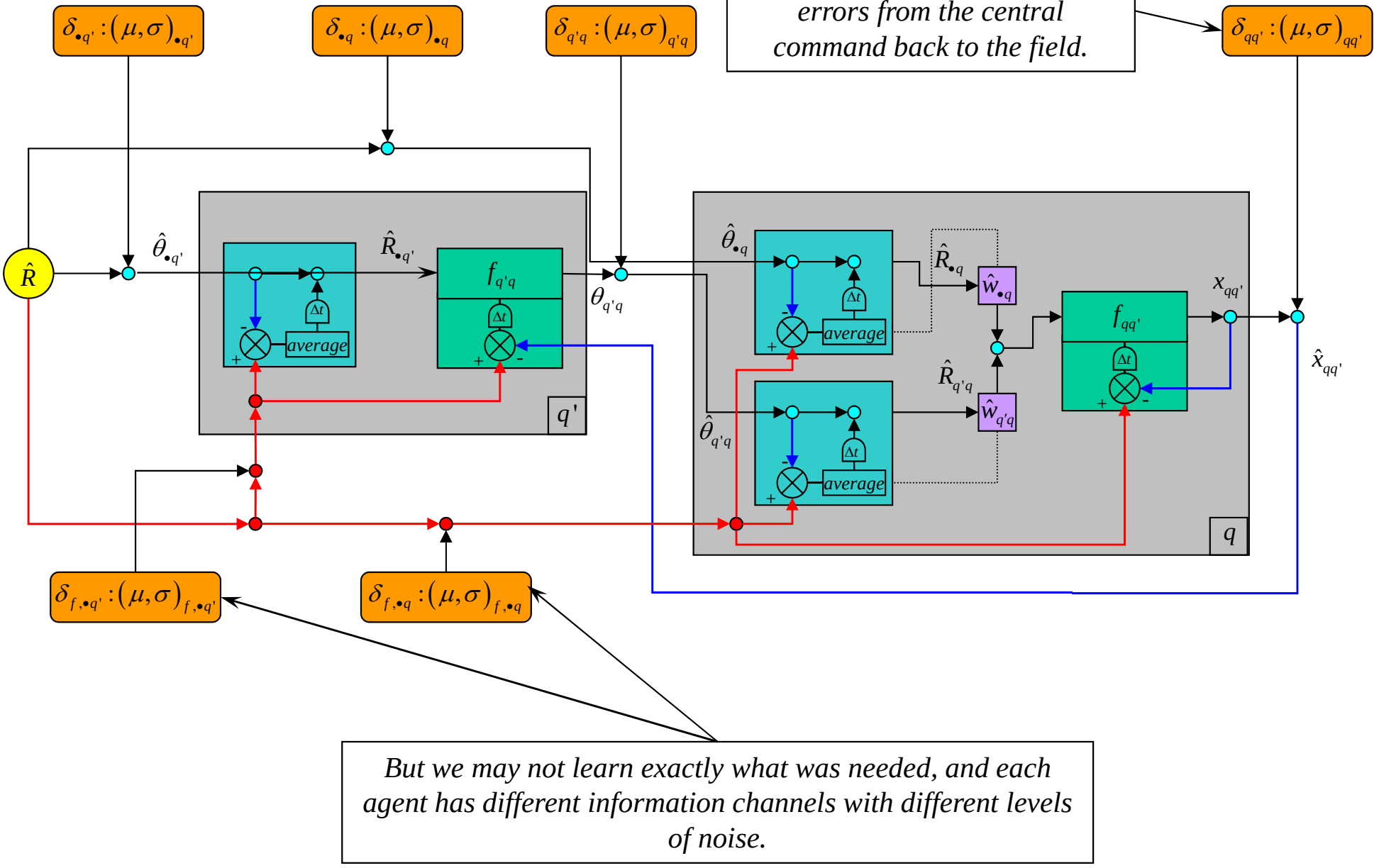
$$s_{\bullet q'}^{2^{k+1}} = \hat{\alpha}_{\bullet q'}\hat{\sigma}_{\bullet q'}^{2^{k+1}} + (1-\hat{\alpha}_{\bullet q'})s_{\bullet q'}^{2^{k}}, \qquad \text{for a given } s_{\bullet q'}^{2^{0}}$$

Two-agent newsvendor schematic

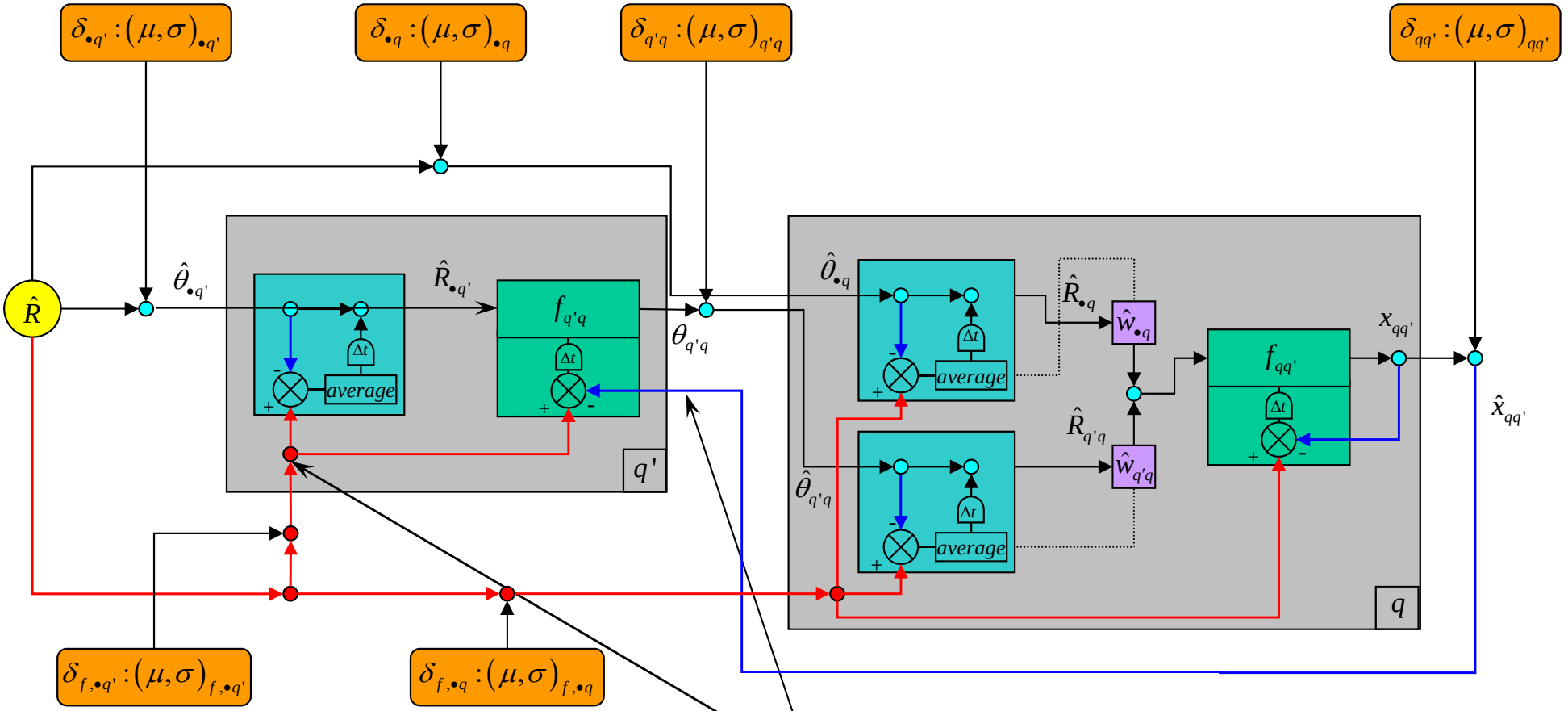


The central command also learns what happens after the fact, and is thus able to estimate biases, variances, etc.

Two-agent newsvendor schematic



Two-agent newsvendor schematic



The agent looks at what was really needed (as best as he can measure it) and compare this to what the central command actually gave him. From this, he can identify biases in the behavior of the central command, and account for this in future requests.

Two-agent newsvendor schematic

Given:

$$\tilde{R}^k = \hat{R}^k + \mu_{f,\bullet q'} + \varepsilon_{f,\bullet q'} \quad \text{where} \quad \varepsilon_{f,\bullet q'} \sim N(0, \sigma^2_{f,\bullet q'})$$

Estimating how much to order:

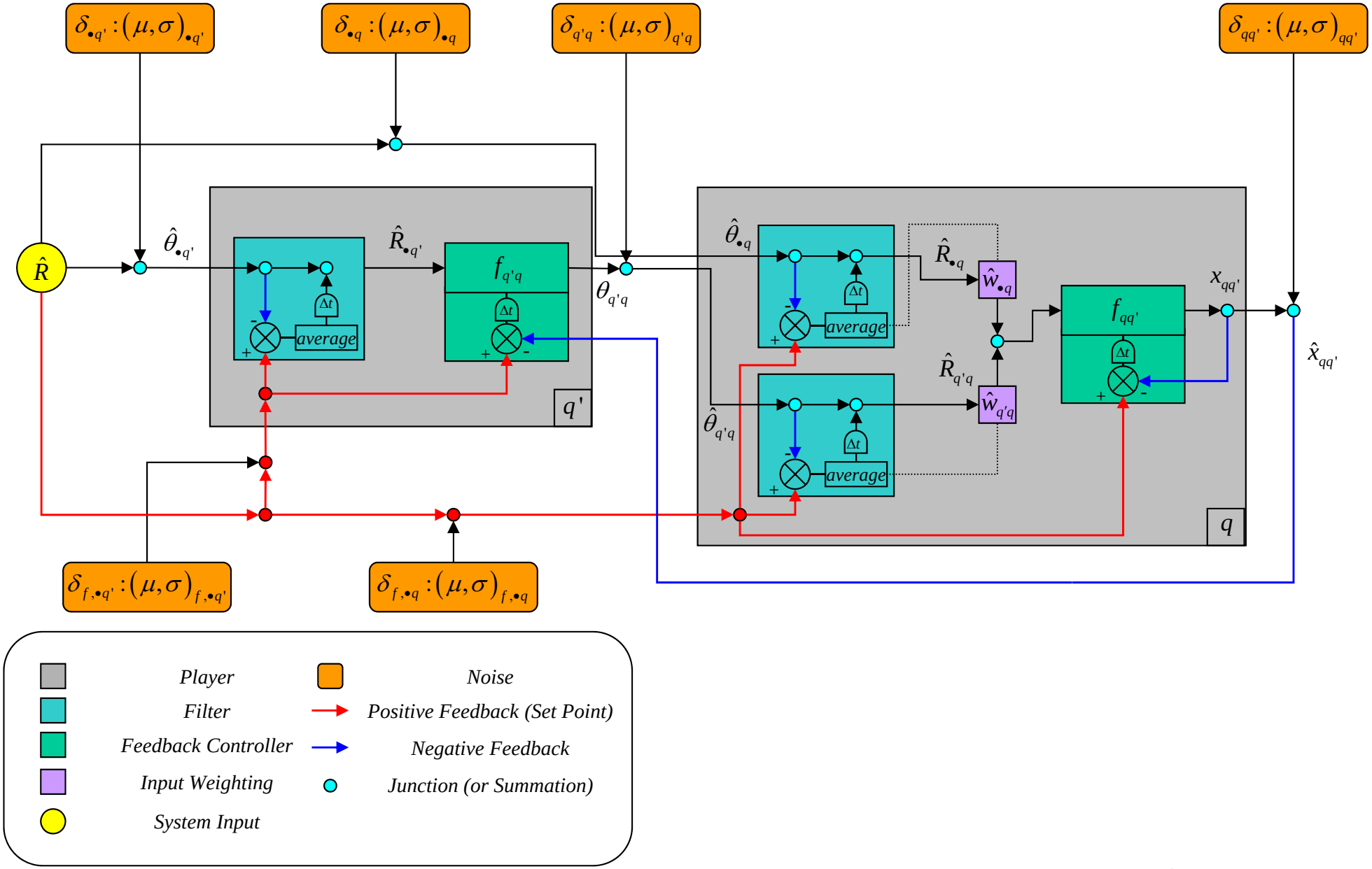
$$\theta^{k+1}_{q'q} = \hat{R}^{k+1}_{\bullet q'} + \beta^{k+1}_{q'q}$$

$$\beta^{k+1}_{q'q} = \beta^{k}_{q'q} - \gamma_{q'} \alpha^k_{q'q} \nabla F\left(\hat{x}^k_{qq'} - \tilde{R}^k, c^o_{q'}, c^u_{q'}\right)$$

where the stochastic gradient can be given by:

$$\nabla F\left(\hat{x}^k_{qq'} - \tilde{R}^k, c^o_{q'}, c^u_{q'}\right) = \left\{ \begin{array}{c} c^o_{q'}, \quad \text{if } \hat{x}^k_{qq'} > \tilde{R}^k \\ -c^u_{q'}, \quad \text{if } \hat{x}^k_{qq'} \le \tilde{R}^k \end{array} \right\}$$

Two-agent newsvendor schematic



Feedback loops and misinformation

Feedback loops and misinformation

- What are sources of misinformation in resource allocation problems?
  - » If we run out of resources, does this mean demand was too high?
  - » If we did not run out of resources, does this mean demand was too low?
  - » How do we estimate productivity? How do we estimate how many resources are required?

The forms of misinformation

- Sources of misinformation in the feedback process:
  - » Bias from information hiding:
    - Censoring:
      - – Too few resources are allocated, all resources are used, and actual demand is hidden.
    - Inflation:
      - – Too many resources are allocated by central command.
      - – Field agent uses resources inefficiently to make it appear that the right amount was requested.
        - » Spend more time finishing a project
        - » Spend money unnecessarily – Ryder example: management bought a lot of furniture and the profits were reduced; company did not want such a large bump in profits – would set unrealistic expectations.
    - Hiding
      - – The resources are there and available to be used, but the field agent does not allow them to be recorded.
        - » Hiding people: make them appear as if they are assigned to a project
        - » Hiding money: put it in a budget allocated to a project

The forms of misinformation

- Sources of misinformation in the feedback process:
  - » Bias from information hiding (cont'd)
    - "Shrinkage"
      - Resources are reduced due to theft, spoilage, breakage.
    - Rationing/hoarding:
      - Too few resources are allocated, but resources are still left over. Agent has held resources for future potential uses, even if demand is not satisfied. So, it appears we have resources, but in reality we would have run out. We were "hiding" low priority demands.
  - » Measurement error
    - May be reduced through investment in information technology.
      - Sensors on equipment; auditing accounts; etc.

Designing policies

Anchor and adjustment for the beer game

Behavioral dynamics

- How did you make decisions?
  - » What was your thought process?
  - » How would you classify your decision-making progress in terms of the four types of policies introduced earlier?
    - Myopic policy
    - Lookahead
    - Policy function approximation (which kind?)
    - Policy based on value function approximation

Behavioral dynamics

- A common policy structure is called **anchor** and **adjustment**:
  - » **Anchor**: A fixed action, possibly based on a ***plan*** (a set of actions based on a particular set of future events or an expected system state).
  - » **Adjustment**: Changes made to the "anchor" based on deviations from planned future events or deviation from an expected system state.

Behavioral dynamics

- What information did we have?

Physical state:

$R(t)$ = Current inventory (or backorder if $R_t < 0$) at time t

Activity variables:

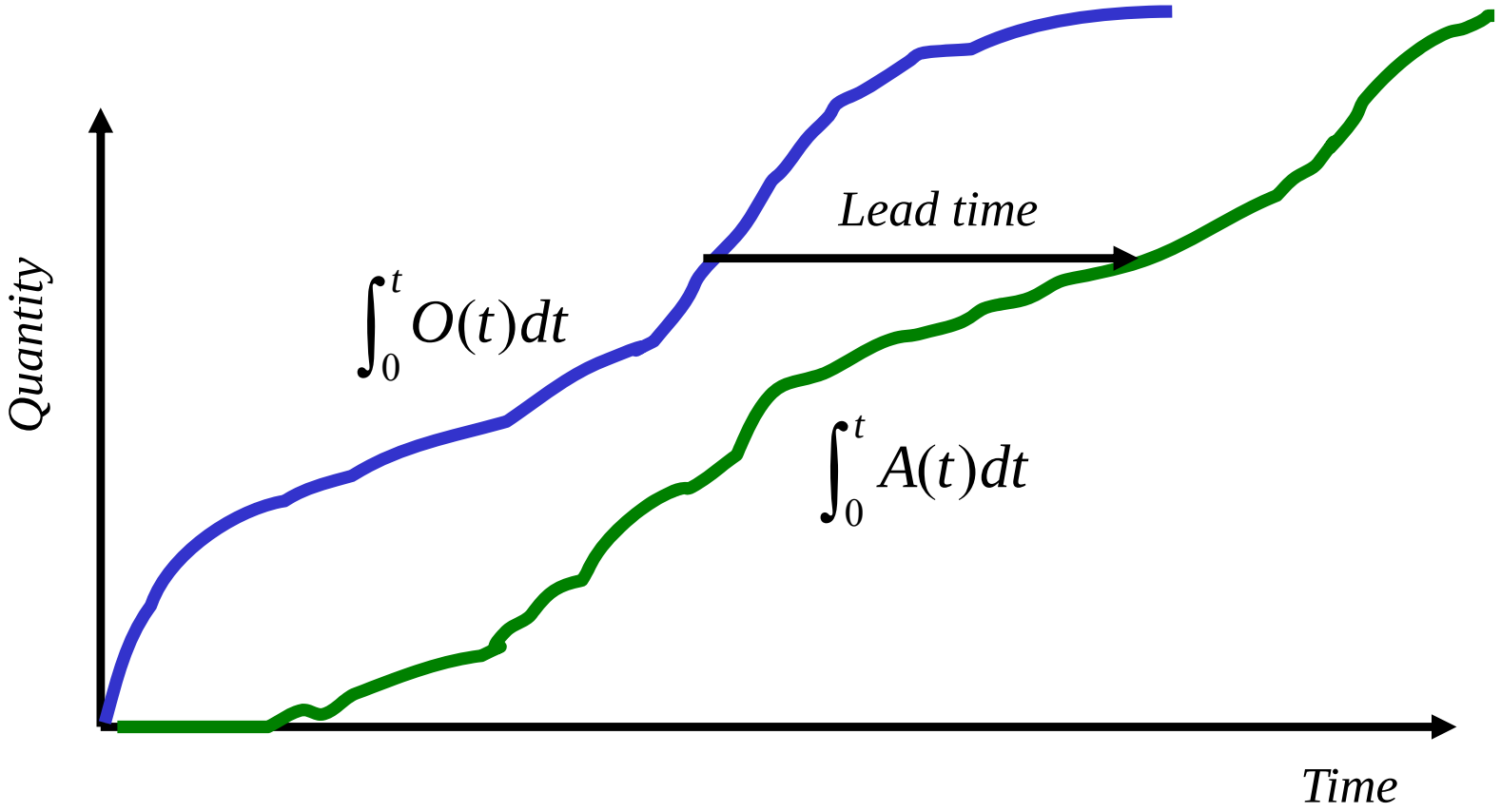
$A(t)$ = Arrival rate

$L(t)$ = Loss rate

$O(t)$ = Order rate

$R^{transit}(t)$ = In-transit inventory (inbound)

Behavioral dynamics



$$R(t) = \int_0^t \big(A(t) - L(t)\big)dt + R(0) = \text{ Current inventory}$$

$$R^{transit}(t) = \int_0^t \big(O(t) - A(t)\big)dt + R^{transit}(0) = \text{ Supply line}$$

Behavioral dynamics

- Observation:
  - » If there was no lead time (and no fixed order cost), what would you do?

$$\tilde{O}(t) = \text{"Indicated order rate" (how much you should order)}$$
$$= L(t-1) \quad \text{(i.e. order what you lost in the previous time period)}$$

  - » Why would you do otherwise?
    - Uncertainties in order lead times?
    - Need to build up larger buffer inventory?
    - Anticipation of higher demand?
    - Current stock under/over what you think you need?

We need to make "adjustments" to our basic model.

Behavioral dynamics

- Refining our model:
  - » Behavioral strategy - anchor and adjustment (Tversky)
    - Anchor - Create a known reference point
    - Adjustment - Change based on position relative to the anchor.
  - » Let:

$$\tilde{L}(t) = \text{Perceived loss rate}$$

$$\delta R(t) = \text{Adjustment to order based on current stock.}$$

$$\delta R^{transit}(t) = \text{Adjustment to order based on current supply line.}$$

  - » New behavioral model:

$$\tilde{O}(t) = \tilde{L}(t) + \delta R(t) + \delta R^{transit}(t)$$

  - » Since this may be negative, our new ordering model is:

$$O(t) = \left[\tilde{O}(t)\right]^{+}$$

Behavioral dynamics

- Calculating the perceived loss rate:
  - » Possible models:

Reactive:

$\tilde{L}(t) = L(t-1)$ Perceived loss equal to last period's actual loss.

Stable:

$\tilde{L}(t) = L^*$ Equal to a constant

Regressive expectations

$$\tilde{L}(t) = \gamma L(t-1) + (1-\gamma)L^*$$

Adaptive:

$$\tilde{L}(t) = (1-\gamma)\tilde{L}(t-1) + \gamma L(t-1)$$

Behavioral dynamics

- Adjustments:
  - » For current stock:

$$\delta R(t) = \alpha\left(R^* - R(t)\right)$$

where:

α = damping factor.

R^* = desired stock level (in itself a function)

$R(t)$ = current stock

  - » What is R^* a function of?

Behavioral dynamics

- Adjustments:
  - » For supply line:

$$\delta R^{transit}(t) = \alpha_{SL}\left(R^{transit*}(t) - R^{transit}(t)\right)$$

where:

α_{SL} = damping factor for the supply line.

$R^{transit*}(t)$ = desired supply line level (in itself a function).

$R^{transit}(t)$ = current supply line.

The target supply line might be:

$$R^{transit*}(t) = T^{lag}(t)P^{*}(t)$$

where:

$T^{lag}(t)$ = estimated time lag between order and arrival.

$P^{*}(t)$ = desired product throughput.

Behavioral dynamics

- Observation:
  - » Anchoring and adjustment is an exceptionally powerful model.
  - » General method:
    - Find an anchor by solving a simple version of the problem
    - Develop adjustment mechanisms based on available information.
  - » Examples:
    - What quantity to order (product, investment, energy storage)?
      - – Anchor – based on expectation
      - – Adjustment – over time, build up an idea of safety stock

Week 9 - Wednesday

Thanksgiving!

Week 12 – Blood management

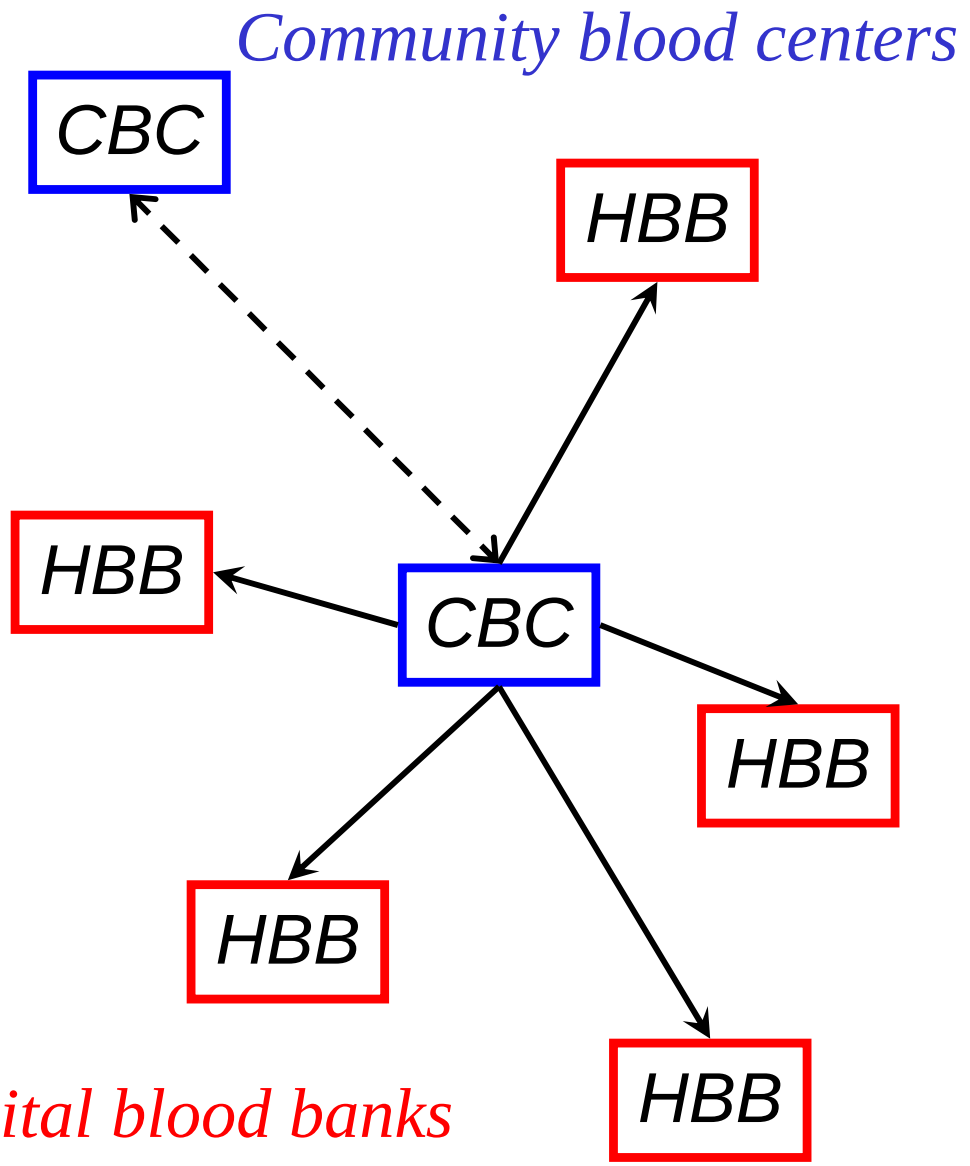
Narrative

Blood management

- Blood collection organizations:
  - » American Red Cross
    - 36 regional centers and five testing facilities
    - Highly centralized
  - » America's Blood Centers
    - Started in competition to Red Cross.
    - Serves independent blood centers.
    - Each handles about half of civilian supply
  - » Department of Defense
    - Runs blood bank program for the military.

Blood management

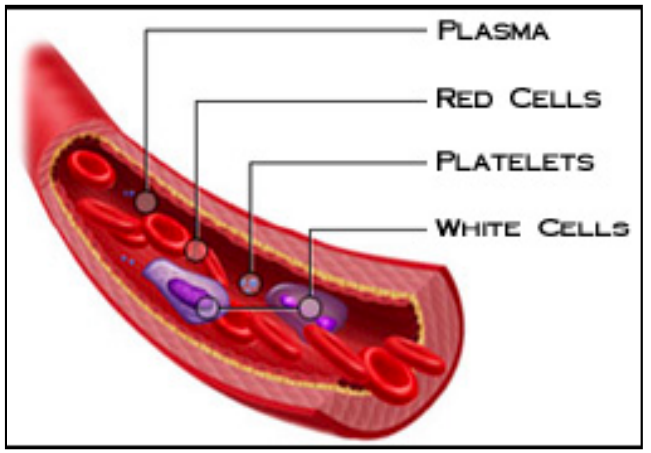
- Donation Process
  - » Eligibility Requirements
  - » Blood donors vary by race, which affects the distribution of blood types.
  - » Some donated blood is lost due to contamination.
- Transfusion Process
  - » Community blood centers distribute to hospitals
  - » Crossmatching process



Blood management

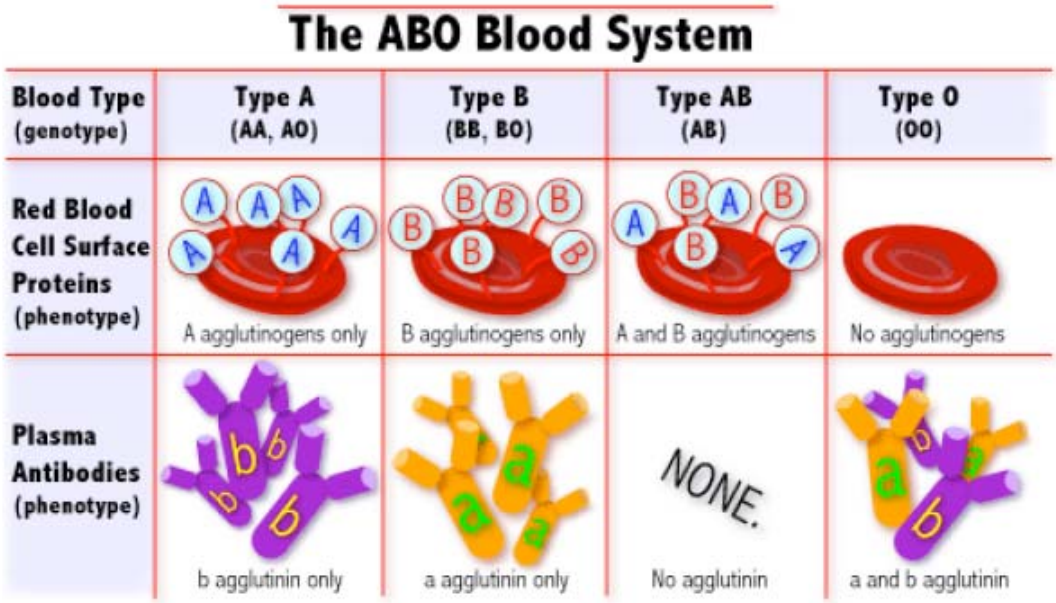
- Components of Blood

- Red Blood Cells:
  - » 42 day shelf-life
  - » 8 blood types with various substitution possibilities
  - » Different products such as frozen RBC



Blood management

- Blood types:
  - » A and B antigens
    - Produces A, B, AB and O
  - » The Rh antigen
    - Present (+)
    - Not present (-)



Blood management

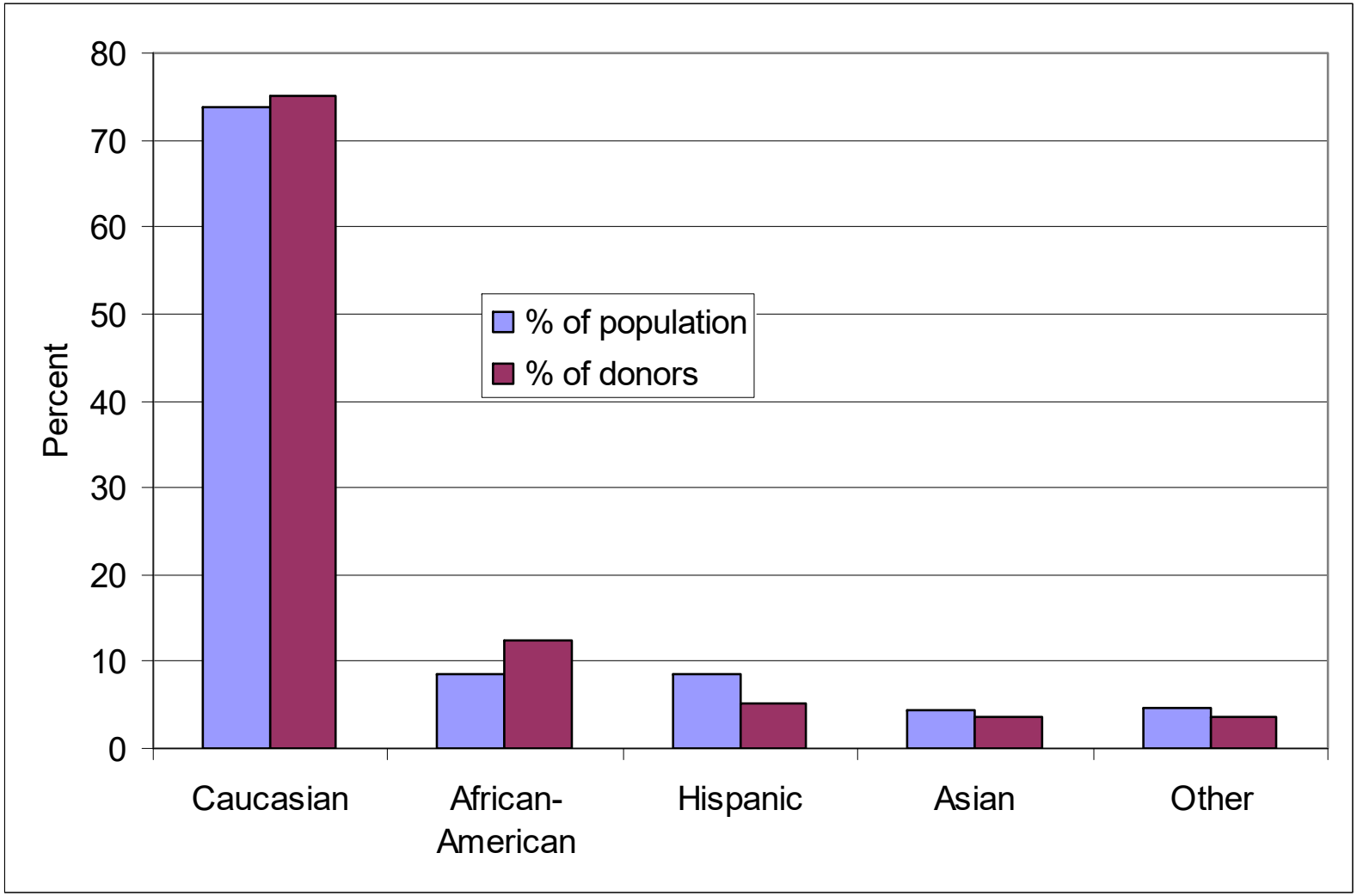
Table 2-1: Blood Type Distribution by Ethnic Group

Ethnic Group	Type O	Type A	Type B	Type AB
US (Caucasian)	45	40	11	4
US (African American)	49	27	20	4
Hawaiian	37	61	2	1
Chinese-Canton	46	23	25	6
Brazilians	47	41	9	3
Germans	41	43	11	5
French	43	47	7	3
Peru (Indians)	100	0	0	0
Russians	33	36	23	8

Source: Blood Book Website [2]

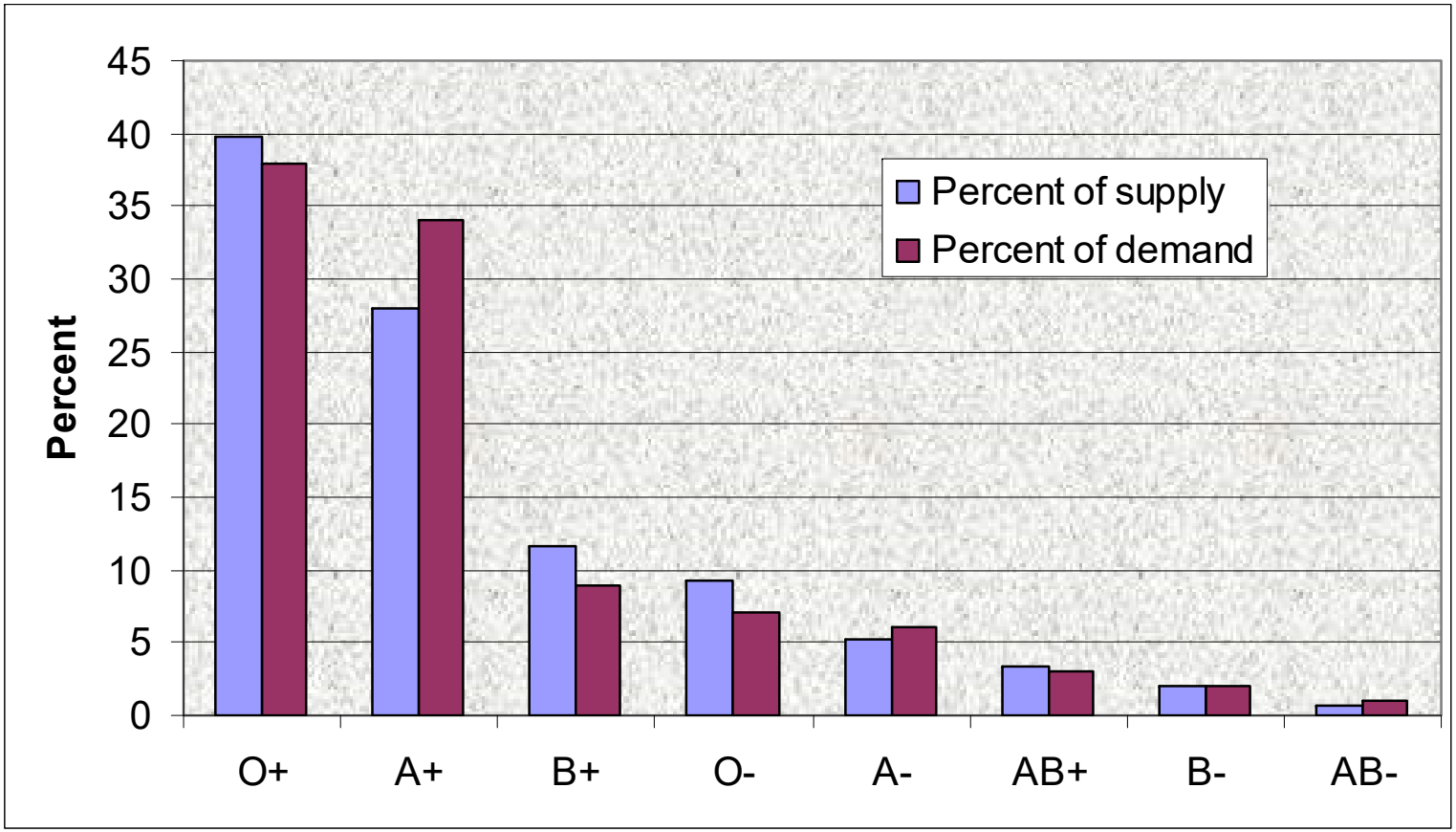
Blood management

- Donations by ethnicity

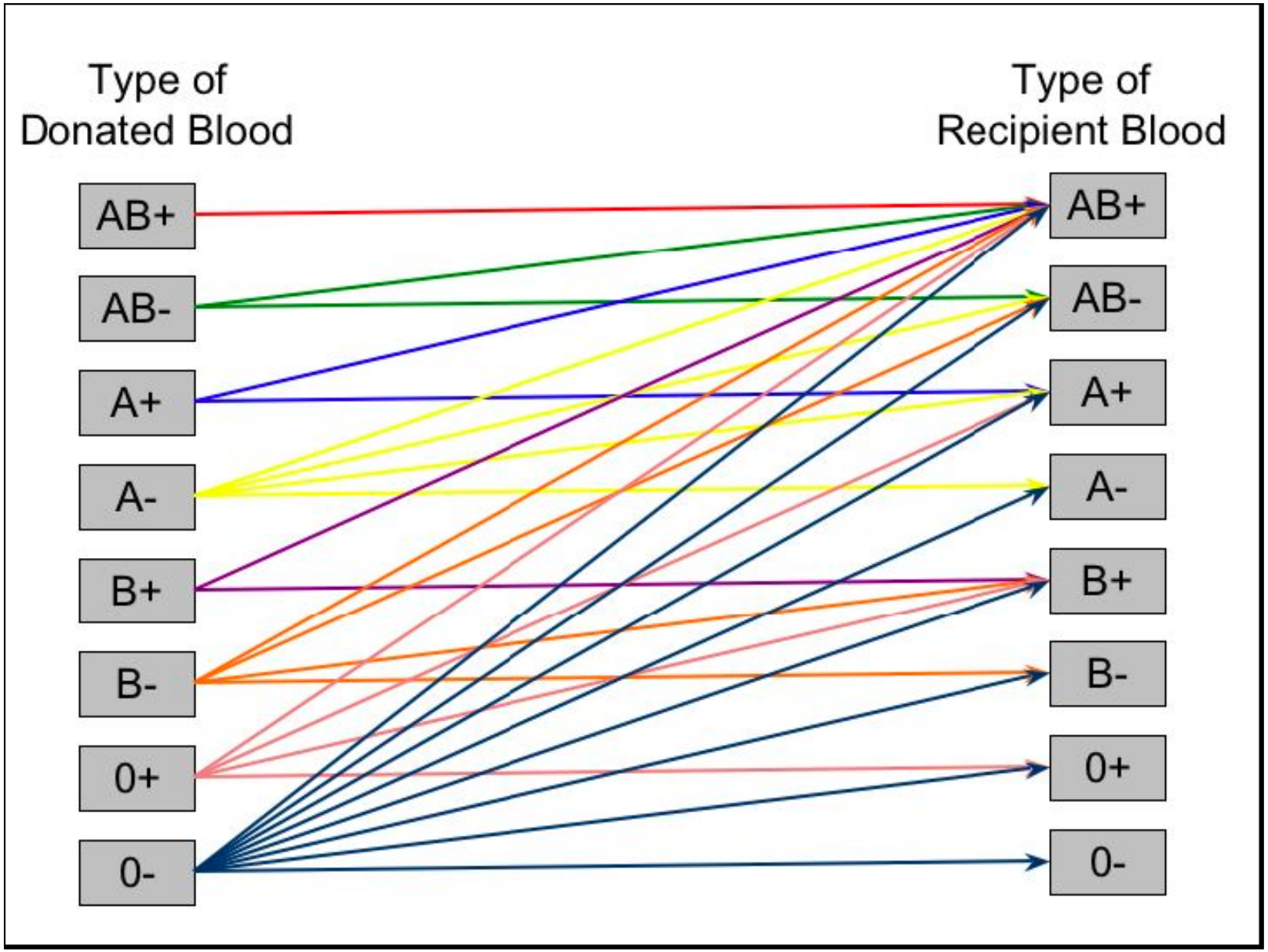


Blood management

- Supply and demand distributions
  - » Donations
    - Depends on age, race, income and geography



Blood management



Blood management

- Substitution
  - » O- is universal donor
  - » AB+ is universal recipient
  - » Substitution is not allowed for some operations
    - Mothers giving birth – infants cannot handle substitution of any type.
    - Preference of surgeon for some operations

Blood management

- Storage
  - » Red blood cells (RBC's) require constant metabolic energy to survive.
  - » Blood Ph has to be maintained in range of 6.4 to 7.2.
  - » Maximum (safe) shelf life is 42 days.
    - Determined by requirement that 75 percent of blood cells must be alive 24 hours after transfusion.
  - » Must be refrigerated – temperature must be maintained between 2 and 6 degrees C.
  - » Freezing
    - Frozen blood may be stored indefinitely.
    - Requires 60 to 90 minutes to thaw.
    - Must be used within 24 hours after thawing.

Basic model

Blood management

State variables

We can model the blood problem as a heterogeneous resource allocation problem. We are going to start with a fairly basic model which can be easily extended with almost no notational changes. We begin by describing the attributes of a unit of stored blood using

$$b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} \text{Blood type } (A+, A-, \dots) \\ \text{Age (in weeks)} \end{pmatrix},$$

$$\mathcal{B} = \text{Set of all attribute types.}$$

We will limit the age to the range $0 \leq a_2 \leq 6$. Blood with $a_2 = 6$ (which means blood that is already six weeks old) is no longer usable. We assume that decision epochs are made in one-week increments.

Blood inventories, and blood donations, are represented using

$$\begin{aligned} R_{tb} &= \text{Units of blood of type } b \text{ available to be assigned or held at time } t, \\ R_t &= (R_{tb})_{b \in \mathcal{B}}, \end{aligned}$$

The pre- and post-decision state variables are given by

$$\begin{aligned} S_t &= (R_t, \hat{D}_t), \\ S_t^x &= (R_t^x). \end{aligned}$$

Blood management

- Blood types

$$a = \begin{pmatrix} type \\ location \\ age \\ previously\ frozen? \end{pmatrix} \in \mathcal{A}$$

- Demand types

$$b = \begin{pmatrix} type \\ location \\ surgery\ type \\ substitution? \end{pmatrix} \in \mathcal{B}$$

Blood management

Resource state

$R_{ta} =$ the number of units of blood at time t with attribute a

$R_t = \left(R_{ta} \right)_{a \in \mathcal{A}}$

Demands

$\hat{D}_{tb} =$ the number of units of blood needed at time t with attribute b.

$\hat{D}_t = \left(\hat{D}_{tb} \right)_{b \in \mathcal{B}}$

Blood management

Decision variables

We act on blood resources with decisions given by:

$$\begin{aligned} x_{tbd} &= \text{Number of units of blood with attribute } b \text{ that we assign to a demand of type } d, \\ x_t &= (x_{tad})_{b\in\mathcal{B},d\in\mathcal{D}}. \end{aligned}$$

The feasible region $\mathcal{X}_t$ is defined by the following constraints:

$$\sum_{d\in\mathcal{D}} x_{tbd} = R_{tb}, \tag{16.1}$$

$$\sum_{b\in\mathcal{B}} x_{tbd} \leq \hat{D}_{td}, \quad d \in \mathcal{D}, \tag{16.2}$$

$$x_{tbd} \geq 0. \tag{16.3}$$

Blood management

- Decisions

$\mathcal{D}^B$ = Set of decisions to serve demands of any given type.

For each decision $d \in \mathcal{D}^B$, there is a blood type $b_d \in \mathcal{B}$.

d^h = Decision to hold a unit of blood in inventory until next week.

$\mathcal{D} = \mathcal{D}^B \cup d^h$

x_{tad} = The # of units of blood with attribute a acted on at time t by a decision of type d (with demand attribute b)

$x_t = \left(x_{tad}\right)_{a \in \mathcal{A}, d \in \mathcal{D}}$

Blood management

Exogenous information

$$
\begin{aligned}
\hat{R}_{tb} &= \text{Number of new units of blood of type } b \text{ donated between } t-1 \text{ and } t, \\
\hat{R}_t &= (\hat{R}_{tb})_{b \in \mathcal{B}}.
\end{aligned}
$$

The attributes of demand for blood are given by

$$
d = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix} = \begin{pmatrix} \text{Blood type of patient} \\ \text{Surgery type: urgent or elective} \\ \text{Is substitution allowed?} \end{pmatrix},
$$

$$
\begin{aligned}
d^{\phi} &= \text{Decision to hold blood in inventory (“do nothing”)}, \\
\mathcal{D} &= \text{Set of all demand types } d \text{ plus } d^{\phi}.
\end{aligned}
$$

The attribute d_3 captures the fact that there are some operations where a doctor will not allow any substitution. One example is childbirth, since infants may not be able to handle a different blood type, even if it is an allowable substitute. For our basic model, we do not allow unserved demand in one week to be held to a later week.

The new demands for blood are modeled using

$$
\begin{aligned}
\hat{D}_{td} &= \text{Units of demand with attribute } d \text{ that arose between } t-1 \text{ and } t, \\
\hat{D}_t &= (\hat{D}_{td})_{d \in \mathcal{D}}.
\end{aligned}
$$

Blood management

Transition function

Blood that is held simply ages one week, but we limit the age to six weeks. Blood that is assigned to satisfy a demand can be modeled as being moved to a blood-type sink, denoted, perhaps, using $b_{t,1} = \phi$ (the null blood type). The blood attribute transition function $r^M(b_t, d_t)$ is given by

$$b_{t+1} = \begin{pmatrix} b_{t+1,1} \\ b_{t+1,2} \end{pmatrix} = \begin{cases} \begin{pmatrix} b_{t,1} \\ \min\{6, b_{t,2}+1\} \end{pmatrix}, & d_t = d^\phi, \\ \begin{pmatrix} \phi \\ - \end{pmatrix}, & d_t \in \mathcal{D}. \end{cases}$$

To represent the transition function, it is useful to define

$$\begin{aligned} \delta_{b'}(b,d) &= \begin{cases} 1 & b_t^x = b' = r^M(b_t, d_t), \\ 0 & \text{otherwise}, \end{cases} \\ \Delta &= \text{Matrix with } \delta_{b'}(b,d) \text{ in row } b' \text{ and column } (b,d). \end{aligned}$$

We note that the attribute transition function is deterministic. A random element would arise, for example, if inspections of the blood resulted in blood that was less than six weeks old being judged to have expired. The resource transition function can now be written

$$\begin{aligned} R^x_{tb'} &= \sum_{b\in\mathcal{B}} \sum_{d\in\mathcal{D}} \delta_{b'}(b,d) x_{tbd}, \\ R_{t+1,b'} &= R^x_{tb'} + \hat{R}_{t+1,b'}. \end{aligned}$$

In matrix form, these would be written

$$R^x_t = \Delta x_t, \tag{16.4}$$

$$R_{t+1} = R^x_t + \hat{R}_{t+1}. \tag{16.5}$$

Blood management

Objective function

There is no real "cost" to assigning blood of one type to demand of another type (we are not considering steps such as spending money to encourage additional donations, or transporting inventories from one hospital to another). Instead, we use the contribution function to capture the preferences of the doctor. We would like to capture the natural preference that it is generally better not to substitute, and that satisfying an urgent demand is more important than an elective demand. For example, we might use the contributions described in table 16.2. Thus, if we use $O-$ blood to satisfy the needs for an elective patient with $A+$ blood, we would pick up a -\$10 contribution (penalty since it is negative) for substituting blood, a +\$5 for using $O-$ blood (something the hospitals like to encourage), and a +\$20 contribution for serving an elective demand, for a total contribution of +\$15.

The total contribution (at time t) is finally given by

$$C_t(S_t, x_t) = \sum_{b \in \mathcal{B}} \sum_{d \in \mathcal{D}} c_{tbd} x_{tbd}.$$

As before, let $X_t^\pi(S_t)$ be a policy (some sort of decision rule) that determines $x_t \in \mathcal{X}_t$ given S_t. We wish to find the best policy by solving

$$\max_{\pi \in \Pi} \mathbb{E} \sum_{t=0}^{T} \gamma^t C_t(S_t, x_t). \tag{16.6}$$

Blood management

- Contributions

c_{tad} = Contribution earned by acting on blood of type a with a decision of type d.

$$c_t = \left(c_{tad}\right)_{a \in \mathcal{A}, d \in \mathcal{D}}$$

$$C_t(x_t) = \sum_{a \in \mathcal{A}} \sum_{d \in \mathcal{D}} c_{tad} x_{tad}$$

- Notes:
  - » "contributions" express preferences.
  - » Matching blood type to demand might produce a "contribution" of 0. Small negative contribution when types do not match.
  - » Larger negative contributions if we hold blood longer.

Blood management

Condition	Description	Value
if $d = d^{\phi}$	Holding	0
if $b_1 = b_1$ when $d \in \mathcal{D}$	No substitution	0
if $b_1 \neq b_1$ when $d \in \mathcal{D}$	Substitution	-10
if $b_1 =$ O- when $d \in \mathcal{D}$	O- substitution	5
if $d_2 =$ Urgent	Filling urgent demand	40
if $d_2 =$ Elective	Filling elective demand	20

Table 16.2 Contributions for different types of blood and decisions

Uncertainty modeling

Blood management

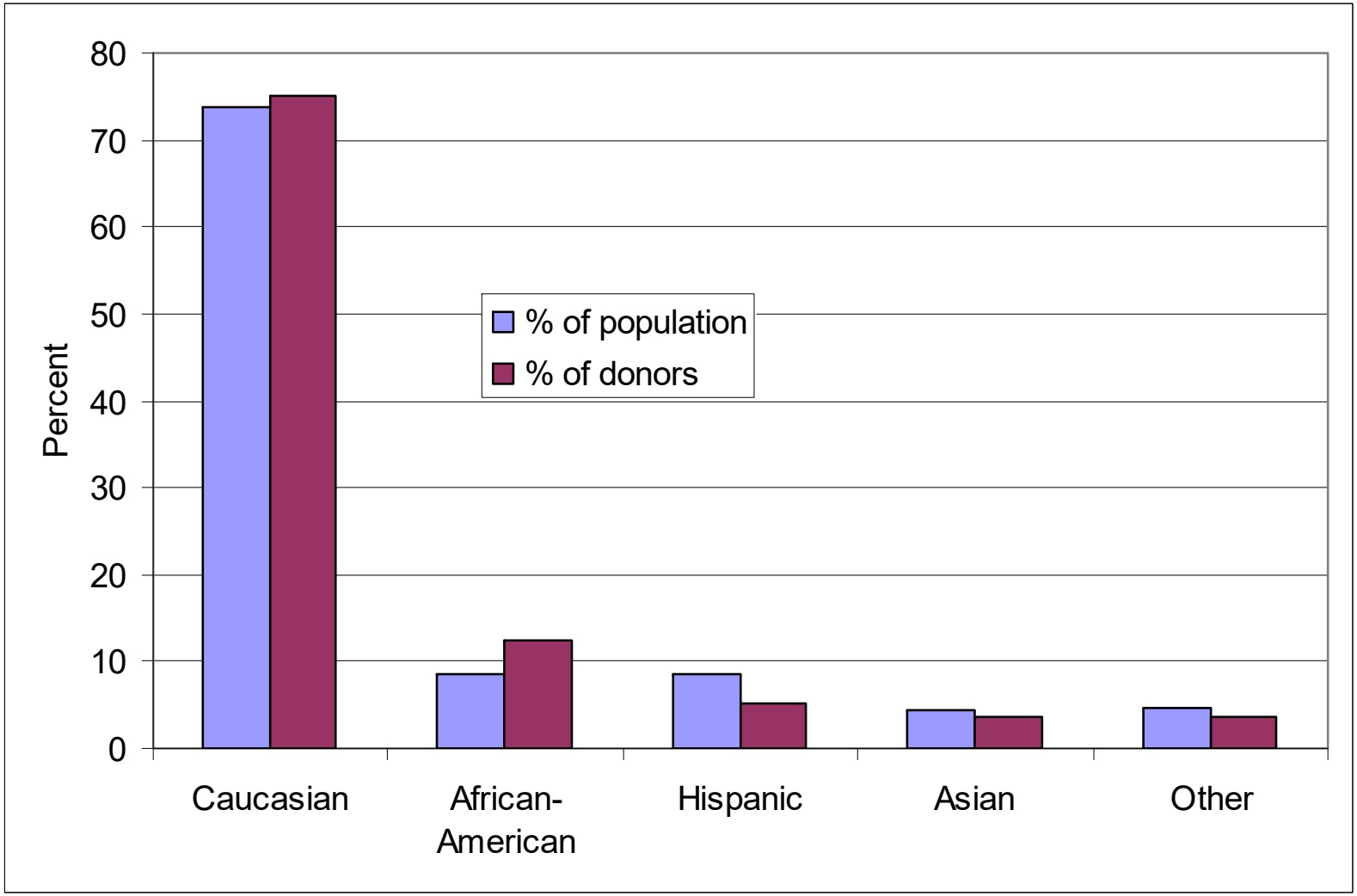
Table 2-1: Blood Type Distribution by Ethnic Group

Ethnic Group	Type O	Type A	Type B	Type AB
US (Caucasian)	45	40	11	4
US (African American)	49	27	20	4
Hawaiian	37	61	2	1
Chinese-Canton	46	23	25	6
Brazilians	47	41	9	3
Germans	41	43	11	5
French	43	47	7	3
Peru (Indians)	100	0	0	0
Russians	33	36	23	8

Source: Blood Book Website [2]

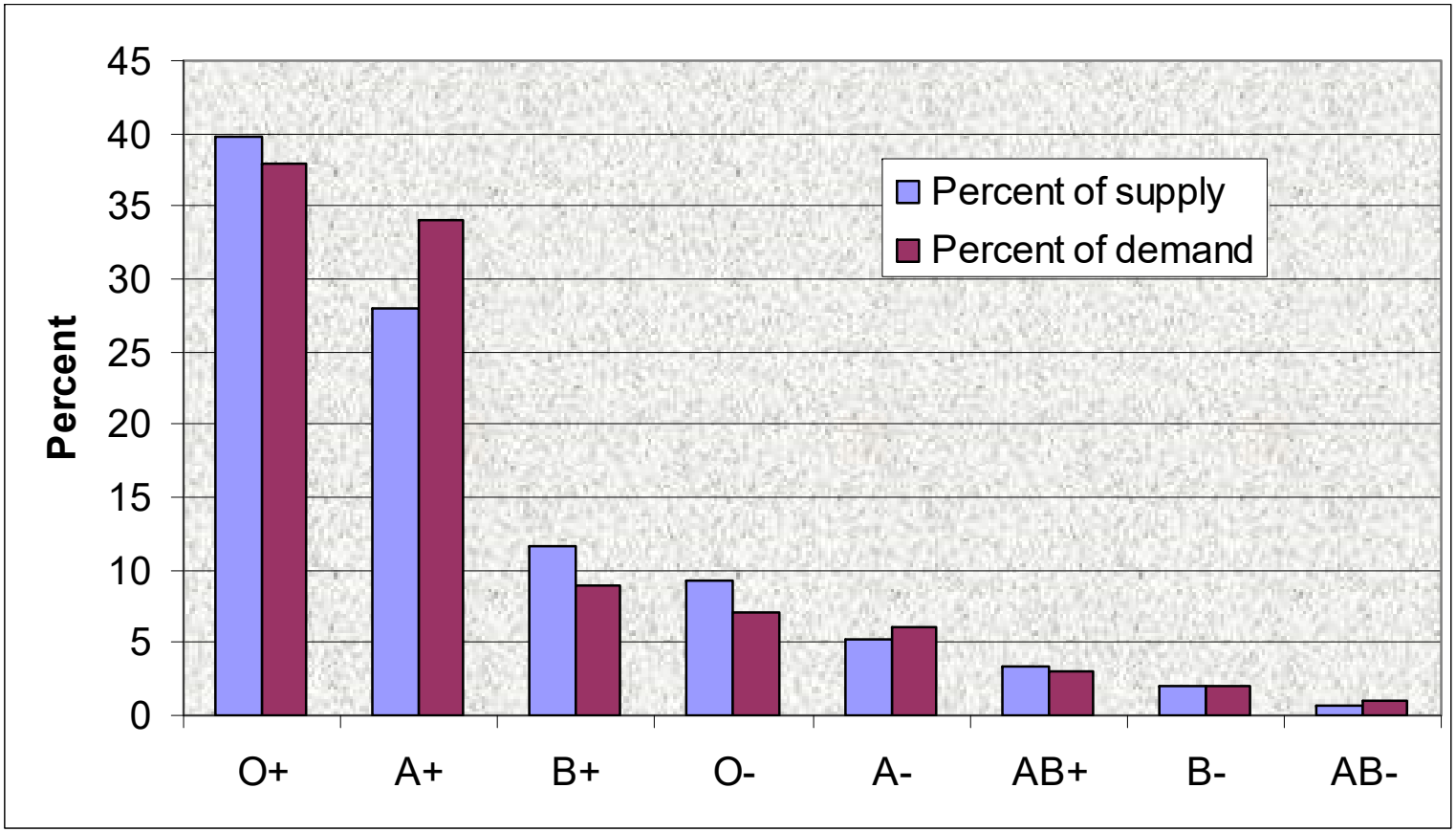
Blood management

- Donations by ethnicity



Blood management

- Supply and demand distributions
  - » Donations
    - Depends on age, race, income and geography



- Supplies
  - » Might assume that donations follow a Poisson distribution with means given by historical donation rates (this is what you might learn in ORF 309).
  - » The Poisson is generally appropriate when describing rates that come from a large population.
  - » Potential errors:
    - Donations may be triggered by mass requests or highly visible events.
    - Capturing correlations between donations because of exogenous sources.
    - Correlations can come in different forms: spatial, temporal, and across ethnicities (??)
  - » Next week we are going to see a model where we assume a Poisson distribution, but where the Poisson is not actually a good fit.

- Demands
  - » Same issues as donations, although the underlying stochastic model would have different elements.
  - » E.g. bursts of demands could be due to:
    - Weather events (snow storms, very hot weather)
    - Major sports events

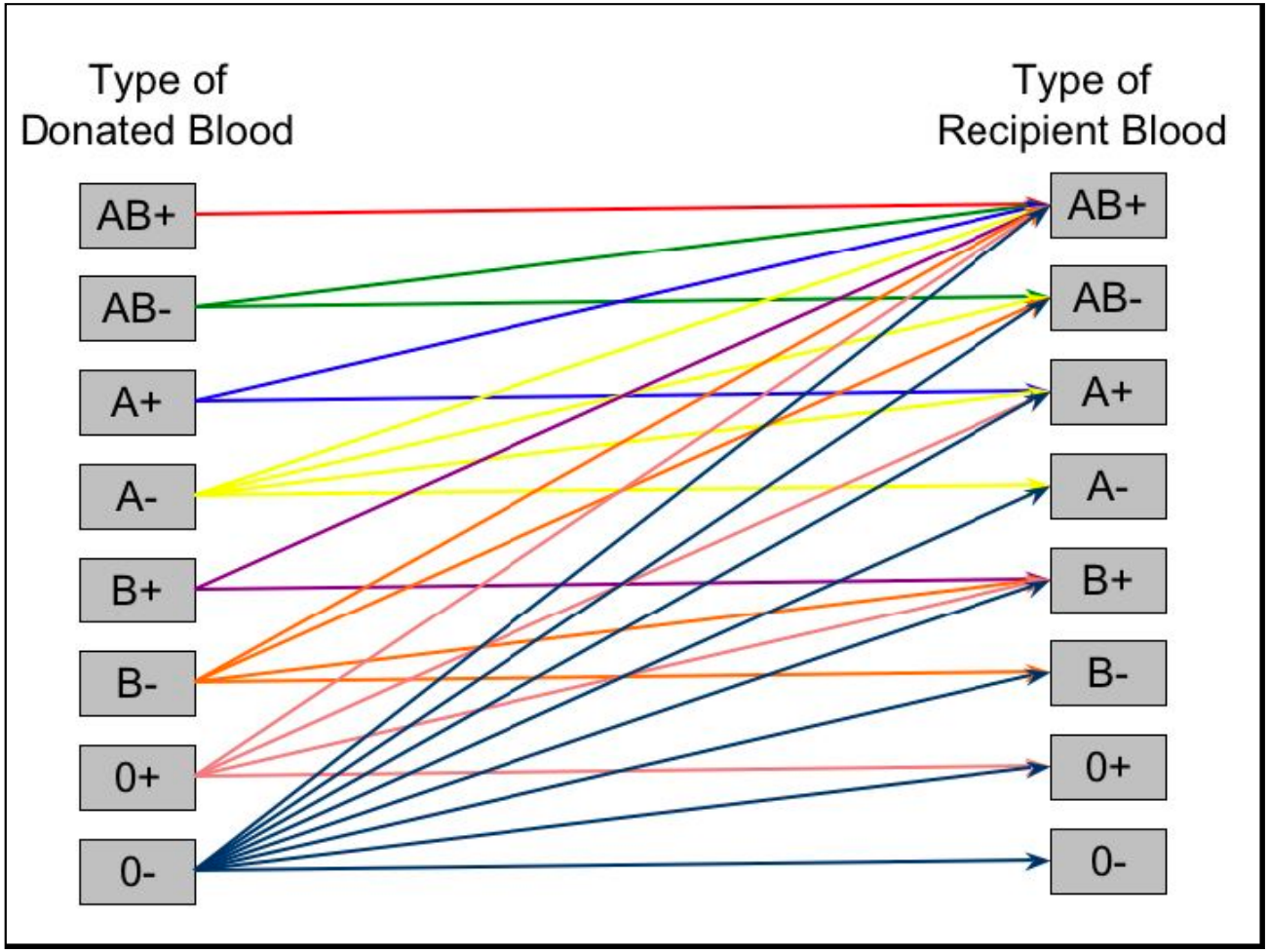
- Notes:
  - » Proper stochastic modeling is key if we wish to focus on the risk of outages.

Designing policies

Myopic policy

Blood management

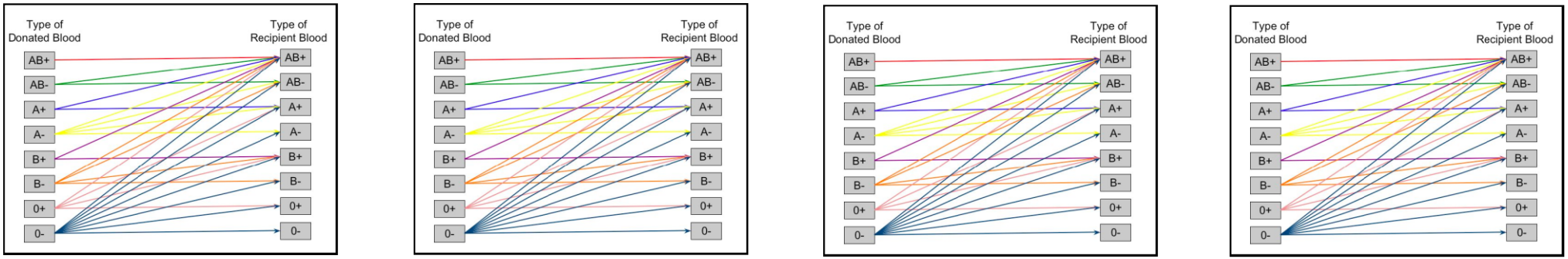
- Managing blood inventories

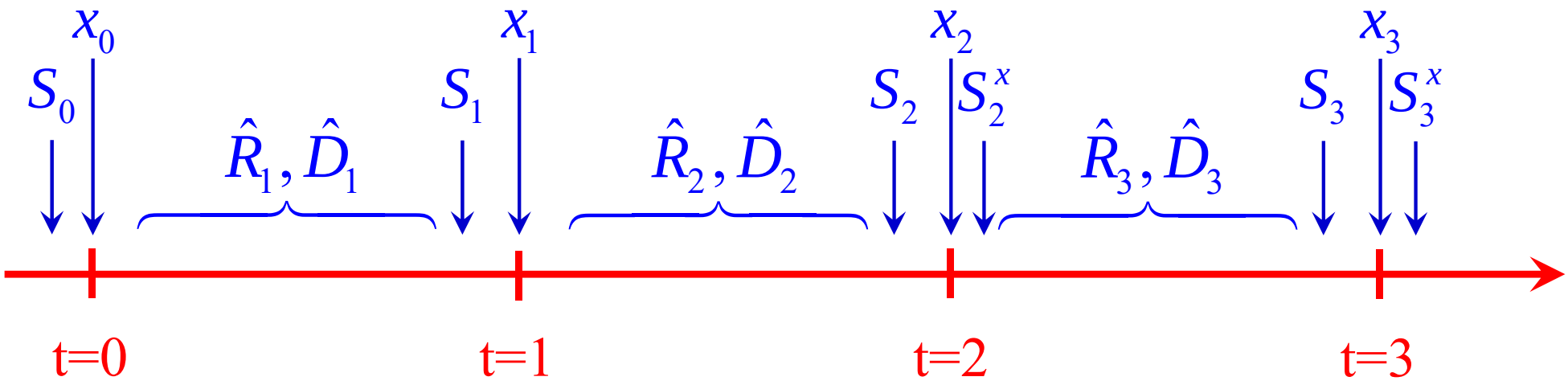


Blood management

- Managing blood inventories over time

Week 0 | Week 1 | Week 2 | Week 3





- Myopic policy
  - » We can use bonuses and penalties to encourage different behaviors.
    - We may wish to try to hold onto O-minus blood for the future since it can be used to meet any need.
    - We may wish to discourage covering elective surgery if we feel we need to hoard blood for urgent surgeries in the future.
  - » Challenges:
    - Tuning these bonuses and penalties can be a real pain!
    - Very tricky when we are trying to optimize across different metrics (e.g. limiting the coverage of elective surgeries, holding onto O-minus).
    - Hard to handle time-dependent policies.

