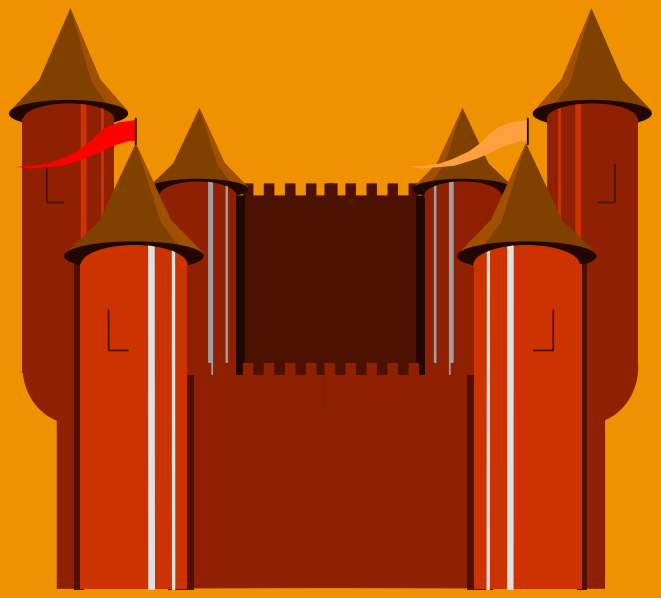
ORF 544

Stochastic Optimization and Learning

Spring, 2019



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

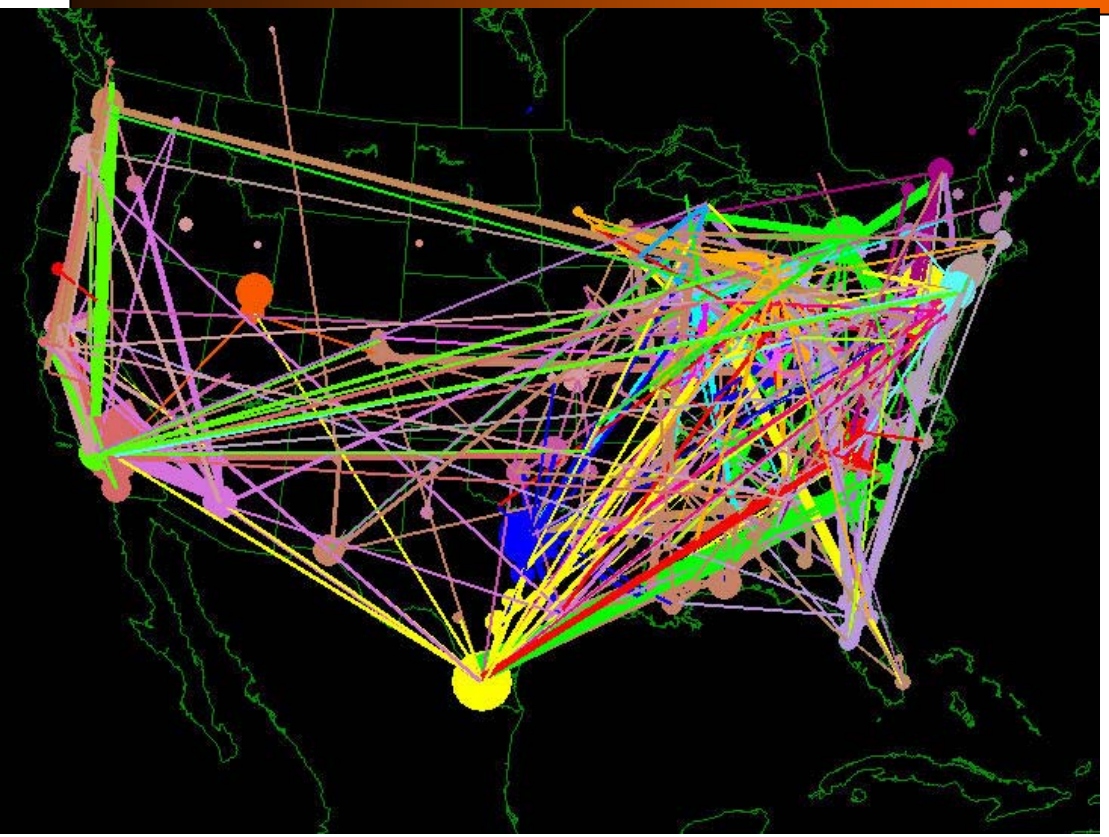
Week 6

General sequential decision problems

Universal modeling framework

Applications

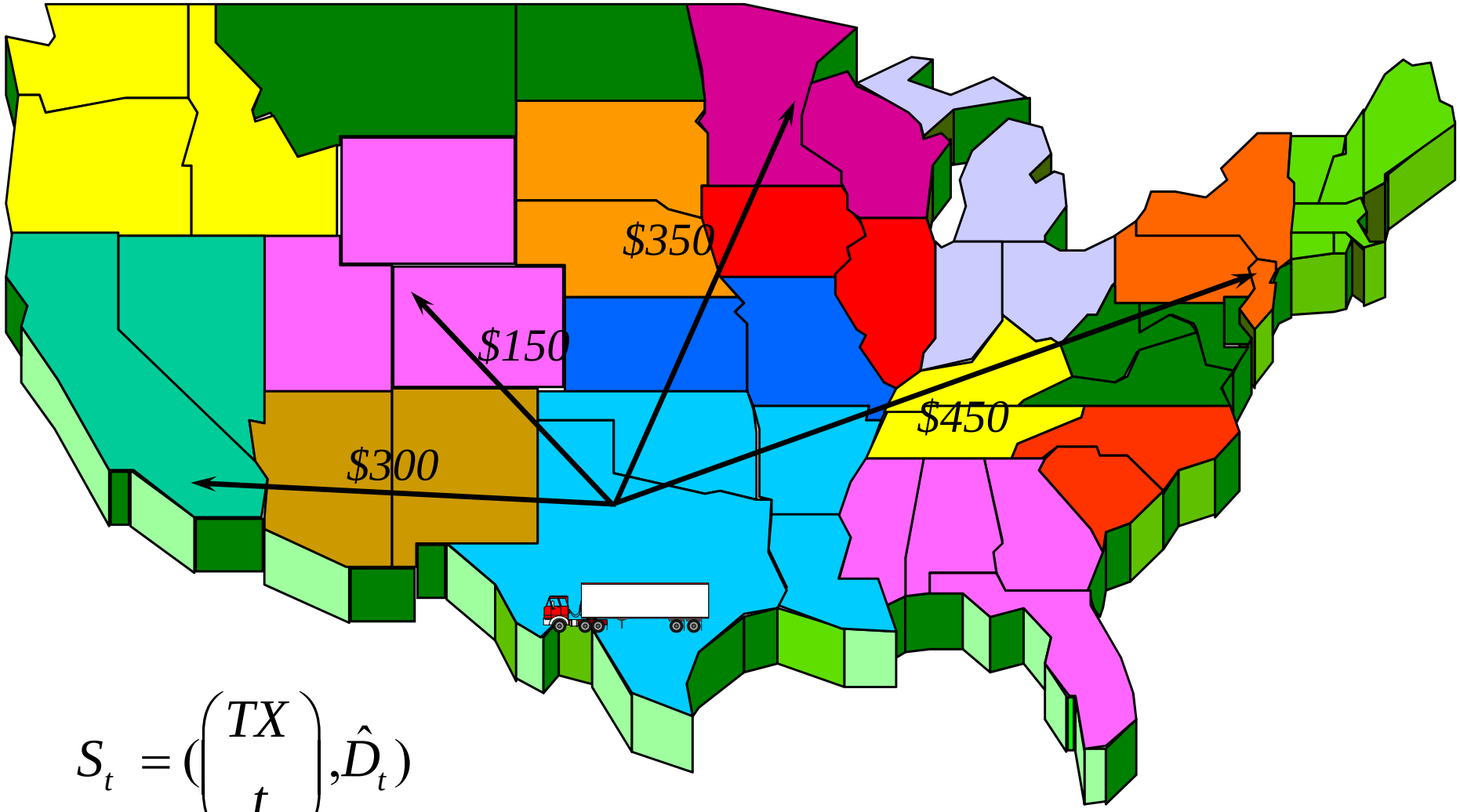
Fleet management



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

Nomadic trucker illustration

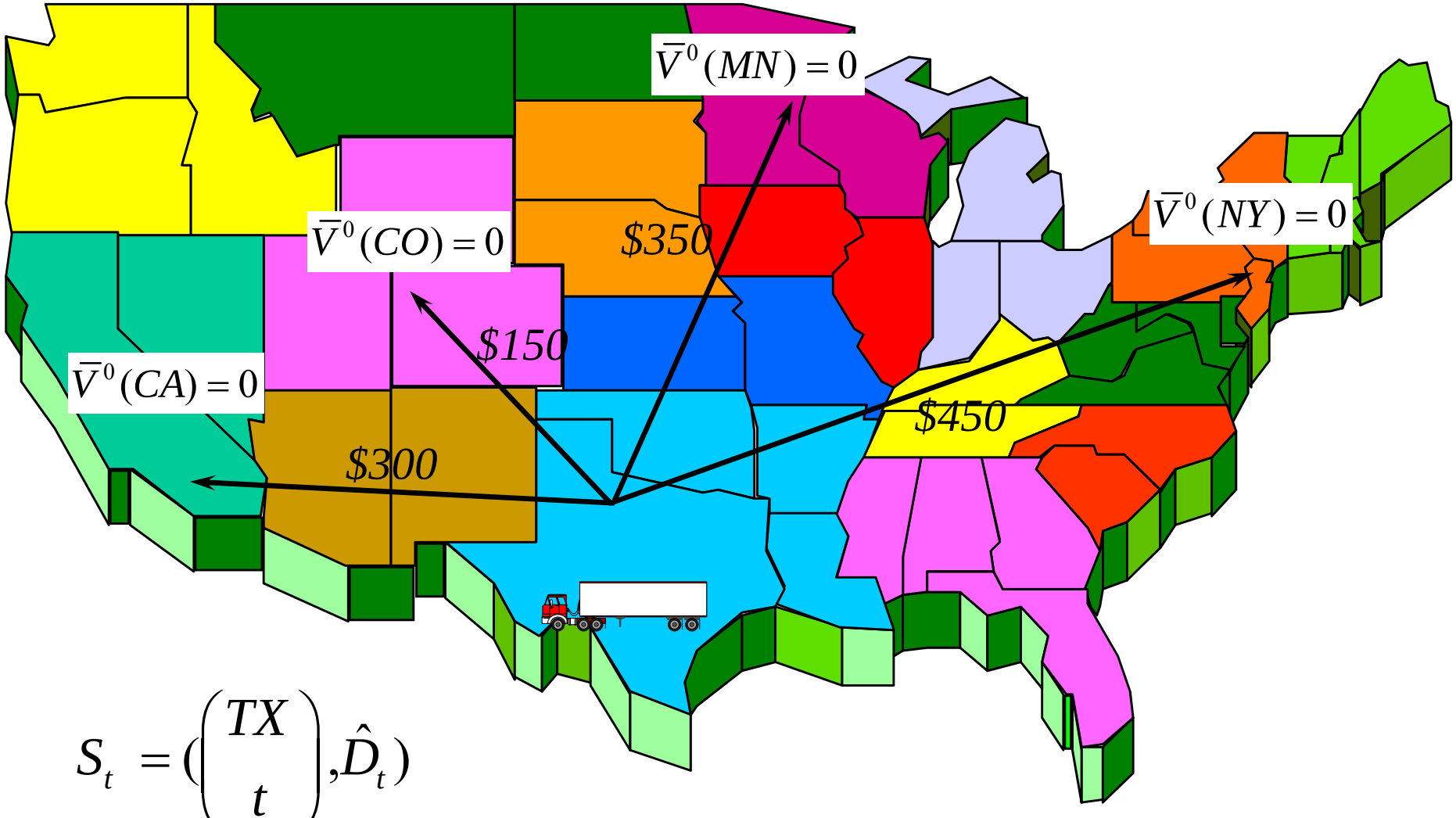
- Pre-decision state: we see the demands



$$S_t = \left(\begin{pmatrix} TX \\ t \end{pmatrix}, \hat{D}_t\right)$$

Nomadic trucker illustration

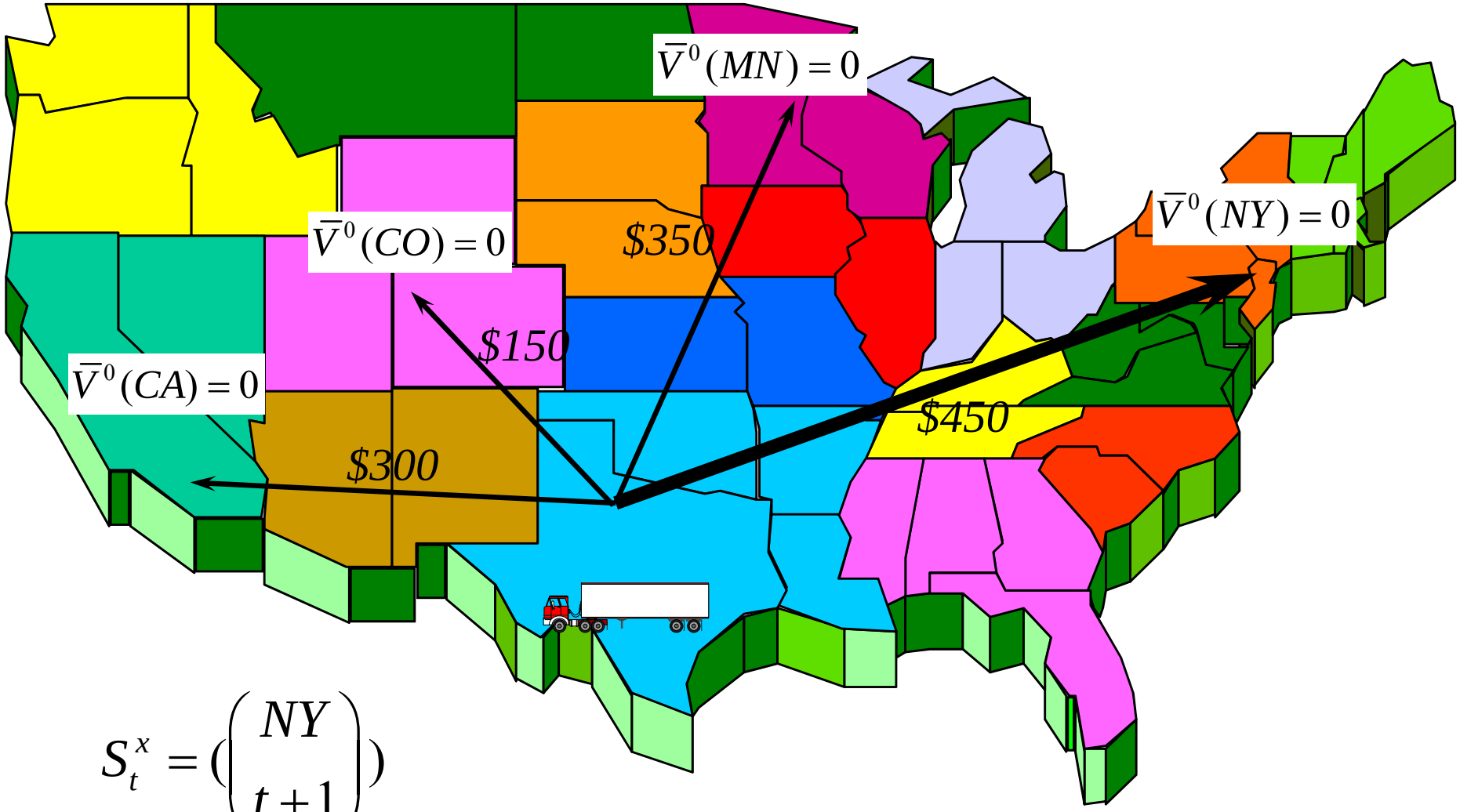
- We use initial value function approximations…



$$S_t = \left(\begin{pmatrix} TX \\ t \end{pmatrix}, \hat{D}_t\right)$$

Nomadic trucker illustration

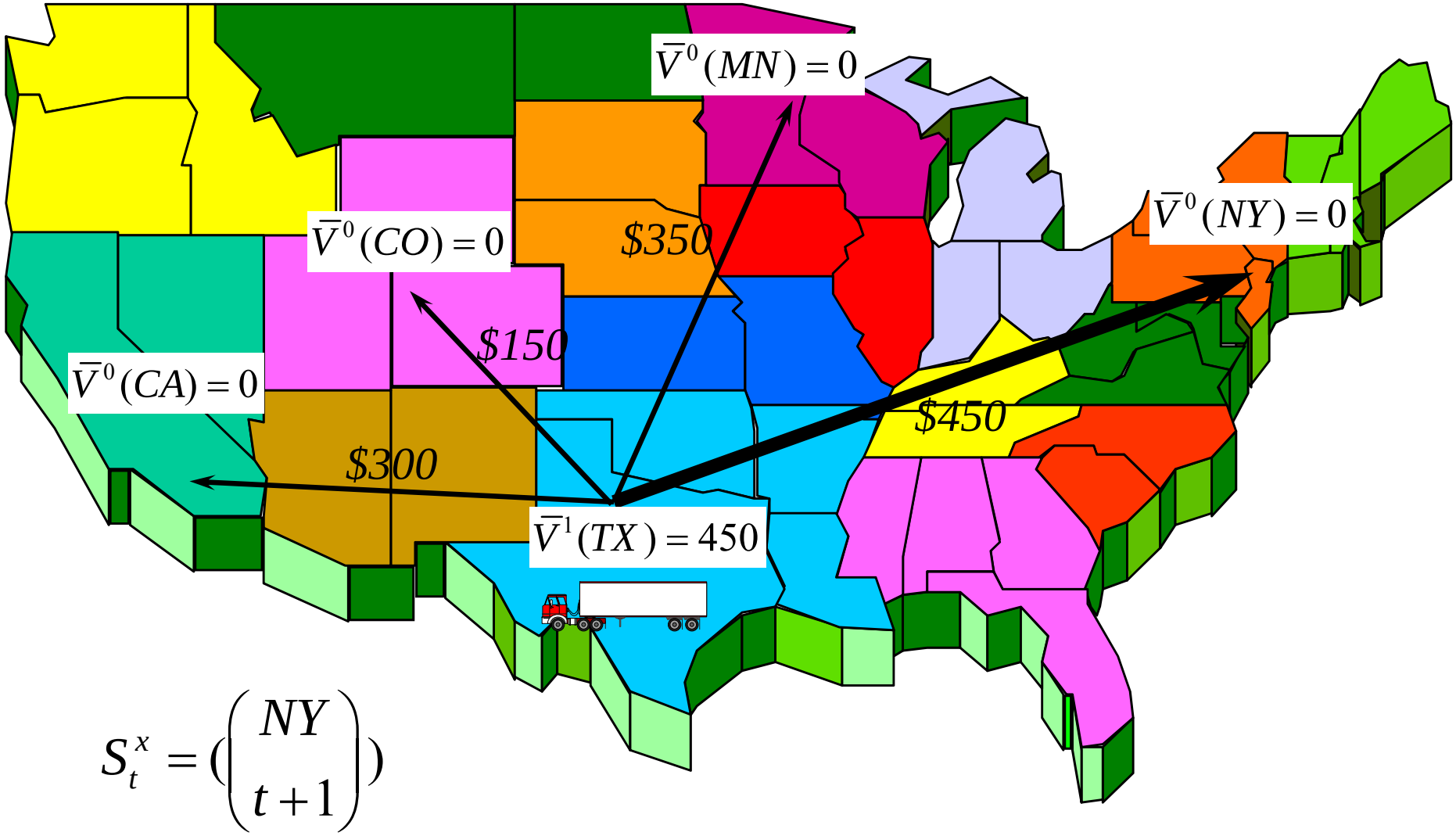
- … and make our first choice: x^1



$$S_t^x = \begin{pmatrix} NY \\ t+1 \end{pmatrix}$$

Nomadic trucker illustration

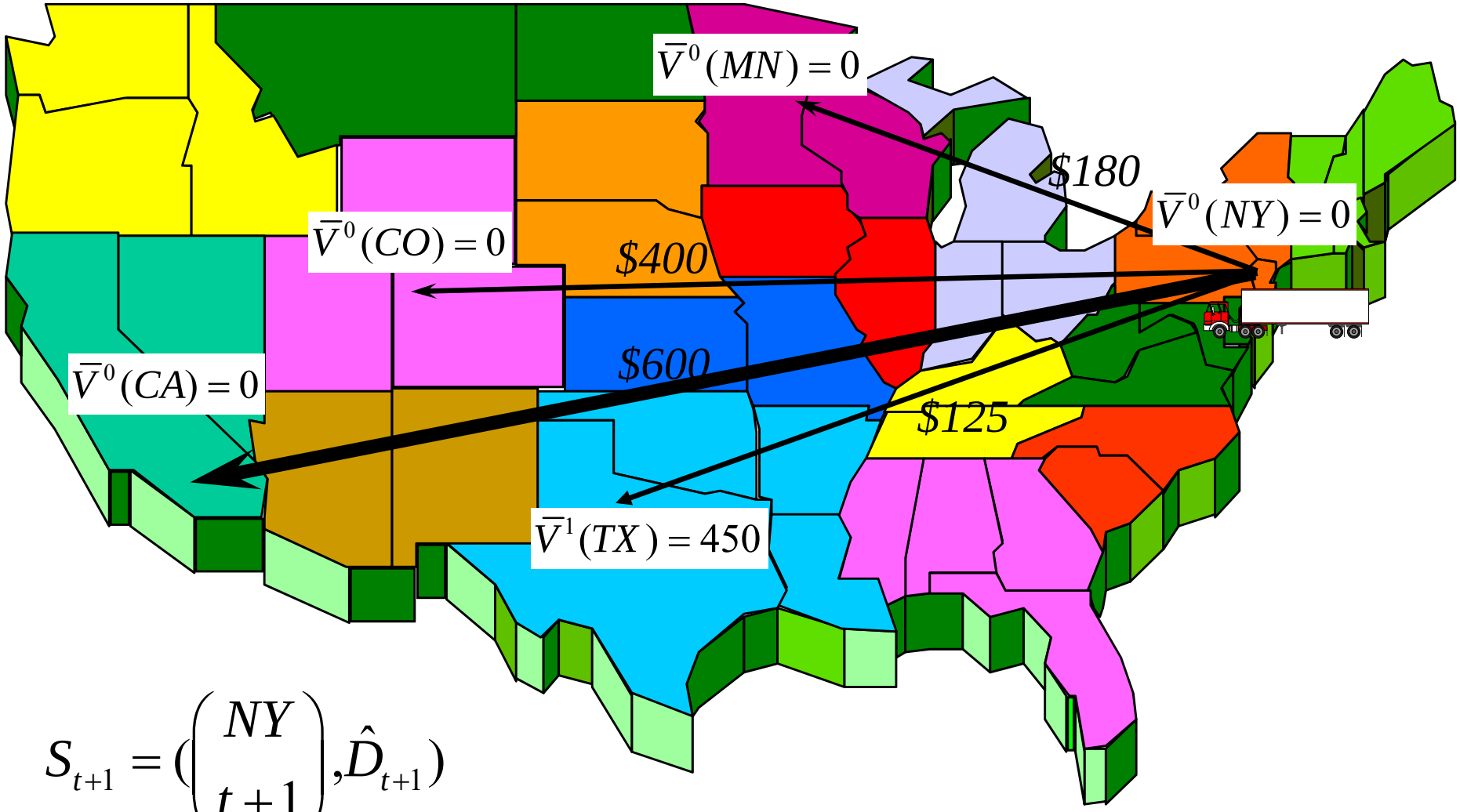
- Update the value of being in Texas.



$$S_t^x = \begin{pmatrix} NY \\ t+1 \end{pmatrix}$$

Nomadic trucker illustration

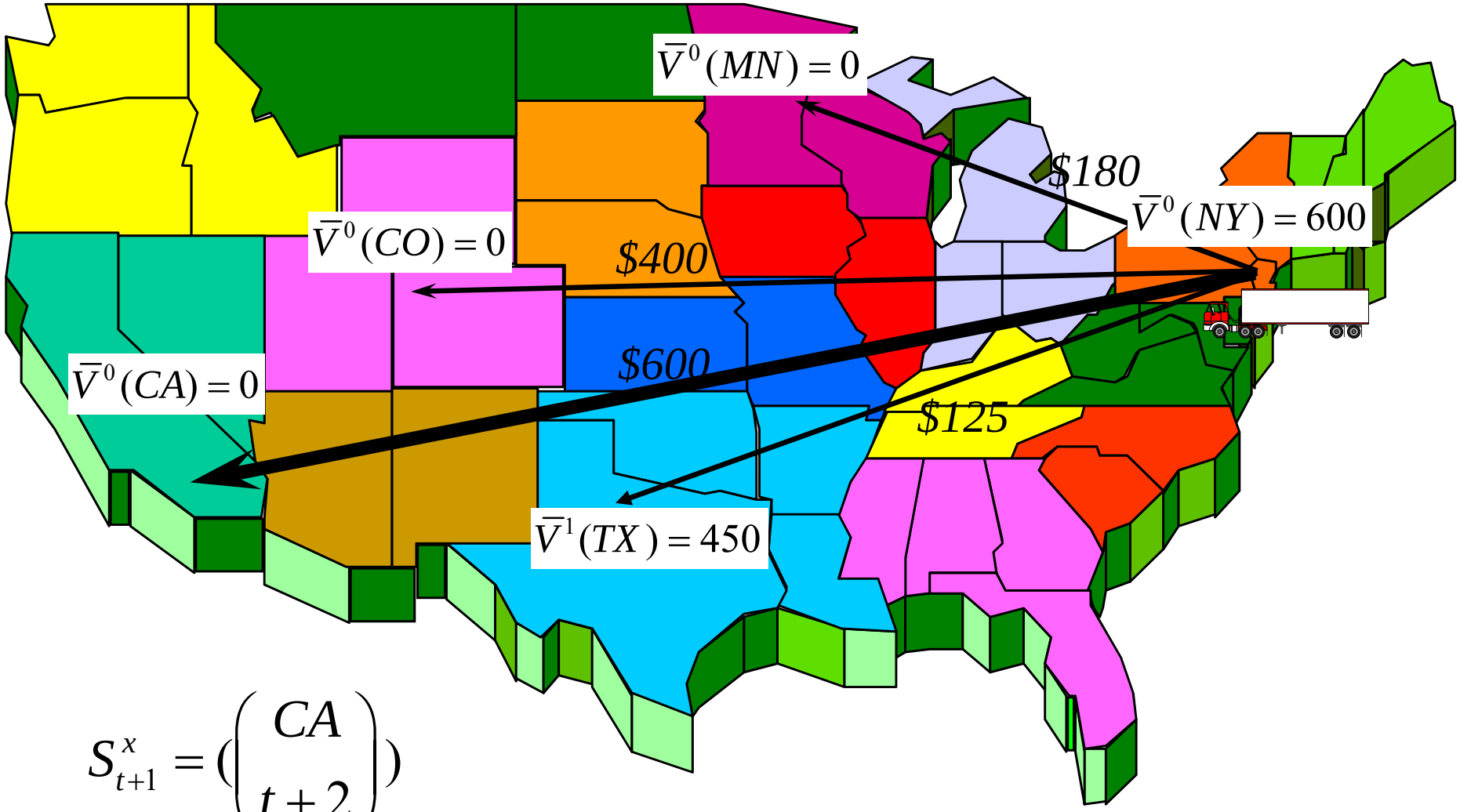
- Now move to the next state, sample new demands and make a new decision



$$S_{t+1} = \left(\begin{pmatrix} NY \\ t+1 \end{pmatrix}, \hat{D}_{t+1}\right)$$