Condition	Description	Value
if $d = d^{\phi}$	Holding	0
if $b_1 = b_1$ when $d \in \mathcal{D}$	No substitution	0
if $b_1 \neq b_1$ when $d \in \mathcal{D}$	Substitution	-10
if $b_1 =$ O- when $d \in \mathcal{D}$	O- substitution	5
if $d_2 =$ Urgent	Filling urgent demand	40
if $d_2 =$ Elective	Filling elective demand	20

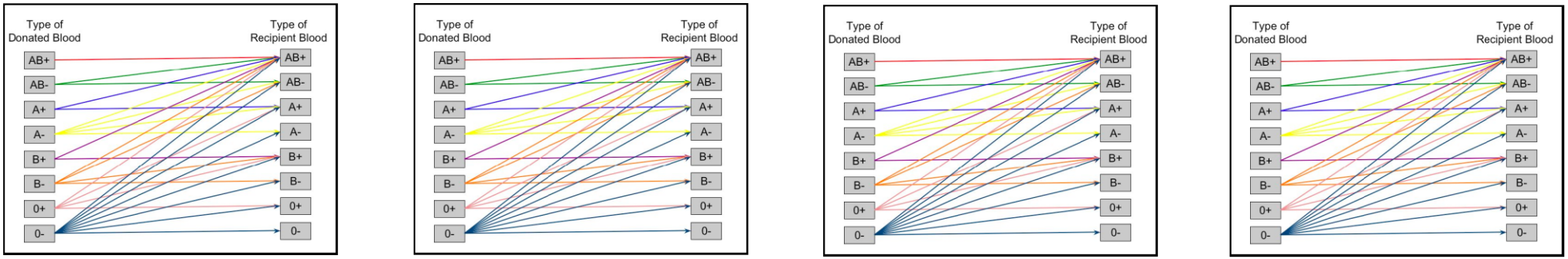
Designing policies

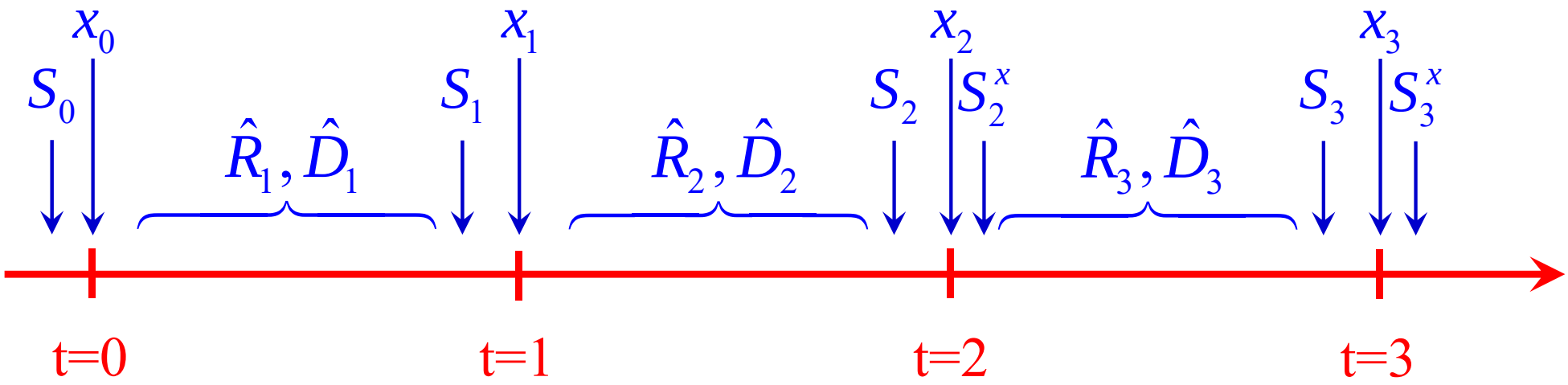
Approximate dynamic programming

Blood management

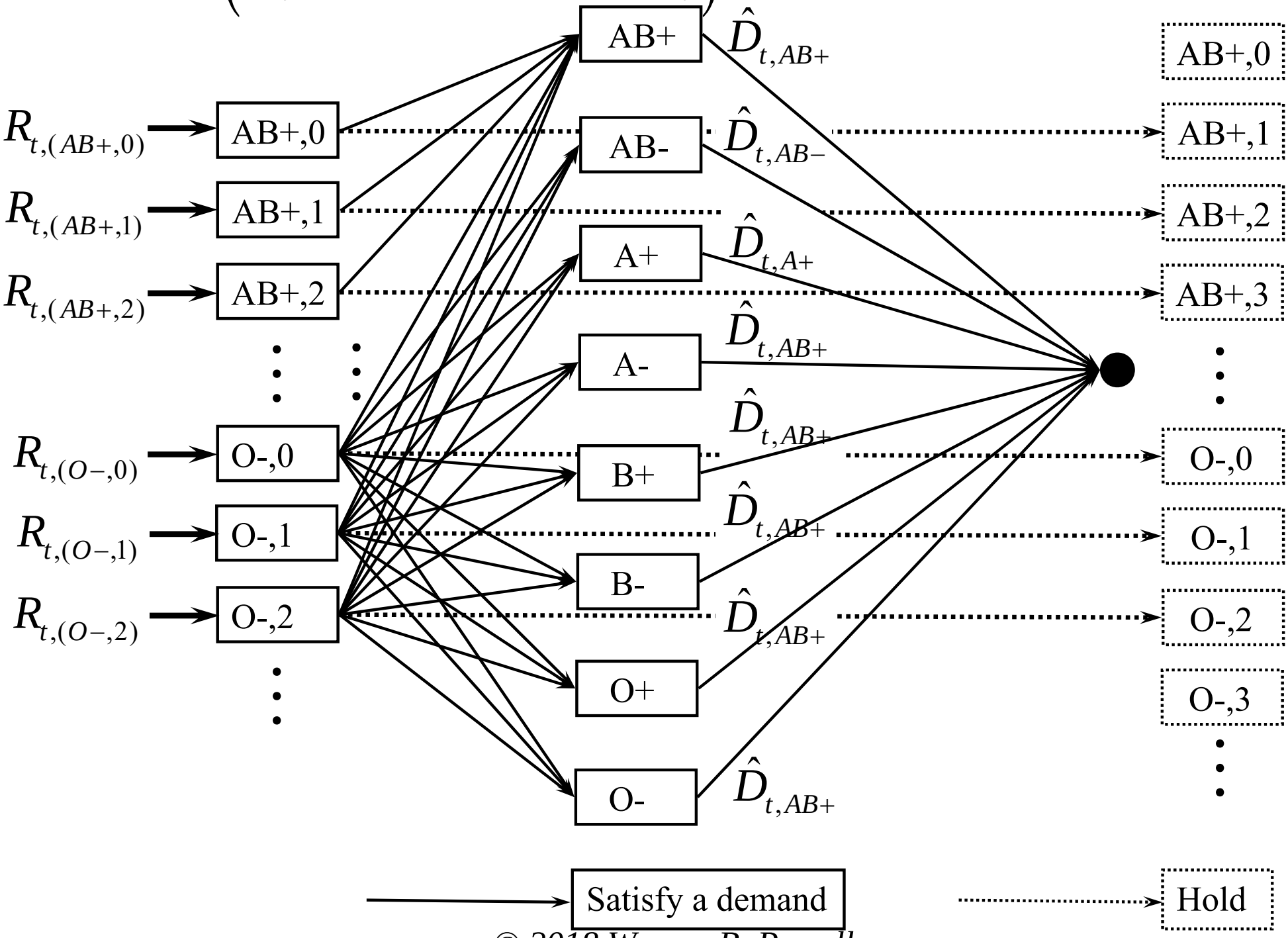
- Managing blood inventories over time

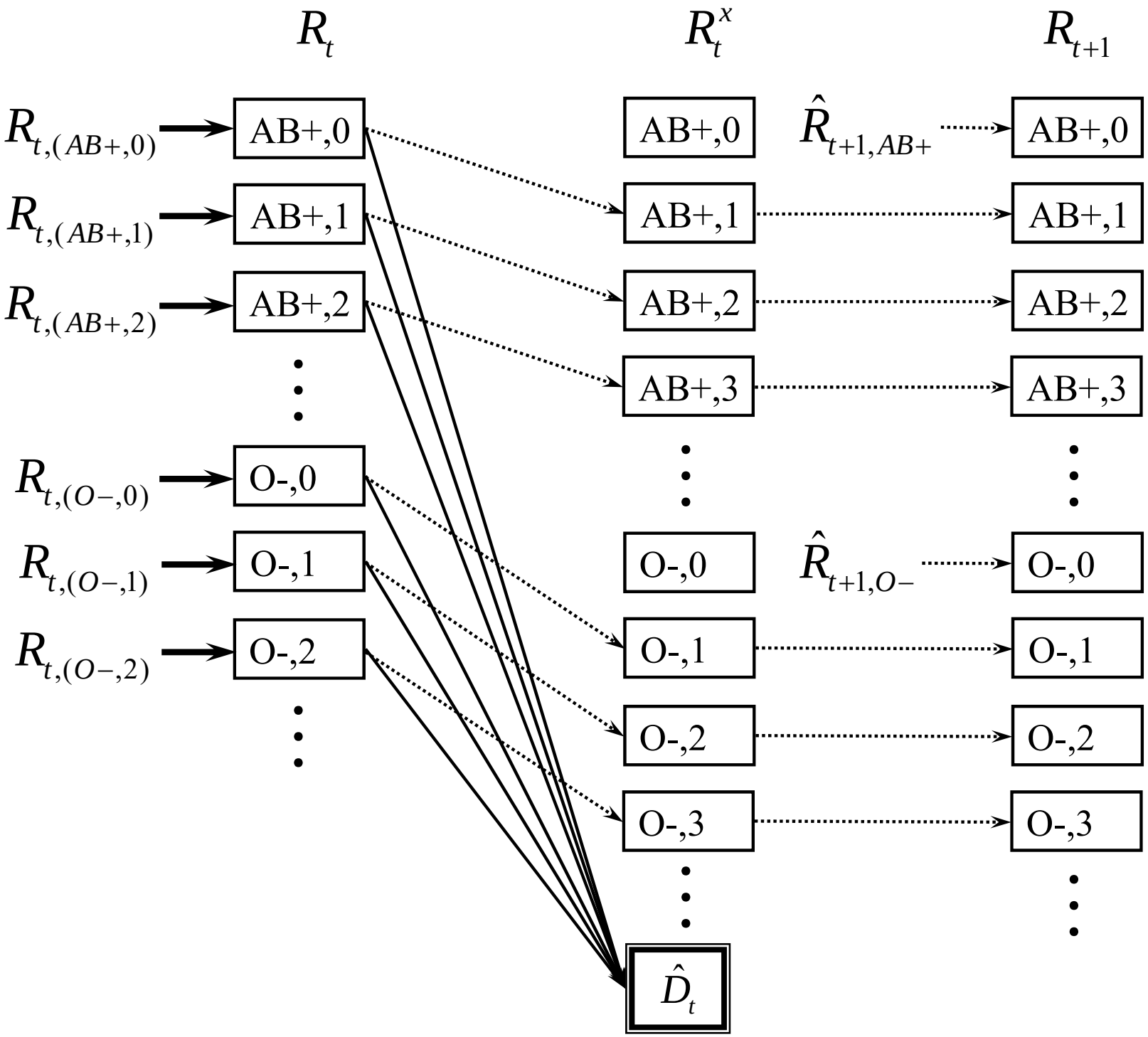
Week 0 Week 1 Week 2 Week 3

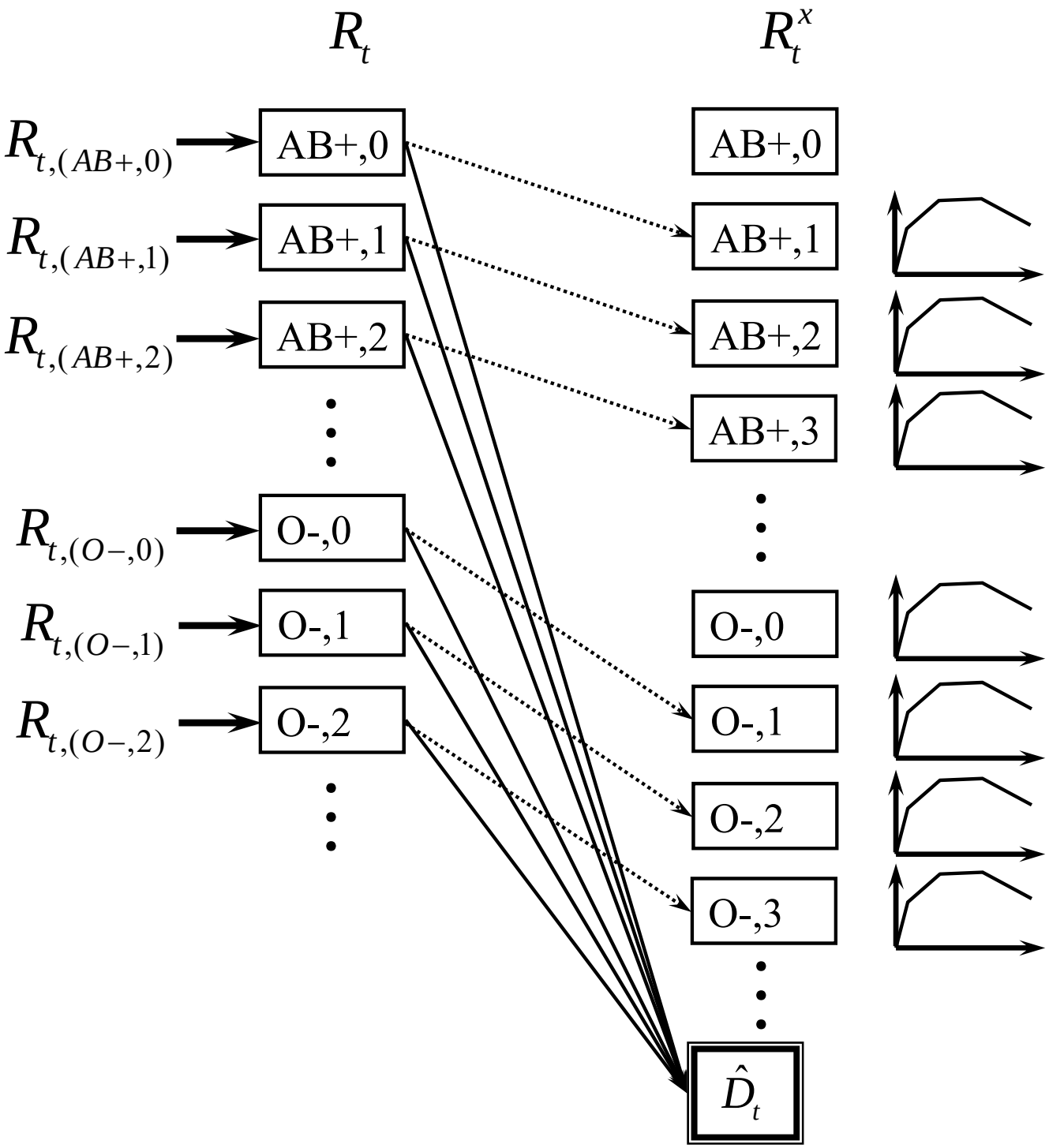


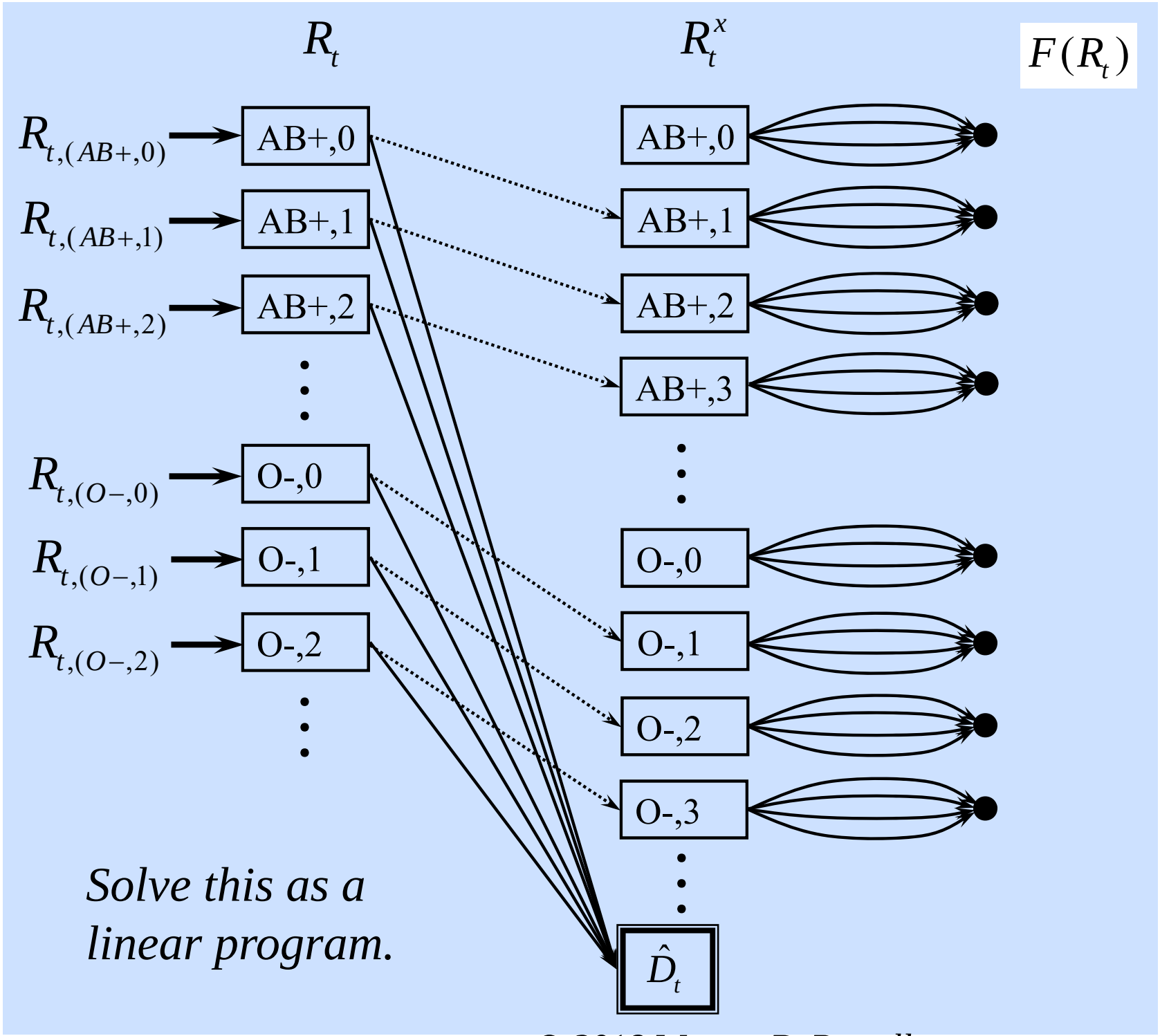


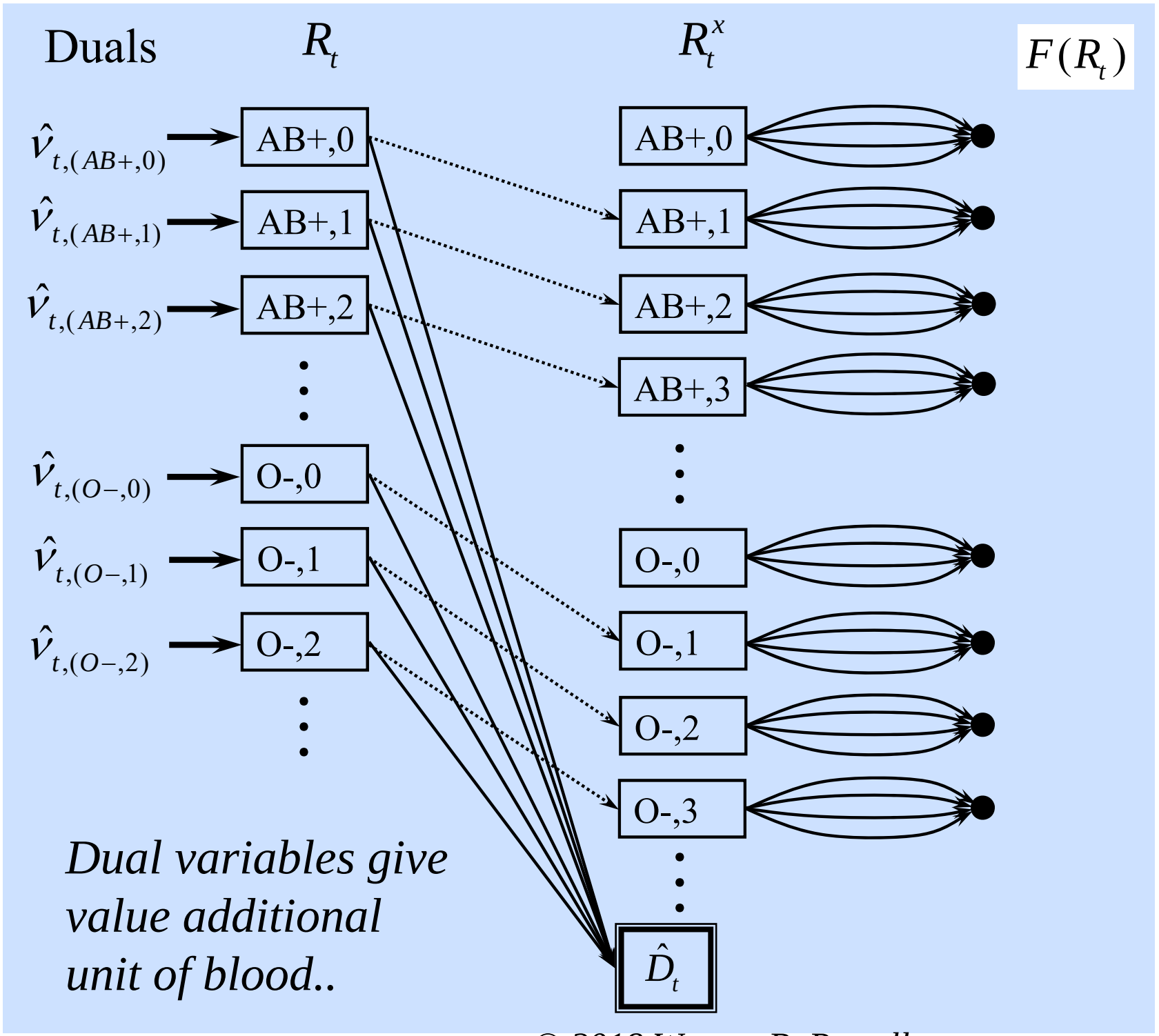
$S_t = \left(R_t \quad , \quad \hat{D}_t \right)$ $\quad R_t^x$





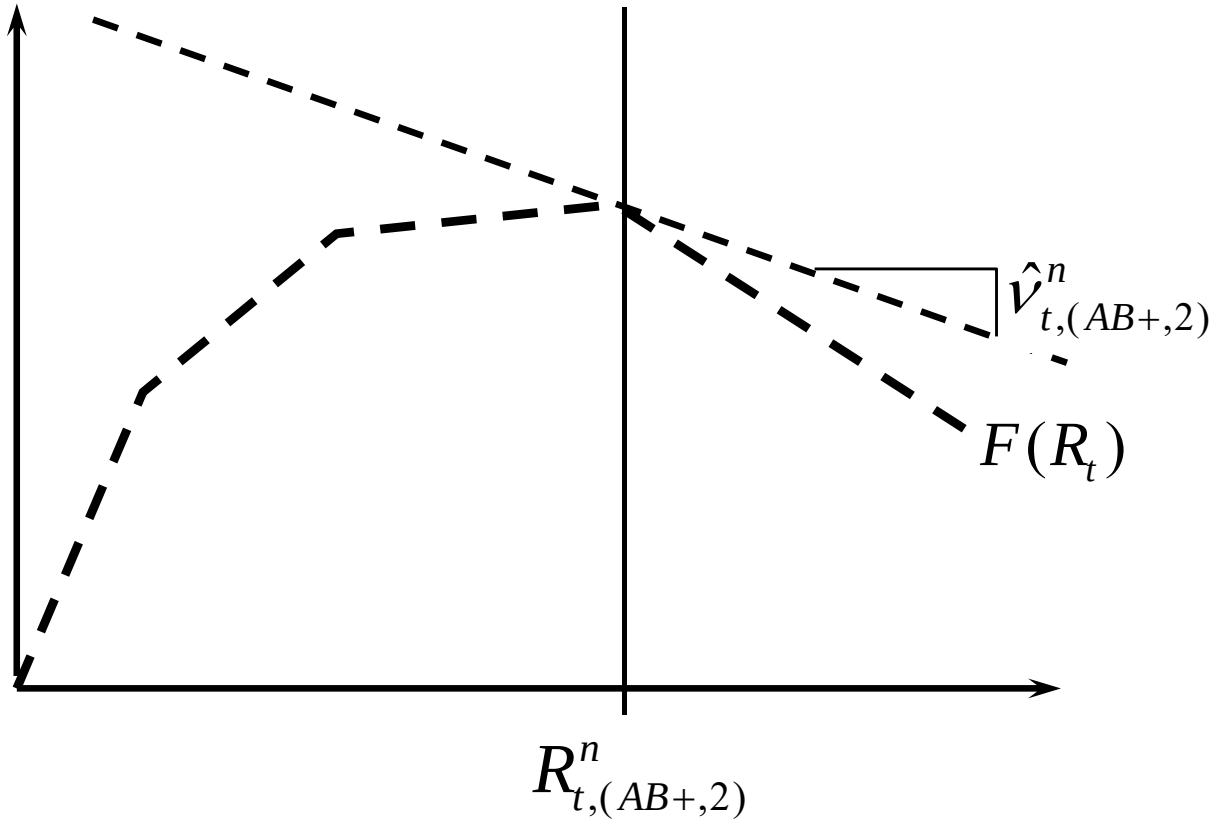






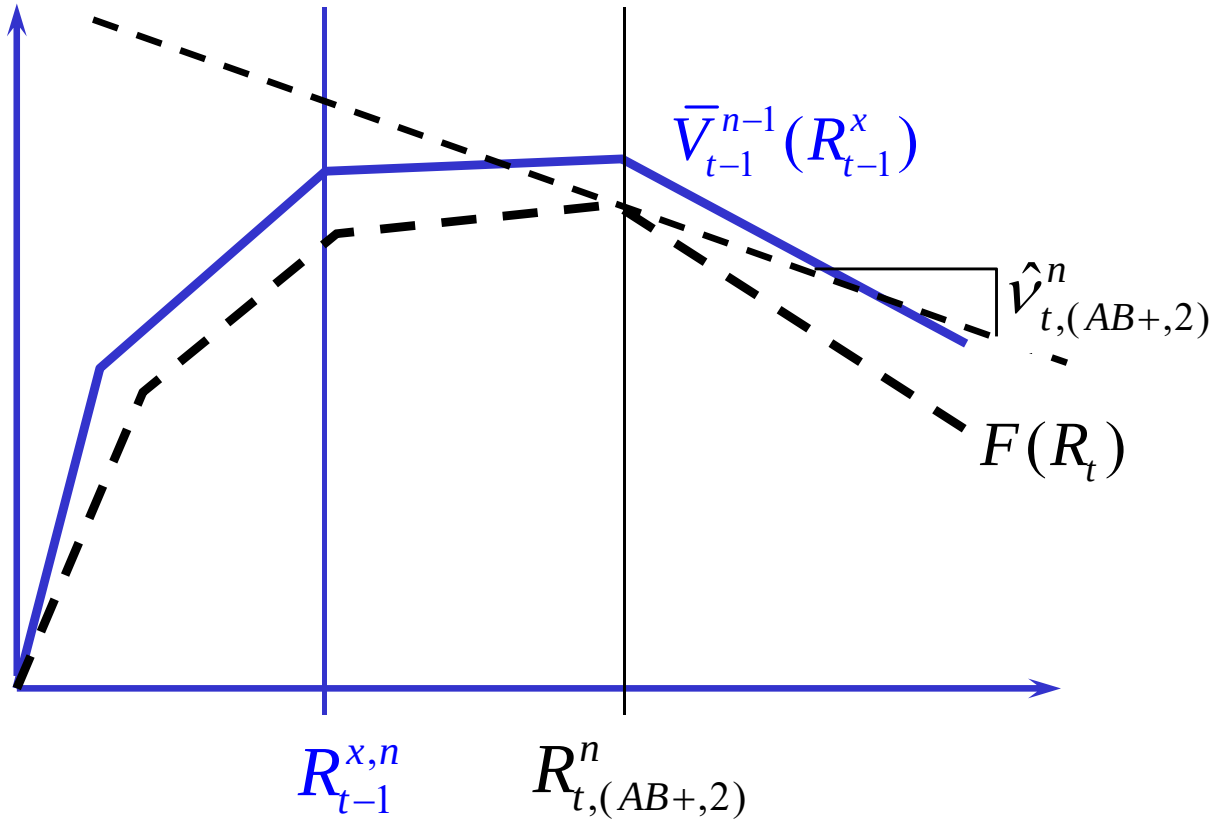
Updating the value function approximation

- Estimate the gradient at R_t^n



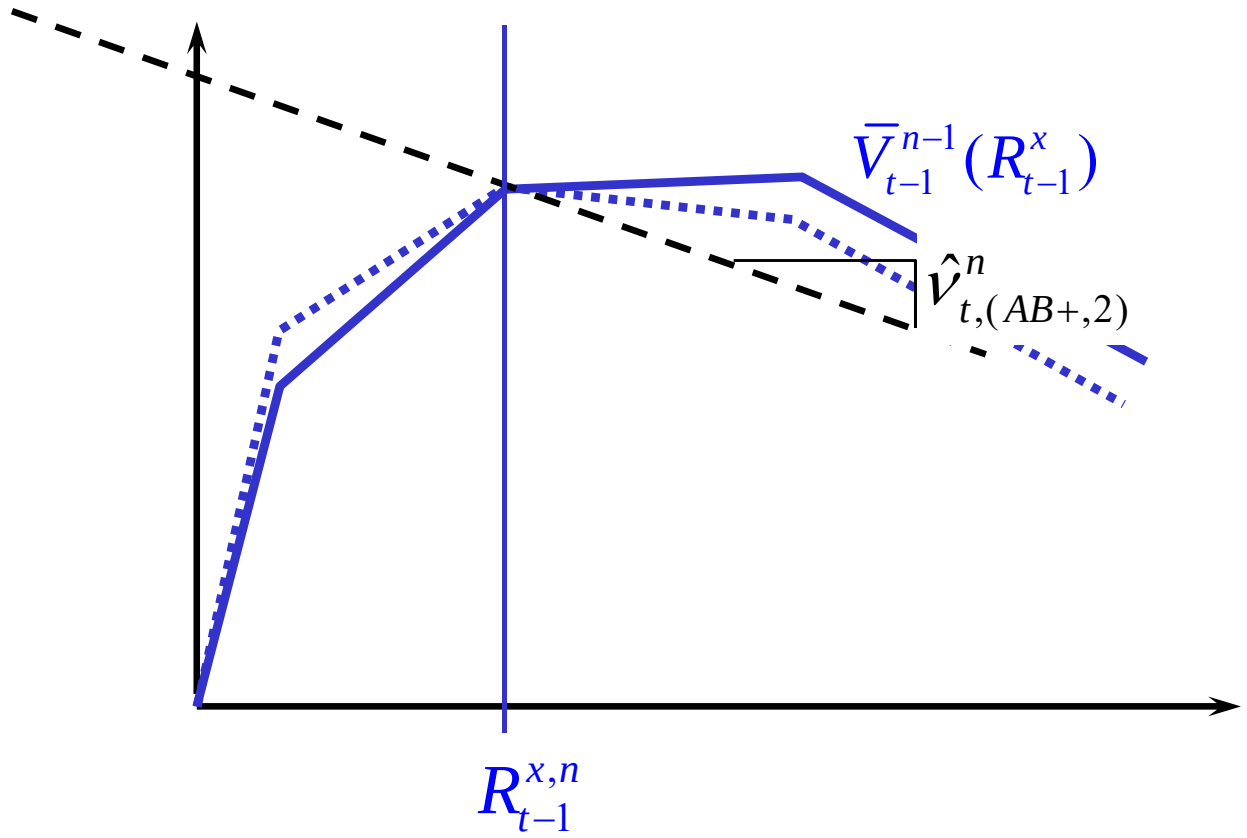
Updating the value function approximation

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



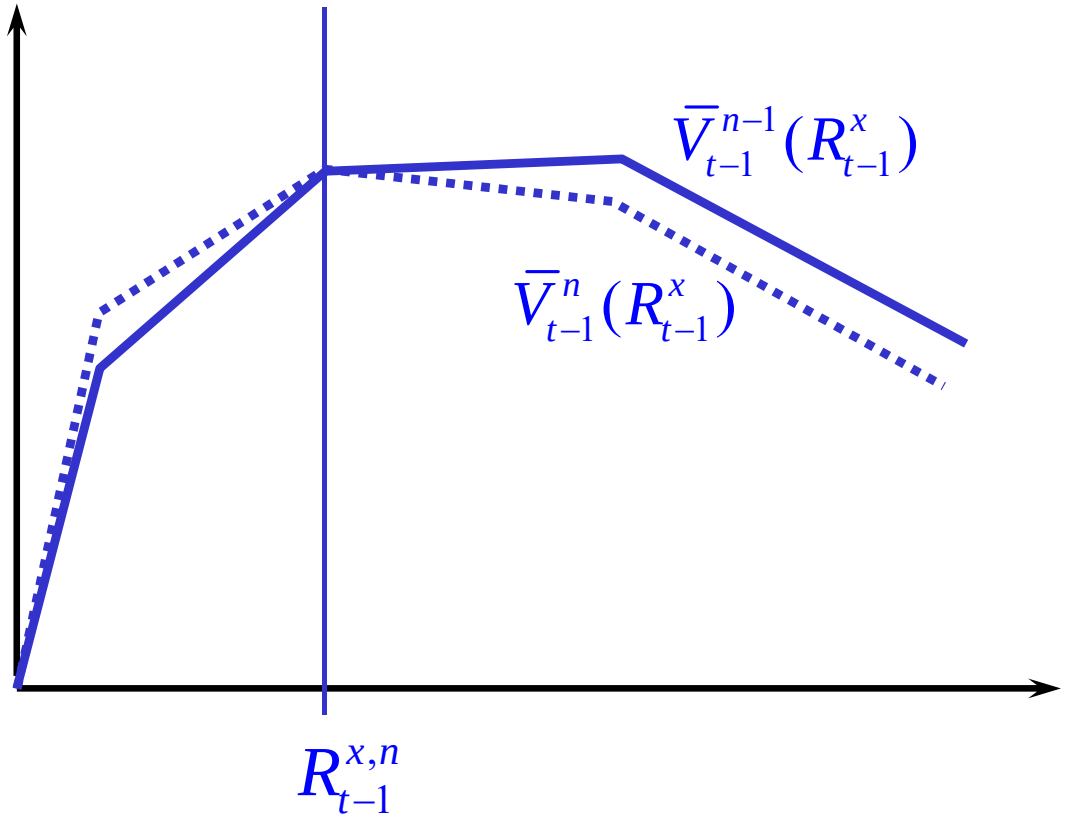
Updating the value function approximation

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



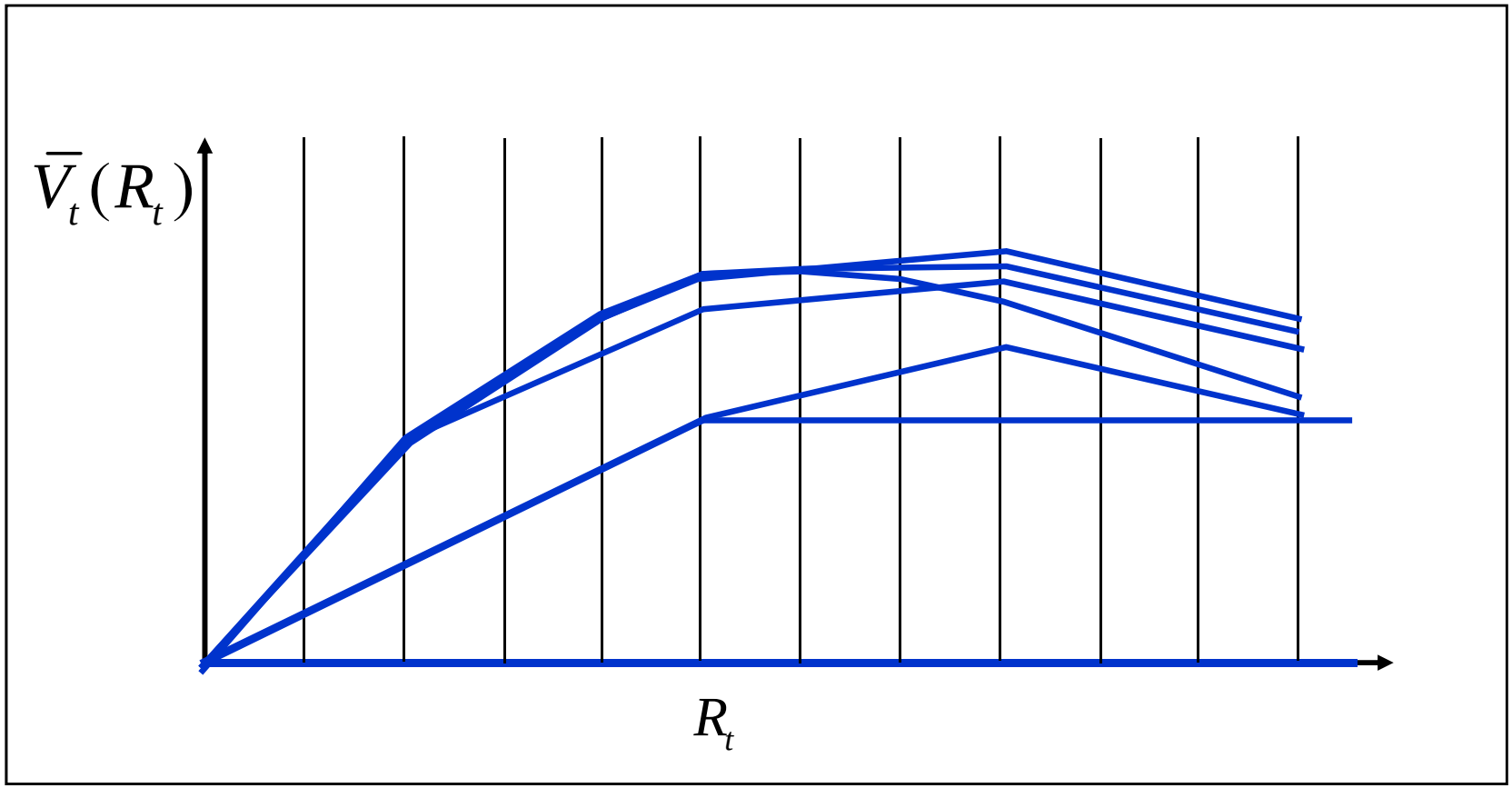
Updating the value function approximation

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



Exploiting concavity

- Derivatives are used to estimate a piecewise linear approximation



Approximate value iteration

Step 1: Start with a pre-decision state S_t^n

Step 2: Solve the deterministic optimization using an approximate value function:

$$\max_x \left(C_t(S_t^n, x_t) + \bar{V}_t^{n-1}(S^{M,x}(S_t^n, x_t)) \right)$$

to obtain x_t^n and dual variables $\left(\hat{v}_{ti}^n\right)$.

Deterministic optimization

Step 3: Update the value function approximation

$$\bar{V}_{t-1}^n(S_{t-1}^{x,n}) = (1-\alpha_{n-1})\bar{V}_{t-1}^{n-1}(S_{t-1}^{x,n}) + \alpha_{n-1}\hat{v}_t^n$$

Recursive statistics

Step 4: Obtain Monte Carlo sample of $W_t(\omega^n)$ and compute the next pre-decision state:

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

Simulation

Step 5: Return to step 1.

Approximate value iteration

Step 1: Start with a pre-decision state S_t^n

Step 2: Solve the deterministic optimization using an approximate value function:

$$\max_x \left(C_t(S_t^n, x_t) + \bar{V}_t^{n-1}(S^{M,x}(S_t^n, x_t)) \right)$$

to obtain x_t^n and dual variables $\left(\hat{v}_{ti}^n\right)$.

Deterministic optimization

Approximate value iteration

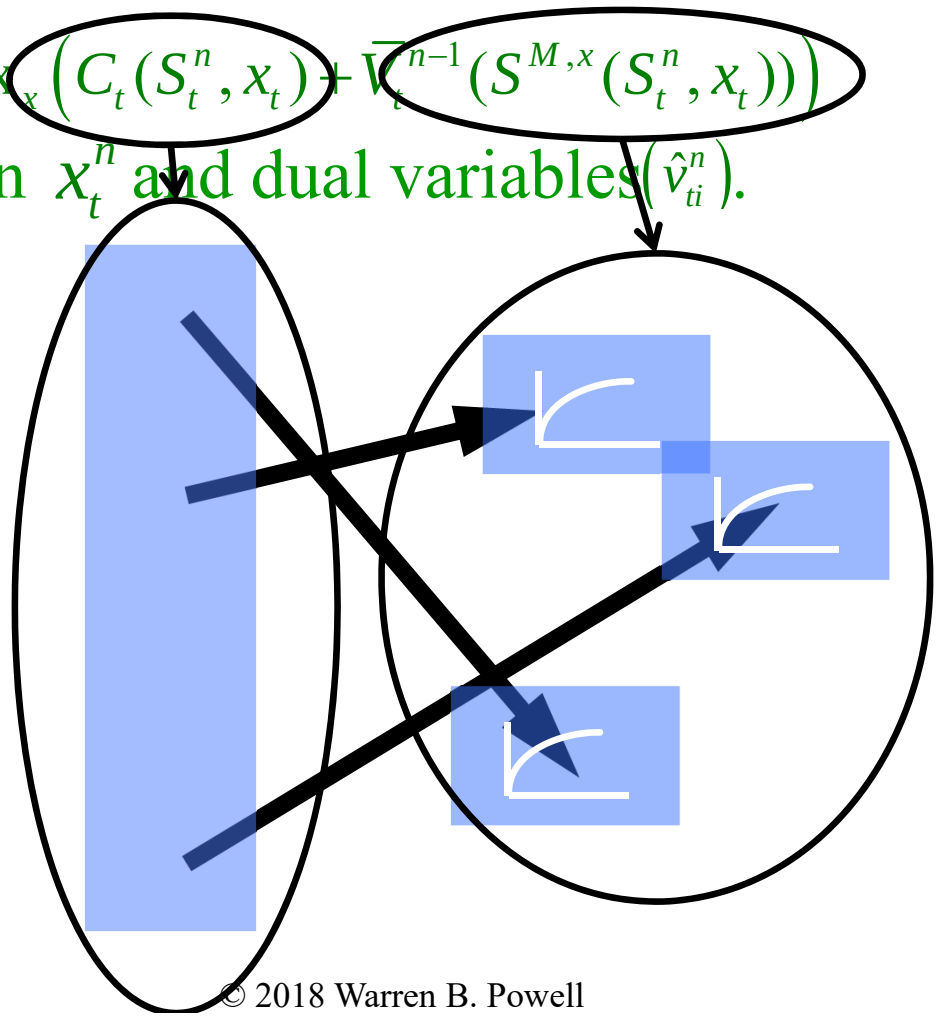
Step 1: Start with a pre-decision state S_t^n

Step 2: Solve the deterministic optimization using an approximate value function:

$$\max_x \left(C_t(S_t^n, x_t) + \overline{V}_t^{n-1}(S^{M,x}(S_t^n, x_t)) \right)$$

to obtain x_t^n and dual variables $\left(\hat{v}_{ti}^n\right)$.

Deterministic optimization



Approximate value iteration

Step 1: Start with a pre-decision state S_t^n

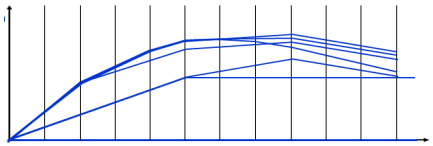
Step 2: Solve the deterministic optimization using an approximate value function:

$$\max_x \left(C_t(S_t^n, x_t) + \bar{V}_t^{n-1}(S^{M,x}(S_t^n, x_t)) \right)$$

to obtain x_t^n and dual variables $\left(\hat{v}_{ti}^n\right)$.

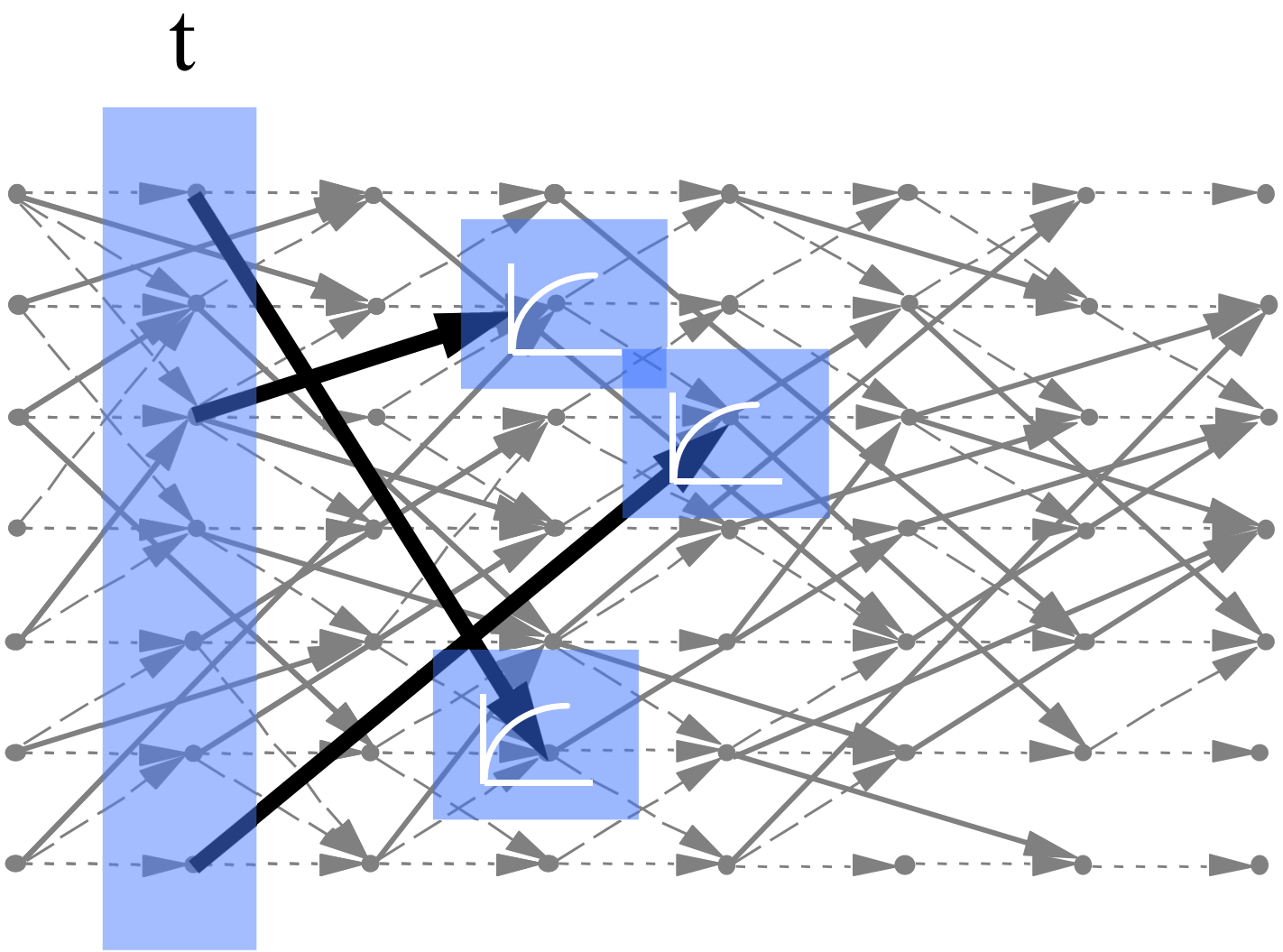
Deterministic optimization

Step 3: Update the value function approximation

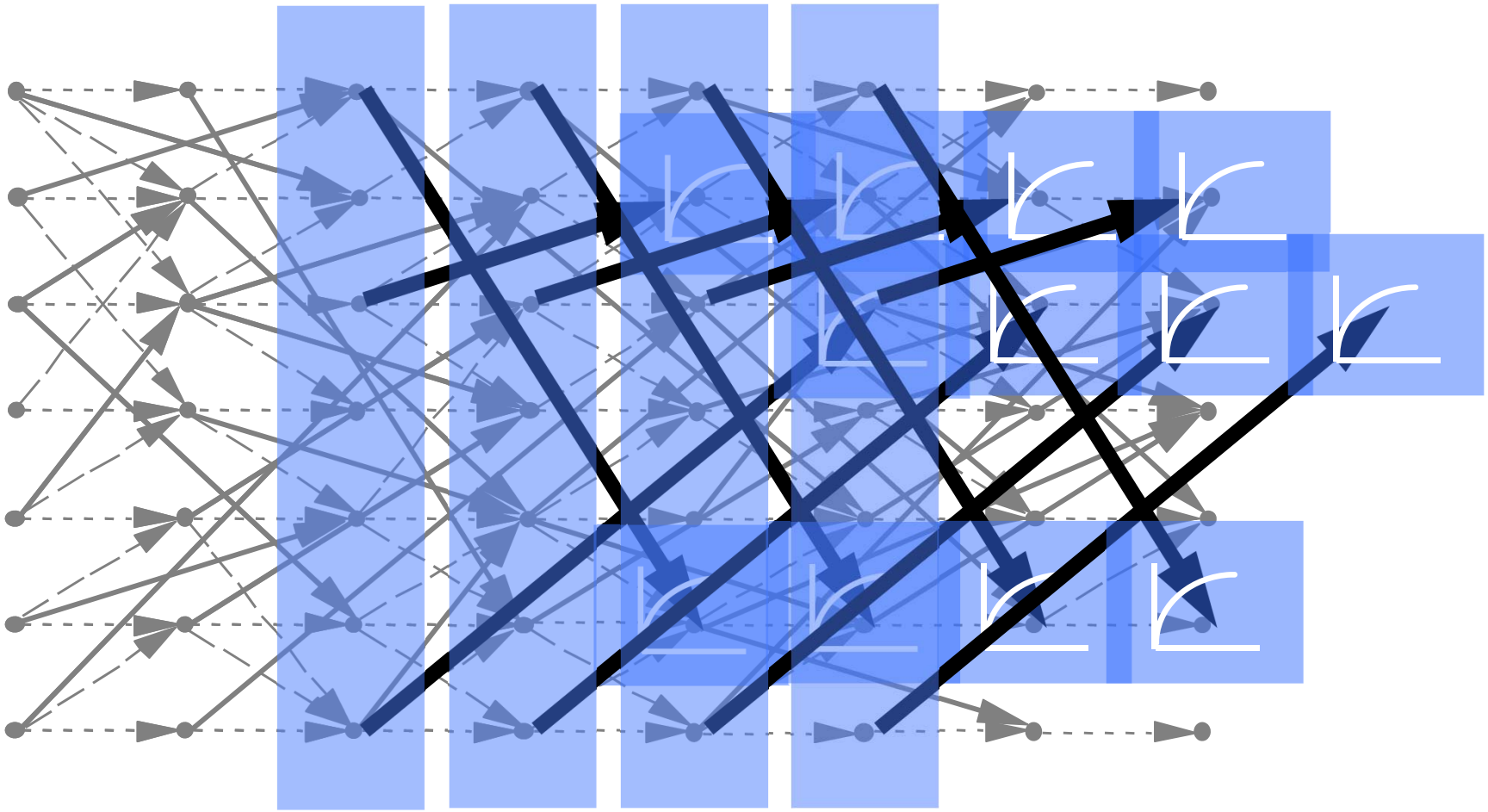


Recursive statistics

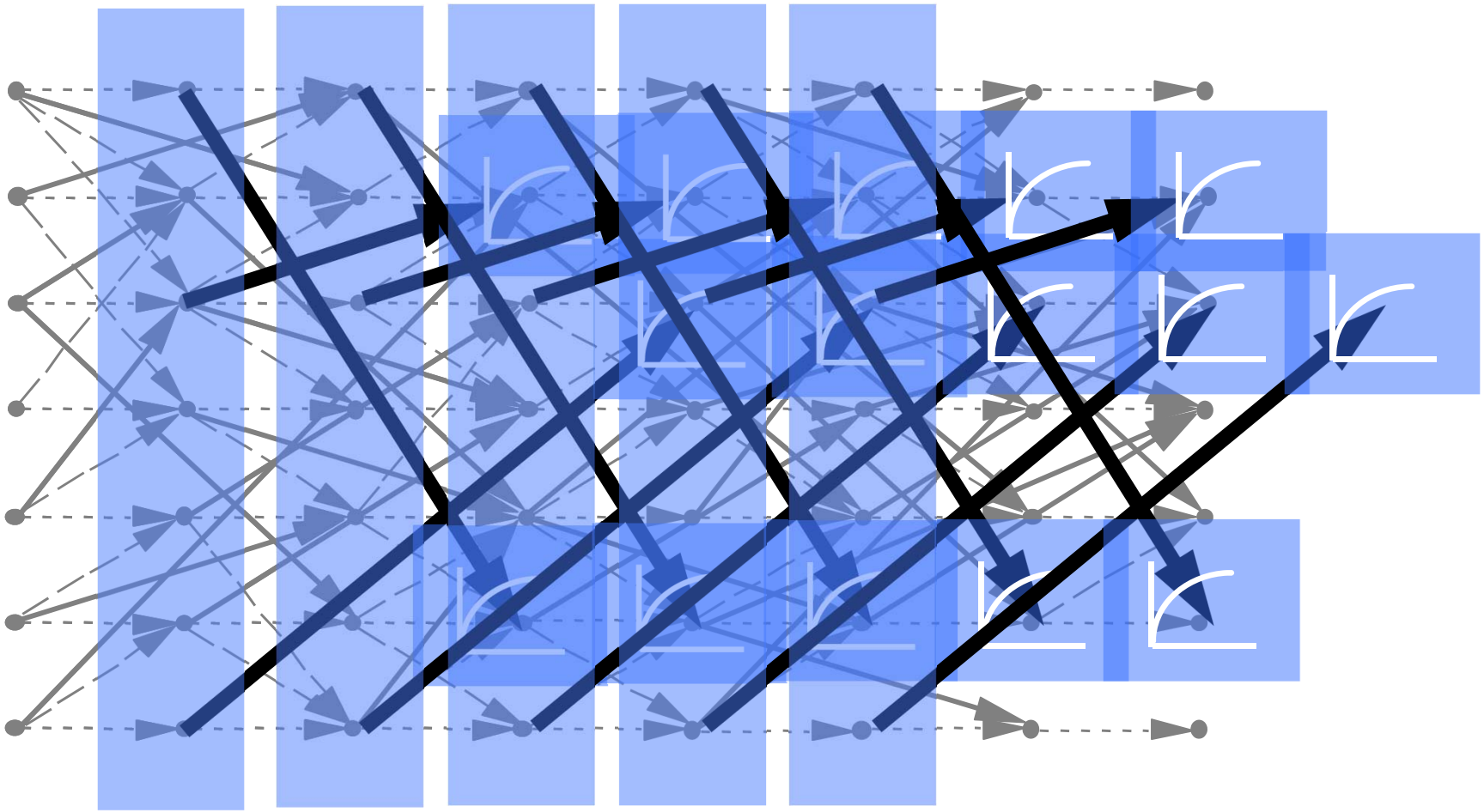
Iterative learning



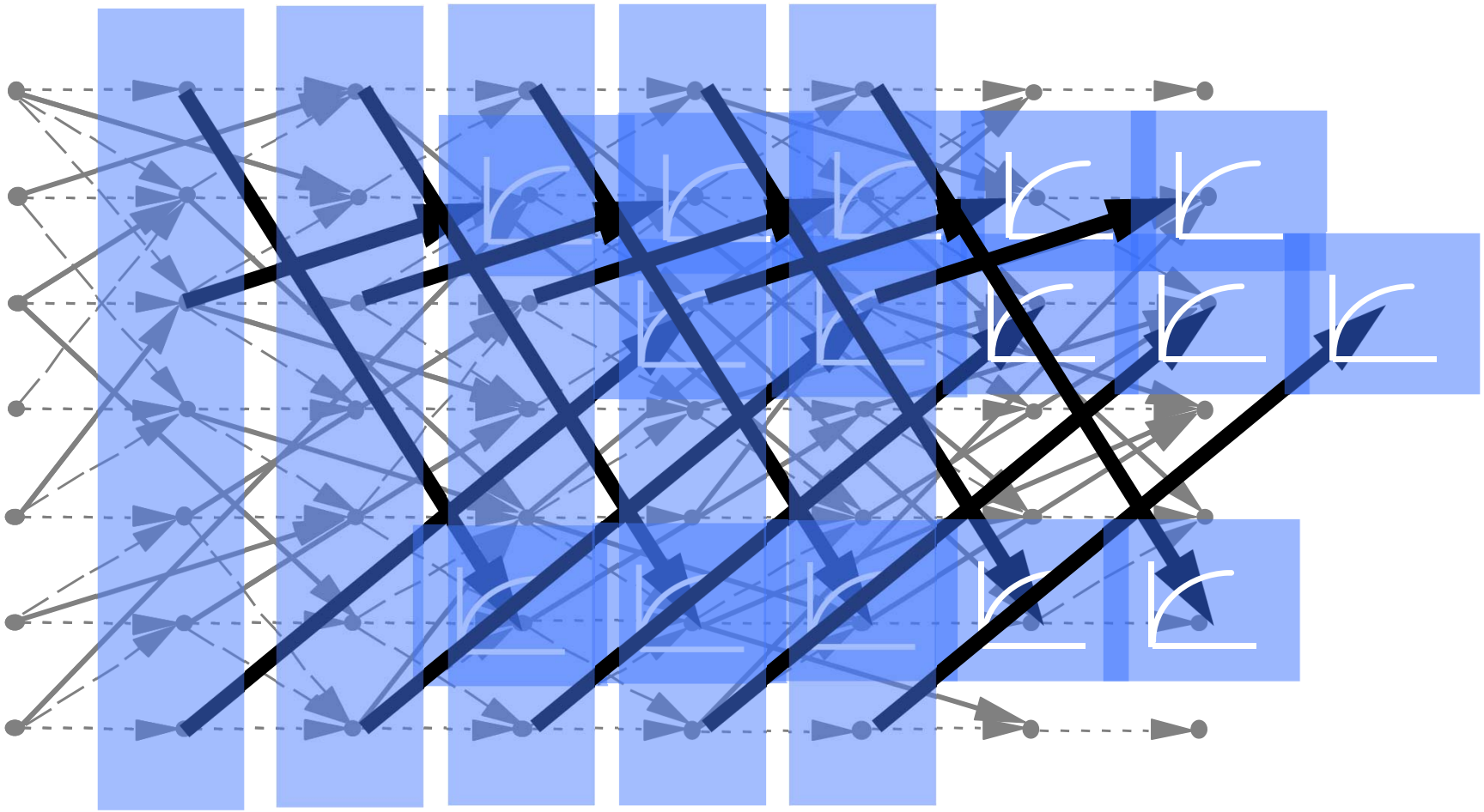
Iterative learning



Iterative learning

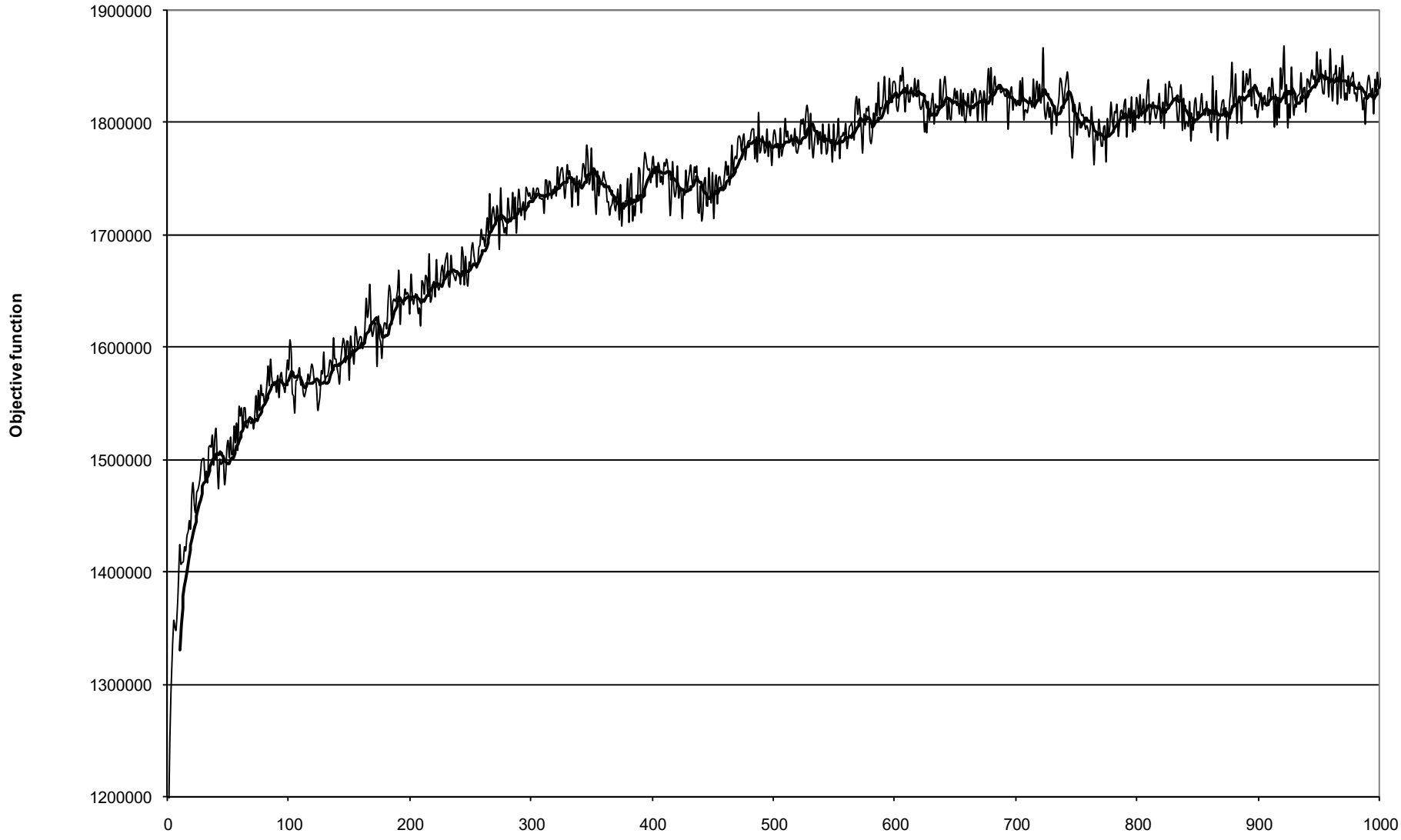


Iterative learning



Approximate dynamic programming

- … a typical performance graph.



Designing policies

Numerical experiments for homework

Week 10 - Wednesday

Autonomous fleets of Evs

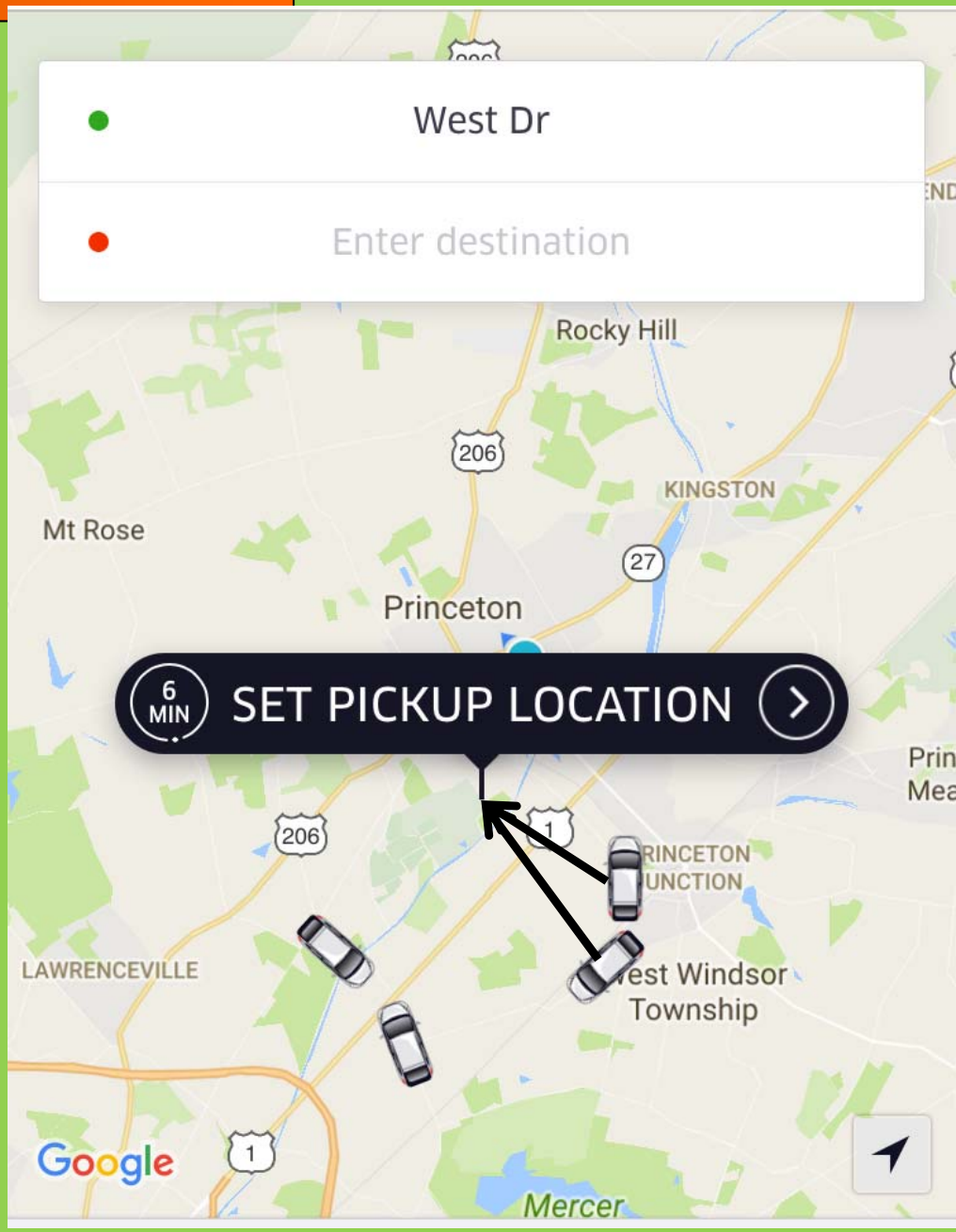
(short presentation)

Autonomous fleets of EVs

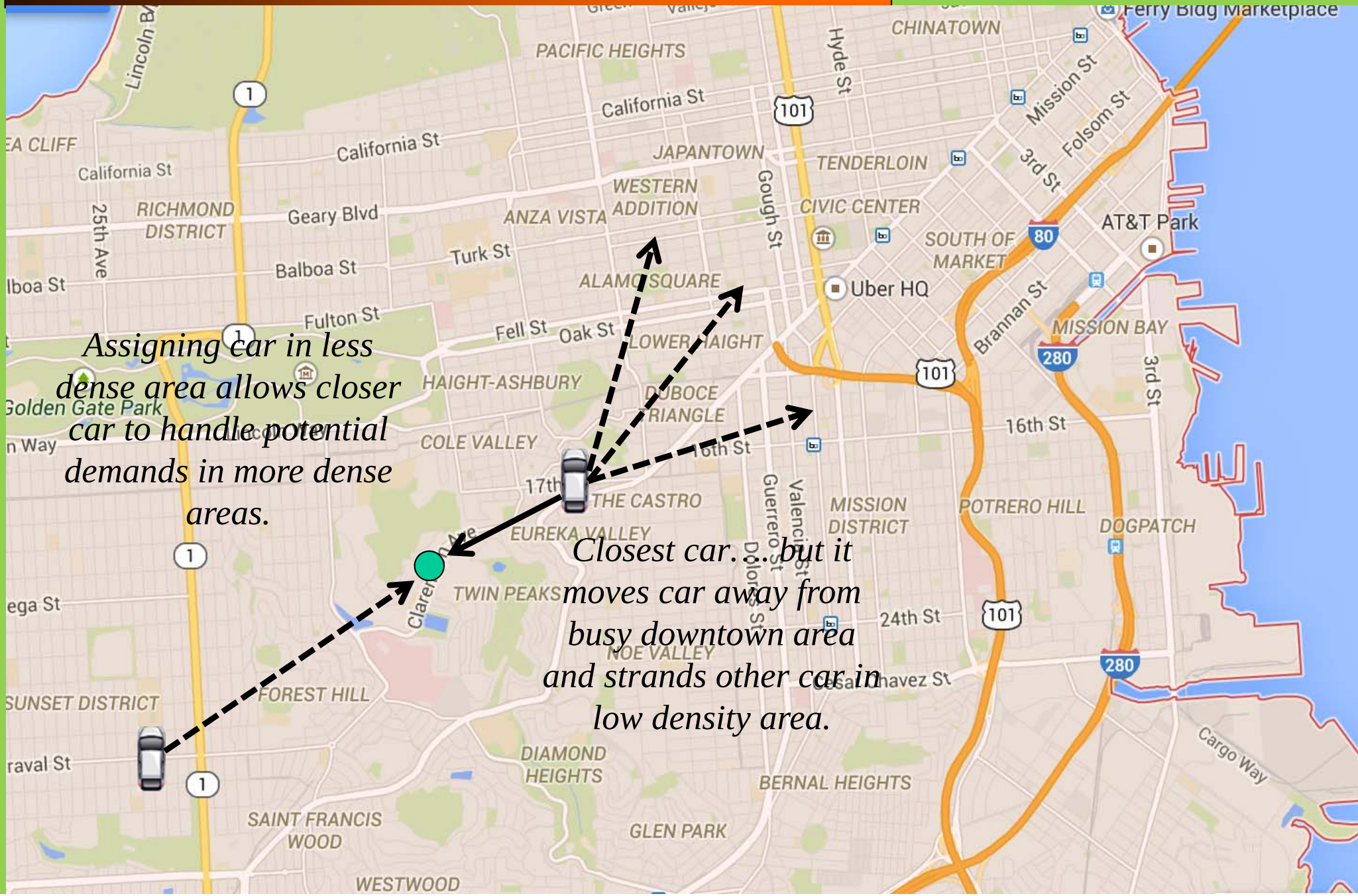
… if time allows.

Real-time logistics

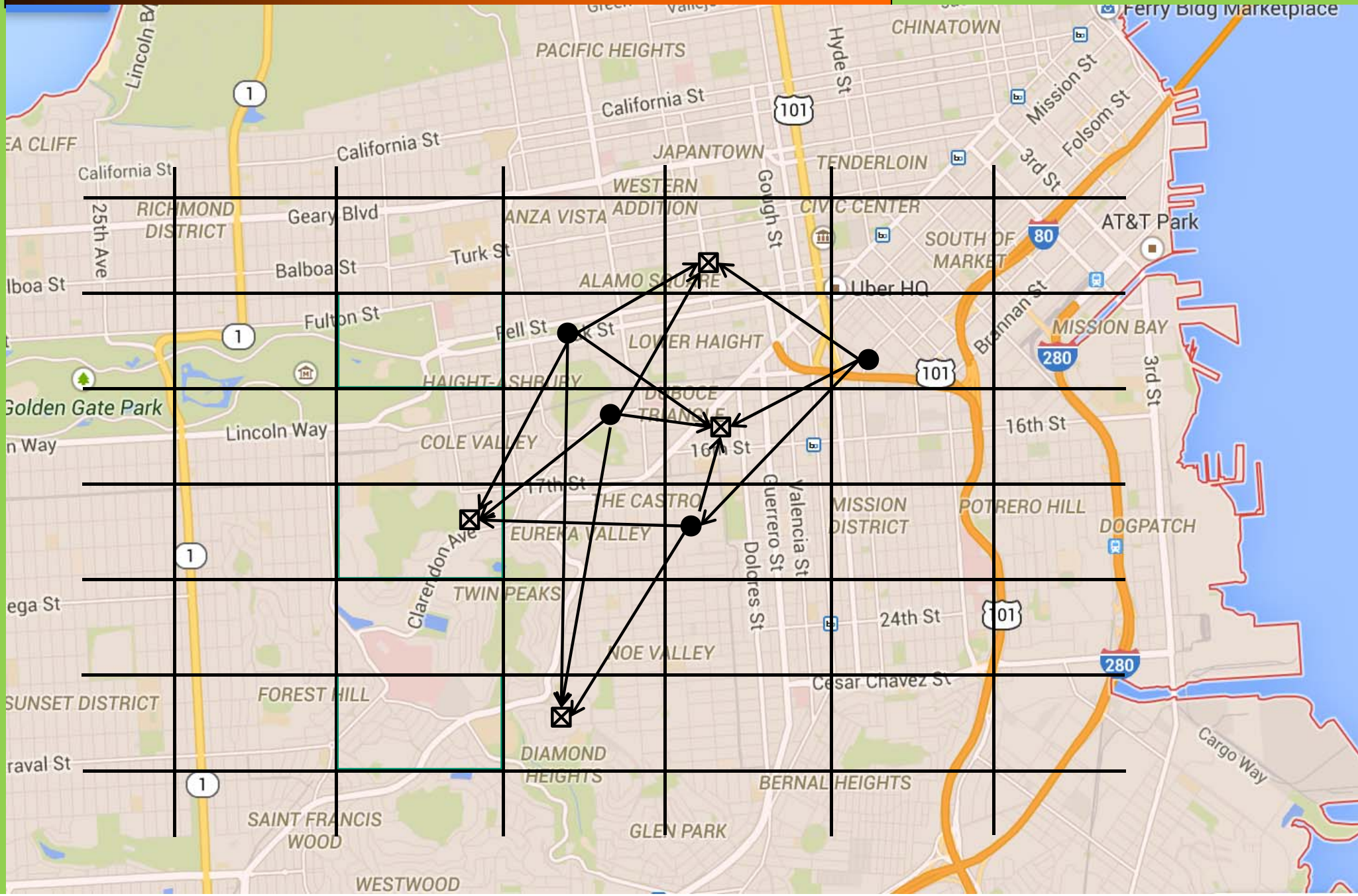
- Current Uber logic:
  - » Show nearest 8 drivers.
  - » Contact closest driver to confirm assignment.
  - » If driver does not confirm, contact second closest driver.
- Limitations:
  - » Ignores potential future opportunities for each driver.



Effect of Current Decision on the Future



From Local to Global Assignment

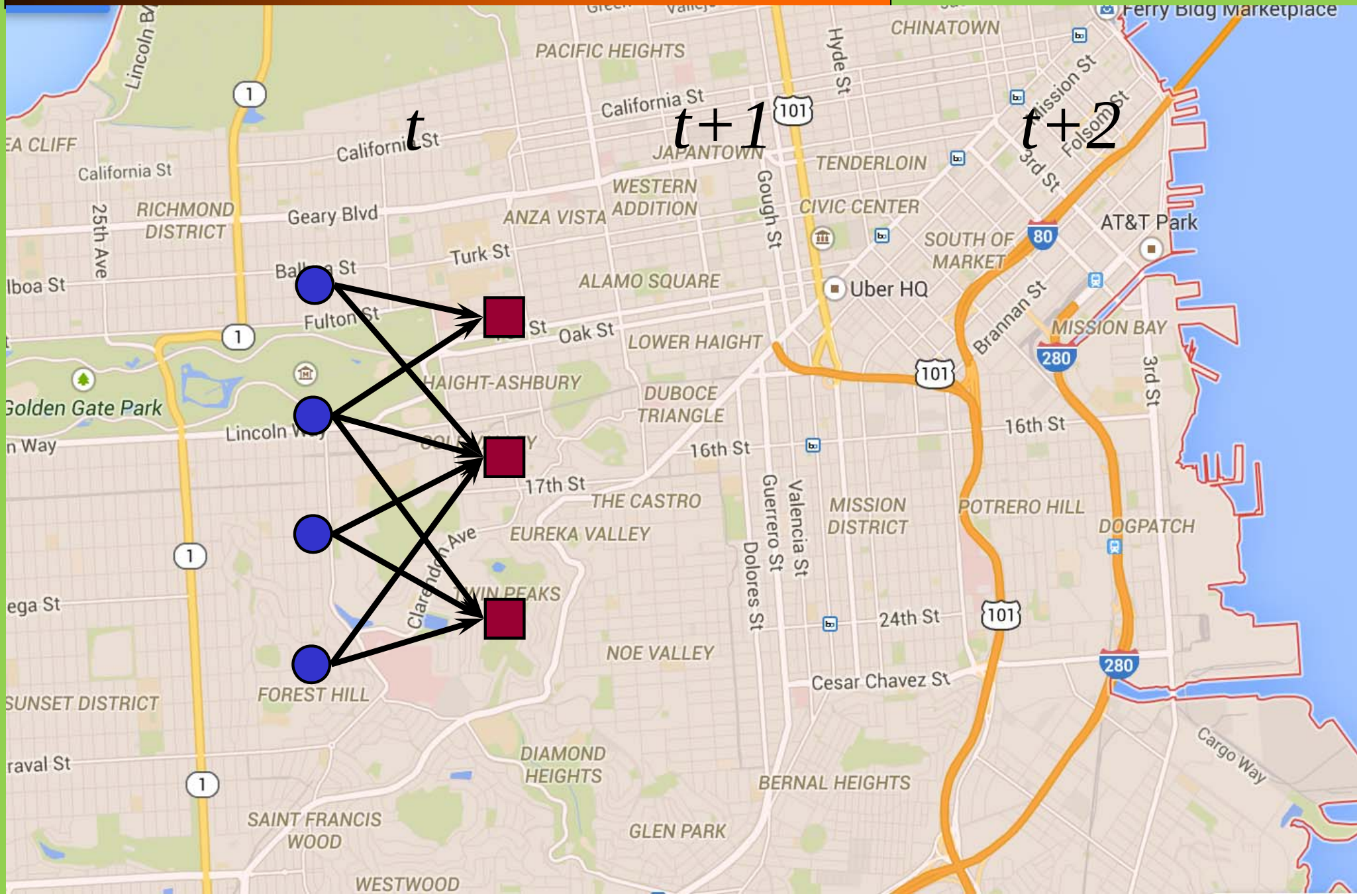


Decisions in Driverless Fleets of EVs

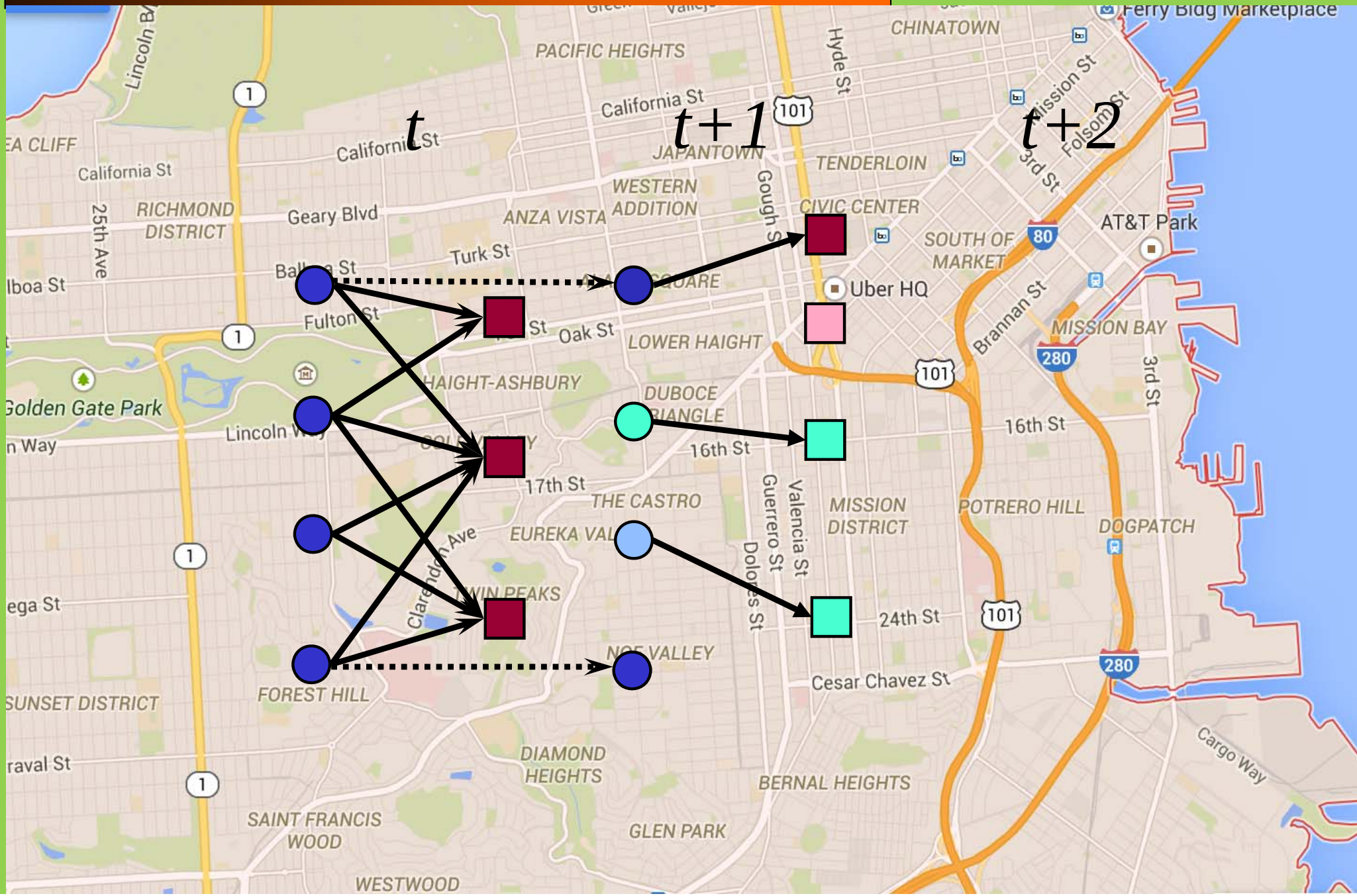
- The central operator should think about:
  - Should it accept this trip?
  - What is the best car to assign to the trip considering
    - The type of car
    - The charge level of the battery

» If it doesn't assign a car to a trip, should it:
  - Sit where it is?
  - Reposition to a better location?
  - Recharge the battery?
  - Move to a parking facility?

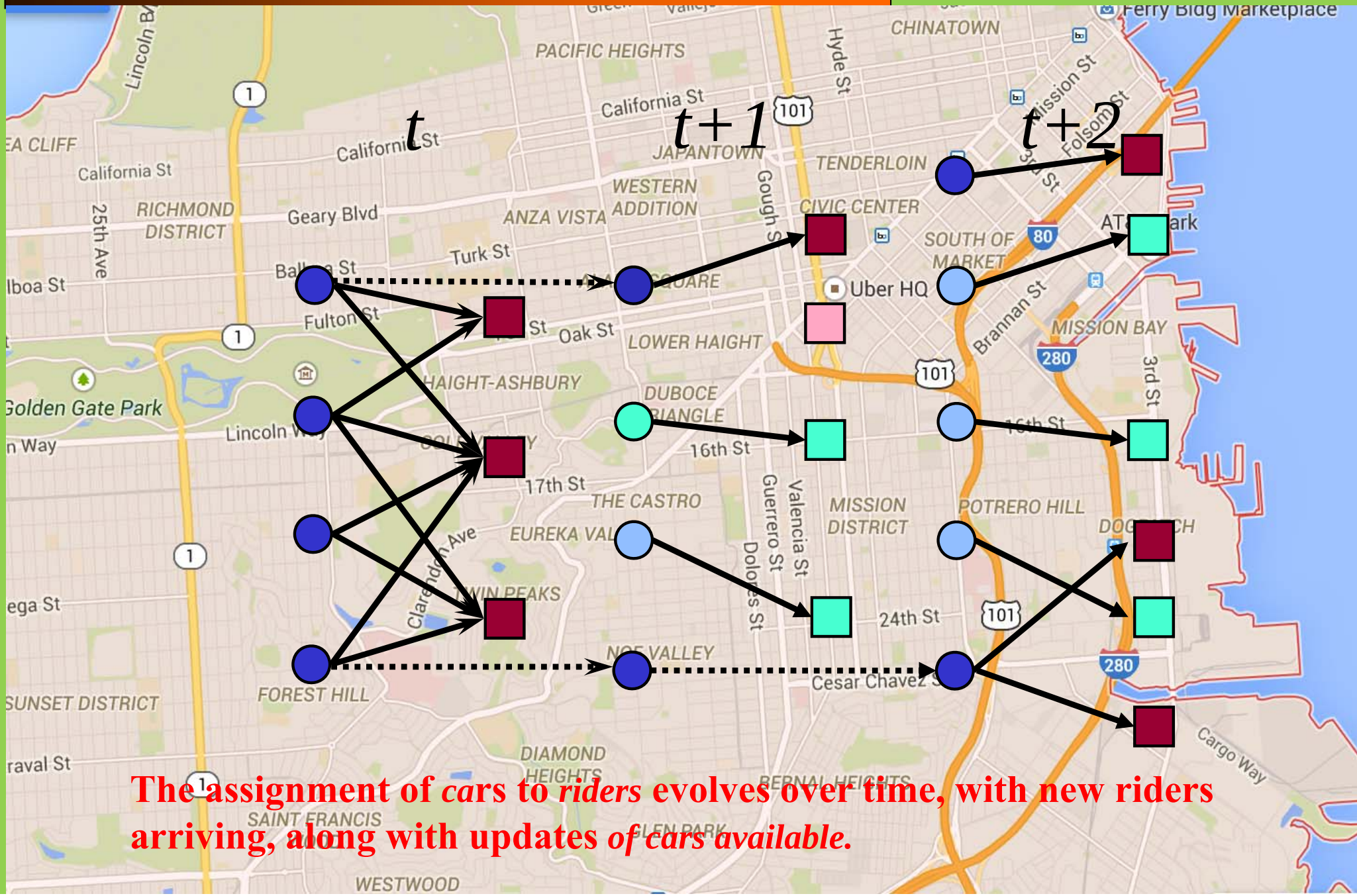
Optimizing over time



Optimizing over time



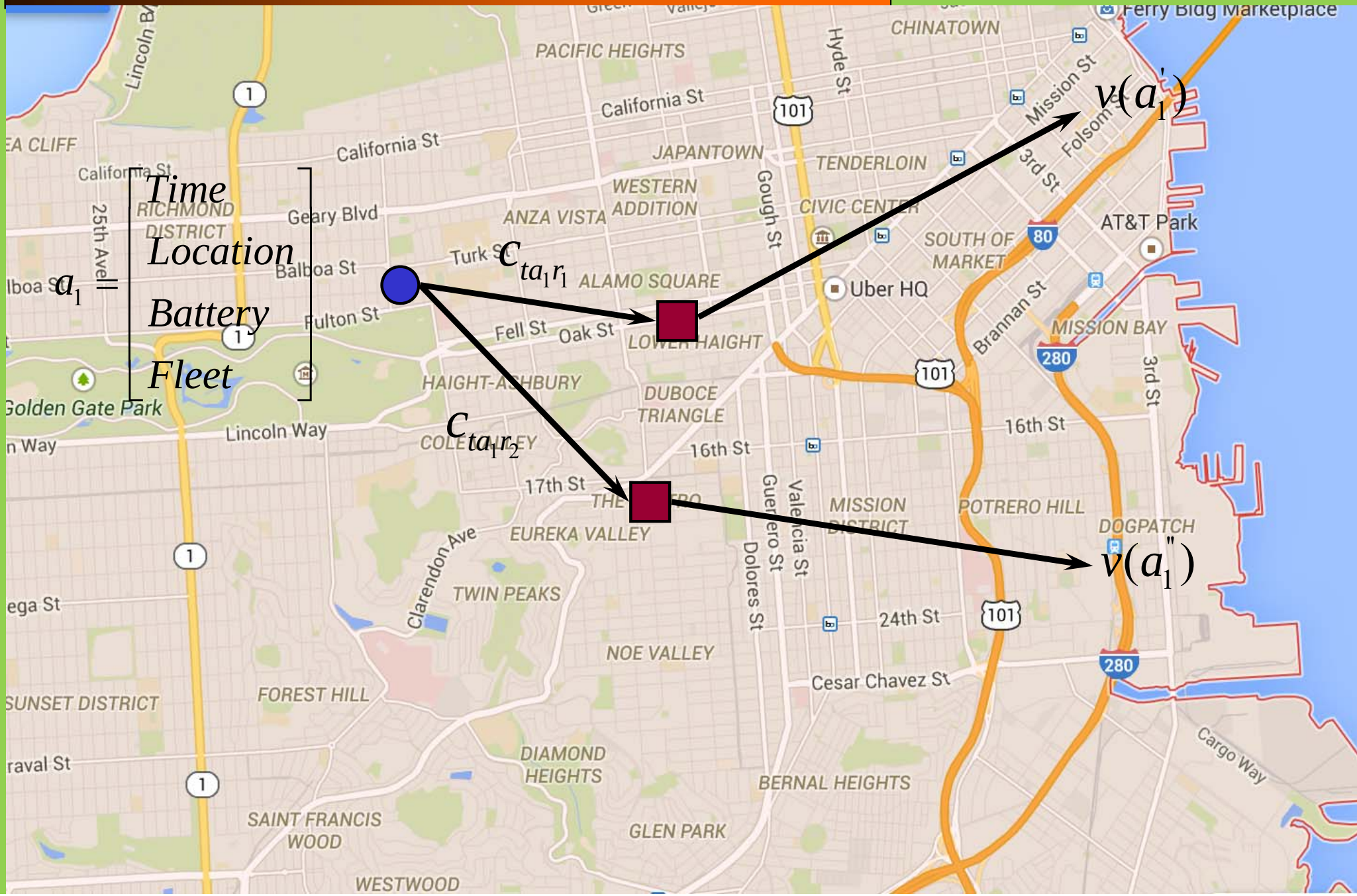
Optimizing over time



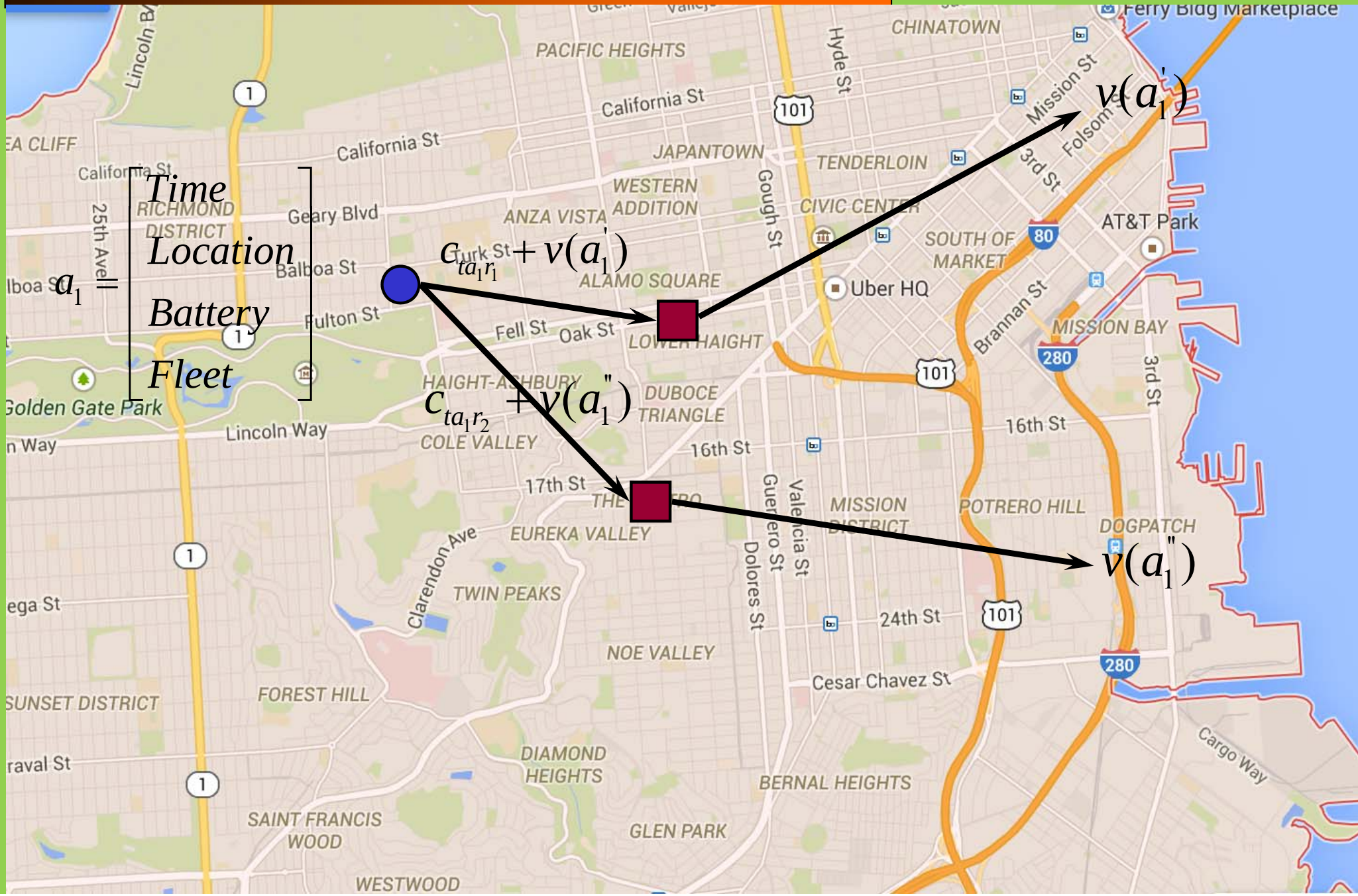
The assignment of *cars* to *riders* evolves over time, with new riders arriving, along with updates *of cars available.*

- Computing value functions
  - » Use the same basic strategy as for blood, but now we are estimating the value of a vehicle.