Nomadic trucker illustration

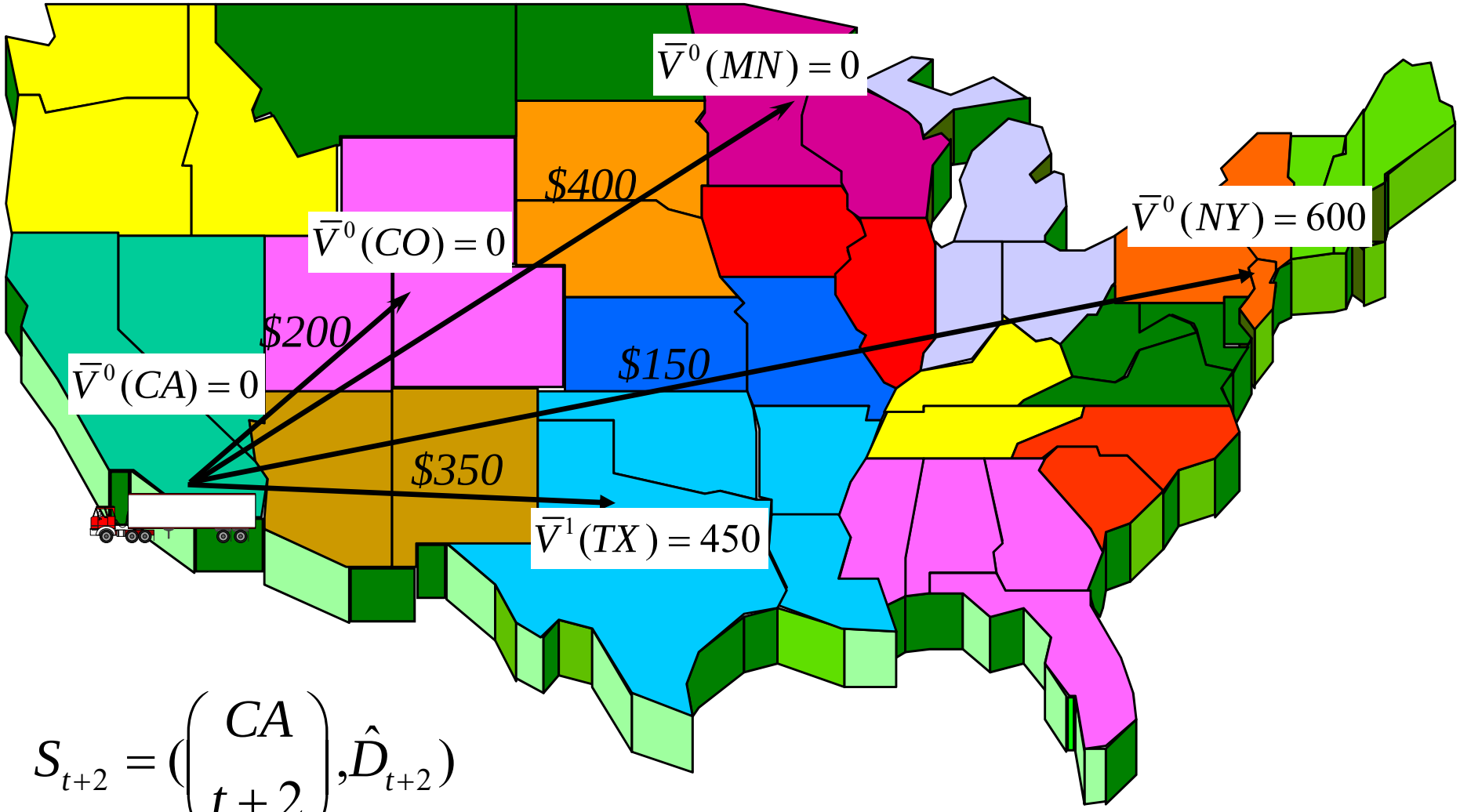
- Update value of being in NY



$$S_{t+1}^x = \begin{pmatrix} CA \\ t+2 \end{pmatrix}$$

Nomadic trucker illustration

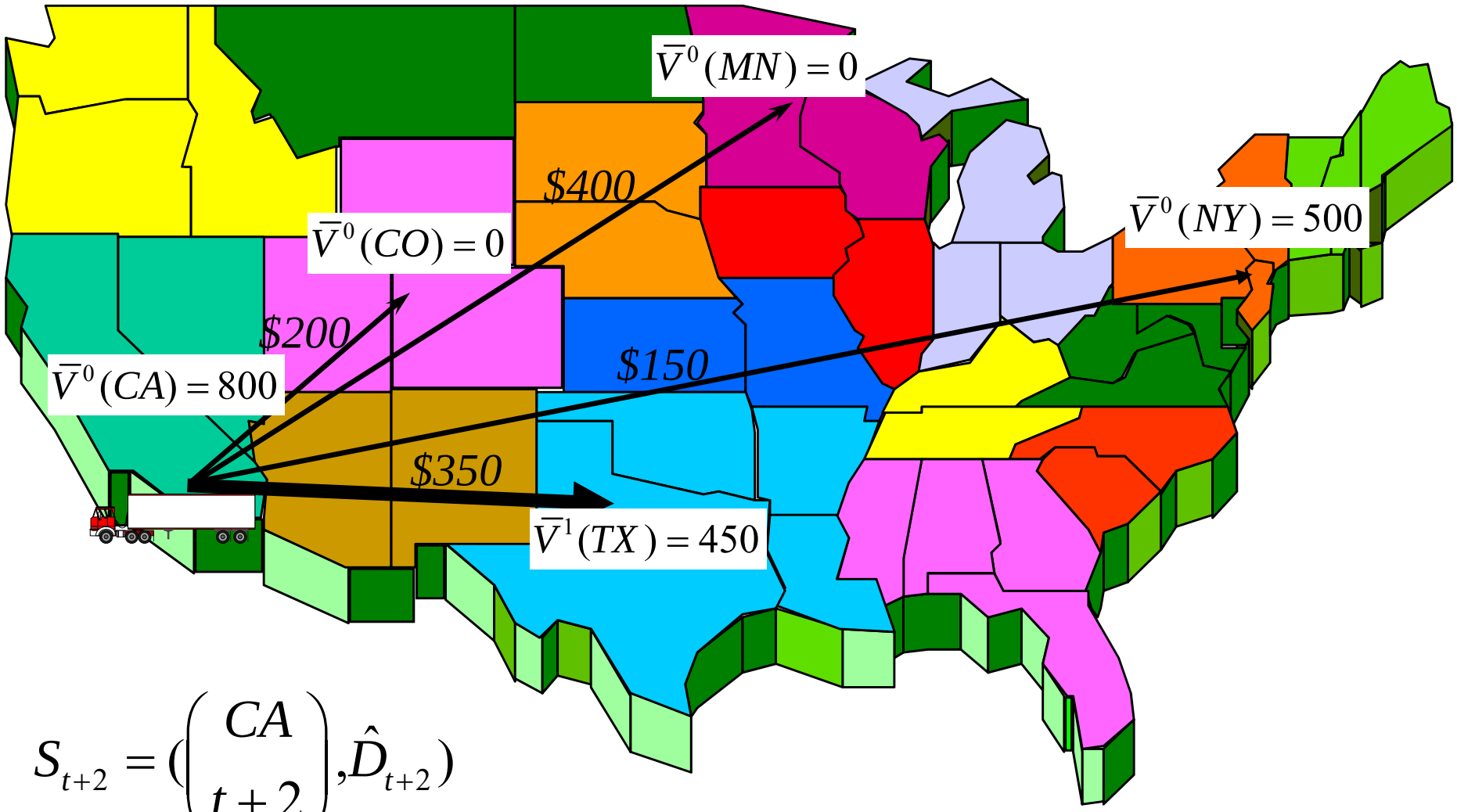
- Move to California.



$$S_{t+2} = \left(\begin{pmatrix} CA \\ t+2 \end{pmatrix}, \hat{D}_{t+2}\right)$$

Nomadic trucker illustration

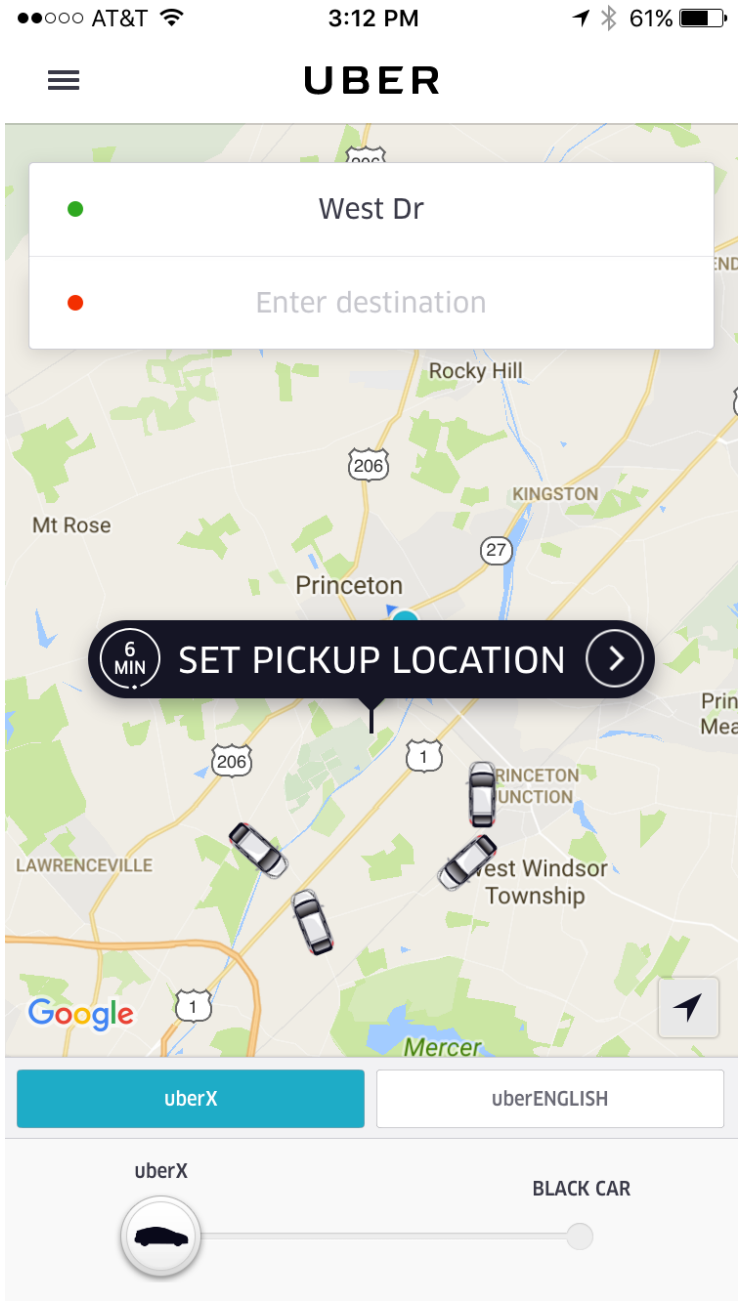
- Make decision to return to TX and update value of being in CA

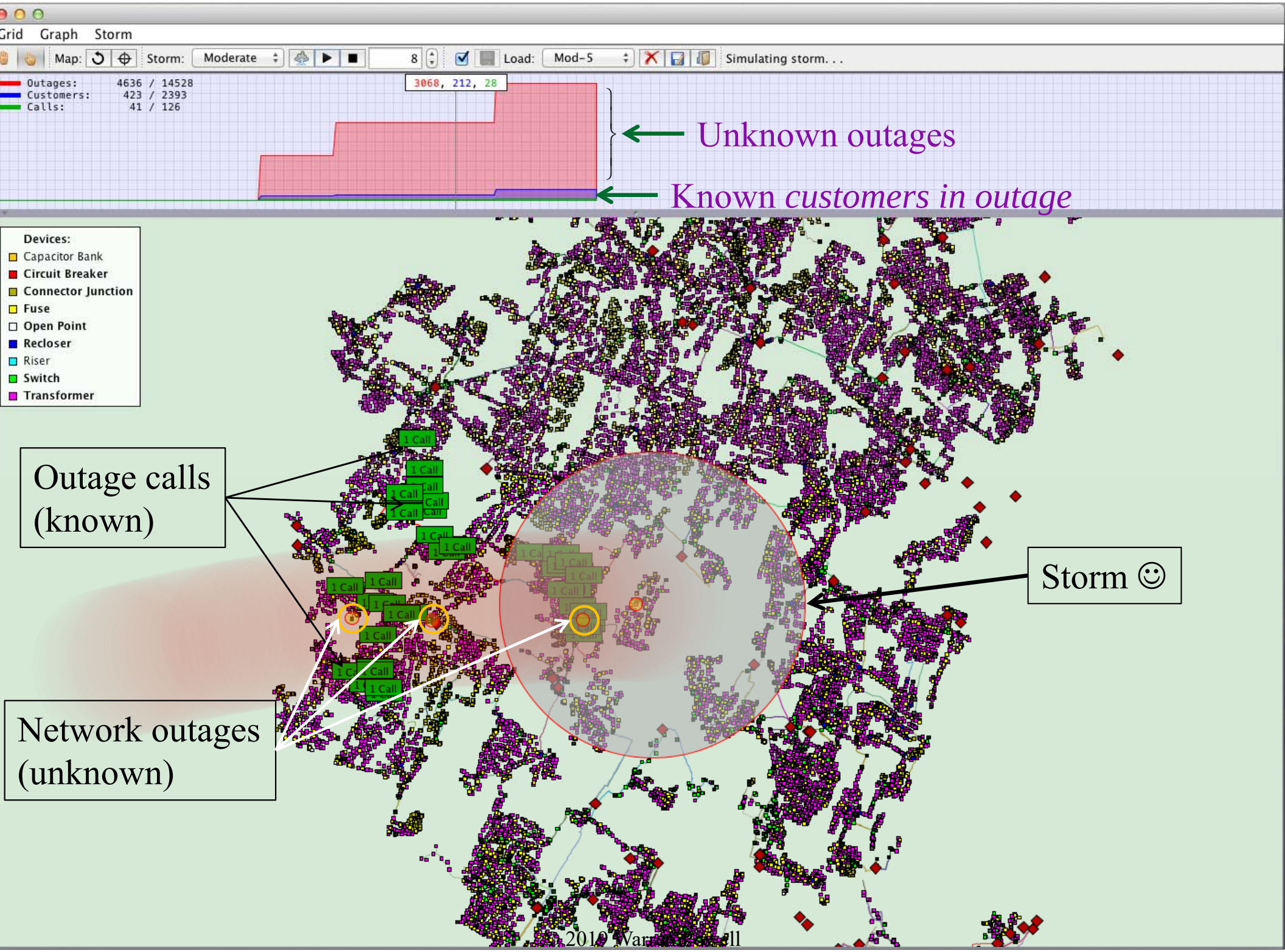


$$S_{t+2} = \left(\begin{pmatrix} CA \\ t+2 \end{pmatrix}, \hat{D}_{t+2}\right)$$

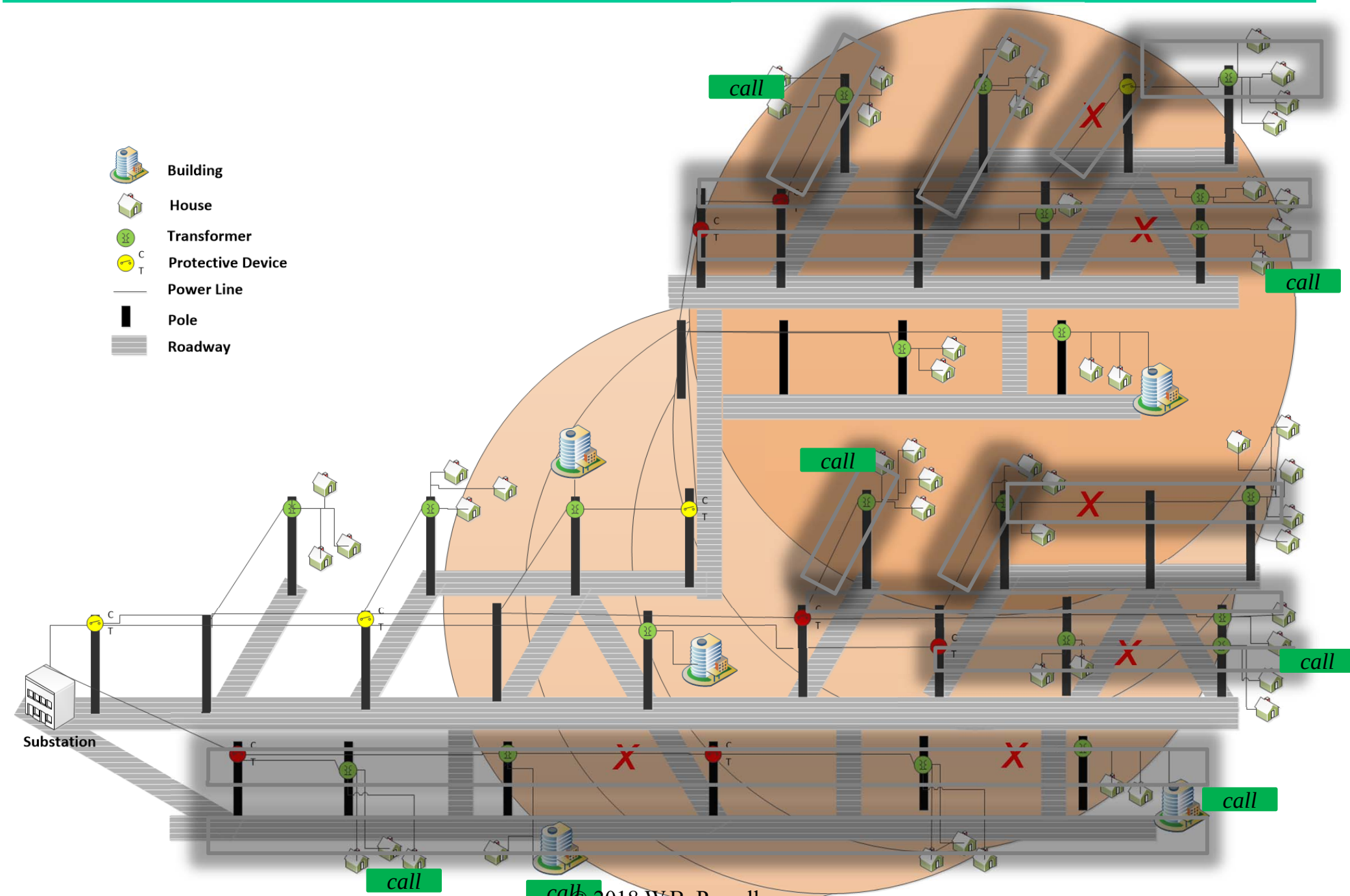
Fleet management

- 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, …)

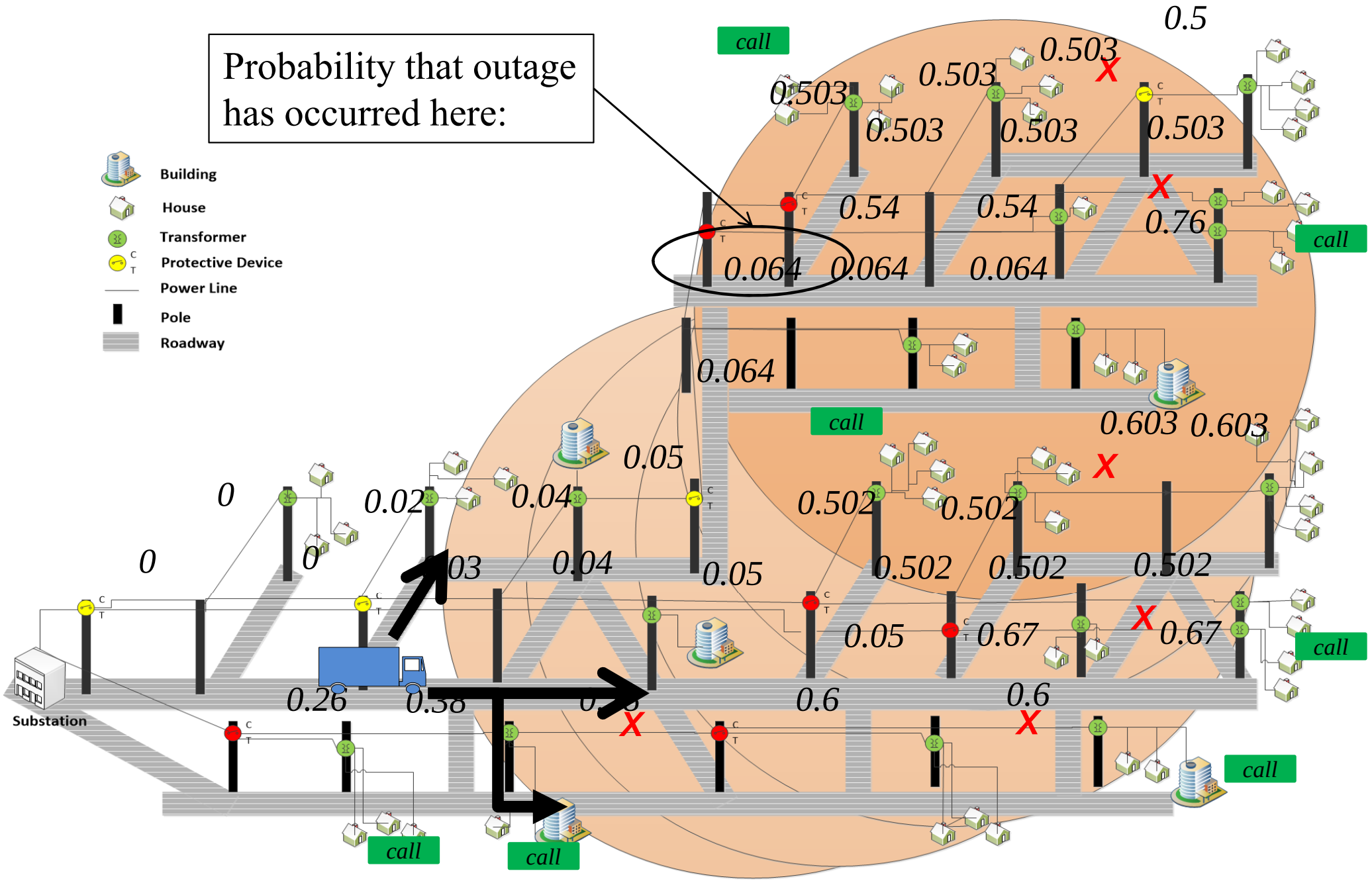



Grid Graph Storm
Map: Storm: Moderate 8 Load: Mod-5 Simulating storm. . .
Outages: 4636 / 14528
Customers: 423 / 2393
Calls: 41 / 126
3068, 212, 28
Unknown outages
Known customers in outage
Devices:
Capacitor Bank
Circuit Breaker
Connector Junction
Fuse
Open Point
Recloser
Riser
Switch
Transformer
Outage calls (known)
Storm ☺
Network outages (unknown)
1 Call

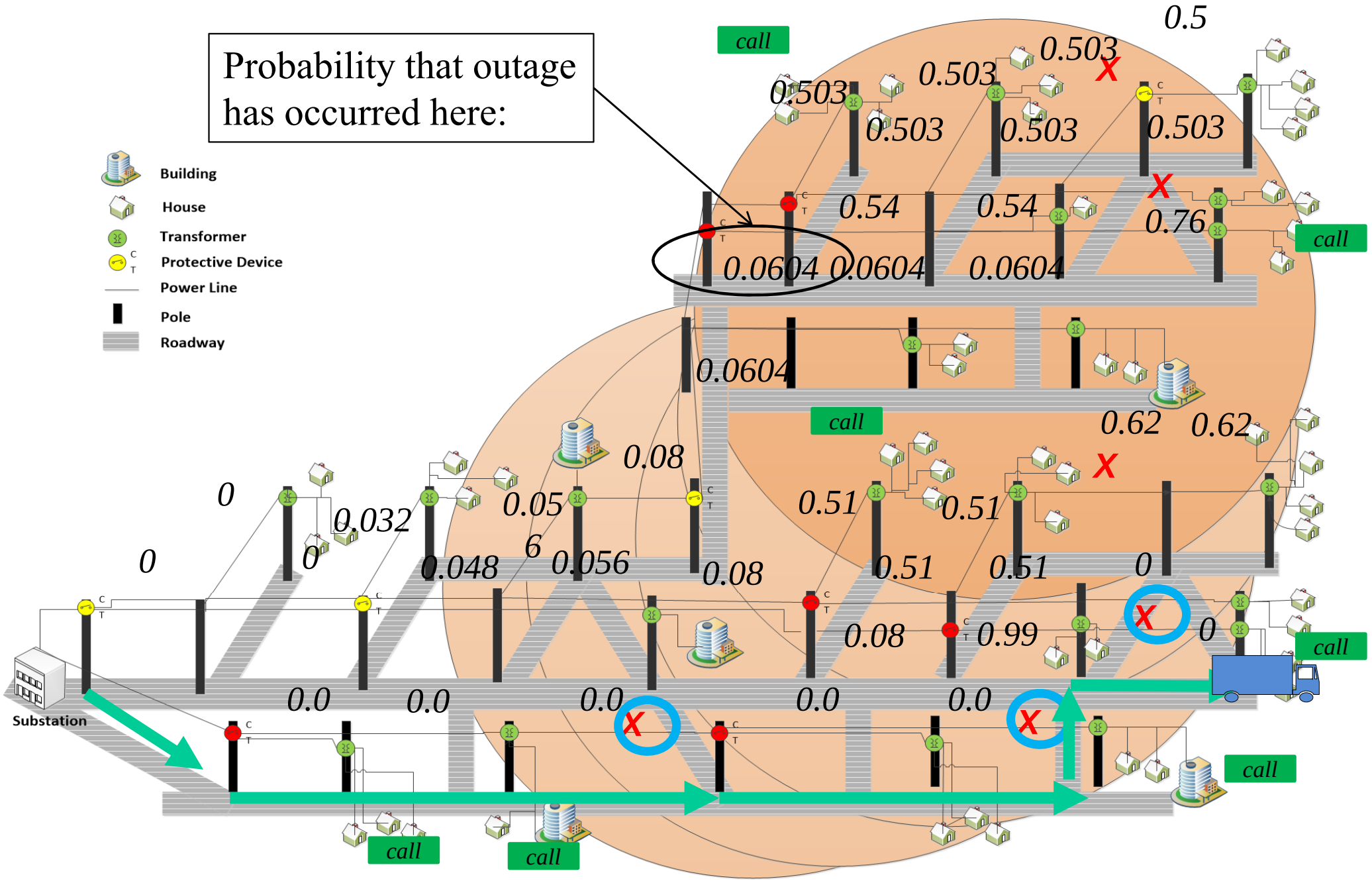
Problem Description - Emergency Storm Response