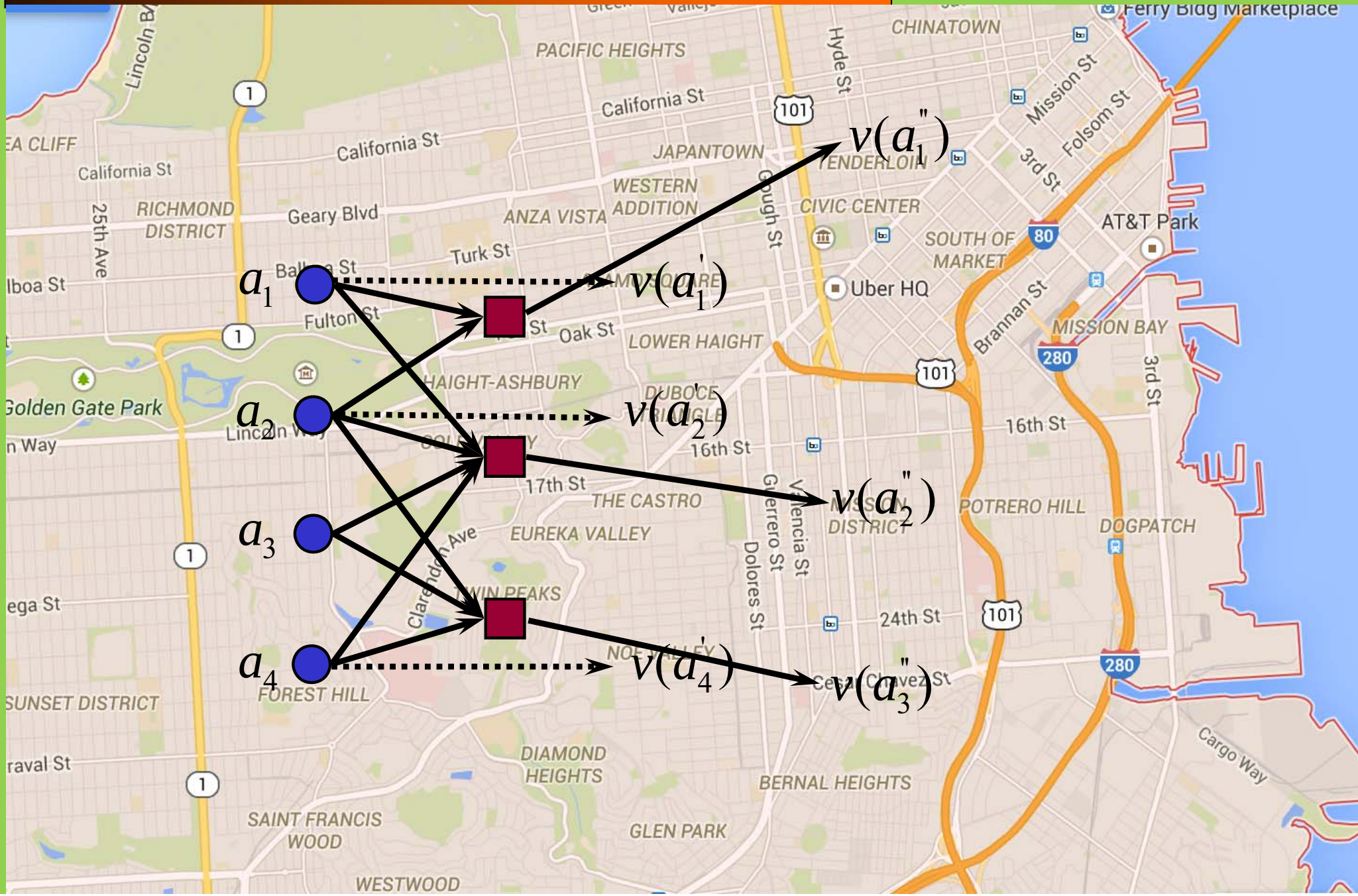
Value function approximations



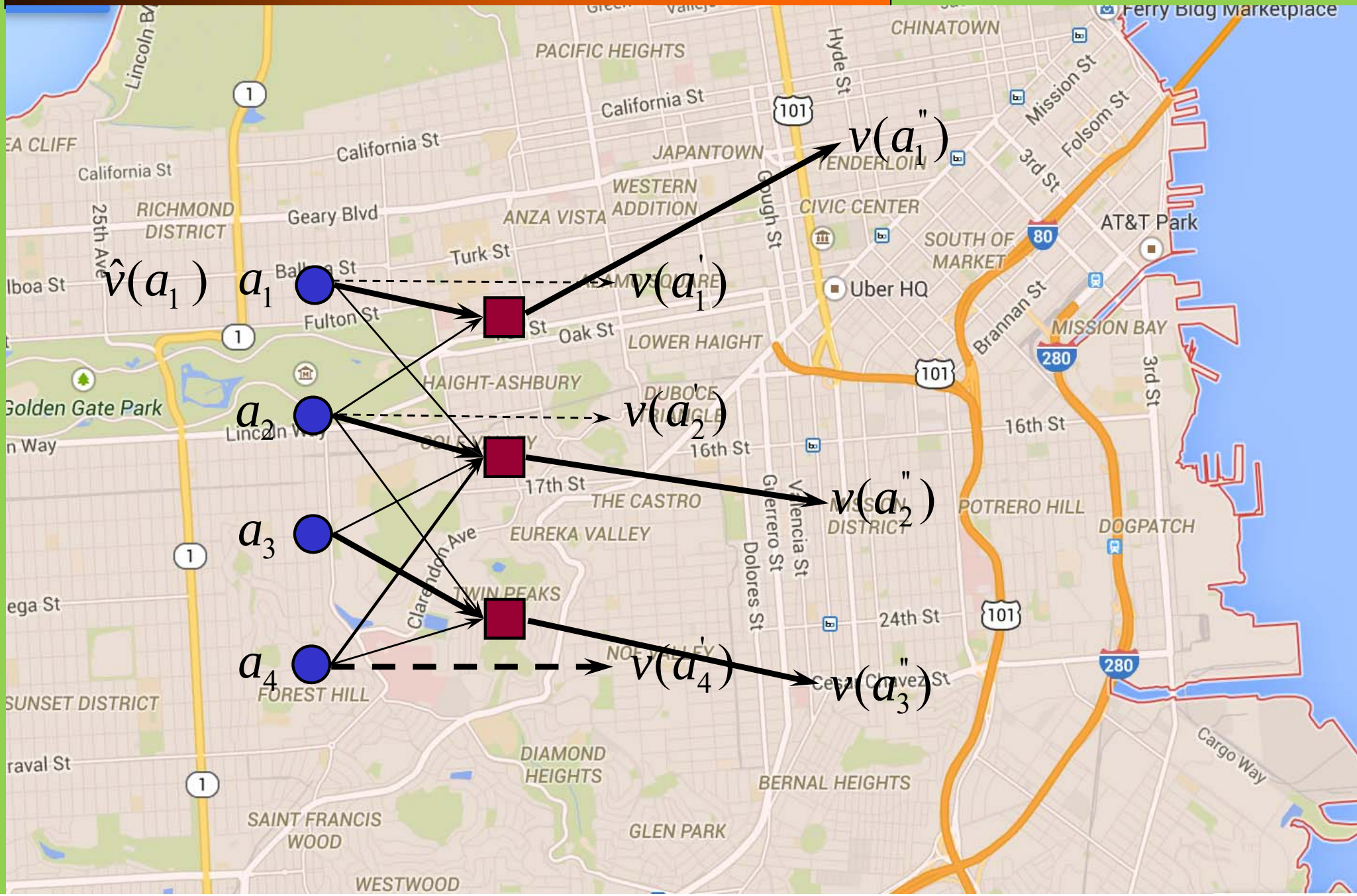
Value function approximations



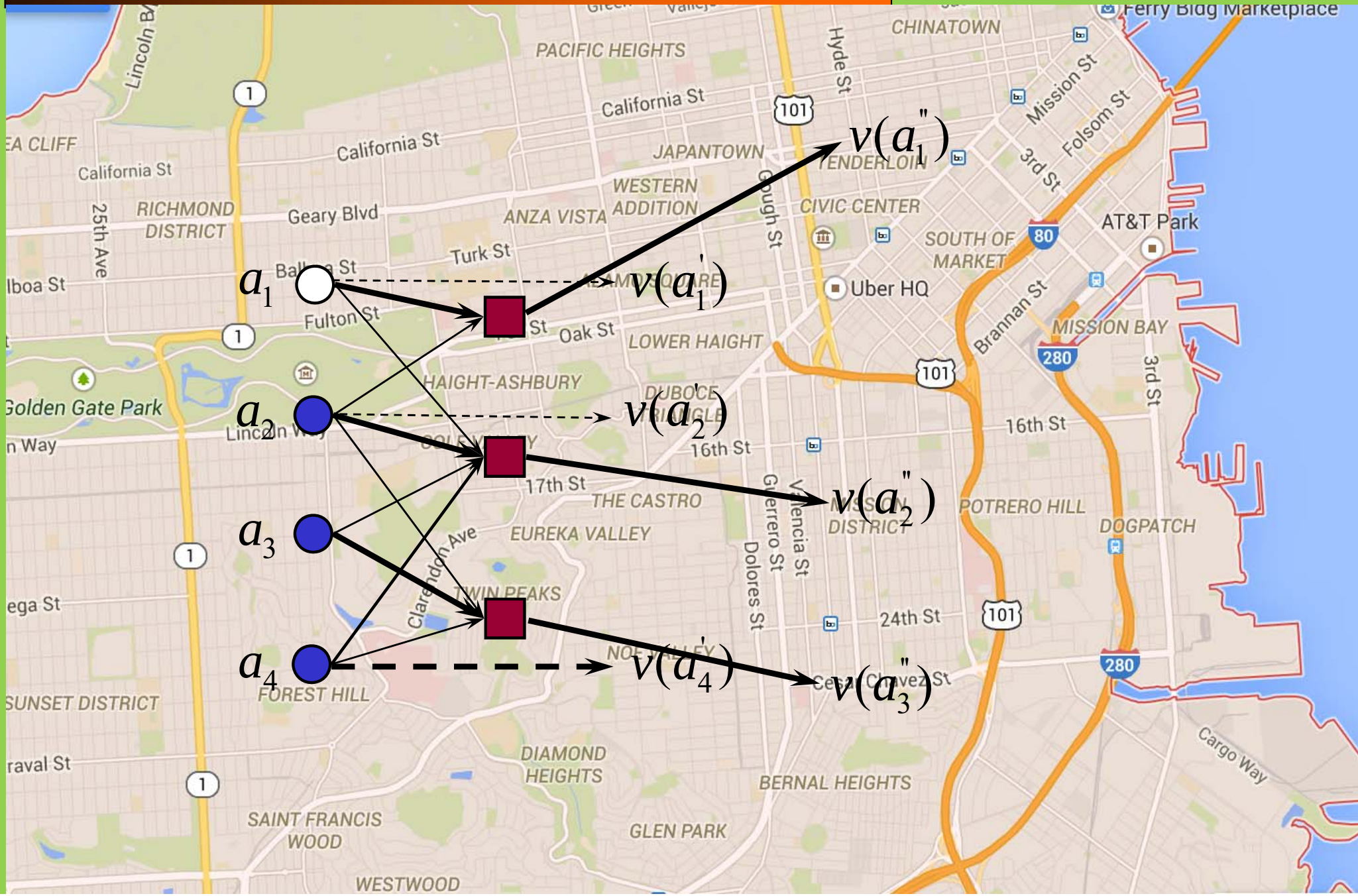
Value function approximations



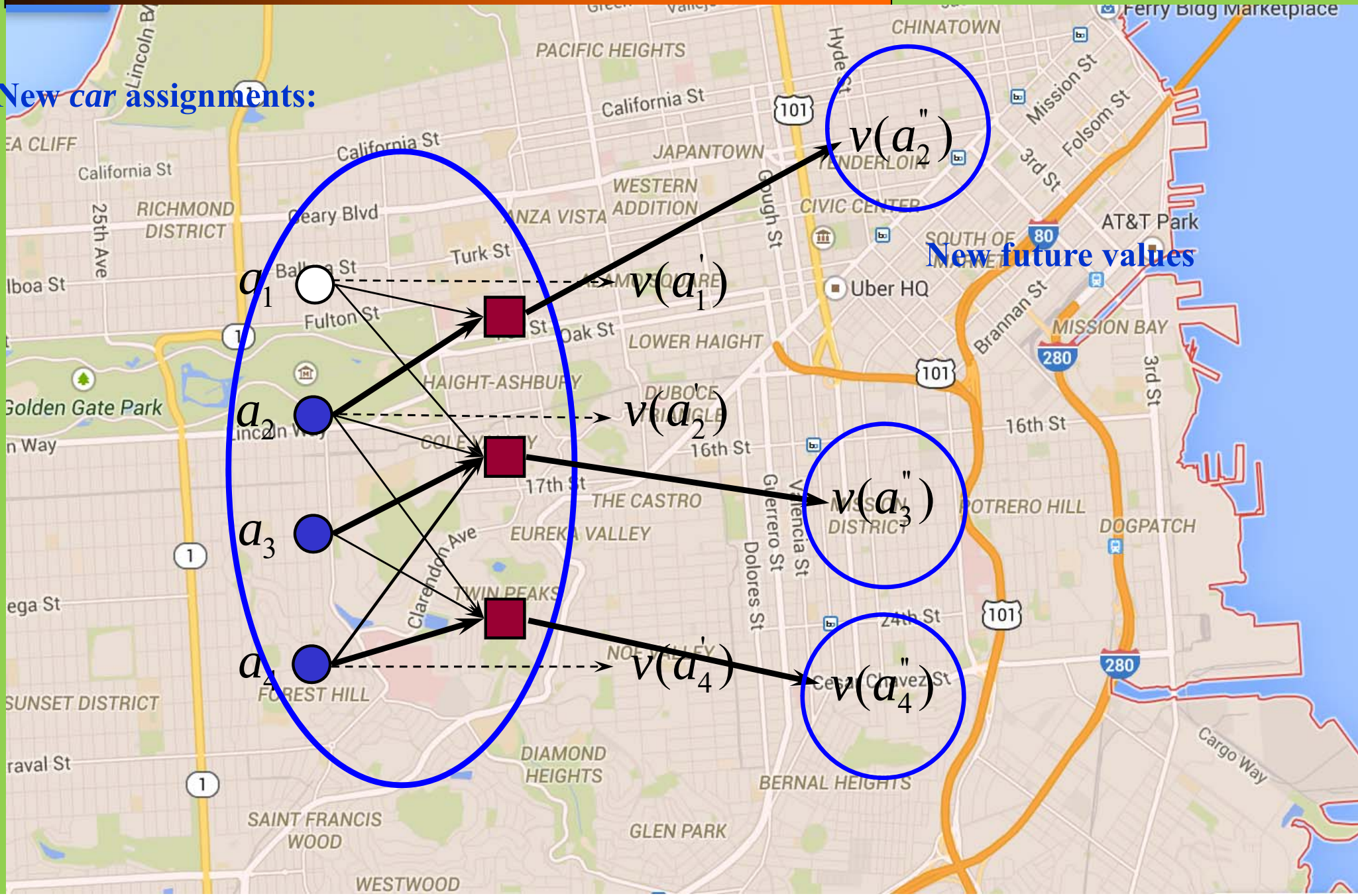
Value function approximations



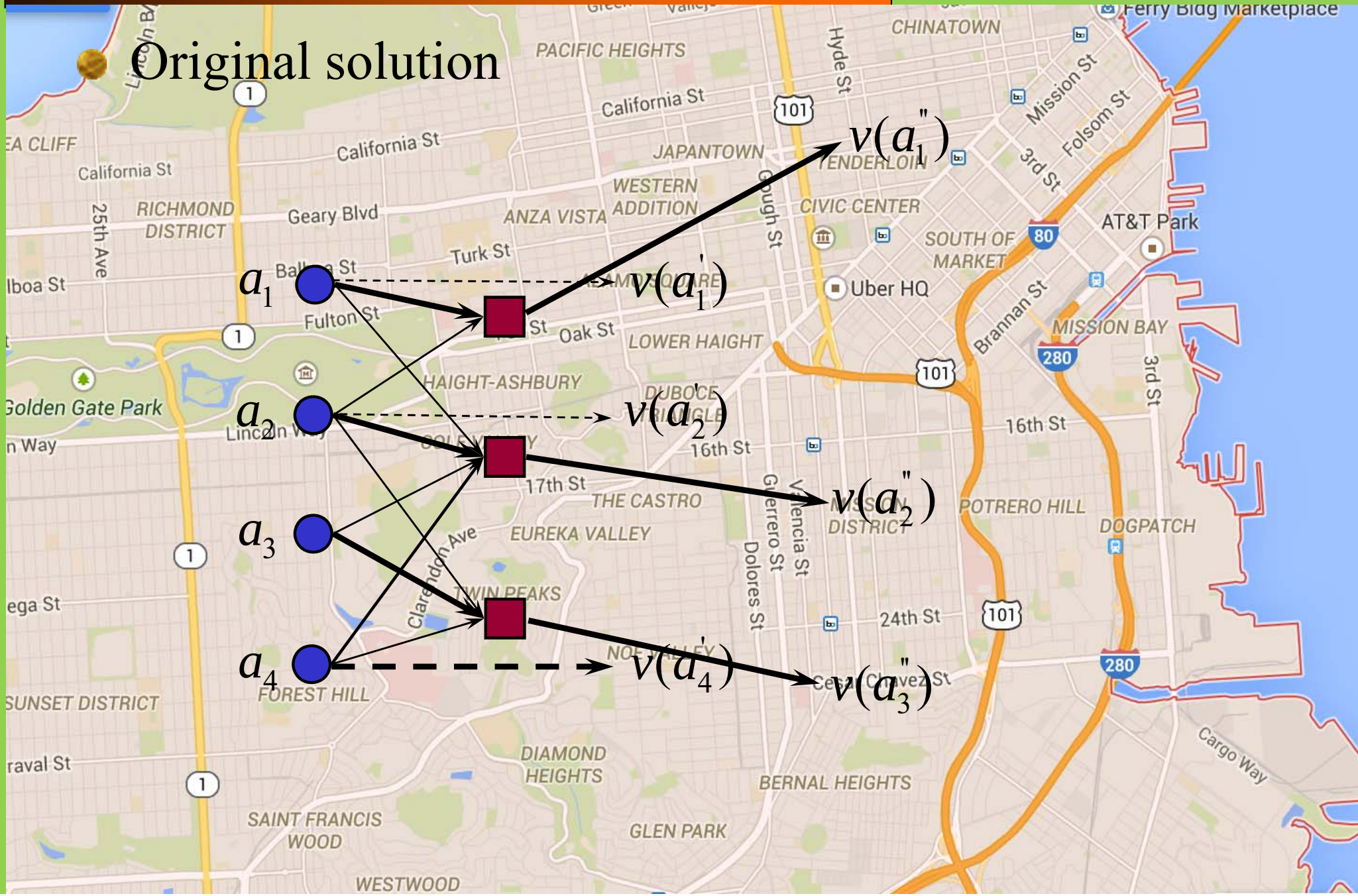
Value function approximations



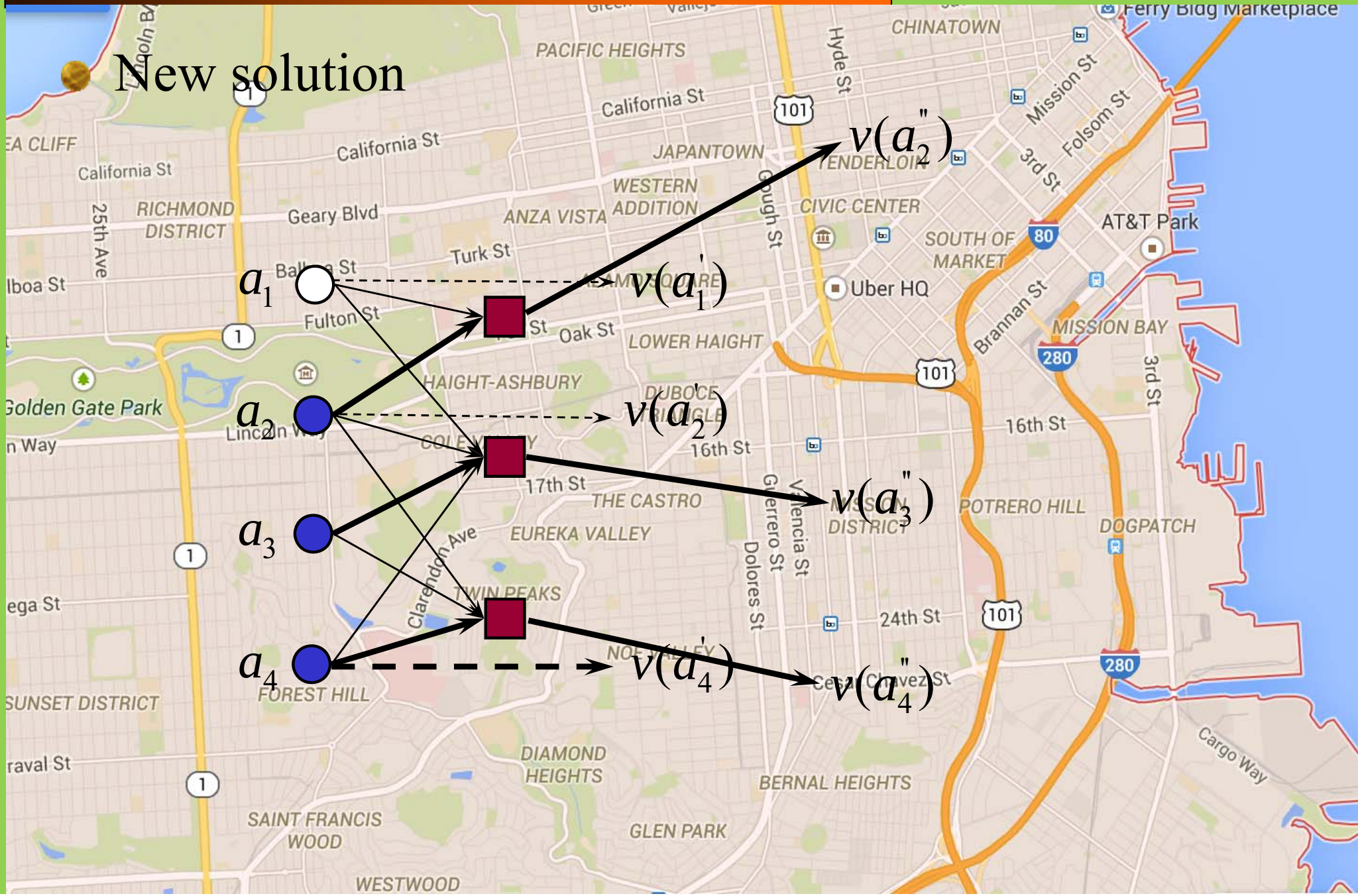
Value function approximations



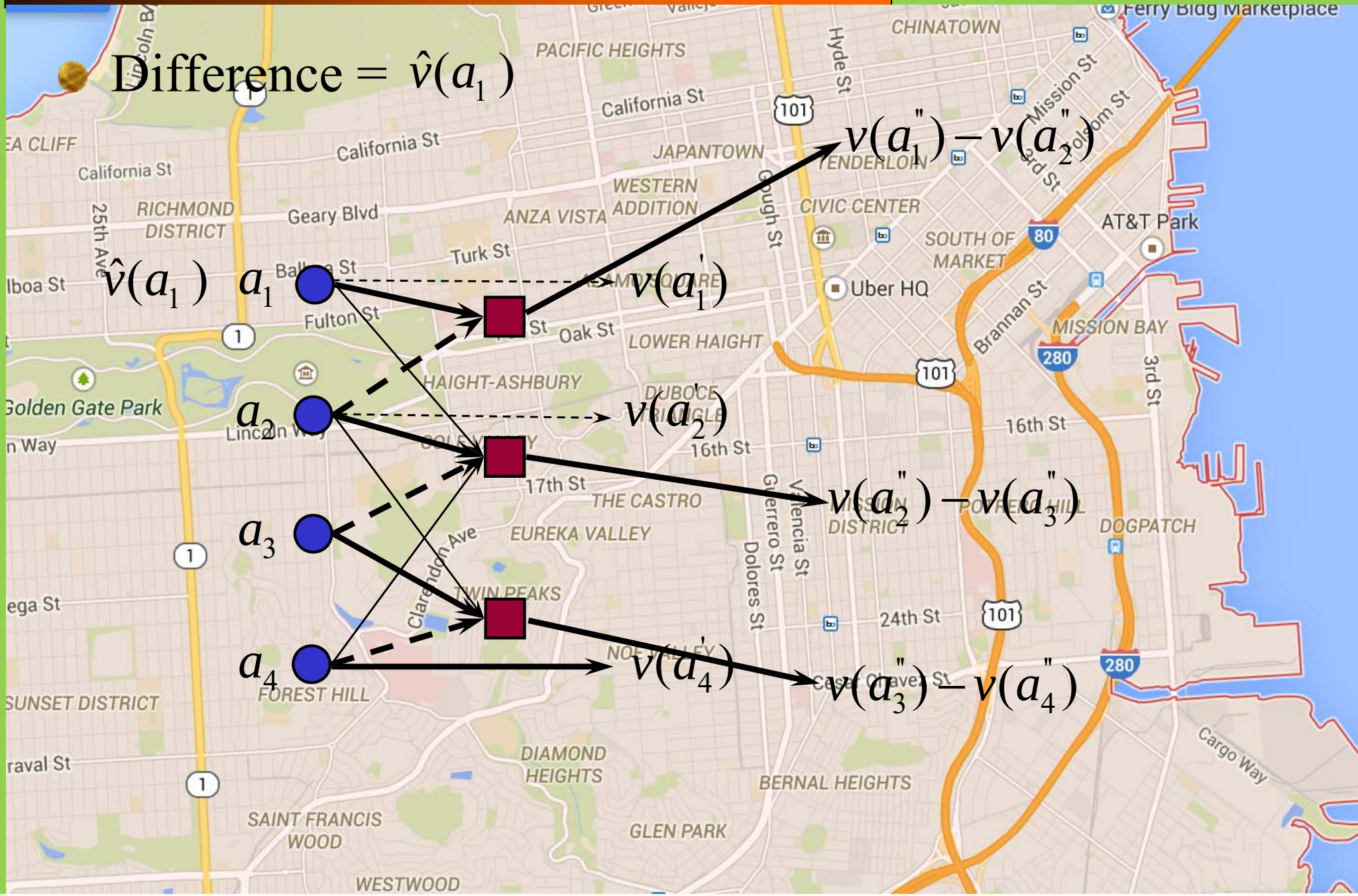
Value function approximations



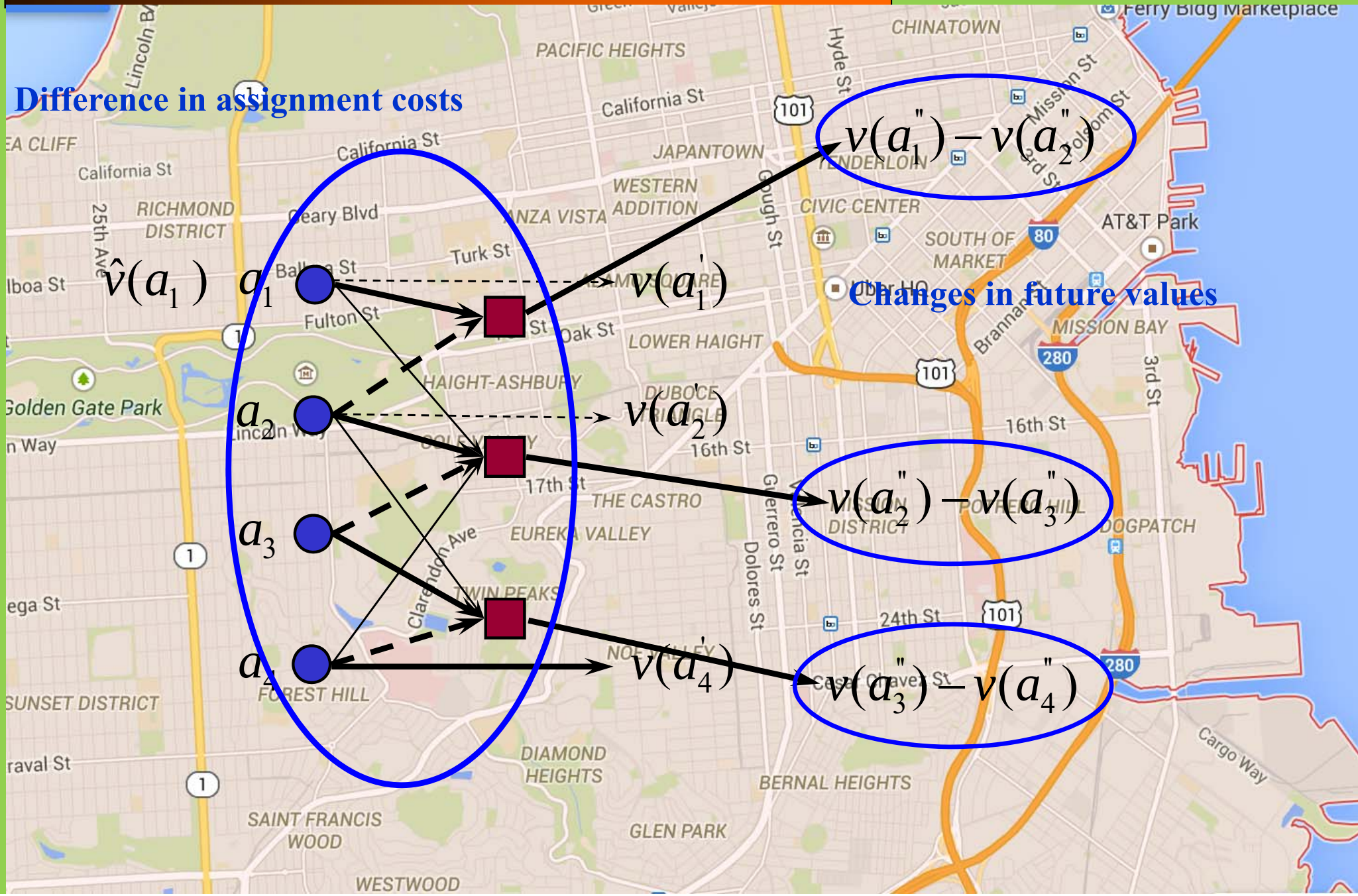
Value function approximations



Value function approximations

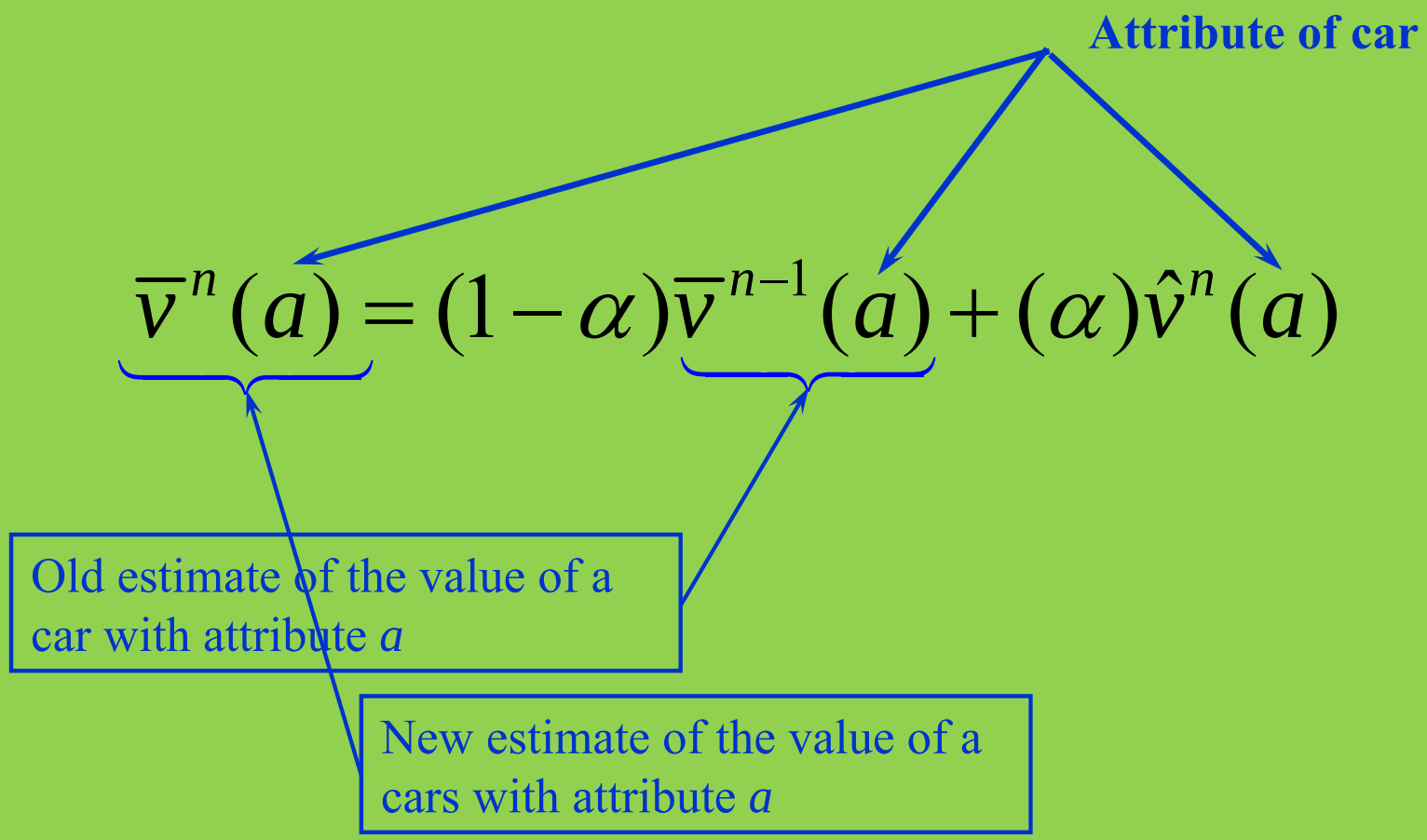


Value function approximations



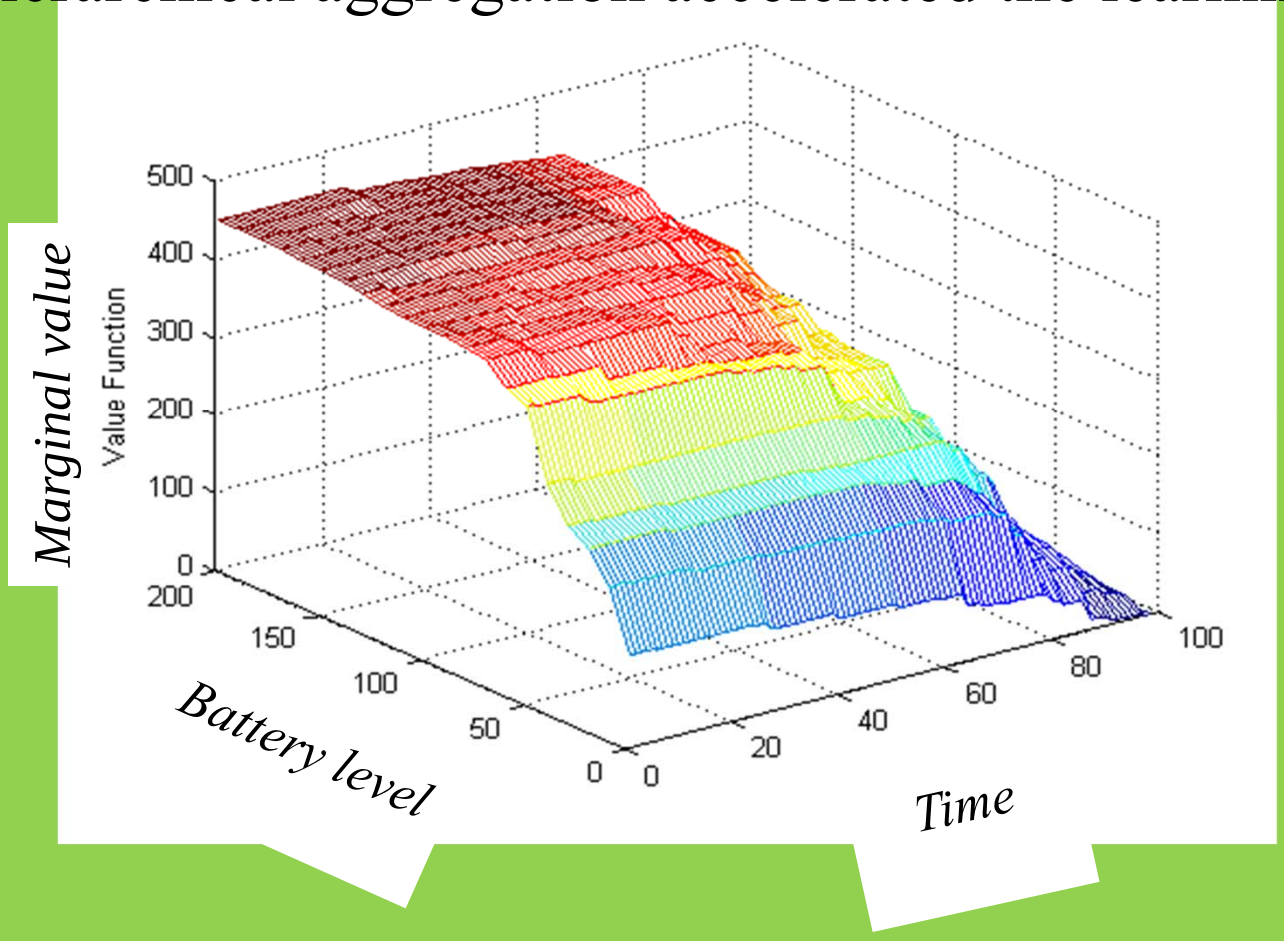
Value function approximations

- Estimating average values:



Driverless fleets of EVs using ADP

- The value of a vehicle in the future
  - » Value function approximation captures charge level, as well as time and location.
  - » Hierarchical aggregation accelerated the learning process



Approximate value iteration

Step 1: Start with a pre-decision state S_t^n

Step 2: Solve the deterministic optimization using an approximate value function:

$$\hat{v}_t^n = \min_x \left(C_t(S_t^n, x_t) + \bar{V}_t^{n-1}(S^{M,x}(S_t^n, x_t)) \right)$$

to obtain x^n.

Deterministic optimization

Step 3: Update the value function approximation

$$\bar{V}_{t-1}^n(S_{t-1}^{x,n}) = (1-\alpha_{n-1})\bar{V}_{t-1}^{n-1}(S_{t-1}^{x,n}) + \alpha_{n-1}\hat{v}_t^n$$

Recursive statistics

Step 4: Obtain Monte Carlo sample of $W_t(\omega^n)$ and compute the next pre-decision state:

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

Simulation

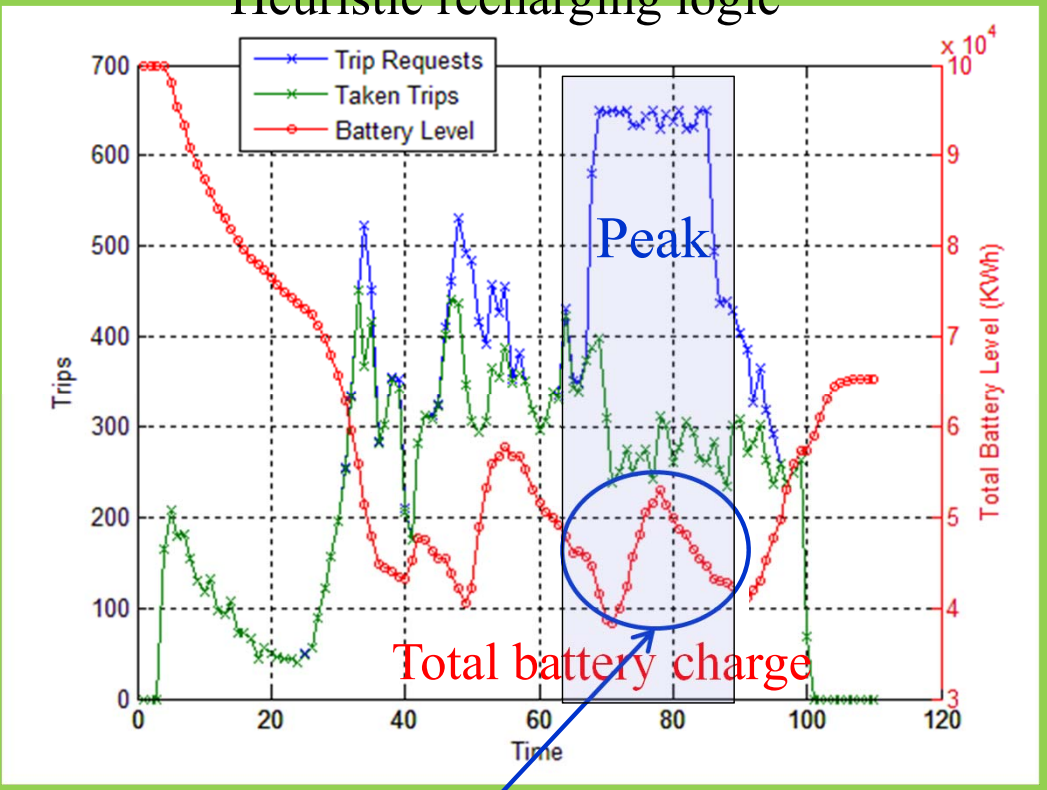
Step 5: Return to step 1.

"on policy learning"

Driverless fleets of EVs using ADP

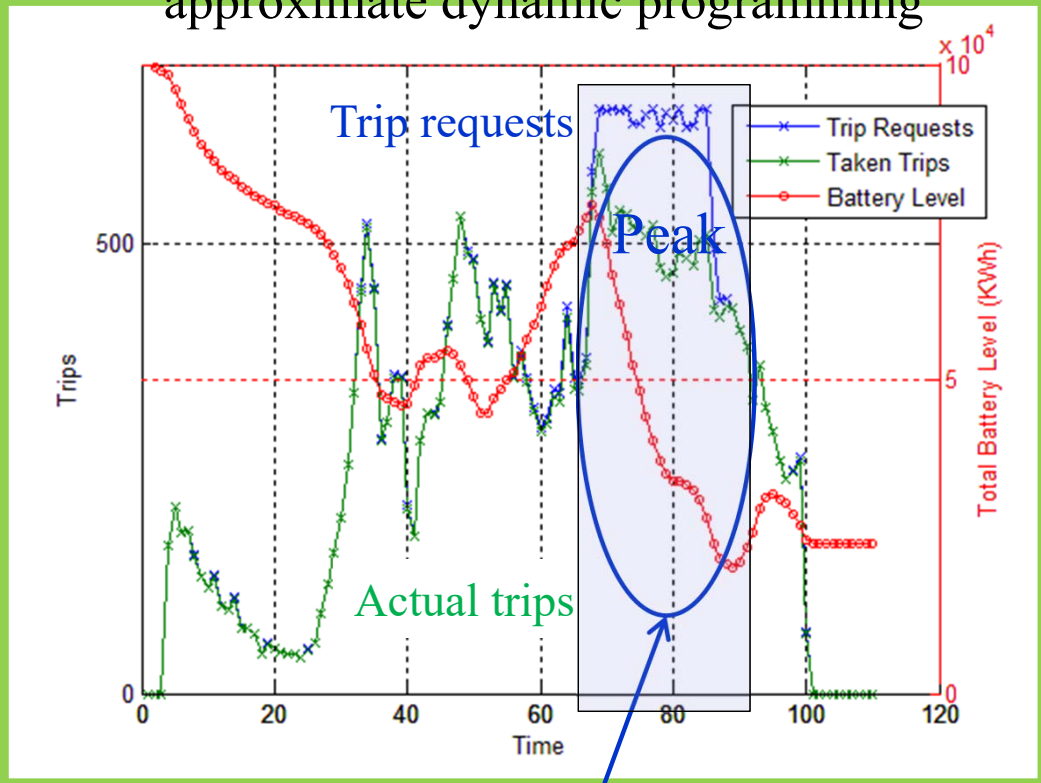
- Heuristic dispatch vs. ADP-based policies
  - » Effect of value function approximations on recharging

Heuristic recharging logic



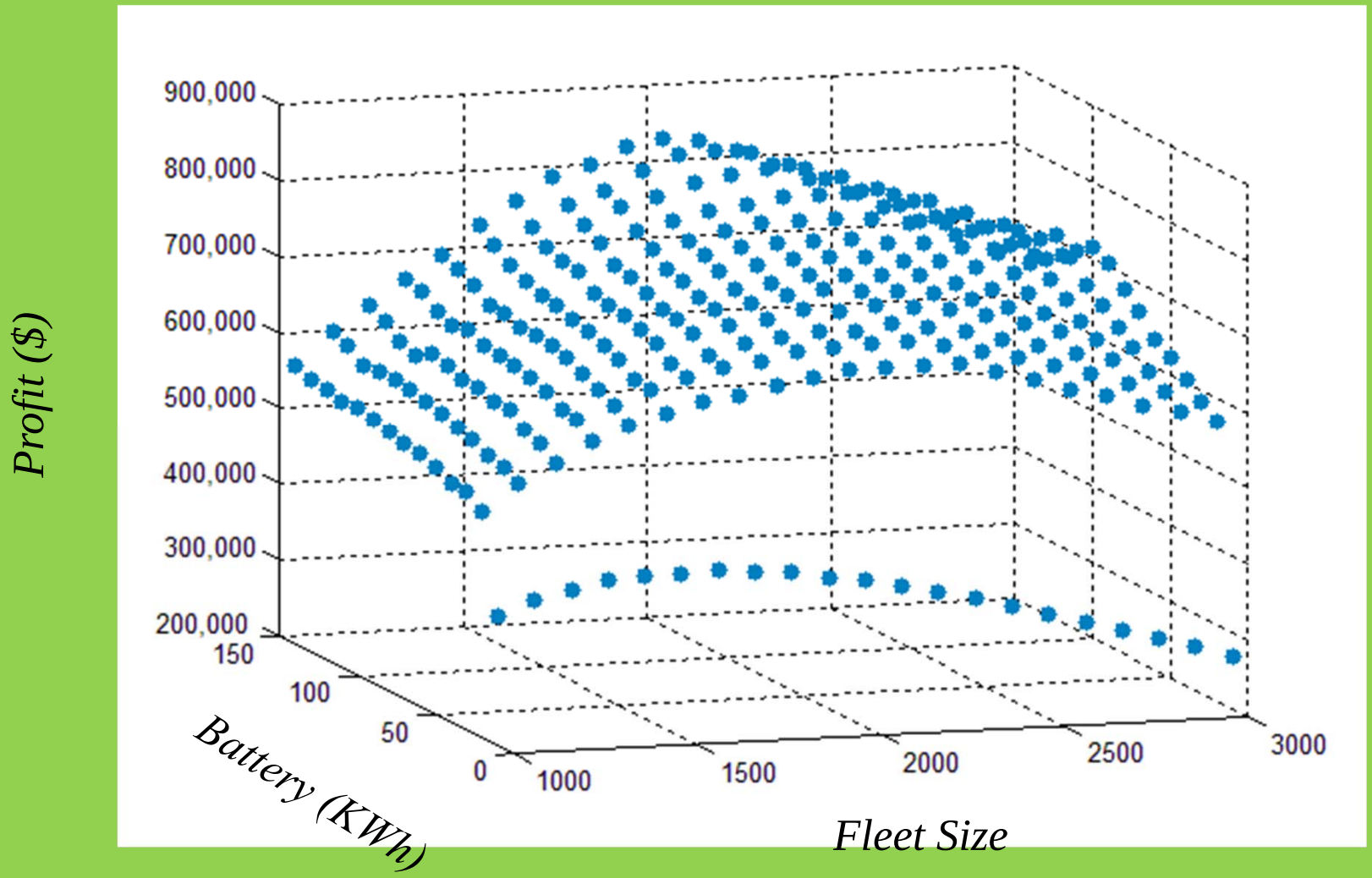
Recharging during peak period

Recharging controlled by approximate dynamic programming



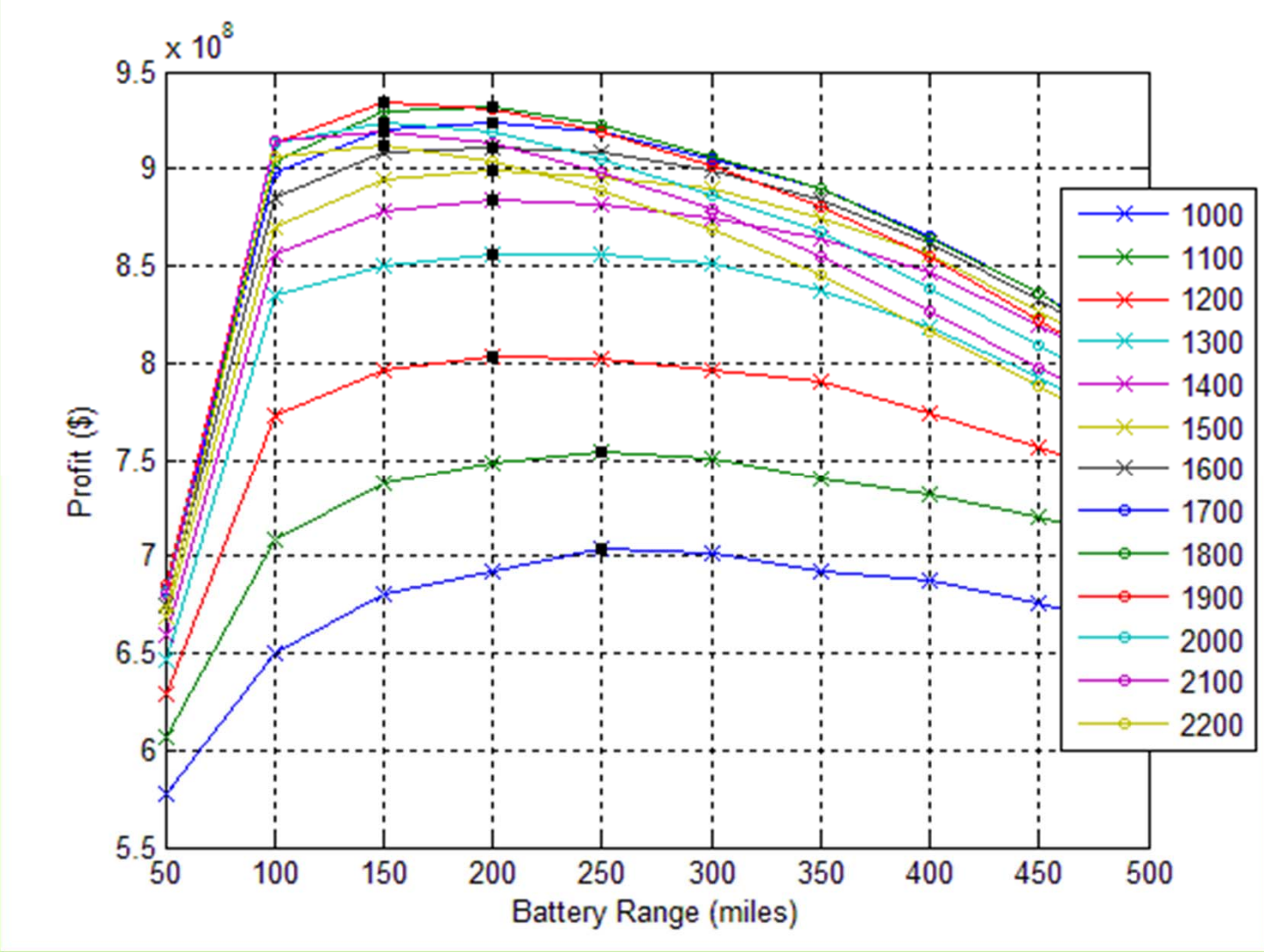
No recharging during peak period

The economics of driverless fleets



We can simulate different fleet sizes and battery capacities, properly modeling recharging behaviors given battery capacity.

Results: Optimizing Fleet and Battery Sizes



Thank You!

Week 10 - Wednesday

Revenue management for hotels

Narrative

Hotel revenue management

- Decision:
  - » Determine the best price to charge for a particular "stay date" to maximize total revenue on that date.
  - » We assume prices can be changed daily (some change prices weekly, some several times a day, and others not at all).
- The booking process
  - » Customers book rooms, primarily through "online travel agents" (OTAs), choosing hotels based on location, price, reputation, and amenities/services.
- Issues:
  - » We do not know how customers will respond to price.
  - » Demand rates vary as a function of the number of days until the stay date, and also depends on the stay date.
  - » The booking pattern also depends (in an unknown way) on the stay date due to special events and weather.

Hotel revenue management

- Broader setting
  - » Customers are organized into three broad classes:
    - "Transients" – individuals booking their own rooms
    - Corporate – Customers associated with corporate accounts
    - Groups – such as conferences.
  - » Booking behaviors can be highly seasonal (ski resorts, beach resorts, hotels near major outdoor activities)
  - » While hotel revenue managers like to maximize revenue, meeting monthly revenue targets is particularly important.
    - Beating the target is fine, but underperforming can pose serious cash flow problems.
    - Hotels in seasonal business lose money during periods of the year. Cash is a valuable (and constraining) resource.
  - » The popularity of the hotel will generally depend on its position relative to hotels in its "competitive set."

Basic model

- State variables
  - » R_t=number of rooms booked
  - » p_{tk}=probability that booking pace curve k is the correct one given what we know by day t.
- Decision variables
  - » p_t=Price to charge for a room
  - » Policy $P^{\pi}(S_t)$
- Exogenous information
  - » $\hat{R}_{t+1}$ =Reservations made on day t+1.
  - » $\hat{R}_{t+1}$ modeled as a logistic regression plus error.

- Transition function
  - » $R_{t+1} = \min\{R^{Max}, R_t + \hat{R}_{t+1}\}$
  - » p_{t+1} updated using Bayes theorem.
- Objective function
  - » $\max_{\pi} \mathbb{E} \sum_{t=T-H}^{T} P^{\pi}(S_t)(R_{t+1} - R_t)$

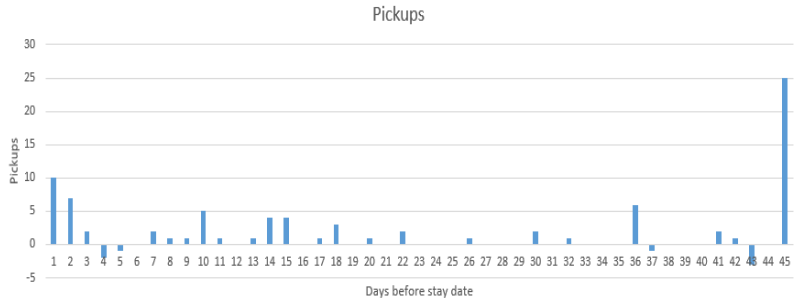
    or we may index time 0,…, T:

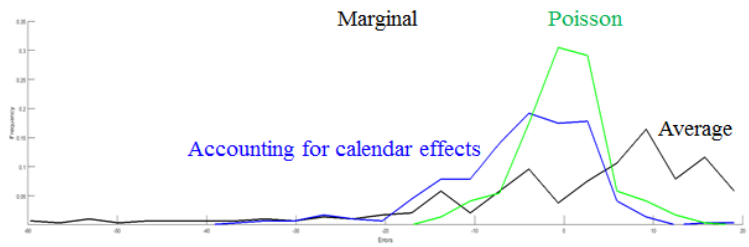
  - » $\max_{\pi} \mathbb{E} \sum_{t=0}^{T} P^{\pi}(S_t)(R_{t+1} - R_t)$

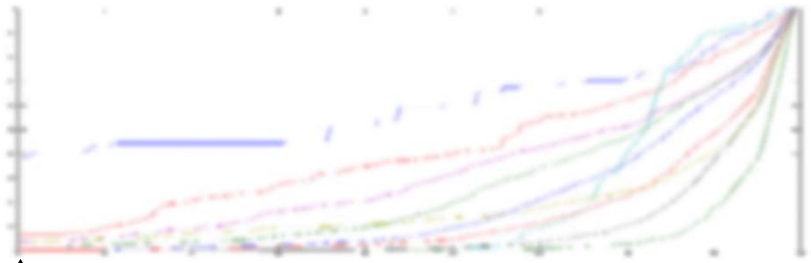
Uncertainty modeling

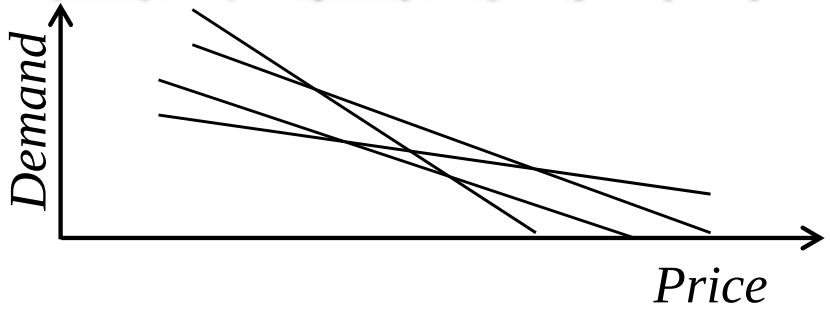
Revenue management

- Types of uncertainty
  - » Randomness in the number of bookings each day due to the underlying Poisson process
  - » Uncertainty in the total number of bookings on the stay date.
  - » Uncertainty in the booking pace curve.
  - » Uncertainty in how the market will respond to price.







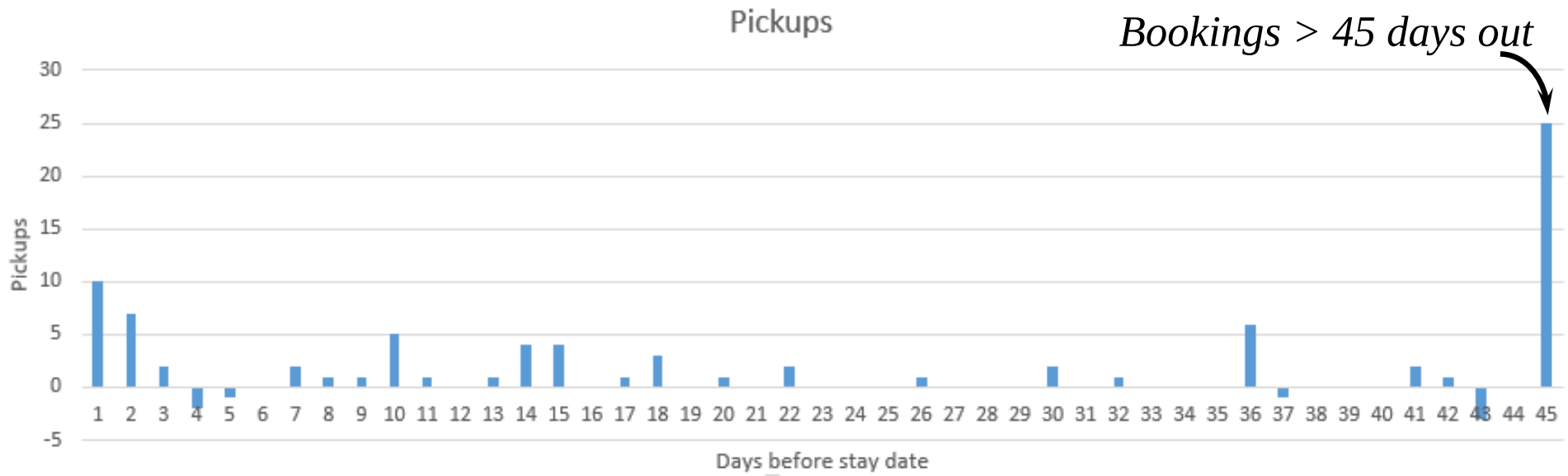


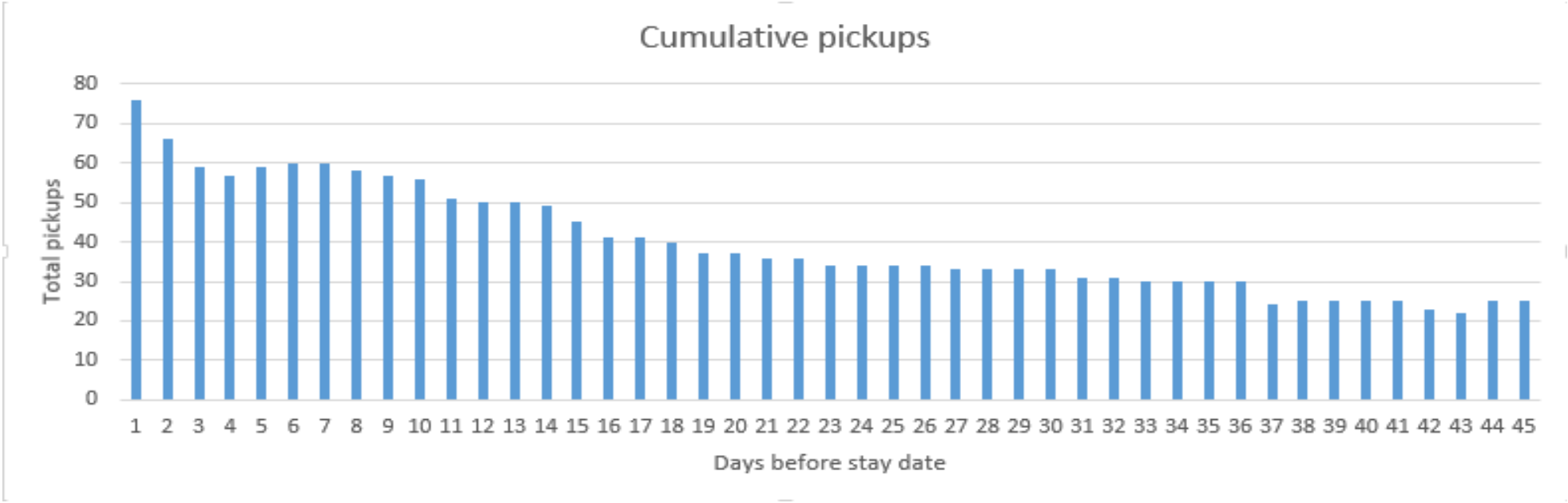
Demand forecasting

Forecasting seasonality

- The booking process
  - » Customers book through:
    - Online travel agents (OTAs)
    - "Property management system" (PMS)
  - » Booking pace (the rate of bookings) reflects:
    - Type of hotel:
      - Business hotels tend to exhibits bookings primarily in last two weeks
      - Resort hotels can have bookings >180 days out.
    - Season
      - Resort hotels have very strong peak seasons
    - Day of week
    - Special events (e.g. graduation) can have bookings much farther (even 365 days out).
  - » Cancellations
    - We only observe net bookings = bookings minus cancellations

Forecasting seasonality





Forecasting seasonality

- Components of the booking model
  - » Start with a base forecast of the total number of customers who will book on a stay-date T
  - » Let

$$f_{tT} = \text{Forecast of total bookings on stay date } T\text{, given current estimates as of time (week) } t.$$

$$= B_t \theta^W_{t,w(T)} \theta^D_{t,d(T)}$$

where

B_t = Estimate of baseline bookings/day as of week t.

$w(T)$ = Week of year corresponding to stay date T, $1 \le w \le 52$

$d(T)$ = Day of week corresponding to stay date T, $1 \le d \le 7$

$\theta^W_{t,w(T)}$ = Adjustment factor for week $w(T)$.

$\theta^D_{t,d(T)}$ = Adjustment factor for week $d(T)$.

Forecasting seasonality

- Use first year to initialize the model
  - » B_0 = Average pickups/day over first year
  - » $\theta^W_{0w} = \dfrac{N_w}{7B_0}$ $1 \le w \le 52$ N_w = Bookings in week w in first year
  - » $\theta^D_{0d} = \dfrac{N_d}{B_0}$ $1 \le d \le 7$ N_d = Avg. bookings on day d in first year
- Notes
  - » Initializing the baseline to the yearly average can introduce a significant distortion since we will allow the baseline to adapt.

Forecasting seasonality

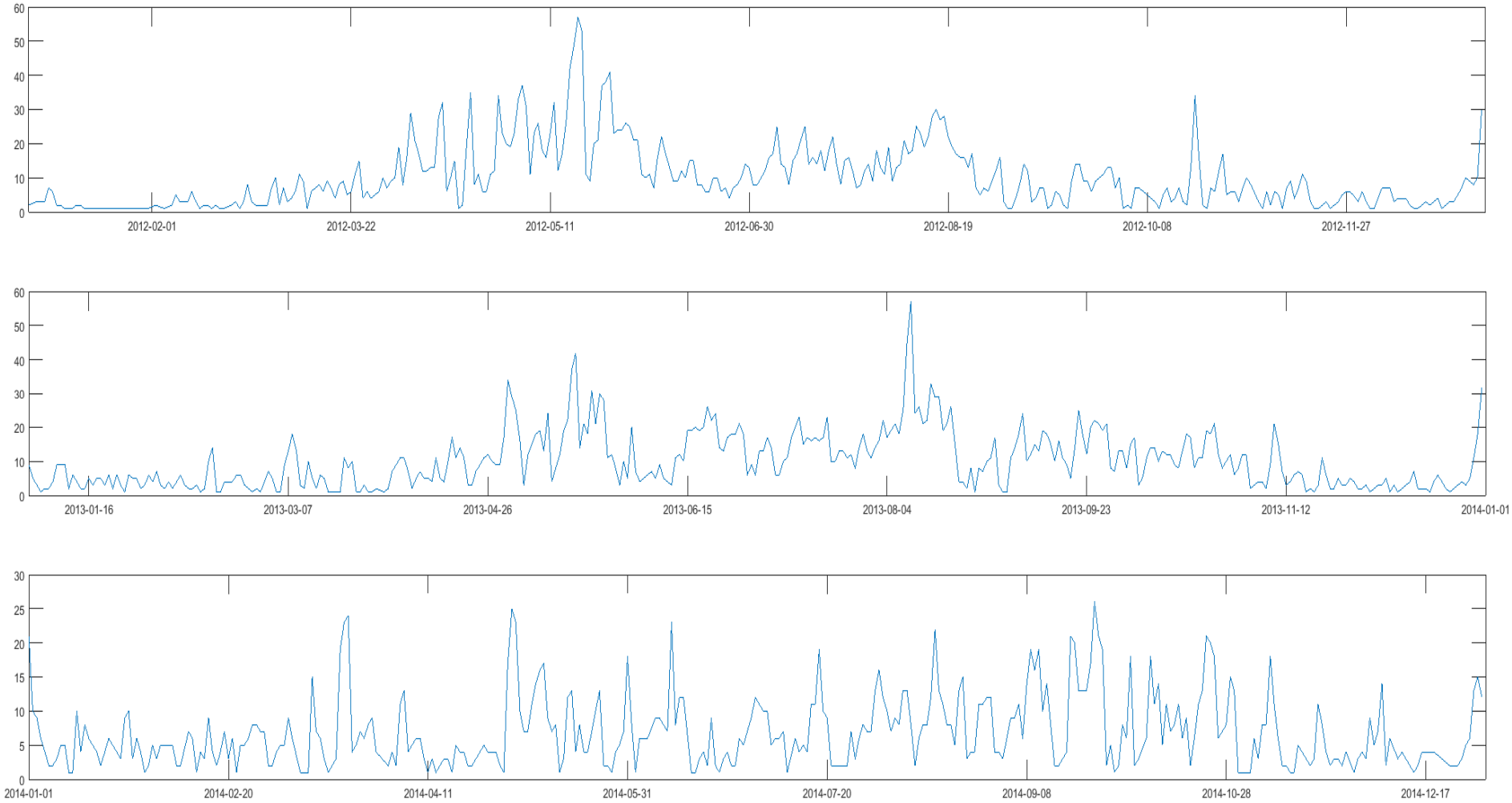
- Adaptive updating
  - » Given the initial estimates, we then update our parameters each week, starting with the first year.
  - » Let *t* be the week (counting from beginning of the dataset), and let *w(t)* be the week within the year, where we have observed
    - $N_{w(t)}$ pickups (bookings) during week *w(t)*.
    - $N_{d(t)}$ pickups on each day *d(t)* of week *t*.
  - » Updating formulas:
    - $B_t = (1-\alpha^B)B_{t-1} + \alpha^B \frac{N_{w(t)}}{7}$
    - $\theta^W_{t,w} = (1-\alpha^W)\theta^W_{t-1,w} + \alpha^W \frac{N_{w(t)}}{7B_t} \quad w = w(t)$
    - $\theta^D_{t,d} = (1-\alpha^D)\theta^D_{t-1,d} + \alpha^D \frac{N_{d(t)}}{B_t} \quad 1 \le d = d(t) \le 7$

Forecasting seasonality

- Notes on smoothing factors
  - » Baselines reflect changes in the market, economy. Typically these are allowed to adapt more quickly. But because they are updated weekly, need to be careful not to overreact to special events.
    - Typical smoothing values range (.05 - .15)
  - » Weekly factors are updated once each year, so the updating has to be more aggressive.
    - Typical smoothing values might be (.10 - .20)
    - Might reduce this once model has stabilized.
  - » Daily factors are updated each week. Day-of-week effects should be fairly stable, so smoothing values are smaller (.02-.05).