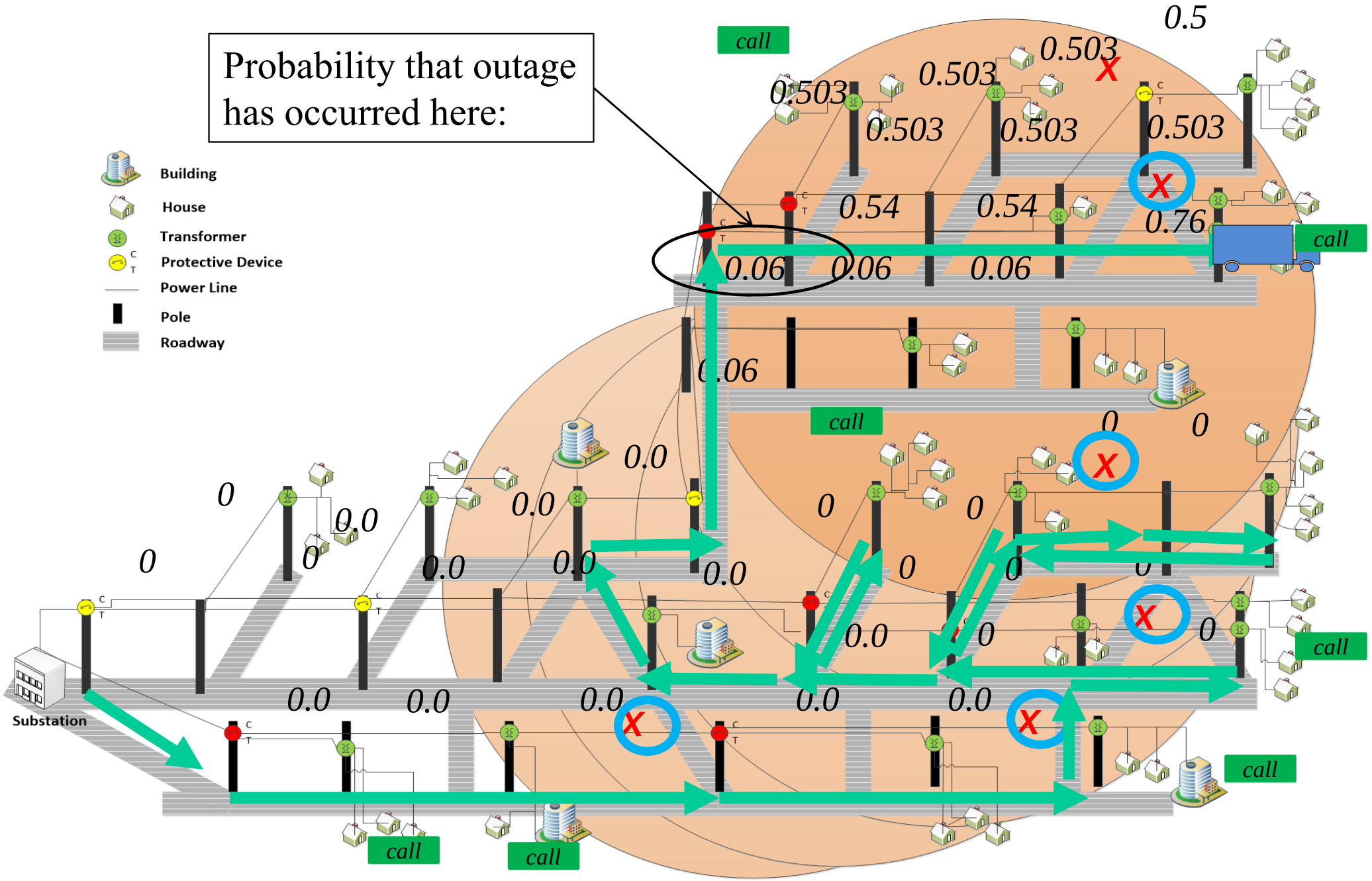
Emergency storm response



Emergency storm response



Emergency storm response

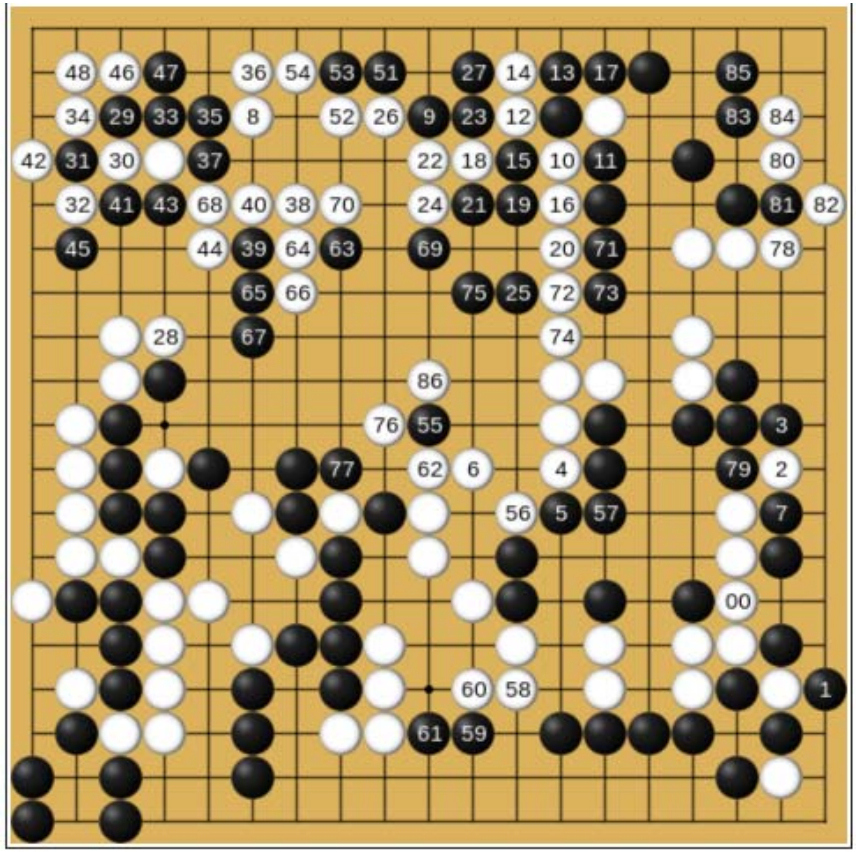


Monte Carlo tree search

- AlphaGo

» State variables:
  - State of the board
  - Beliefs about opponent strategies

» Uses hybrid of policies:
  - MCTS
  - PFA
  - VFA



Planning cash reserves

- How much money should we hold in cash given variable market returns and interest rates to meet the needs of a business?

Stock prices

Bonds

Energy storage

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

A note on notation

Notational style

- The challenge of communicating:
  - » “The elevator lift is being fixed for the next day. During that time we regret that you will be unbearable.” - Sign in a Bucharest business.
  - » “Please do not feed the animals. If you have any suitable food, give it to the guard on duty.” - Sign at a Budapest zoo.
  - » “You are invited to take advantage of the chambermaid.” - A Japanese hotel.
  - » In an Acapulco hotel: “The manager has personally passed all the water served here.”
  - » From a tailor shop: “Place orders early because in a big rush we will execute customers in strict rotation.”

Notational style

- Principles of good mathematical notation:
  - » Notation is a language - if the language is hard to learn, others will have difficulty speaking it.
  - » Minimize the number of variables you introduce (i.e. keep your vocabulary small).
  - » Make variables as mnemonic as possible.
  - » Follow consistent, standard conventions.
  - » Organize your variables into natural groupings:

Notational style

- Variables
  - » A basic variable - lower case and script: χ
  - » Use subscripts to identify elements of a vector:

$$\chi_{tij} \qquad \chi_{ti} \qquad \chi_{t}$$

  - » Use superscripts to create different flavors of a variable:

	Decisions :	**Costs :**
purchase:	x^p	c^p
sell :	x^s	c^s
hold :	x^h	c^h

$$x = \left\{x^p, x^s, x^h\right\} \qquad c = \left\{c^p, c^s, c^h\right\}$$

$$\textbf{Total costs} = c^p x^p + c^s x^s + c^h x^h = cx$$

Notational style

- Variables
  - » *Never* use variables with more than one letter:

    PC = Purchasing cost (is it P times C ?)

    BC_t = Battery charge at time t

  - » Using multiple letter variables is popular in the business community. If you are presenting an equation in a memo to a management audience, it is perfectly reasonable to write:

    $$Fleet_t = Const_0 + Const_1 \times Tons_t + Const_2 \times speed_t$$

    This is acceptable in this setting, because the variables define themselves, and you are never going to manipulate this equation.

Notational style

- Sequencing subscripts:
  - » When you have multiple subscripts:
    - Roughly sequence them in the order that you would place them in a summation:

$$x_{ti} = \text{Dollars invested in asset } i \text{ in time period } t.$$

$$C(x) = \sum_t \sum_i c_{ti} x_{ti}$$

$$x_t = (x_{ti})_{i \in \mathcal{I}}$$

    - Always model discrete time periods as a subscript, because it is an element of a vector over different time periods.

Notational style

- Arguments, superscripts and hats:
  - » Use arguments only when there is a functional dependence:

$$F(x) \quad \text{or} \quad a(x)$$

  - » If we are estimating a variable iteratively, again use a superscript to identify different versions of the variable:

$$x^{n+1} = F(x^n)$$

Use n for iterations. If you need outer and inner iterations, use n and m.

Notational style

- Choice of letters:
  - » Use hats and bars (etc) to indicate different estimates/observations of the same variable:

    $\bar{x}, \quad \hat{x}$

  - » Constants:
    - General rule: *a*, *b*, *c*, *d* and *e*
  - » Physical parameters (ratios, speeds, …):
    - Use Greek letters:

      $\alpha \quad \beta \quad \gamma \quad \lambda \quad \rho \quad \delta$

    - Avoid unusual Greek letters, e.g.

      $\xi \quad \zeta$ *(hard to pronounce/remember)*

Notational style

- Time:
  - » Always use *t* or its variants.
    - A flight going from city *i* to city *j* might start at time *t* and arrive a time *t'*.
    - You might purchase an option at time *t* which can be exercised at time *t'*.
  - » Use τ to indicate an interval of time, such as the time required to complete an action.
  - » Let $t = 0$ represent "here and now."

Notational style

- Modeling new information:
  - » It helps to identify variables that represent new information arriving in a time period.
  - » Suggest using "hats" to indicate information that *first becomes known at time t :*

$\hat{D}_t$ = Customer demand for a product in time t.

$\hat{p}_t$ = Market price of a stock at time t.

$\hat{b}_t$ = A baseball player's batting average in year t.

  - » Use "bars" for statistics calculated from exogenous information:

$$\bar{D}_t = \frac{1}{t}\sum_{t'=1}^{t} \hat{D}_{t'}$$

$$\bar{p}_t = (1-\alpha)\bar{p}_{t-1} + \alpha \hat{p}_t$$

$$\bar{b}^N = \frac{1}{N}\sum_{n=1}^{N} \hat{b}^n$$

Notational style

- Sets:
  - » Sets - Calligraphic (or script) capital letters

    $x \in \mathcal{X}$ $\quad a \notin \mathcal{A}$ *"Monotype Corsiva" in equation editor*

    Calligraphic in Latex
    - Subsets - suggest using:

      $\mathcal{X}_i \quad \mathcal{A}_b$
  - » Use sets for summations:
    - Sets are especially useful when there is not a natural indexing:

      $\sum_{i \in \mathcal{I}} x_i$ is better than: $\sum_{i=1}^{n} x_i$

      Let a be a multiattribute vector, where:

      $a \in \mathcal{A}$

      It is then easy to write:

      $\sum_{a \in \mathcal{A}} x_a$
    - We can also write vectors:

      $x = (x_t)_{t \in \mathcal{T}} \qquad x_t = (x_{ti})_{i \in \mathcal{I}}$

Notational style

- Notational style
  - » Functions:

    Argument notation (preferred in engineering):

    $F(x)$

    $F(x, y)$

    Mapping notation (preferred in mathematics):

    $F : \mathcal{X} \to \Re$

    $F : \mathcal{X} \times \mathcal{Y} \to \Re$
  - » Use lower case letters for individual functions; use upper case for sums:

$$C(x) = \sum_{t \in \mathcal{T}} c_t(x_t)$$

Notational style

- Notational style
  - » Matrices:
    - Square, capital letters:
      A, B
  - » Multiplication of matrices and vectors:

    Use cx when it is understood that we want an inner product. The "transpose" is implicit. There is no need to write:

    $$c^T x$$

Modeling “time”

Modeling "time"

- There are two fundamental ways for modeling the timing of information and decisions

  » Counters n – Here we count experiments, iterations, arrivals of customers, …
    - We index counters in superscripts: S^n, x^n, W^n

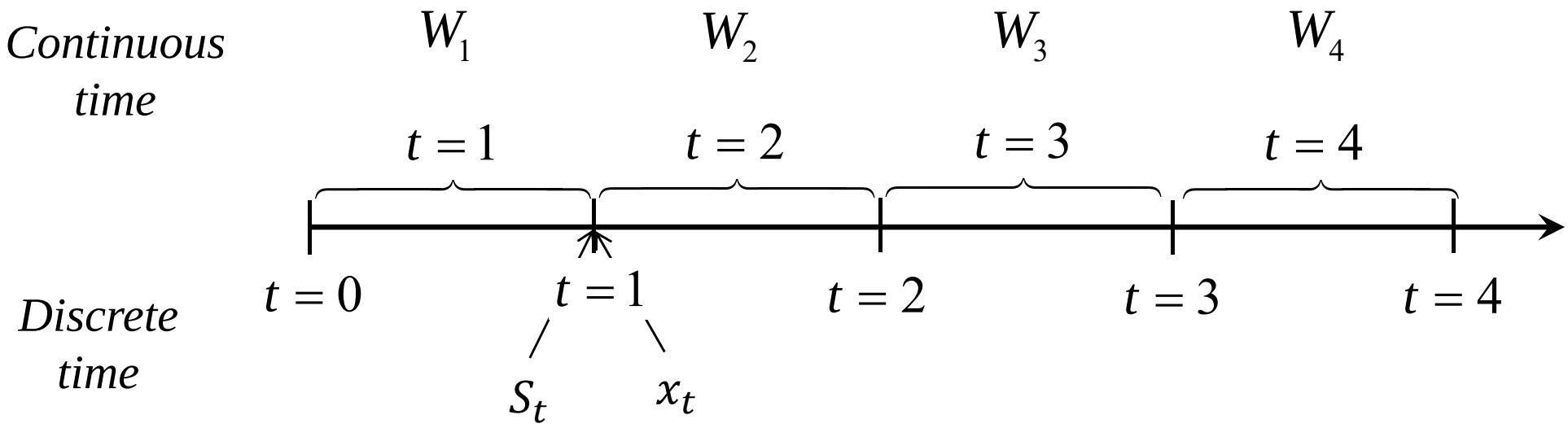
  » Time t – This is always in discrete units:
    - Seconds, minutes, hours, days, weeks, months, years
    - We index time in subscripts: S_t, x_t, W_t

  » Combinations
    - We may want to index the time t within the nth simulation:

$$S_t^n, \qquad x_t^n, \qquad W_t^n$$

Modeling "time"

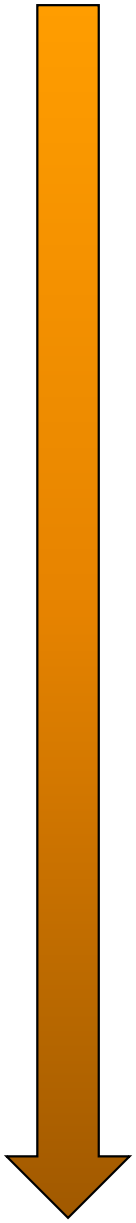
- We need a system for indexing time. In particular, it is important to know the mapping between discrete and continuous time.



It is useful to think of information as arriving continuously over time.
Functions (states, decisions) are measured at a point in time.
At time t, anything $t' \leq t$ is known, anything $t' > t$ is unknown.

Time scales in energy

- Frequency regulation
- Power quality management
- Battery arbitrage
- Energy shifting
- Demand peak management - Many utilities impose charges based on peak usage over a month, quarter or even a year.
- Peak management for avoiding capacity expansion
- Backup power for outages



Modeling "time"

- Modeling multiple time scales:

  » Strategy 1: If we wish to model hour h within day d, we can let time be represented by (d, h).

  » Strategy 2: Better strategy: Let t index the finest increment (e.g. hour), and then define:
  - Let $d(t)$ =the day in which hour t falls.
  - Define sets that capture appropriate points in time (e.g. if we want to make certain decisions at noon each day):
  - $\mathcal{T}^{daily}$=Set of time periods that correspond to noon each day. So, we can say "if $t \in \mathcal{T}^{daily}$ then do…"

Modeling sequential decision problems

Modeling

- We propose to model problems along five fundamental dimensions:
  - » State variables
  - » Decision variables
  - » Exogenous information
  - » Transition function
  - » Objective function
  - » This framework draws heavily from Markov decision processes and the control theory communities, but it is not the standard form used anywhere.

Modeling dynamic problems

- The system state:

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

Belief about traffic delays

Belief about the status of equipment

Bizarrely, only the controls community has a tradition of actually defining state variables. We return to state variables later.

Modeling dynamic problems

- Decisions:

Markov decision processes/Computer science

a_t = Discrete action

Control theory

u_t = Low-dimensional continuous vector

Operations research

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

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

Problem classes

- Types of decisions
  - » Binary
    $$x \in X = \{0,1\}$$
  - » Finite
    $$x \in X = \{1,2,...,M\}$$
  - » Continuous scalar
    $$x \in X = [a,b]$$
  - » Continuous vector
    $$x = (x_1,...,x_K), \quad x_k \in \mathbb{R}$$
  - » Discrete vector
    $$x = (x_1,...,x_K), \quad x_k \in \mathbb{Z}$$
  - » Categorical
    $$x = (a_1,...,a_I), \quad a_i \text{ is a category (e.g. red/green/blue)}$$

Modeling dynamic problems

- Exogenous information:

W_t = New information that first became known at time t

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

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

New drivers being hired to the network

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

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

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

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

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

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

Modeling dynamic problems

- The transition function

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

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

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

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

Also known as the:

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

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

Modeling dynamic problems

- The objective function

Dimensions of objective functions

- » Type of performance metric
- » Final cost vs. cumulative cost
- » Expectation or risk measures
- » Mathematical properties (convexity, monotonicity, continuity, unimodularity, …)
- » Time to compute (fractions of seconds to minutes, to hours, to days or months)

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

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

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:

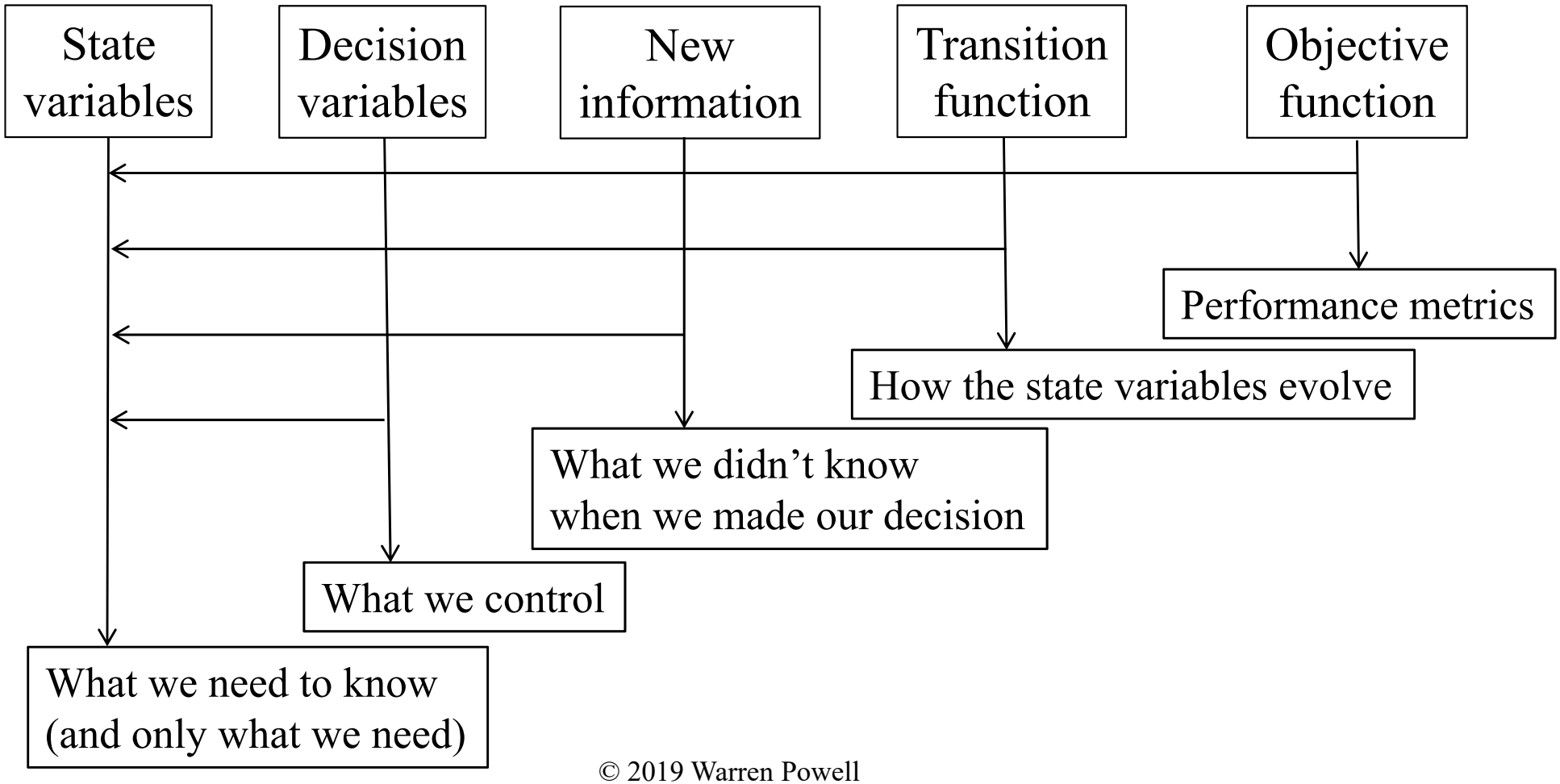
$$S_{t+1} = S^M\left(S_t, x_t, W_{t+1}\right)$$

  - » Exogenous information:

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

Elements of a dynamic model

- The modeling process
  - » I conduct a conversation with a domain expert to fill in the elements of a problem:



State variables

The state variables

- What is a state variable?
  - » Bellman's classic text on dynamic programming (1957) describes the state variable with:
    - "… we have a physical system characterized at any stage by a small set of parameters, the *state variables*."
  - » The most popular book on dynamic programming (Puterman, 2005, p.18) "defines" a state variable with the following sentence:
    - "At each decision epoch, the system occupies a *state*."
  - » Wikipedia:
    - "State commonly refers to either the present condition of a system or entity" or….
    - A state variable is one of the set of variables that are used to describe the mathematical 'state' of a dynamical system

The state variables

- What is a state variable?
  - » Kirk (2004), an introduction to control theory, offers the definition:
    - A state variable is a set of quantities $x_1(t), x_2(t), \ldots$ which if known at time $t = t_0$ are determined for $t \geq t_0$ by specifying the inputs for the system for $t \geq t_0$.
    - … or "all the information you need to model the system from time *t* onward." True, but vague (and only for deterministic problems.
  - » Cassandras and Lafortune (2008):
    - The *state* of a system at time t_0 is the information required at t_0 such that the output [cost] *y(t)* for all $t > t_0$ is uniquely determined from this information and from $u(t)$, $t \geq t_0$.
    - Again, consistent with the statement "all the information you need to model the system from time t_0 onward," but then why do they later talk about "Markovian" and "non-Markovian" queueing systems?

The state variables

- From *Probability and Stochastics* by Erhan Cinlar (2011):

> The definitions of "time" and "state" depend on the application at hand and on the demands of mathematical tractability. Otherwise, if such practical considerations are ignored, every stochastic process can be made Markovian by enhancing its state space sufficiently.

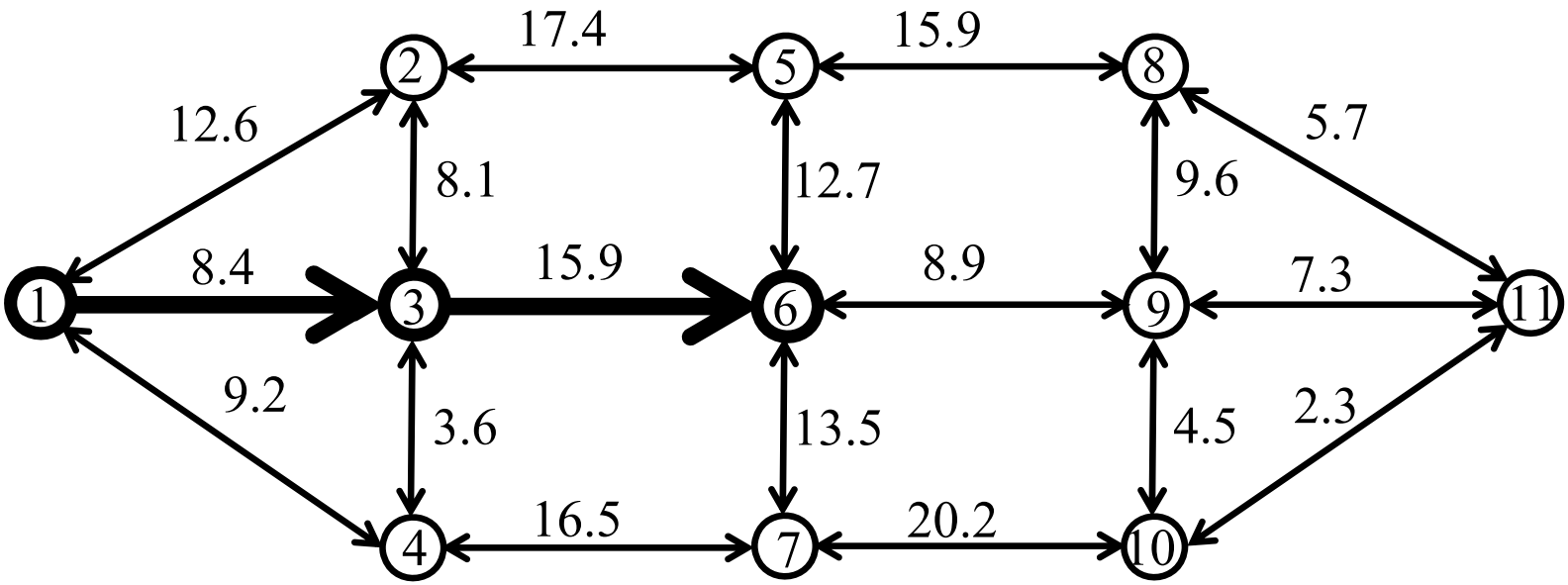
  - » Question: Why would you ever model a stochastic process where you intentionally left needed information out of the state variable?

The state variables

- There appear to be two ways of approaching a state variable:
  - » The mathematician's view – The state variable is a given, at which point the mathematician will characterize its properties ("Markovian," "history-dependent," …)
  - » The modeler's view – The state variable needs to be constructed from a raw description of the problem. Information should not be excluded due to computational tractability until *after* a solution strategy has been designed.