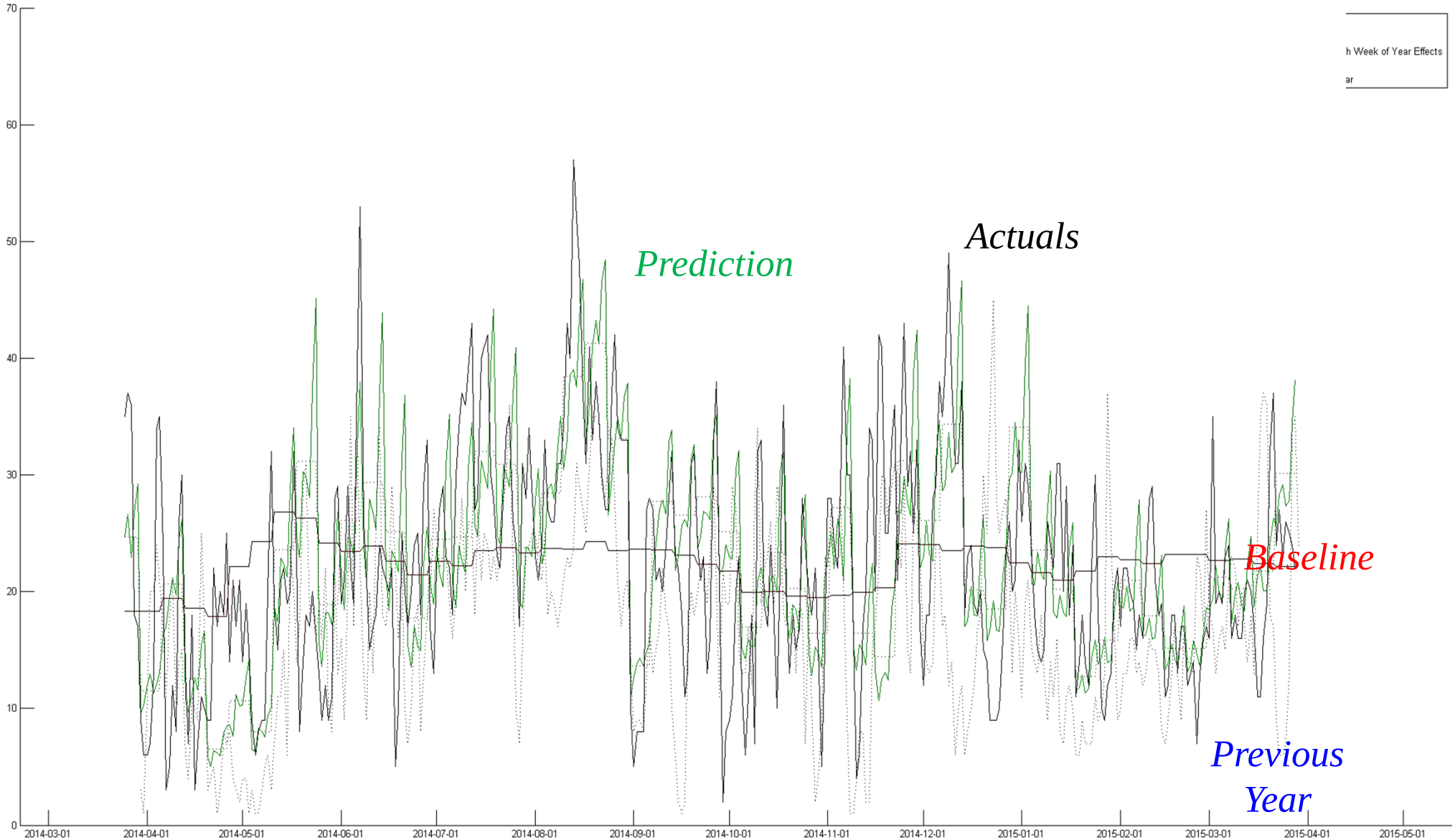
Forecasting seasonality

- Historical bookings by stay date, 2012-2014



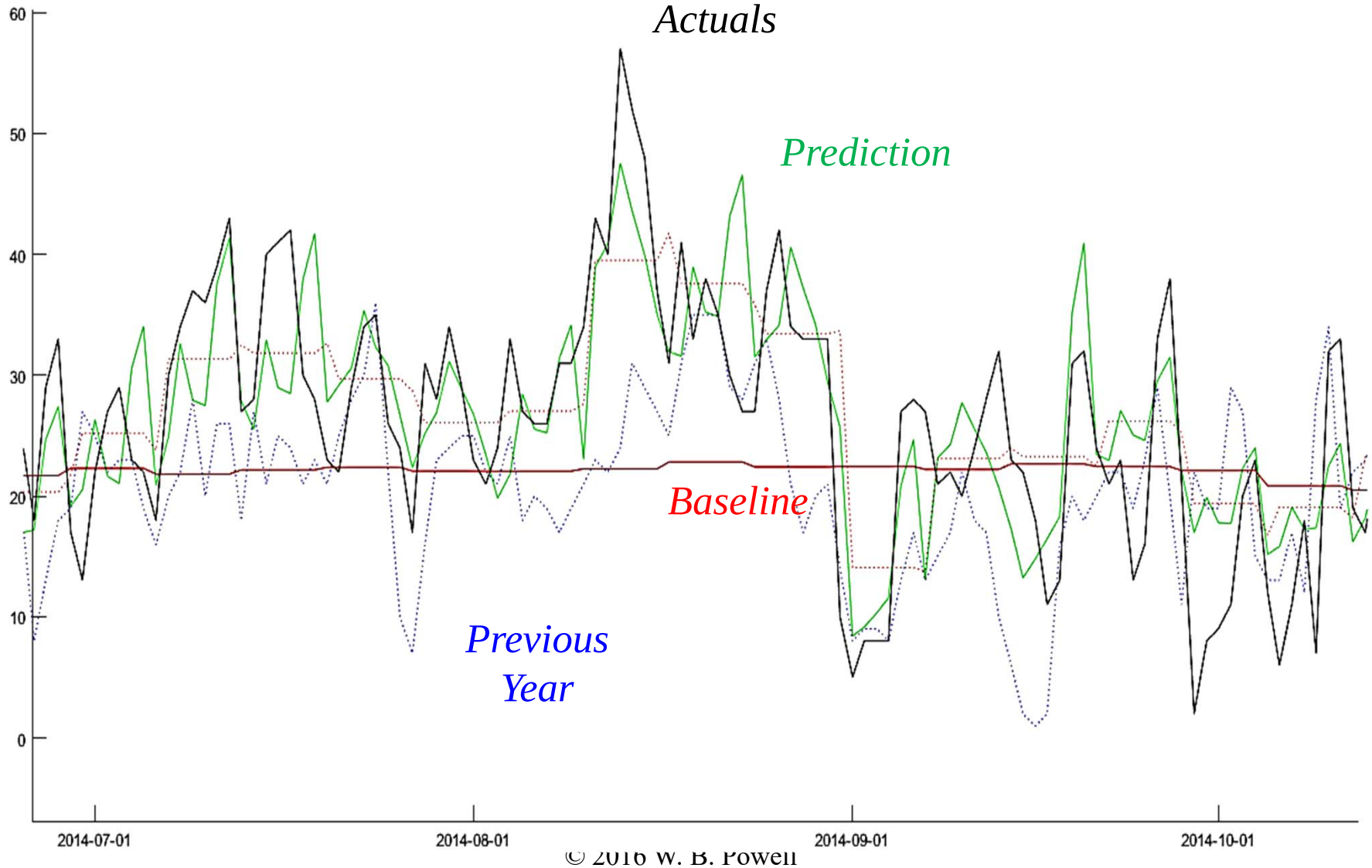
Forecasting seasonality

- Actual vs. forecast



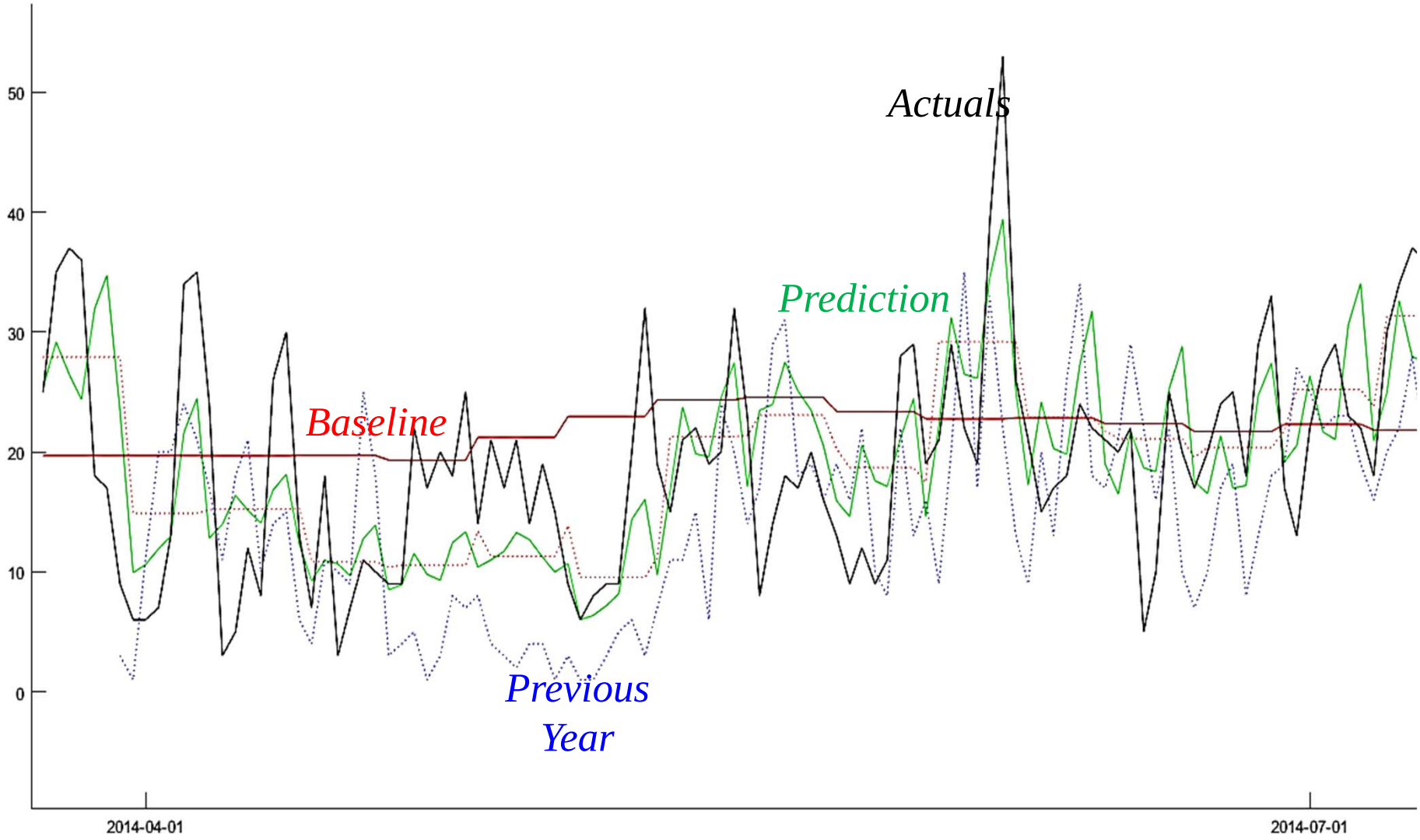
Forecasting seasonality

- Actual vs. forecast



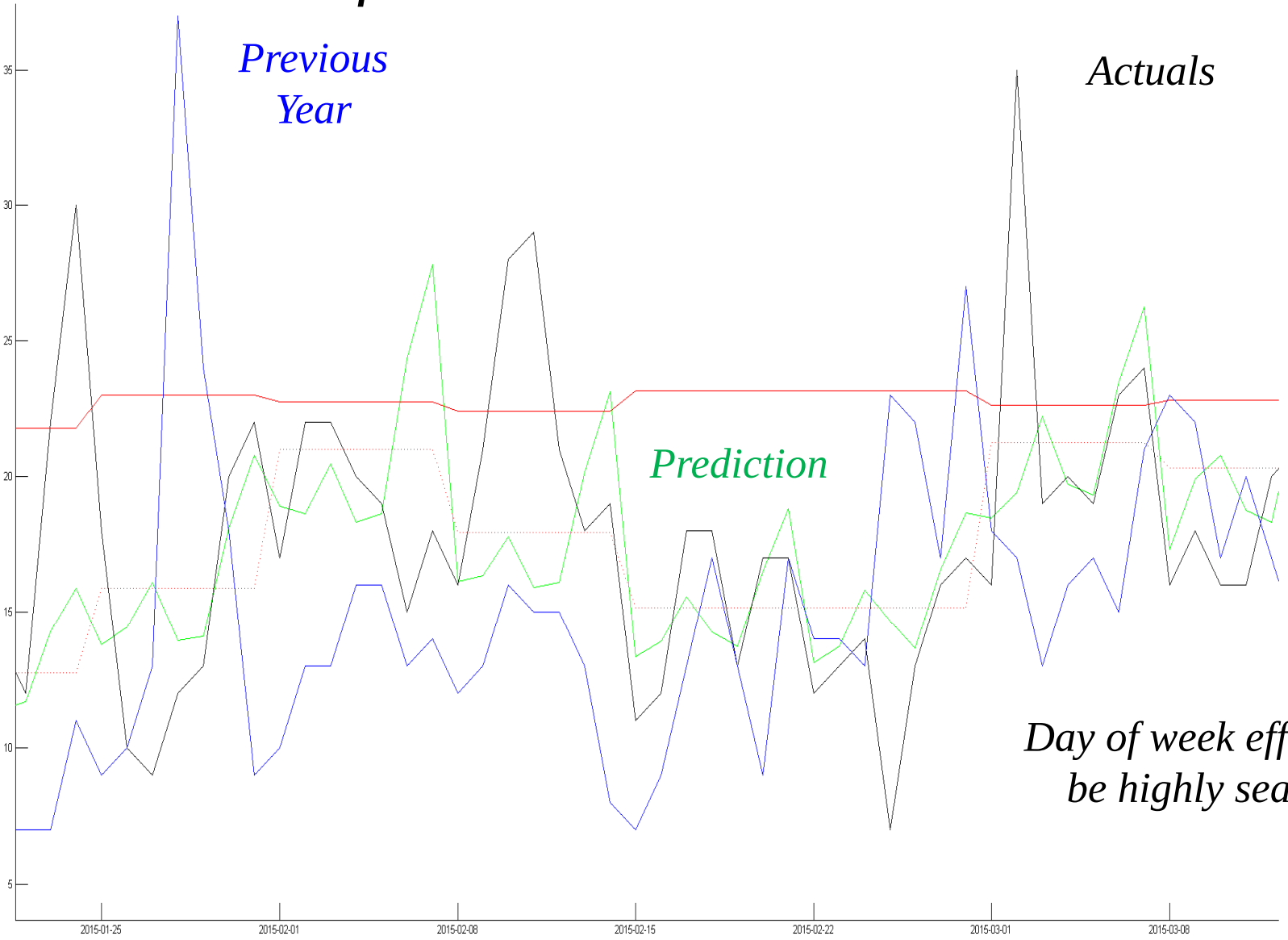
Forecasting seasonality

- Actual vs. forecast



Forecasting seasonality

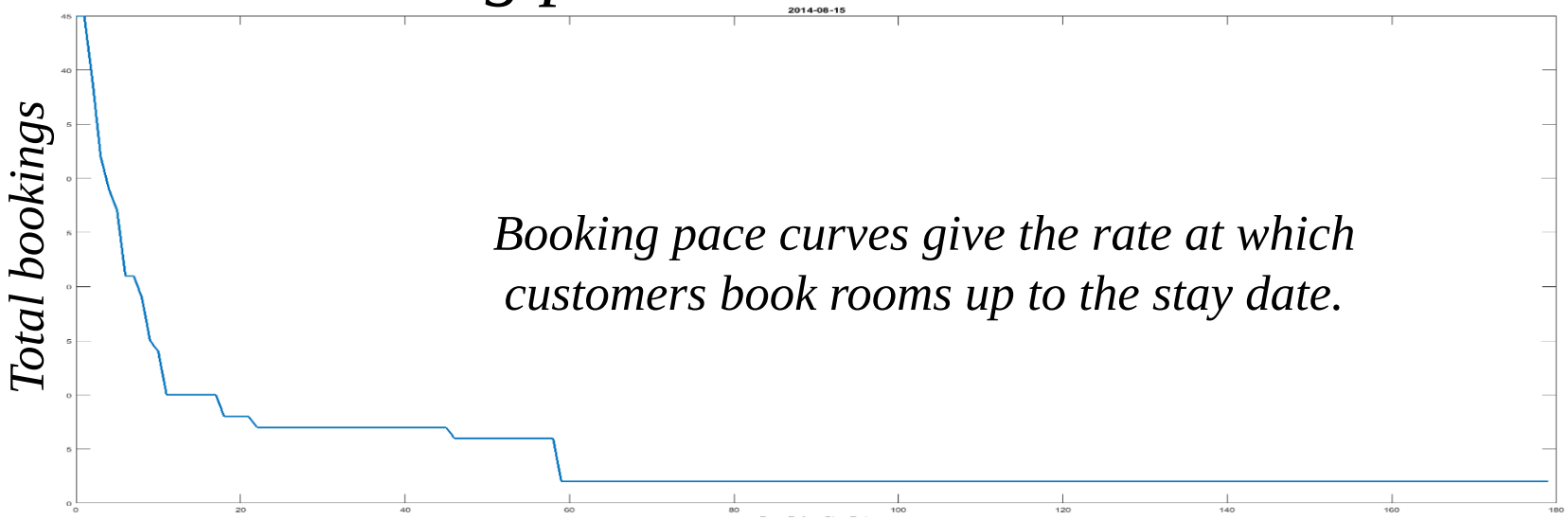
- *Actual vs. forecast*



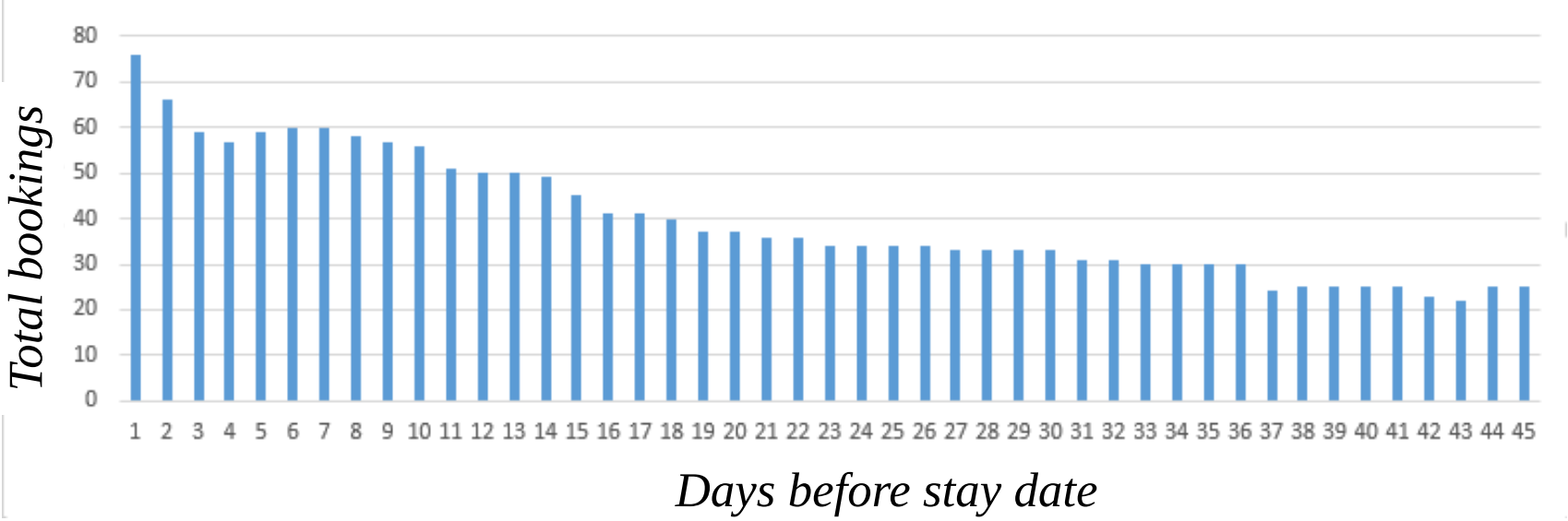
Booking pace curves

Booking pace curves

- *Some booking pace curves*



Booking pace curves give the rate at which customers book rooms up to the stay date.



Booking pace curves

- The booking pace curve
  - » This is the rate at which customers call in for a particular stay date
  - » We assume that there is a family of booking pace curves (for a particular hotel) given by

g_τ^k = Fraction of customers who will call in τ days in advance for curve $k \in \mathcal{K}^{BP}$ =set of booking pace curves.

  - » Now we have the problem of identifying these curves.

Booking pace curves

- Identifying booking pace curves
  - » Get booking pace curves from history for a 1-2 year horizon.
  - » Let

$$h_{t,T} = \frac{N_{t,T}^{C}}{N_T}$$

$$= \frac{\text{Cumulative number of bookings by } t \text{ for state date T}}{\text{Total bookings on stay date T}}$$

$$= \text{Fraction of bookings } \tau = T - t \text{ days before stay date.}$$

  - » Now compute a "distance" $d_{T,T'}$ between two curves using

$$d_{T,T'} = \sum_{\tau=0}^{\tau^{\max}} | h_{T-\tau,T} - h_{T'-\tau,T'} |$$

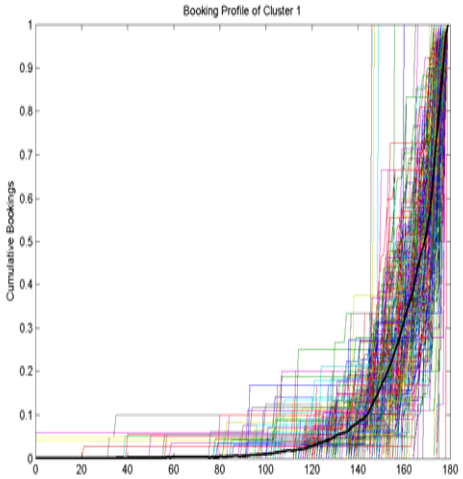
  - » Next, use k-means clustering (available in Matlab) to cluster curves into *K* representative booking pace curves:

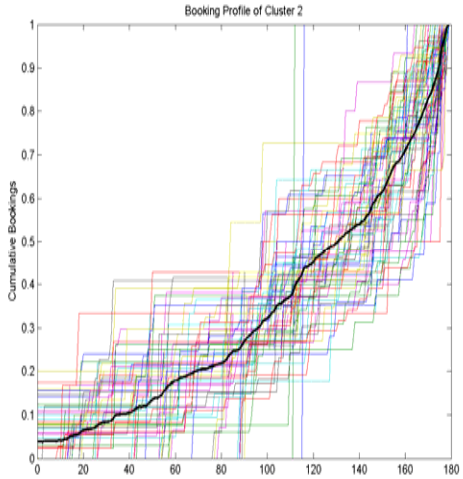
$$h_{\tau}^{k} = \text{Fraction of customers who call in } \tau \text{ or more days in advance}$$

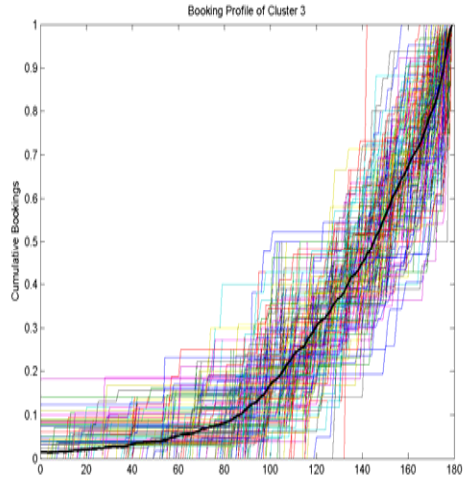
$$\text{according to booking pace curve } k \in \{1,...,K\}$$

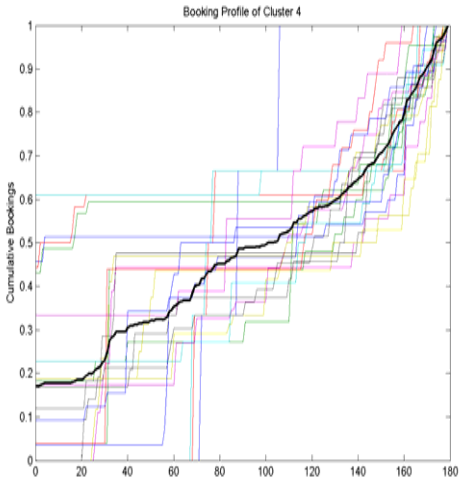
Booking pace curves

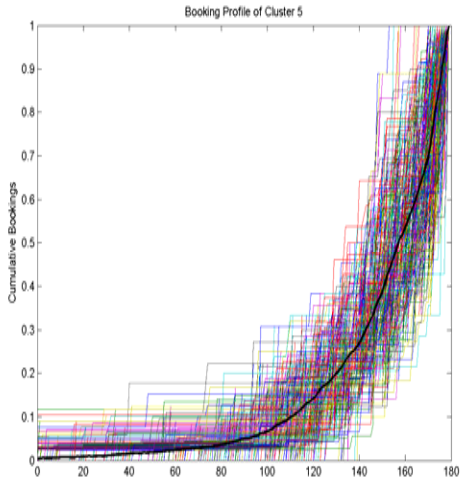
- *5 clusters (derived from 2 years of history)*





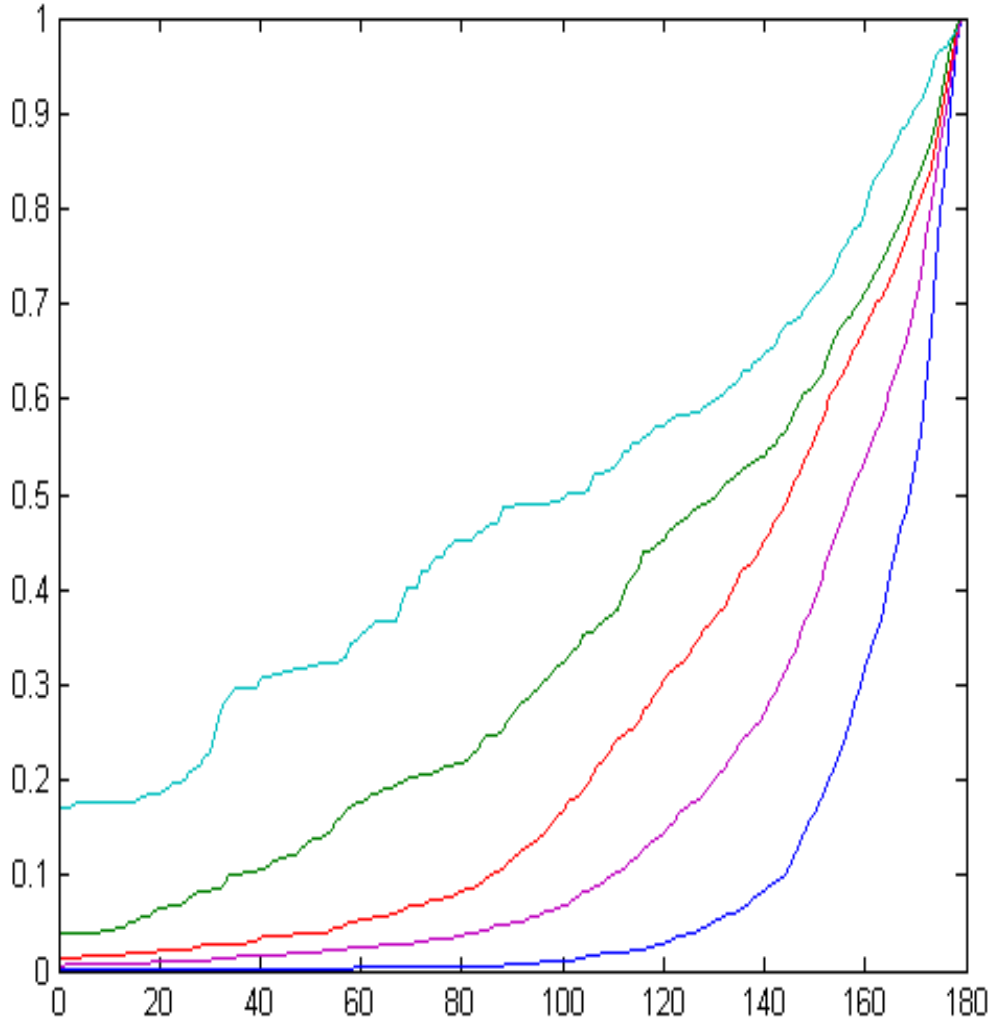






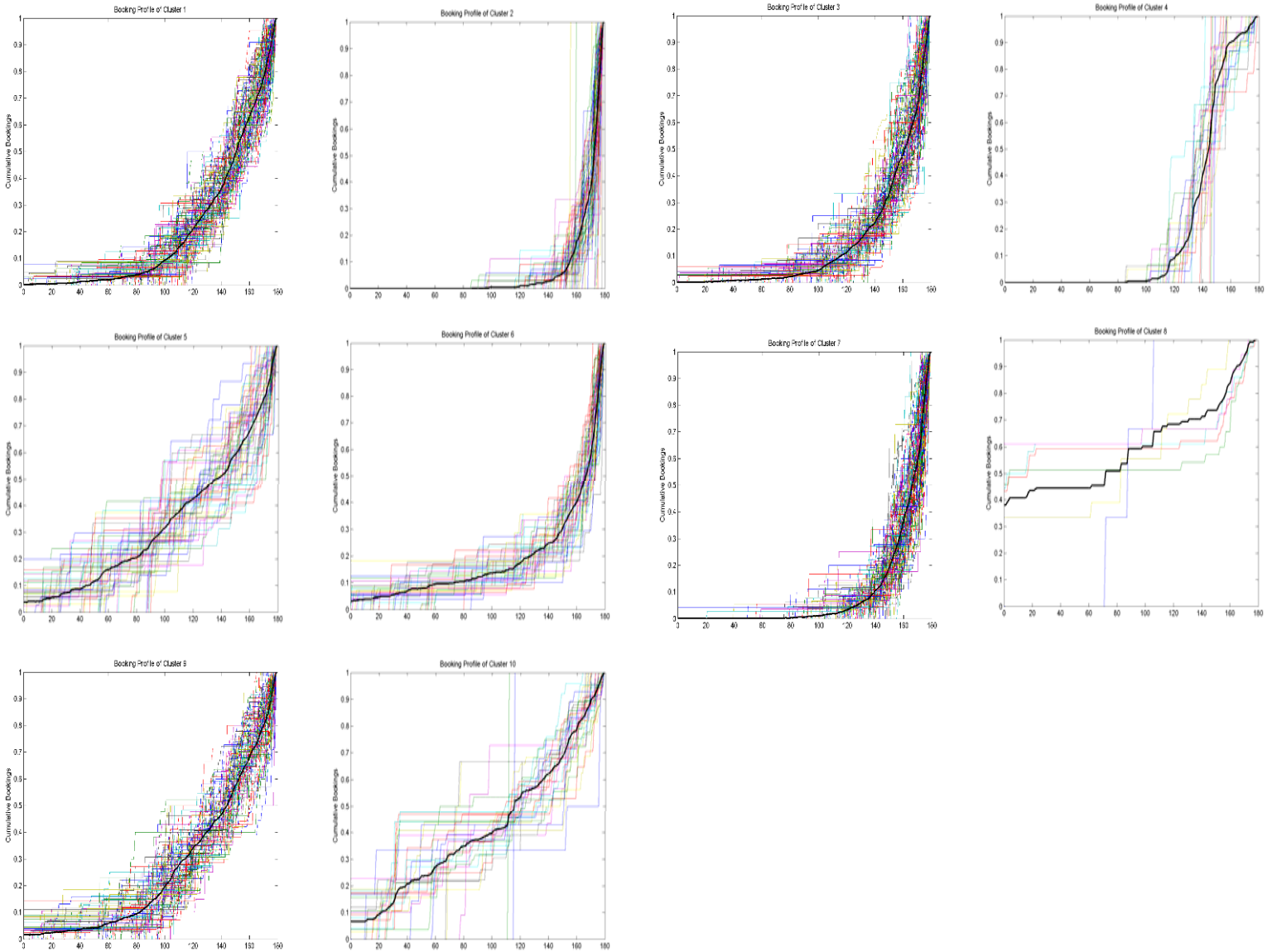
Booking pace curves

- 5 clusters (derived from 2 years of history)



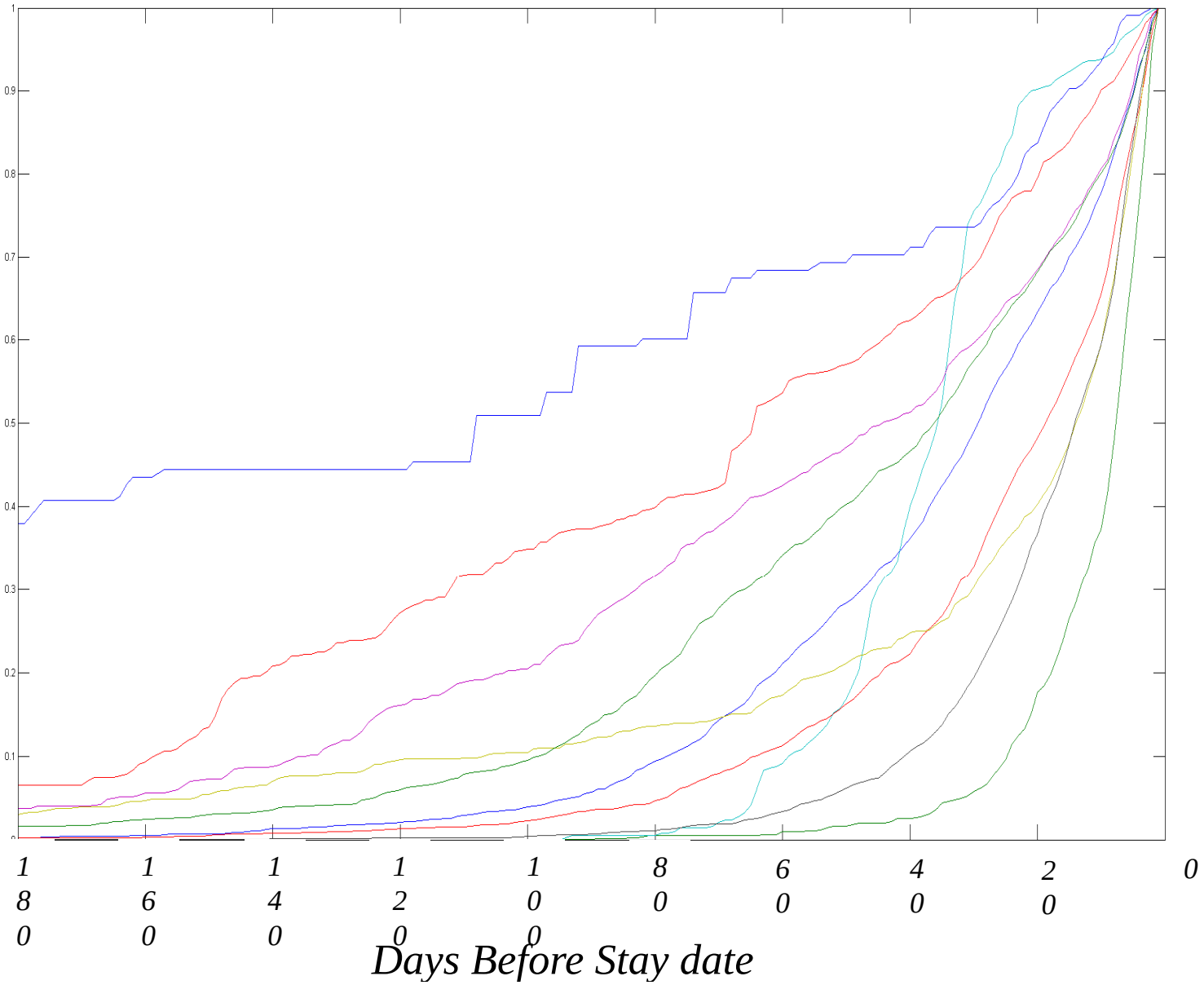
Booking pace curves

- *10 clusters (derived from 2 years of history)*



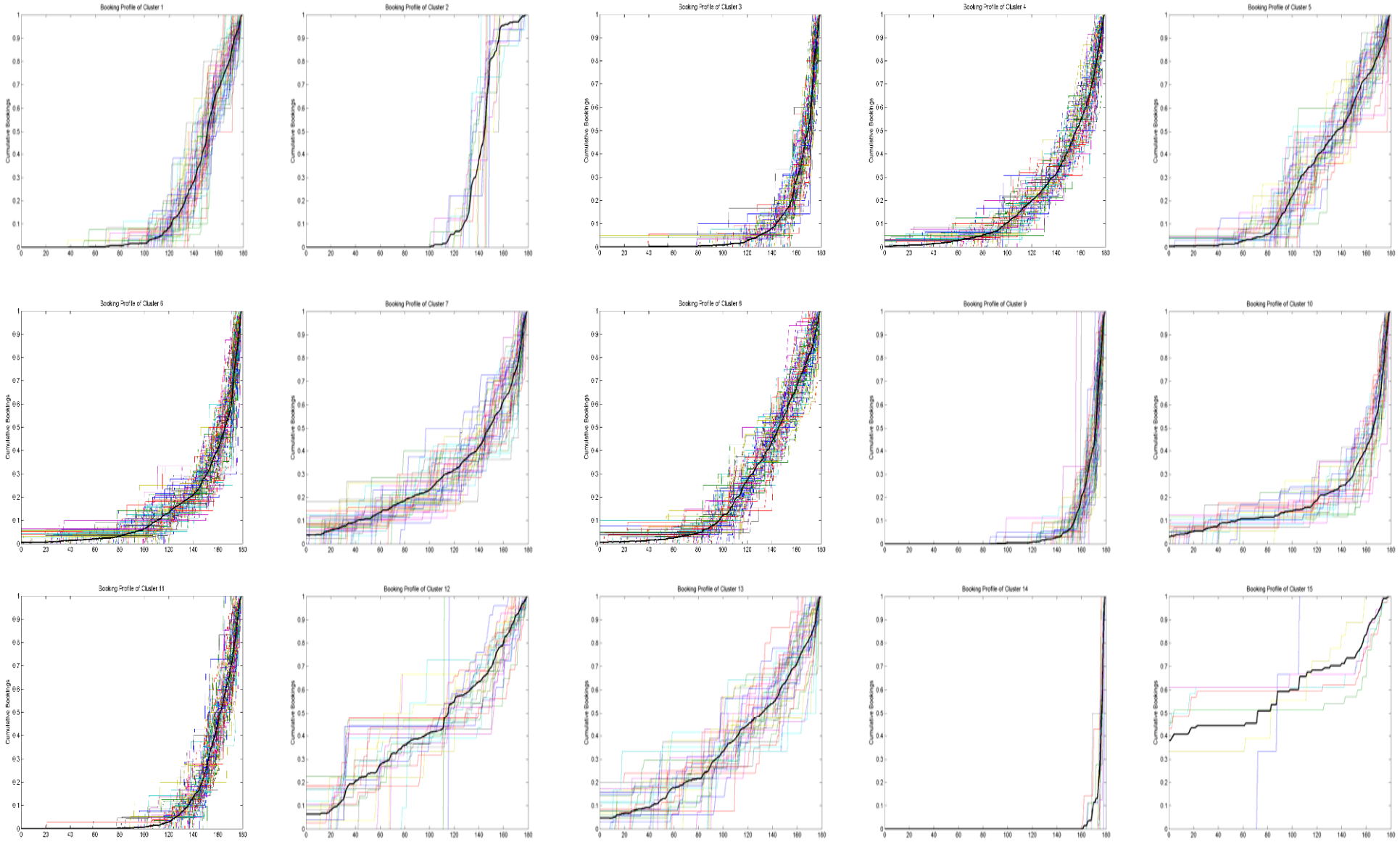
Booking pace curves

- 10 clusters (derived from 2 years of history)



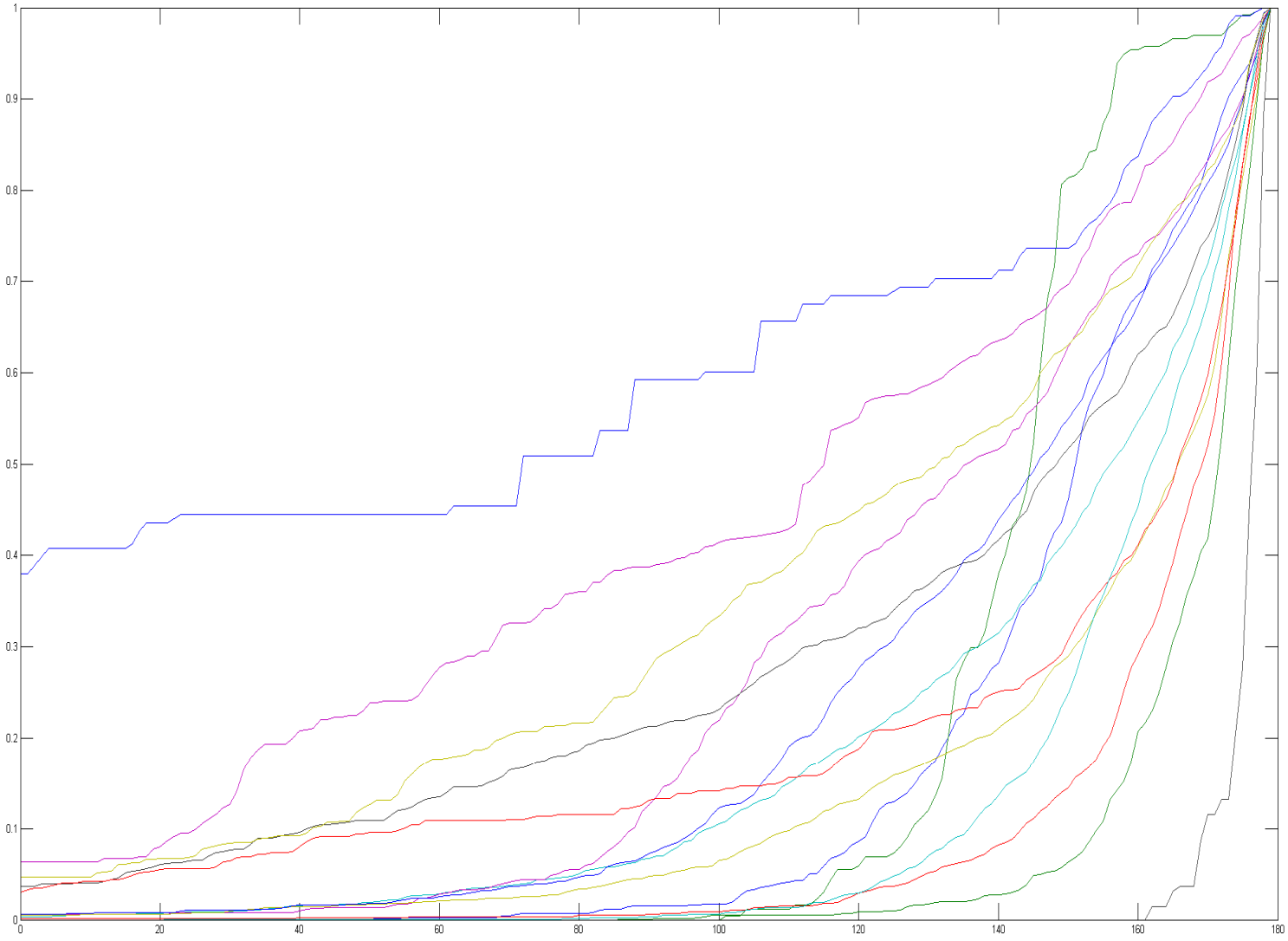
Booking pace curves

- *15 clusters (derived from 2 years of history)*



Booking pace curves

- *15 clusters (derived from 2 years of history)*



Booking pace curves

- The marginal booking pace curve
  - » Most of the time we are going to need the fraction of customers that call in on a particular day. For this we define

$$g_\tau^k = \text{Fraction of customers who call in } \tau \text{ days in advance according to booking pace curve } k \in \{1,...,K\}$$
$$= h_\tau^k - h_{\tau-1}^k$$

  - » We can now compute the expected number of customers who will call in $\tau = T - t$ days in advance according to booking pace curve k:

$$\lambda_{t,T}^k = g_{T-t}^k f_{t,T}$$

  - » Note that the forecast of total bookings $f_{t,T}$ is indexed by t because we keep updating it (but it should not change by much).

Booking pace curves

- The marginal booking pace curve
  - » For example, imagine that we are 20 days before the stay date and that:

$$f_{t,T} = 80 \text{ customers}, \; g^k_{T-t} = g^k_{20} = .03$$

$$\lambda^k_T(20) = (.03)(80)$$

$$= 2.4 \text{ will book on the 20th day before the stay date.}$$

  - » Further assume that

$$\sum_{t'=T-19}^{T} g^k_{T-t'} = .18 \Rightarrow 18 \text{ percent will call in over the period 19, ..., 0}$$

days before the stay date.

  - » This means that

$$\left(\sum_{t'=t+1}^{T} g^k_{T-t'} \right) f_{t,T} = (.18)80 = 14.4$$

are expected to call in between now and the stay date.

Booking process

Predicting the booking process

Predicting the booking pace curves

- We need to estimate which booking pace curve is correct for a given stay date T.
  - » Let C_T^{BP} be a random variable denoting the booking pace curve for stay date T.
  - » Let $H_t = (N_t, N_{t-1}, ...)$ be the history of bookings.
  - » Let

$$q_{t,T}^k = \text{probability that curve } k \text{ is the right curve, after observing data up through day } t \text{ for stay date } T.$$

$$= \text{Prob}[C_T^{BP} = k \mid H_t]$$

  - » Initialize $q_{0,T}^k$ = the fraction of stay-dates where the booking curve fell into cluster k.
  - » Let N_t = the number of pickups on day t. We are going to assume that N_t follows a Poisson distribution.

Predicting the booking pace curves

- Challenge problem for the teams:
  - » We need an updating equation for $q^k_{t,T}$ given:
    - The prior $q^k_{t-1,T}$
    - The observed number of rooms booked on day t, given by N_t
  - » Thus, the problem is to use Bayes' theorem to find the conditional probability

$$q^k_{t,T} = \text{Prob}[C^{BP}_T = k \mid N_t = n, H_{t-1}]$$

given $q^k_{t-1,T}$ and N_t .

Predicting the booking pace curves

- Updating the booking pace curve distribution.
  - » We first need to compute the distribution of N_t. If we condition on the booking pace curve *k*, we know that the mean is

$$\lambda_{t,T}^k = g_{T-t}^k f_{t,T}$$

  - » Assuming that N_t is Poisson, we obtain

$$\text{Prob}[N_t = n] = \frac{\left(\lambda_{t,T}^k\right)^n e^{-\lambda_{t,T}^k}}{n!}$$

  - » Bayes' theorem tells us

$$q_{t,T}^k = \text{Prob}[C_T^{BP} = k \mid N_t = n, H_{t-1}] = \frac{\text{Prob}[N_t = n \mid C_T^{BP} = k]\text{Prob}[C_T^{BP} = k \mid H_{t-1}]}{\text{Prob}[N_t = n]}$$

Predicting the booking pace curves

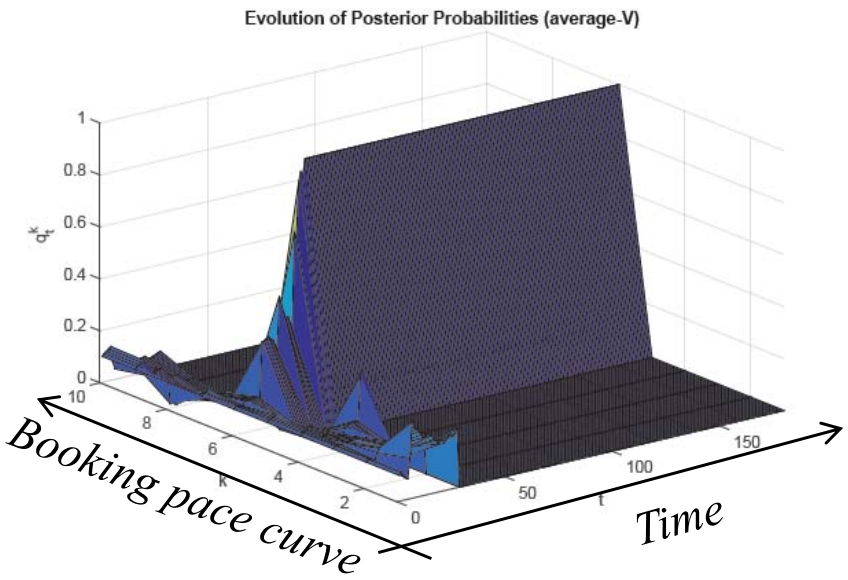
- Updating the booking pace curve distribution.
  - » Substituting in the various probabilities

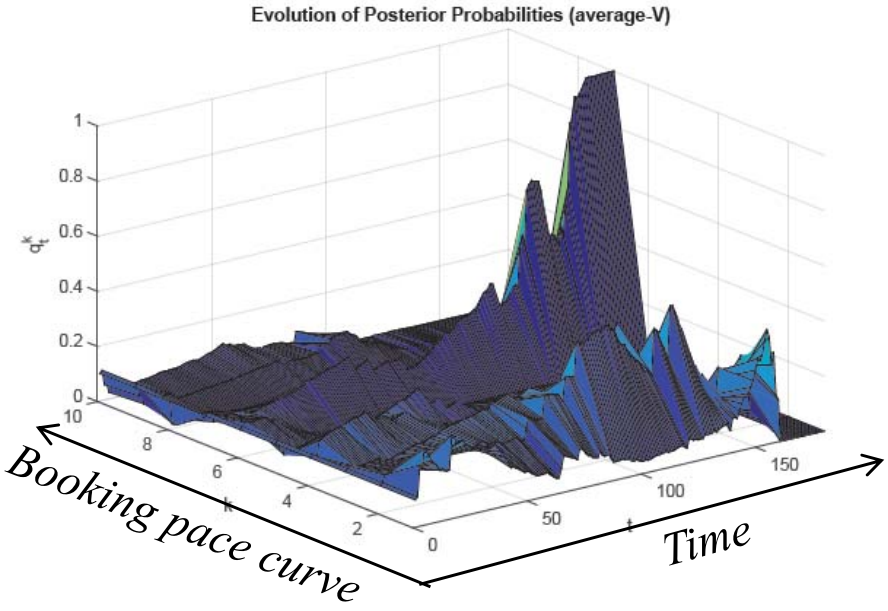
$$q_{t,T}^k = \frac{\text{Prob}[N_t = n \mid C_T^{BP} = k]\text{Prob}[C_T^{BP} = k \mid H_{t-1}]}{\text{Prob}[N_t = n]}$$

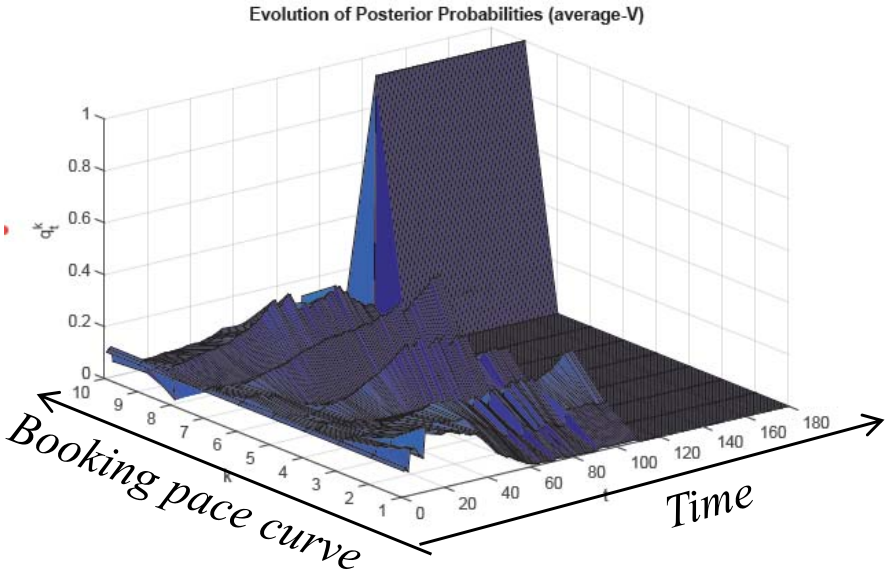
$$= \frac{\frac{\left(\lambda_{t,T}^k\right)^n e^{-\lambda_{t,T}^k}}{n!} q_{t-1,T}^k}{\sum_{k=1}^{K}\left(\frac{\left(\lambda_{t,T}^k\right)^n e^{-\lambda_{t,T}^k}}{n!} q_{t-1,T}^k\right)}$$

Predicting the booking pace curves

- Samples of process of learning booking pace curves
  - » Sometimes we learn the right curve quickly
  - » Sometimes we never learn the right curve

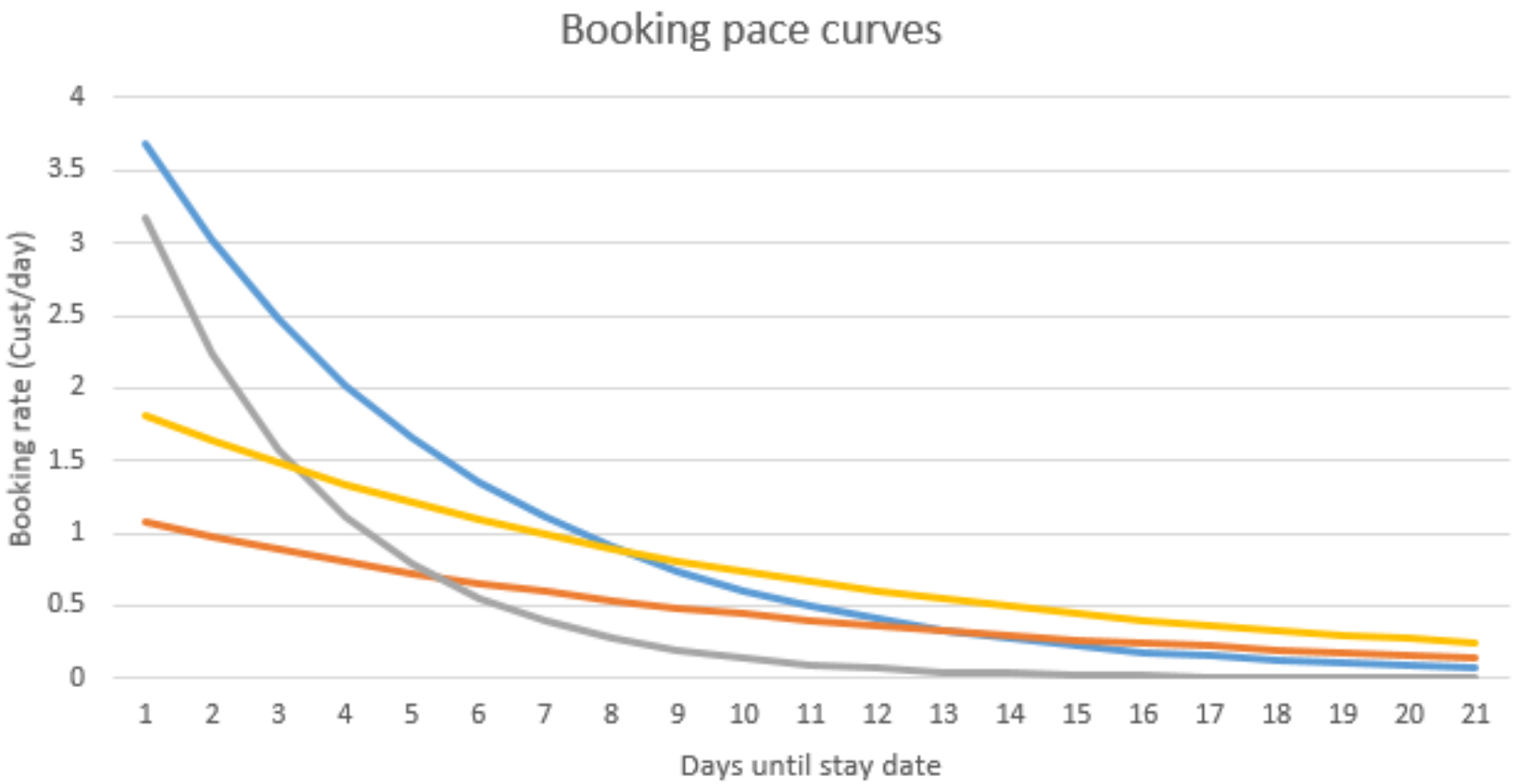






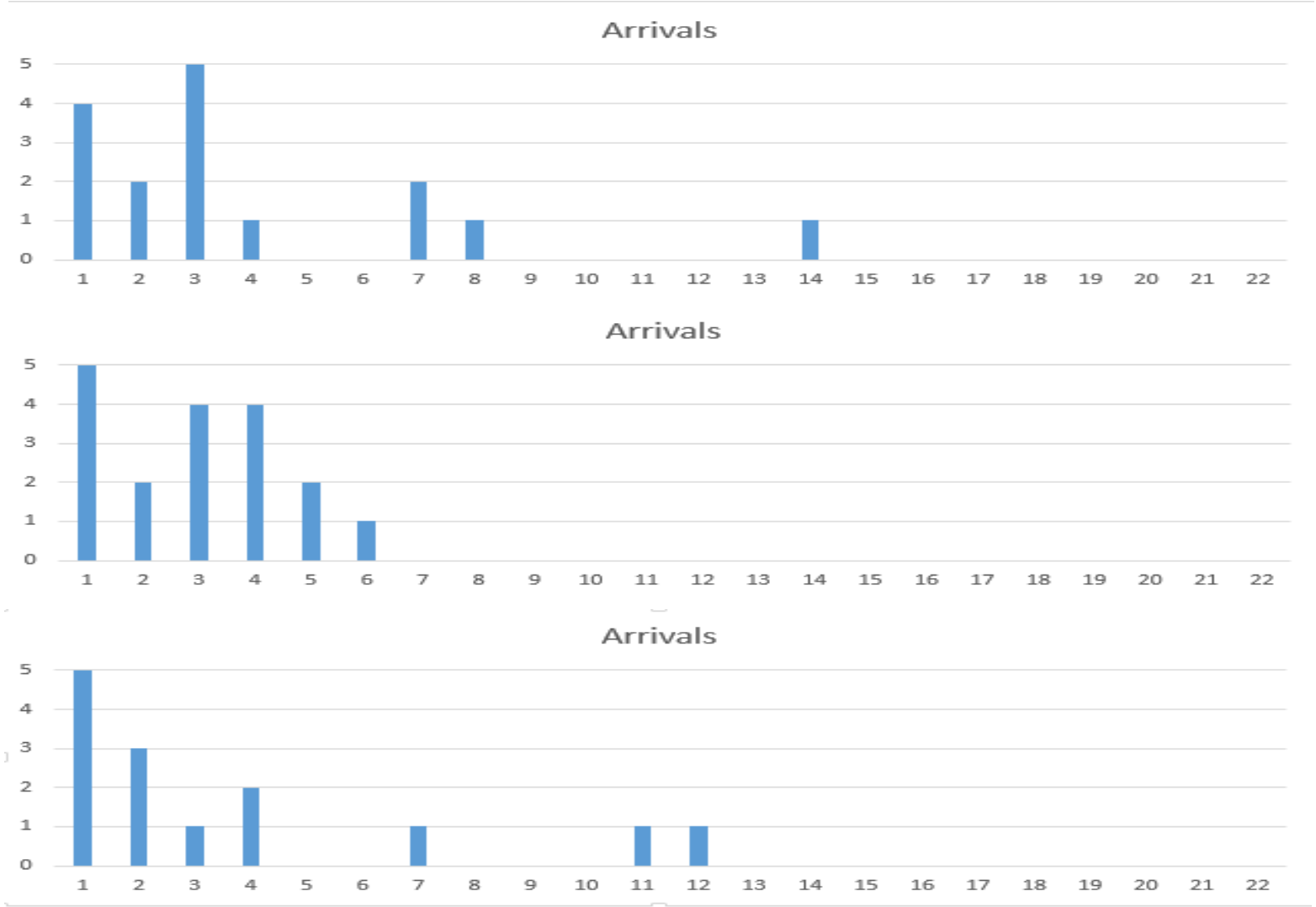
Predicting the booking pace curves

- Booking pace curves



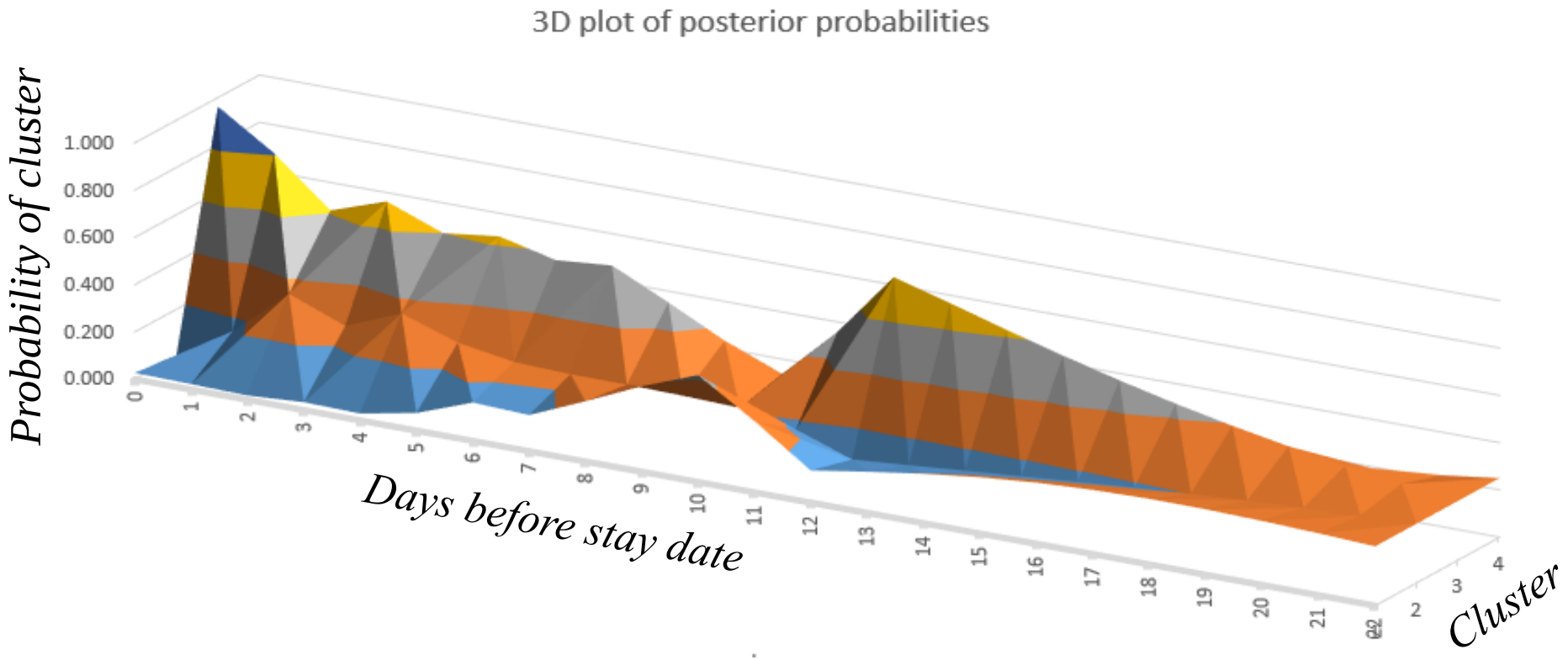
Predicting the booking pace curves

- Some sample paths for bookings



Predicting the booking pace curves

- Posterior probabilities of booking profiles



» Caution: these probabilities are fairly sensitive to the actual sample realizations.

Click here for spreadsheet

Predicting the booking pace curves

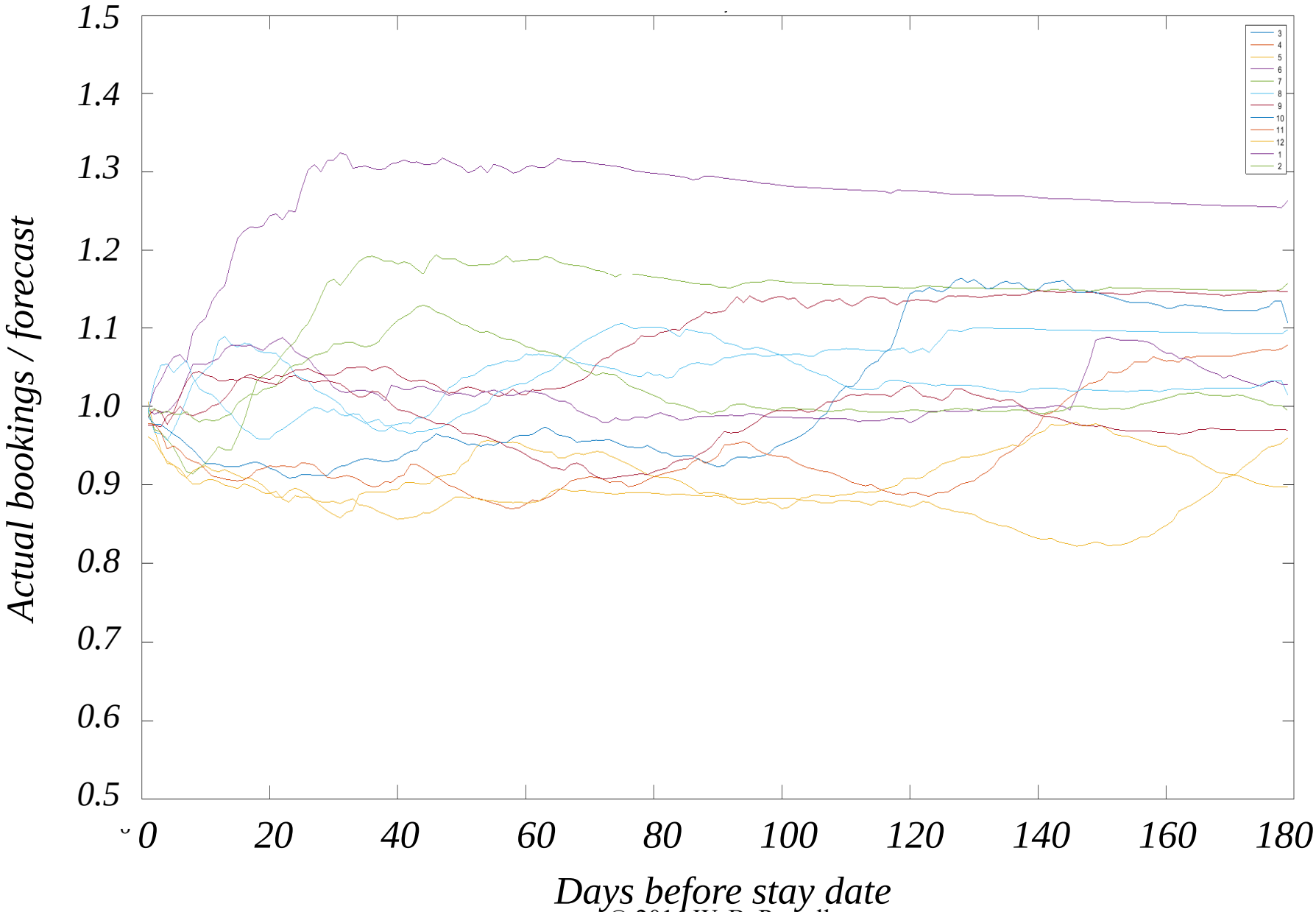
- The booking pace curve
  - » Given the booking pace probabilities, we obtain the expected booking pace at time t, given stay date T:

$$\lambda_{t,T} = \sum_{k=1}^{K} q_{T-t}^{k} \lambda_{t,T}^{k}$$

  - » Now let's create a forecast of the total number of bookings on the state date T, given the history of bookings $H_t = (N_t, N_{t-1}, ...,)$. We have to add the bookings made so far, to the bookings we expected will be made in the future:

$$F_{t,T} = \underbrace{\sum_{t' \le t} N_{t'}}_{\text{Known bookings}} + \underbrace{\sum_{k=1}^{K} q_{t,T}^{k}}_{\text{Probability of booking pace curve } k} \underbrace{\left(\sum_{t'=t+1}^{T} g_{T-t'}^{k} \right)}_{\text{Fraction not yet called in according to booking pace curve } k} \underbrace{f_{t,T}}_{\text{Original forecast of total bookings}}$$

Predicting the booking pace curves



Booking process

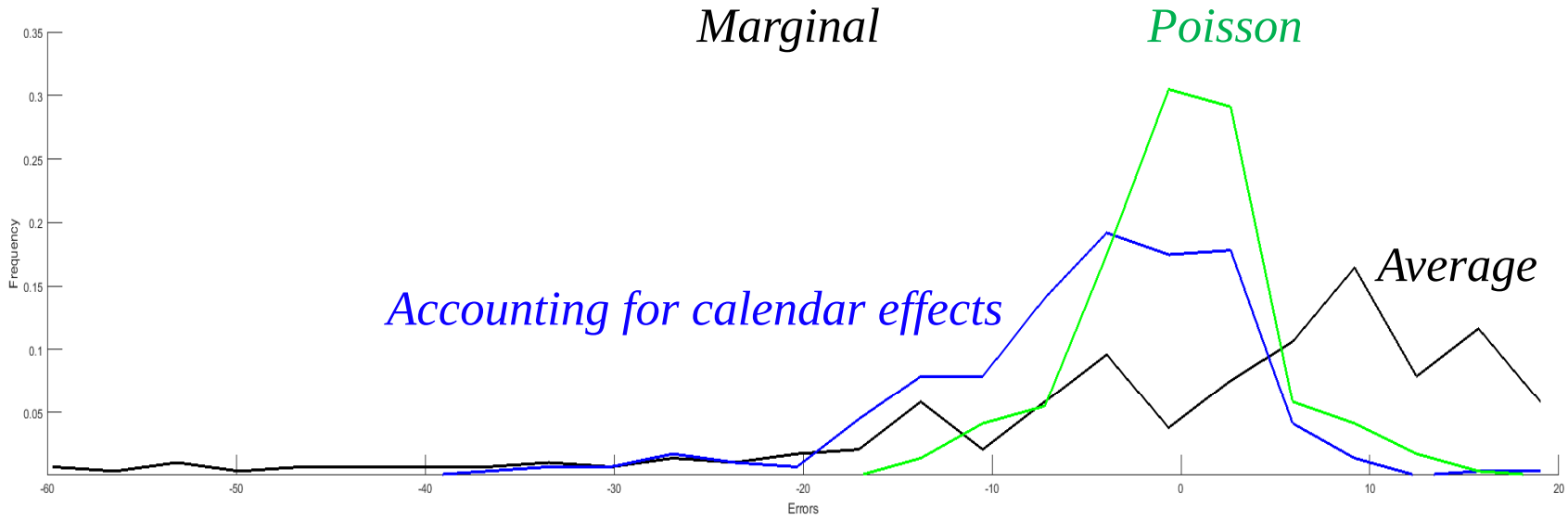
Randomizing arrival process

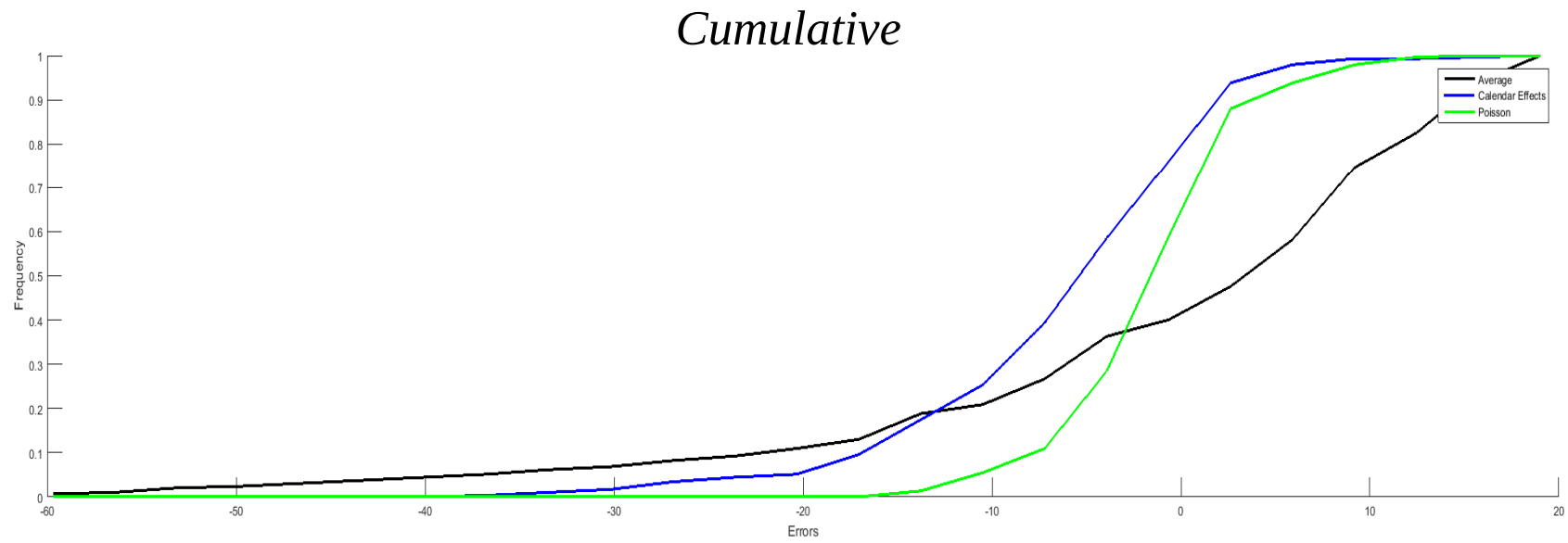
Booking process

- Error distributions:
  - » Ways of estimating the error distribution:
    - Relative to the forecast (presumably made before any bookings have been made, but we can define a forecast at any time up to the stay date).
    - Relative to a base average (which loses the benefit of seasonal adjustments (this would be the worst case).
  - » Best case, with a perfect forecast, is that we should observe an error distribution that follows a Poisson.
    - This assumes we can perfectly predict the *booking rate*.
    - Of course, we cannot perfectly predict the booking rate, so the observed error distribution will show a variance higher than that predicted by a Poisson.

Booking process

- Error distributions





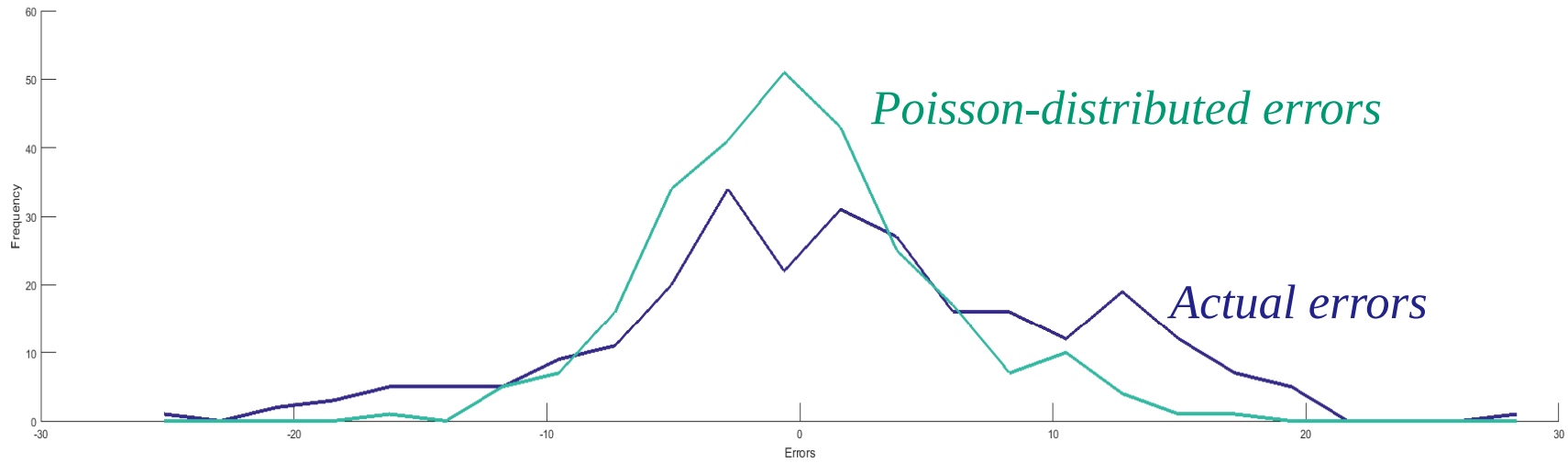
Booking process

- Fixing the variability
  - » We need a booking model that produces the same degree of variability as we observe in real data.
  - » Assuming a Poisson distribution does not introduce enough variability (but we still have to assume Poisson arrivals for our Bayesian updating formula).
  - » We are going to introduce additional variability by randomizing the arrival rate λ_t.
  - » You will work this out on your problem set!

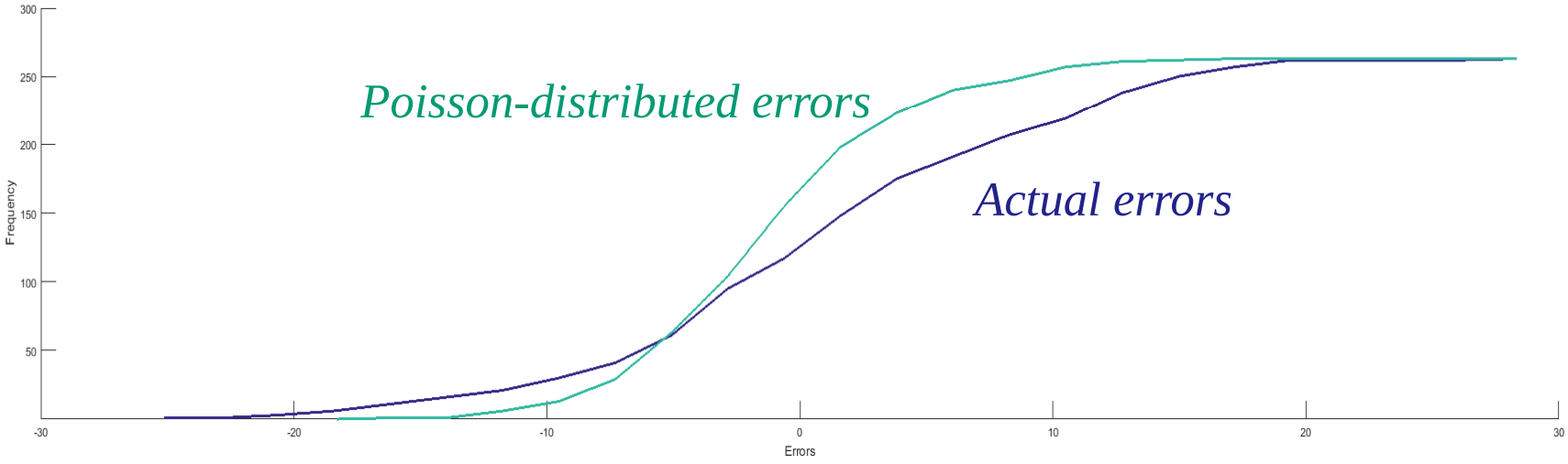
Booking process

- Actual error compared to Poisson error

Marginal

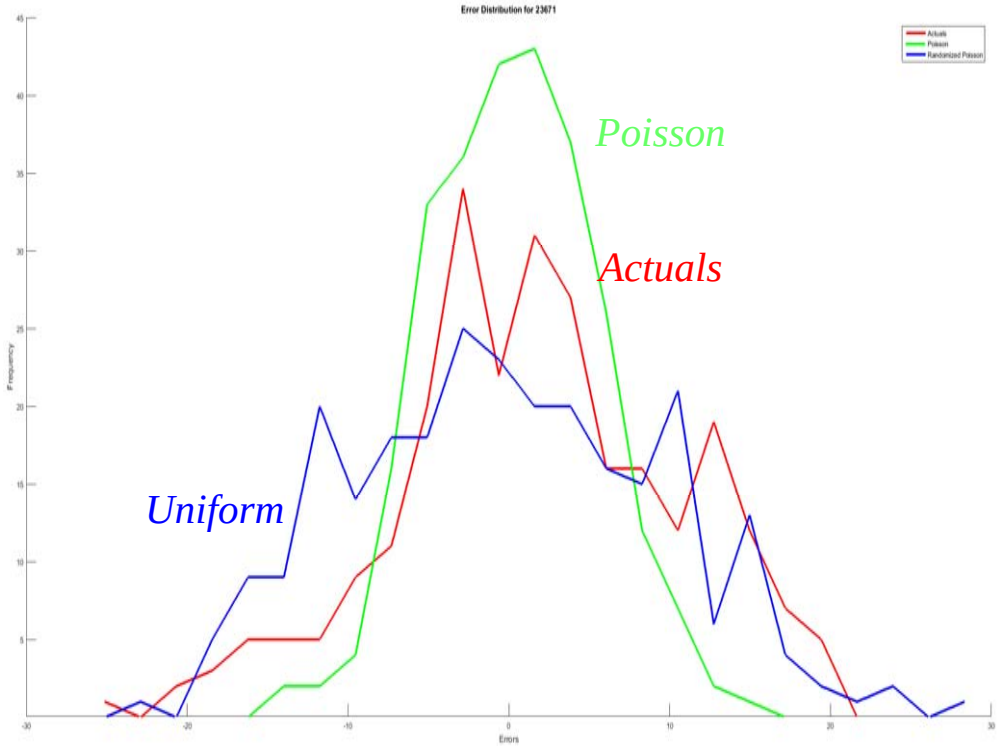


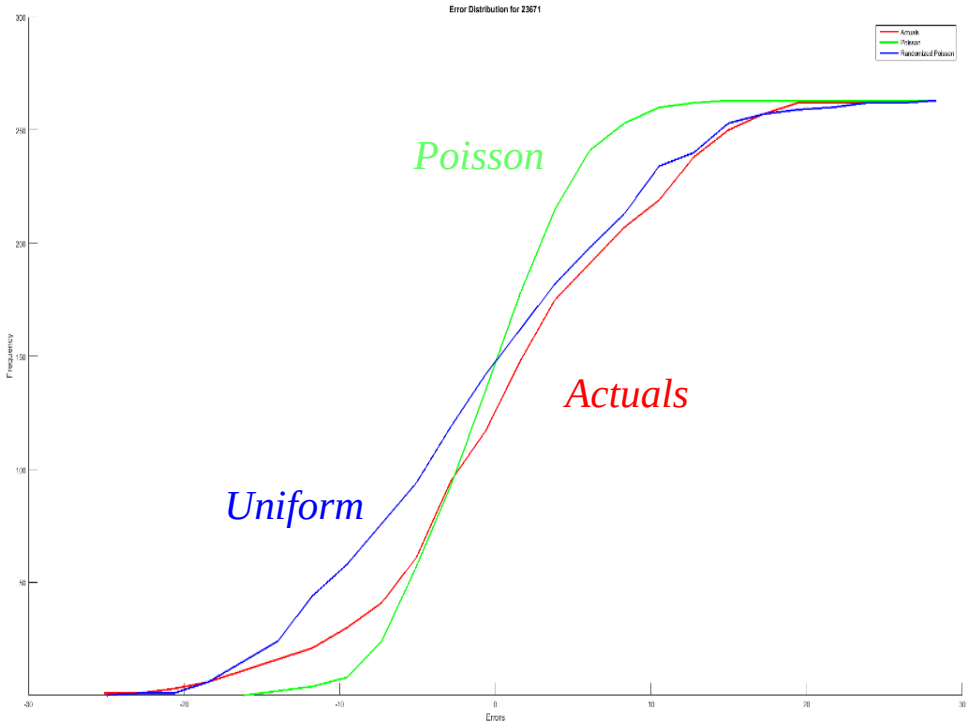
Cumulative



Booking process

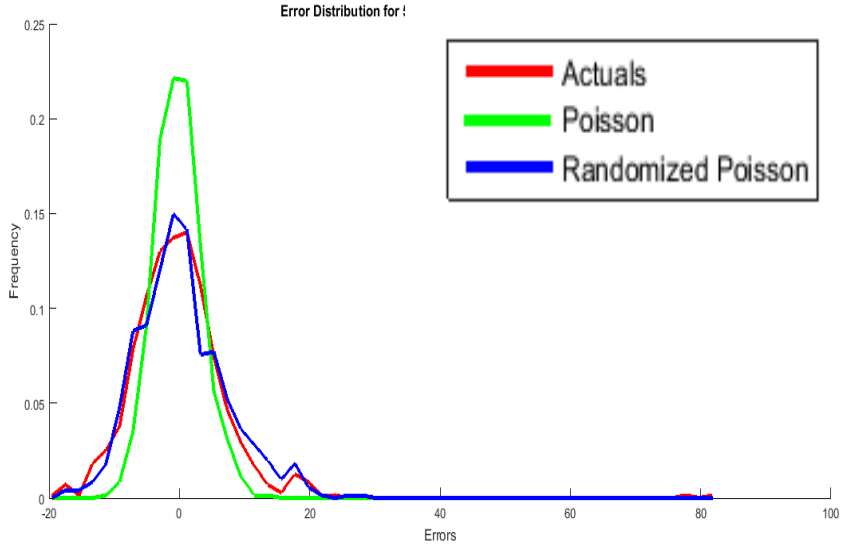
- Fix using uniformly distributed λ

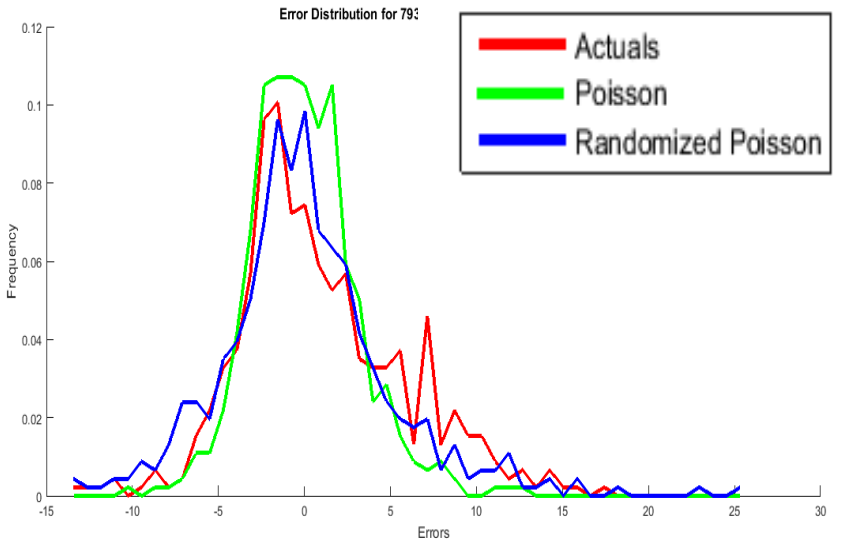


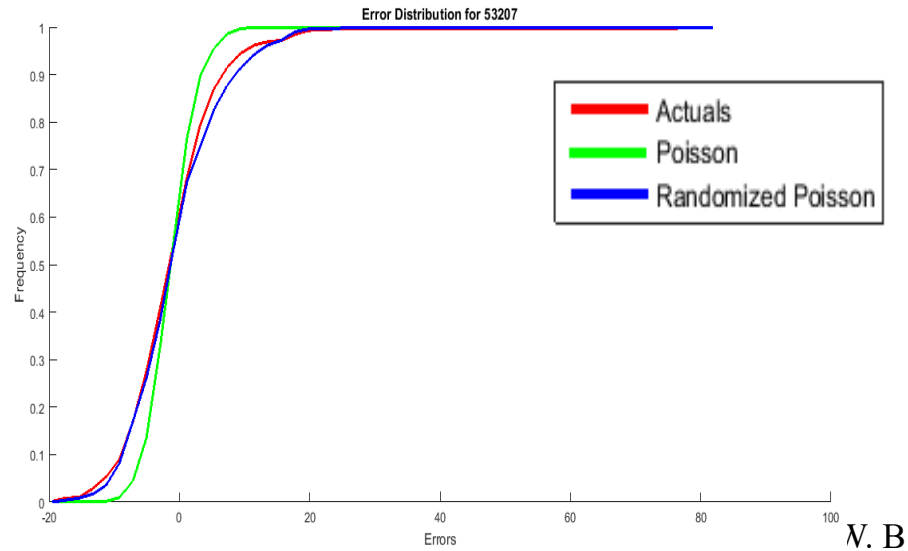


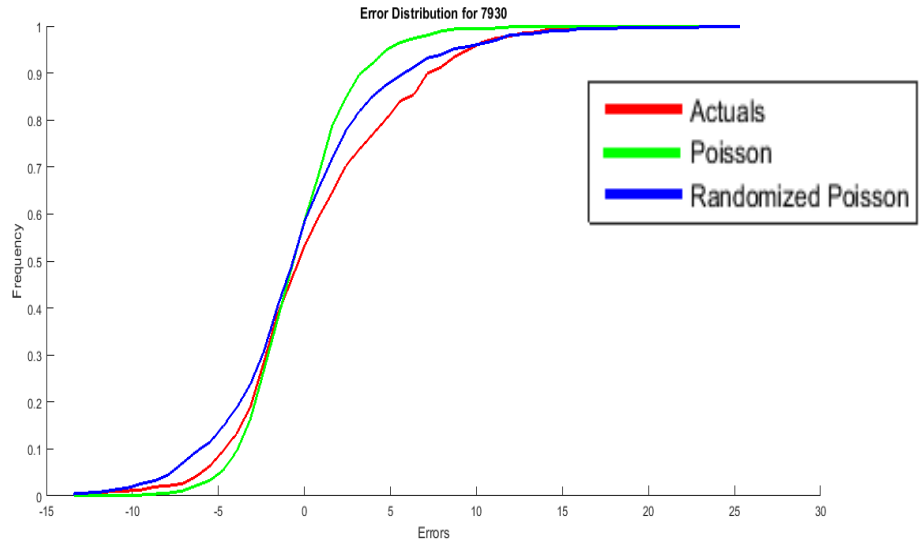
Booking process

- Fix using beta-distributed λ









N. B

Designing policies

Revenue management

- The problem
  - » We want to maximize revenue by adjusting the base price, possibly on a daily basis.
  - » We have to make these decisions in the presence of different types of uncertainty.
  - » We are then going to compare two classes of policies.