The state variable

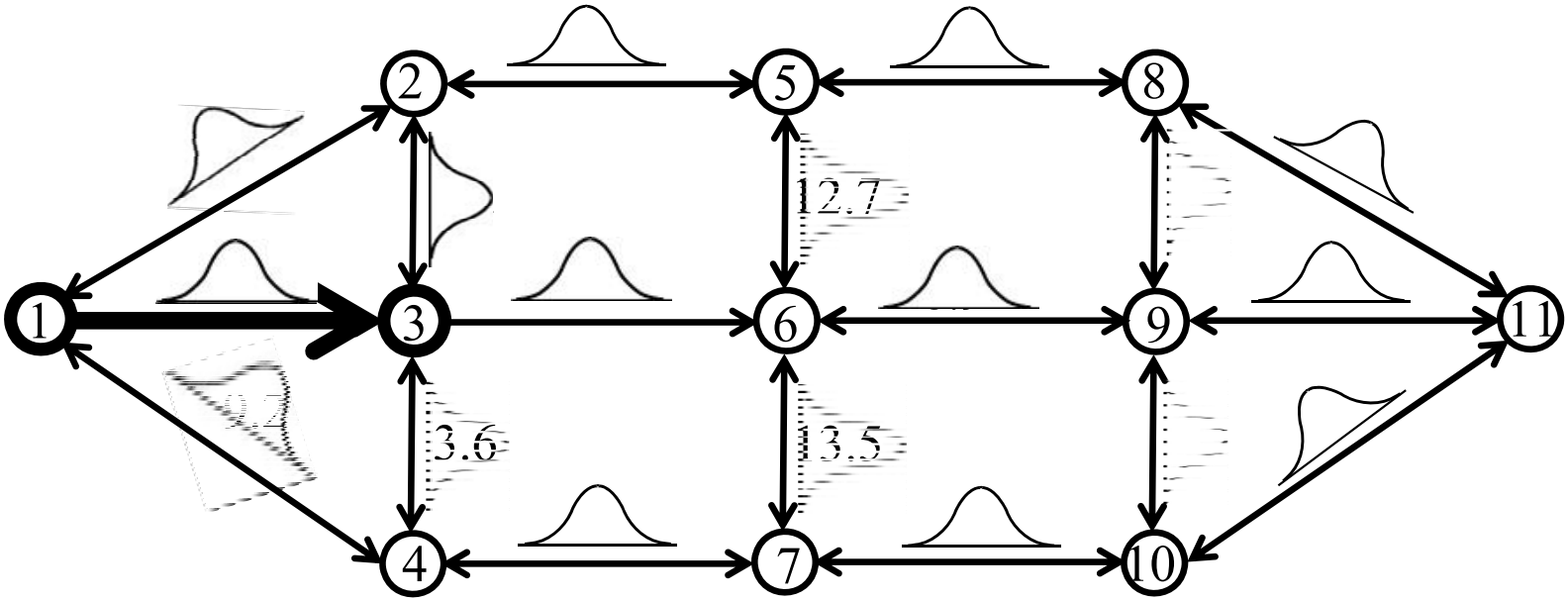
- Illustrating state variables
  - » A deterministic graph



$$S_t = (N_t) = 6$$

The state variable

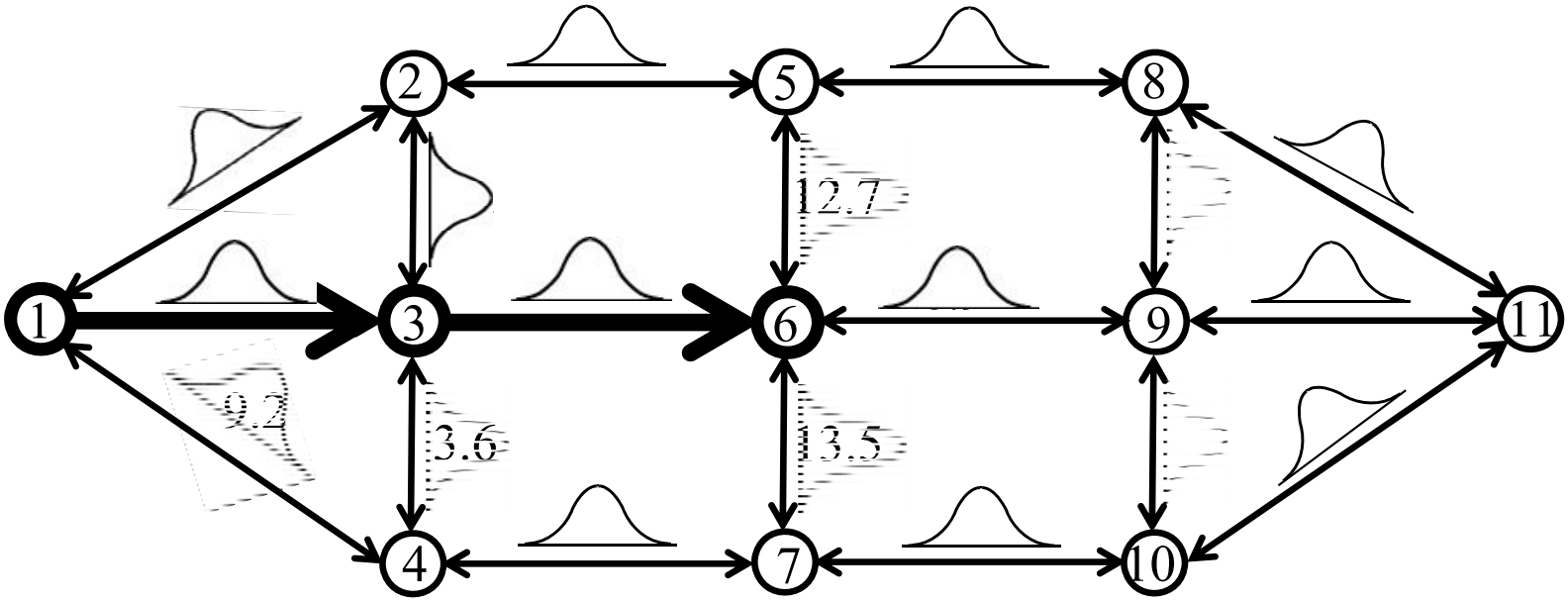
- Illustrating state variables
  - » A stochastic graph



$S_t = ?$

The state variable

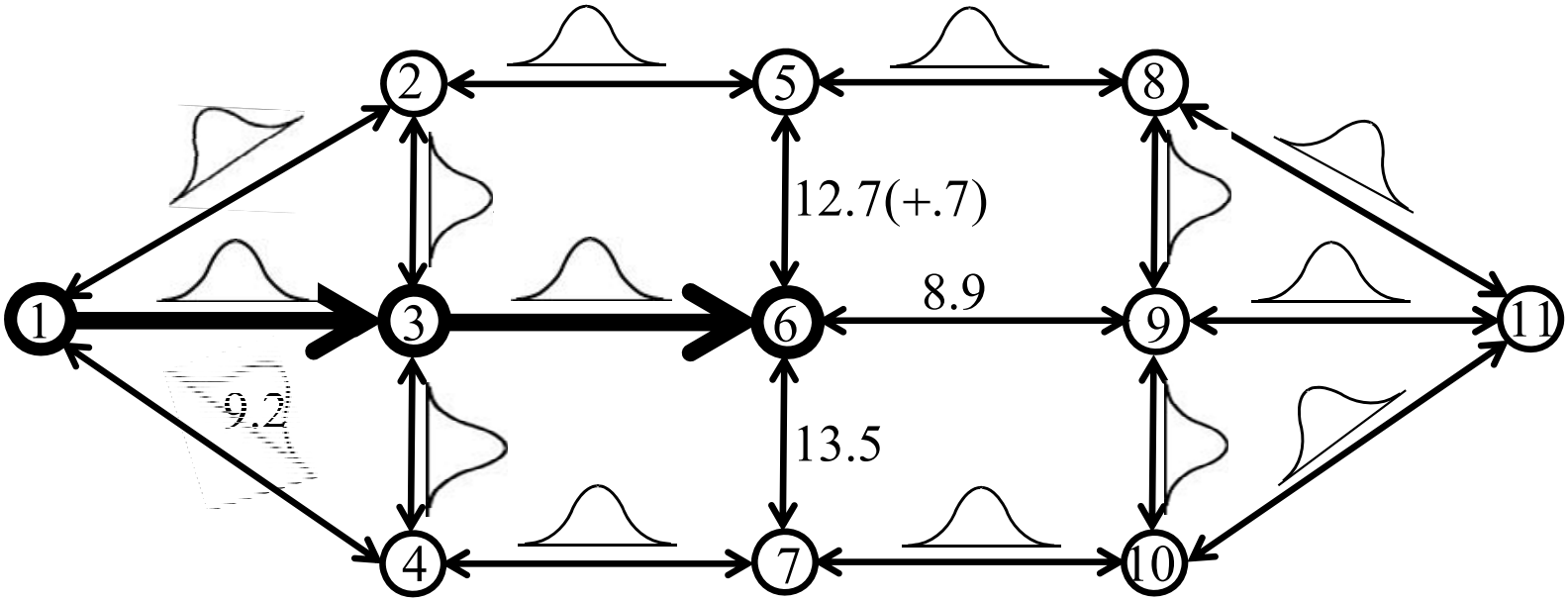
- Illustrating state variables
  - » A stochastic graph



$$S_t = \left(\underbrace{N_t}_{R_t}, \underbrace{\left(c_{t,N_t,j} \right)_j}_{I_t} \right) = \left(6, \left(12.7, 8.9, 13.5 \right) \right)$$

The state variable

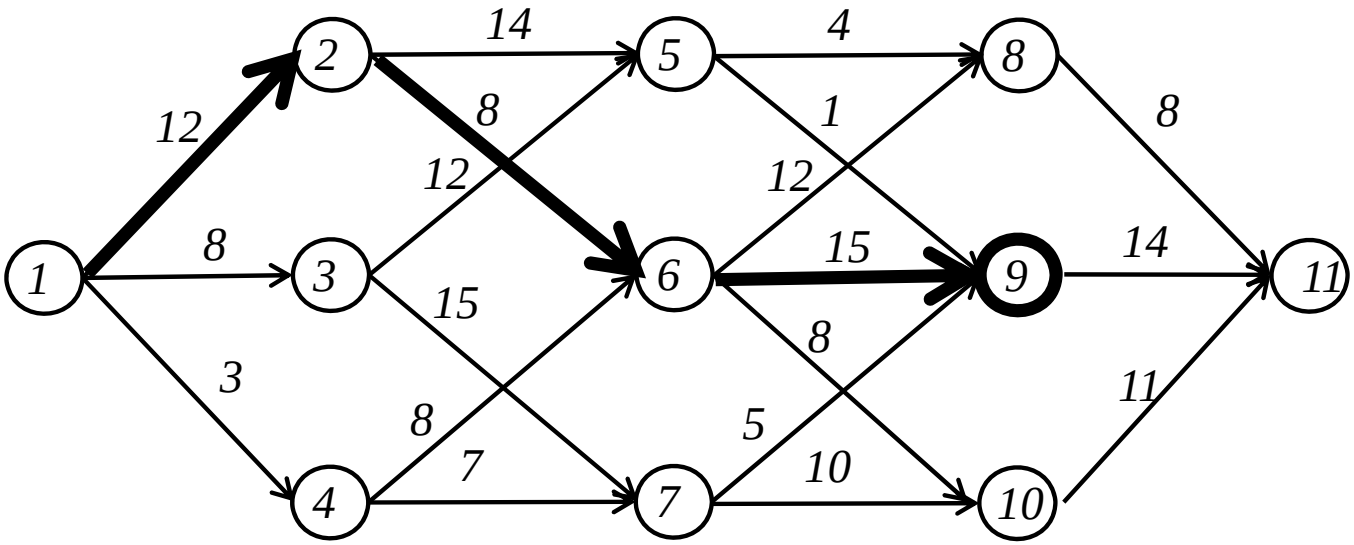
- Illustrating state variables
  - » A stochastic graph with left turn penalties



$$S_t = \left(\underbrace{N_t}_{R_t}, \underbrace{\left(c_{t,N_t,j} \right)_j, N_{t-1}}_{I_t} \right) = \left(6, \left(12, 7, 8.9, 13.5 \right), 3 \right)$$

The state variable

- Variant of problem in Puterman (2005):
  - » Find best path from 1 to 11 that minimizes the *second highest arc cost* along the path:

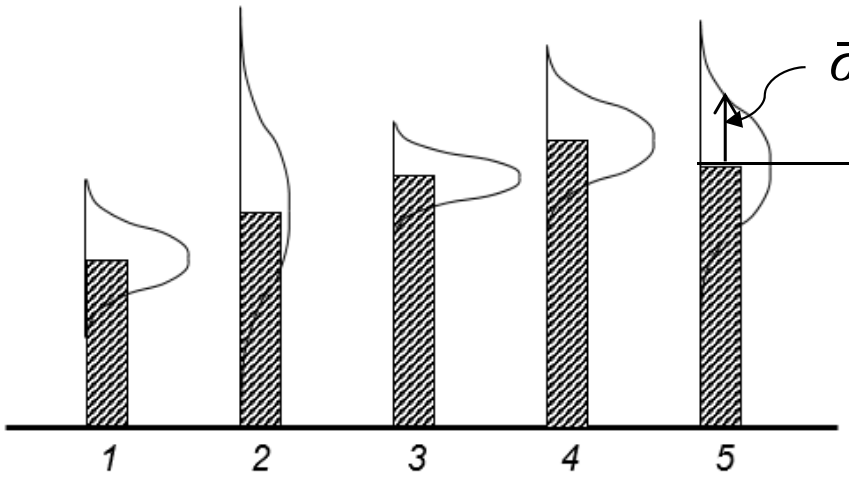


  - » If the traveler is at node 9, what is her state?

$$S_t = (N_t, \text{highest}, \text{second highest}) = (9,15,12)$$

The state variable

- Bellman's equation for our learning problem
  - » A belief state



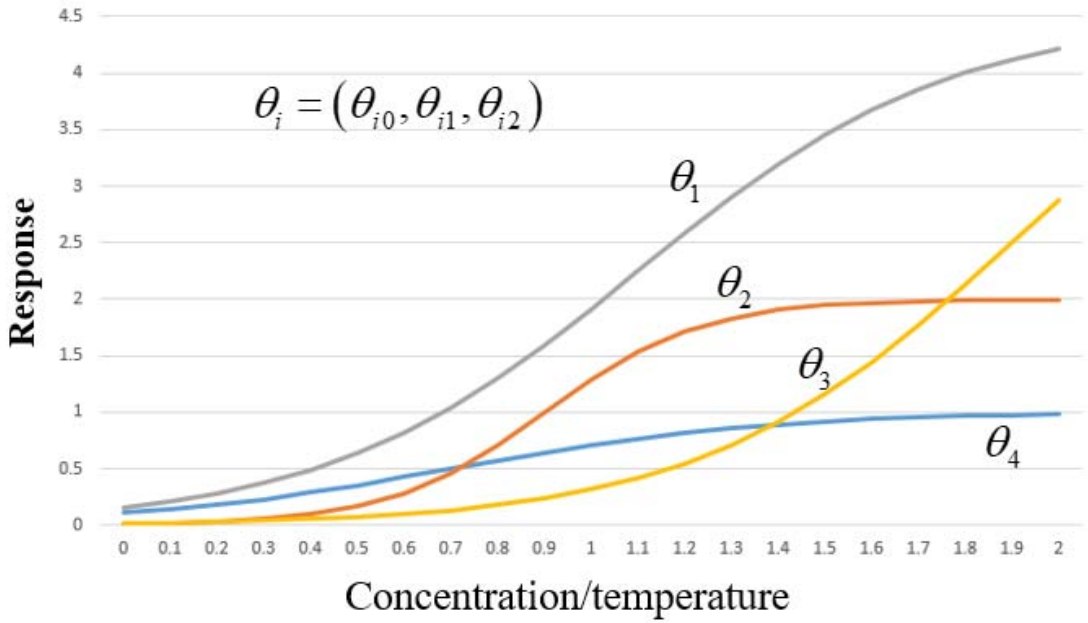
$$\beta_5^n = \frac{1}{\bar{\sigma}_5^{2,n}}$$

  - » State variable (belief state)
    - $S^n = (\bar{\mu}_x^n, \beta_x^n)_{x \in X}$
  - » Bellman equation

$$V(S) = \max_x \left(C(S,x) + \mathbb{E}_W V(S^M(S,x,W))\right).$$

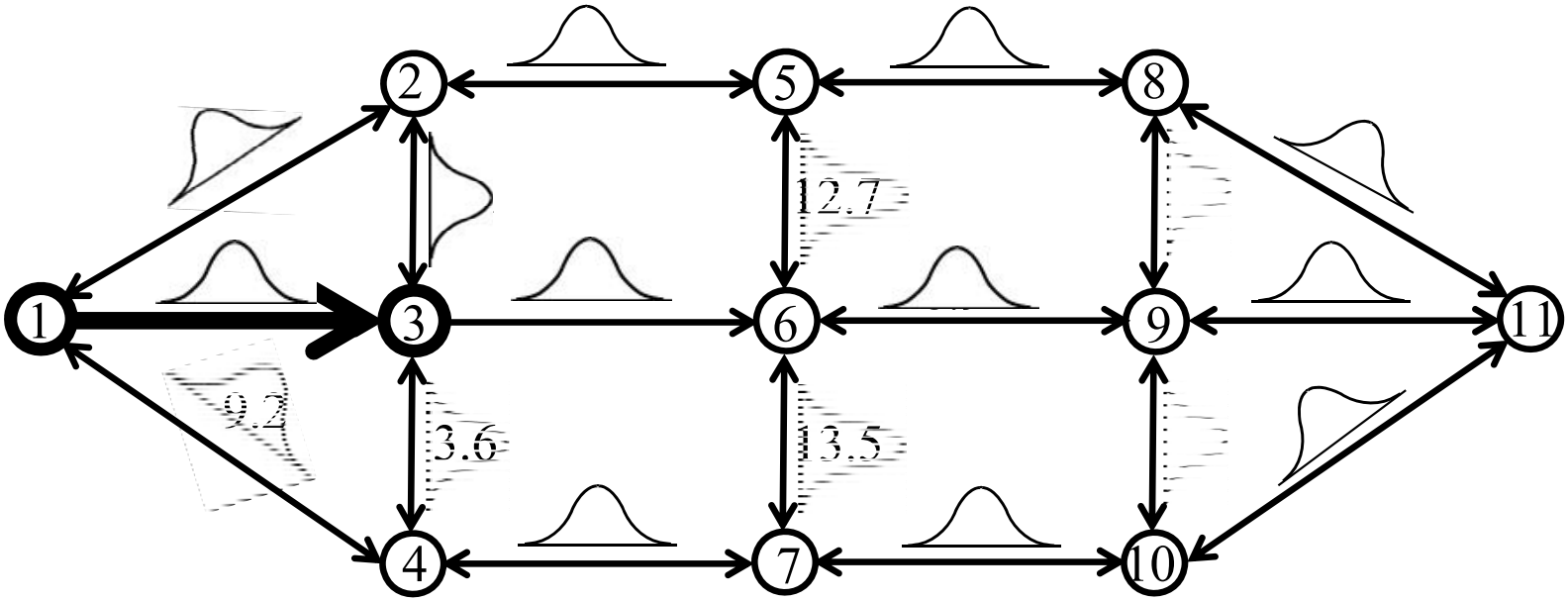
The state variable

- Belief states
  - » Lookup table – Frequentist or Bayesian
    - Independent beliefs - $S^n = (\bar{\mu}_x^n, \beta_x^n)_{x \in X}$
    - Correlated beliefs $S^n = (\bar{\mu}^n, \Sigma^n)$
  - » Parametric with sampled beliefs
    - $S^n = (p_k^n)_{k=1}^K$ where $p_k^n = Prob(\theta = \theta^k)$



The state variable

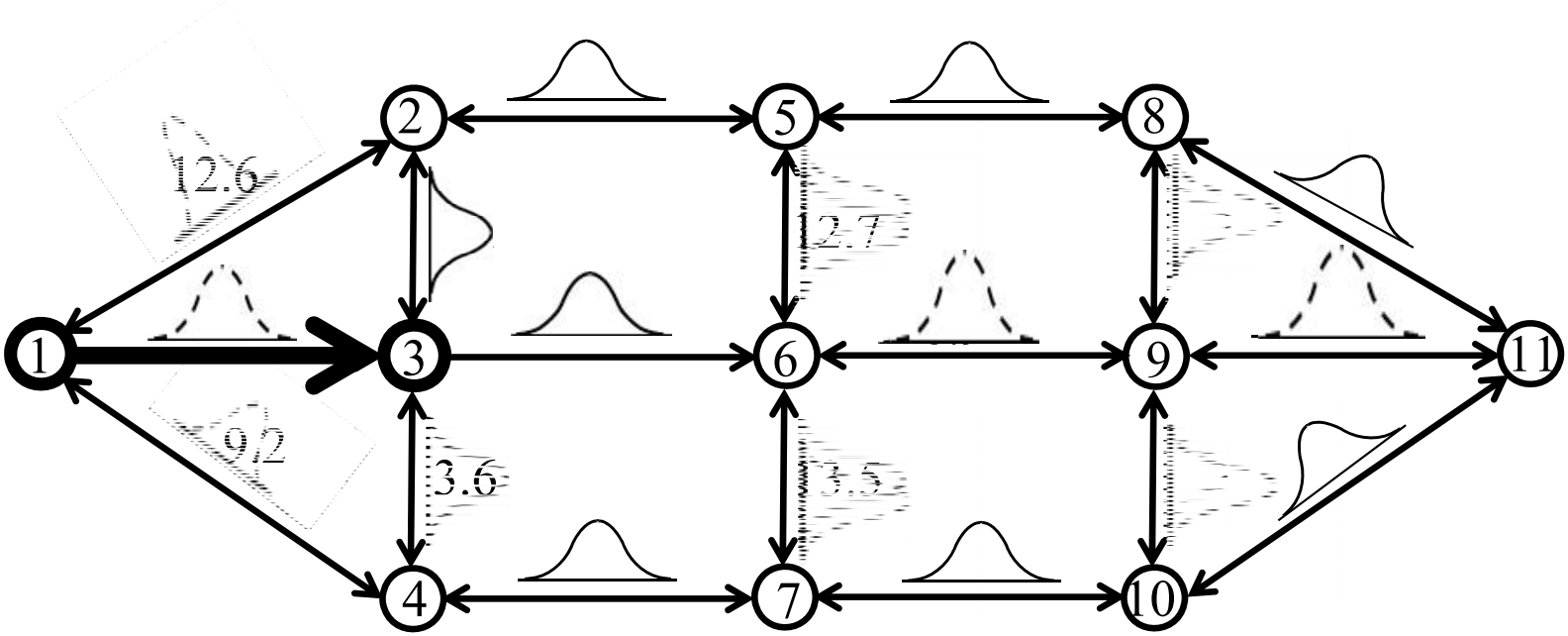
- Illustrating state variables
  - » A stochastic graph with generalized learning



$$S_t = ?$$

The state variable

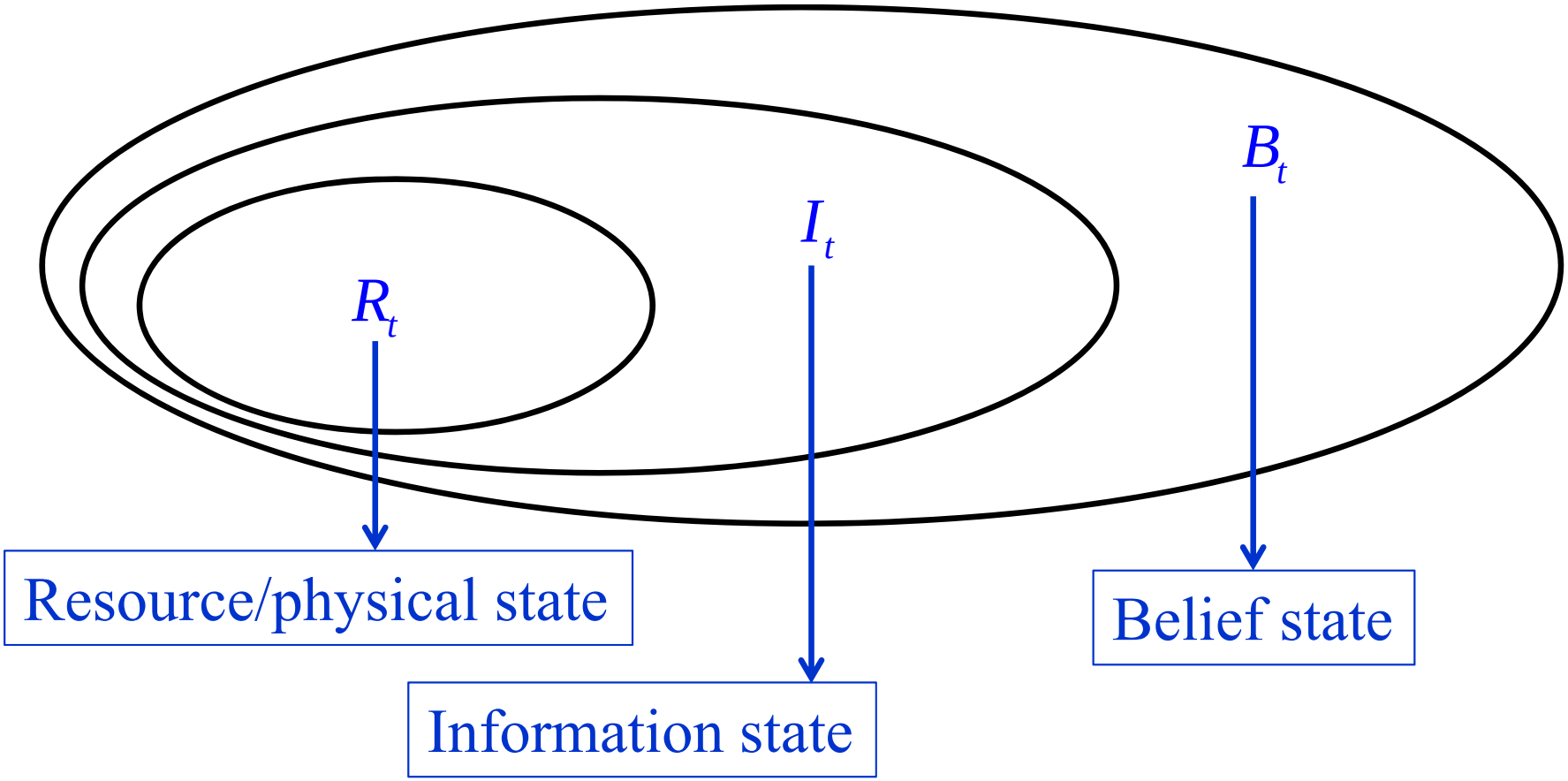
- Illustrating state variables
  - » A stochastic graph with generalized learning



$$S_t = \Big(\underbrace{N_t}_{R_t}, \underbrace{\big(c_{t,N_t,j}\big)_j}_{I_t}, \underbrace{\quad}_{B_t} \Big)$$

The state variable

- Classes of state variables



The state variable

- Notes:
  - » The distinction between R_t, I_t, and B_t is imprecise. After all, they are three types of information.
  - » There are communities (e.g. operations research) that have a tendency to interpret "physical state" as "the state" (the graph example is a good illustration).
  - » So, we can think of R_t as just a special case of I_t.
  - » The belief B_t is the parameters of a probability distribution about parameters we do not know perfectly. But we can think of I_t (and therefore R_t) as simply point estimates of parameters we know perfectly.

The state variable

- My definition of a state variable:

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

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

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

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

The state variable

- Email received March 15, 2019:

Hello Warren,

As I was cleaning up my email folders at the end of the quarter, I realized that I had drafted an email but had not sent it. My apologies, as I appreciate you reaching out and for Charlie connecting us.

I just finished up a stochastic optimization course taught by John Birge, and your work came up several times. This was in reference to the overhaul of the freight industry in addition to the potential for state variables to be properly modified to fit under the Markovian framework. My research is moving more towards stochastic programming, as I look to develop medical center stocking policies in the presence of regular transportation outages. Thanks again for passing on the jungle material and sorry for the long delay.

Best,

addition to the potential for state variables to be properly modified to fit under the Markovian framework. I

The state variable

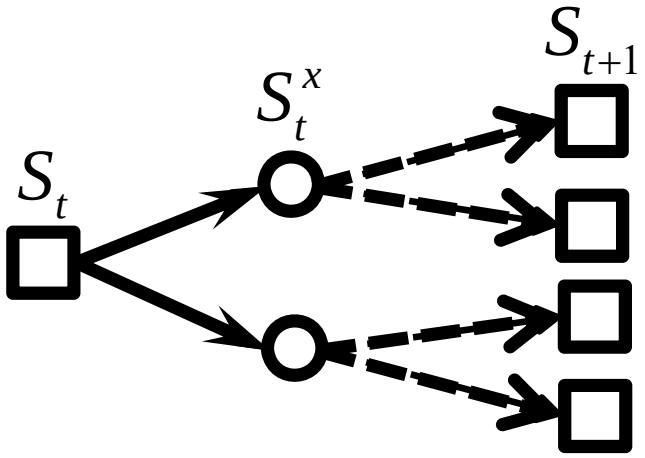
- Pre- and post-decision states
  - » The "pre-decision" state variable:
    - S_t = The information required to make a decision x_t
    - Same as a "decision node" in a decision tree.
  - » The "post-decision" state variable:
    - S_t^x = The state of what we know immediately after we make a decision.
    - Same as an "outcome node" in a decision tree. Also known as "end of period state" or "after state".
  - » The information and decision sequence
    - $(S_0, x_0, S_0^x, W_1, S_1, x_1, S_1^x, W_2, S_2, \ldots S_t, x_t, S_t^x, W_{t+1}, \ldots)$

The state variable

- Representations of the post-decision state:
  - » Decision trees:

$$S_t^x = S^{M,x}\left(S_t, x_t\right)$$

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



  - » Q-learning:

$$S_t^x = \left(S_t, x_t\right)$$ State-action pair

  - » Transition function with expectation:

$$S_t^x = S^M\left(S_t, x_t, \bar{W}_{t,t+1}\right) \qquad \bar{W}_{t,t+1} = \text{Forecast of } W_{t+1} \text{ at time } t.$$

The state variable

- Resource states (single layer)
  - » R_t=Amount of resource on hand at time t.
  - » R_{ti} =Amount of resource of type i on hand at time t.
  - » R_{ta} =Resources with attribute vector a available at time t.
- Two layer problems:
  - » Supplies and demands (where demands can be held if they are not served)
  - » Uber drivers and riders
  - » Pilots and aircraft
  - » (we won't be touching multi-layer problems in this course).

The post-decision state

- A scalar resource variable:
  - » Our basic inventory equation (water, money, …)

$$R_{t+1} = \max\left\{0, R_t + x_t - \hat{D}_{t+1}\right\}$$

  - » where

R_t = Inventory on hand at time t

x_t = Amount ordered

$\hat{D}_{t+1}$ = Demand in next time period

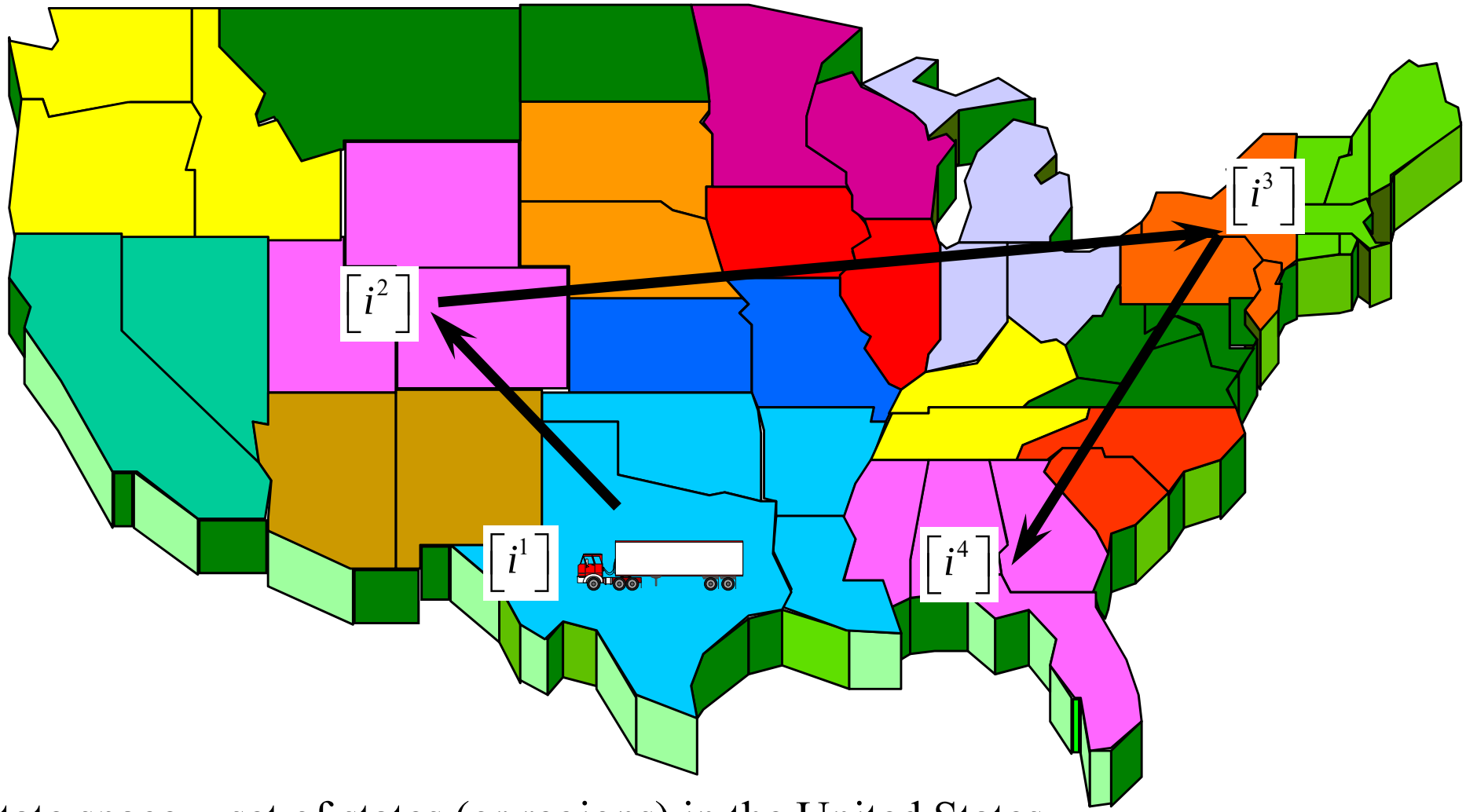
  - » Using pre- and post-decision states:

$$R_t^x = R_t + x_t \qquad \text{Pre- to post-}$$

$$R_{t+1} = \max\left\{R_t^x - \hat{D}_{t+1}\right\} \qquad \text{Post- to pre-}$$

Modeling resources

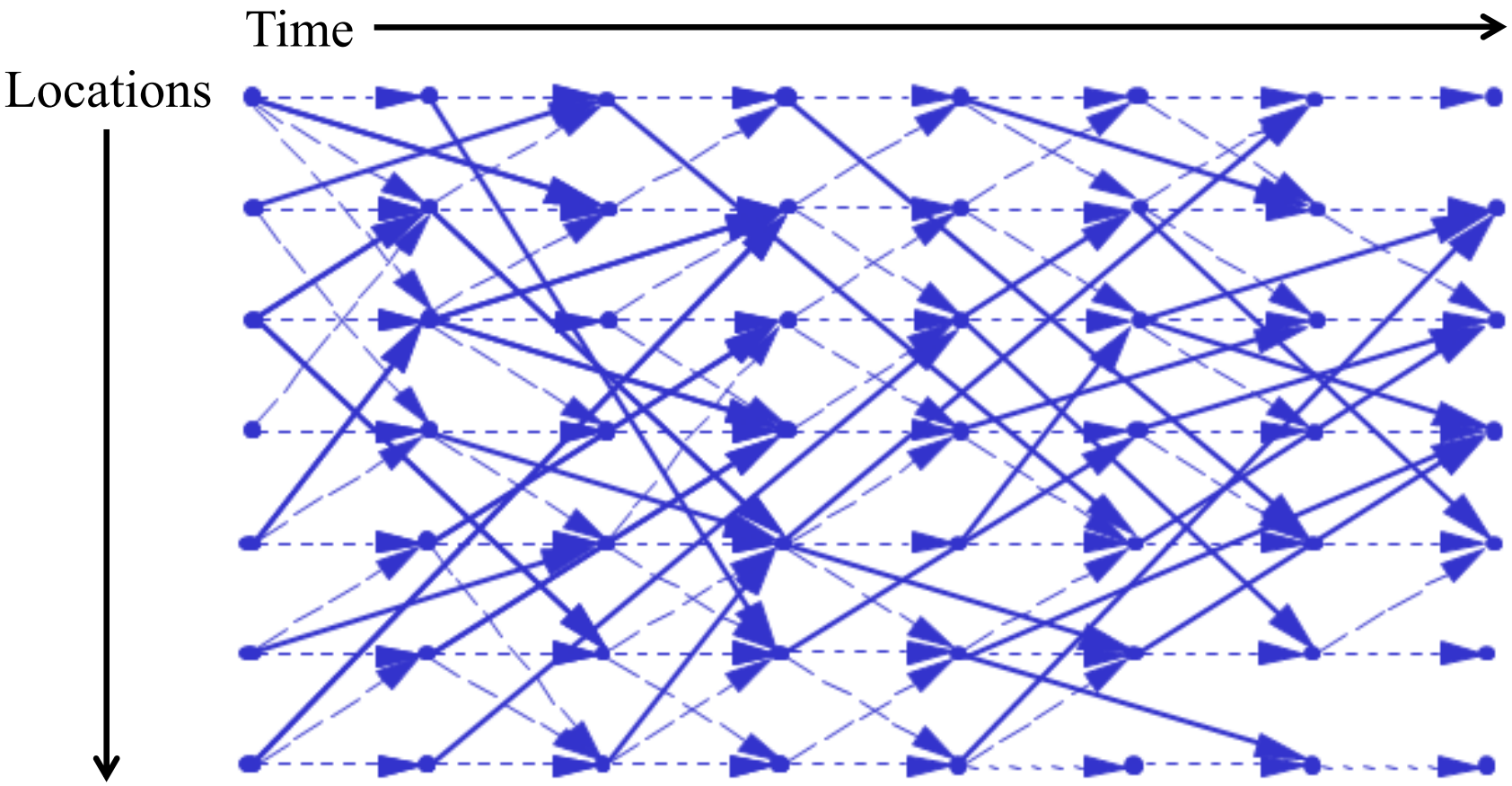
- Single resource type, dynamic attribute
  - » $R_t = (R_{ti})_{i \in I}$



State space = set of states (or regions) in the United States

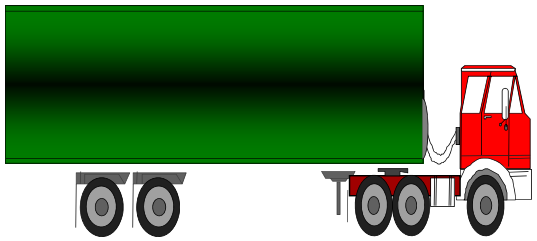
Modeling resources

- Single commodity flow problems (100-1000 rows)

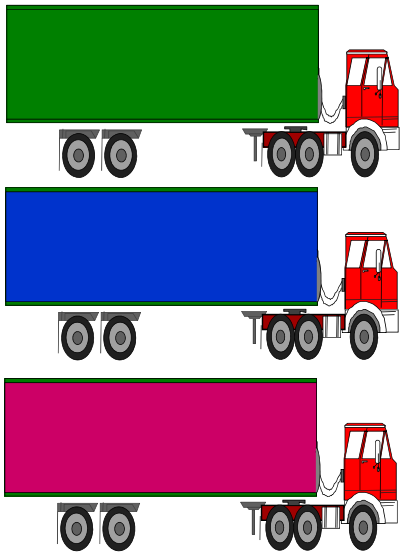


Modeling resources

A single commodity:

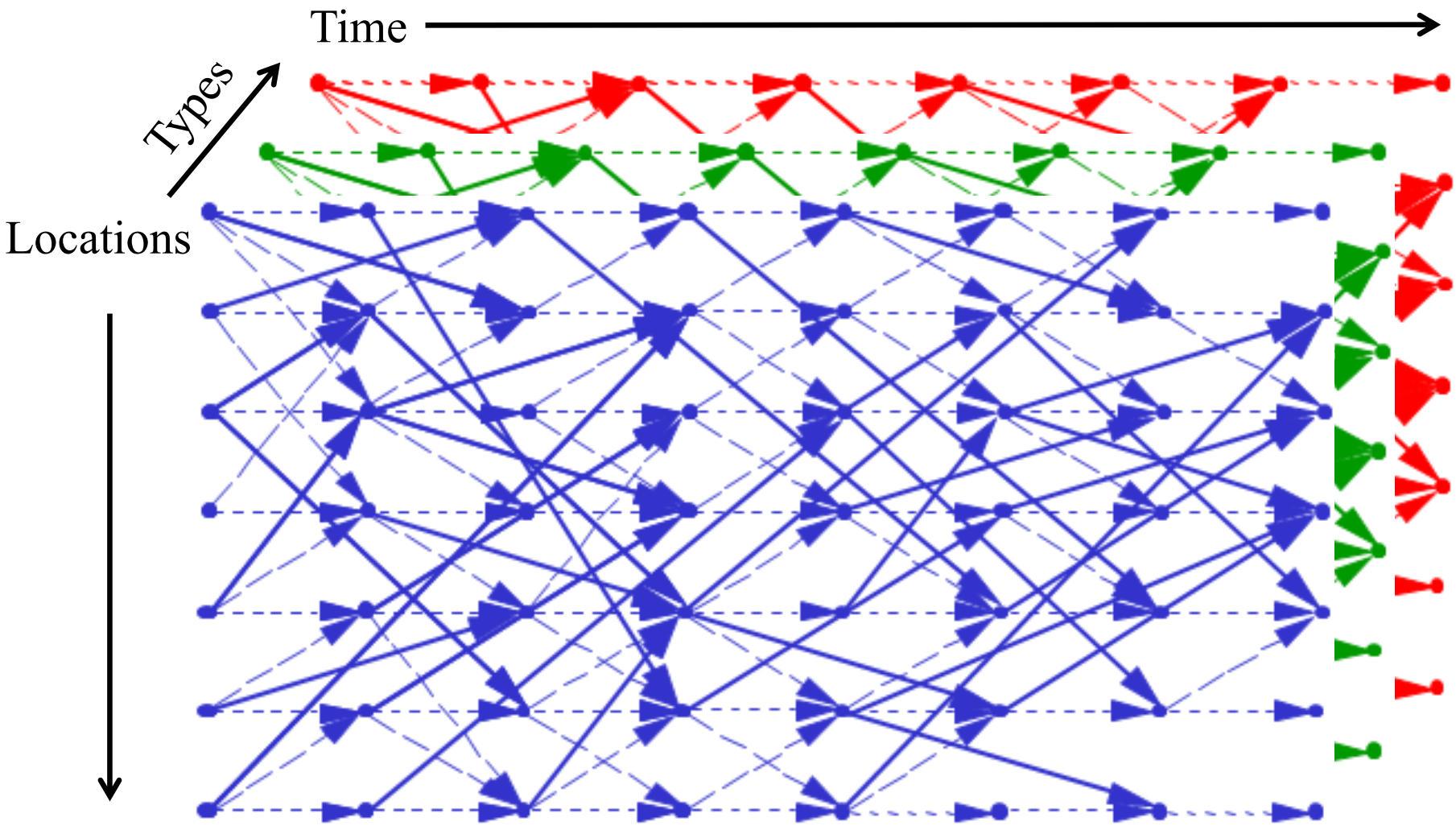


Multiple commodities:



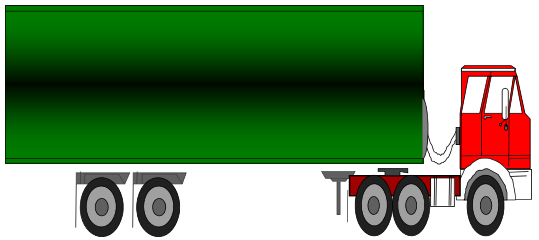
Modeling resources

- Multicommodity flow models

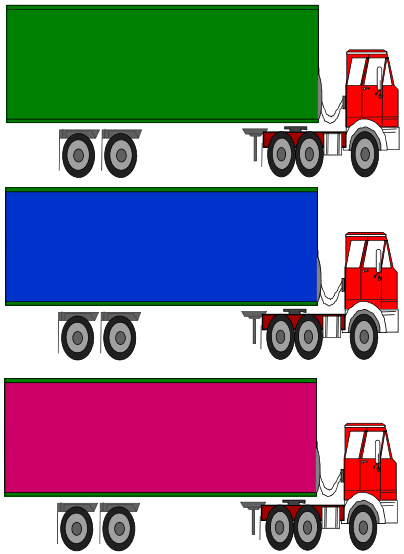


Modeling resources

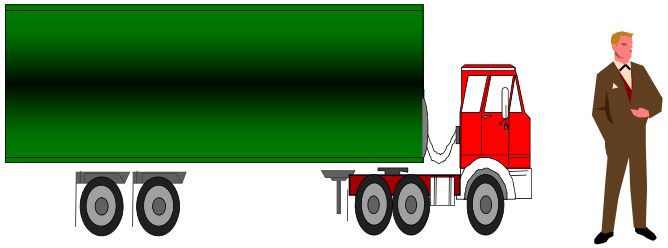
A single commodity:



Multiple commodities:

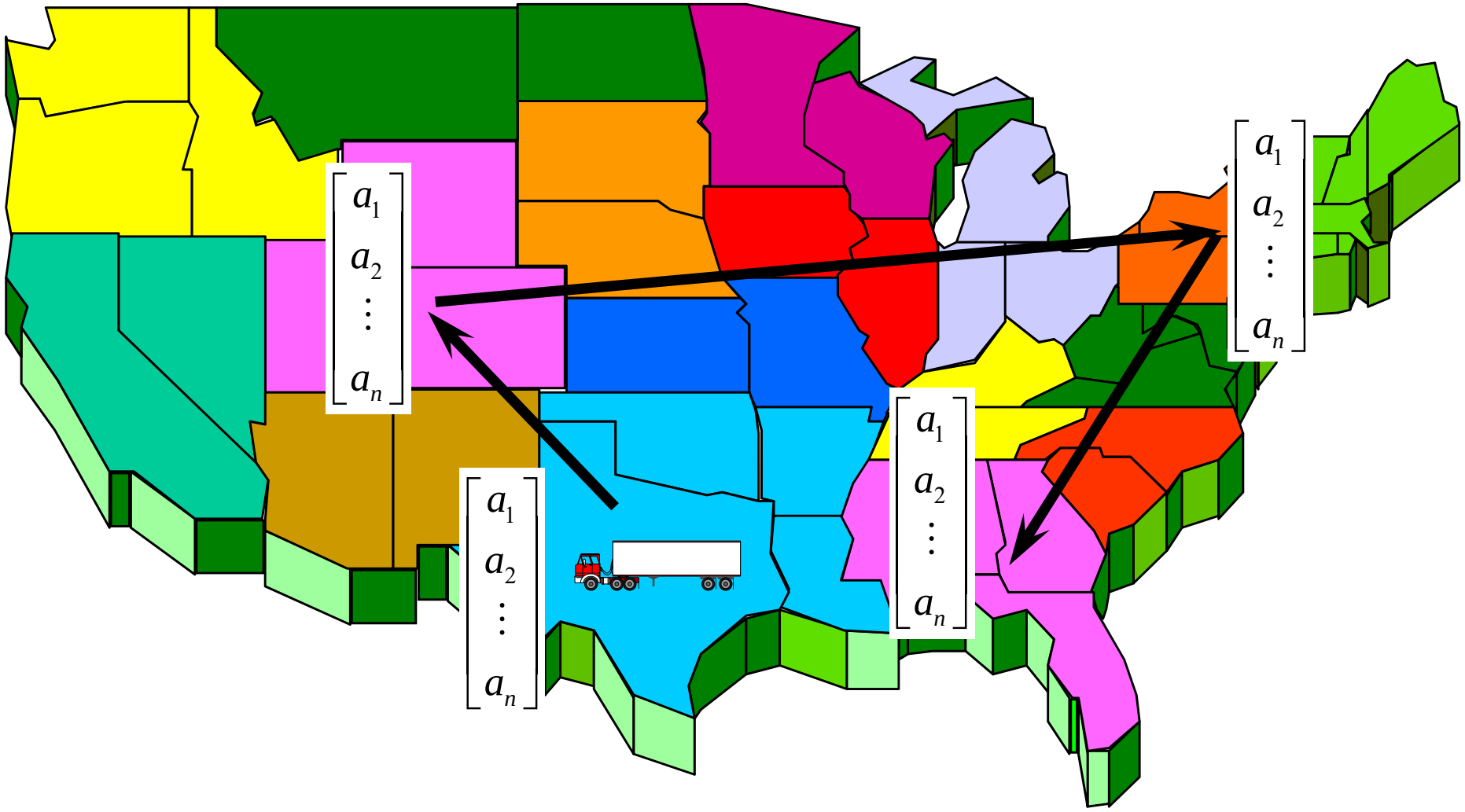


A multiattribute resource:



$$a = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ \vdots \\ a_N \end{bmatrix}$$

Modeling resources



Modeling resources

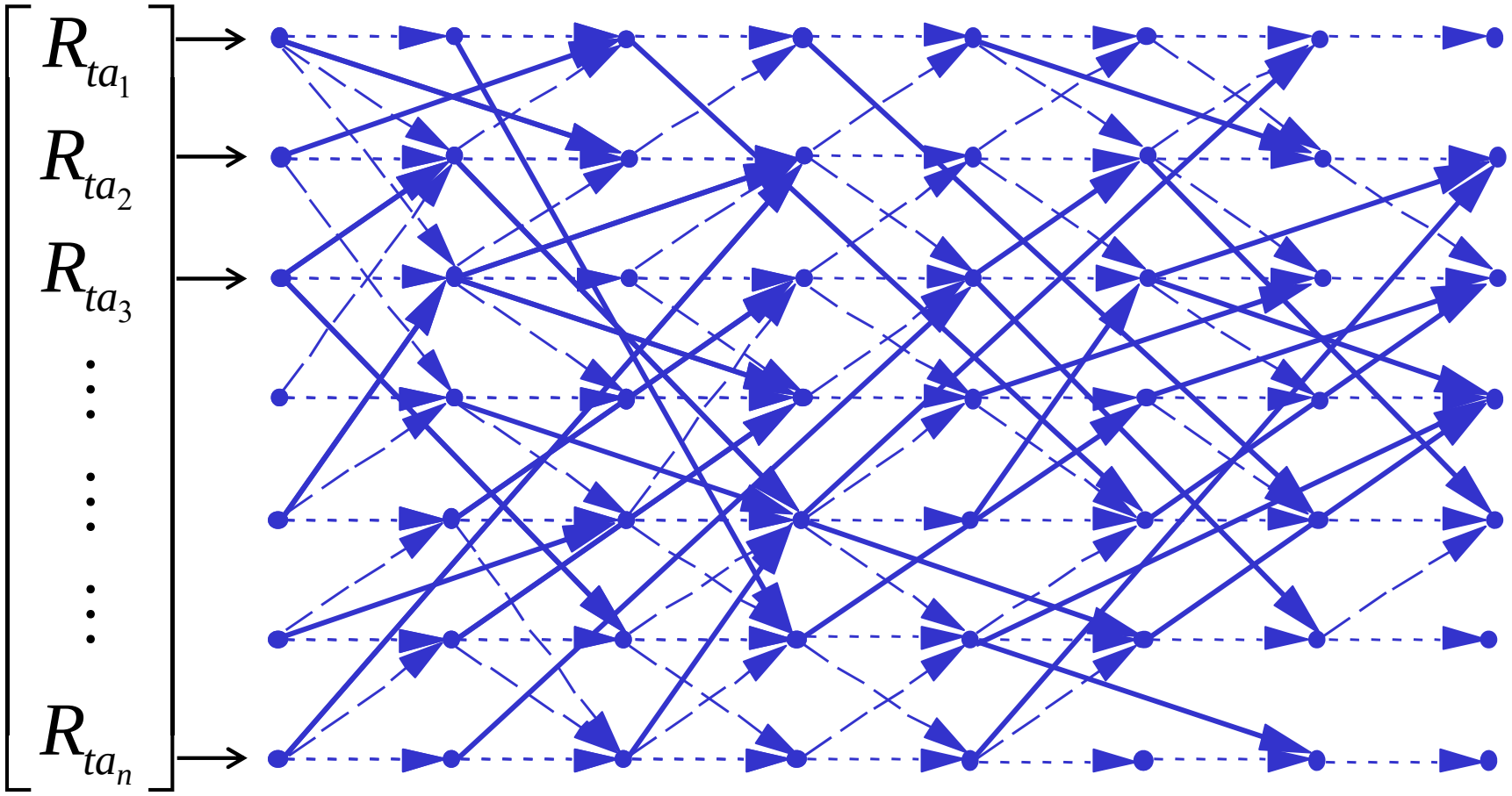
- The evolution of attributes:

$$a = \begin{bmatrix} \text{Time} \\ \text{Location} \end{bmatrix} \begin{bmatrix} \text{Time} \\ \text{Location} \\ \text{Equip type} \end{bmatrix} \begin{bmatrix} \text{Time} \\ \text{Location} \\ \text{Equip type} \\ \text{Time to dest.} \end{bmatrix} \begin{bmatrix} \text{Time} \\ \text{Location} \\ \text{Equip type} \\ \text{Time to dest.} \\ \text{Repair status} \end{bmatrix} \begin{bmatrix} \text{Time} \\ \text{Location} \\ \text{Equip type} \\ \text{Time to dest.} \\ \text{Repair status} \\ \text{Hrs of service} \end{bmatrix}$$

$$|\mathcal{A}| \approx \quad \textit{4,000} \qquad \textit{40,000} \qquad \textit{1,680,000} \qquad \textit{5,040,000} \qquad \textit{50,400,000}$$

Modeling resources

- Multiattribute resource allocation problem



Number of rows equals size of attribute space $\sim 10^6 - 10^{20}$

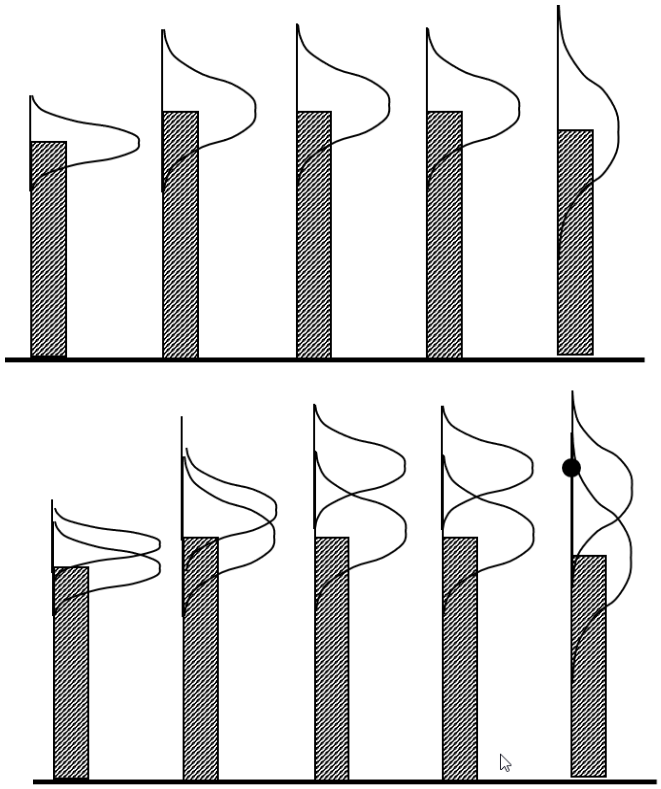
The state variable

- Other information states I_t
  - » The "information state" variable captures the state of any other parameter that we can observe perfectly.
  - » Examples:
    - Weather (e..g temperature, humidity)
    - Prices
    - Economy
    - Traffic conditions
    - Forecasts
  - » The "information state" is simply any information about perfectly observable parameters that we do not model as a resource state.