Revenue management

- We are considering two classes of policies:
  - » A policy function approximation based on expert judgment:
    - This is a set of rules designed by an experienced (and award winning) revenue manager, Zak Ali, to price a policy we are calling "Zak's rules."
    - These rules are being coded by our own Zach (Zach Koerbl) as part of his senior thesis. Zach's job is to optimize Zak's rules to make them perform as well as possible.
  - » A policy based on value functions derived using dynamic programming
    - This is the "high tech" approach.
  - » Both policies will be tested in a realistic simulator that captures the actual booking process.

Designing policies

Zak's rules – industry practice

A practical policy

yieldPlanet | Maximizing hotel revenue. Simply

English Login

Products | Become a partner | YieldPlanet | Blog | Contact us

We help **hotels** manage their **online channels** from a single **point of control**

YieldPlanet provides online Distribution and Revenue Management solutions for more than 2500 hotels of all types and sizes around the world. With advanced capabilities built into easy to use tools, we will help you to increase occupancy, grow revenue and maximize profit.

Discover our solutions

Apartments, vacation | Independent hotels | Hotel chains

A practical policy

- Basic idea
  - » Hotels have a range of room layouts as well as reservation plans, all with their own rates. But one is known as the "BAR" rate (best available rate).
  - » We assume that the hotel has worked out a reasonable solution of how the different room rate plans should be priced relative to the BAR rate.
  - » The "practical policy" is to create a lookup table policy that maps the "state" (what we know) to an adjustment of the BAR rate.
    - State variable requires using expert judgment to identify important factors
    - State variables require combining numerical and qualitative factors (such as competition).

A practical policy

- Zak's rules
  - » Developed by award-winning Revenue Manager Zak Ali from YieldPlanet
  - » Lookup table approach to price-setting
  - » Derived from 6 binary factors and time to stay date
    - Based on intuition and refined through observations
    - Not gradient-based: same effect regardless of small changes within individual factors
    - Minimal look-back: Depends heavily on comparing data from "this year" (TY) to "last year" (LY).

A practical policy

- ABOBTY vs. ABOBLY
  - » ABOB = Actual Business on the Books
    - Refers to the number of reservations made for the stay date
  - » Compare to the equivalent date last year *at the same number of days prior to arrival*
- APRICETY vs. APRICELY
  - » APRICE = Actual Price
  - » Takes into account the current BAR (Best Available Rate), determined from business booked and forecasting relative to budget

A practical policy

- Average Weekly Pickup TY vs. LY
  - » The average rate that business is booked for stay date
    - Use data from the previous 7 days
- FTY vs. FLY
  - » Forecast this year is total final occupancy forecasted in terms of number of rooms booked for stay date
    - Based on (more quantitatively derived) forecasting estimates calculated by dividing customers into segments
    - Done by separate forecasting team
  - » Forecast last year is final realized occupancy

A practical policy

- COMPPRICE vs. MYPRICE
  - » COMPPRICE = Prices of competition
  - » How do we determine competitors?
    - Difficult question to answer
    - Geography, Star Rating, Similar Price Range?
    - Historically been based on qualitative/anecdotal evidence
- Is Price Elastic?
  - » Elasticity: % Change in Quantity / % Change in Price
  - » In practice, harder to determine
    - Use price range of competition for comparable room type/rate plan combination
    - Competition not always available (or could be irrelevant due to special convention or sold out status)

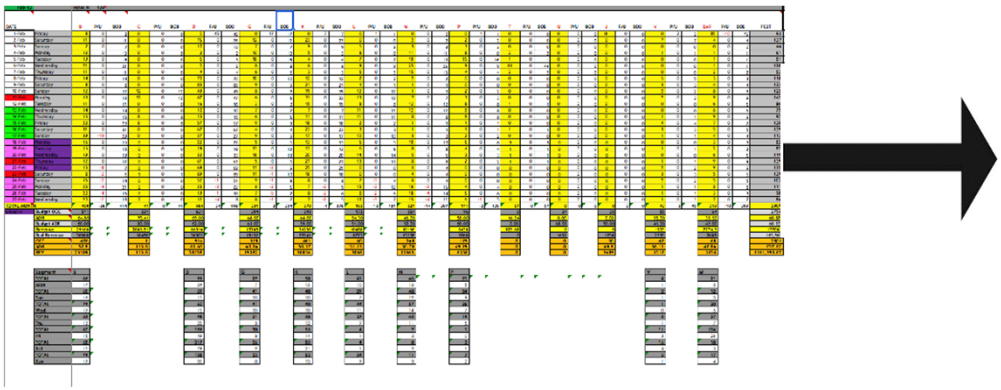
A practical policy

- Pricing spreadsheet
  - » First six columns determine “state”
  - » Last column is recommended price above/below “BAR” rate

1	ABOBTY/ABOBLY	APRICETY/APRICELY	AVGPICKTY/AVGPICKLY(WEEKLY)	FLY/FTH	MYPRICE/COMPPRICE	ELASTIC(YES/NO)	PERCENTAGE
3	1	1	1	1	1	1	20.00%
4	1	1	1	1	1	-1	-11.00%
5	1	1	1	1	-1	1	16.00%
6	1	1	1	1	-1	-1	-3.00%
7	1	1	1	-1	1	1	9.00%
8	1	1	1	-1	1	-1	1.00%
9	1	1	1	-1	-1	1	6.00%
10	1	1	1	-1	-1	-1	-8.00%
11	1	1	-1	1	1	1	5.50%
12	1	1	-1	1	1	-1	-15.00%
13	1	1	-1	1	-1	1	14.00%
14	1	1	-1	1	-1	-1	-8.00%
15	1	1	-1	-1	1	1	12.00%
16	1	1	-1	-1	1	-1	-2.00%
17	1	1	-1	-1	-1	1	12.00%
18	1	1	-1	-1	-1	-1	-8.00%
19	1	-1	1	1	1	1	15.00%
20	1	-1	1	1	1	-1	-8.00%
21	1	-1	1	1	-1	1	7.00%
22	1	-1	1	1	-1	-1	-8.00%
23	1	-1	1	-1	1	1	7.50%

A practical policy

- Example
  - » Assume we are 5 days away from stay date



Sample data

	2014	2015
ABOB	6423.42	7302.64
BAR	126.4	130.2
Avg Pickup	4.32	2.43
Forecast	7800	8400
Actual	7645.23	n/a
Comp. Price	140.5	145.5
Elasticity	1.43	1.37

  - » Use numbers to compute state
    - If ABOBTY > ABOBLY then $S_{t1} = 1,$ else $= -1$
    - If APRICETY > APRICELY then $S_{t2} = 1,$ else $= -1$
    - Pickup TY < Pickup LY then $S_{t3} = 1,$ else $= -1$
    - FTY > FLY then $S_{t4} = 1,$ else $= -1$
    - COMPRICE > MYPRICE then $S_{t5} = 1,$ else $= -1$
    - Unit demand is elastic [>1] then $S_{t6} = 1,$ else $= -1$

$$S_t = (S_{t1}, S_{t2}, S_{t3}, S_{t4}, S_{t5}, S_{t6}) = \text{"state" at time } t$$

A practical policy

- The lookup table policy for Zak's rules:

$$P^{\pi}(S_t) = p^{BAR} \cdot \left(1 - \delta\, p^{ZAK}(S_t)\right)$$

S_t

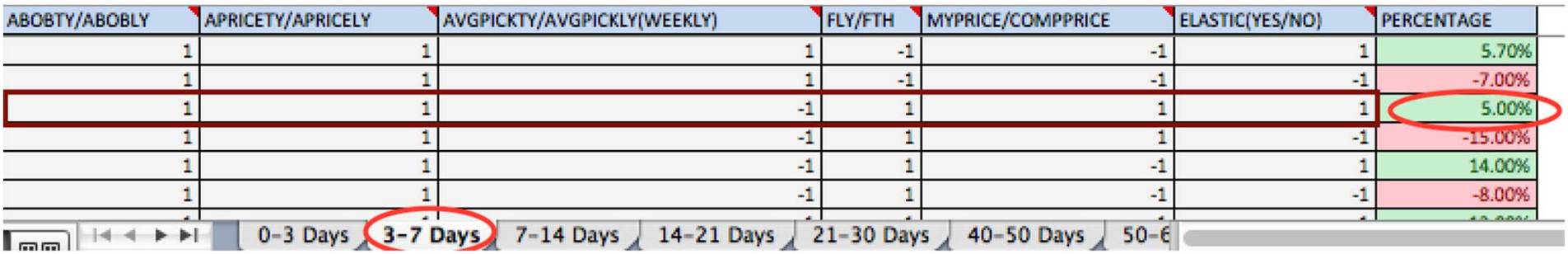
1	ABOBTY/ABOBLY	APRICETY/APRICELY	AVGPICKTY/AVGPICKLY(WEEKLY)	FLY/FTH	MYPRICE/COMPPRICE	ELASTIC(YES/NO)	PERCENTAGE
3	1	1	1	1	1	1	20.00%
4	1	1	1	1	1	-1	-11.00%
5	1	1	1	1	-1	1	16.00%
6	1	1	1	1	-1	-1	-3.00%
7	1	1	1	-1	1	1	9.00%
8	1	1	1	-1	1	-1	1.00%
9	1	1	1	-1	-1	1	6.00%
10	1	1	1	-1	-1	-1	-8.00%
11	1	1	-1	1	1	1	5.50%
12	1	1	-1	1	1	-1	-15.00%
13	1	1	-1	1	-1	1	14.00%
14	1	1	-1	1	-1	-1	-8.00%
15	1	1	-1	-1	1	1	12.00%
16	1	1	-1	-1	1	-1	-2.00%
17	1	1	-1	-1	-1	1	12.00%
18	1	1	-1	-1	-1	-1	-8.00%
19	1	-1	1	1	1	1	15.00%
20	1	-1	1	1	1	-1	-8.00%
21	1	-1	1	1	-1	1	7.00%
22	1	-1	1	1	-1	-1	-8.00%
23	1	-1	1	-1	1	1	7.50%

A practical policy

- Zak's rules:
  - » Use binary factors and number of days prior to arrival (0-3, 4-7, 8-14, etc.) to determine percentage increase/decrease in price
  - » Price changes become more dramatic closer to stay date
    - -20% to 20% range in lookup table corresponding to 0-3 days prior to arrival
    - -8% to 8% range on decision tree corresponding to 71-90 days prior to arrival
  - » The numbers themselves are more based on intuition rather than quantitative analysis
    - Percentages are typically integer increases/decreases, rather than more precise numerical estimates

A practical policy

- Example (cont'd)
  - » Lookup table relative to 4-7 days away from the stay date



ABOBTY/ABOBLY	APRICETY/APRICELY	AVGPICKTY/AVGPICKLY(WEEKLY)	FLY/FTH	MYPRICE/COMPPRICE	ELASTIC(YES/NO)	PERCENTAGE
1	1	1	-1	-1	1	5.70%
1	1	1	-1	-1	-1	-7.00%
1	1	-1	1	1	1	5.00%
1	1	-1	1	1	-1	-15.00%
1	1	-1	1	-1	1	14.00%
1	1	-1	1	-1	-1	-8.00%

0-3 Days | 3-7 Days | 7-14 Days | 14-21 Days | 21-30 Days | 40-50 Days | 50-6

  - » Conclusion: Take BAR and increase price by 5%
  - » Lookup table rests complete on binary node values
    - Note that if we had Elasticity of 1.01 that changes to 0.99, the percentage adjustment changes from 5% to -15%

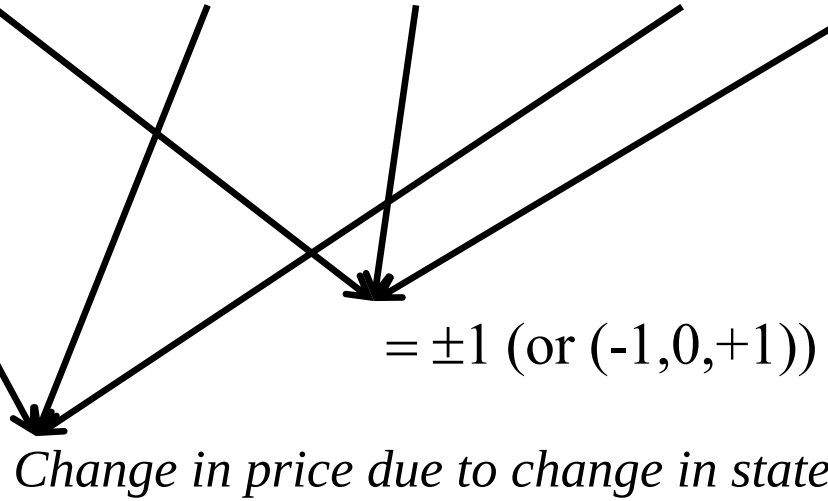
A practical policy

- Discussion
  - » Strengths:
    - Simple, easy to understand, easy to compute
    - Easy to add new factors
    - Considerable amount of expert judgment in the choice of how to adjust the rates
  - » Weaknesses:
    - The "state" of the system is highly aggregated – simply binary classification based on relation between this year and last
    - Despite this aggregation, there are quite a few parameters that have to be specified by judgment ($2^6 = 64$), doubling with each additional factor introduced into the state.
    - Yet even this lookup table is ignoring factors built into our booking model (seasonality, booking pace curves).

A practical policy

- An alternative policy
  - » Just to illustrate the difference between lookup tables and parametric policies, consider the policy:

$$P^{\pi}(S_t) = p^{BAR}\left(\theta^{ABOB}S_t^{ABOB} + \theta^{APRICE}S_t^{APRICE} + \theta^{ABOB}S_t^{ABOB} + \ldots\right)$$



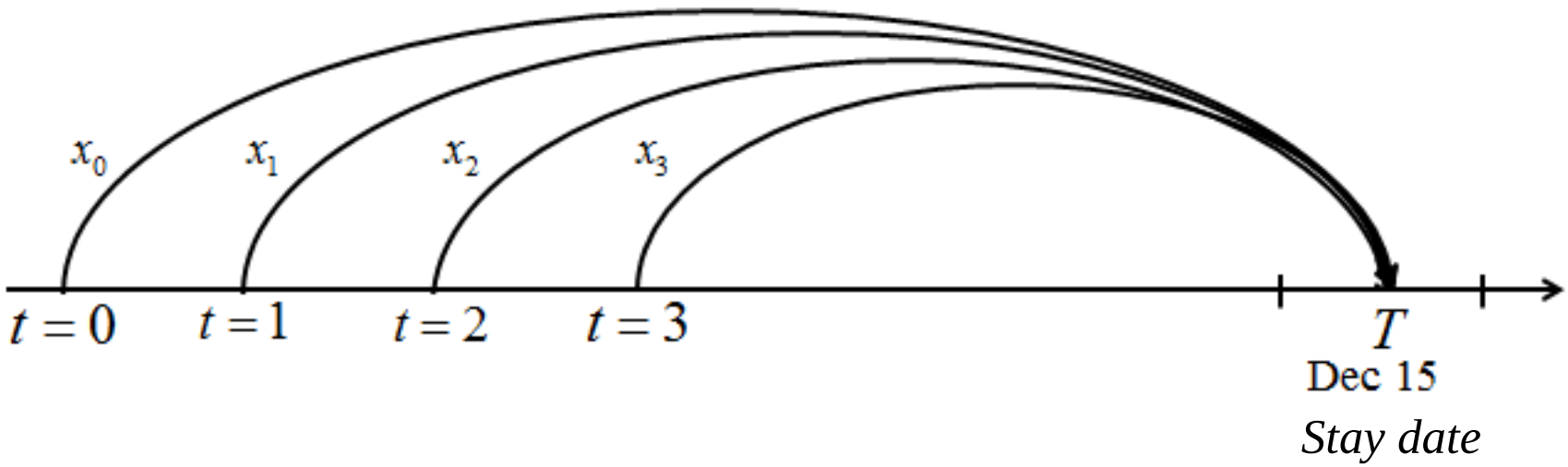
  - » Now there are 6 parameters to specify, rather than 2^6. But you have to live with a linear model.

Designing policies

A direct lookahead policy

Dynamic programming

- Recall the lagged ordering problem:
  - » We can order x_t now to arrive at time T in the future.
  - » With the hotel problem, we do not directly control the quantity x_t; now, we control the price p_t that indirectly influences how many rooms are booked.



Dynamic programming

- Dynamic programming formulation
  - » Let's start by modeling a single state variable:

R_t = Total number of rooms that have been booked on day t and earlier.

  - » This allows us to write Bellman's equation as

$$V_t(R_t) = \max_{p_t} \mathbb{E}_\theta \mathbb{E}_{D_{t+1}|\theta} \left(p_t \underbrace{D_{t+1}(p_t \mid \theta)}_{\text{Random demand on day t+1 given the price we set at the end of day t.}} + V_{t+1}(\underbrace{R_t + D_{t+1}(p_t \mid \theta)}_{\text{Total rooms booked at the end of day t+1}}) \right)$$

θ = Booking pace curve, demand rate

Dynamic programming

- Dynamic programming formulation

$$V_t(R_t) = \max_{p_t} \mathbb{E}_{\theta}\mathbb{E}_{D_{t+1}|\theta}\left(p_t D_{t+1}(p_t \mid \theta) + V_{t+1}(R_t + D_{t+1}(p_t \mid \theta))\right)$$

Poisson distribution given the demand, and given the booking pace curve.

Expectation over:
Different booking pace curves with prob $q^k_{t,T}$
Different values of the forecasted demand

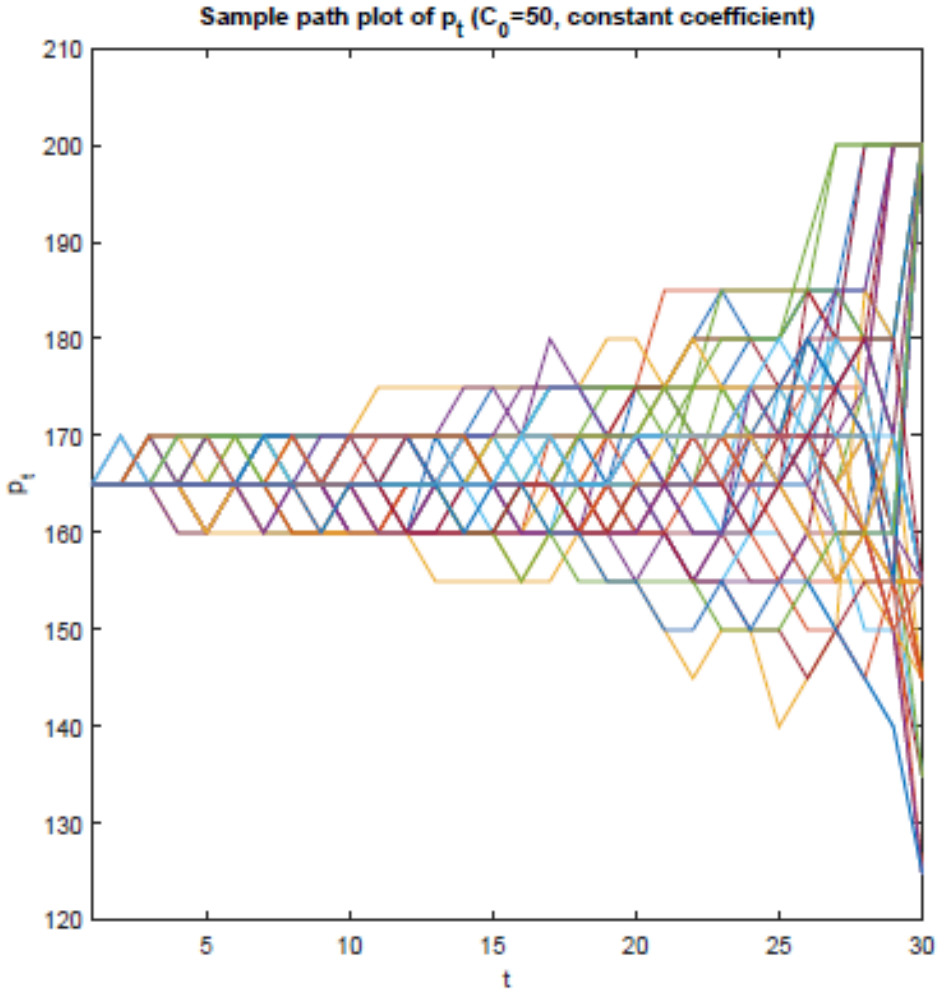
  - » This is fairly easy to execute in a backward dynamic programming recursion.
  - » For now we are ignoring the uncertainty in the demand response.

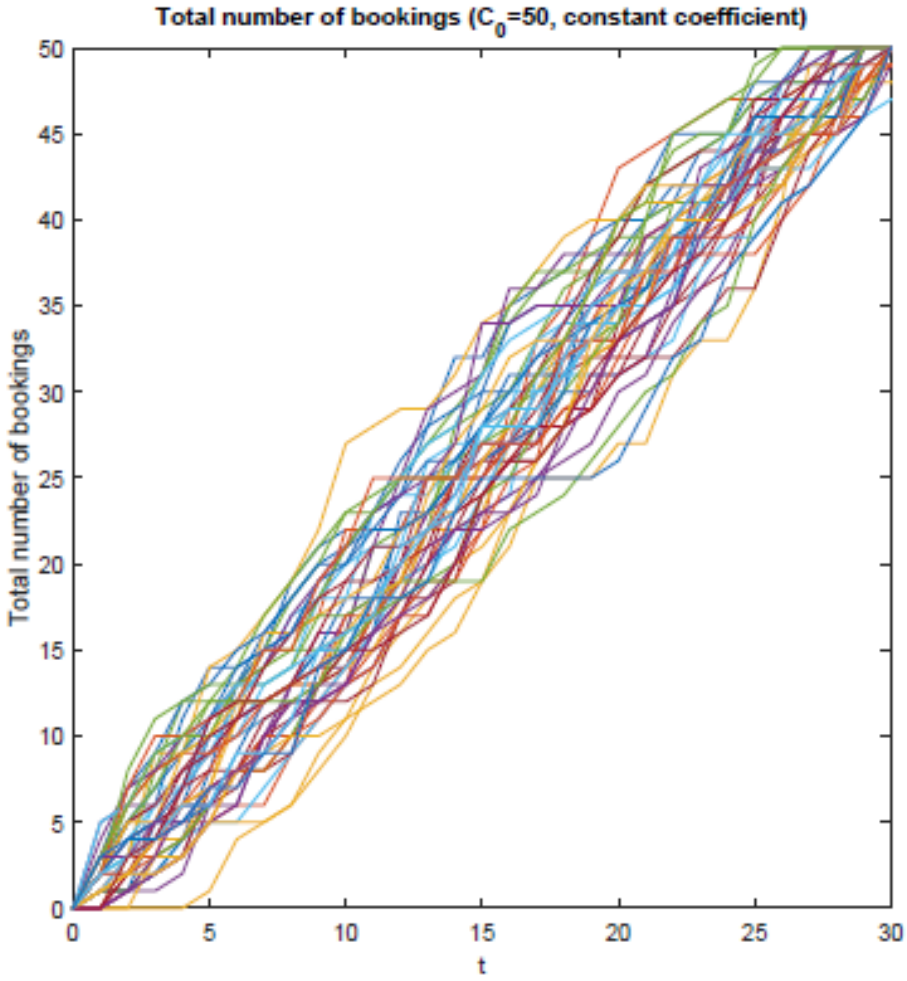
Dynamic programming

- In the slides that follow
  - » We show sample paths of prices, and the corresponding sample paths of the total rooms booked.
  - » They are shown for three hotel sizes: 50, 100 and 150 rooms.
  - » Notice the drop in prices required to fill the larger hotels.
  - » Also notice how the prices are varying as we observe the total booking rate.

Dynamic programming

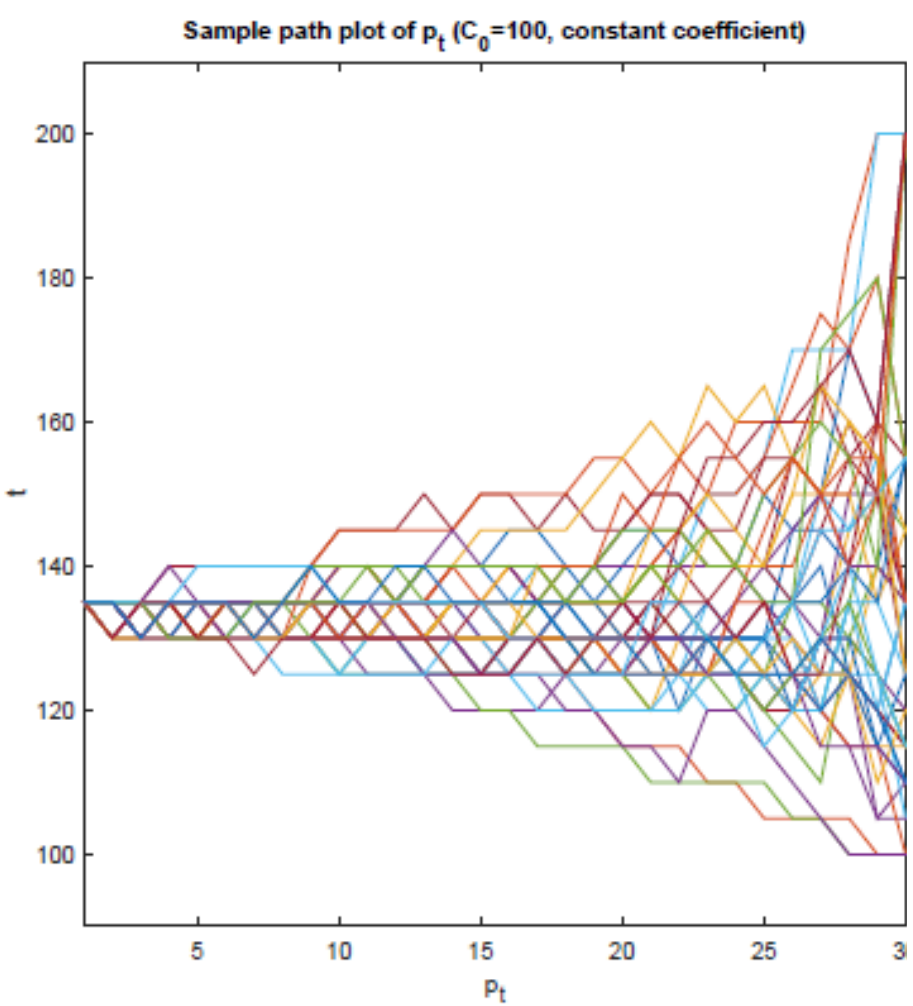
- Capacity = 50 rooms

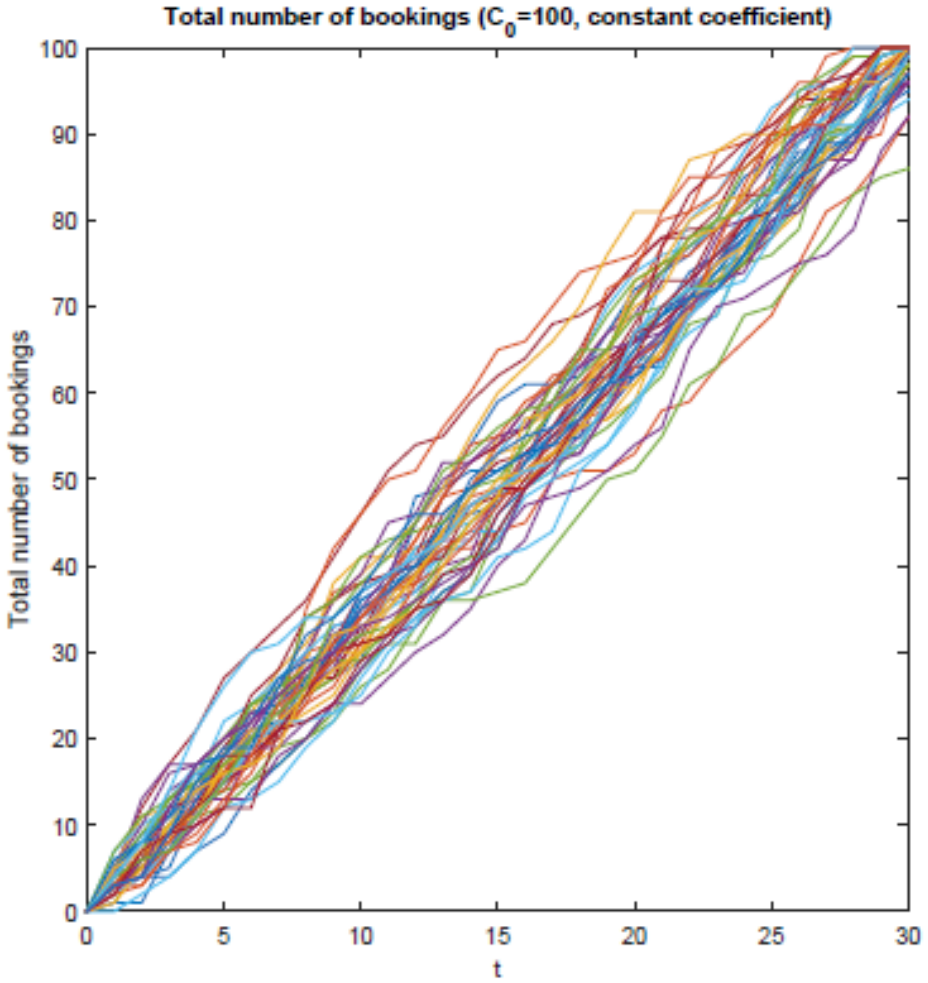




Dynamic programming

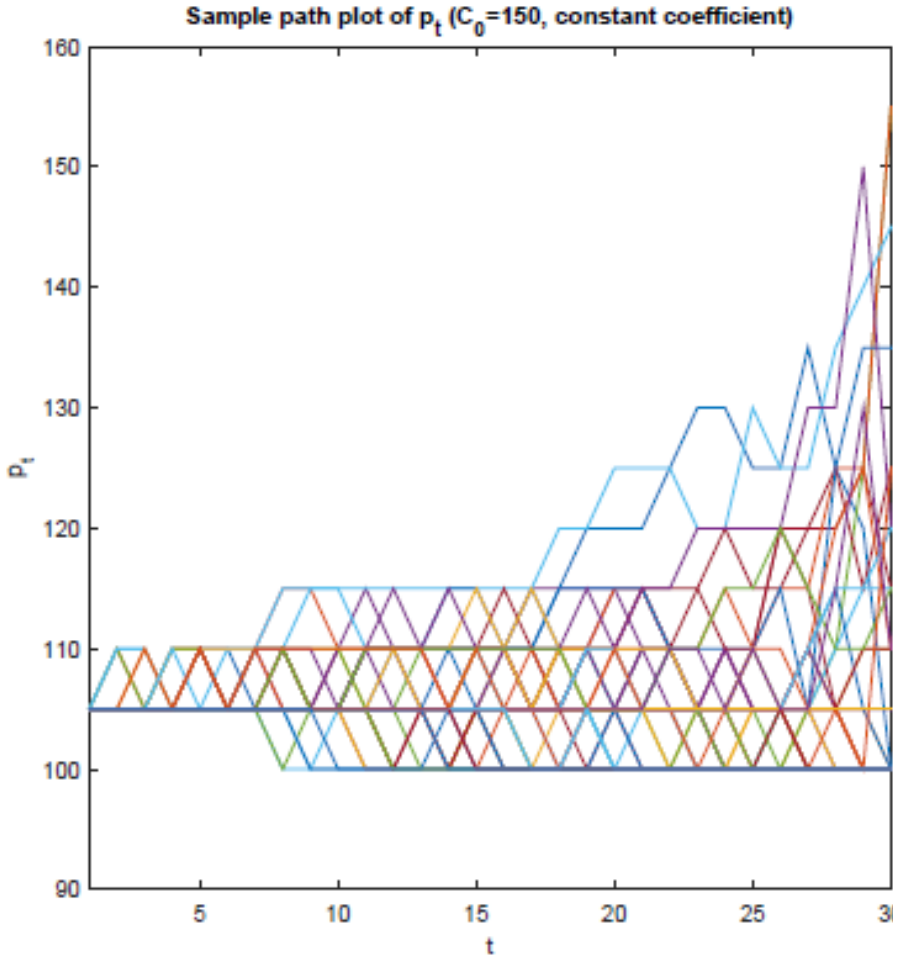
- Capacity = 100 rooms

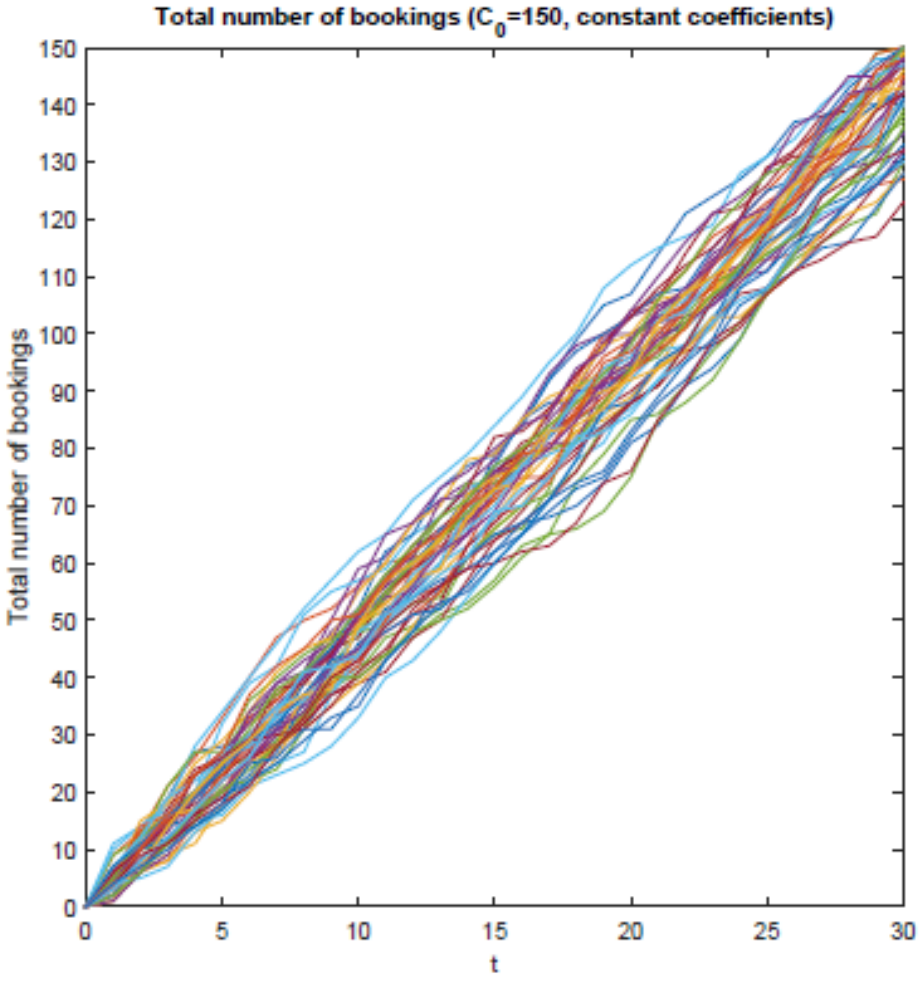




Dynamic programming

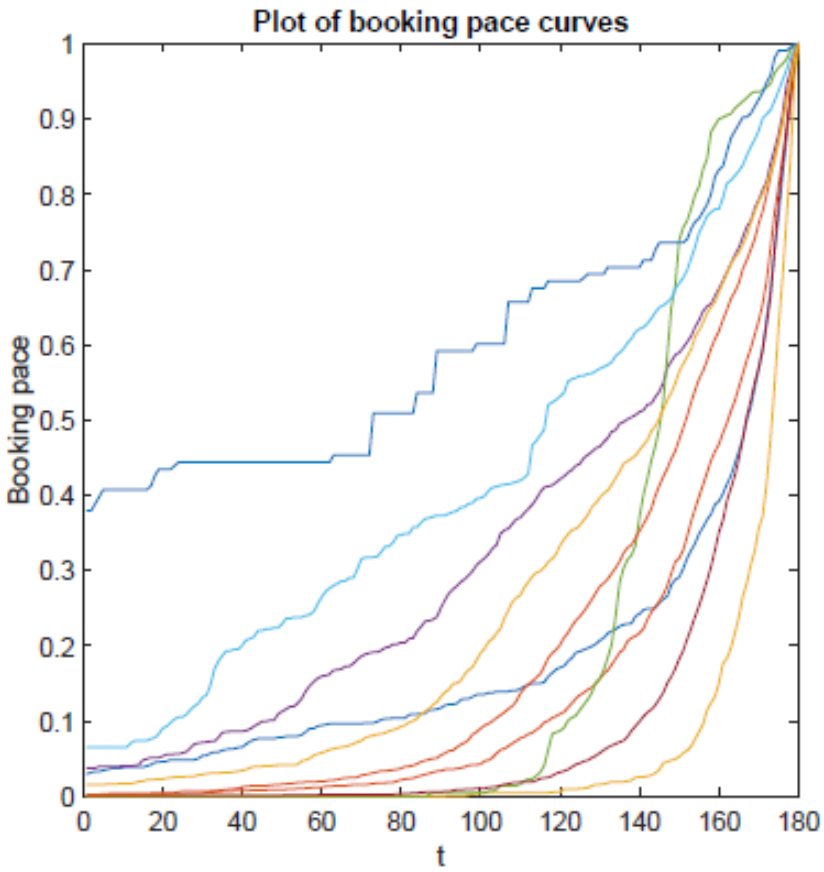
- Capacity = 150 rooms





Dynamic programming

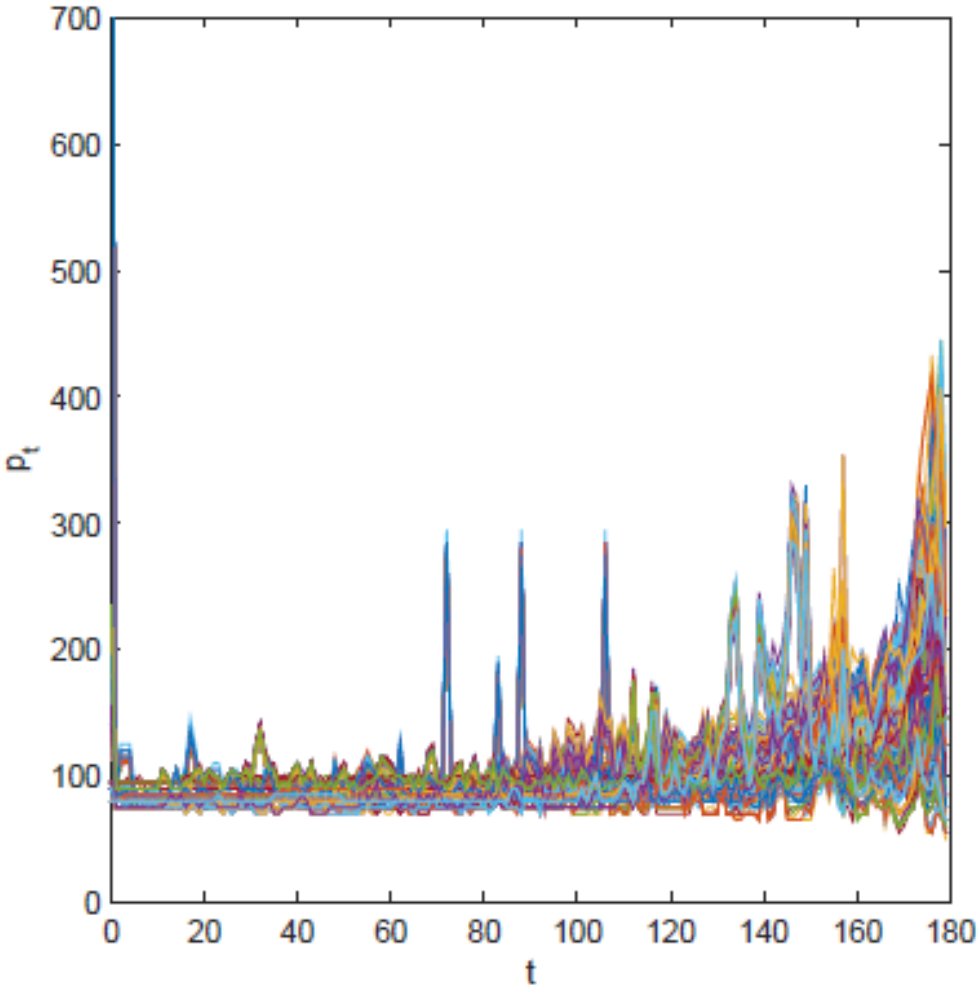
- Random booking pace curves
  - » When we take random booking pace curves into account, we have to include the booking curve probabilities $\left(q_{t,T}^{k}\right)$ in the state variable.
  - » Suddenly, we can no longer solve the dynamic program optimally.
- Solution strategies
  - » Upper bound by assuming that we know which booking pace curve is correct
  - » Solve DP for *each* BP curve and then simulate using *averaged* value function.
  - » Optimal fixed price



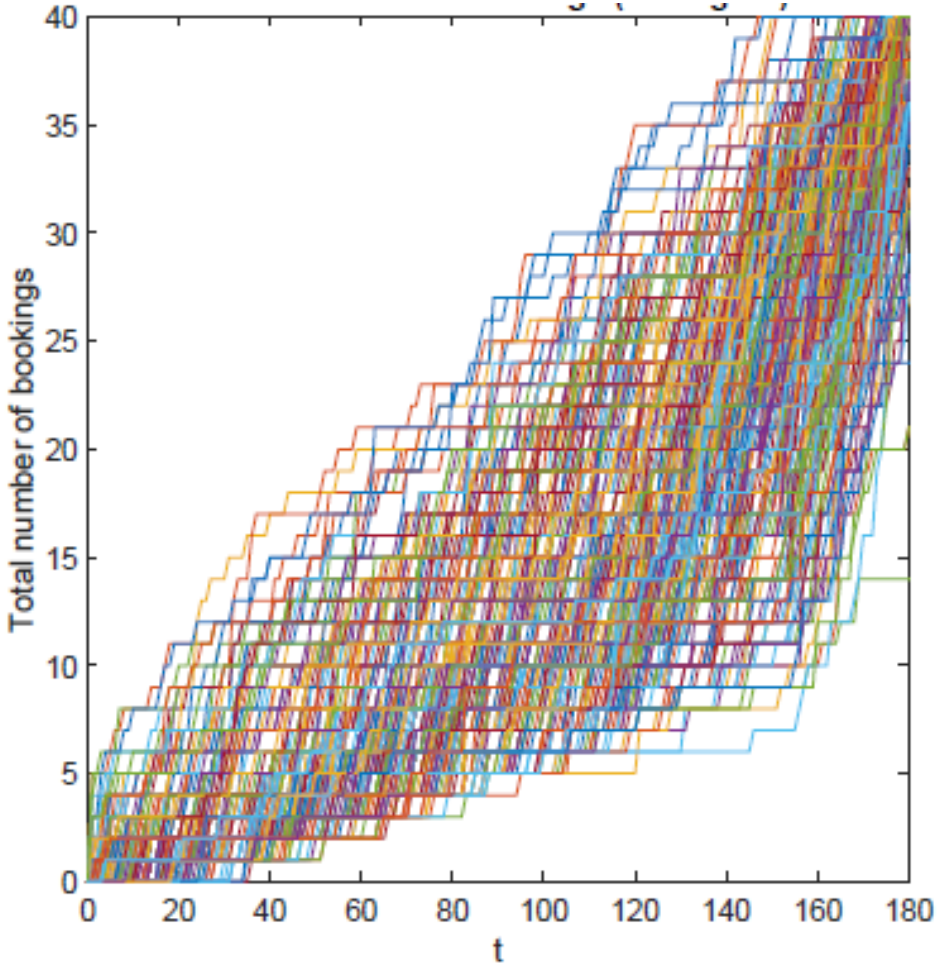
Dynamic programming

- Behavior from optimal upper bound policy (BP curves are known):

Price paths



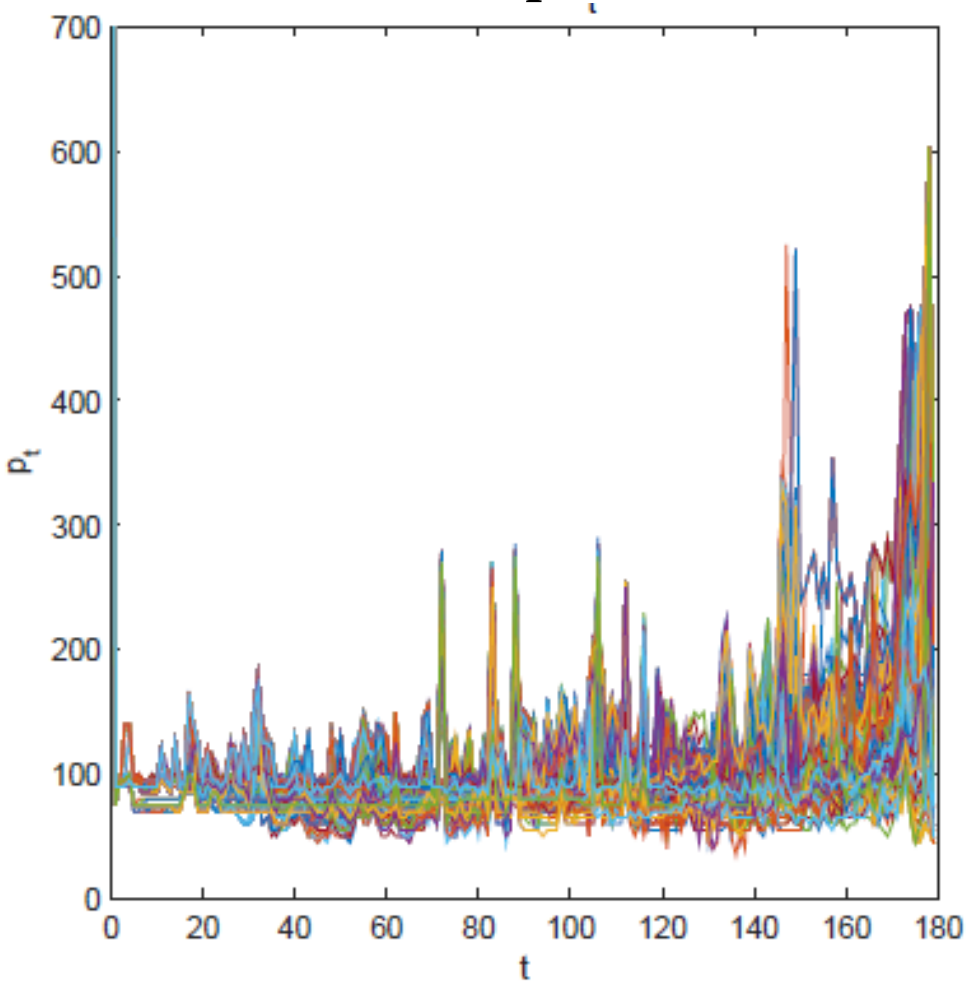
Total bookings



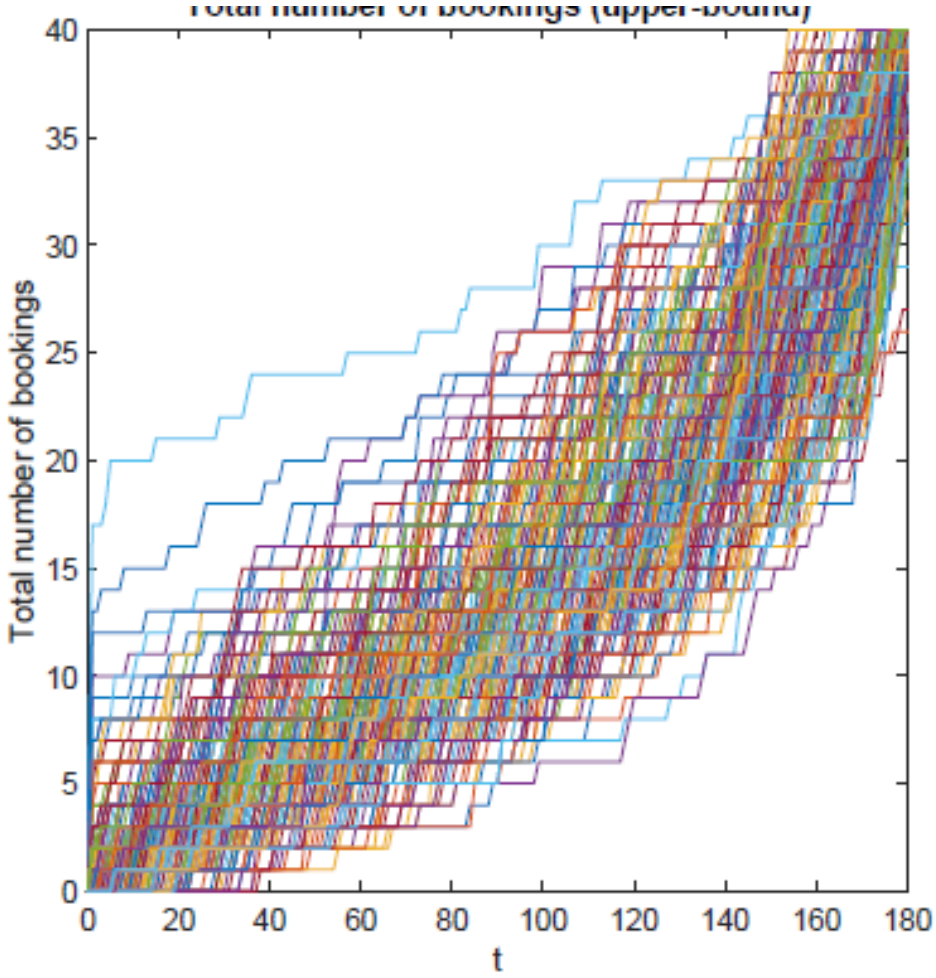
Dynamic programming

- Behavior from averaging value functions across booking pace curves:

Price paths



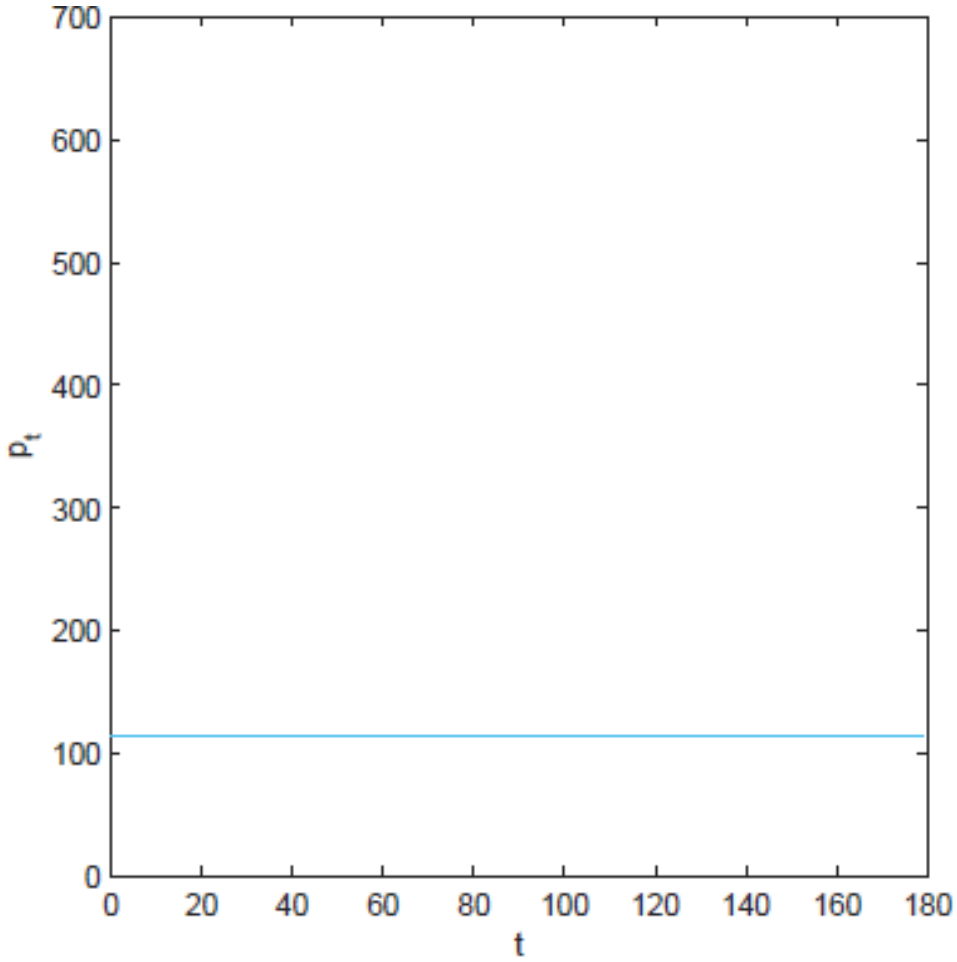
Total bookings



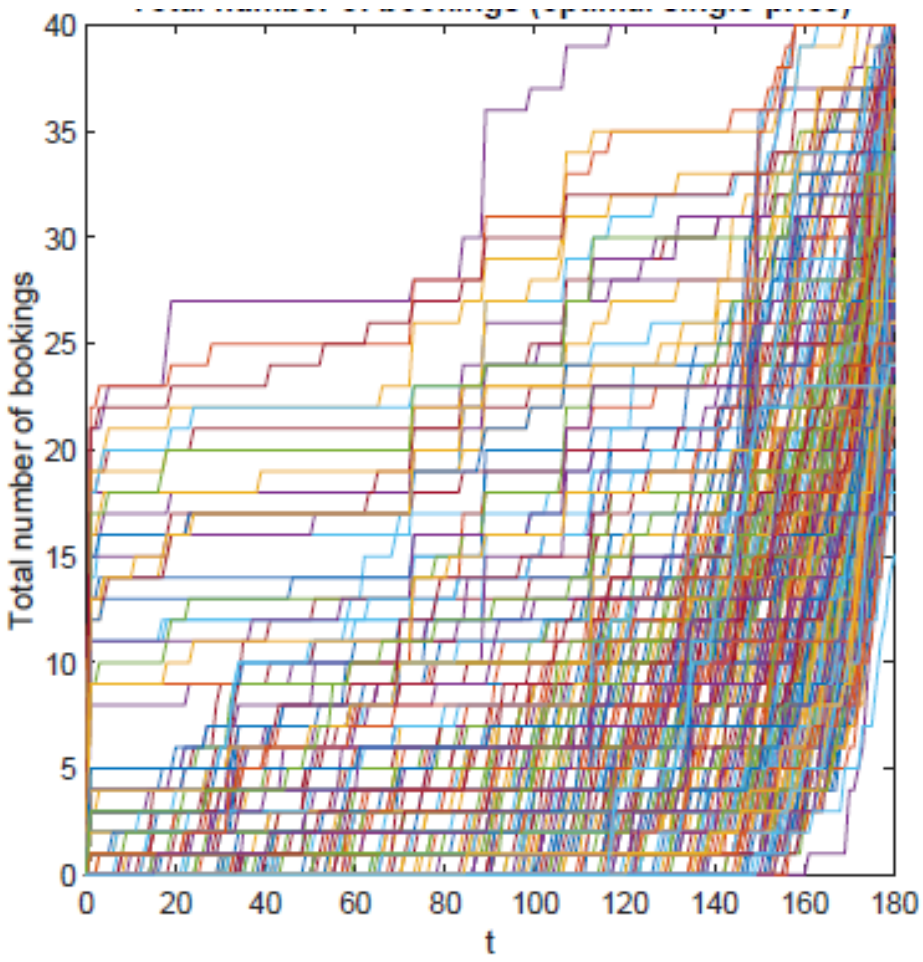
Dynamic programming

- Behavior when we assume we have to keep the price fixed:

Price paths



Total bookings



Dynamic programming

- Performance

Table: Mean total revenue under different policies (300 sample paths)

Policy	Mean total revenue	Relative to the UB
Upper bound	5419.0	100%
Averaged-V	4787.2	88.34%
Optimal single-price	3456.9	63.79%

» The large gap between the "average-V" policy and the upper bound (which only assumes that the booking pace curve is known) hints at the potential for improvements.

Week 11 - Monday

Student decision problems

Student decision problem

Booking airline reservations

When to book a flight?

- Narrative
  - » Need to book on day t to fly on day $t > T$. Similar to asset selling problem
  - » Price fluctuates day by day.
  - » Want to minimize the price we pay (and still get a seat).

- State variables:
  - » p_t=Price of a seat at time t
  - » R_t=1 if we are still trying to book a seat, 0 otherwise.
  - » $f_{tt'}$ =forecast of prices at time t' (if available).

When to book a flight?

- Decision variable
  - » $x_t = 1$ if we accept the price p_t and book a seat, 0 otherwise.
  - » Constraint: $x_t \leq R_t = 1$ if we are still trying to book a seat.
  - » Policy: $X^{\pi}(S_t)$
- Exogenous information
  - » $\hat{p}_{t+1}$ =Change in price
  - » $\hat{R}_{t+1}$ =1 means seats are still available, 0 means flight is sold out.
  - » $W_{t+1} = (\hat{R}_{t+1}, \hat{p}_{t+1})$

When to book a flight?

- Transition function
  - » $p_{t+1} = p_t + \hat{p}_{t+1}$
  - » $R_{t+1} = (R_t - x_t)\hat{R}_{t+1}$
  - » For now…
- Objective function
  - » c^{pen}=penalty for not booking a flight
  - » $\min_{\pi} \mathbb{E} \sum_{t=0}^{T} p_t X^{\pi}(S_t)$

When to book a flight?

- Uncertainty model
  - » We can use a basic model
  - » $p_{t+1} = p_t + \hat{p}_{t+1}$
  - » Where $\hat{p}_{t+1}$ is independent and identically distributed.
  - » Alternatives:
    - Mean reversion?
    - Mean reversion with jumps?

When to book a flight?

- Uncertainty model
  - » What about a rolling forecast?
    - $f_{tt'} = \mathbb{E}p_{t'}$ given that we are at time t.
    - Forecasts may be updated arbitrarily (we are given a new forecast at each time period), or we may assume that they evolve:
    - $f_{t+1,t'} = f_{tt'} + \hat{f}_{t+1,t'}$ Now we have to think about how forecasts change over time.
    - The vector $\hat{f}_{t+1}$ will be correlated. We might assume that
    - $Cov\left(\hat{f}_{t+1,t'}, \hat{f}_{t+1,t''}\right) = \sigma^2 e^{\beta|t'-t''|}$
    - Let Σ be the full covariance matrix of the changes in forecasts.
    - Assume that the expected change is 0. Let L be the lower triangular Cholesky decomposition: $L = chol(\Sigma)$. Let Z be a column vector of independent N(0,1) random variables.

$$\begin{bmatrix} \\ \hat{f}_{t+1} \\ \\ \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + L \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \end{bmatrix}$$

When to book a flight?

- Designing policies - PFA
  - » Buy if the price p_t is more than θ below
    - p_{t-1}
    - A smoothed estimate $\bar{p}_t$
  - » Might have to make θ depend on how many days remaining: θ_τ, where $\tau = T - t$.
  - » Or we can buy if price is β_τ below the previous (smoothed) price and let $\beta_\tau = \theta_1(1 - e^{-\theta_2 \tau})$. Now we just have to find (θ_1, θ_2).
- Need to tune.

When to book a flight?

- Possible CFA …
  - » Discuss logic from diabetes example.
  - » We could estimate the value of reserving on each day before the departure date. You can learn within a simulator, or using actual experience.
    - Simplify – instead of day t, use weeks.

When to book a flight?

- Tuning:
  - » Method 1 – Build a simulator, and test different values of θ.
    - Building simulators is hard.
    - Simulator never matches reality.
  - » Method 2 – Test different values of θ each time you take a flight.
    - Really slow!
    - You are going to need to design a policy to update θ.

When to book a flight?

- Designing policies – DLA - Deterministic
  - » Now imagine that we have a forecast $f_{tt'}$ of how prices will evolve.
  - » Typically this forecast will be trending up, suggesting we should buy our ticket now, but this ignores the possibility of downward spikes. This is an example of where a deterministic lookahead will not work well.
  - » If prices are trending up (but there are random spikes downward), should we reserve now?
    - We might want to wait to take advantage of a downward spike.
- Parameterized lookahead:
  - » Create estimated lookahead: $\bar{p}_t(\theta) = \sum_{t'=t}^{t+H} \theta_{t'-t} f_{tt'}$. Reserve if $p_t < \bar{p}_t(\theta)$.

When to book a flight?

- Designing policies – DLA - Stochastic
  - » We can simulate a PFA into the future for each choice (reserve now, wait).

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

  - » The lookahead policy $\tilde{\pi}$ could be a PFA.
  - » Compute the approximate expectation $\tilde{\mathbb{E}}$ could be estimated using simulation.
  - » Do this for each possible decision x_t=(reserve, wait).

Student decision problem

Financial trading

Narrative

17.1 NARRATIVE

We are trading a basket of financial instruments on a stock exchange. At each point in time, we can buy or sell some shares of any instruments in the basket, utilizing forecasts of the future stock prices with various leading horizons. The planning horizon is usually the length of the trading hours in a day, since once the market is closed we evaluate the risks based on our holding positions in the stocks.

One type of decision that we have to make is the number of shares to buy or sell in each of the stocks. These decisions have to be made under some constraints. For instance, the remaining capital has to be sufficient to finance the buying decisions. These decisions (including both buying and selling) are considered "liquidity-taking," because such decisions take the best prices available in the market and usually widen the bid-ask spread, thus reducing the liquidity of the stock. Another type of decisions is considered as "liquidity-providing," which is often faced by market makers. These decisions involve quoting prices at which a market maker is willing to buy (bid price) and sell (ask price). For the first type of decisions the decision maker has to pay a transaction cost such as the spread, while the second type of decision will not incur any transactions if no one is willing to trade at the provided prices and the market maker earns the spread in the case of a trade. In the following basic model, we focus on the first type of decisions.

State variables

State variables

Our process has two state variables: the "physical state," which captures our positions in each of the stocks, and an "informational state" which contains the prices of the stocks and the forecasts of the future prices.

Assume there are M stocks in the basket. Our "physical state" is given by our holdings in each stock, which we represent using

$$
\begin{aligned}
\mathcal{I} &= \text{The set of stocks we may hold a position in, with } i = 0 \text{ referring to cash.} \\
R_{ti} &= \text{Our position (in shares) in a particular stock } i \in \mathcal{I}, \text{ where } R_{ti} \text{ can be either positive (for a long position) or negative (for a short position). We denote by } R_{t,0} \text{ our cash position.} \\
R_t &= (R_{ti})_{i \in \mathcal{I}}
\end{aligned}
$$

Our "information state" variables include:

$$
\begin{aligned}
p_{ti} &= \text{The price of stock } i, \\
p_t &= (p_{ti})_{i \in \mathcal{I}}, \\
f_{tt'i} &= \text{The forecast, generated at time } t, \text{ of the price of stock } i \text{ at time } t' \text{ over a horizon } t' = t, \ldots, t + H, \\
f_t &= (f_{tt'i})_{i \in \mathcal{I}, t' = t, \ldots, t+H}.
\end{aligned}
$$

Our state variable is then

$$S_t = (R_t, p_t, f_t)$$

Decision variables

Decision variables

The decision variable is

x_{ti} = the number of shares that we trade for each of the stocks. We use $x_{ti} > 0$ to represent the number of shares we buy for stock i, and $x_{ti} < 0$ to represent a selling decision. We assume that short-selling is allowed so that the number of shares to sell can exceed our holding position in that stock.

The decision is constrained by the requirement that we have enough cash on hand to finance the purchasing decisions:

$$\sum_{i=1}^{M} x_{ti} p_{ti} \leq R_{t,0}.$$

We let $X^{\pi}(S_t)$ be the policy that determines x_t which satisfies this constraint.

Exogenous information

Exogenous information

The random processes in our basic model include both the change in price and the change in forecasts. Let

$$\begin{aligned}\hat{p}_{ti} &= \text{The change in the price of stock } i \text{ between } t-1 \text{ and } t,\\ \hat{p}_t &= (\hat{p}_{ti})_{i\in\mathcal{I}}.\end{aligned}$$

For the forecasts, the new information is contained in the new forecasts $f_{t+1,t',i}$. We would then write our exogenous information W_{t+1} as

$$W_{t+1} = (\hat{p}_{t+1}, f_{t+1}).$$

To simulate our process, we need to assume a probabilistic model for $\hat{p}_{t+1}$. A simple model would be to assume that $\hat{p}_{t+1}$ is normally distributed with mean 0 and variance σ^2. The evolution of the forecasts is a more complex process and we will return to this topic later.

Transition function

Transition function

The transition equation for the position in a stock R_{ti} is given by

$$R_{t+1,i} = R_{ti} + x_{ti}. \tag{17.1}$$

The transition equation for the cash position $R_{t,0}$ is given by

$$R_{t+1,0} = R_{t0} - \sum_{i=1}^{M} x_{ti} p_{ti}. \tag{17.2}$$

The transition function for the price p_t would be given by

$$p_{t+1,i} = p_{ti} + \hat{p}_{t+1,i}. \tag{17.3}$$

Also, since the new forecasts are contained in the exogenous information, we can combine equations (17.1), (17.2), and (17.3) as

$$S_{t+1} = S^M(S_t, X^\pi(S_t), W_{t+1}), \tag{17.4}$$

where $X^\pi(S_t)$ denotes a policy that maps a state to a decision.

Objective function

We evaluate our performance at the end of each day. Over the course of the day we just incur transactional costs. Then, at the end of the day, we evaluate the performance of our positions.

Let

$$c^{trans} = \text{The transaction cost per dollar.}$$

The transaction cost per period is given by

$$C_t(S_t, x_t) = -c^{trans} \sum_{i=1}^{M} |x_{ti}| p_{ti}, \text{ for } t = 0, \ldots, T-1,$$

At the end of the planning horizon, we evaluate our performance based on the net profit and the risk associated with our ending positions. We evaluate the risk using

$$\rho(R_T) = R_T' \Sigma R_T \tag{17.5}$$

where Σ denotes the covariance matrix of the returns. For example, it can be the variance in the overnight return of our portfolio. The final-period contribution function is then given by

$$C_T(S_T, x_T) = R_{T0} + \sum_{i=1}^{M} R_{Ti} p_{Ti} - \rho(R_T),$$

Finally, our objective is to find the best policy that maximizes the cumulative contribution,

$$\max_{\pi} \mathbb{E} \left\{ \sum_{t=0}^{T} C_t(S_t, X_t^{\pi}(S_t)) \middle| S_0 \right\}. \tag{17.6}$$

- Uncertainty model
  - » The process is data-driven – no uncertainty model required. No simulator has been used, and there is no lookahead component.

Designing policies

17.4 DESIGNING POLICIES

A popular class of policies used in the industry is the cost function approximations (CFA). We introduce a particular CFA policy that can be applied to solve the trading problem.

At each time t, we choose a decision according to the following policy,

$$X_t^\pi(S_t|\theta) = \arg\max_{x_t} \left(\sum_{i=1}^{M} \left((R_{ti} + x_{ti})(\tilde{f}_{ti}(\theta) - p_{ti}) - c^{trans}|x_{ti}|p_{ti} \right) - \rho(R_t + x_t) \right), \tag{17.7}$$

where $\tilde{f}_{ti}(\theta) = \sum_{s=1}^{H} \theta_s f_{t,t+s,i}$ represents an overall prediction of the future price using all available forecasts with different horizons and a tunable parameter vector $\theta = (\theta_1, \ldots, \theta_H)$. This policy maximizes a utility function that balances the trade-off between return and risk. It can be seen that for the risk function (17.5), the policy can be computed efficiently by solving a convex optimization problem.

Week 11 – Wednesday

Clinical trials

Narrative

1. Introduction

The development of new drugs for the treatment of disease has been central to the improvement of health care over the past 40 years. Annually, worldwide pharmaceutical firms spend more than $140 billion in research and development (R&D) of new drugs according to the International Federation of Pharmaceutical Manufacturers and Associations (IFPMA 2015). Pharmaceutical Research and Manufacturers of America (PhRMA 2015) reports that the average cost of development of a new drug including the cost of failures is $2.6 billion from 2000s to early 2010s and can take 10 to 15 years. As part of this R&D process, and in order to gain the approval of marketing a new drug from the Food and Drug Administration (FDA), firms must test the new drugs on human subjects to evaluate the efficacy, safety, and tolerability of the drug.

Clinical testing consists of three phases. Phase I tests the drug on healthy volunteers, Phase II tests the drug on a larger sample of several hundred patients to evaluate efficacy and identify optimal dosing, Phase III uses a much larger group of patients to confirm the efficacy and safety observed in Phase II. Clinical testing typically spans six to seven years and accounts for 50 percent of total development cost (IFPMA 2015). During this time, the drug company has to work with high patient dropout rates, which can exceed 30 percent (NRC 2010) and the challenge of evaluating the performance of treatment and control groups. In addition to the pure cost of the testing, time required is subtracting from the 20 year period during which a company enjoys patent protection.

policy of evaluating test data at intermediate points of the trial, at which time the sponsor firm has to choose one of the following three actions at each interim analysis: stopping with the conclusion that the investigated drug is effective, stopping to abandon the trial, or continuing to the next interim analysis (Kelly et al. 2005). Further, if warranted by the interim analysis, the firm may start a Phase III trial or file an New Drug Application (NDA) or Biologics License Applications (BLA) right after the interim. Interim analyses have become popular in clinical trials, which is a mechanism to reduce cost as well as allowing firms to make decisions more promptly, and hence we incorporate these in our model.

NIH U.S. National Library of Medicine

ClinicalTrials.gov

Find Studies | About Studies | Submit Studies | Resources | About Site

ClinicalTrials.gov is a database of privately and publicly funded clinical studies conducted around the world.

Explore 291,505 research studies in all 50 states and in 207 countries.

ClinicalTrials.gov is a resource provided by the U.S. National Library of Medicine.

IMPORTANT: Listing a study does not mean it has been evaluated by the U.S. Federal Government. Read our disclaimer for details.

Before participating in a study, talk to your health care provider and learn about the risks and potential benefits.

Find a study (all fields optional)

Status

- ○ Recruiting and not yet recruiting studies
- ● All studies

Condition or disease (For example: breast cancer)

Other terms (For example: NCT number, drug name, investigator name)

Country

Search | Advanced Search

2017 Studies found for: **alzheimers**

List | By Topic | On Map | Search Details

Hide Filters

Download | Subscribe to RS

Showing: 1-10 of **2,017** studies 10 studies per page

Show/Hide Columns

Filters

Apply | Clear

Status

Recruitment :
- Not yet recruiting
- Recruiting
- Enrolling by invitation
- Active, not recruiting
- Suspended
- Terminated
- Completed
- Withdrawn
- Unknown status†

Expanded Access :

Eligibility Criteria

Age :
years OR

Age Group:
- Child (birth–17)
- Adult (18–64)
- Older Adult (65+)

Sex :
- All

Row	Saved	Status	Study Title	Conditions	Interventions	Locations
1		Recruiting	Early Onset **Alzheimer's** Disease Genomic Study	• **Alzheimer** Disease	• Genetic: Genetic Testing	• Baylor Scott & White AT&T Memory Center Dallas, Texas, United States
2		Recruiting	The Chinese Familial **Alzheimer's** Network	• **Alzheimer** Disease • Familial **Alzheimer** Disease (FAD)		• The First Affiliated Hospital of Anhui Medical University Hefei, Anhui, China • Beijing Geriatric Hospital Changping, Beijing, China • Beijing Chao Yang Hospital Chaoyang, Beijing, China • (and 62 more...)
3		Not yet recruiting	Memantine Treatment in **Alzheimer's** Disease Patients	• **Alzheimer** Disease	• Drug: Memantine Hydrochloride	
4		Completed	Feasibility and Efficacy of the Ketogenic Diet in **Alzheimer's** Disease	• **Alzheimer** Disease	• Behavioral: Ketogenic Diet	
5		Recruiting	Study to Evaluate the Effect of CT1812 Treatment on Amyloid Beta Oligomer Displacement Into CSF in Subjects With Mild to Moderate **Alzheimer's** Disease	• **Alzheimer** Disease	• Drug: CT1812 • Drug: Placebo	• University of Pennsylvania Philadelphia, Pennsylvania, United States
6		Completed	Therapeutic Role of Transcranial DCS in **Alzheimer**	• **Alzheimer** Disease	• Device: tDCS • Device: Sham tDCS	
7		Completed	ATP in **Alzheimer** Disease	• **Alzheimer's** Disease	• Drug: ADENOSINE TRIPHOSPHATE • Drug: PLACEBO	• Fundació ACE Barcelona, Spain • Hospital Sanitas CIMA Barcelona, Spain
8		Not yet recruiting	ADvance II Study: DBS-f in Patients With Mild **Alzheimer's** Disease	• **Alzheimer** Disease	• Device: DBS-f On • Device: DBS Off	

16.1 NARRATIVE

Pharmaceutical companies run thousands of clinical trials each year testing new medications. Drug testing occurs in three phases:

Phase I These are tests run on 10 to 30 volunteers over a few months to determine the dose, identify side effects and perform an initial assessment of drug response and side effects.