The state variable

- Belief states:
  - » Look up tables, Bayesian belief, independent beliefs:

$$B^n = (\bar{\mu}_x^n, \beta_x^n)_{x \in X}$$

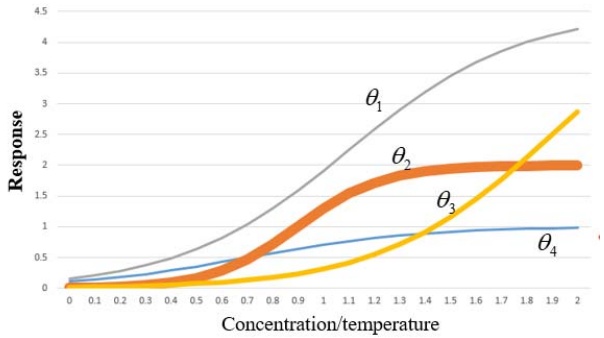


  - » Correlated beliefs

$$B^n = (\bar{\mu}^n, \Sigma^n)$$

  - » Sampled beliefs, nonlinear model $f(x|\theta_k)$:

$$B^n = (p_k^n)_{k=1}^K, p_k^n = Prob[\theta = \theta_k]$$

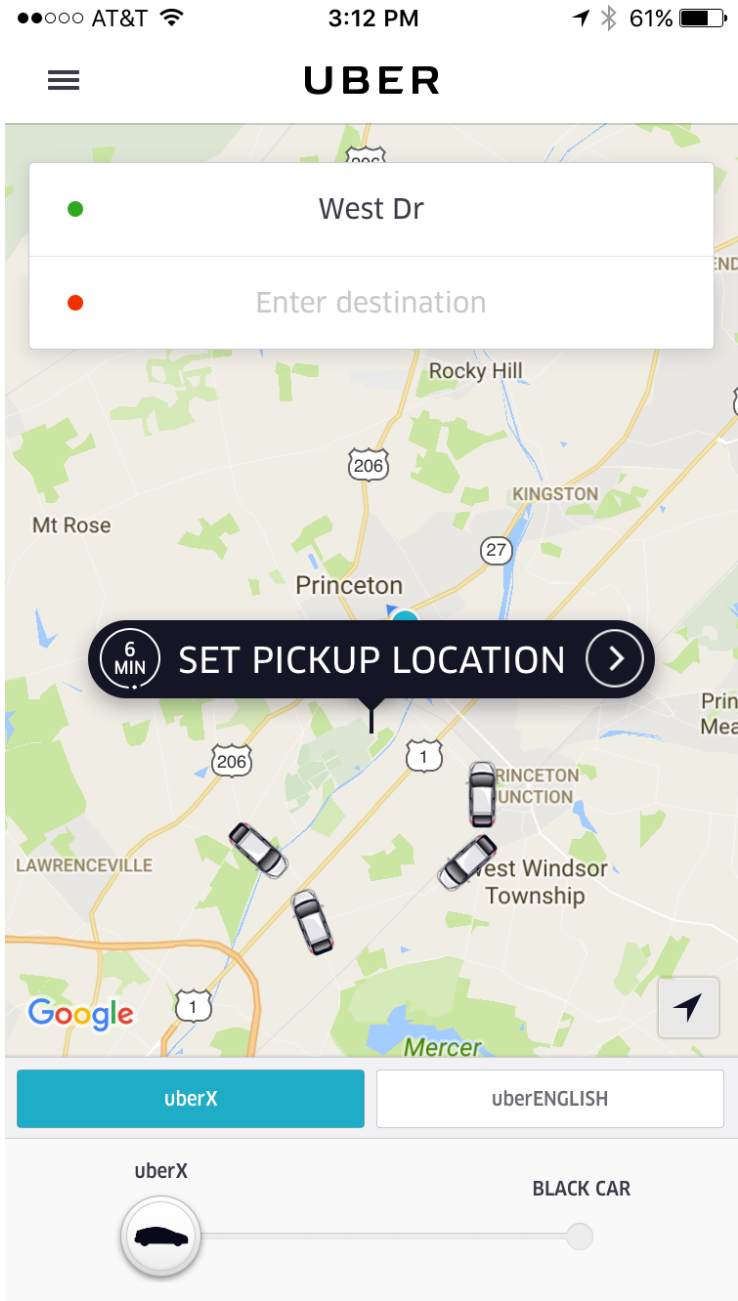


The state variable

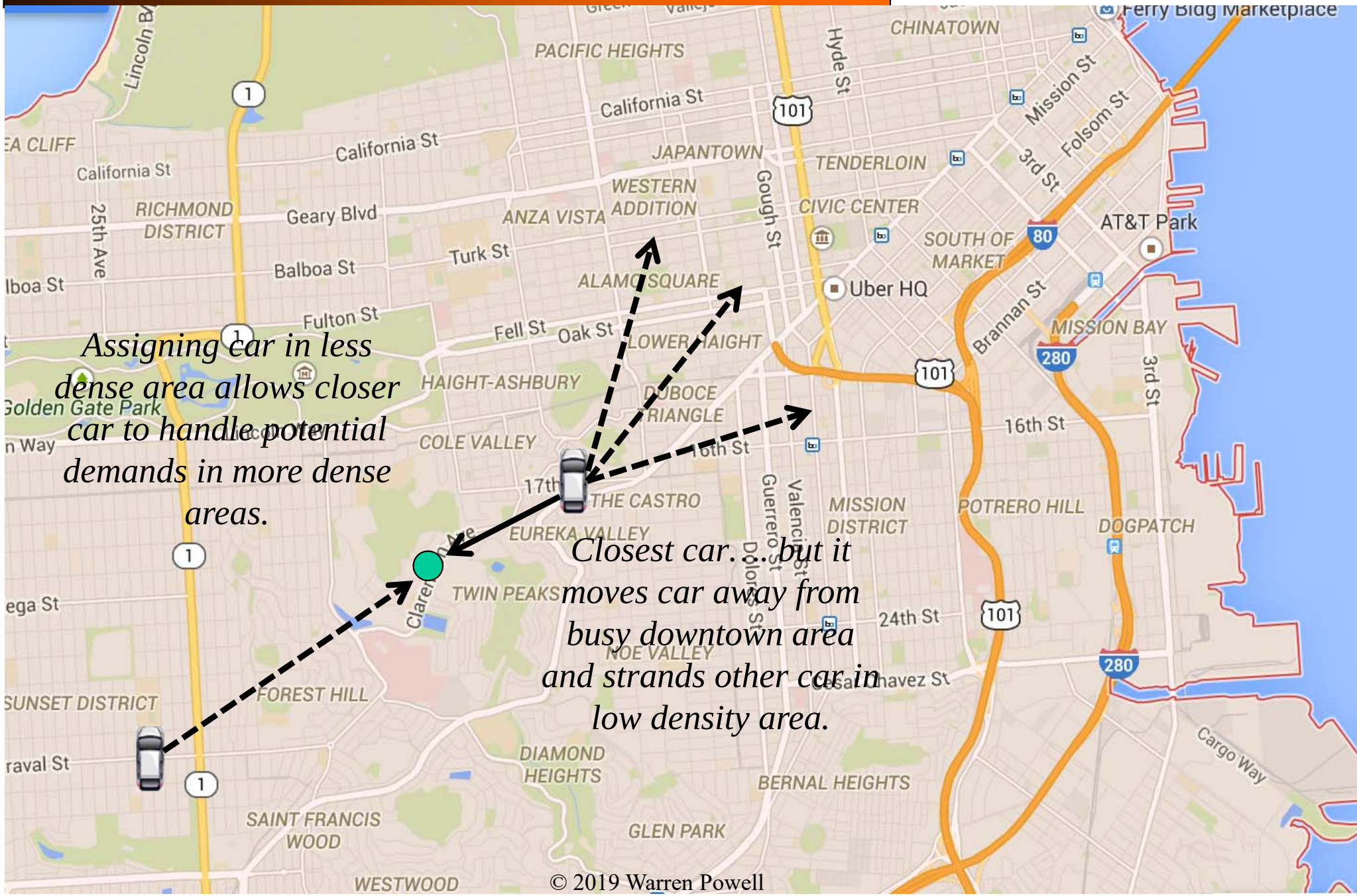
- Multi-layer problems
  - » Two-layers
    - Supplies and demands – Note that this is only a two-layer problem if both supplies and demands are held if they are not used or served.
    - Uber drivers and customers – if we step forward in 15 minute increments, and assume that customers not served in the first time period are lost, then this is a one-layer problem.

Fleet management

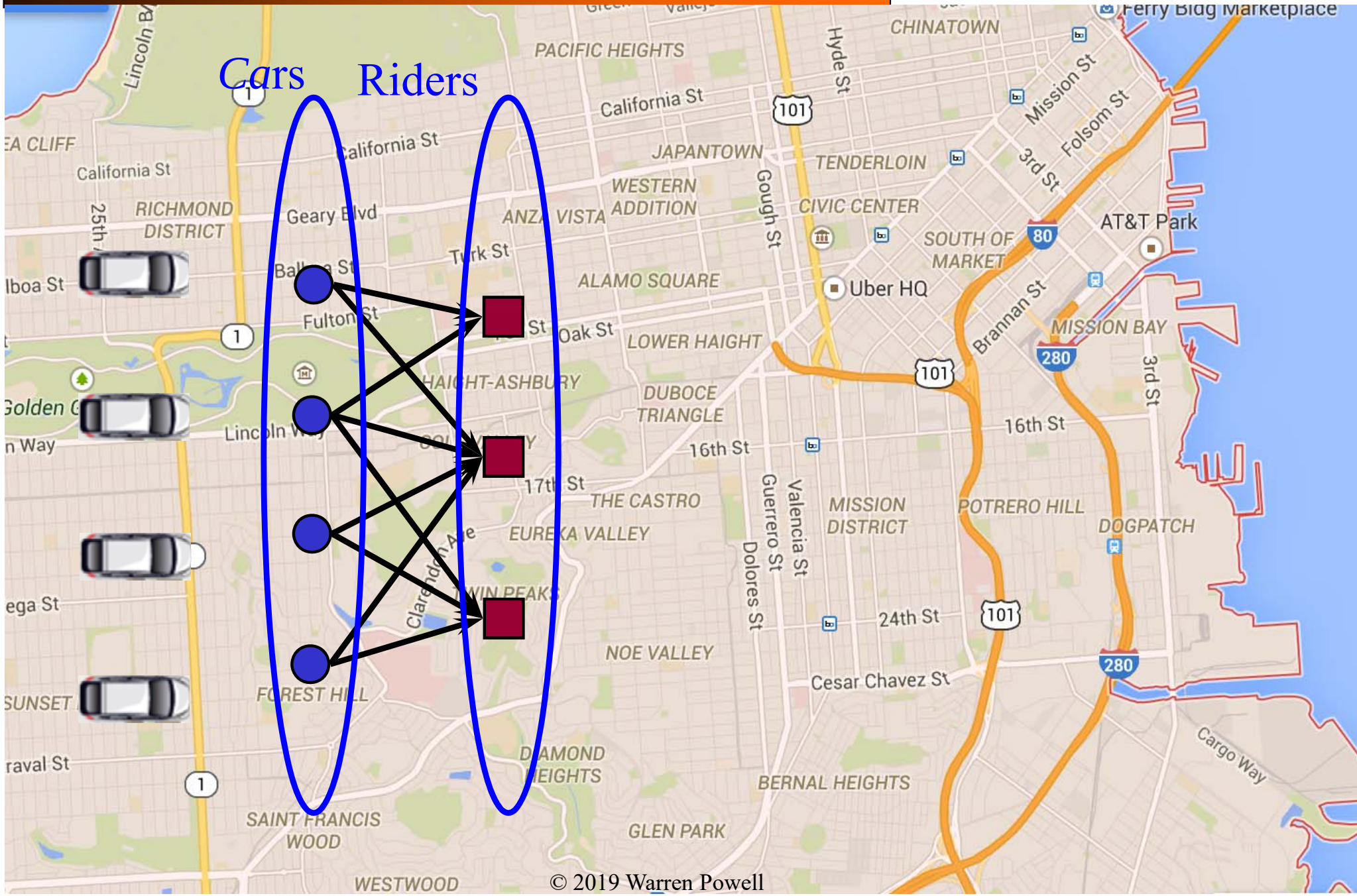
- Uber
  - » Provides real-time, on-demand transportation.
  - » Drivers are encouraged to enter or leave the system using pricing signals and informational guidance.
- Decisions:
  - » How to price to get the right balance of drivers relative to customers.
  - » Assigning and routing drivers to manage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



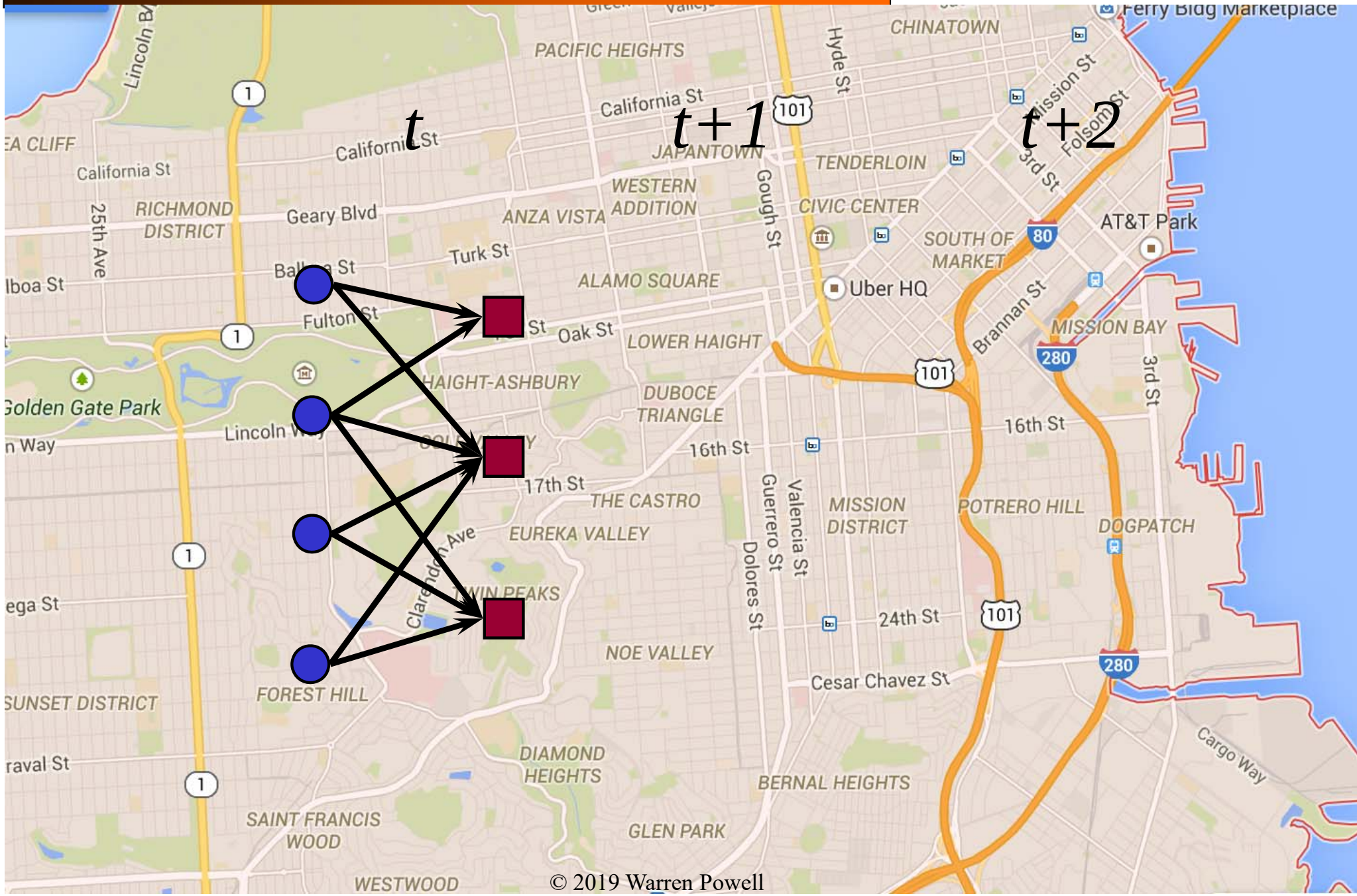
Fleet management



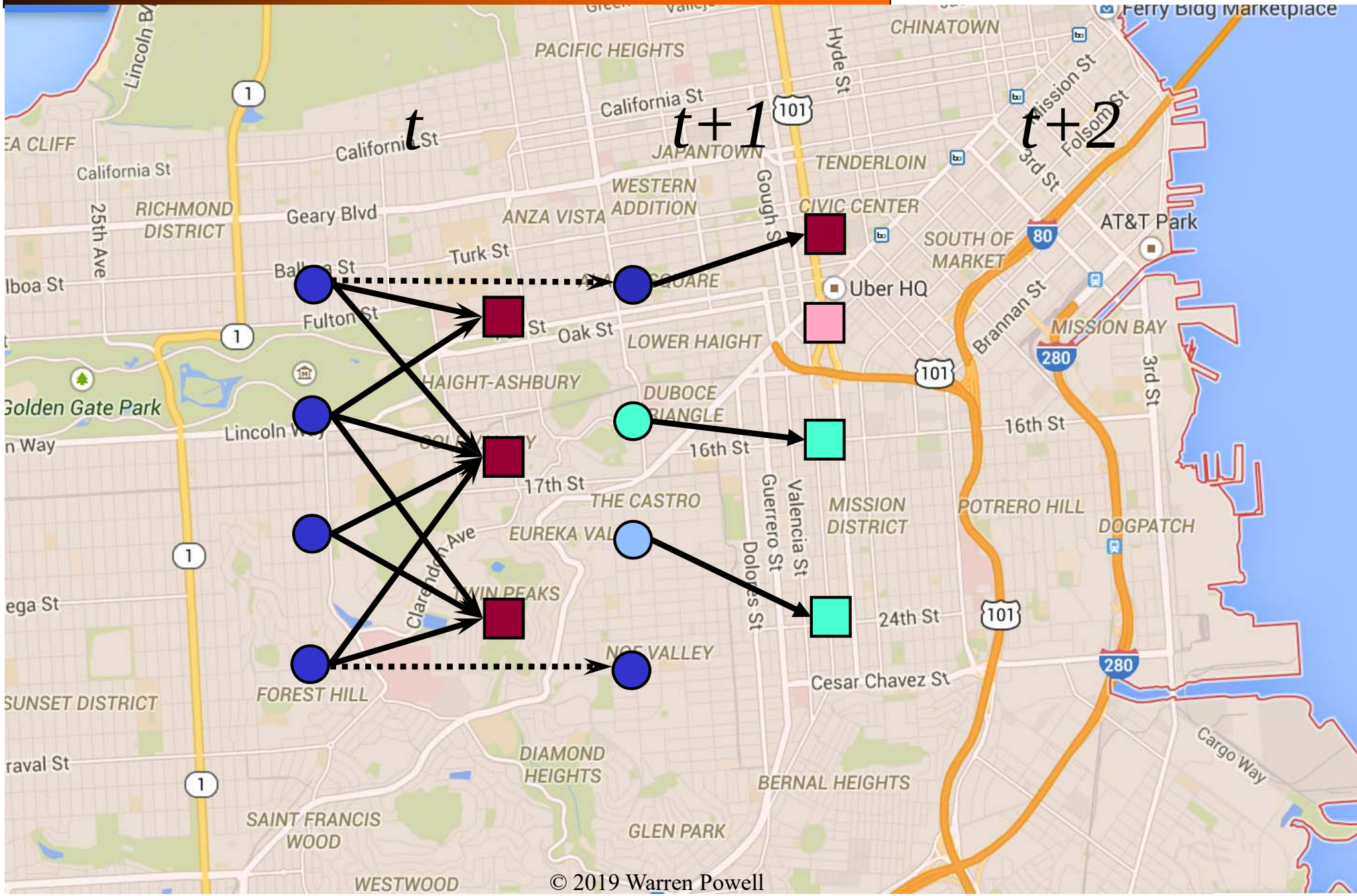
Fleet management



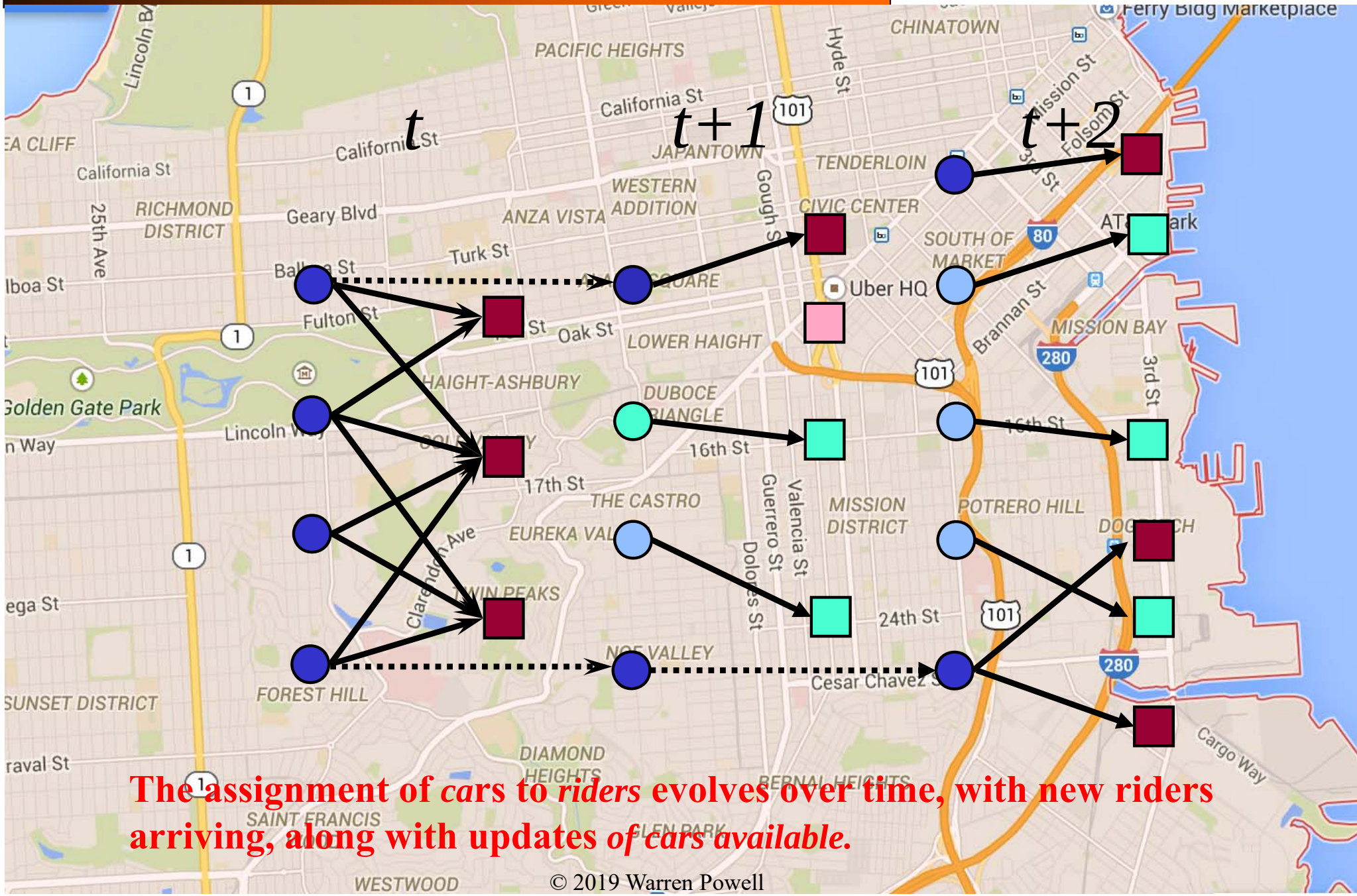
Fleet management



Fleet management



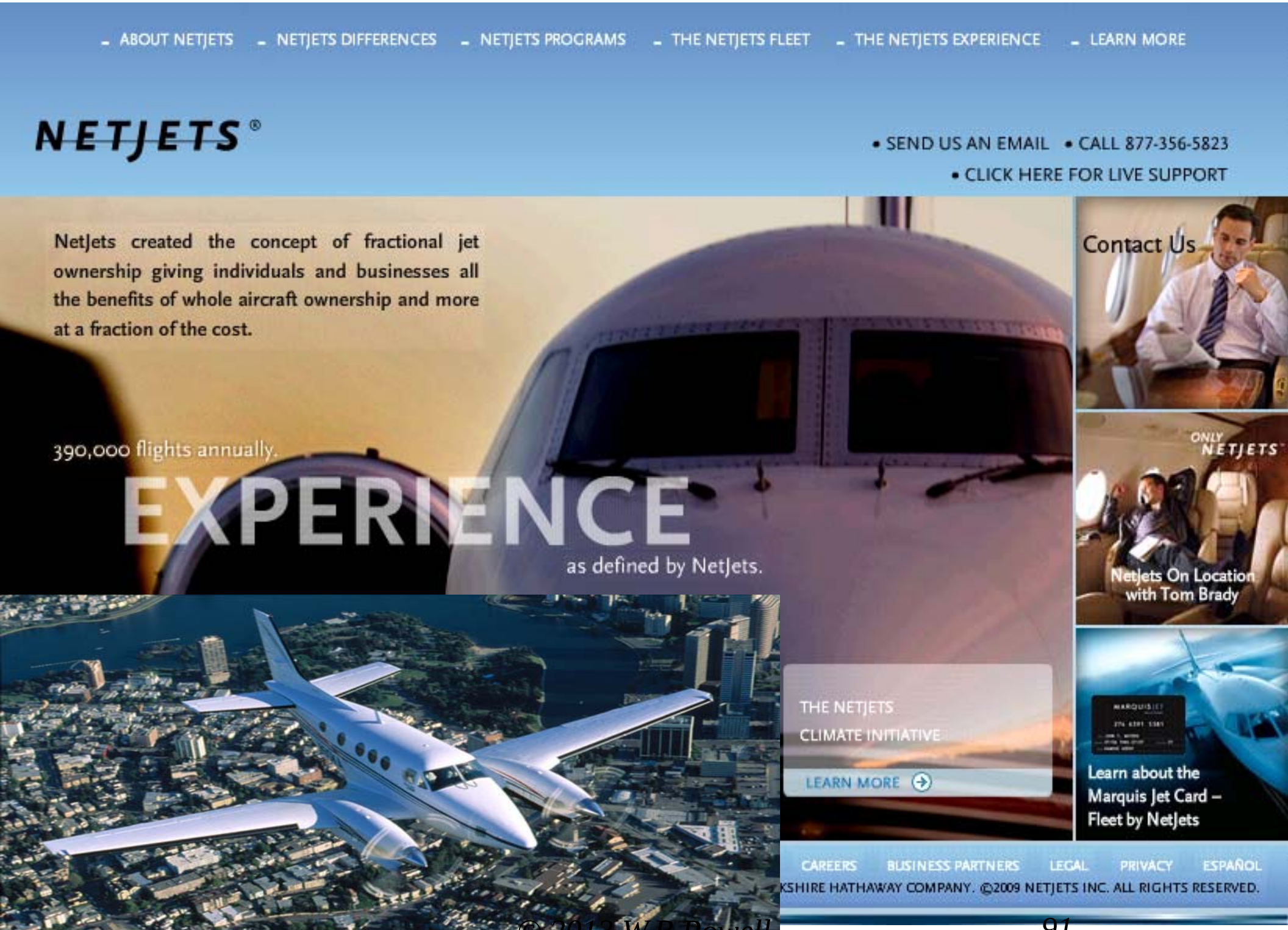
Fleet management



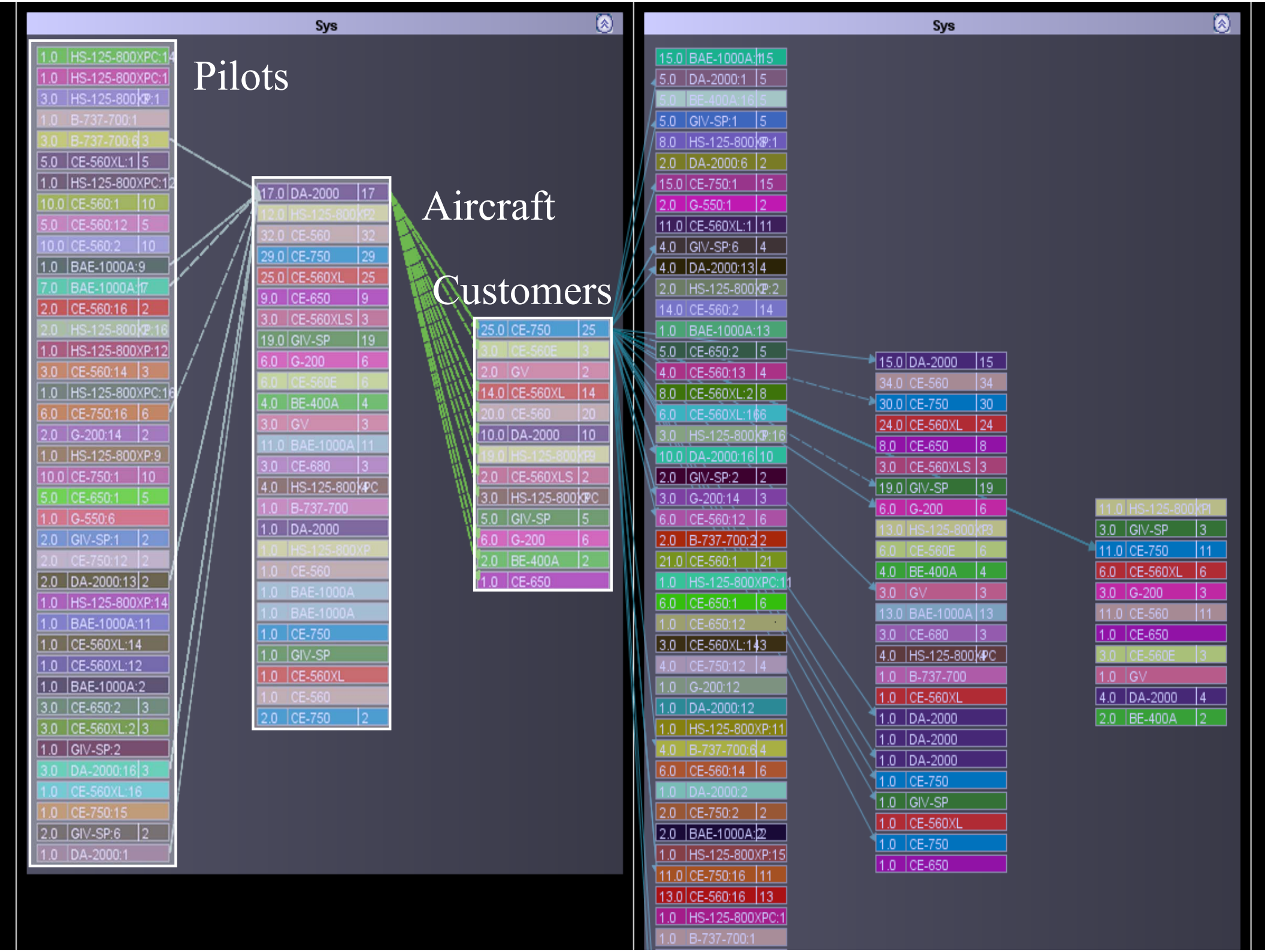
The assignment of *cars* to *riders* evolves over time, with new riders arriving, along with updates *of cars available.*

The state variable

- Multi-layer problems
  - » Three-layers - Application at Netjets. Assigning
    - Pilots
    - Aircraft
    - Customers
  - » All three layers involved complex (multiattribute) resources.




© Netjets


Sys
Pilots
Aircraft
Customers
Sys

State variables

- Notes:
  - » There is tremendous confusion in the stochastic optimization literature about state variables.
  - » One of the worst misconceptions is that if the state variable is multidimensional, then you have the "*curse of dimensionality*."
  - » The curse of dimensionality *only* arises when you use lookup table representations of functions that depend on the state variable.
  - » Most of the examples above have extremely large (or infinite) state spaces, but we will still be able to find practical solutions to these problems.

State variables

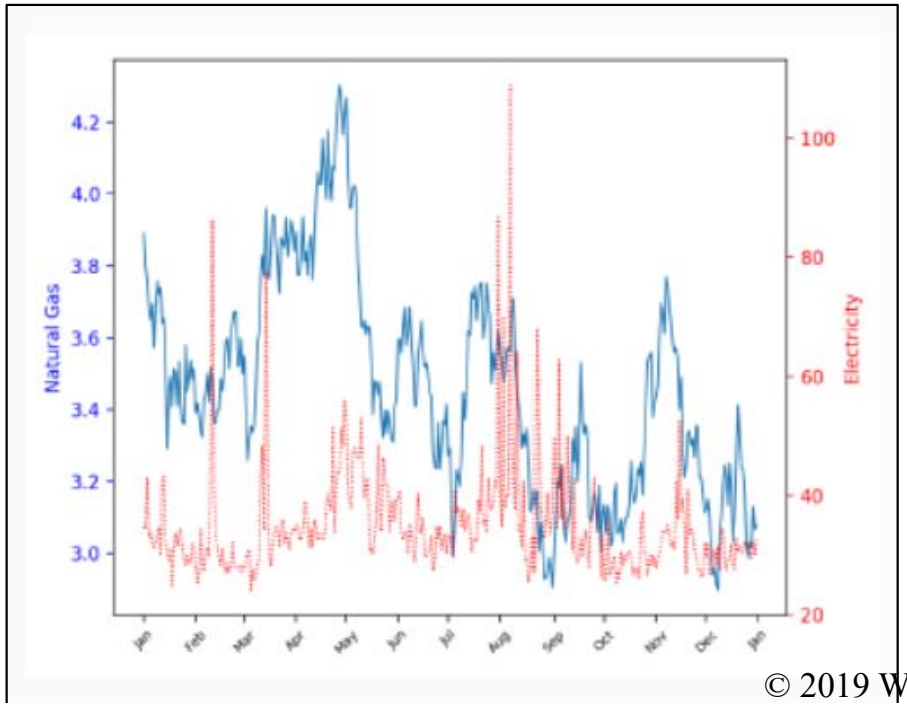
Lagged information

The state variable

- Lagged problems
  - » Resources
    - $R_{tt'}$ =Resources that will be available at time $t' \geq t$ given what you know at time t.
  - » Decisions
    - $x_{tt'}$ =Decision made at time t to do something at time t':
      - – Planning energy generation
      - – Purchasing a futures contract in month t for natural gas to exercise in month t'.
  - » Forecasts
    - $f_{tt'}^{X}$=Forecast of activity $X_{t'}$ at time t' given what we know at time t.

Future contracts for natural gas

- Air Liquide
  - » Largest industrial gases company with 64,000 employees.
  - » Consumes 0.1 percent of *global* electricity.



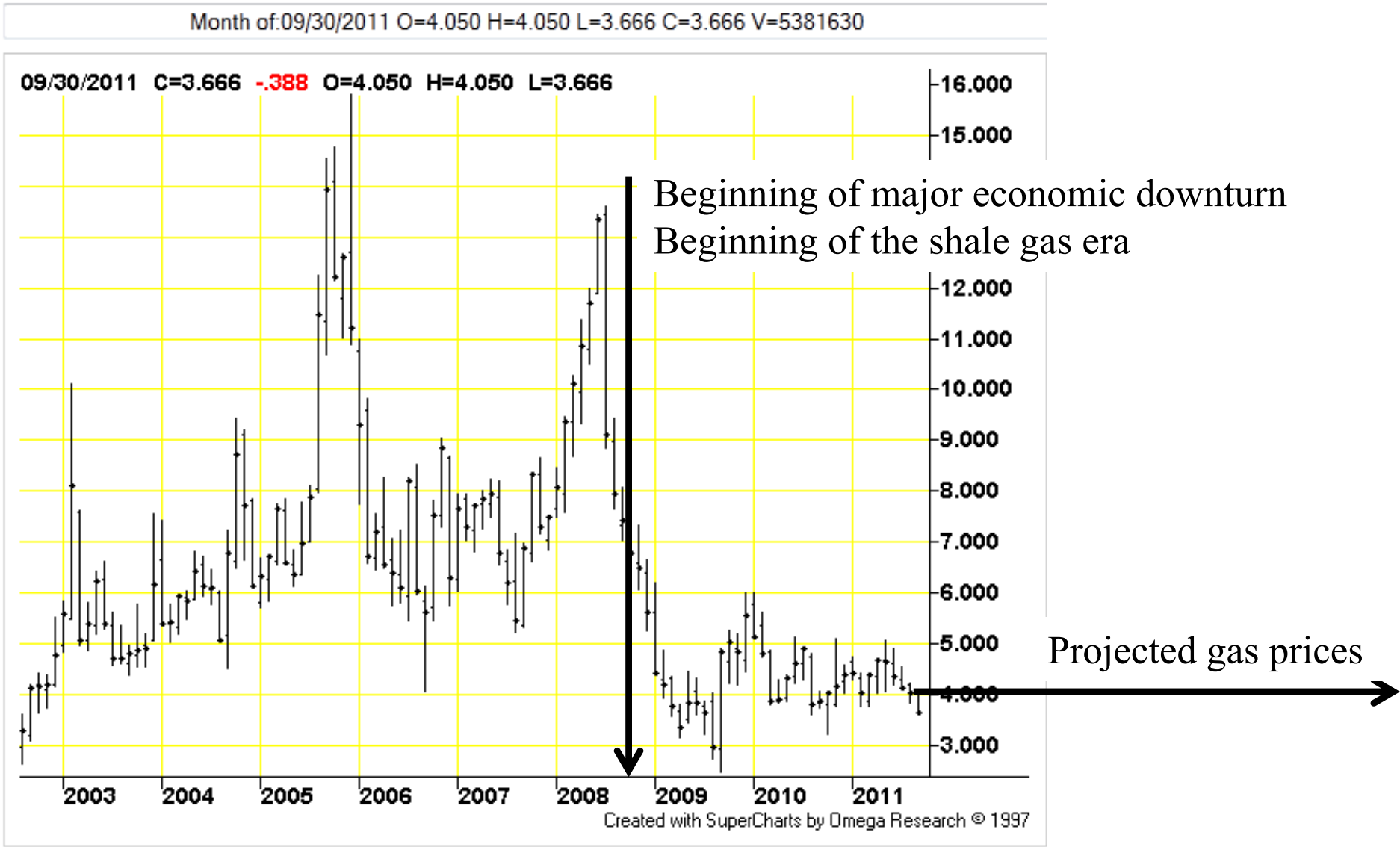
- Challenges
  - » Faces a variety of challenges to manage risk:
  - » Spikes in natural gas prices, electricity prices.
  - » Pipeline outages due to storms.

Future contracts for natural gas

- Basics
  - » Natural gas is a relatively clean, economical source of energy with approximately half the CO2 footprint of coal.
  - » Widely used in the generation of electricity:
    - In steam generators
    - In combustion turbine generators, which can be turned on and off relatively quickly in response to unexpected changes in demands.
  - » Over the last 10 years, a new manufacturing process has made it possible to economically tap gas locked in shale

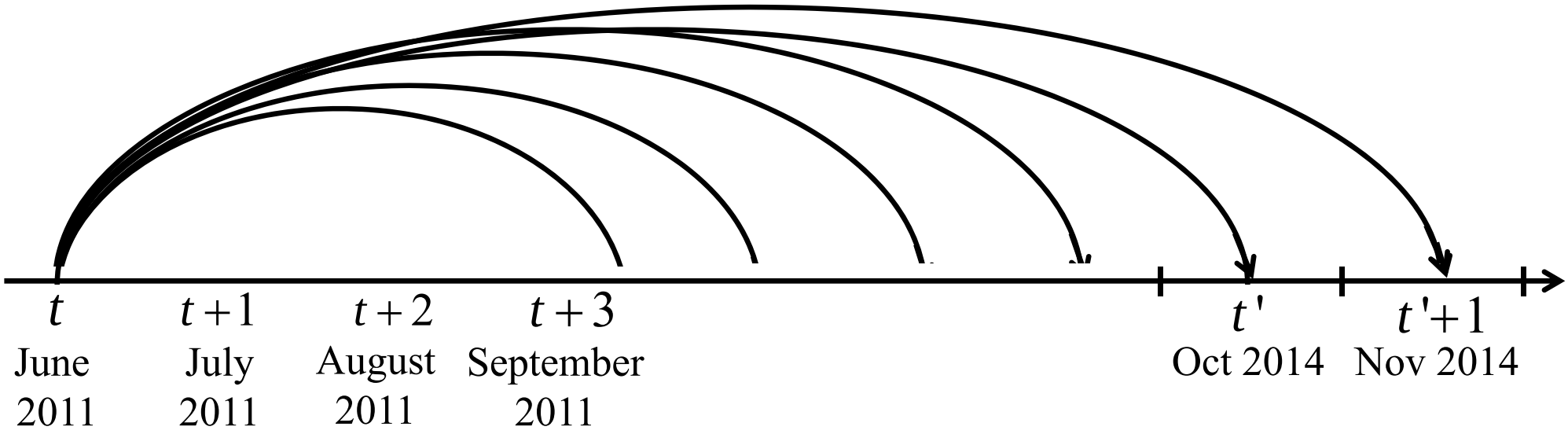
Future contracts for natural gas

- Historical natural gas prices



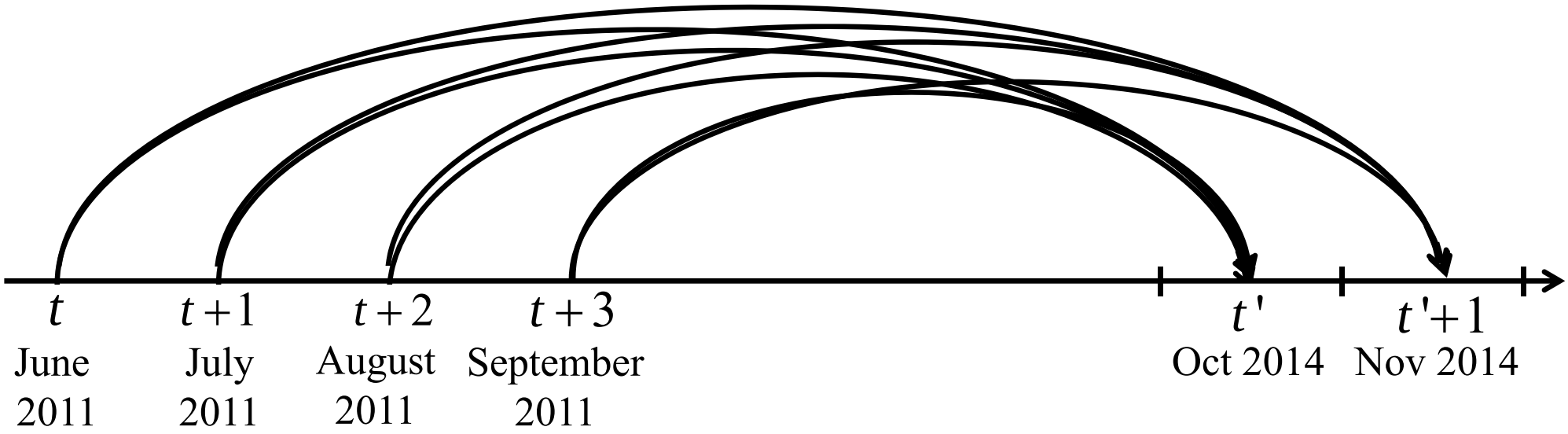
Future contracts for natural gas

- A myopic policy only looks at decisions we make at time t. It ignores decisions we might make at times t+1, t+2, …



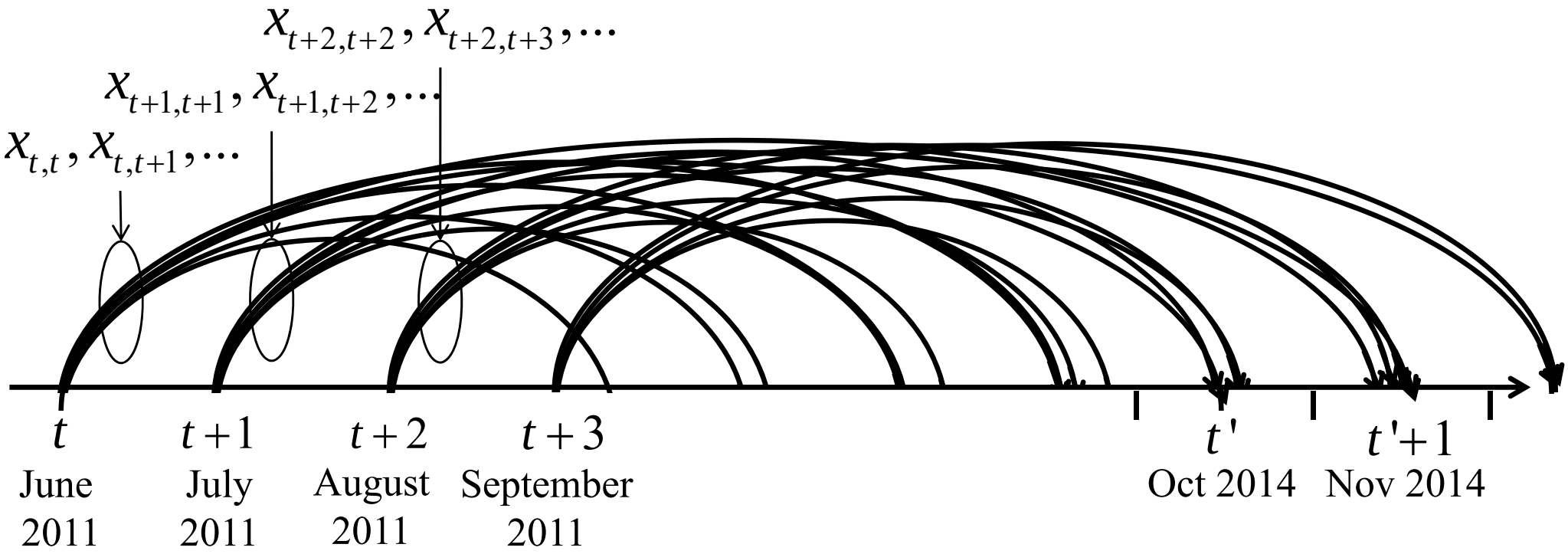
Future contracts for natural gas

- We can also buy hedges for different points in time in the future.



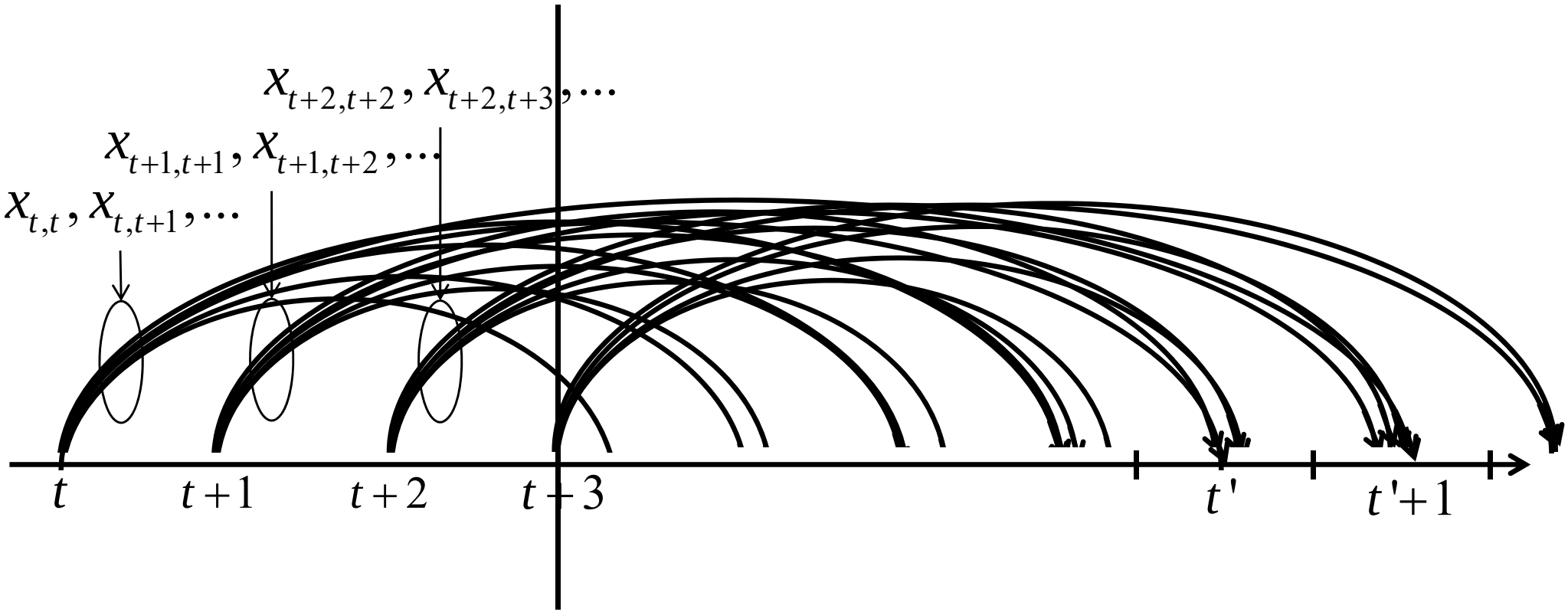
Future contracts for natural gas

- A myopic policy only looks at decisions we make at time t. A lookahead policy considers decisions we *may* make at times t+1, t+2, …, t+H



The state variable

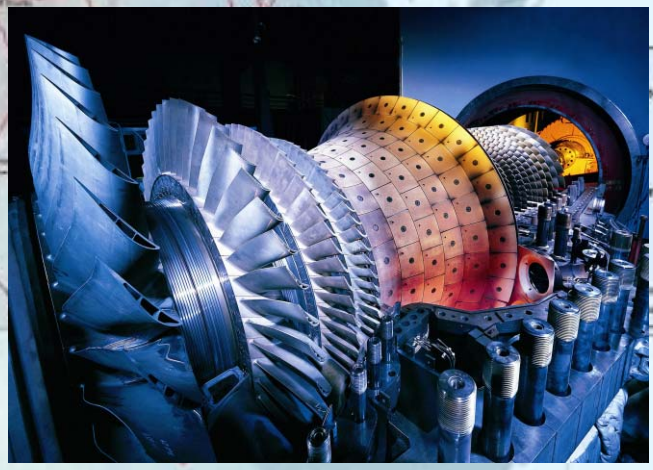
- State variable for lagged problems:



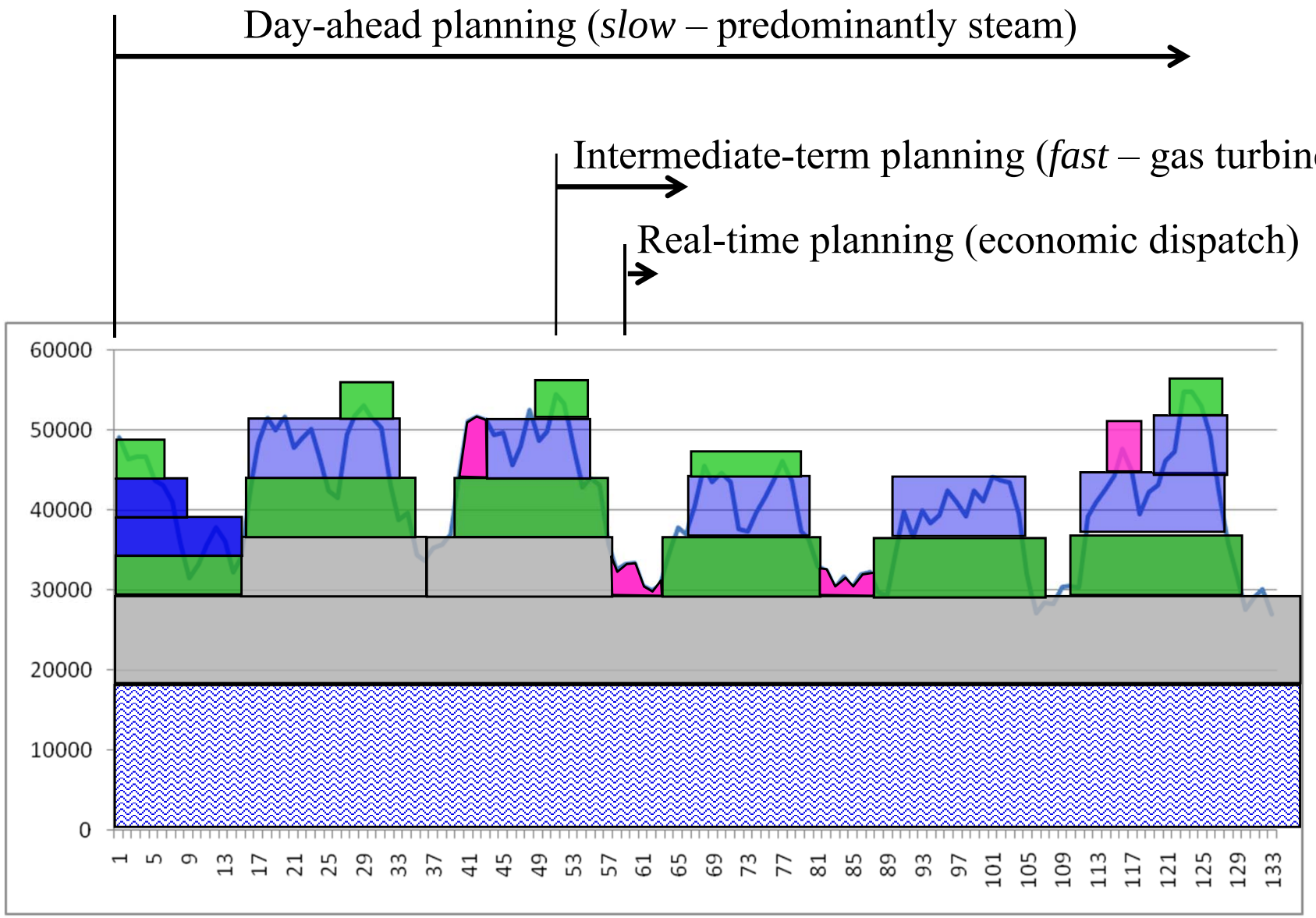
$R_{tt'}$ =Contracts signed prior to time t that become due at time $t' \geq t$.