Phase II These are larger trials that might involve over 100 patients spanning two years. The trial will typically involve comparing a group of patients with the standard treatment against one or two groups with new treatments.

Phase III These are trials involving hundreds of patients spanning multiple years to assess effectiveness and safety.

Both Phase II and Phase III trials require identifying patients with the proper characteristics to enter the trial, at which point they are randomly assigned to one of the groups for comparison.

In our exercise, we assume that we enroll a set of hospitals and clinics each week to achieve a *potential* population, from which patients will be identified as candidates for the program based on paper records. There is an up-front administrative cost to enroll a hospital or clinic in the program. The administrative cost reflects the pool of patients that the facility might have for the study. For example, it might cost $250,000 to sign up a group of hospitals and clinics with a total potential population of 500 patients. In our model, we will simply set a $500 per patient signup cost, keeping in mind that this is for a total potential population, from which we draw actual signups.

Once we have signed up a facility, we then advertise the clinical trial from which patients (or their physicians) step forward. At that point, a patient is then subjected to a more detailed evaluation, which determines who is accepted into the program. Ineligible patients are then dropped.

For the purpose of this exercise, we are going to assume that each patient is given a medication at the beginning of the week. By the end of the week, we know if the patient is responding or not. Patients that respond are designated as successes, the rest as failures. Each week, then, requires an entirely new set of patients, but these are drawn from the base population that we have signed up. We can only increase this population by entering more capacity into the program, and paying the up-front administrative cost.

- In this lecture, we are going to explore three versions of "reality":
  - » The real world – this is what we experience when we implement our policies in practice.
  - » The base model – This is going to be our representation of the real world. We would like to make this model as realistic as possible, but simplifications are inevitable.
  - » The lookahead model – We will be using a lookahead policy, where we introduce additional approximations to make it easier to solve.

- Notes:
  - » The lookahead model is by far the most subtle in terms of modeling.
  - » When the lookahead model is deterministic (but being used in a stochastic setting), then it is clear that this is an approximation.
  - » But when the lookahead model is stochastic, it can be hard to know if you are solving a lookahead approximation, or the base model.
  - » There are many instances where people will solve a problem using a stochastic lookahead model, without recognizing that it is just a lookahead model.

Basic model

State variables

We have the following state variables:

R_t = The potential population of patients that are in the hospitals and clinics that have been signed up,

α_t = The number of successes for the treatment by week t over the course of the clinical trial,

β_t = The number of failures for the treatment by week t.

λ_t = Estimated fraction of potential patients who elect to join the trial

Using this information, we can estimate the probability that our treatment is successful using

ρ_t = The probability that the treatment is successful given what we know by the end of week t,

$$= \frac{\alpha_t}{\alpha_t + \beta_t}.$$

This means that our state variable would be

$$S^n = (R_t, (\alpha_t, \beta_t)).$$

For the initial state, it is reasonable to assume that $R_0 = 0$, but we may wish to start with an initial estimate of the probability of success (based on prior clinical trials) which we can represent by assuming nonzero values for α_0 and β_0.

Decision variables

We model the number of potential patients that are signed up using

$$x_t^{enroll} = \text{The number of new potential patients that are signed up at time } t \text{ (we can think of this as at the end of week } t\text{, to be implemented for week } t+1.$$

We also have the decision of when to stop the program represented by

$$x_t^{program} = \begin{cases} 1 & \text{Continue the trial} \\ 0 & \text{Stop the trial} \end{cases}$$

If $x_t^{program} = 0$, then we are going to set $R_{t+1} = 0$, which shuts down the program. We assume that once we have stopped the trial, we cannot restart it, which means we will require that

$$x_t^{program} = 0 \text{ If } R_t = 0.$$

If we stop the trial, we have to declare whether the drug is a success or a failure,

$$x_t^{drug} = \begin{cases} 1 & \text{If the drug is declared a success} \\ 0 & \text{If the drug is declared a failure} \end{cases}$$

We will create policies $X^{\pi^{enroll}}(S_t)$, $X^{\pi^{program}}(S_t)$ and $X^{\pi^{drug}}(S_t)$ which determine x_t^{enroll}, $x_t^{program}$ and x_t^{drug}. We can then write

$$X^\pi(S_t) = (X^{\pi^{enroll}}(S_t), X^{\pi^{program}}(S_t), X^{\pi^{drug}}(S_t)).$$

Exogenous information

We first identify new patients and patient withdrawals to and from the program using

$$\hat{R}_{t+1}(x_t^{enroll}) = \text{The number of new patients joining the program during week } t+1, \text{ which depends on the number of potential patients } x_t^{enroll}.$$

We next track our successes with

$$\begin{aligned} \hat{X}_t &= \text{The number of successes during week } t, \\ \hat{Y}_t &= \text{The number of failures during week } t. \end{aligned}$$

The number of failures during week t can be calculated as

$$\hat{Y}_t = \hat{R}_t - \hat{X}_t.$$

These variables depend on the number of patients R_t in the system at the end of week t. As always, we defer to the section on uncertainty modeling the development of the underlying probability models for these random variables.

Our exogenous information process is then

$$W_t = (\hat{R}_t, \hat{X}_t),$$

where we exclude $\hat{Y}_t$ because it can be computed from the other variables.

Transition function

The transition equations for each state variable are given by

$$R_{t+1} = x_t^{program}(R_t + x_t^{enroll}), \quad (16.1)$$

$$\alpha_{t+1} = \alpha_t + \hat{X}_{t+1}, \quad (16.2)$$

$$\beta_{t+1} = \beta_t + (\hat{R}_{t+1} - \hat{X}_{t+1}). \quad (16.3)$$

We note that at this point, we do not need a formal probability model for $\hat{X}_{t+1}$ which this is a random variable that we assume that we are going to simply observe.

Objective function

We have to consider the following costs:

c^{enroll} = The cost of maintaining a patient in the program per time period,

$c^{program}$ = The ongoing administrative overhead costs of keeping the program, going (this stops when we stop testing)

$p^{success}$ = The (large) revenue gained if we stop and declare success, which typically means selling the patient to a manufacturer.

The profit (contribution) in a time period would then be given by

$$C(S_t, x_t) = (1 - x_t^{program}) x_t^{drug} p^{success} - x_t^{program}(c^{program} + c^{enroll} x_t^{enroll}). \tag{16.4}$$

Our objective function, then, would be our canonical objective which we state as

$$\max_{\pi} \mathbb{E}\left\{\sum_{t=0}^{T} C(S_t, X^{\pi}(S_t))|S_0\right\}, \tag{16.5}$$

where we recognize that our policy is a composition of the patient enrollment policy $X^{\pi^{enroll}}(S_t)$, the program continuation policy $X^{\pi^{program}}(S_t)$, and the drug success/failure policy $X^{\pi^{drug}}(S_t)$.

Uncertainty modeling

16.3 MODELING UNCERTAINTY

There are two potential reasons to develop a formal uncertainty model. The first is for the base model, which we can use both to design policies as well as to run studies. The second might be if we want to use a stochastic lookahead model, where we have to remember that if we are creating a stochastic lookahead model, we are allowed to introduce simplifications.

We are going to begin by working to develop a probabilistic base model, which means we are going to make a best effort to model the real problem, recognizing that all mathematical models are approximations of the real world.

We need to model three random variables:

- The number of customers $\hat{R}_{t+1}(x_t^{enroll})$ that sign up for the trial.
- The (unobservable) success rate ρ^{true}.
- The number of successes $\hat{X}_{t+1}$, which we do observe.

We address each of these below.

The patient enrollment process $\hat{R}_t$

We are going to use the simple model that we make choices (e.g. by signing up hospitals and clinics) that allow us to expect to sign up x_t^{enroll} patients for week $t+1$. The reality will be different. We propose to model the actual number of arrivals by assuming that they are Poisson with a mean of $\lambda^{response}(R_t + x_t^{enroll})$ where $0 < \lambda^{response} < 1$ is the fraction of potential patients who elect to join the trial (which is unknown). This means that we can write

$$Prob[\hat{R}_{t+1}(R_t) = r] \quad = \quad \frac{(\lambda^{response}(R_t + x_t^{enroll}))^r e^{-\lambda^{response}(R_t + x_t^{enroll})}}{r!}. \tag{16.6}$$

In practice we will not know $\lambda^{response}$, but we can generate estimates $\bar{\lambda}_t^{response}$ by using the ratio of $\hat{R}_{t+1}$ (which we observe) and x_t^{enroll} (which we choose). We can then use a truncated Poisson distribution for $\hat{R}_{t+1}$, where we have to recognize that the number of patients that join the program is limited by the number of potential patients given by $R_{t+1} = R_t + x_t^{enroll}$. Let

$$\bar{R}_t = \bar{\lambda}_t^{response}(R_t + x_t^{enroll})$$

be the expected number of patients that will volunteer for the program (given R_t) and

$$P_{\hat{R}_{t+1}}(r|x_t^{enroll}, \bar{R}_t) \quad = \quad Prob[\hat{R}_{t+1}(x_t^{enroll}) = r|\bar{R}_t].$$

We write $P_t^{\hat{R}}(r|x_t^{enroll}, \bar{R}_t)$ as a function of x_t^{enroll} and $\bar{R}_t$ to reflect its dependence on the decision and on the number $R_{t+1} = R_t + x_t^{enroll}$.

The truncated Poisson distribution is then given by

$$P_{\hat{R}_{t+1}}(r|x_t^{enroll},\bar{R}_t) = \begin{cases} \frac{(\bar{R}_t)^r e^{-\bar{R}_t}}{r!} & r = 0,1,\ldots,x_t^{enroll}-1 \\ 1-\sum_{r=0}^{x_t^{enroll}-1} P_{\hat{R}_{t+1}}(r|x_t^{enroll},\bar{R}_t) & \text{If } r = x_t^{enroll} \end{cases} \tag{16.7}$$

For a population process such as this, a Poisson process is a good starting point. It enjoys the property that the mean equals the variance which equals x_t^{enroll}. It is not uncommon to find that real problems of this type exhibit a higher variance than the Poisson would predict. One way to increase the variance is to add a random δx to the mean x_t^{enroll} that might be uniformly distributed in some range that needs to be tuned.

The success probability ρ^{true}

The successes are driven by the underlying, but unobservable, probability that the treatment will create a success in a patient during a week. We use the Bayesian style of assigning a probability distribution to ρ^{true}. There are three ways to represent the distribution of our belief about ρ^{true}:

- A uniform prior, where we would assume that ρ^{true} is uniformly distributed between 0 and 1.
- A beta distribution with parameters (α_0, β_0).
- A sampled distribution, where we assume that ρ^{true} takes on one of the set of values $(\rho_1, \ldots, \rho_K)$, where we let our initial distribution be

$$p^{\rho}_{0k} = Prob[\rho^{true} = \rho_k].$$

  We might let $p_{0k} = 1/K$ (this would be comparable to using the uniform prior). Alternatively, we could estimate these from the beta distribution.

For now, we are going to use our sampled distribution since it is easiest to work with.

The success process $\hat{X}_t$

The random number of successes $\hat{X}_{t+1}$, given what we know at time t, depends first on the random variable $\hat{R}_{t+1}$ giving the number of patients who entered the trial, and the unknown probability ρ^{true} of success in the trial. The way to create the distribution of $\hat{X}_{t+1}$ is to use the power of conditioning. We assume that $\hat{R}_{t+1} = r$ and that $\rho^{true} = \rho_k$.

Given that r patients enter the trial and assuming that the probability of success is ρ_k, the number of successes $\hat{X}_{t+1}$ is the sum of r Bernoulli (that is, 0/1) random variables. The sum of r Bernoulli random variables is given by a binomial distribution, which means

$$Prob[\hat{X}_{t+1} = s|\hat{R}_{t+1} = r, \rho^{true} = \rho_k] = \binom{r}{s}\rho_k^s(1-\rho_k)^{r-s}.$$

We can find the unconditional distribution of $\hat{X}_{t+1}$ by just summing over r and k and multiplying by the appropriate probabilities, giving us

$$Prob[\hat{X}_{t+1} = s|\bar{R}_t] = \sum_{k=1}^{K}\left(\sum_{r=0}^{R_t} Prob[\hat{X}_{t+1} = s|\hat{R}_{t+1} = r, \rho^{true} = \rho_k]P_{\hat{R}_{t+1}}(r|x_t^{enroll}, \bar{R}_t)\right)p_{tk}^{\rho}. \tag{16.8}$$

Using explicit probability distributions such as the one for $\hat{X}_{t+1}$ in equation (16.8) is nice when we can find (and compute) them, but there are many complex problems where this is not possible. For example, even equation (16.8) required that we use the trick of using a sampled representation of the continuous random variable ρ^{true}. Without this, we would have had to introduce an integral over the density for ρ^{true}.

Another approach, which is much easier and extends to even more complicated situations, uses Monte Carlo sampling to generate $\hat{R}_{t+1}$ and $\hat{X}_{t+1}$. This process is outlined in figure 16.1, which produces a sample $\hat{X}_{t+1}^1, \ldots, \hat{X}_{t+1}^N$ (and corresponding $\hat{R}_{t+1}^1, \ldots, \hat{R}_{t+1}^N$). We can now approximate the random variable $\hat{X}_{t+1}$ with the set of outcomes $\hat{X}_{t+1}^1, \ldots, \hat{X}_{t+1}^N$, each of which may occur with equal probability.

Step 1. Loop over iterations $n = 1, \ldots, N$:

Step 2a. Generate a Monte Carlo sample $r^n \sim \hat{R}_{t+1}(x^{enroll})$ from the Poisson distribution given by equation (16.7).

Step 2b. Generate a Monte Carlo sample of the true success probability $\rho^n \sim \rho^{true}$.

Step 2c. Given r^n and ρ^n, loop over our r^n patients and generate a 0/1 random variable which is 1 (that is, the drug was a success) with probability ρ^n.

Step 2d. Sum the successes and let this be a sample realization of $\hat{X}^n_{t+1}$.

Step 4. Output the sample $\hat{X}^1_{t+1}, \ldots, \hat{X}^N_{t+1}$.

Figure 16.1 A Monte Carlo-based model of the clinical trial process.

Week 12 – Monday

Finish clinical trials

More decision problems

Designing policies

Clinical trials

16.4 DESIGNING POLICIES

We are going to use this problem to really understand our full stochastic lookahead policy which we first introduced in chapter 8, given by

$$X^*(S_t) = \arg\max_{x_t \in \mathcal{X}} \Bigg(C(S_t, x_t) + \mathbb{E}_{W_{t+1}} \Bigg\{ \max_{\pi} \mathbb{E}_{W_{t+2},\ldots,W_T} \Bigg\{ \sum_{t'=t+1}^{T} C(S_{t'}, X^{\pi}_{t'}(S_{t'})) | S_{t+1} \Bigg\} | S_t, x_t \Bigg\} \Bigg). \tag{16.9}$$

In particular, we are going to focus on what is meant by that maximization over policies π embedded within the policy.

For our clinicial trials application, we have to design policies for the three different decisions: the number of patients to enroll, whether or not to continue the trial, and whether or not the drug is declared a success when the trial is stopped. We are going to begin by designing simple policy function approximations for the decisions of whether to stop or continue, and if we stop, whether we declare the drug a success or a failure. We then address the more difficult decision of how many patients to enroll in the trial.

16.4.1 Stopping the trial and declaring success or failure

We begin by using our belief about ρ^{true} given by the beta distribution with parameters (α_t, β_t), which gives us an estimate of

$$\bar{\rho}_t = \frac{\alpha_t}{\alpha_t + \beta_t}$$

Now introduce the parameters $\theta^{stop-low}$ and $\theta^{stop-high}$, where we are going to stop the trial and declare success if $\bar{\rho}_t > \theta^{stop-high}$, while we will stop the trial and declare failure if $\bar{\rho}_t < \theta^{stop-low}$. Let $\theta^{stop} = (\theta^{stop-low}, \theta^{stop-high})$. We use these rules to define the policy for stopping the trial as

$$X_t^{trial}(S_t|\theta^{stop}) = \begin{cases} 1 & \text{If } \theta^{stop-low} \leq \bar{\rho}_t \leq \theta^{stop-high} \\ 0 & \text{Otherwise.} \end{cases}$$

If we stop the trial, then the policy for declaring success (1) or failure (0) is given by

$$X_t^{drug}(S_t|\theta^{stop}) = \begin{cases} 1 & \text{If } \bar{\rho}_t > \theta^{stop-high} \\ 0 & \text{If } \bar{\rho}_t < \theta^{stop-low} \end{cases}$$

16.4.2 The patient enrollment policy

To create a full lookahead model as we described in section 16.3, we would create variables such as $\bar{\lambda}_{tt'}$ for the lookahead version of $\bar{\lambda}_t^{response}$, $\bar{\rho}_{tt'}$ for $\bar{\rho}_t$, and $(\bar{\alpha}_{tt'}, \bar{\beta}_{tt'})$ for (α_t, β_t). Otherwise, all the logic would be the same as the original uncertainty model.

While we can use the full uncertainty model, we can choose to simplify the model in different ways. These choices include:

- The enrollment rate λ_t - We have two options:
  - We can continue to estimate λ_t, where we would introduce the notation $\bar{\lambda}_{tt'}$ as the estimate at time t' in the lookahead model of the enrollment rate λ.
  - We could fix $\bar{\lambda}_{tt'} = \bar{\lambda}_t^{response}$, which is our estimate at time t in the base model.
- The drug success rate ρ^{true} - We again have two options:
  - We can continue to estimate the success rate. For this, we would define the variables $(\bar{\alpha}_{tt'}, \bar{\beta}_{tt'})$ for accumulating successes and failures in the lookahead model.
  - This means we would fix $(\bar{\alpha}_{tt'}, \bar{\beta}_{tt'}) = (\alpha_t, \beta_t)$ within the lookahead model.

Using our choices for modeling uncertainty, we can suggest three different strategies for designing a lookahead model:

Model A Deterministic lookahead model - Here, we are going to assume that the enrollment rate $\bar{\lambda}_{tt'} = \bar{\lambda}_t^{response}$, which means that the enrollment rate is fixed at the estimate at time t when we create the lookahead model. We then assume that the true drug success probability is fixed at

$$\bar{\rho}_{tt'} = \bar{\rho}_t = \frac{\alpha_t}{\alpha_t + \beta_t},$$

which is our estimate at time t in the base model.

Model B We fix our estimate of the enrollment rate at $\bar{\lambda}_{tt'} = \bar{\lambda}_t^{response}$, but assume that we continue learning about the effectiveness of the drug.

Model C We model the process of learning the enrollment rate $\bar{\lambda}_{tt'}$ and the drug effectiveness $\bar{\rho}_{tt'}$.

Note that we did not include the potential fourth model where we fix the drug effectiveness but continue learning the patient enrollment rate (we will see in a minute how silly this model would be).

We are going to use these three models to illustrate the process of designing a lookahead model.

Model A

Model A is a deterministic problem, since we are fixing both the estimated enrollment rate $\bar{\lambda}_{tt'} = \bar{\lambda}_t^{response}$, and $\bar{\rho}_{tt'} = \bar{\rho}_t$. The good news is that this is basically a deterministic shortest path problem, where the number of patients we have signed up (in the lookahead model), given by $\bar{R}_{tt'}$, is like a node in a network, and the decision $\bar{x}_{tt'}^{enroll}$ is a link that takes us to node $\bar{R}_{t,t'+1} = \bar{R}_{tt'} + \bar{x}_{tt'}^{enroll}$.

To see this, recall equation (5.1) for our deterministic shortest path problem, which we repeat here

$$v_i = \min_{j \in \mathcal{N}_i^+} (c_{ij} + v_j).$$

Now we just replace v_i for the value at node i, with $\widetilde{V}_{tt'}(\bar{R}_{tt'})$ which is the value of having $\bar{R}_{tt'}$ patients signed up (remember we are in our lookahead model). The decision to go to node j is replaced with the decision to sign up $\bar{x}_{tt'}^{enroll}$ patients. Instead of this taking us to node j, it takes us to node $\bar{R}_{tt'} + \bar{x}_{tt'}^{enroll}$. So Bellman's equation becomes

$$\widetilde{V}_{tt'}(\bar{R}_{tt'}) = \min_{\bar{x}_{tt'}^{enroll}} \left(\bar{C}(\bar{R}_{tt'}, \bar{x}_{tt'}^{enroll}) + \widetilde{V}_{t,t'+1}(\bar{R}_{tt'} + \bar{x}_{tt'}^{enroll})\right). \quad (16.9)$$

The one-period profit function $\bar{C}(\bar{R}_{tt'}, \bar{x}_{tt'}^{enroll})$ is adapted from the same function for our base model (see equation (16.4)).

There is only one problem with our deterministic lookahead model: we would never stop, because our policy for stopping requires that our estimate of $\tilde{\rho}_{tt'}$ move into the "success" or "fail" regions (it would have to start in the "continue" region, since otherwise we would have stopped the base model). However, this does not mean that we cannot use the deterministic lookahead model: we just have to fix a horizon H and stop when $t' = t + H$.

Using this strategy, we solve our deterministic shortest path problem over the horizon $t' = t, \ldots, t + H$, and then from this find $\tilde{x}^*_{tt}$. Our enrollment policy is then

$$X^{\pi^{enroll}}(S_t) \quad = \quad \tilde{x}^*_{tt}.$$

We are not claiming that this will be an effective policy. We are primarily illustrating the types of modeling approximations that can be made in a lookahead model.

Model B

Now we are going to fix our estimate of the the response rate $\bar{\lambda}_{tt'}$ at our estimate $\bar{\lambda}_t^{response}$ at time t in the base model. To simplify our model, we are going to assume that the number of enrollments $\bar{\hat{R}}_{t,t'+1}$ equals the expected number of patients that will volunteer $\bar{\bar{R}}_{tt'}$. The enrollments $\bar{\hat{R}}_{t,t'+1}$ are generated deterministically from

$$\bar{\hat{R}}_{t,t'+1} = \lfloor \bar{\lambda}_t^{response}(\bar{R}_{tt'} + \bar{x}_{tt'}^{enroll}) \rfloor,$$

Remember that lambda is the estimate from the base model, so it has a bar, not a tilde.

where $\lfloor x \rfloor$ means to round x down to the nearest integer. We then compute the distribution of $\bar{\hat{X}}_{t,t'+1}$ using $Prob[\hat{X}_{t+1} = s|\bar{R}_t]$ but where we replace $\bar{R}_t$ with $\bar{\bar{R}}_{tt'}$.

We still have to generate the successes $\bar{\hat{X}}_{t,t'+1}$ from a simulated truth $\bar{\rho}_{tt'}$, from which we will update $(\bar{\alpha}_{tt'}, \bar{\beta}_{tt'})$.

$$\begin{aligned}
\bar{\alpha}_{t,t'+1} &= \bar{\alpha}_{tt'} + \bar{\hat{X}}_{t,t'+1}, \\
\bar{\beta}_{t,t'+1} &= \bar{\beta}_{tt'} + \bar{\bar{R}}_{tt'} - \bar{\hat{X}}_{t,t'+1}.
\end{aligned}$$

We model the distribution of $\bar{\hat{X}}_{t,t'+1}$ using $Prob[\bar{\hat{X}}_{t,t'+1} = s|\bar{\bar{R}}_{tt'}]$ in equation (16.8) but conditioning on $\bar{\bar{R}}_{tt'}$ instead of $\bar{R}_t$ (remember that we can also use the sampled distribution using the method in figure 16.1 instead of the Poisson distribution).

We can solve the lookahead model by adapting Bellman's equation for Model A in equation (16.9) for $t' = t + H, \ldots, t$:

$$\tilde{V}_{tt'}(\tilde{S}_{tt'}) = \min_{\tilde{x}^{enroll}_{tt'}} \left(\tilde{C}(\tilde{S}_{tt'}, \tilde{x}^{enroll}_{tt'}) + \sum_{s=0}^{\bar{\tilde{R}}_{tt'}} Prob[\hat{\tilde{X}}_{t,t'+1} = s | \bar{\tilde{R}}_{tt'}] \tilde{V}_{t,t'+1}(\tilde{S}_{t,t'+1} | \hat{\tilde{X}}_{t,t'+1} = s) \right). \quad (16.10)$$

where $\tilde{S}_{t,t'+1} = (\tilde{R}_{t,t'+1}, \tilde{\alpha}_{t,t'+1})$ is conditioned on the number of successes $\hat{\tilde{X}}_{t,t'+1} = s$, and where $Prob[\hat{\tilde{X}}_{t,t'+1} = s | \bar{\tilde{R}}_{tt'}]$ comes from equation (16.8). We have to keep in mind that the evolution of $\tilde{R}_{tt'}$ has to reflect if we have decided to stop or continue the trial within the lookahead model.

Our physical state variable (total potential patients) $\tilde{R}_{t,t'+1}$ is given by

$$\tilde{R}_{t,t'+1} = \begin{cases} \tilde{R}_{tt'} + \tilde{x}^{enroll}_{tt'} & \text{If } \tilde{X}^{trial}(\tilde{S}_{tt'} | \theta^{stop}) = 1 \\ 0 & \text{Otherwise} \end{cases}.$$

Note that as with our base model, the number of potential patients drops to zero if we stop the trial in the lookahead model.

Our current estimate of the success of the drug (in the lookahead model) is computed using

$$\bar{\bar{\rho}}_{tt'} = \frac{\bar{\alpha}_{tt'}}{\bar{\alpha}_{tt'} + \bar{\beta}_{tt'}}.$$

In the expectation, if we condition on the number of successes being $\bar{\hat{X}}_{t,t'+1} = s$, then the updated belief state $(\bar{\alpha}_{tt'}, \bar{\beta}_{tt'})$ is

$$\bar{\alpha}_{t,t'+1} = \bar{\alpha}_{tt'} + s, \tag{16.11}$$

$$\bar{\beta}_{t,t'+1} = \bar{\beta}_{tt'} + (\bar{\bar{R}}_{tt'} - s). \tag{16.12}$$

We now have to solve the lookahead model using Bellman's equation in equation (16.10). For this problem, it makes sense to use a large-enough horizon H so that we can confidently assume that we would have stopped the trial by then (that is $\bar{X}^{trial}(\bar{S}_{tt'}|\theta^{stop}) = 0$). This means we can assume that $\widetilde{V}_{t,t+H}(\bar{S}_{t,t+H}) = 0$, and work backward from there to time t. Once we have solved the dynamic program, we can pull out our enrollment decision using

$$X_t^{enroll}(S_t) = \arg\min_{\bar{x}_{tt}^{enroll}} \left(\bar{C}(\bar{S}_{tt}, \bar{x}_{tt}^{enroll}) + \sum_{s=0}^{\bar{\hat{R}}_{t,t+1}} Prob[\bar{\hat{X}}_{t,t'+1} = s|\bar{\bar{R}}_{tt}] \widetilde{V}_{t,t+1}(\bar{S}_{t,t+1}|\bar{\hat{X}}_{t,t'+1} = s) \right).$$

- Model C
  - » Here we have modeled the dynamics (including uncertainty) the same as the base model.
  - » The only approximation is that we are using the PFAs that we introduced for the decision of when to stop, and the decision of whether to declare the drug a success. So, we may not be searching over the entire class of policies.
  - » If we could show that the PFAs for these decisions include the optimal policy, then Model C would produce an optimal policy for the base model.

Student decision problem

Rowing strategy

Sadie McGirr

Optimal Training Plan for Athletic Performance: A Sequential Decision Problem

Sadie McGirr
Advisor: Warren Powell

June 2019

- Narrative
  - » Provided by Sadie…
  - » The problem is to optimize the timing of different segments of a rowing race. The race is divided into four 500-yard segments, but the first and last segment are then divided into two, creating an initial “sprint” segment and a final closing segment (both are usually spring segments). Rowers work in terms of “splits” which is the time required to finish each segment.

State variables

State Variables

$$S_t = (L_t, H_t, A_t, (\hat{\tau}_t)_{t=1,\dots,t}, (\bar{\tau}_t)_{t=t,\dots,T})$$

Where:

$L_t = the\ amount\ of\ lactic\ acid\ buildup$

$H_t = the\ heart\ rate$

$A_t = addreline\ level\ W$

$(\hat{\tau}_t)_{t=1,\dots,t-1} = actual\ split\ times\ for\ increments\ 1, \dots, t-1$

$(\bar{\tau}_t)_{t=t,\dots,T} = forcasted\ splits\ for\ times\ t, \dots, T$ (probably do not need this is state var?)

$$Y_t = \sum_{t'=1}^{t-1} \hat{\tau}_{t'} + \sum_{t'=t}^{T} \bar{\tau}_{tt'}$$

$Y_t = projected\ total\ time$

$\hat{\tau} = actual\ split\ times$

$\bar{\tau} = projected\ split\ times$

- Decision variables

Decision Variables

$x_t = target\ split$

- Exogenous information

Exogenous Information

$W_{t+1} = (\epsilon^{\tau}_{t+1}, \epsilon^{L}_{t+1}, \epsilon^{H}_{t+1})\ or\ W_{t+1} = \epsilon^{\tau}_{t+1}?$

$\epsilon^{\tau}_{t+1} = deviation\ in\ your\ forcast\ of\ split\ and\ actual\ split$

$F^{\tau}(S_t, x_t) = \mathbb{E}[\tau_{t+1}|S_t, x_t] = forcast\ of\ your\ split\ time$

$\hat{\tau}_{t+1} = F^{\tau}(S_t, x_t) + \epsilon^{\tau}_{t+1} \rightarrow the\ actual\ split\ is\ your\ forcast + deviation$

$F^{\tau}(S_t, x_t) \rightarrow (\bar{\tau}_t)_{t=t+1,\ldots,T}$

- Think about what a function for $F^{\tau}(S_t, x_t)$. This function will describe projected split time, given how much lactic acid is built up, heart rate, and adrenaline.
- L_{t+1} and H_{t+1} are monotonic, try to find papers on relationship between lactic acid, heart rate, and exercise.
- Epsilons should have mean zero or you need to change your model
- H_{t+1} could possibly not have an equation, model-free

- Transition function

$$L_{t+1} = L_t + \ell(\hat{\tau}_t) - \eta + \epsilon^L_{t+1}$$

$$\ell(\hat{\tau}_t) = function\ of\ how\ lactic\ acid\ level\ depends\ on\ split$$

$$H_{t+1} = H_t + \cdots + \epsilon^H_{t+1}$$

$$A_{t+1} \begin{cases} 1\ with\ probability\ p\ in\ first\ or\ last\ 250 \\ 0\ otherwise \end{cases}$$

?

  - Think more about equations for H_{t+1} and A_{t+1}
  - It is possible that H_{t+1} is just observed and has no equation, model free
  - Need to approximate function $\ell(\hat{\tau}_t)$ using parametric, non-parametric, or lookup table
  - $\ell(\hat{\tau}_t)\ vs.\ Q_t\ (quickness)$
  - L_{t+1} is a hidden state variable
  - Add parameters $L_{t+1} = L_t + \ell(\hat{\tau}|\theta^L) - \eta, \theta^L and\ \eta\ are\ tunable\ parameters$

» What does the function $\ell(\hat{\tau}_t)$ look like?

» How do we even get an estimate of what L_t is?

» Could we replace lactic acid with splits, with a nonlinear model that captures "tiredness" if the split is above your steady state pace.

- Objective function

Objective Function

$$\min_{\tau} \sum_{t=1}^{T-1} \hat{\tau}_t$$

» How to write this correctly:

- $\hat{\tau}_t(x_t)$ needs to depend on the split, which is specified by the policy $x_t = X^\pi(S_t)$. We could write

$$\hat{\tau}_t(x_t) = X^\pi(S_t) + \varepsilon_{t+1}(S_t)$$

- where the noise ε_{t+1} depends on the state S_t. May not have mean 0.

» So we could write our objective as

$$\min_{\pi} E \sum_{t=0}^{T} \hat{\tau}_t(X^\pi(S_t))$$

Uncertainty model

- Uncertainty model

Chapter 3: Uncertainty Modeling

Resting Heart Rate

Perceived Exertion Rate

Recovery

Sleep

Stress

Injury

» Race uncertainties:
  - Wind, choppiness of the water
    - Introduce randomness in speeds
  - Performance of the competition – does this have a psychological effect? How to model?
    - State variable indicating: behind, tied/close, ahead

Designing policies

- Designing policies
  - » Deterministic policy
    - Choose all the splits in advance regardless of within-race performance.
    - Discuss – is this realistic? What types of adjustments are made within the race.
  - » Policy function approximation
    - Split = initial plan + adjustment
    - Adjustments:
      - If behind, adjust remaining splits to finish at target time.
      - If ahead, no adjustment (assumes we cannot keep beating the splits).

- Designing policies
  - » Direct lookahead
    - Optimize over rest of race using nonlinear lactic acid function.
    - Can describe this as a graph if you discretize the splits. Show as decision tree where each time you set a split, you then realize what actually happens over the segment.
    - This can be solved as a decision tree, which is really a dynamic program where you use value functions.

- Extensions
  - » A nonlinear model for lactic acid should discourage sprinting at the beginning.
  - » If we really believe that sprints work, what are we missing?
  - » We often have an intuitive idea of what we should be doing, without knowing a reason why.
  - » This intuitive behavior can be handled with PFAs, but it does not mean it is right!

Student decision problem

Hiring interns

- Narrative
  - » 1) Should a company have an internship program (the decision would be whether to hire interns or not)
  - » 2) Should a company hire a student with specific attributes (e.g. school, major, GPA, …).
  - » Let's pursue the second one.
- State variables
  - » List of intern applications (this is the "physical state").
    - $R_{ta} = 1$ if we have an applicant with attribute a
  - » Estimate of probability that an intern with attribute a ends up being hired by the company, and accepts the offer.

- Decision variables
  - » Whether to hire an intern.
  - » Whether to offer an intern a full time job at the end of the summer.
  - » Specify policies (to be designed)
- Exogenous information
  - » Intern applications
  - » Performance of the intern.
  - » Whether the intern accepts the offer.

- Transition function
  - » Updating of beliefs of probabilities

- Objective function
  - » Metric: acceptance of the offer by the intern

- Uncertainty model
  - » Probability model of applications from different schools
  - » Probability model of performance by school
  - » Probability model of acceptance of offer

- Designing policies
  - » If we ignore learning, we are simply going to hire an intern if the probability of extending an offer, and then the offer being accepted, is high enough.
    - May also want to include the probability that the employee is viewed as a successful hire after one year, versus hiring someone from a standard interviewing process.
  - » This will be an active learning problem. Need to capture the uncertainty in an estimate, and the value of information to guide future decisions.

- Designing policies
  - » PFA policies:
    1) Hire any intern with success probability over some amount.
    2) (1) plus randomization on probability.
  - » CFA policies:
    1) Hire interns with highest probability of eventual success.
    2) Same as (1), but add some randomization (this is Thompson sampling).
    3) Interval estimation – maximize probability plus tunable parameter times standard deviation of estimated probability.
  - » VFA policies ??
    - Unlikely.
  - » DLA policies:
    - One-period lookahead – value of information
    - Multiperiod extension

- Extensions
  - » ??

Student decision problem

Twitter trading – Senior thesis of Raj Patel

- Narrative
  - » Hunter Johnson asked:
    - How to invest effectively to profit off of Trump's twitter posts
  - » This is a decision problem on when to buy/sell stocks using the information in twitter feeds.
  - » This is in the same problem class as the original asset selling problem (chapter 2) or when to purchase a seat on a flight. The challenge is designing the stochastic model of prices using twitter data.

PRINCETON UNIVERSITY

SENIOR THESIS

Twitter Trading: Modeling Twitter Processes and Finding an Optimal Trading Policy

---

Author:
Raj PATEL

Supervisor:
Dr. Warren B. POWELL

Twitter Trading

- Table of contents

Twitter Trading

- Table of contents

Twitter Trading

- Uncertainty model

Twitter Trading

4.1 Vector Autoregressive Model

In this section, we will fit a standard autoregressive time series model in hopes of capturing the joint behavior of the three time series central to this thesis:

(I) Twitter bull ratio in 15 minute intervals;

(II) Twitter tweet volume in 15 minute intervals;

(III) Stock returns in 15 minute intervals.

These three time series exhibit significant correlation with each other and thus lend themselves to the vector autoregression (VAR) model (Carmona, 2014). Let $\boldsymbol{Y}_t$ be our 3-dimensional time series vector. That is, $\boldsymbol{Y}_i$ is a 3 x 1 vector which entries 1, 2, and 3 corresponding to the i^{th} observation of time series (I), (II), and (III) respectively. We can model the behavior of $\boldsymbol{Y}_t$ as an autoregressive series of lag order p with the following formula:

$$\boldsymbol{Y}_t = A_1\boldsymbol{Y}_{t-1} + A_2\boldsymbol{Y}_{t-2} + \ldots + A_p\boldsymbol{Y}_{t-p} + \boldsymbol{W}_t \tag{4.1}$$

Twitter Trading

- Twitter bullish ratio time series

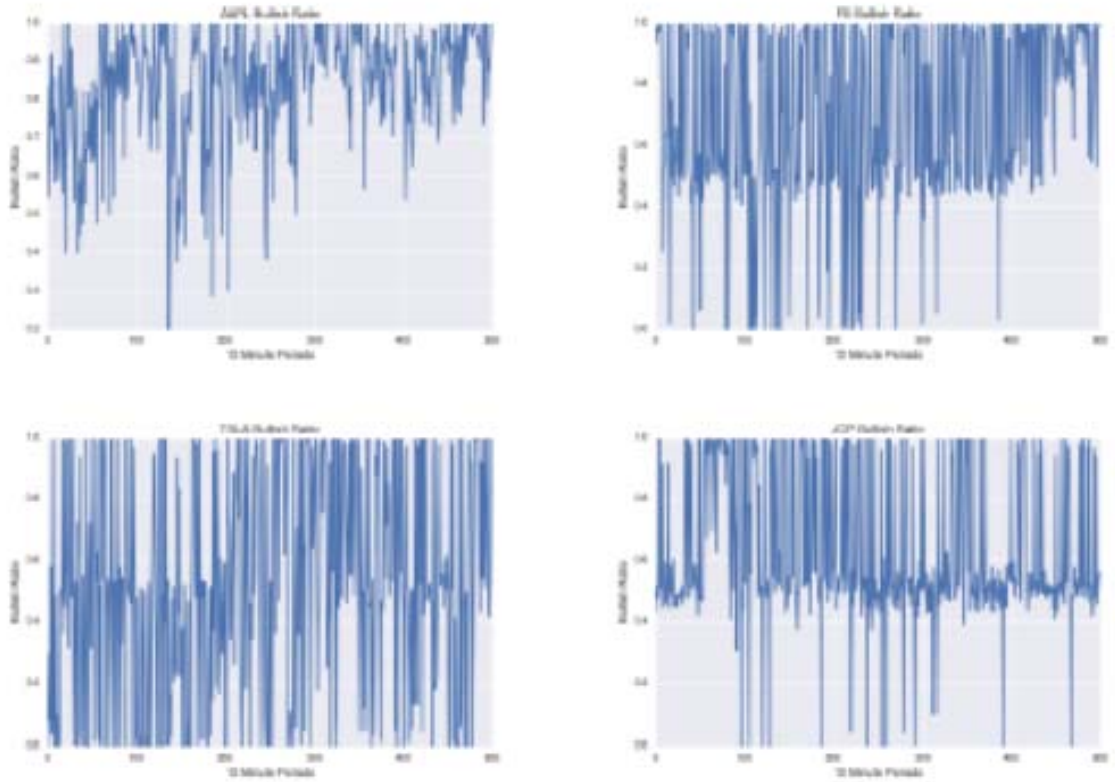


FIGURE 4.1: Bullish Ratio time series for the first 500 15-minute periods of 2015. Only four of the eight stocks are shown (AAPL, FB, TSLA, JCP).

Twitter Trading

- Twitter volume time series

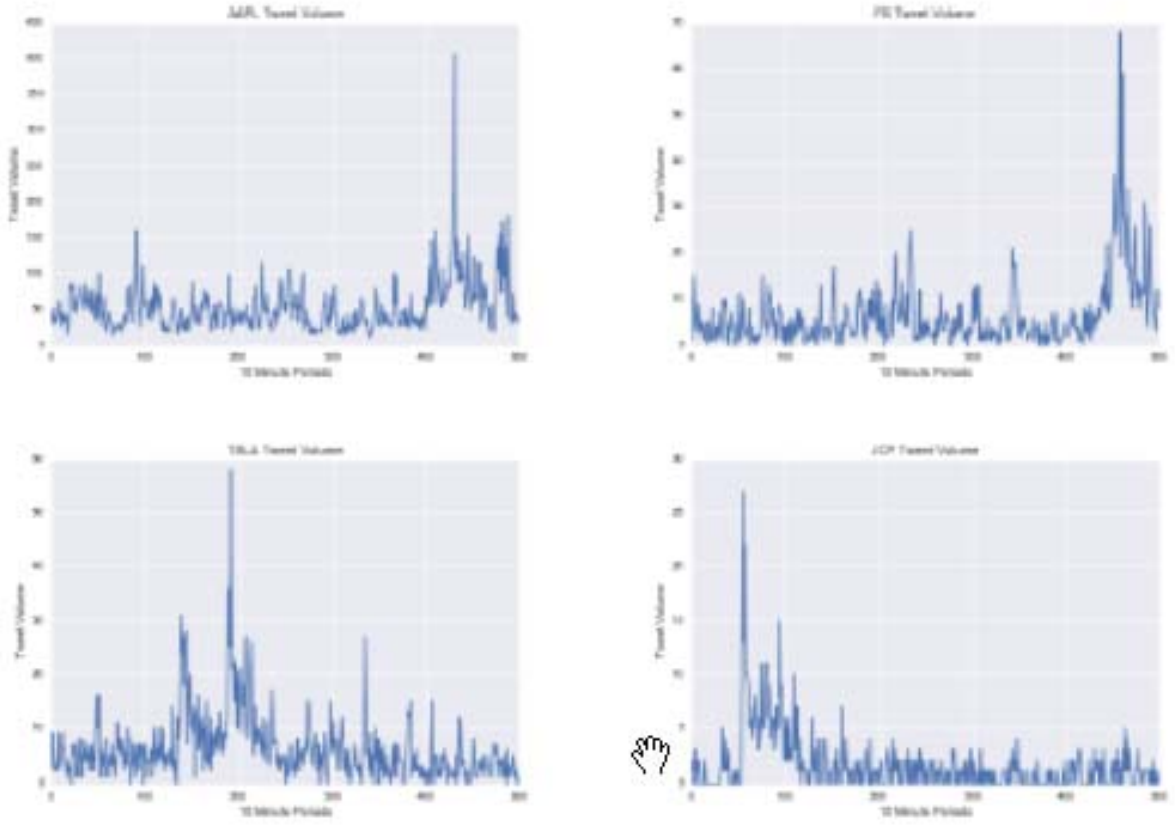


FIGURE 4.2: Twitter volume time series for the first 500 15-minute periods of 2015. Only four of the eight stocks are shown (AAPL, FB, TSLA, JCP).

Twitter Trading

- Stock price time series

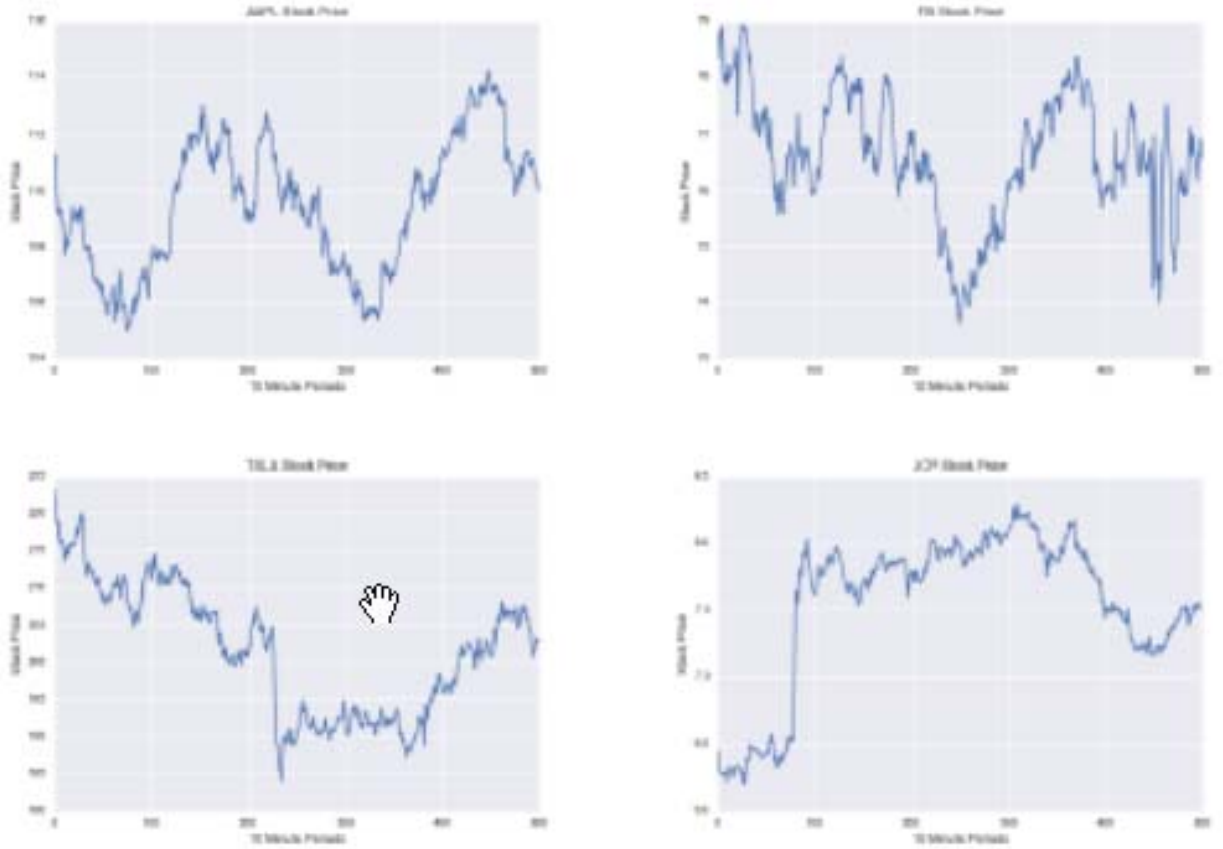


FIGURE 4.3: Stock price time series for the first 500 15-minute periods of 2015. Only four of the eight stocks are shown (AAPL, FB, TSLA, JCP).

Twitter Trading

- Fitting vector-auto regressive model

4.1.4 Time Series Model Fitting

We aim to fit the following model to our three time series with the intent of forming a joint stochastic model we can use for simulation:

$$\boldsymbol{Y}_t = A_1\boldsymbol{Y}_{t-1} + A_2\boldsymbol{Y}_{t-2} + \ldots + A_p\boldsymbol{Y}_{t-p} + \boldsymbol{W}_t \tag{4.1}$$

where

$$\boldsymbol{Y}_t = \begin{bmatrix} Bullish\_Ratio_t \\ Tweet\_Volume_t \\ Stock\_Diff_t \end{bmatrix}$$

Twitter Trading

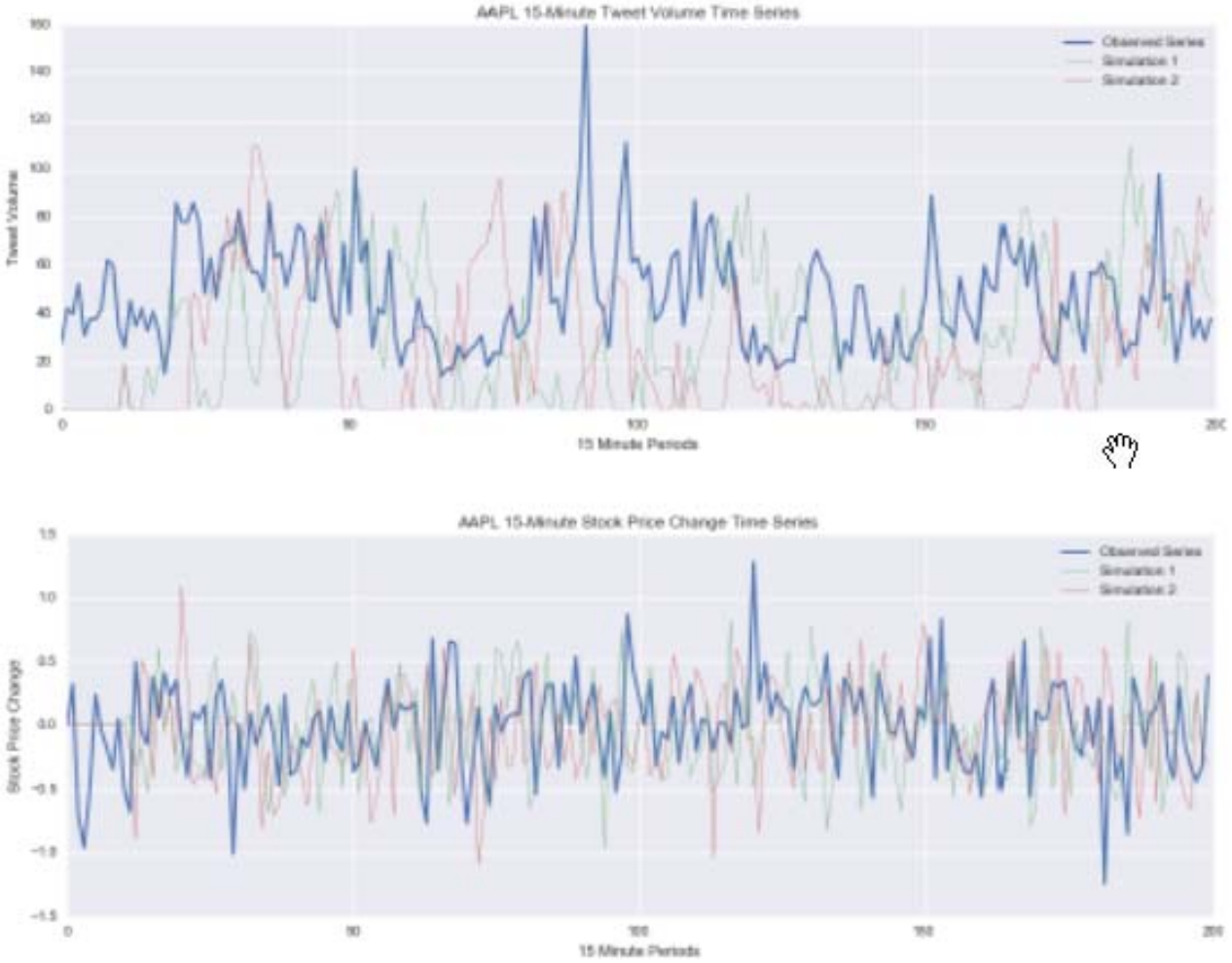


FIGURE 4.5: 200-period VAR simulations of bullish ratio, tweet volume, and first-difference of stock price for Apple, Inc. The simulations are plotted alongside a sample of the true series.

Twitter Trading

- Error distribution:

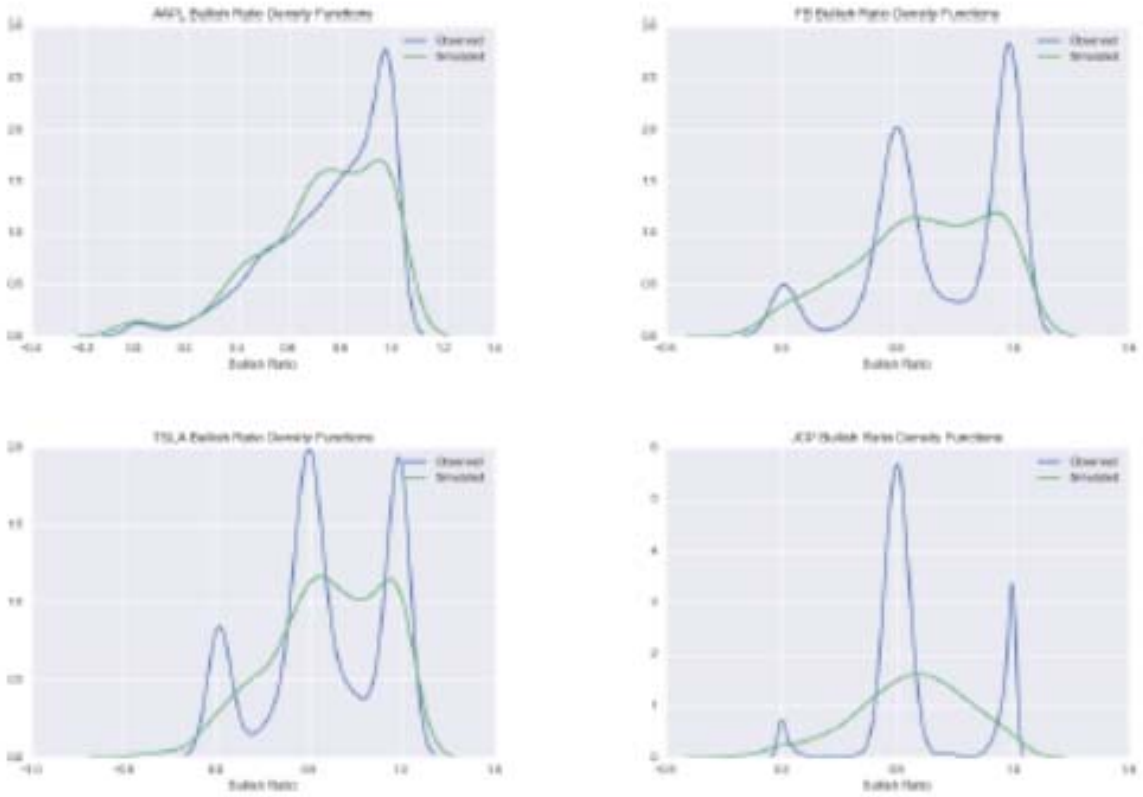
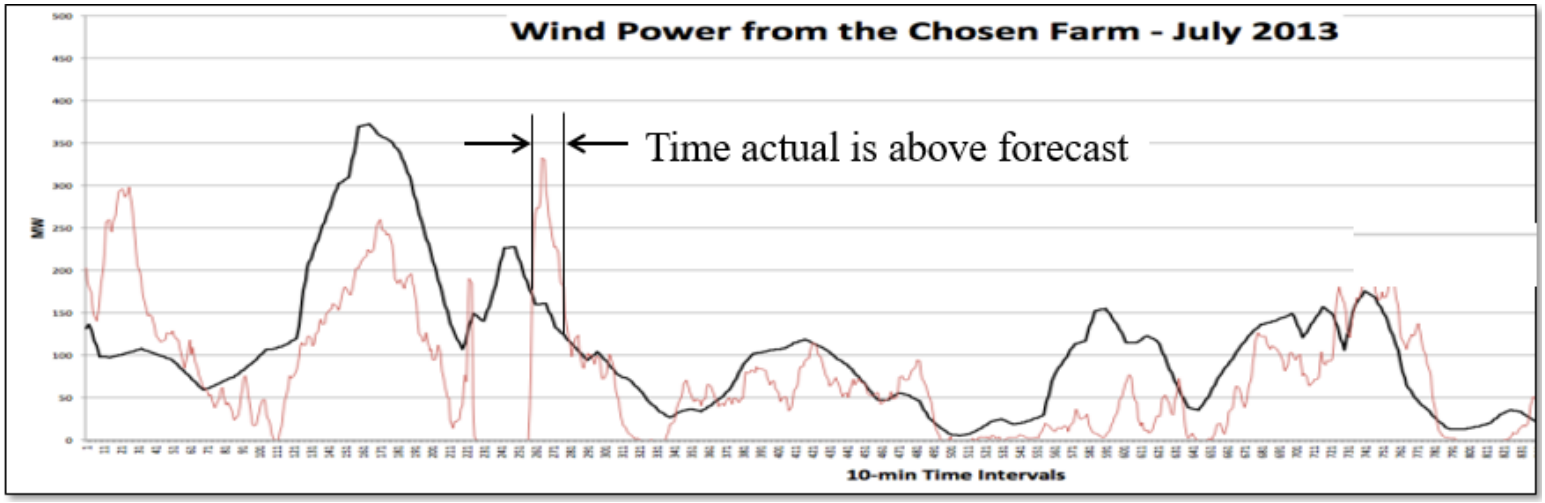
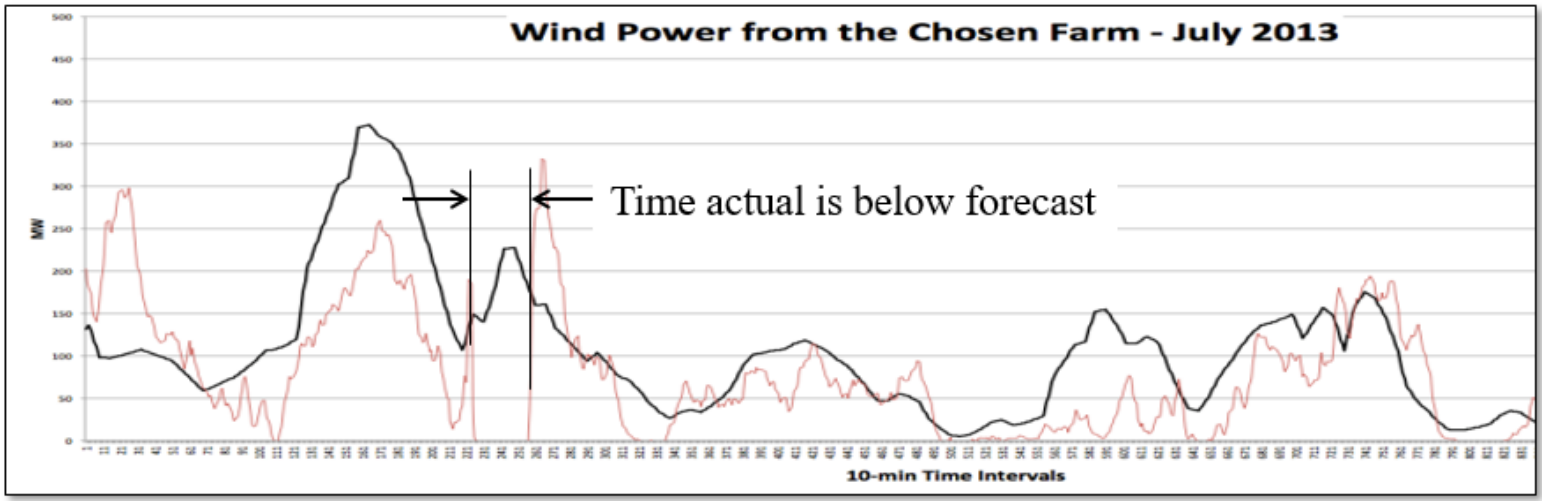


FIGURE 4.6: Empirical density functions for simulated and observed bullish ratio time series for AAPL, FB, TSLA, and JCP. Appendix B contains these the empirical density functions for all eight stocks in the sample.

Twitter Trading

- Crossing times:





Twitter Trading

- The vector autoregressive model did not reproduce the crossing times:

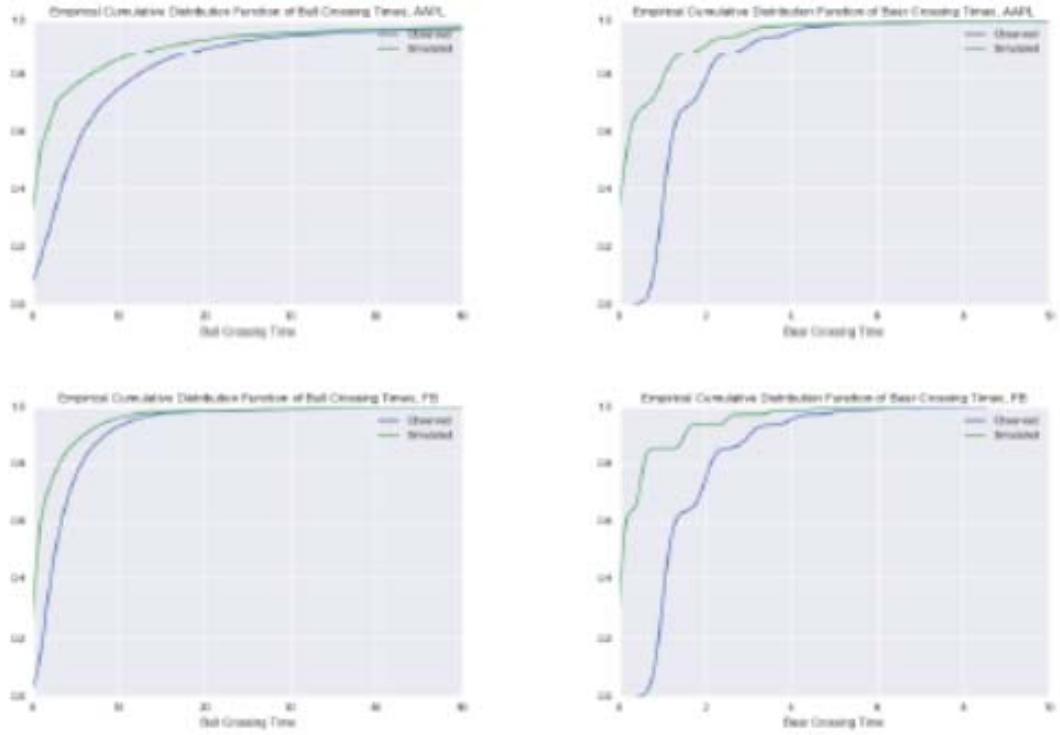


FIGURE 4.7: Empirical CDFs for simulated and observed bullish ratio time series crossing times. The top two plots for are for AAPL and the bottom two are for FB. Appendix C contains these the empirical CDFs for all eight stocks in the sample.

Twitter Trading

- We use a hybrid Markov chain model with two stage variables:
  - » The crossing state S_t^C:

$$S_t^C = \begin{cases} A\,|\,(\text{S/M/L}) & \text{If we are in an "above the forecast" state } |\text{ S/M/L} \\ B\,|\,(\text{S/M/L}) & \text{If we are in a "below the forecast" state } |\text{ S/M/L} \end{cases}$$

A/B means above or below
S/M/L means a short, medium or long crossing distribution

$\mathbb{P}(S_{t+1}^C \,|\, S_t^C)$ is estimated from historical data.

  - » The wind speed W_t given the crossing state:

W_t = Wind speed at time t.

W_t^g = Wind speed aggregated into 5 ranges.

$\mathbb{P}(W_{t+1} \,|\, W_t^g, S_t^C)$ = Density of W_{t+1} given W_t^g and S_t^C

Twitter Trading

- Hidden-state Markov chain

$$P^{CS} = \begin{array}{c} \\ (Bullish,1) \\ (Bullish,2) \\ (Bullish,3) \\ (Bearish,1) \\ (Bearish,2) \\ (Bearish,3) \end{array} \begin{array}{c} \begin{array}{cccccc} (Bullish,1) & (Bullish,2) & (Bullish,3) & (Bearish,1) & (Bearish,2) & (Bearish,3) \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0.72 & 0.11 & 0.17 \\ 0 & 0 & 0 & 0.26 & 0.63 & 0.11 \\ 0 & 0 & 0 & 0.44 & 0.39 & 0.17 \\ 0.79 & 0.08 & 0.13 & 0 & 0 & 0 \\ 0.55 & 0.14 & 0.31 & 0 & 0 & 0 \\ 0.82 & 0.09 & 0.09 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

FIGURE 4.12: Crossing state transition matrix for $Q = 3$. Rows represent crossing states before a transition, and columns represent crossing states after a transtion. Entry $P_{i,j}$ corresponds to $P((S_{t+1}^{CS} = j|S_t^{CS} = i)$.

Twitter Trading

- Error distribution:

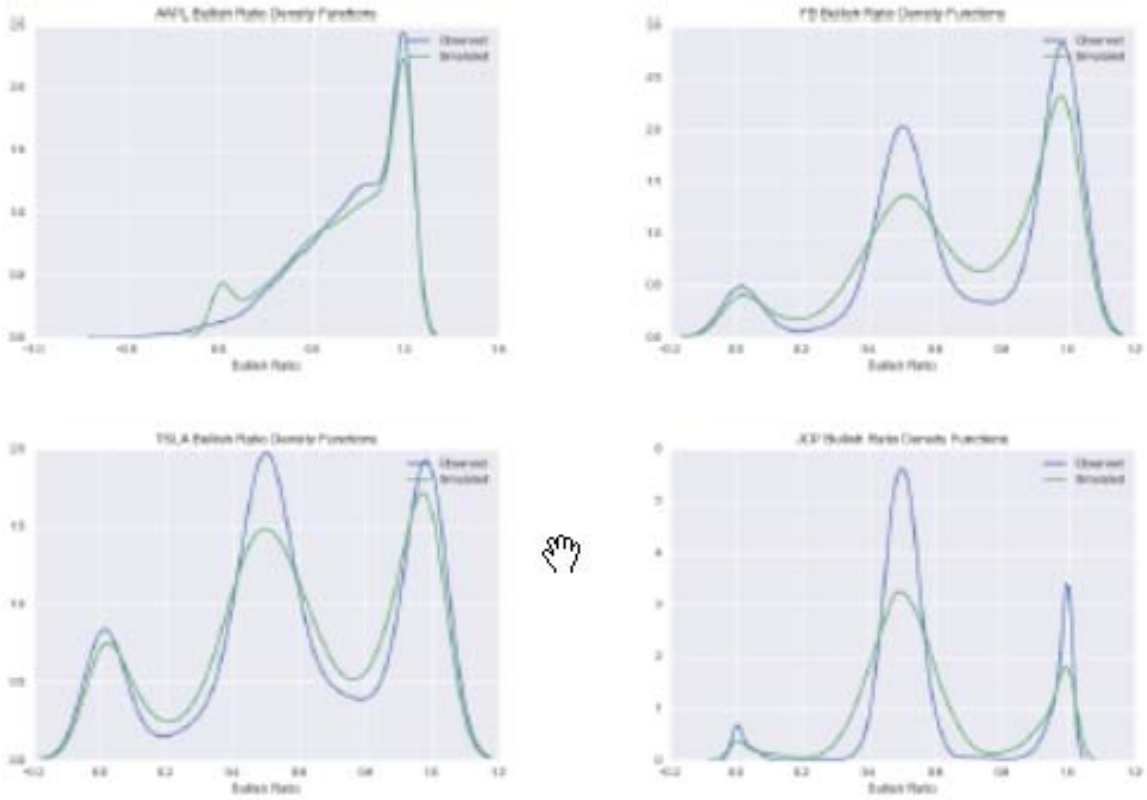


FIGURE 4.14: Empirical density functions for HSMM simulated and observed bullish ratio time series for AAPL, FB, TSLA, and JCP. Appendix G contains these the empirical density functions for all eight stocks in the sample.

Twitter Trading

- Crossing time distribution

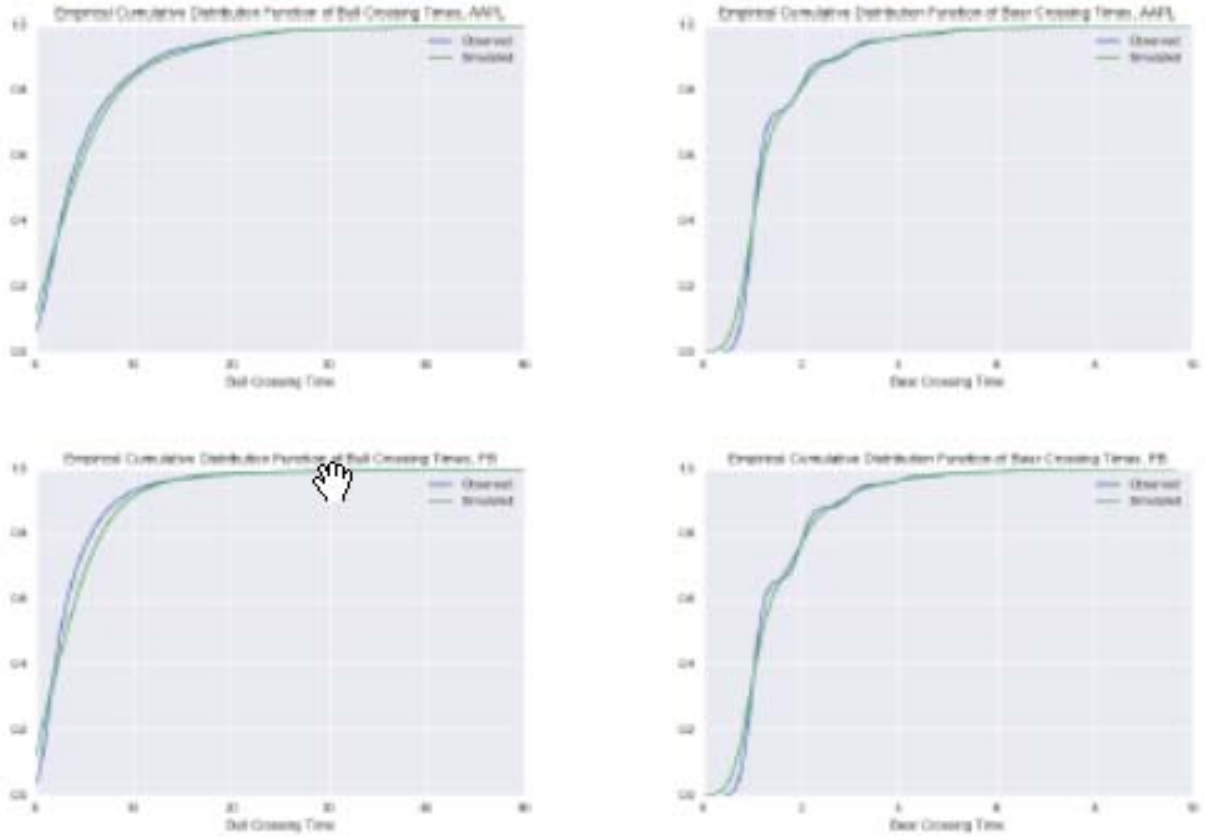
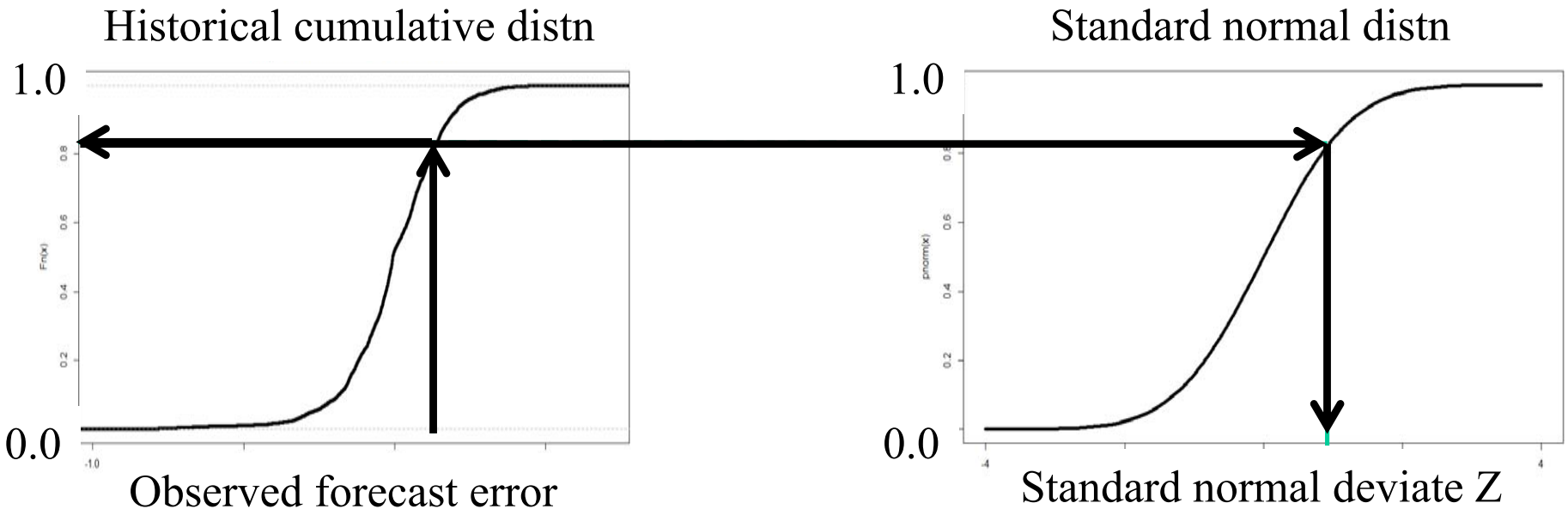


FIGURE 4.15: Empirical CDFs for simulated and observed bullish ratio time series crossing times. The top two plots for are for AAPL and the bottom two are for FB.. Appendix H contains these the empirical CDFs for all eight stocks in the sample.

Twitter Trading

- Note:
  - » The error distribution is accomplished by using the transformation from actual observations (e.g. prices, feeds) to uniform to normal, doing statistical modeling on the normal deviates, and then transforming back from normal through the empirical distribution to get actuals.



Twitter Trading

Twitter Trading

- State variables

The state variable S_t takes the following form for company A:

$$S_t^A = (R_t^A, b_t^A, v_t^A, p_t^A, D_t^A)$$

- R_t: the amount of company A stock held at time t

$$R_t^A = \begin{cases} 1 & \text{if 1 unit of stock is currently held} \\ 0 & \text{if 0 units of stock are currently held} \end{cases}$$

- b_t^A: the Twitter bullish ratio for company A at time t. $b_t^A \in [0, 1]$ for all t.
- v_t^A: the Twitter volume (total number of tweets) about company A at time t. $v_t^A \geq 0$ for all t.
- p_t^A: company A's stock price at time t. $p_t^A \geq 0$ for all p_t^A.
- D_t^A: current amount of cash on hand at time t. $D_0^A = 0$ and is impacted by buy/sell decisions.

Twitter Trading

- Decision variables
  - At each time t, we must decide whether to buy stock, hold our current position, or sell stock. This decision, x_t, has the following potential values:

$$x_t \in \begin{cases} \{0,1\} & \text{if } R_t = 0 \\ \{-1,0\} & \text{if } R_t = 1 \end{cases}$$

That is, if no values of stock are held at time t, the decision x_t at time t can only be to continue holding no stock or purchasing one unit of stock. If one unit of stock is held at time t, the decision x_t at time t can only be to continue holding one stock or selling one unit of stock.

Twitter Trading

- Exogenous information
  - » Data from twitter feed

5.3 Exogenous Information

The processes of Twitter bull ratio, tweet volume, and stock price are exogenous. We model this exogenous information using the HSMM method developed in Chapter 4. The transition functions of these processes b_t^A, v_t^A, p_t^A are described in the following section.

Twitter Trading

Transition function

- Twitter bullish ratio, tweet volume, and the stock price move exogenously. In our optimization, these movements will be simulated by the HSMM model developed to simulate these three correlated series. We represent the transition of these processes using the following transition functions.

$$b^A_{t+1} = b^A_t + \hat{b}^A_t$$

$$v^A_{t+1} = v^A_t + \hat{v}^A_t$$

$$p^A_{t+1} = p^A_t + \hat{p}^A_t$$

- The current stock holding R^A_t changes according to our policy decision at time t. Therefore, R_t transitions according to the following function:

$$R^A_{t+1} = R^A_t + x_t$$

- Buying and selling stock impacts our cash. Buying stock reduces our cash by the current stock price and selling stock increases our cash.

$$D^A_{t+1} = D^A_t - x_t p_t$$

Twitter Trading

- Objective function

To determine the optimal parameter vector θ, we maximize the following objective function:

$$\max_{\theta} F(\theta) = \boldsymbol{E} \sum_{t=0}^{T} C_t(S_t, A^{\pi}(S_t|\theta)) = \boldsymbol{E}[D_T + R_T p_T|\theta]$$

This objective function represents our goal of maximizing profit at the end of the time period T. The value at the end of the period is measured by the value of our cash plus the value of any held stock.

The optimal policy is determined by solving the optimization problem with respect to $\theta = (\theta_{BR}^{Buy}, \theta_{VOL}^{Buy}, \theta_{STOCK}^{Buy}, \theta_{BR}^{Sell}, \theta_{VOL}^{Sell}, \theta_{STOCK}^{Sell})$. To maximize the $F(\theta)$, we use a version of the stochastic gradient algorithm. The optimization procedure is outlined in the next section.

Twitter Trading

- Designing policies

Our goal is determine an optimal trading strategy that maximizes our profit. To do so, we will implement a parametric policy model, a subclass of policy function approximation (Powell, 2010). Our policy will determine whether to buy, hold, or sell a unit of stock at a given time based on a six-dimensional parameter vector: $\theta = (\theta^{Buy}_{BR}, \theta^{Buy}_{VOL}, \theta^{Buy}_{STOCK}, \theta^{Sell}_{BR}, \theta^{Sell}_{VOL}, \theta^{Sell}_{STOCK})$. This parametric policy is used to make decisions in the following way.

If $R^A_t = 0$:

- $x_t = 1$ if $b^A_t > \theta^{Buy}_{BR} \quad \wedge \quad v^A_t > \theta^{Buy}_{VOL} \quad \wedge \quad p^A_t < \theta^{Buy}_{STOCK}$
- $x_t = 0$ otherwise

If $R^A_t = 1$:

- $x_t = -1$ if $b^A_t < \theta^{Sell}_{BR} \quad \wedge \quad v^A_t > \theta^{Sell}_{VOL} \quad \wedge \quad p^A_t > \theta^{Sell}_{STOCK}$
- $x_t = 0$ otherwise

Twitter Trading

- Evaluating policies
  - » In a simulator:
    - Using data from history
      - Limited to what we observe
      - Need to be careful to keep training and testing data separate
    - Using samples from a mathematical model
      - This requires using our stochastic model to generate samples.
  - » In the real world
    - Similar to running simulations from history

Twitter Trading

- Extensions

Student decision problem

Template

- Narrative

- State variables

- Decision variables

- Exogenous information

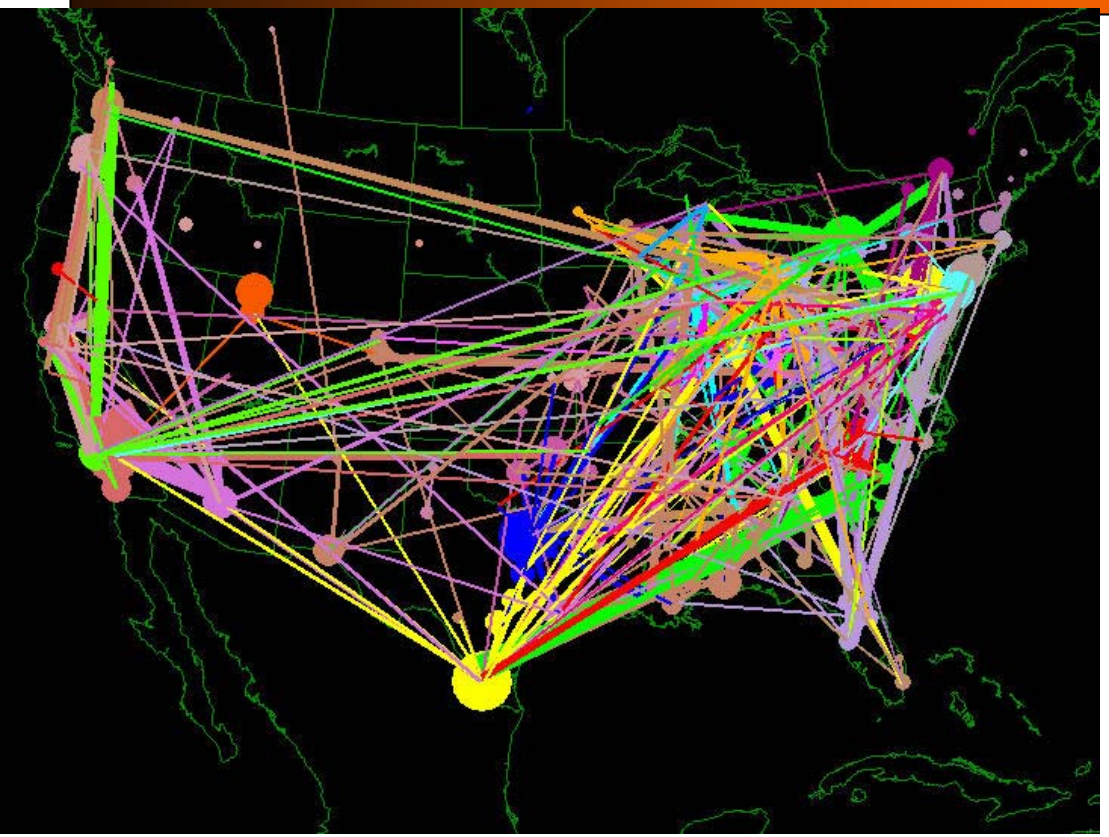
- Transition function

- Objective function

Week 12 – Wednesday

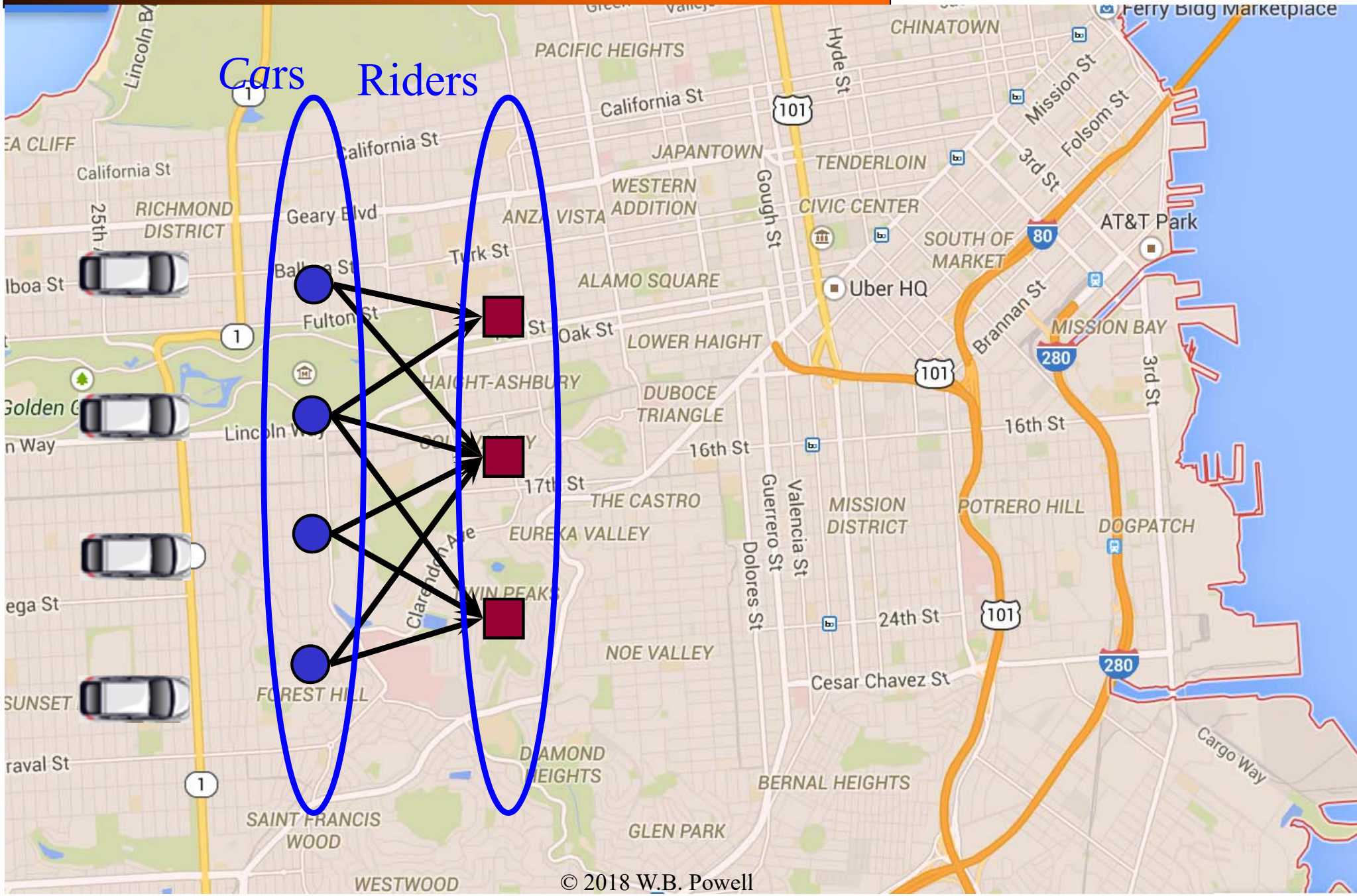
Summary lecture

Fleet management

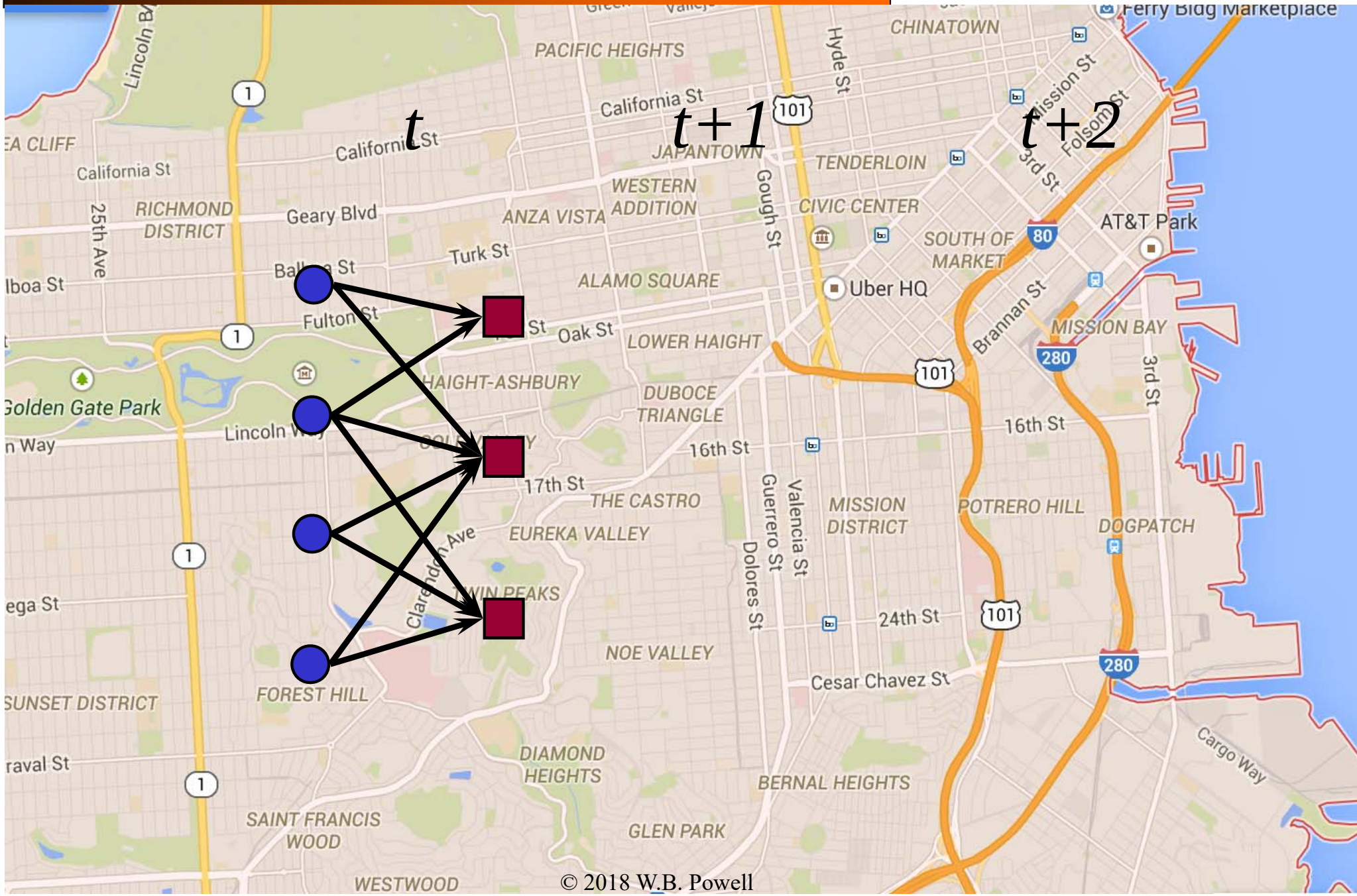


- Fleet management problem
  - » Optimize the assignment of drivers to loads over time.
  - » Tremendous uncertainty in loads being called in

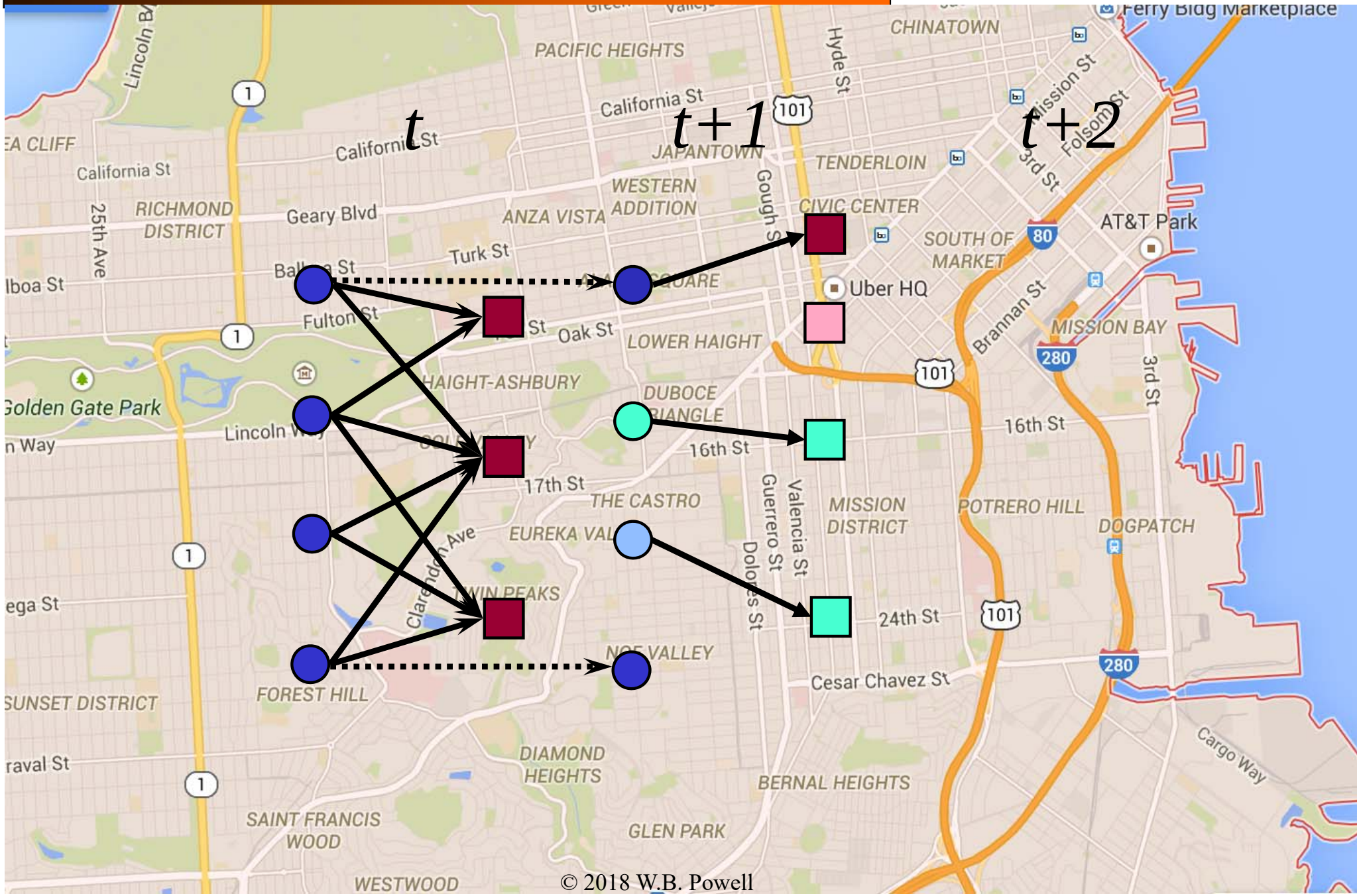
Cost function approximations



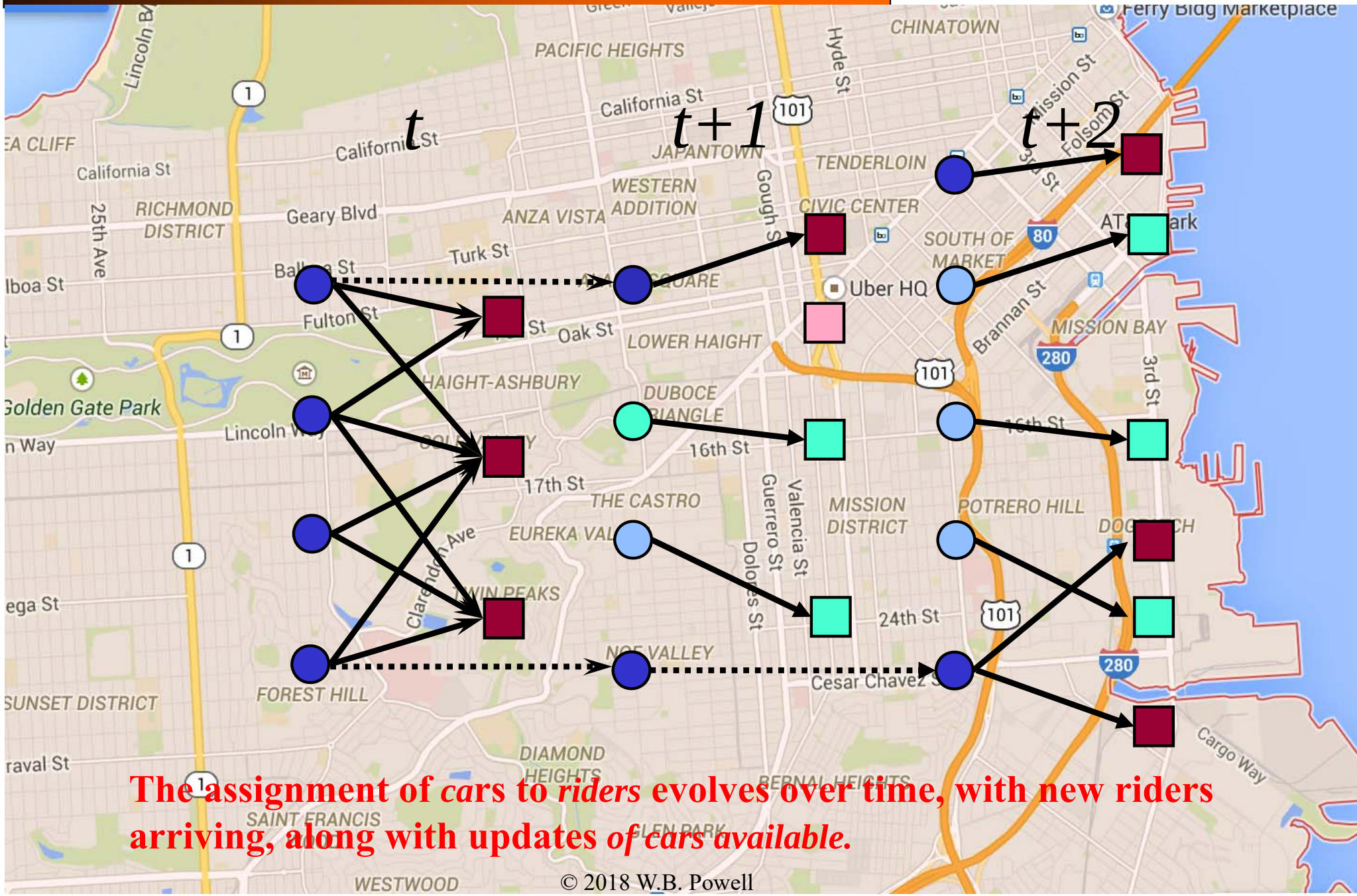
Optimizing over time



Optimizing over time

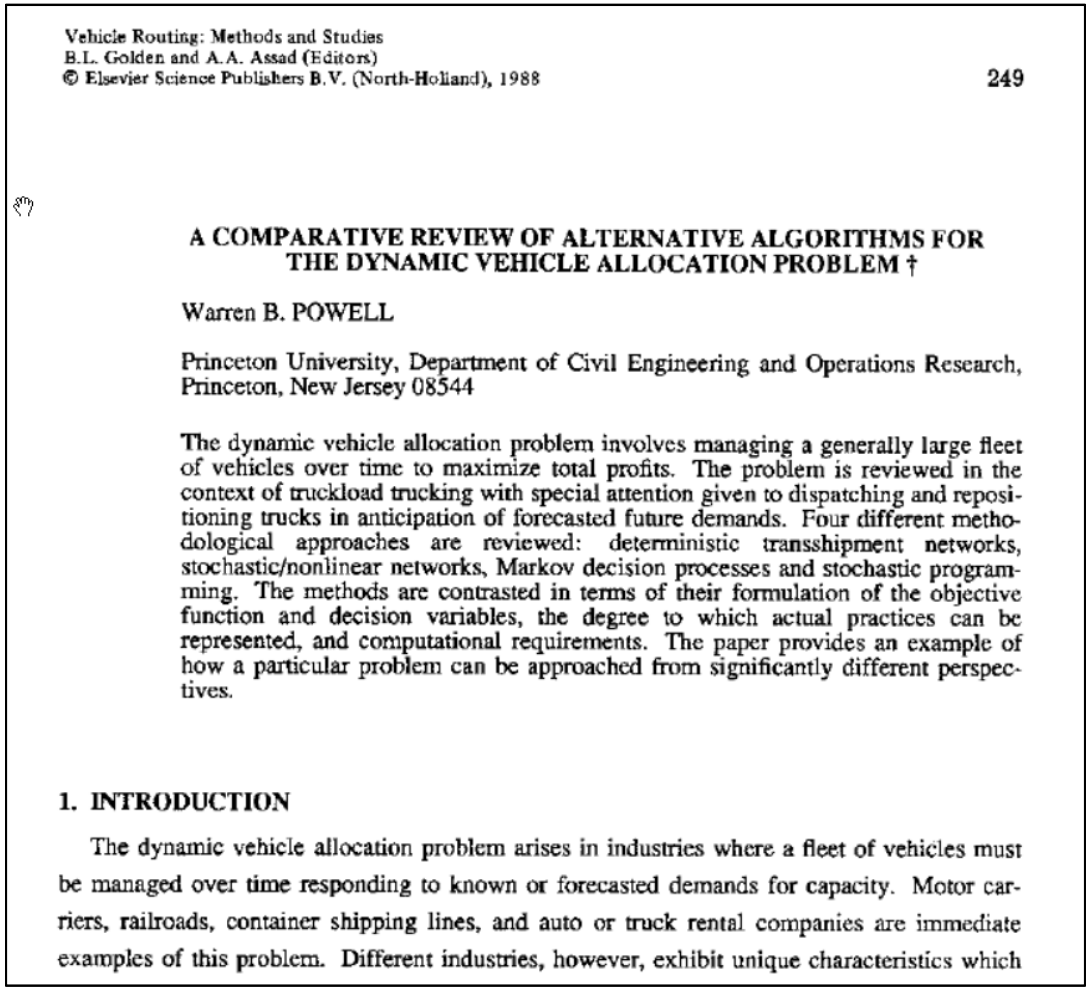


Optimizing over time



The assignment of *cars* to *riders* evolves over time, with new riders arriving, along with updates *of cars available.*

- A bit of history
  - » The dynamic assignment problem (assigning drivers to loads) in truckload trucking.
  - » Today, we would recognize this as the problem faced by Uber and Lyft.
  - » The challenge was sequencing decisions (assigning drivers to loads) and information (e.g. new customers calling in).
  - » Four completely different models, and none came close to allowing me to solve the problem.



Vehicle Routing: Methods and Studies
B.L. Golden and A.A. Assad (Editors)

A COMPARATIVE REVIEW OF ALTERNATIVE ALGORITHMS FOR THE DYNAMIC VEHICLE ALLOCATION PROBLEM †

Warren B. POWELL

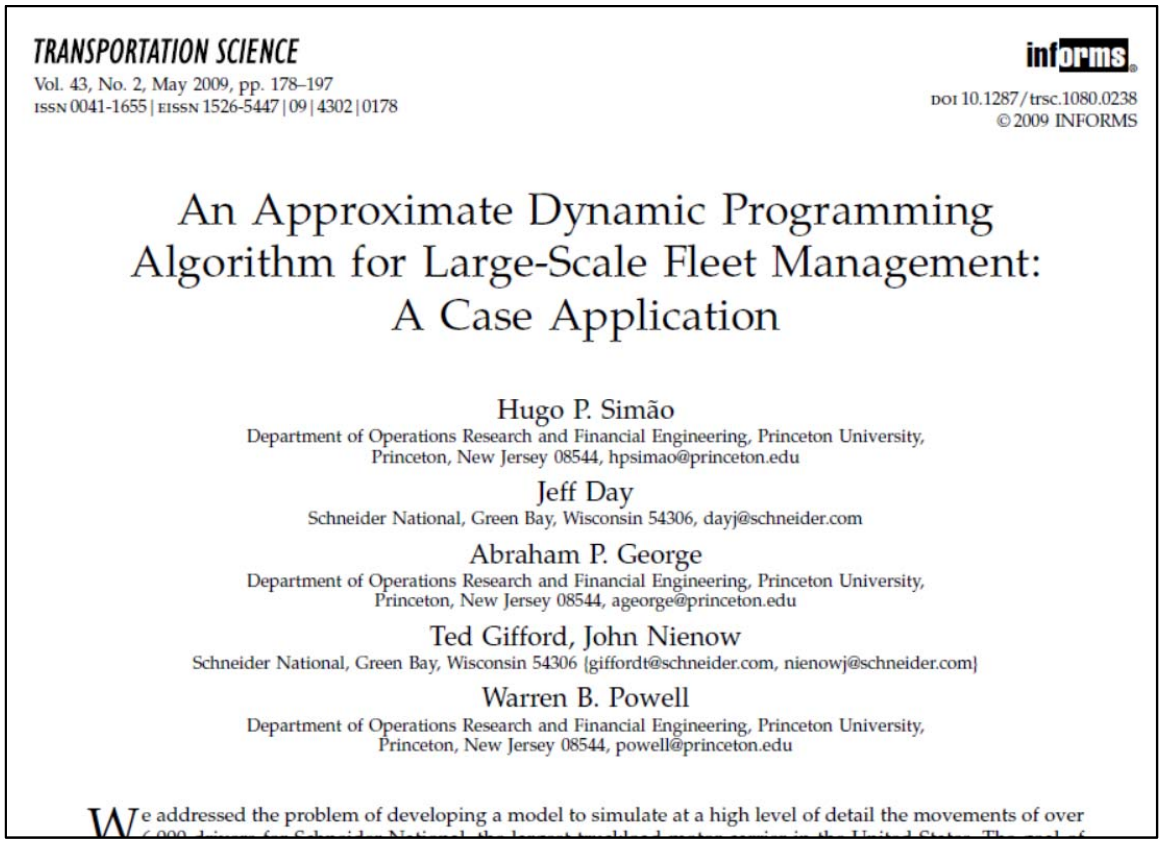
Princeton University, Department of Civil Engineering and Operations Research, Princeton, New Jersey 08544

The dynamic vehicle allocation problem involves managing a generally large fleet of vehicles over time to maximize total profits. The problem is reviewed in the context of truckload trucking with special attention given to dispatching and repositioning trucks in anticipation of forecasted future demands. Four different methodological approaches are reviewed: deterministic transshipment networks, stochastic/nonlinear networks, Markov decision processes and stochastic programming. The methods are contrasted in terms of their formulation of the objective function and decision variables, the degree to which actual practices can be represented, and computational requirements. The paper provides an example of how a particular problem can be approached from significantly different perspectives.

1. INTRODUCTION

The dynamic vehicle allocation problem arises in industries where a fleet of vehicles must be managed over time responding to known or forecasted demands for capacity. Motor carriers, railroads, container shipping lines, and auto or truck rental companies are immediate examples of this problem. Different industries, however, exhibit unique characteristics which

- Over 20 years later:
  - » Winner, 2009 Daniel H. Wagner Prize.
  - » Winner, 2010 Best paper prize in Transportation Science and Logistics.
  - » 2016, licensed to Optimal Dynamics where it is being marketed as SMART-TL.



TRANSPORTATION SCIENCE
Vol. 43, No. 2, May 2009, pp. 178–197
ISSN 0041-1655 | EISSN 1526-5447 | 09 | 4302 | 0178

informs

DOI 10.1287/trsc.1080.0238

An Approximate Dynamic Programming Algorithm for Large-Scale Fleet Management: A Case Application

Hugo P. Simão
Department of Operations Research and Financial Engineering, Princeton University, Princeton, New Jersey 08544, hpsimao@princeton.edu

Jeff Day
Schneider National, Green Bay, Wisconsin 54306, dayj@schneider.com

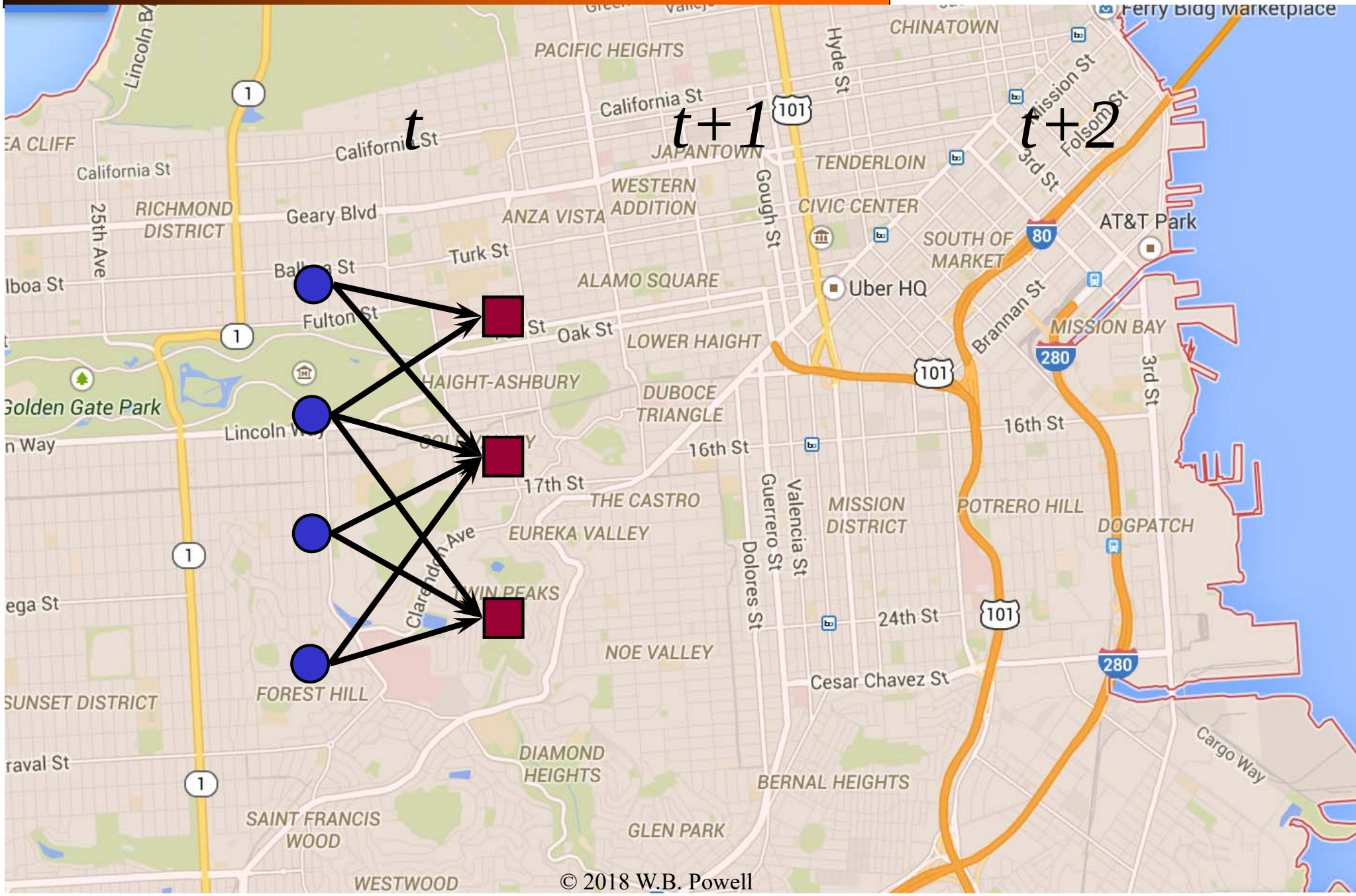
Abraham P. George
Department of Operations Research and Financial Engineering, Princeton University, Princeton, New Jersey 08544, ageorge@princeton.edu

Ted Gifford, John Nienow
Schneider National, Green Bay, Wisconsin 54306 {giffordt@schneider.com, nienowj@schneider.com}

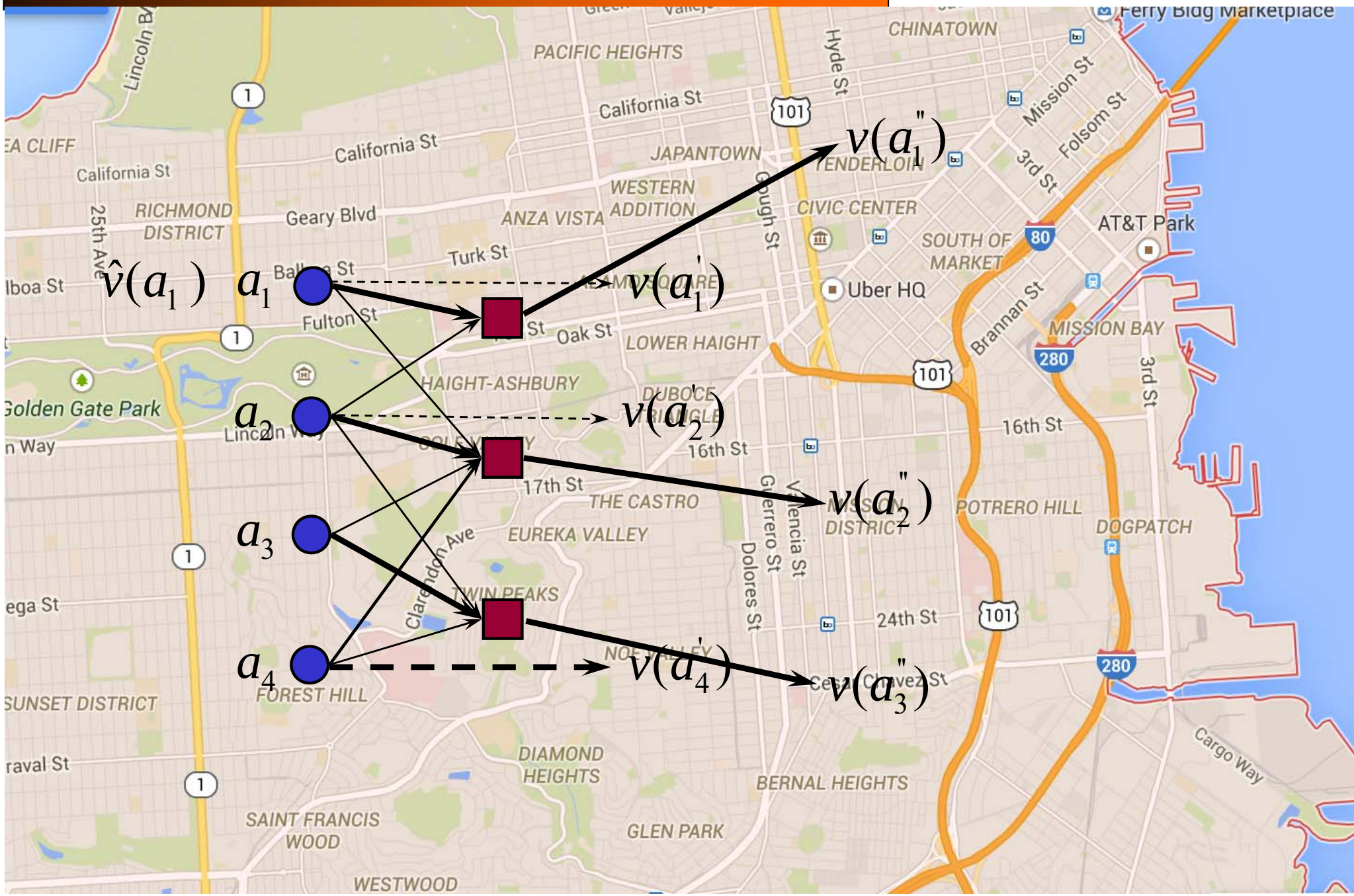
Warren B. Powell
Department of Operations Research and Financial Engineering, Princeton University, Princeton, New Jersey 08544, powell@princeton.edu

We addressed the problem of developing a model to simulate at a high level of detail the movements of over

Driverless fleets of EVs

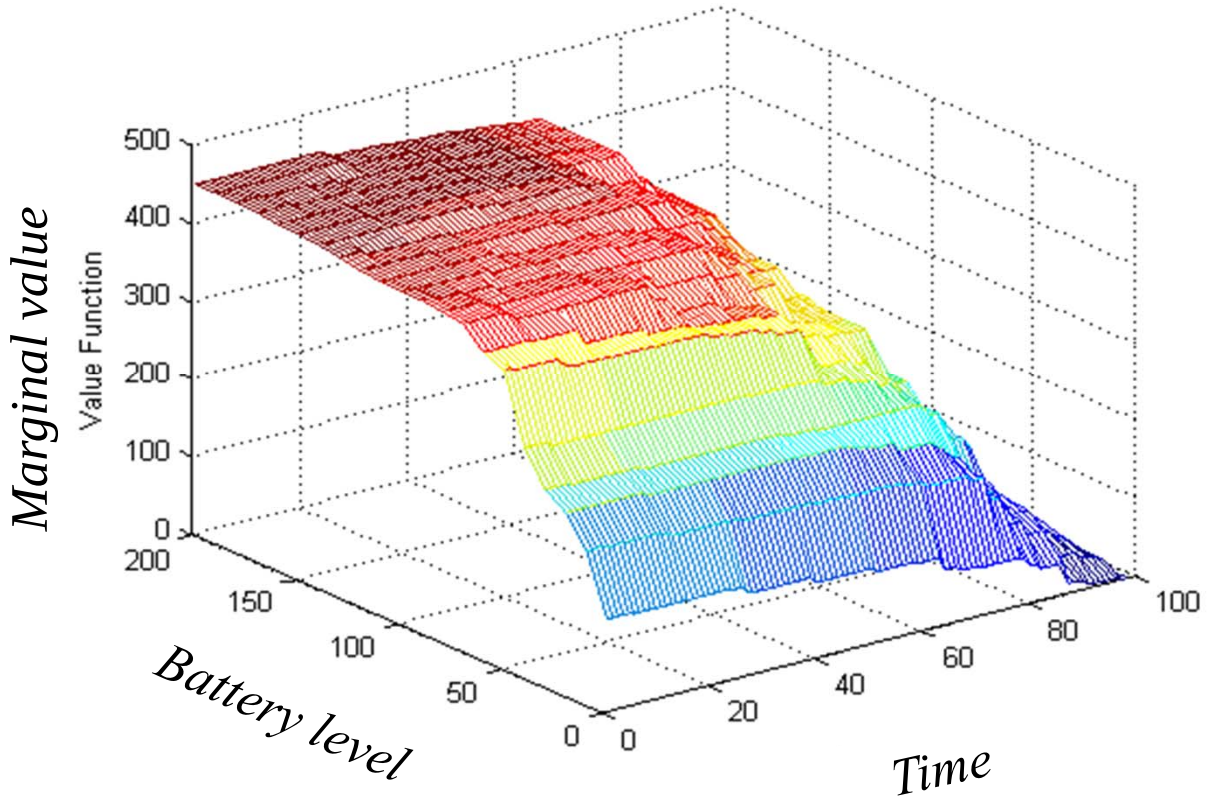


Driverless fleets of EVs



Driverless fleets of EVs using ADP

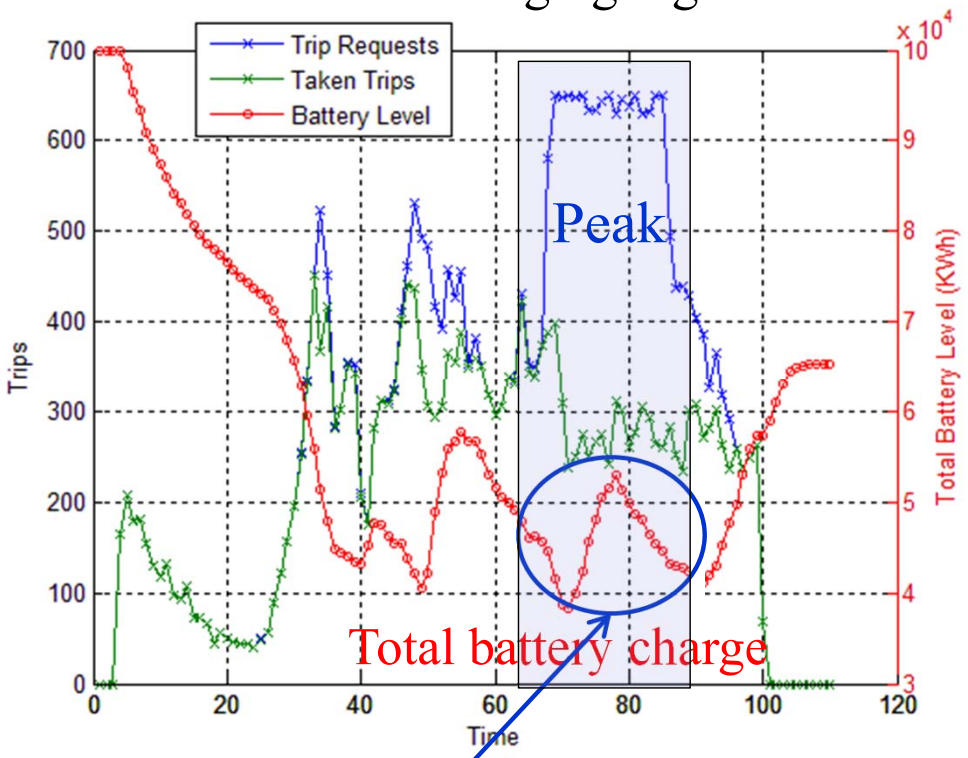
- The value of a vehicle in the future
  - » Value function approximation captures charge level, as well as time and location.
  - » Hierarchical aggregation accelerated the learning process



Driverless fleets of EVs using ADP

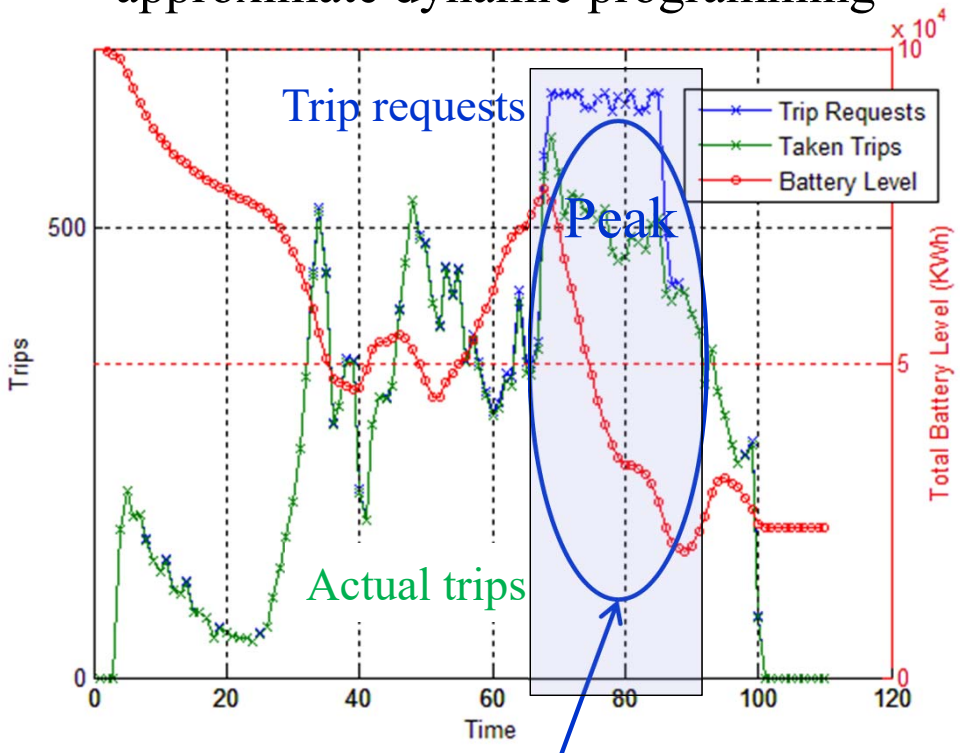
- Heuristic dispatch vs. ADP-based policies
  - » Effect of value function approximations on recharging

Heuristic recharging logic



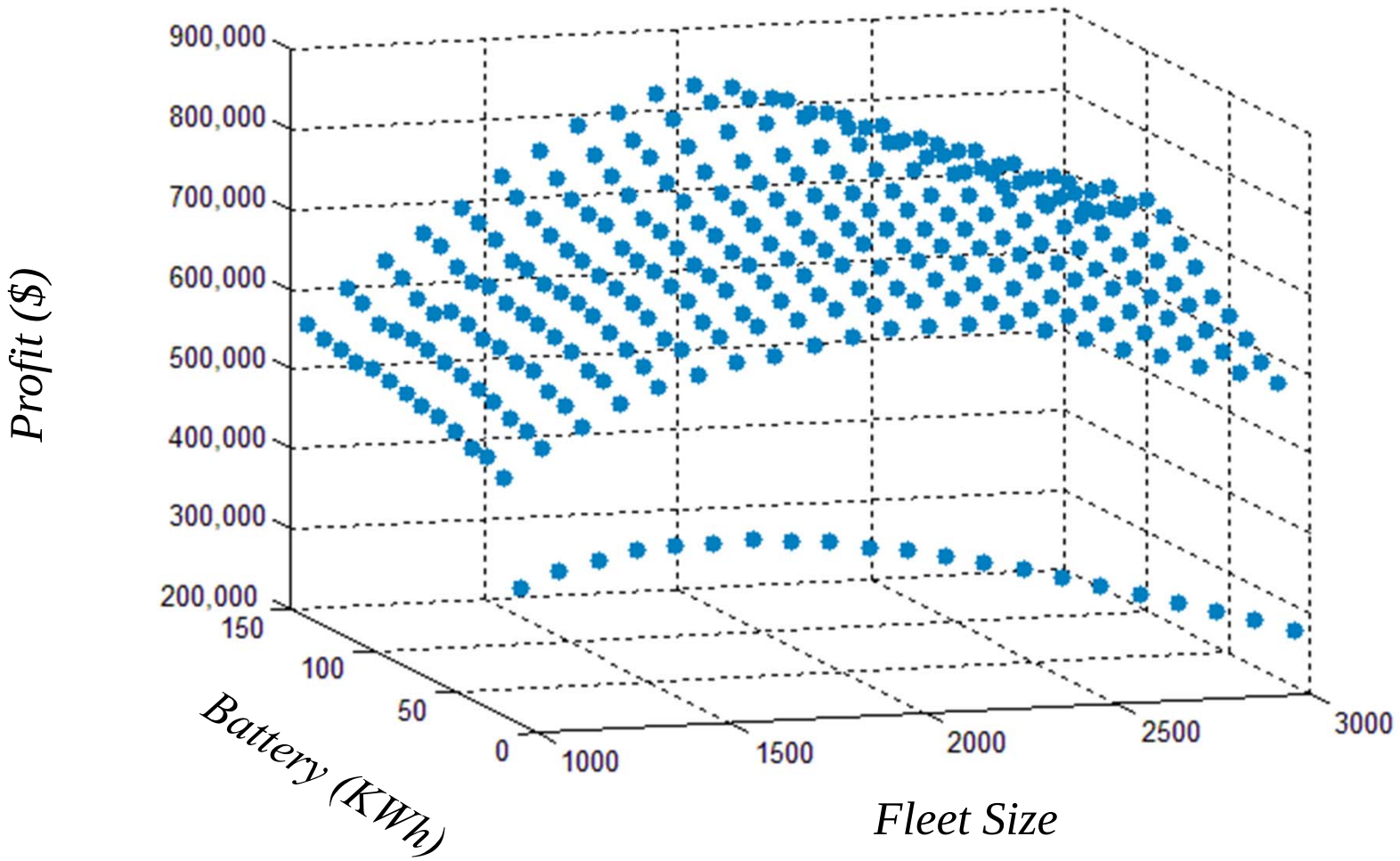
Recharging during peak period

Recharging controlled by approximate dynamic programming



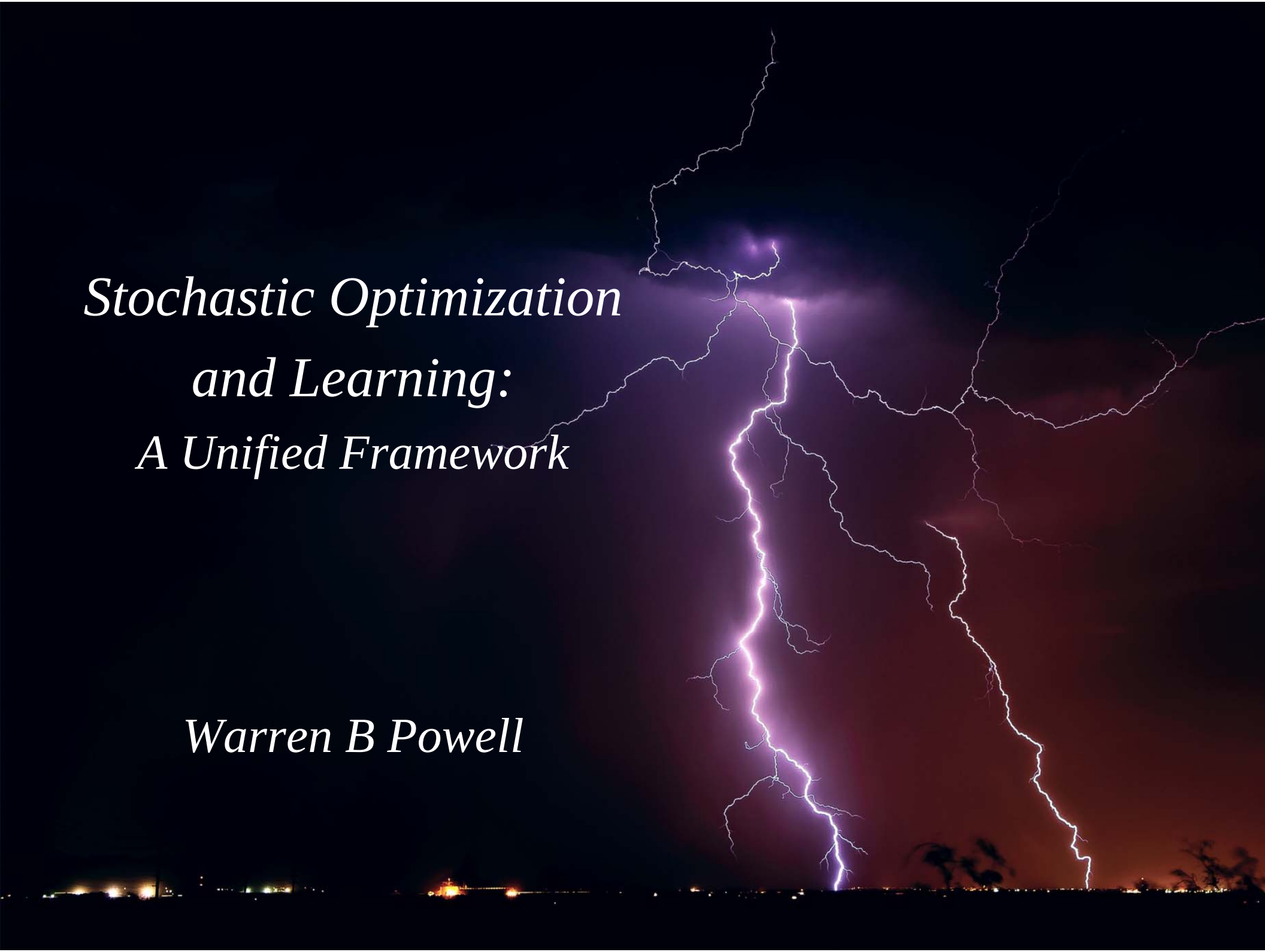
No recharging during peak period

The economics of driverless fleets



We can simulate different fleet sizes and battery capacities, properly modeling recharging behaviors given battery capacity.


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
VOLUME 2 • 4th EDITION
Dynamic Programming and Optimal Control
APPROXIMATE DYNAMIC PROGRAMMING
Wiley Series in Probability and Statistics
Optimal Learning
SECOND EDITION
Model Predictive Control
OPTIMAL CONTROL
Dimitri P. Bertsekas
Athena Scientific
WILEY
Interscience Series in Discrete Mathematics and Optimization
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
Reinforcement Learning
An Introduction
second edition
Richard S. Sutton and Andrew G. Barto
WILEY
Markov Decision Processes
Discrete Stochastic Dynamic Programming
MARTIN L. PUTERMAN
Vol. 1
Online Computation and Competitive Analysis
Allan Borodin
Ran El-Yaniv
43
Jiongmin Yong
Xun Yu Zhou
Stochastic Controls
Hamiltonian Systems and HJB Equations
Modelling and Applied Probability
STOCHASTIC SIMULATION OPTIMIZATION
An Optimal Computing Budget Allocation
Chun-Hung Chen • Loo Hay Lee



Stochastic Optimization and Learning:

A Unified Framework

Warren B Powell

SEQUENTIAL DECISION ANALYTICS AND MODELING:

Modeling exercises with python

Warren B. Powell

December 12, 2018

- Sequential decision problems
  - » Any sequential decision problem can be modeled as

$$S_0, x_0, W_1, S_1, x_1, W_2, S_2, \dots, S_t, x_t, W_{t+1}, \dots$$

  - » Decisions are made with *policies* which are represented using

$$x_t = X^{\pi}(S_t)$$

  - » The system is driven by exogenous information from
  - »

$$S_0, W_1, W_2, \dots, W_t,$$

  - » The goal is to find the best policy.

Elements of a dynamic model

- 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?

Elements of a dynamic model

- The state variable:

Controls community

x_t = "Information state"

Operations research/MDP/Computer science

$S_t = (R_t, I_t, B_t)$ = 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 traffic delays

Belief about the status of equipment

Elements of a dynamic model

- 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 $\theta \in \Theta^f$.

Elements of a dynamic model

- 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 $W_1, W_2, \ldots$.

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

Elements of a dynamic model

- The transition function

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

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

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

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

Also known as the:

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

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

Elements of a dynamic model

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

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

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

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

    - We only care about how well the final decision $x^{\pi,N}$ works.
  - » Risk (not covered in this course):

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

Elements of a dynamic model

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

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

    - Final reward ("offline learning")

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

    - Risk:

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

  - » Transition function:

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

  - » Exogenous information:

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

Problem classes

- State independent problems
  - » The *problem* does not depend on the state of the system.

$$\max_x \mathbb{E}F(x,W) = \mathbb{E}\left\{p\min(x,W) - cx\right\}$$

  - » The only state variable is what we know (or believe) about the unknown function $\mathbb{E}F(x,W)$.
- State dependent problems
  - » Now the *problem* may depend on what we know at time *t*:

$$\max_{0\le x\le R_t} \mathbb{E}\,C(S,x,W) = \mathbb{E}\left\{p_t\min(x,W) - cx\right\}$$

Problem classes

- Offline (final reward)
  - » We can iteratively search for the best solution.
  - » We only care about the final solution.
  - » Asymptotic formulation:

$$\max_x \mathbb{E} F(x, W)$$

  - » Finite horizon formulation:

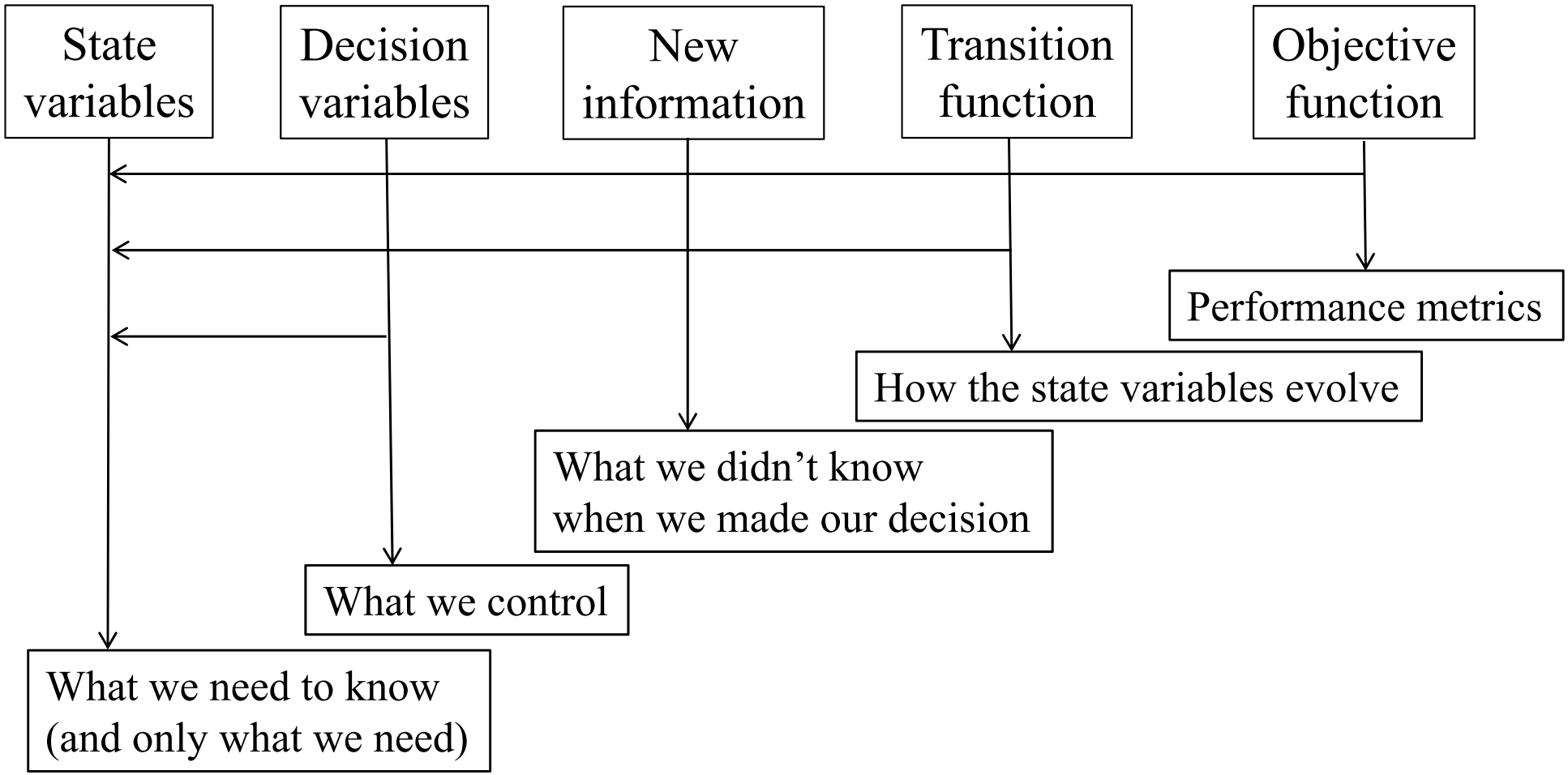
$$\max_\pi \mathbb{E} F(x^{\pi,N}, W)$$

- Online (cumulative reward)
  - » We have to learn as we go

$$\max_\pi \mathbb{E} \sum_{n=0}^{N-1} F(X^\pi(S^n), W^{n+1})$$

Elements of a dynamic model

- The modeling process
  - » I conduct a conversation with a domain expert to fill in the elements of a problem:



- Course themes
  - » The universal modeling framework
    - The five elements of any sequential decision problem.
    - "Model first, then solve."
  - » Statistical learning
    - Five classes of learning problems
  - » Uncertainty modeling
    - What are the types of uncertainty?
    - How to model them?
  - » Designing policies
    - Policy search class (PFAs and CFAs)
    - The lookahead class (VFAs and DLAs)
    - Hybrids
  - » Teach by example style …
    - Need to learn how to transfer methods from one setting to the next.

Basic energy modeling problem

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.

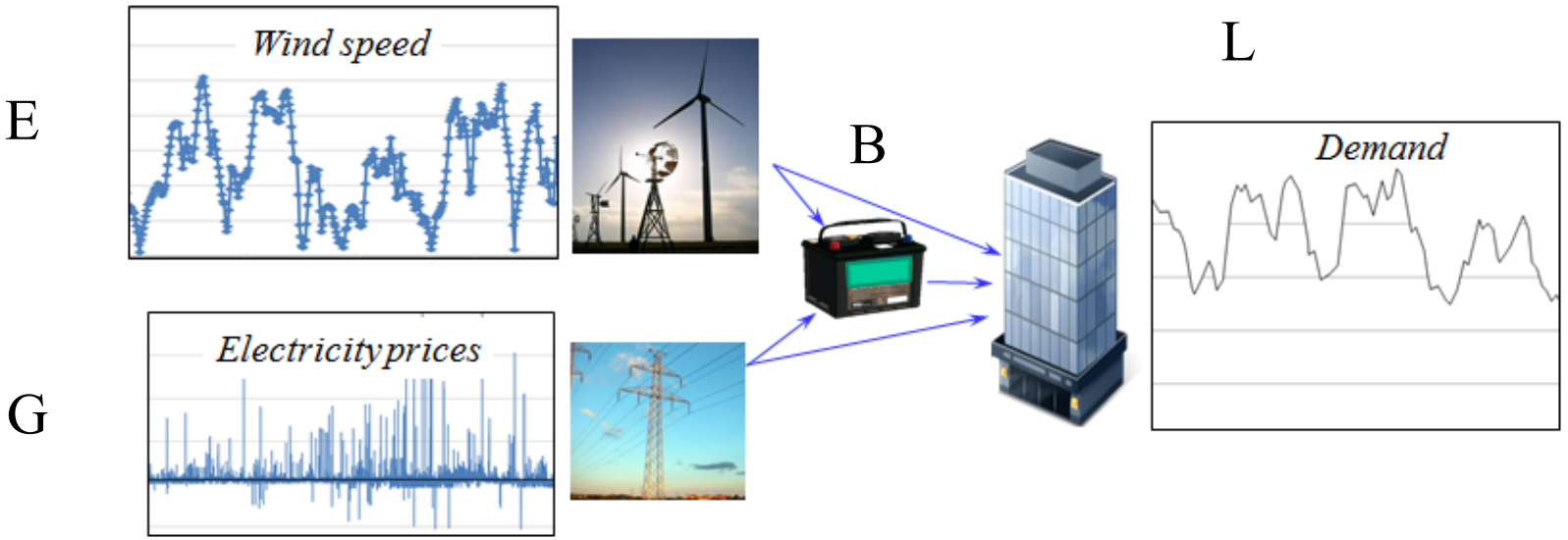
- Notes:
  - » We are going to describe three variations of our energy storage problem:
    - No learning
    - Passive learning
    - Active learning

An energy storage problem

- A model of our problem
  - » State variables
  - » Decision variables
  - » Exogenous information
  - » Transition function
  - » Objective function

An energy storage problem

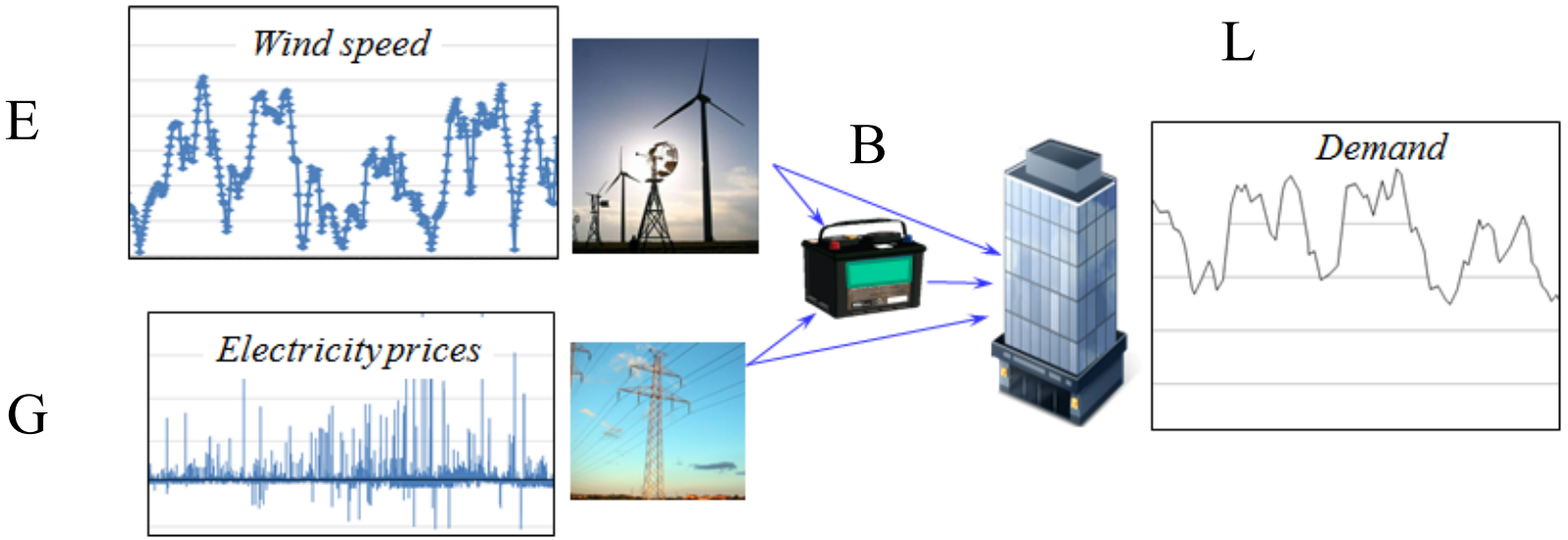
- State variables



» Lay out the five columns of the modeling framework and work through the model.

An energy storage problem

- Decision variables



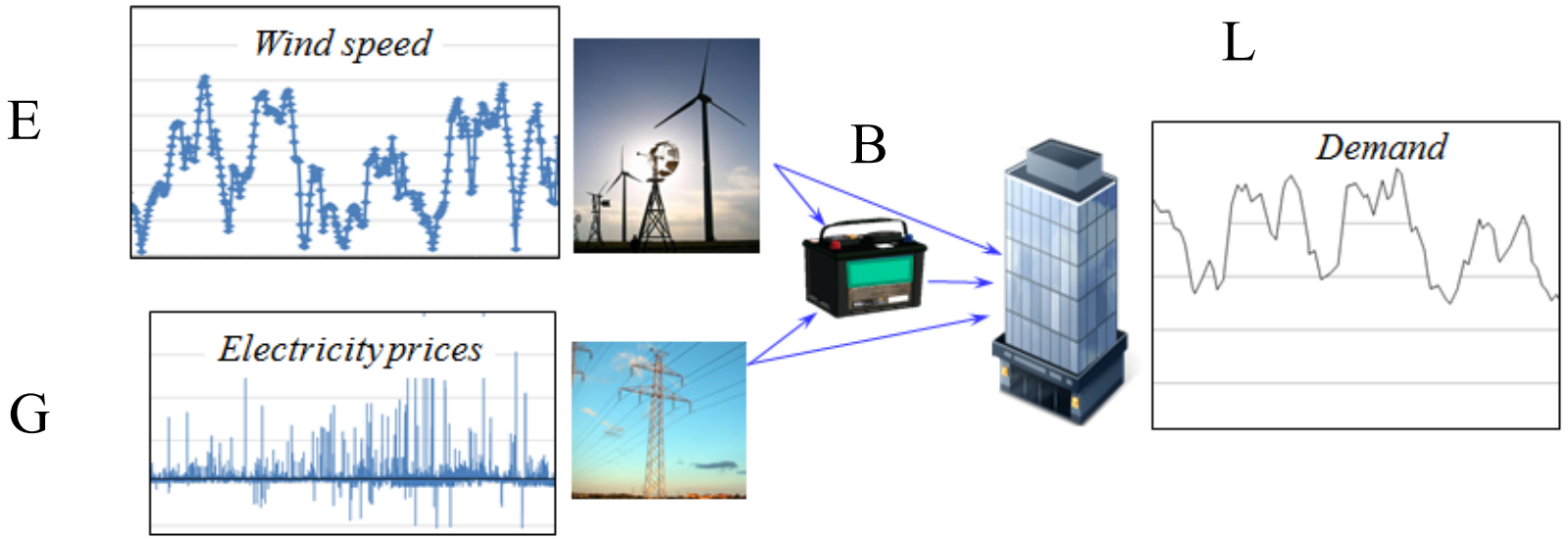
$$x_t = \left(x_t^{EL}, x_t^{EB}, x_t^{GL}, x_t^{GB}, x_t^{BL}, \right)$$

» Constraints;

$$\begin{aligned} x_t^{EL} + x_t^{EB} &\leq E_t, \\ (x_t^{GL} + x_t^{EL} + x_t^{BL}) &= L_t, \\ x_t^{BL} &\leq R_t, \end{aligned}$$

An energy storage problem

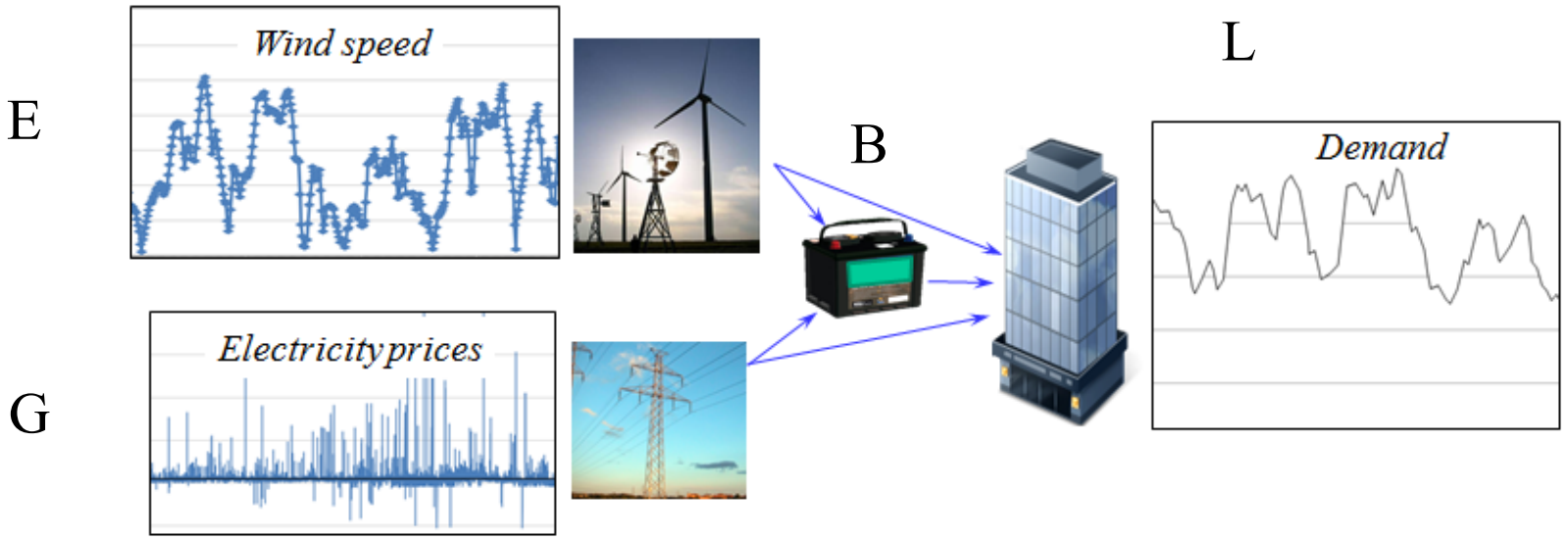
- Exogenous information



$$W_t = \begin{cases} \hat{E}_t & = \text{Change in energy from wind between } t-1 \text{ and } t \\ \varepsilon_t^p & = \text{Noise in the price process between } t-1 \text{ and } t \\ \varepsilon_t^D & = \text{Difference between actual demand and forecast} \end{cases}$$

An energy storage problem

- Transition function



$$E_{t+1} = E_t + \hat{E}_{t+1}$$

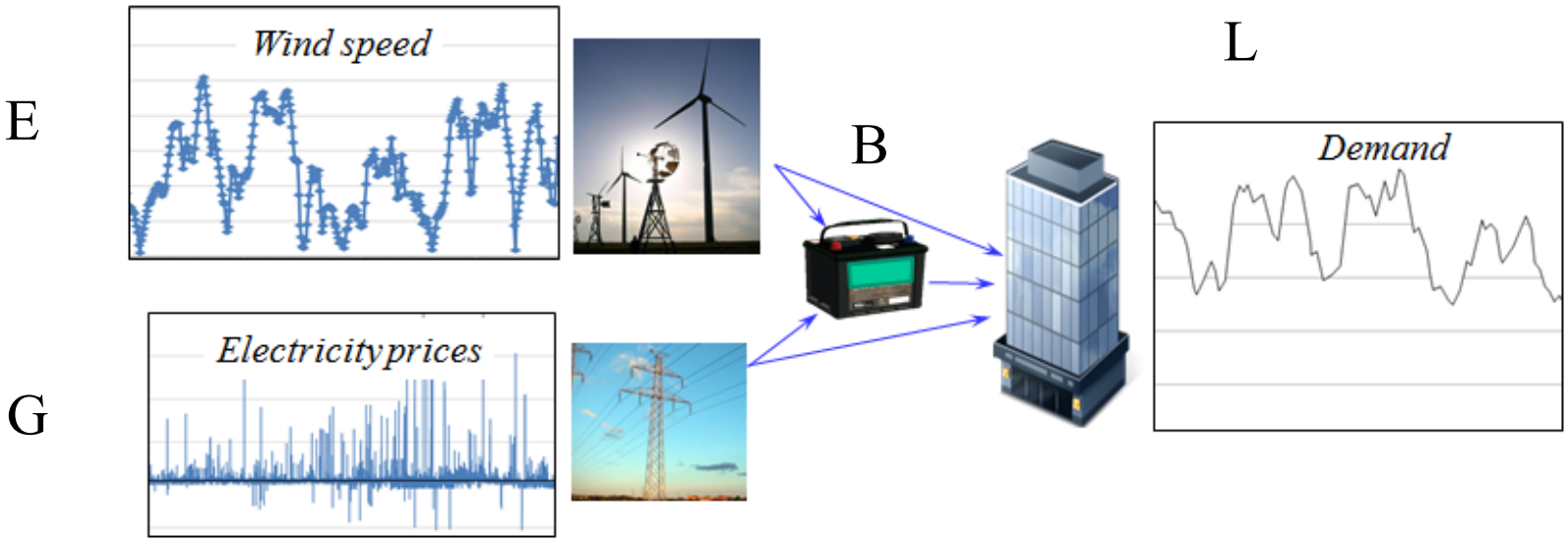
$$p_{t+1} = \theta_0 p_t + \theta_1 p_{t-1} + \theta_2 p_{t-2} + \varepsilon^p_{t+1} = (\theta)^T \bar{p}_t + \varepsilon^p_{t+1} \qquad \bar{p}_t = \begin{pmatrix} p_t \\ p_{t-1} \\ p_{t-2} \end{pmatrix}$$

$$D_{t+1} = D_t + \varepsilon^D_{t+1}$$

$$R^{battery}_{t+1} = R^{battery}_t + x_t$$

An energy storage problem

- Objective function



$$C(S_t, x_t) = p_t \left(x_t^{GB} + x_t^{GL} \right)$$

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

An energy storage problem

- State variables
  - » Cost function

    p_t = Price of electricity

  - » Decision function

    Constraints:

$$S_t = \left(\underbrace{E_t, L_t, R_t}, \underbrace{(p_t, p_{t-1}, p_{t-2})} \right)$$

$$\begin{aligned} x_t^{EL} + x_t^{EB} &\leq E_t \\ (x_t^{GL} + x_t^{EL} + x_t^{BL}) &= L_t \\ x_t^{BL} &\leq R_t \end{aligned}$$

  - » Transition function

$$p_{t+1} = \theta_0 p_t + \theta_1 p_{t-1} + \theta_2 p_{t-2} + \varepsilon_{t+1}^p$$

Basic energy modeling problem

Passive learning of price model

An energy storage problem

- Types of learning:
  - » No learning ($\theta's$ are known)

$$p_{t+1} = \theta_0 p_t + \theta_1 p_{t-1} + \theta_2 p_{t-2} + \varepsilon^p_{t+1}$$

  - » Passive learning (learn θs from price data)

$$p_{t+1} = \bar{\theta}_{t0} p_t + \bar{\theta}_{t1} p_{t-1} + \bar{\theta}_{t2} p_{t-2} + \varepsilon^p_{t+1}$$

Learning in stochastic optimization

- Updating the demand parameter
  - » Let p_{t+1} be the new price and let

$$\bar{F}_t^{price}(\bar{p}_t \mid \bar{\theta}_t) = (\bar{\theta}_t)^T \bar{p}_t = \bar{\theta}_{t0} p_t + \bar{\theta}_{t1} p_{t-1} + \bar{\theta}_{t2} p_{t-2}$$

  - » We update our estimate $\bar{\theta}_t$ using our recursive least squares equations:

$$\bar{\theta}_{t+1} = \bar{\theta}_t - \frac{1}{\gamma_{t+1}} B_t \bar{p}_t \varepsilon_{t+1}$$

$$\varepsilon_{t+1} = \bar{F}_t^{price}(\bar{p}_t \mid \bar{\theta}_t) - p_{t+1},$$

$$B_{t+1} = B_t - \frac{1}{\gamma_{t+1}} \left(B_t \bar{p}_t (\bar{p}_t)^T B_t \right)$$

$$\gamma_{t+1} = 1 + (\bar{p}_t)^T B_t \bar{p}_t$$

An energy storage problem

- State variables
  - » Cost function

    p_t = Price of electricity

  - » Decision function

    Constraints:

$$S_t = \left(\underbrace{E_t, L_t, R_t}, \underbrace{(p_t, p_{t-1}, p_{t-2})}, \underbrace{(\bar{\theta}_t, B_t)} \right)$$

$$\begin{aligned} x_t^{EL} + x_t^{EB} &\leq E_t \\ (x_t^{GL} + x_t^{EL} + x_t^{BL}) &= L_t \\ x_t^{BL} &\leq R_t \end{aligned}$$

  - » Transition function

$$p_{t+1} = \bar{\theta}_{t0} p_t + \bar{\theta}_{t1} p_{t-1} + \bar{\theta}_{t2} p_{t-2} + \varepsilon_{t+1}^p$$

$$D_{t+1} = D_t + \varepsilon_{t+1}^D$$

$$\bar{\theta}_{t+1} = \bar{\theta}_t - \frac{1}{\gamma_{t+1}} B_t \phi(x_t) \varepsilon_{t+1}$$

Basic energy modeling problem

Active learning of price model

An energy storage problem

- Types of learning:
  - » No learning ($\theta's$ are known)

$$p_{t+1} = \theta_0 p_t + \theta_1 p_{t-1} + \theta_2 p_{t-2} + \varepsilon^p_{t+1}$$

  - » Passive learning (learn θs from price data)

$$p_{t+1} = \bar{\theta}_{t0} p_t + \bar{\theta}_{t1} p_{t-1} + \bar{\theta}_{t2} p_{t-2} + \varepsilon^p_{t+1}$$

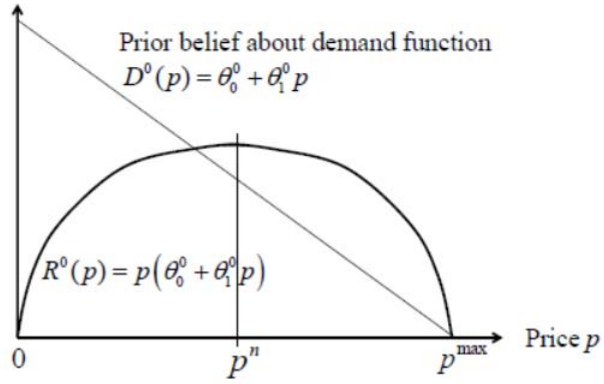
  - » Active learning ("bandit problems")

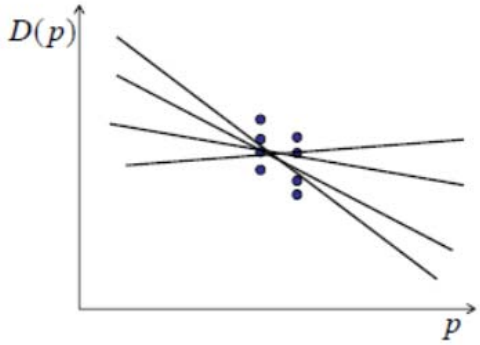
$$p_{t+1} = \bar{\theta}_{t0} p_t + \bar{\theta}_{t1} p_{t-1} + \bar{\theta}_{t2} p_{t-2} + \bar{\theta}_{t3} x^{GB}_t + \varepsilon^p_{t+1}$$

Buy/sell decisions

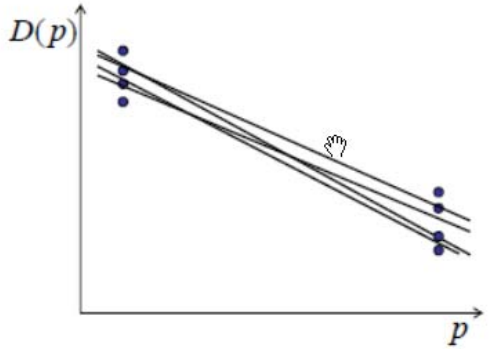
Linear belief models

- Dynamic pricing





Sampling behavior focusing on exploitation



Sampling behavior focusing on explortation

» Need to balance exploration and exploitation.

- Notes:
  - The addition of an active learning component does not change the basic model, but it changes the policies that will work best.
  - The ability of decisions to affect what we learn makes it possible to encourage more variability in the data.
  - If we are doing online learning (maximizing cumulative reward), we have to balance the costs of trying choices that do not appear to be best against the potential future benefits.
  - This tradeoff depends on how long the future is.

- Policies
  - » PFA
    - Take what appears to be best, and add in a noise term.
  - » CFA
    - Use UCB policy to search over discretized set of actions
  - » VFA
    - Solve using dynamic programming, but it is hard to capture the belief state in the value function.
  - » DLA
    - Use a decision tree that can capture entire state variable.
  - » Hybrid:
    - VFA plus noise term
    - DLA to handle a forecast, with noise added to costs (or the optimal decisions) to encourage exploration.

Learning

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

  $$\bar{V}_t(S_t|\theta) \approx V_t(S_t).$$

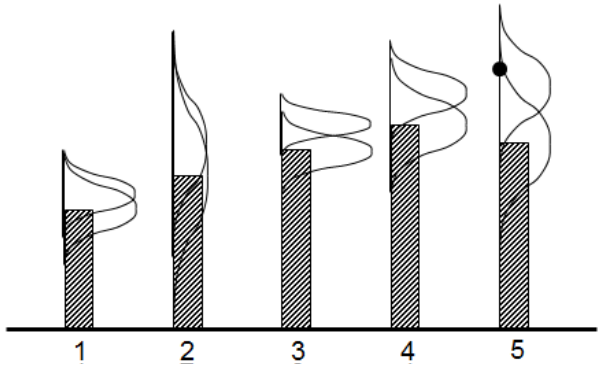
  4) Designing a cost function approximation:
     - The objective function $\bar{C}^{\pi}(S_t, x_t|\theta)$.
     - The constraints $X^{\pi}(S_t|\theta)$

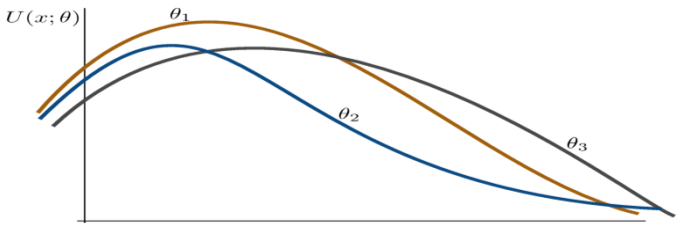
  5) Approximating the transition function

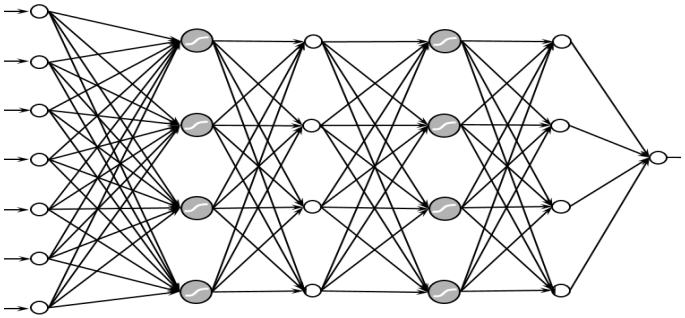
  $$\bar{S}^M(S_t, x_t, W_{t+1}|\theta) \approx S^M(S_t, x_t, W_{t+1})$$

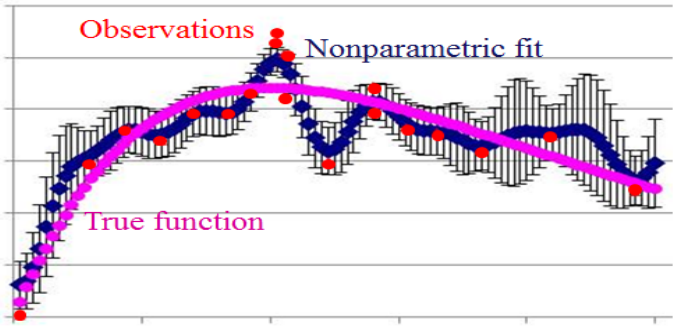
Approximation strategies

- 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





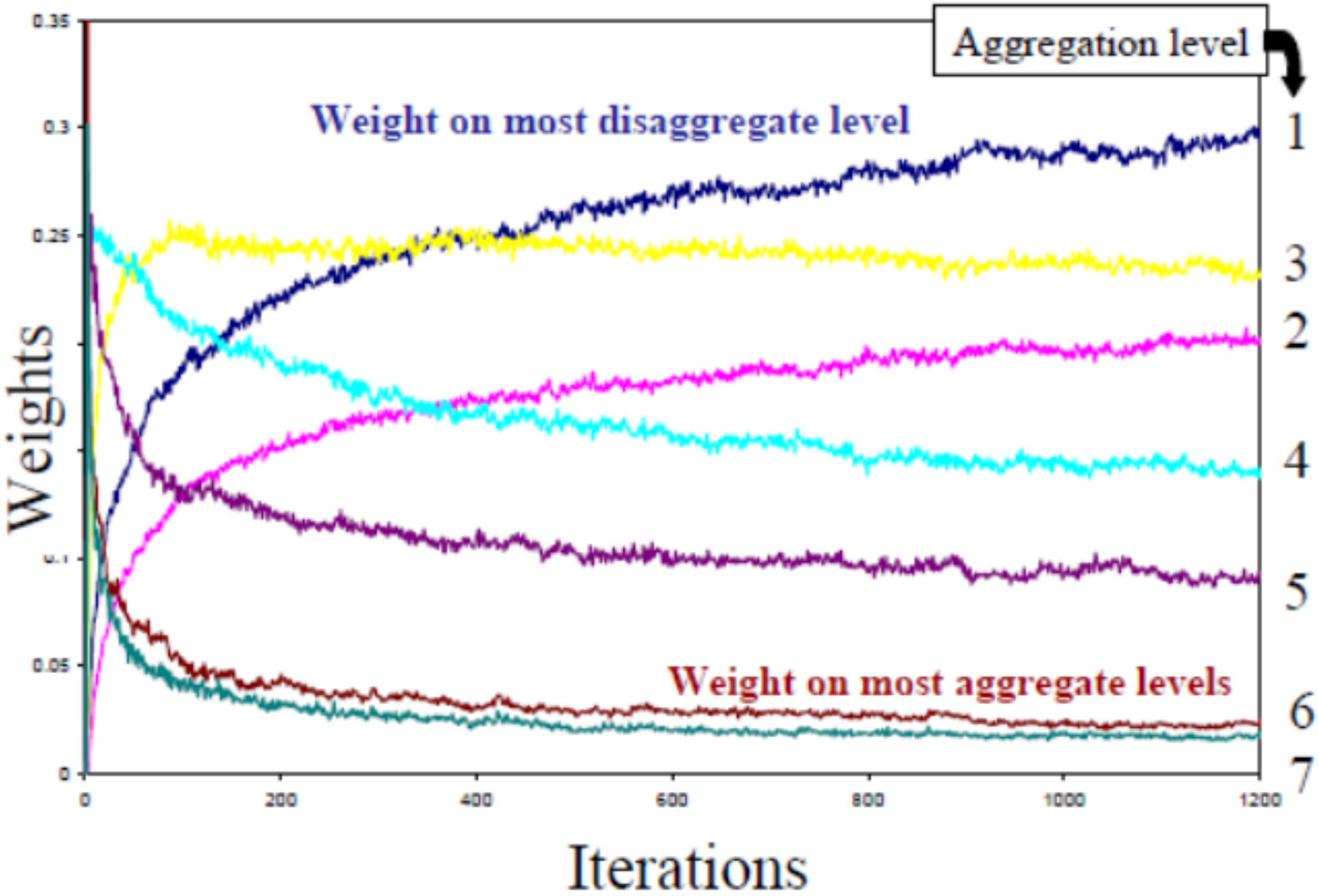




- Notes:
  - » In the context of adaptive search algorithms, we need to do online learning, which means starting with little or no data (perhaps just a belief) and transitioning to more data.
  - » This means starting with low-dimensional models and transitioning to higher dimensional models.

Learning challenges

- Variable-dimensional learning



Uncertainty modeling

- Notes:
  - » Think of "learning" as the problem of creating a point estimate.
  - » Uncertainty modeling can be viewed as the error in the point estimate.

Modeling uncertainty

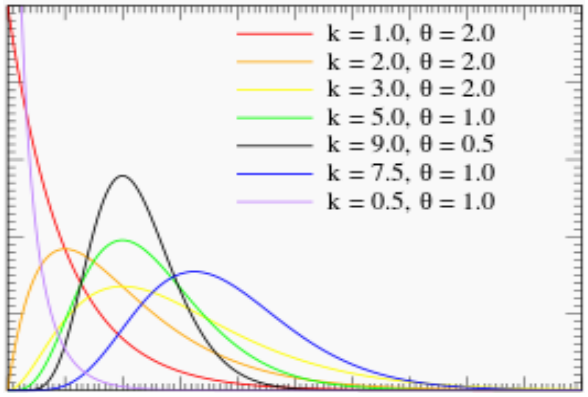
- Classes of uncertainty:
  - » Observational uncertainty
    - Noise in estimate of how many people carry a disease
    - Observation of how well an energy storage policy works
  - » Exogenous uncertainty
    - Changes in weather, prices, economy
  - » Prognostic uncertainty (forecasting)
    - Difference between actual and forecasted quantities
  - » Inferential uncertainty
    - Difference between an estimate of a parameter (e.g. demand response to price) and the actual.
  - » Experimental noise/variability
    - Variation between successive experiments
  - » Model uncertainty
    - Uncertainty in the structure of a model (e.g. transition function)
    - Uncertainty in the parameters of a model
    - Uncertainty in the parameters of a probability distribution (e.g. arrival rate)
    - Uncertainty in the objective function

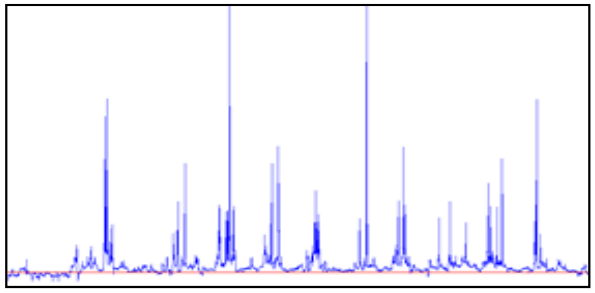
Modeling uncertainty

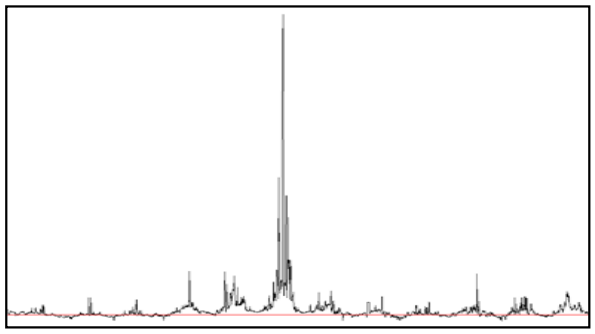
- Classes of uncertainty:
  - » Transitional uncertainty
    - Difference between the expected state that a control takes you to, and what actually happens (e.g. wind pushing a rocket)
  - » Control/implementation uncertainty
    - Difference between the decision you make and what gets implemented.
  - » Communication errors/biases
    - For multiagent systems, there can be noise introduced when communicating, along with biases.
  - » Algorithmic instability
    - Noise due to multiple optima, pre-mature stopping (epsilon-optimality)
    - Errors due to model truncation
  - » Goal uncertainty
    - Difference between the utility function we are using, and the objectives of a real system
  - » Political/regulatory uncertainty
    - Uncertainty in the regulatory environment

Distributions

- Types of distributions
  - » Exponential families
    - Normal, log-normal
    - Exponential
    - Gamma
    - ….
  - » Heavy-tailed distributions
    - Jump diffusion (mixed normals)
    - Cauchy
  - » Frequency:
    - Spikes
    - Bursts
    - Rare events

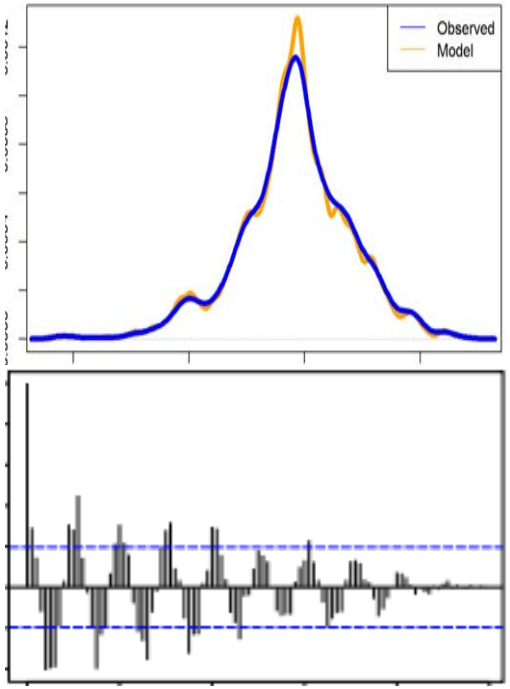


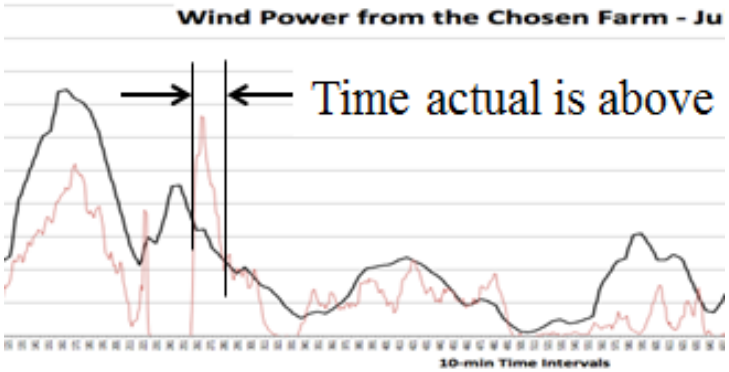




Distributions

- Statistical features
  - » Error distributions
    - Distribution of deviation from forecast.
  - » Autocorrelation functions (ACF)
    - Correlation between observations at different times.
  - » Crossing times
    - Distribution of time simulated is above or below the forecast.





Modeling uncertainty

- Fitting distributions
  - » Moment matching
    - E.g. use mean and variance from data to match the mean and variance of a distribution.
    - Beta distribution:

$$f(x|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

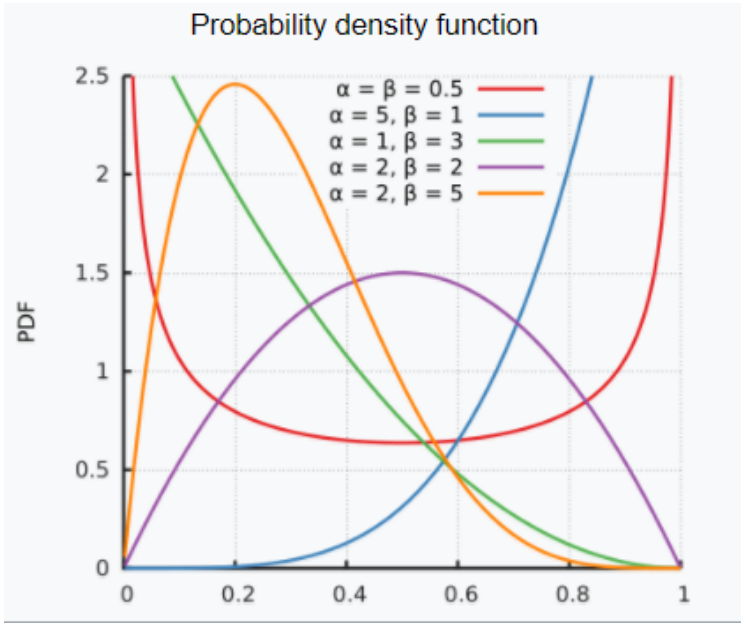
    - Mean:

$$E[X] = \frac{\alpha}{\alpha+\beta}$$

    - Variance

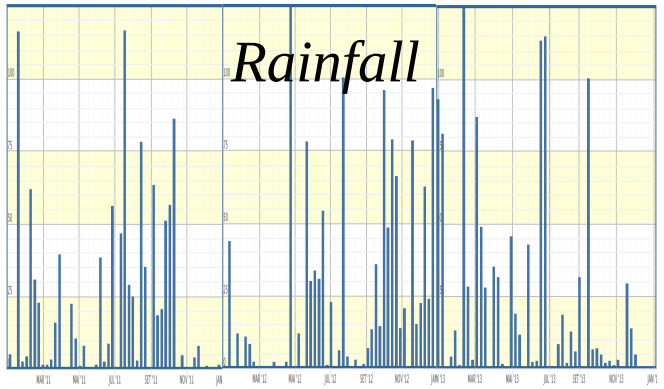
$$Var[X] = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$$

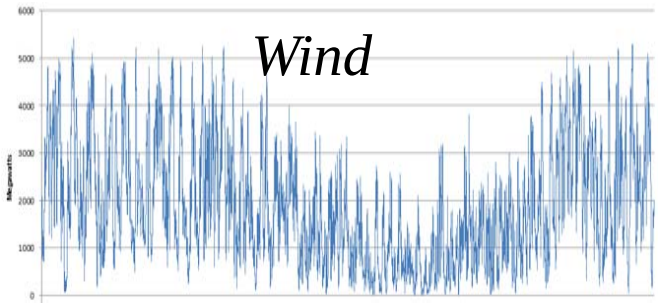
    - Use these formulas to find α and β



Distributions

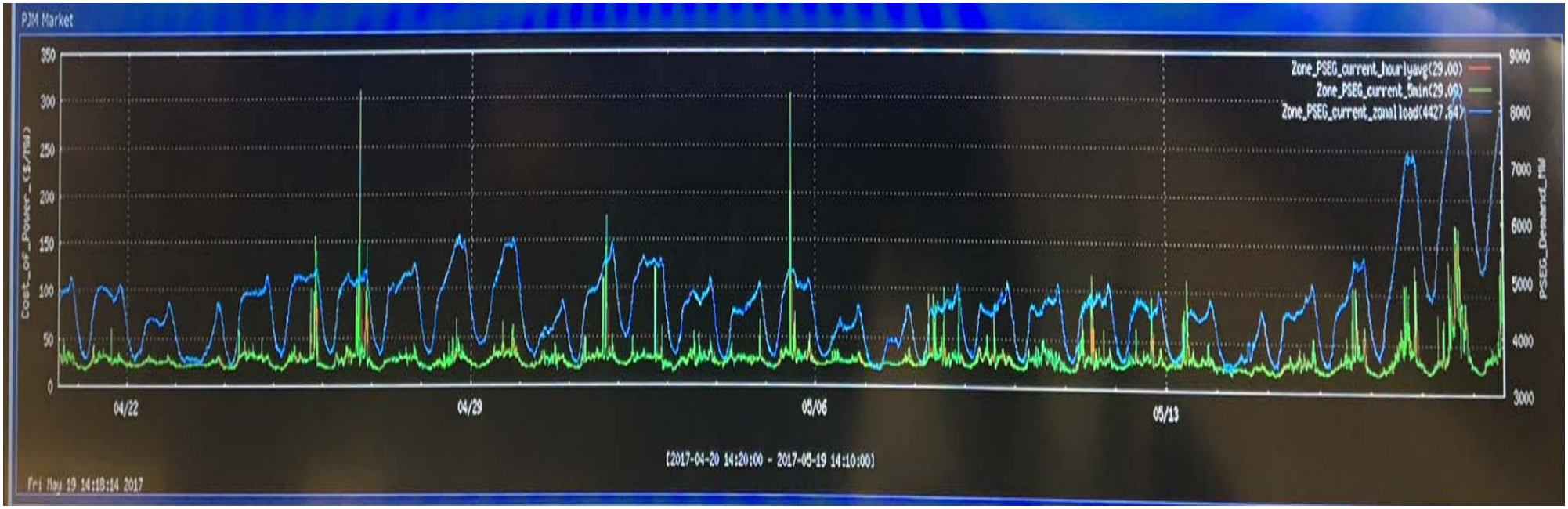
- Sampling from history
  - » We can create a probability distribution of the randomness of stochastic processes (e.g. wind, solar) by simply using samples drawn from different periods in history.
  - » Sampling from history retains complex intertemporal relationships.





Modeling uncertainty

- Some data can be very heavy tailed
  - » Snapshot of electricity prices for New Jersey:

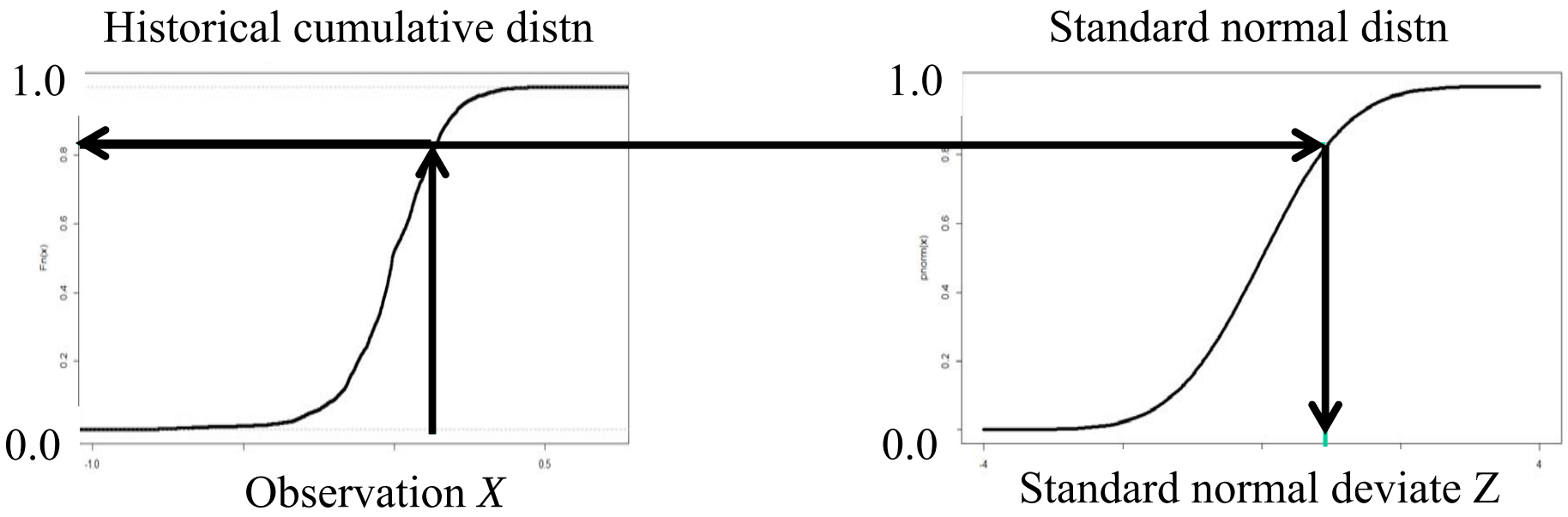


Modeling uncertainty

- Transforming distributions
  - » There are many problems where the distribution may be hard to fit. It can be useful to transform observations into normally distributed random variables, model these transformed variables, and then transform them back.
  - » Let U be a random variable that is uniformly distributed over [0,1], and let X be an arbitrary random variable with cumulative distribution $F_X(x) = Prob[X \leq x]$. Let $F_X^{-1}(u)$ be the inverse of the cumulative distribution. It is possible to show that
    - $F_X(X) \sim U$ and $F_X^{-1}(U) \sim X$

Modeling uncertainty

- Transforming distributions
  - » Observed errors are transformed to normally distributed errors using quantile mapping:



  - » Very useful for representing unusual distributions (e.g. heavy tailed, bimodal, …)

Policies

- Policies:
  - » Review four classes
  - » Policy search class:
    - just requires that you simulate; complex state variables are not an issue.
    - With online learning, you do not even need a model – you just need the ability to fix parameters and observe a noisy estimate of how well the policy is working.
    - S imulators are nice to have, but can be a lot of work to build. And ultimately, they are never perfect.
  - » Lookahead class:
    - Need to have a model, and then need to approximate the future:
      - – Either through a lookahead model
      - – … or through a value function approximation.

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

- Two fundamental strategies:

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

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

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

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

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:
  - » An optimal policy (based on looking ahead):

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

2a) Approximating the value of being in a downstream state using machine learning ("value function approximations")

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

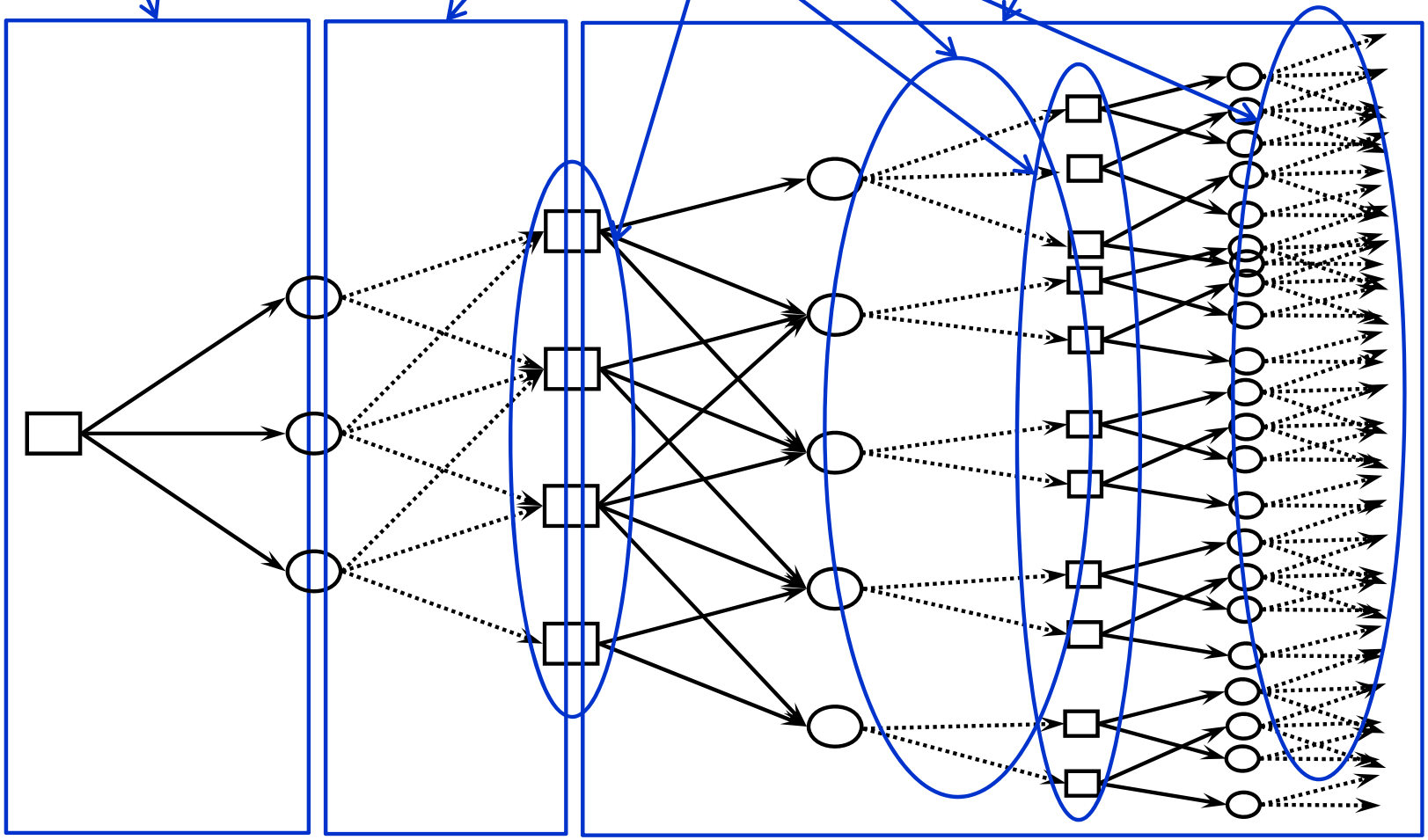
$$X_t^{VFA}(S_t) = \arg\max_{x_t} \left(C(S_t, x_t) + \mathbb{E}\left\{ \bar{V}_{t+1}(S_{t+1}) \mid S_t, x_t \right\} \right)$$

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

Designing policies

- The ultimate lookahead policy is optimal

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



Designing policies

- The ultimate lookahead policy is optimal

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

» 2b) Instead, we have to solve an approximation called the *lookahead model:*

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

» *A lookahead policy* works by approximating the *lookahead model*.

Lookahead policies

- Lookahead models use five classes of approximations:
  - » Horizon truncation – Replacing a longer horizon problem with a shorter horizon
  - » Stage aggregation – Replacing multistage problems with two-stage approximation.
  - » Outcome aggregation/sampling – Simplifying the exogenous information process
  - » Discretization – Of time, states and decisions
  - » Dimensionality reduction – We may ignore some variables (such as forecasts) in the lookahead model that we capture in the base model (these become *latent* variables in the lookahead model).

- Notes:
  - » The imbedded optimization over policies can take a variety of forms:
    - Lookup table (as is done with a decision tree).
    - Parameterized policy
      - Parameters may depend on the state $\tilde{S}_{t,t+1}$ (in theory, it should).
      - We may fix the parameter and determine it as a parameter of the overall policy $X^{\pi}(S_t|\theta)$ in the usual way.

Designing policies

- Types of lookahead approximations
  - » One-step lookahead – Widely used in pure learning policies:
    - Bayes greedy/naïve Bayes
    - Expected improvement
    - Value of information (knowledge gradient)
  - » Multi-step lookahead
    - Deterministic lookahead, also known as model predictive control, rolling horizon procedure
    - Stochastic lookahead:
      - Two-stage (widely used in stochastic linear programming)
      - Multistage
        - » Monte carlo tree search (MCTS) for discrete action spaces
        - » Multistage scenario trees (stochastic linear programming) – typically not tractable.

Four (meta)classes of policies

Policy search

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

» Lookup tables, rules, parametric/nonparametric functions

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

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

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*

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

» *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"*

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

Designing policies

- Finding the best policy
  - » We have to first articulate our classes of policies

$$f \in \mathcal{F} = \{PFAs, CFAs, VFAs, DLAs\}$$

$$\theta \in \Theta^f = \text{Parameters that characterize each family.}$$

  - » So minimizing over $\pi \in \Pi$ means:

$$\Pi = \left\{ f \in \mathcal{F}, \theta \in \Theta^f \right\}$$

  - » We then have to pick an objective such as

$$\max_\pi \mathbb{E} \sum_{t=0}^{T} C\left(S_t, X^\pi(S_t \mid \theta)\right) = \mathbb{E} \sum_{t=0}^{T} F\left(X^\pi(S_t \mid \theta), W_{t+1}\right)$$

or

$$\max_\pi \mathbb{E} C\left(S_T, X_T^\pi\right) = \mathbb{E} F(X_T^\pi, W)$$

Applications

… and policies

Topics

- Asset selling
- Adaptive market planning
- Diabetes problem
- Static shortest path
- Dynamic shortest path
- Energy storage problem (all four variations)
- Newsvendor problem (single/two agent, beer game)
- Blood management problem
- Clinical trials

- PFA for asset selling (sell on drops, surges)
- PFAs for stepsize policies
- CFA (UCB, IE) for diabetes)
- VFA for shortest paths
- DLA/Knowledge gradient for diabetes
- Dynamic shortest paths
- Parameterized stochastic shortest path
- MDP for energy storage
- Lookahead for energy storage
- PFA for newsvendor problem
- CFA for blood management I – parameterized myopic
- VFA for blood management II – VFAs
- Lookaheads for clinical trials
- All parameter search problems

- Asset selling
  - » Sell signal:

$$X^{sell-low}(S_t|\theta^{low}) = \begin{cases} 1 & \text{If } p_t < \theta^{low} \text{ and } R_t = 1 \\ 1 & \text{If } t = T \text{ and } R_t = 1 \\ 0 & \text{Otherwise} \end{cases} \quad (2.1)$$

  - » or

$$X^{high-low}(S_t|\theta^{high-low}) = \begin{cases} 1 & \text{If } p_t < \theta^{low} \text{ or } p_t > \theta^{high} \\ 1 & \text{If } t = T \text{ and } R_t = 1 \\ 0 & \text{Otherwise} \end{cases} \quad (2.14)$$

  - » or

$$X^{track}(S_t|\theta^{track}) = \begin{cases} 1 & \text{If } p_t \geq \bar{p}_t + \theta^{track} \\ 1 & \text{If } t = T \text{ and } R_t = 1 \\ 0 & \text{Otherwise} \end{cases} \quad (2.16)$$

$$\bar{p}_t = (1-\alpha)\bar{p}_{t-1} + \alpha\hat{p}_t. \quad (2.15)$$

Adaptive market planning

- Stepsize policies
  - » Harmonic stepsize formula (deterministic

$$\alpha_n(\theta^{step}) = \frac{\theta^{step}}{\theta^{step} + n - 1}, \tag{3.8}$$

  - » Kesten's rule (stochastic)

$$\alpha_n(\theta^{step}) = \frac{\theta^{step}}{\theta^{step} + K^n - 1}, \tag{3.9}$$

$$K^n = \begin{cases} K^n + 1 & \text{If } (\nabla^x F(x^n, W^{n+1}))^T \nabla^x F(x^{n-1}, W^n) < 0, \\ K^n & \text{Otherwise} \end{cases} \tag{3.10}$$

  - » Discuss:
    - Scaling
    - Handling bias (learning) vs. noise (smoothing)

- Choosing diabetes medication
  - » Upper confidence bounding policies

$$X^{UCB}(S^n|\theta^{UCB}) = \arg\max_{x\in\mathcal{X}} \left(\bar{\mu}_x^n + \theta^{UCB}\sqrt{\frac{\log n}{N_x^n}}\right), \quad (4.5)$$

  - » Interval estimation

$$X^{IE}(S^n|\theta^{IE}) = \arg\max_{x\in\mathcal{X}} \left(\bar{\mu}_x^n + \theta^{IE}\bar{\sigma}_x^n\right), \quad (4.6)$$

  - » Knowledge gradient

$$\nu_x^{KG,n} = \mathbb{E}_\mu \mathbb{E}_{W|\mu} \left\{ \max_{x'} \bar{\mu}_{x'}^{n+1}(x) \,|\, S^n \right\} - \max_{x'} \bar{\mu}_{x'}^n$$

- Shortest paths – static/stationary
  - » Deterministic, or stochastic (but we see costs after we choose the link):

$$X^{\pi}(i) = \arg\min_{j \in \mathcal{N}_i^+} (\bar{c}_{ij} + v_j). \quad (5.9)$$

  - » Stochastic (we see costs before we make a decision)

$$X^{\pi}(S_t) = argmax_j \, (\hat{c}_{ij} + \bar{V}_{tj}^{x,n-1})$$

- Imagine that the lookahead is just a black box:
  - » Solve the optimization problem

$$X_t^{\pi}(S_t^n) = \arg\min \sum_{i \in N} \sum_{j \in N_i^+} \tilde{c}_{tij}^n \tilde{x}_{tij}$$

  - » subject to

$$\sum_j \tilde{x}_{i_t^n, j} = 1 \quad \text{Flow out of current node where we are located}$$

$$\sum_i \tilde{x}_{i,r} = 1 \quad \text{Flow into destination node } r$$

$$\sum_i \tilde{x}_{i,j} - \sum_k \tilde{x}_{j,k} = 0 \quad \text{for all other nodes.}$$

  - » This is a deterministic shortest path problem that we could solve using Bellman's equation, but for now we will just view it as a black box optimization problem.

- The θ −percentile policy.
  - » Solve the linear program (shortest path problem):

$$X_t^\pi(S_t^n \mid \theta) = \arg\min \sum_{i \in N} \sum_{j \in N_i^+} \tilde{c}_{tij}^{\,p}(\theta)\tilde{x}_{tij} \quad \text{(Vector with } x_{tij} = 1 \text{ if decision is to take } (i, j)\text{)}$$

  - » subject to

$$\sum_j \tilde{x}_{t,i_t^n,j} = 1 \quad \text{Flow out of current node where we are located}$$

$$\sum_i \tilde{x}_{tir} = 1 \quad \text{Flow into destination node } r$$

$$\sum_i \tilde{x}_{tij} - \sum_k \tilde{x}_{tjk} = 0 \quad \text{for all other nodes.}$$

  - » This is a deterministic shortest path problem that we could solve using Bellman's equation, but for now we will just view it as a black box optimization problem.

- Energy storage problem
  - » PFA version

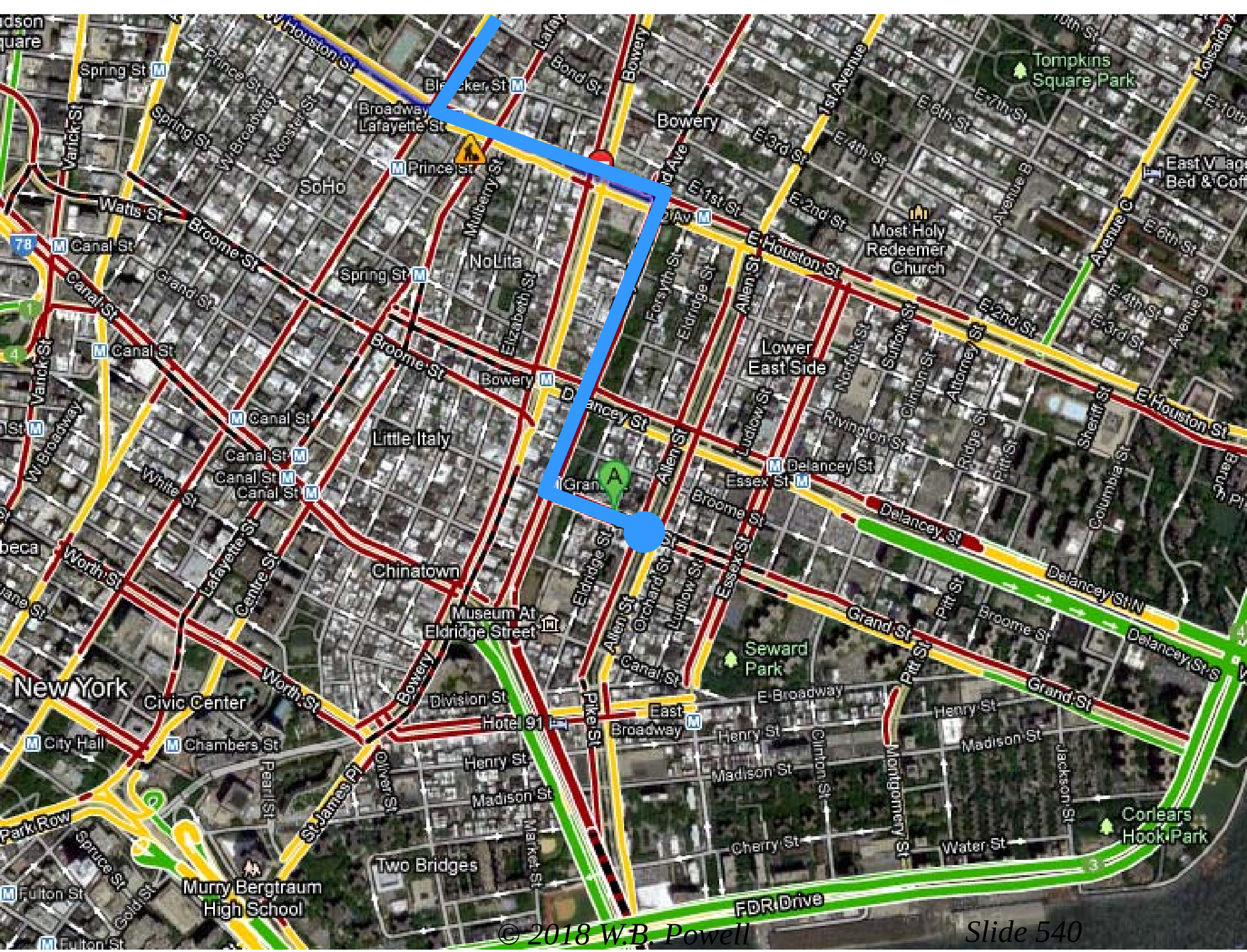
$$X^{\pi}(S_t \mid \theta) = \begin{cases} +1 & \text{if } p_t < \theta^{\text{charge}} \\ 0 & \text{if } \theta^{\text{charge}} < p_t < \theta^{\text{discharge}} \\ -1 & \text{if } p_t > \theta^{\text{charge}} \end{cases}$$

- Use basic policy search:

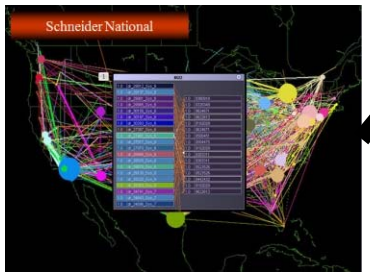
$$\max_{\theta} F(\theta) = \mathbb{E}\sum_{t=0}^{T} \gamma^t C\left(S_t, X_t^{\pi}(S_t \mid \theta)\right)$$

Designing policies

Matching problems to policies



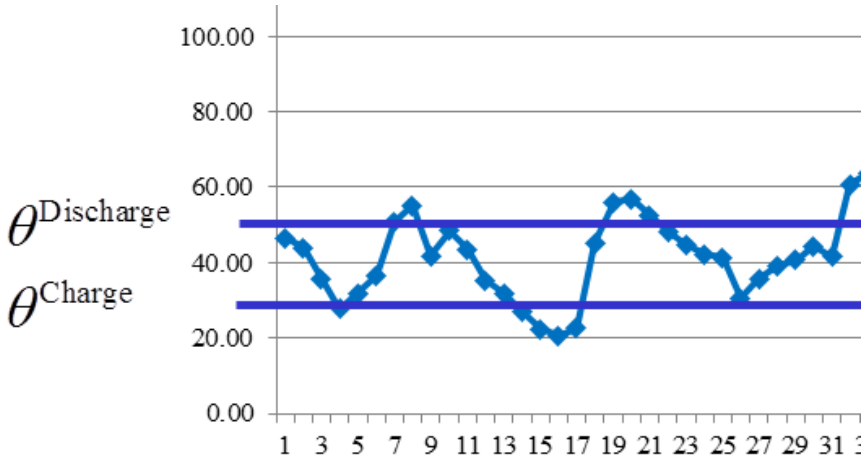
Choosing a policy

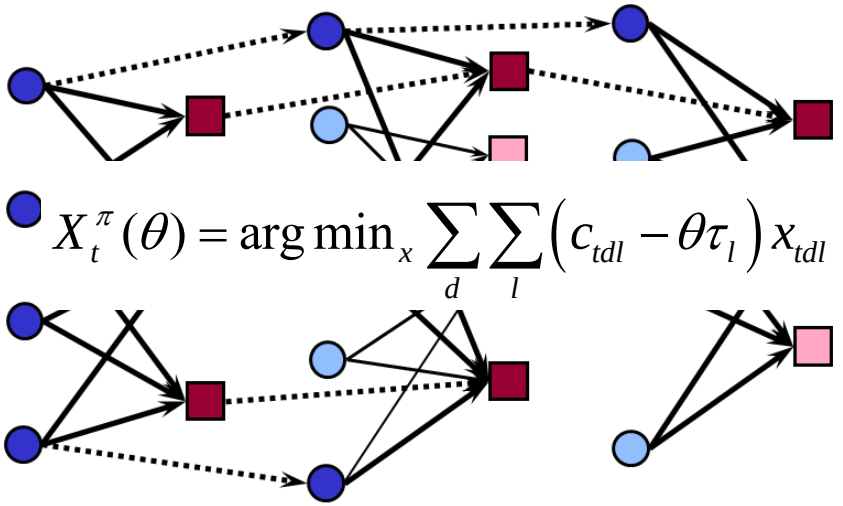


- Robust cost function approximation
- Lookahead policy
- Policy function approximation
- Policy based on value function approximation

Choosing a policy

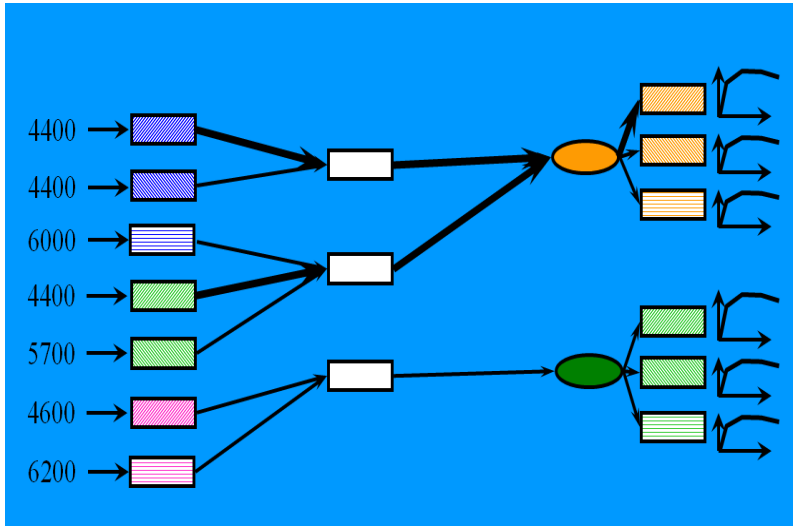
- Which policy to use?
  - » PFAs are best for low-dimensional problems where the structure of the policy is apparent from the problem.
  - » CFAs work for high-dimensional problems, where we can get desired behavior by manipulating the cost function.



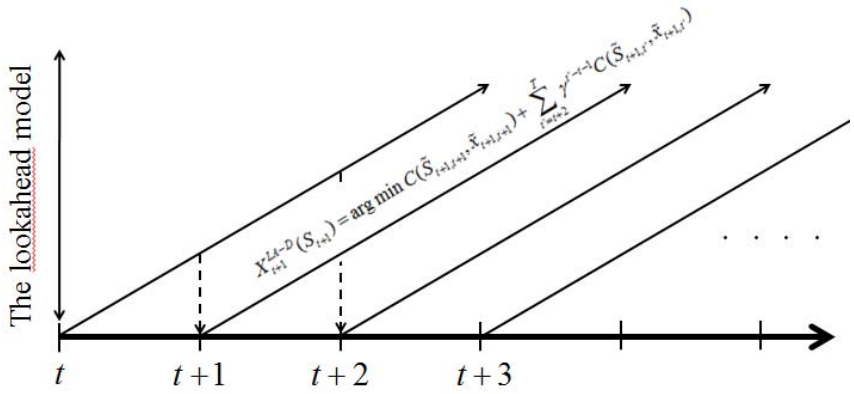


Choosing a policy

- Which policy to use?
  - » VFAs work best when the lookahead model is easy to approximate



  - » Lookahead models should be used only when all else fails (which is often)



Evaluating policies

- "Offline learning"
  - » Evaluate policies in a simulator
  - » Requires building a simulator
  - » Much faster than online, but you have to accept the errors implicit in any simulator.

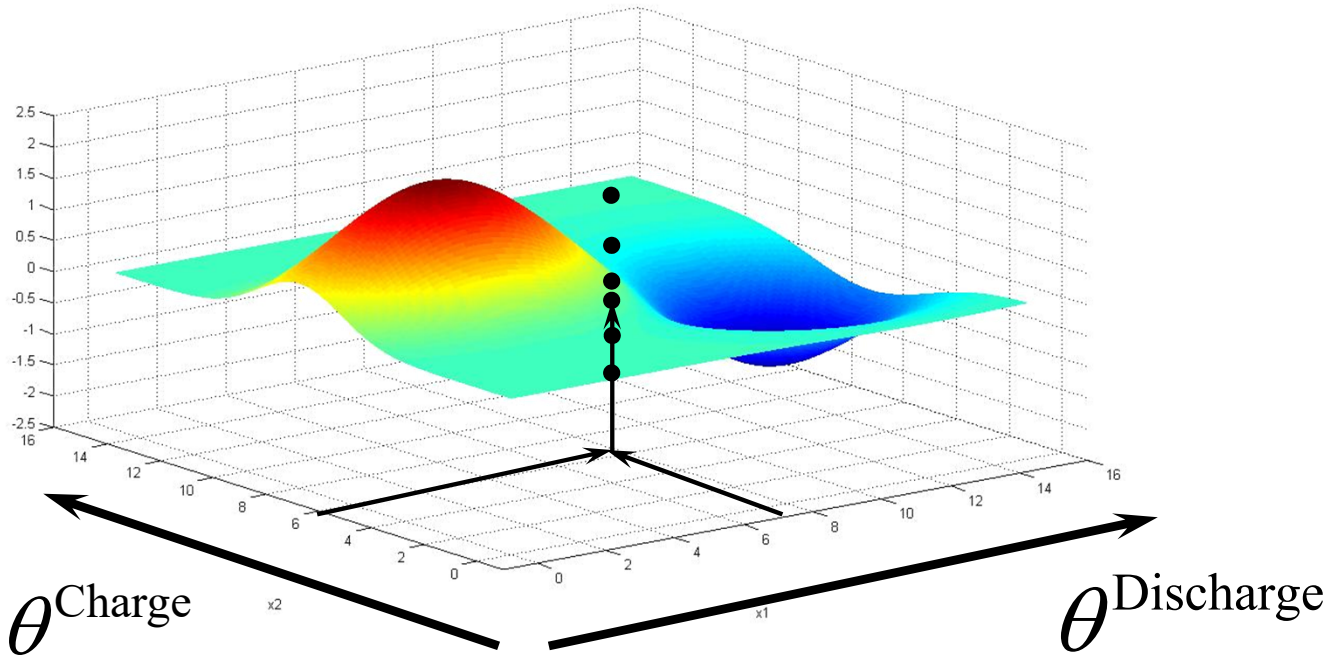
- "Online learning"
  - » Learning in the field
  - » Does not require a simulator…
  - » … but it is very slow (takes a day to simulate a day).

Policy search class

- Finding the best policy
  - » We need to maximize

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

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



- There are two settings for doing policy search:
  - » Offline
    - In a computer simulation
    - In a lab where we do not mind making mistakes
    - We look to optimize the performance of the final design after a series of experiments
  - » Online
    - This means in the field, where we care about how well we do while we are learning.
    - We look to optimize performance along the way, which means we are maximizing cumulative reward.
  - » Caution: "offline" and "online" are terms used in the machine learning community:
    - "offline" means batch learning
    - "online" means continuous updating

- Offline ("final reward") objective
  - » The final reward process works as follows:
    - States, decisions and exogenous information evolve according to

$$(S^0, x^0, W^1, S^1, x^1, W^2, \dots S^n, x^n, W^{n+1}, \dots)$$

    - Decisions are made according to a policy $x^n = X^\pi(S^n)$.
    - States are updated using

$$S^{n+1} = S^M(S^n, x^n, W^{n+1})$$

    - After N experiments, we call our final decision $x^{\pi,N}$, since it depends on our policy π and the budget N.
    - Once we have the final design, we then have to stop and test the design using new observations that we call $\widehat{W}$.

- The value of the policy
  - » We can start by writing
$$F^{\pi} = \mathbb{E}F(x^{\pi,N}, \hat{W})$$
  - » … but this is not very clear.
  - » There are up to three different sources of uncertainty:
    - The initial state S^0, which might have a Bayesian prior about the performance of the different diabetes drugs.
    - The observations $W^1, W^2, \ldots, W^N$
    - The observations made during the final testing $\hat{W}$.
  - » We can write the value of a policy more clearly as
$$F^{\pi} = \mathbb{E}_{S^0}\mathbb{E}_{W^1,W^2,\ldots,W^N|S^0}\mathbb{E}_{\hat{W}|S^0}F(x^{\pi,N}, \hat{W})$$
  - » This is better, but we won't really understand it until we know how to compute it.

- The value of the policy (cont'd)
  - » To compute
    $$F^{\pi} = \mathbb{E}_{S^0}\mathbb{E}_{W^1,W^2,\ldots,W^N|S^0}\mathbb{E}_{\widehat{W}|S^0}F(x^{\pi,N},\widehat{W})$$
  - » We have to simulate the expectations. Let's break down each one:
  - » $\mathbb{E}_{S^0}$: Imagine that S^0 includes a Bayesian prior such as the belief about how a patient responds to a diabetes medication. Let $\mu = (\mu_x)_{x\in X}$ be the truth of the performance of each drug x. Let's just sample $k = 1,\ldots,K$ samples of the vector μ to get $(\mu^1,\mu^2,\ldots,\mu^k,\ldots,\mu^K)$ samples of what each of the truths might be, and let's let $p^0 = \left(p_k^0\right)_{k=1}^K$ be the prior probabilities on these samples (perhaps 1/K).

- The value of the policy (cont'd)
  - » To compute
    $$F^{\pi} = \mathbb{E}_{S^0}\mathbb{E}_{W^1,W^2,\ldots,W^N|S^0}\mathbb{E}_{\widehat{W}|S^0}F(x^{\pi,N},\widehat{W})$$
  - » $\mathbb{E}_{W^1,W^2,\ldots,W^N|S^0}$: Given S^0 means picking one of the $\mu^k = \left(\mu_x^k\right)_{x\in X}$ as the true truth for each drug. Now let's try drug $x = x^n$ after the nth trial, and then observe
    $$W_x^{n+1} = \mu_x^k + \varepsilon_x^{n+1}$$
    where ε_x^{n+1} is the noise in an observation.
  - » Let $W = (W^1, W^2, \ldots, W^N)$ be a sequence of realizations. Again assume we have a series of samples that we will call $W^{(1)}, W^{(2)}, \ldots W^{(\ell)}, \ldots, W^{(L)}$ a sample of all the observations (actually a sample of ε).

- The value of the policy (cont'd)
  - » To compute
    $$F^{\pi} = \mathbb{E}_{S^0}\mathbb{E}_{W^1,W^2,\ldots,W^N|S^0}\mathbb{E}_{\hat{W}|S^0}F(x^{\pi,N},\hat{W})$$
  - » $\mathbb{E}_{\hat{W}|S^0}$: At this point we have our final design $x^{\pi,N}$ which depends on the truth μ^k and the experimental outcomes $W^1, W^2, \ldots, W^L$. We write this as $x^{\pi,N}(k,l)$.
  - » Now we have to test it by observing a new outcome $\hat{W}$. Again assume we sample this, and observe $\hat{W}^1, \hat{W}^2, \ldots, \hat{W}^m, \ldots, \hat{W}^M$ outcomes.
  - » Now simulate F^{π} using
    $$\bar{F}^{\pi} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{L}\sum_{l=1}^{L}\sum_{m=1}^{M}F\,(x^{\pi,N}(k,l),\hat{W}^m)$$

- Online ("cumulative reward") objective
  - » This is how we evaluated our diabetes policy. Rather than evaluating at the end, we evaluate as we proceed, which makes the formula a bit simpler.
  - » The expected performance of a policy would be written

$$F^{\pi} = \mathbb{E}_{S^0}\mathbb{E}_{W^1,W^2,\ldots,W^N|S^0}\sum_{n=0}^{N-1} F(X^{\pi}(S^n), W_x^{n+1})$$

  - » Using samples to approximate the expectations is just as we did for the offline case. The only difference is that we sum our performance, and we do not have to separate the "learning" from the "evaluating."
  - » A more difficult issue is how we do learning in a field situation. This is active research!

- Tuning environments
  - » In a simulator
    - Requires building a mathematical model of the dynamic process and, most important, the uncertainties.
    - Your policy will only be as good as your model of uncertainty.
  - » In the real world
    - No longer need a stochastic model…
    - But it takes a day to simulate a day. This can be very slow.
    - Examples:
      - FAA
      - Grid operators

Week 12 – Monday

Ad-click problem

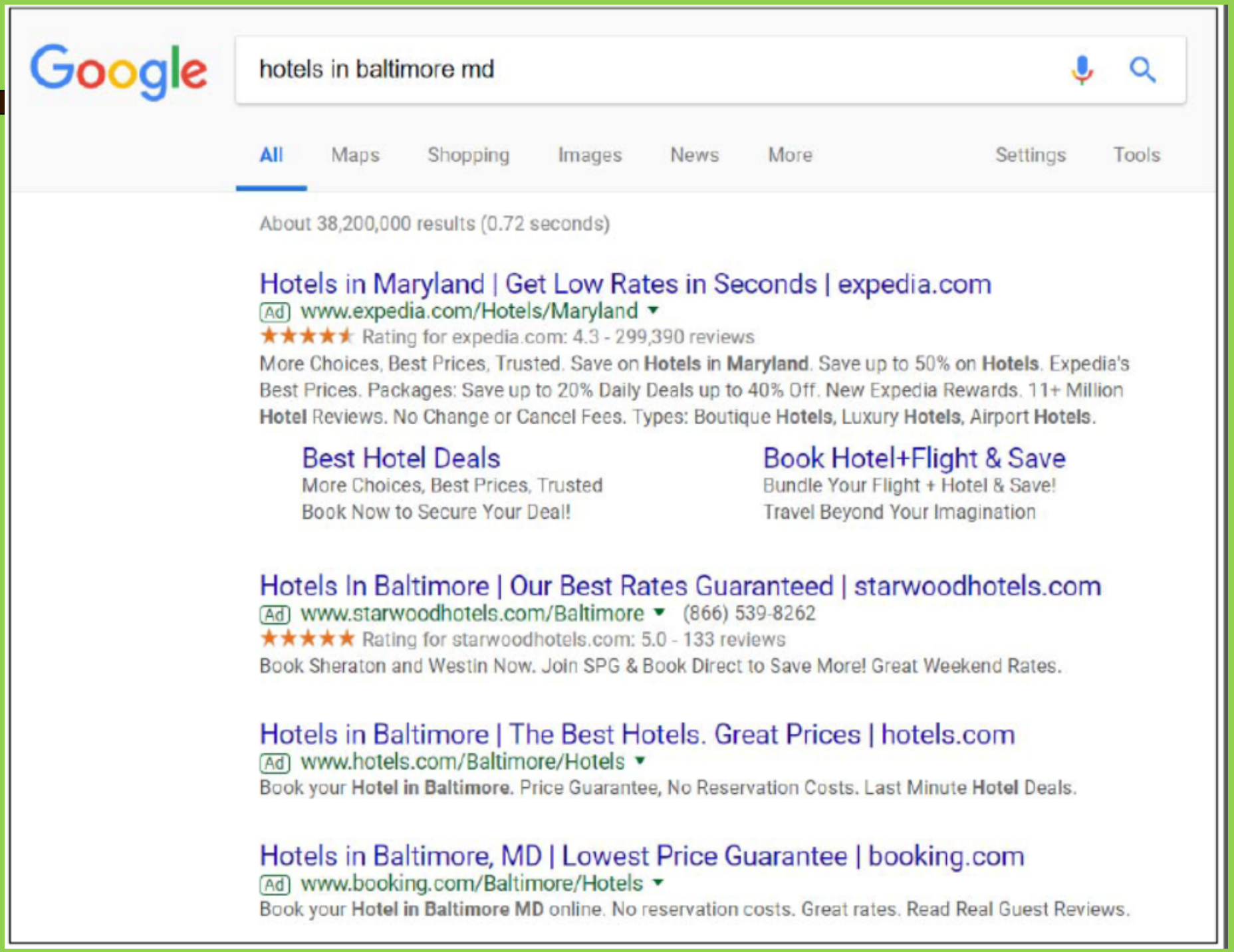
Narrative

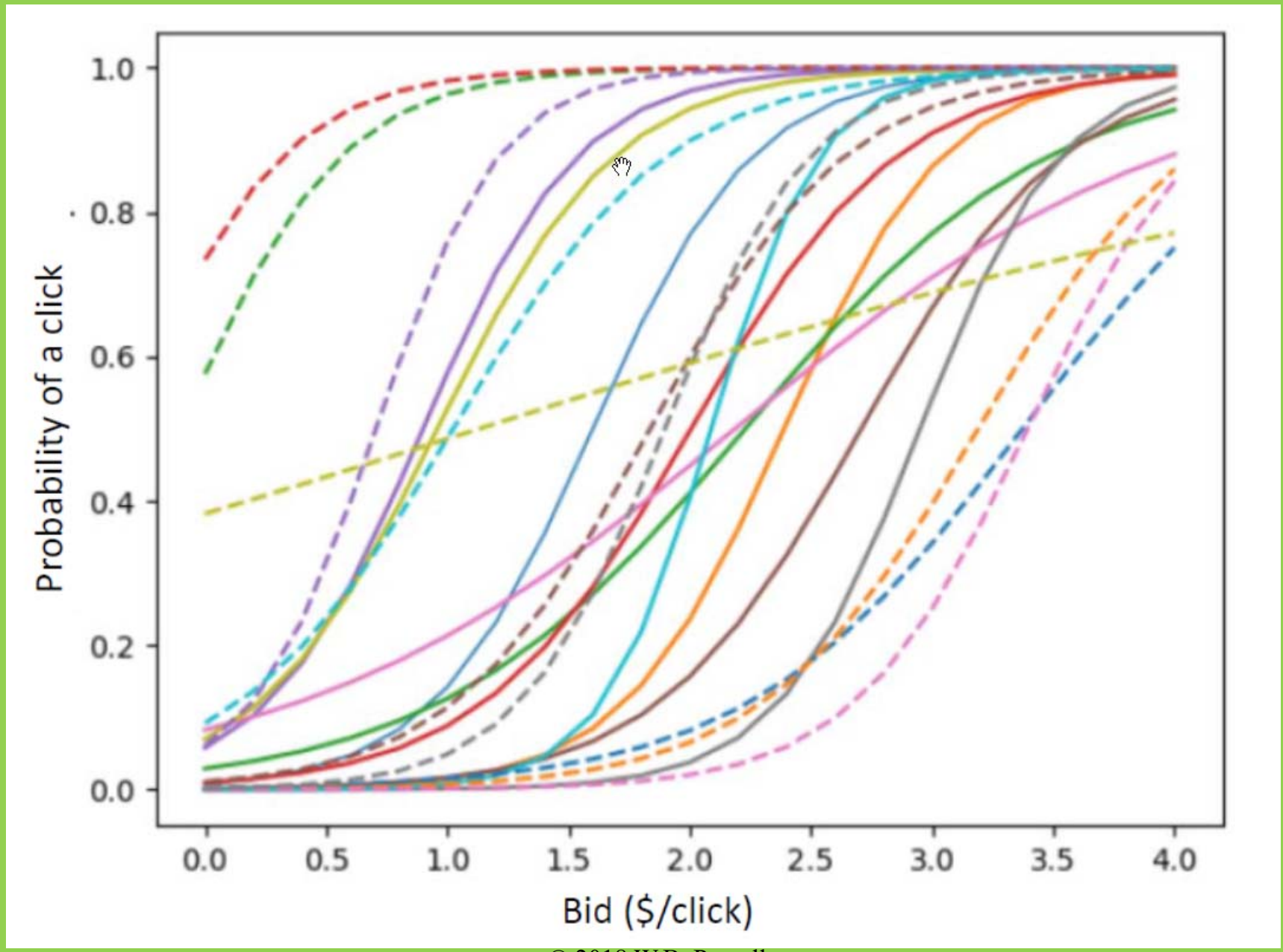
6.1 NARRATIVE

Companies advertising on internet sites such as Google have to bid to get their ads in a visible position (that is, at the top of the list of sponsored ads). When a customer enters a search term, Google identifies all the bidders who have listed the same (or similar) search terms in their list of ad-words. Google then takes all the matches, and sorts them in terms of how much each participant has bid, and runs an auction. The higher the bid, the more likely your ad will be placed near the top of the list of sponsored ads, which increases the probability of a click. Figure 6.1 is an example of what is produced after entering the search terms "hotels in baltimore md."

If a customer clicks on the ad, there is an expected return that reflects the average amount a customer spends when they visit the website of the company. The problem is that we do not know the bid response curve. Figure 6.2 reflects a family of possible response curves. Our challenge is to try out different bids to learn which curve is correct.

We will begin by assuming that we can adjust the bid after every auction, which means we only learn a single response (the customer did or did not click on the link). It is possible that the customer looked at a displayed link and decided not to click on it, or our bid may have been so low that we were not even in the list of displayed ads.





Basic model

6.2 BASIC MODEL

We are going to assume that we use some sort of parameterized model to capture the probability that a customer clicks on an ad. At a minimum this probability will depend on how much we bid for an ad (the more we bid, the higher the ad will appear in the sponsored ad list, which increases the likelihood that a customer will click on it). Let $K^n = 1$ if the nth customer clicks on the ad. Let

$$P^{click}(x|\theta) \quad = \quad Prob[K^{n+1} = 1|\theta = \theta_k]$$

where $P^{click}(x|\theta)$ is given by a logistics function. Let θ be the parameter vector for the logistics function. We do not know what θ is, but we are going to assume that it is one of a sampled set $\Theta = \{\theta_1, \ldots, \theta_K\}$.

State variables

We use the following variables: The initial state S_0 includes

$$\begin{aligned}\Theta &= \{\theta_1, \ldots, \theta_K\} \\ &= \text{The set of possible values that } \theta \text{ may take,} \\ \bar{R}^0 &= \text{Initial estimate of revenue earned when a customer clicks on a link.}\end{aligned}$$

The dynamic state variables S^n includes:

$$\begin{aligned} &= \text{The estimate of the parameter vector } \theta \text{ after } n \text{ auctions,} \\ p_k^n &= \text{Probability that the true } \theta = \theta_k, \\ p^n &= (p_k^n)_{k=1}^K, \\ \bar{R}^n &= \text{Estimate of revenue earned from an ad-click after } n \text{ auctions.}\end{aligned}$$

Our state variable, then, is

$$S^n = (\bar{R}^n, p^n).$$

Decision variables

Our only decision variable is the bid which we define as

$$x^n = \text{The bid (in \$ per click) for the } n+1st \text{ auction.}$$

As before, we let $X^\pi(S^n)$ our generic policy that gives us the bid x^n as a function of the information available to us, represented by S^n.

Exogenous information

In our initial model, we only observe the results of a single auction, which we model using

$$K^{n+1} = \begin{cases} 1 & \text{If the customer clicks on our ad,} \\ 0 & \text{Otherwise.} \end{cases}$$

$$\hat{R}^{n+1} = \text{Revenue earned from the } n+1st \text{ auction.}$$

This means our complete exogenous information vector is

$$W^{n+1} = (\hat{R}^{n+1}, K^{n+1}).$$

Transition function

We are going to update our estimated revenue when a customer clicks on the ad using

$$\bar{R}^{n+1} = \begin{cases} (1-\theta^{revenue})\bar{R}^n + \theta^{revenue}\hat{R}^{n+1} & \text{If } K^{n+1}=1, \\ \bar{R}^n & \text{Otherwise.} \end{cases} \tag{6.1}$$

Thus, we only update our estimated revenue when we get a click. The parameter $\theta^{revenue}$ is a smoothing parameter between 0 and 1 that we fix in advance.

We next address the updating of the probabilities p_k^n. We let H^n be the history of states, decisions and exogenous information

$$H^n = (S^0, x^0, W^1, S^1, x^1, \ldots, W^n, S^n, x^n)$$

use this to write

$$p_k^n = Prob[\theta = \theta_k | H^n].$$

We use Bayes theorem to write

$$\begin{aligned} p_k^{n+1} &= Prob[\theta = \theta_k | W^{n+1}, H^n] \\ &= \frac{Prob[K^{n+1} | \theta = \theta_k, H^n] Prob[\theta = \theta_k | H^n]}{Prob[K^{n+1} | H^n]}. \end{aligned} \tag{6.2}$$

Remember that the history H^n includes the decision x^n which (given a policy for making these decisions) is directly a function of the state S^n. We now use our logistic curve in equation (6.7) to write

$$Prob[K^{n+1} = 1|\theta = \theta_k, H^n] = Prob[K^{n+1} = 1|\theta = \theta_k, x^n] \quad (6.3)$$

$$= \frac{e^{\theta_k^{const} + \theta_k^{bid} x^n}}{1 + e^{\theta_k^{const} + \theta_k^{bid} x^n}}. \quad (6.4)$$

We then note that

$$Prob[\theta = \theta_k|H^n] = p_k^n. \quad (6.5)$$

Finally, we note that the denominator can be computed using

$$Prob[K^{n+1}|H^n] = \sum_{k=1}^{K} Prob[K^{n+1}|\theta = \theta_k, H^n]p_k^n. \quad (6.6)$$

Our use of a sampled representation of the possible outcomes of θ is saving us here. Even if θ has only two dimensions (as it is here, but only for now), performing a two-dimensional integral over a multivariate distribution for θ would be problematic.

Equations (6.4)- (6.6) allow us to calculate our Bayesian updating equation for the probabilities in (6.2). Equations (6.1)-(6.2) make up our transition function

$$S^{n+1} = S^M(S^n, x^n, W^{n+1}).$$

Objective function

We begin by writing the single period profit function as

$$C(S^n, x^n, W^{n+1}) \quad = \quad (\bar{R}^n - x^n)K^{n+1}.$$

Note that this is one of those instances where it is natural to compute the one-period contribution as a function of W^{n+1}. Our objective function can now be written in the canonical form

$$\max_{\pi} \mathbb{E}_{S^0} \mathbb{E}_{W^1,\ldots,W^n|S^0} \sum_{n=0}^{N} C(S^n, X^\pi(S^n), W^{n+1})$$

Uncertainty modeling

6.3 MODELING UNCERTAINTY

We are going to assume that the probability that a bid x^n produces a click K^{n+1} in the $n+1st$ auction is described by a logistics curve

$$P[K^{n+1}=1|x=x^n,\theta] = \frac{e^{\theta^{const}+\theta^{bid}x}}{1+e^{\theta^{const}+\theta^{bid}x}}. \quad (6.7)$$

Designing policies

6.4.1 Pure exploitation

The starting point of any online policy should be pure exploitation, which means doing the best that we can. The expected contribution from bidding x given what we know after n clicks (captured by S^n) is given by

Note that we have used

$$\begin{aligned}\mathbb{E}\hat{R}^{n+1}K^{n+1} &= \mathbb{E}\{\hat{R}^{n+1}|K^{n+1}=1\}Prob[K^{n+1}=1|\theta=\theta_k] \\ &= \bar{R}^n P^{click}(x|\theta).\end{aligned}$$

To find the best bid, we find (after a bit of algebra)

$$\begin{aligned}\frac{d\bar{C}(x)}{dx} &= (\bar{R}^n - x)\frac{dP^{click}(x|\theta)}{d\theta} - P^{click}(x|\theta) \\ &= 0,\end{aligned}$$

where

$$\frac{dP^{click}(x|\theta)}{dx} = \frac{\theta_1 e^{-\theta_0-\theta_1 x}}{(1+e^{-\theta_0-\theta_1 x})^2}$$

Figure 6.3 shows $\frac{d\bar{C}(x|\theta)}{dx}$ versus the bid x, showing the behavior that it starts positive and transitions to negative. The point where it is equal to zero would be the optimal bid, a point which can be found numerically quite easily. Let

$X^{explt}(S^n)$ = A pure exploitation policy which chooses the optimal bid to maximize the single-period profit using the belief distribution p^n for θ.

6.4.2 An excitation policy

A potential limitation of our pure exploitation policy is that it ignores the value of trying a wider range of bids to help with the process of learning the correct values of θ. A popular strategy is to add a noise term, known in engineering as "excitation," giving us the policy

$$X^{excite}(S^n|\rho) \quad = \quad X^{explt}(S^n) + \varepsilon(\rho)$$

where $\varepsilon(\rho) \sim N(0, \rho^2)$. In this policy, ρ is our tunable parameter which controls the amount of exploration in the policy. If it is too small, then there may not be enough exploration. If it is too large, then we will be choosing bids that are far from optimal, possibly without any benefit from learning.

6.4.3 A value of information policy

The pure exploitation and the excitation policies we just introduced are both relatively simple. Now we are going to consider a policy that maximizes the value of information in the future. This seems like a reasonable idea, but it requires that we think about how information now affects what decision we *might* make in the future, and this will be a bit more difficult.

Our exploitation policy assumes that the estimated parameters θ^n after n experiments is the correct value, and chooses a bid based on this estimate. Now imagine that we bid $x^n = x$ and observe K^{n+1} and $\hat{R}^{n+1}$, and use this information to get an updated estimate of θ^{n+1} as well as $\bar{R}^{n+1}$. We can then use these updated estimates to make a better decision. We want to choose the bid x that gives us the greatest improvement in the quality of a decision, recognizing that we do not know the outcome of $W^{n+1} = (\hat{R}^{n+1}, K^{n+1})$ until we actually place the bid.

Let $\theta^{n+1}(x|W^{n+1}$ be the updated estimate of θ assuming we bid $x^n = x$ and observe $W^{n+1} = (\hat{R}^{n+1}, K^{n+1})$. This is a random variable, because we are thinking about placing a bid $x^n = x$ for the $n + 1st$ auction, but we have not yet placed the bid, which means we have not yet observed W^{n+1}.

To simplify our analysis, we are going to assume that the random variable $K^{n+1} = 1$ with probability $P^{click}(x|\theta)$ and $K^{n+1} = 0$ with probability $1 - P^{click}(x|\theta)$. We are then going to assume that $\bar{R}^{n+1} = \bar{R}^n$, which is probably a fairly accurate approximation if we have observed enough auctions to get a good estimate of the revenue we will receive if a customer clicks on our ad.

We can think of this as an approximate lookahead model, where $\bar{R}^n$ does not change. We would then write our exogenous information in our lookahead model as

$$\widetilde{W}^{n,n+1} = K^{n+1},$$

where the double-superscript $(n, n+1)$ means that this is the information in a lookahead model created at time n, looking at what might happen at time $n+1$.

We next use our updating equation (6.2) for the probabilities $p_k^n = Prob[\theta = \theta_k | H^n]$. We can write these updated probabilities as $p_k^{n+1}(K^{n+1})$ to capture the dependence of the updating on K^{n+1} (equation (6.2) is written for $K^{n+1} = 1$). Since K^{n+1} can take on two outcomes (0 or 1) we will have two possible values for $p_k^{n+1}(K^{n+1})$.

Now imagine that we perform our pure exploitation policy $X^{explt}(S^n|\theta^n)$ that we described above, but we are going to do it in our approximate lookahead model (this is where we ignore changes in $\bar{R}^n$). Let $\tilde{S}^{n,n+1}$ represent our state in the lookahead model given by

$$\tilde{S}^{n,n+1}(K^{n+1}) = (\bar{R}^n, p^{n+1}(K^{n+1})).$$

Remember - since $K^{n,n+1}$ is a random variable (we are still at time n), $\tilde{S}^{n,n+1}(K^{n+1})$ is also a random variable, which is why we write its explicit dependence on the outcome K^{n+1}.

The way to think about this lookahead model is as if you are playing a game (such as chess) where you think about a move (for us, that would be the bid x^n) and then, before you make the move, think about what might happen in the future. In this problem, our future only has two outcomes (whether or not a customer clicks on the ad), which means two possible values of $\tilde{S}^{n,n+1}$, which produces two sets of updated probabilities $p^{n+1}(K^{n+1})$.

Finally, this means that there will be two values of the optimal myopic bid (using our pure exploitation policy) $X^{explt}(\tilde{S}^{n,n+1})$. The expected contribution we would make in the future is then given by $\bar{C}(\tilde{S}^{n,n+1}, \tilde{x}^{n,n+1})$ where $\tilde{x}^{n,n+1}$ (this is the decision we are thinking of making in the future) is given by

$$\tilde{x}^{n,n+1} = X^{explt}(\tilde{S}^{n,n+1})).$$

This means there are two possible optimal decisions, which means two different values of the expected contribution $\bar{C}(\tilde{S}^{n,n+1}, X^{explt}(\tilde{S}^{n,n+1}))$. For compactness, let's call these $\tilde{C}^{n,n+1}(1)$ (if $K^{n+1} = 1$) and $\tilde{C}^{n,n+1}(0)$ (if $K^{n+1} = 1$). Think of these as the expected contributions that *might* happen in the future given what we know now. Finally we can take the expectation over K^{n+1} to obtain the expected contribution of placing a bid $x^n = x$ right now, which we can compute using

$$\tilde{C}^n(x) = \sum_{k=1}^{K} \Big(P^{click}(x|\theta = \theta_k)\tilde{C}^{n,n+1}(1) + (1 - P^{click}(x|\theta = \theta_k))\tilde{C}^{n,n+1}(0)\Big)p_k^n.$$

Our policy, then, is to pick the bid x that maximizes $\tilde{C}^n(x)$. Assume that we discretize our bids into a set $\mathcal{X} = \{x_1, \ldots, x_M\}$. Our value of information policy would be written

$$X^{VoI}(S^n) = \arg\max_{x \in \mathcal{X}} \tilde{C}^n(x).$$

Value of information policies are quite powerful, but they are clearly harder to compute. Imagine, for example, doing this computation when there is more than two outcomes. For example, if we had not made our simplification of holding $\bar{R}^n$ constant, we would have to recognize that this state variable is also changing.

- Notes:
  - » Value of information policies are clearly much harder to compute.
  - » Have to use care for some problems (such as this one) where the value of a single observation may be low. This is typical when the random outcome is 0/1.
  - » A better strategy is to assume we are going to fix a bid x for a period of time. Instead of 0 or 1 clicks, we might get a number of clicks described by a binomial distribution.

Week 12 - Monday

Template

Narrative

Basic model

Uncertainty modeling

Designing policies