State $R_t = (R_{tt'})_{t' \geq t}$

An energy generation portfolio

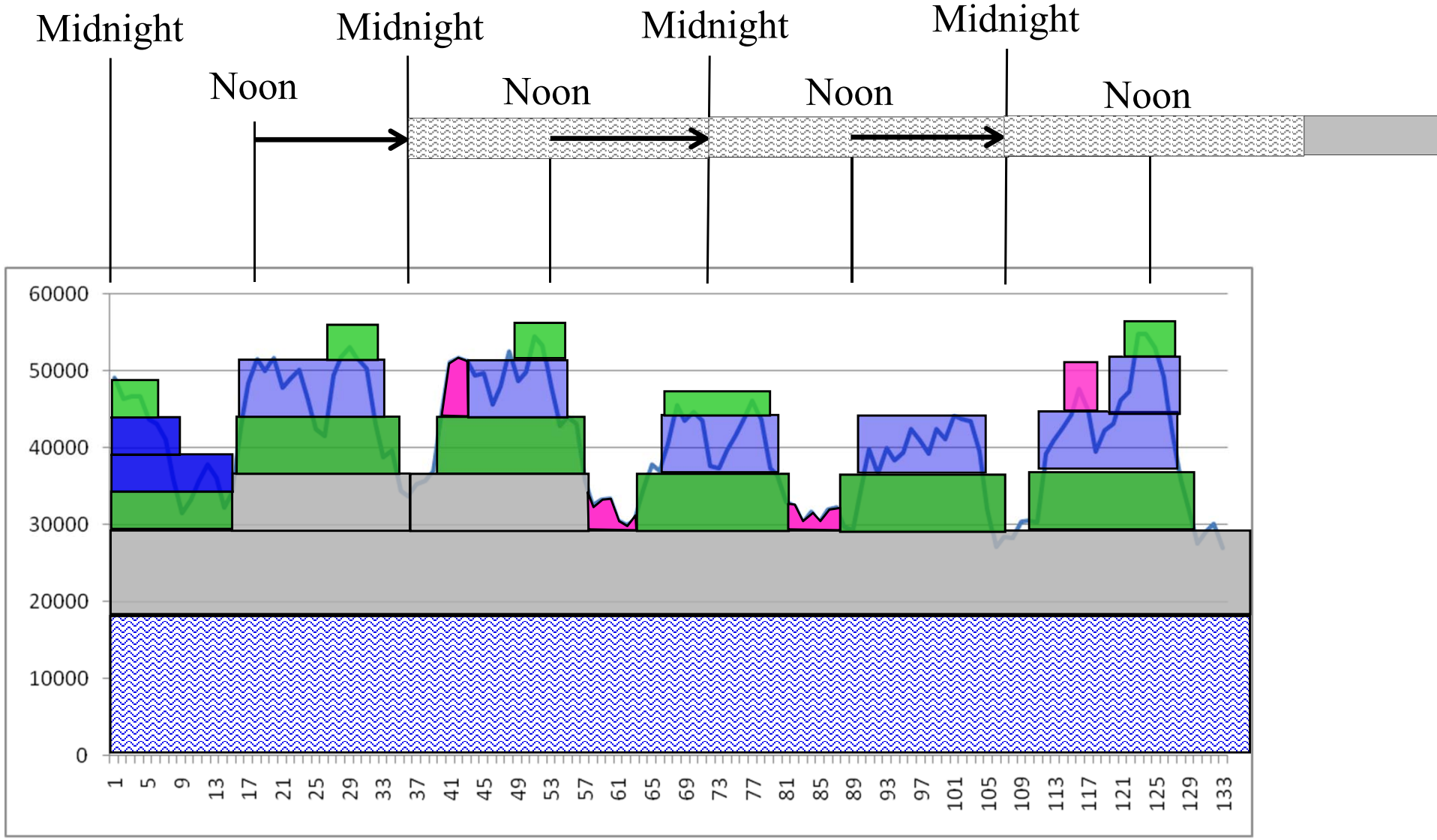


Stochastic unit commitment



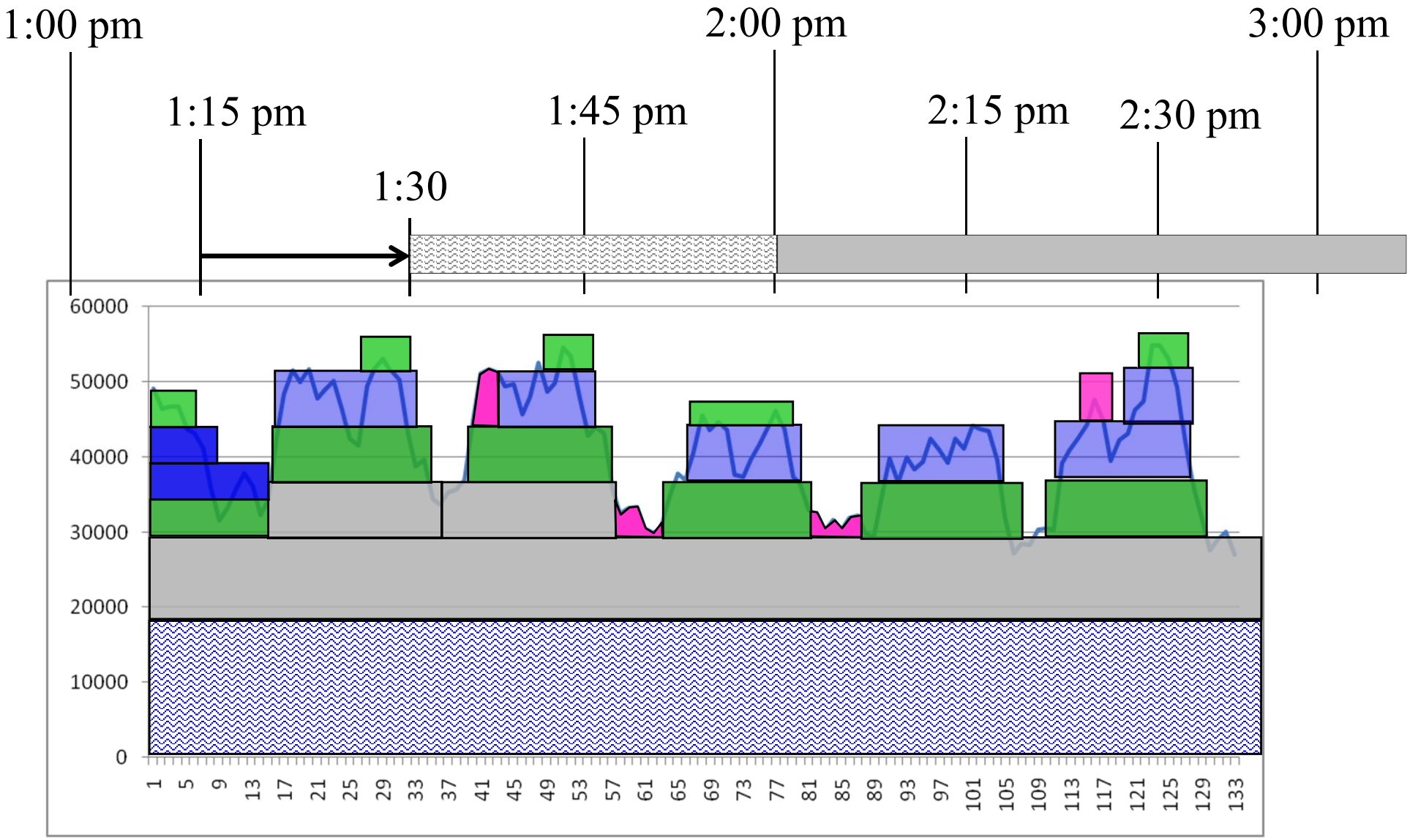
Stochastic unit commitment

- The day-ahead unit commitment problem



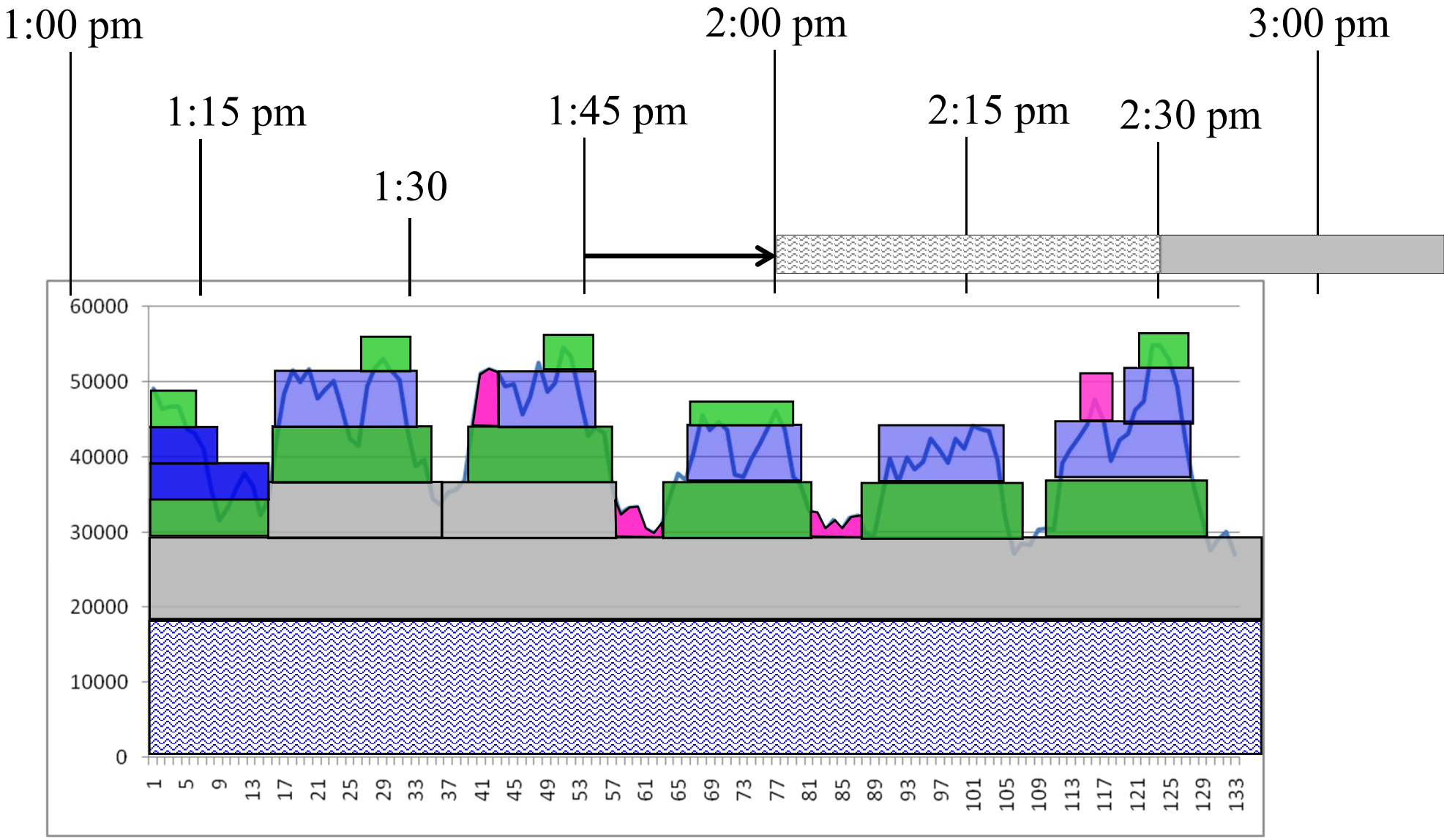
Stochastic unit commitment

- Intermediate-term unit commitment problem



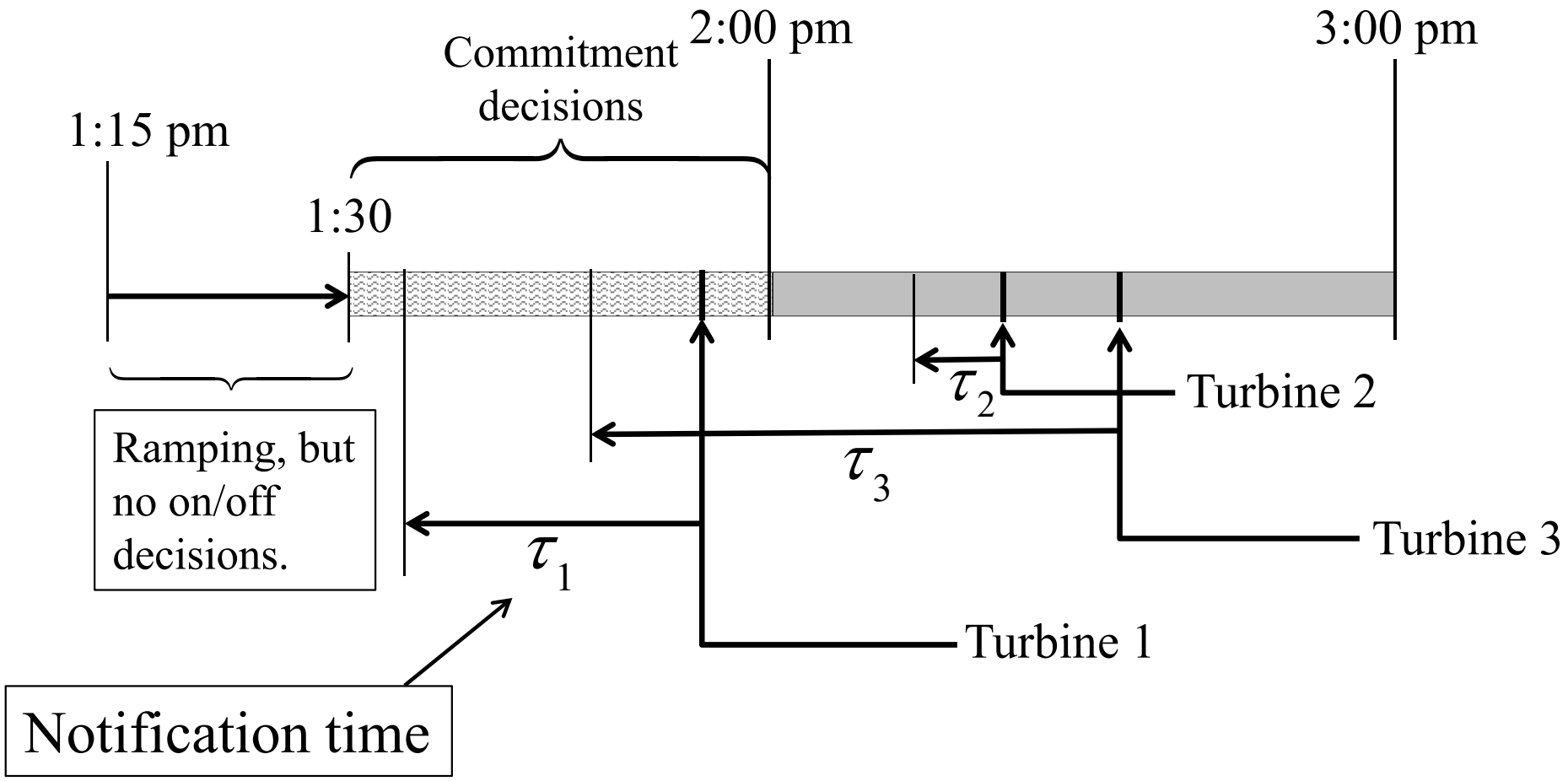
Stochastic unit commitment

- Intermediate-term unit commitment problem



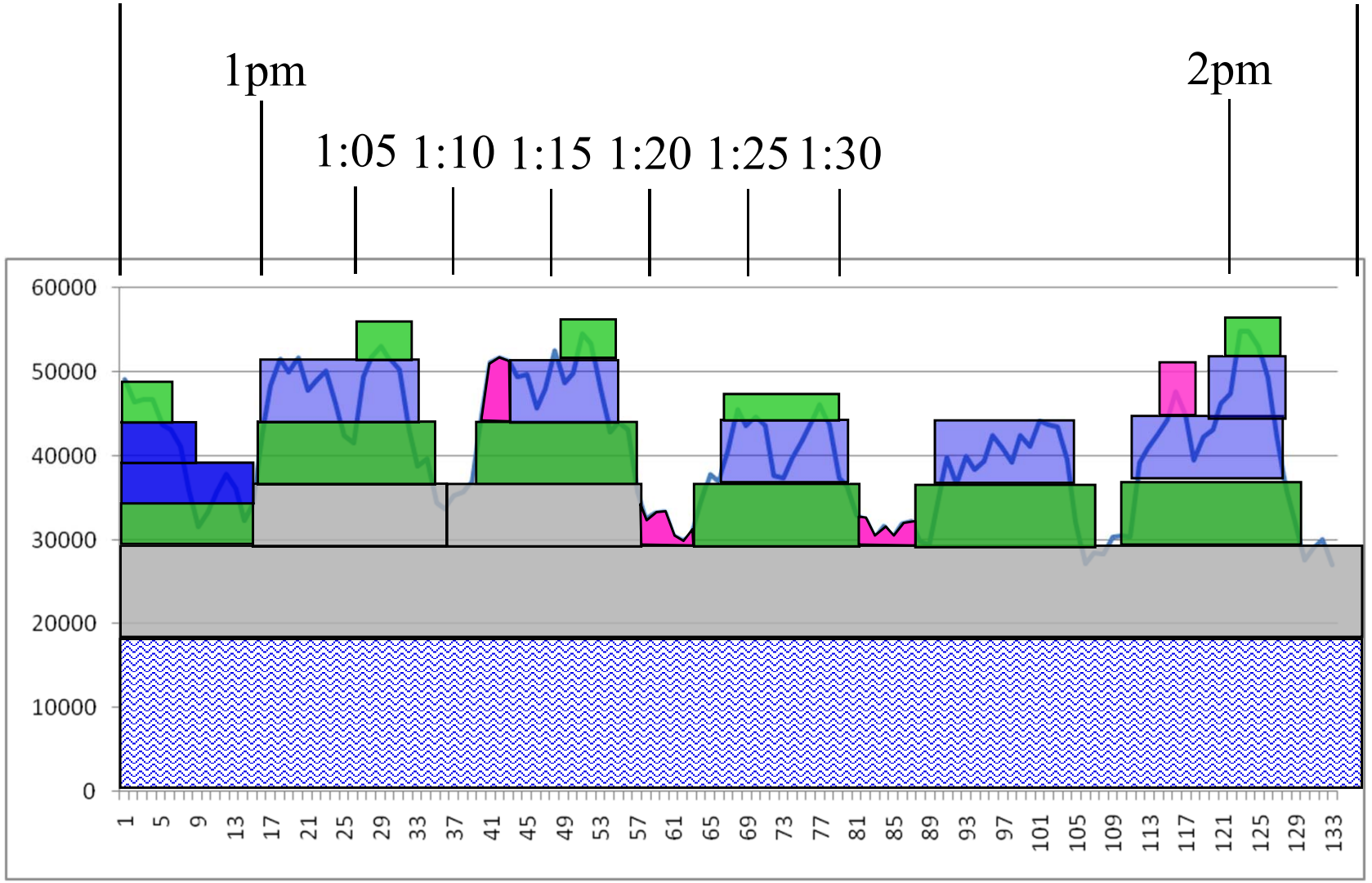
Stochastic unit commitment

- Intermediate-term unit commitment problem



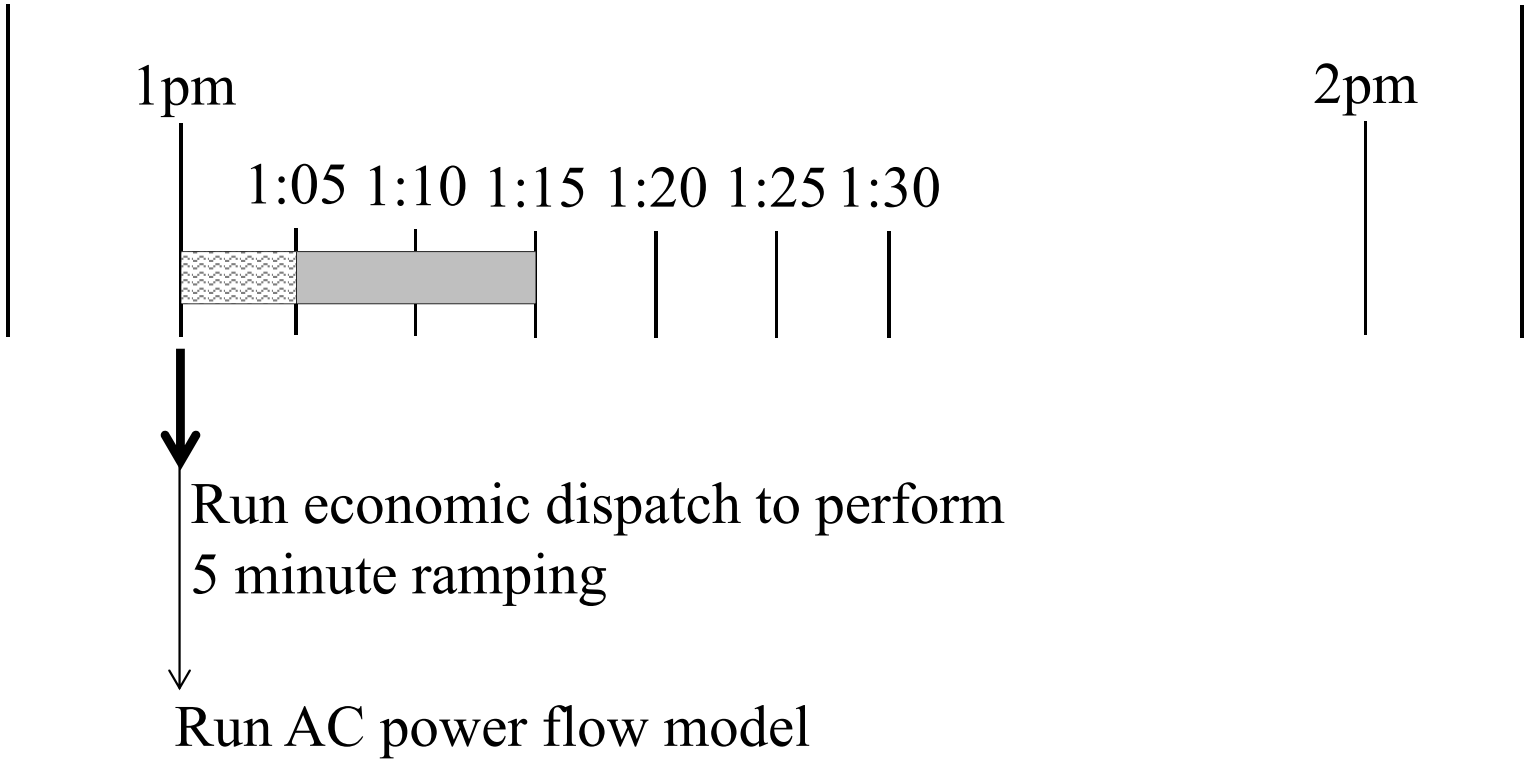
Stochastic unit commitment

- Real-time economic dispatch problem



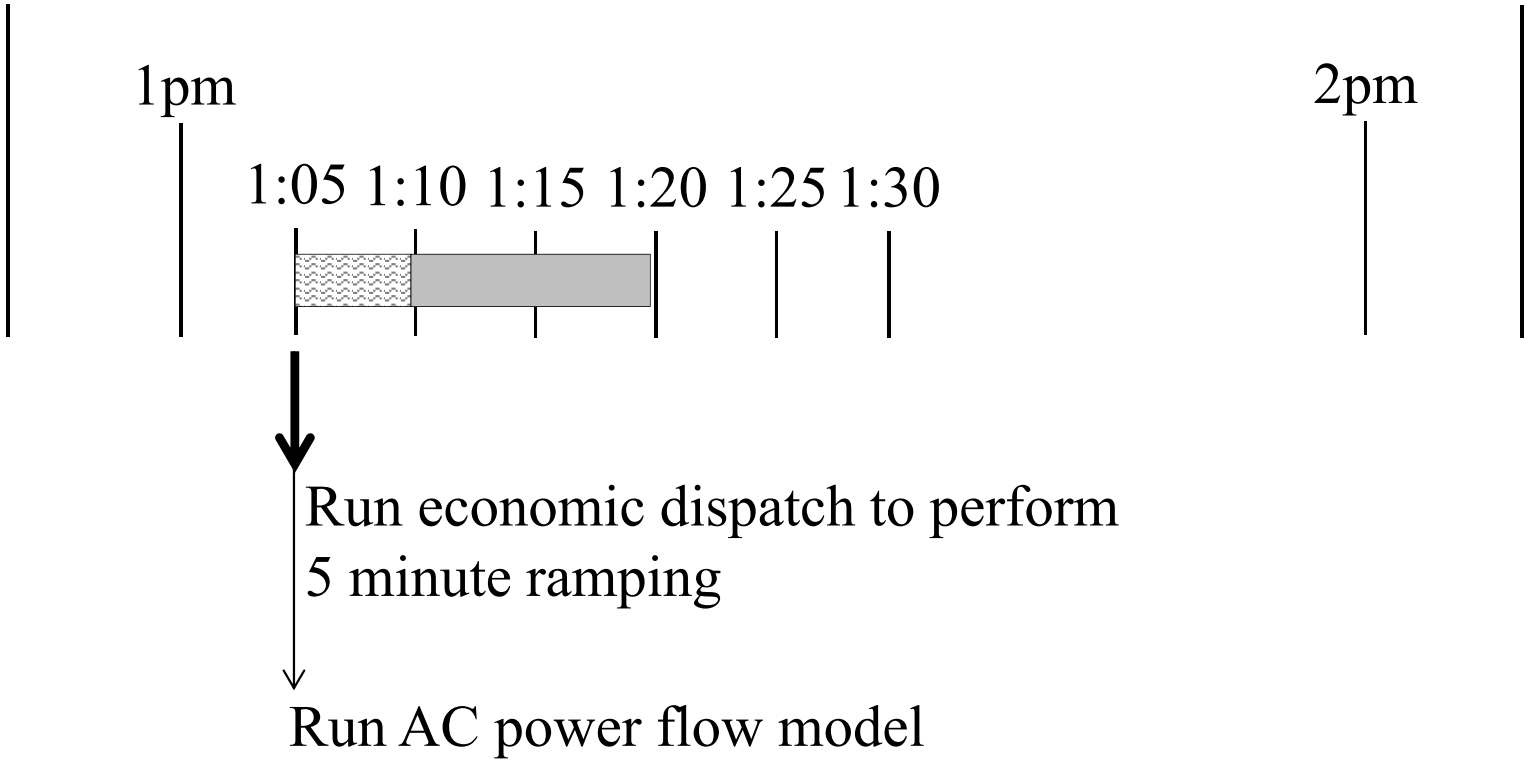
Stochastic unit commitment

- Real-time economic dispatch problem



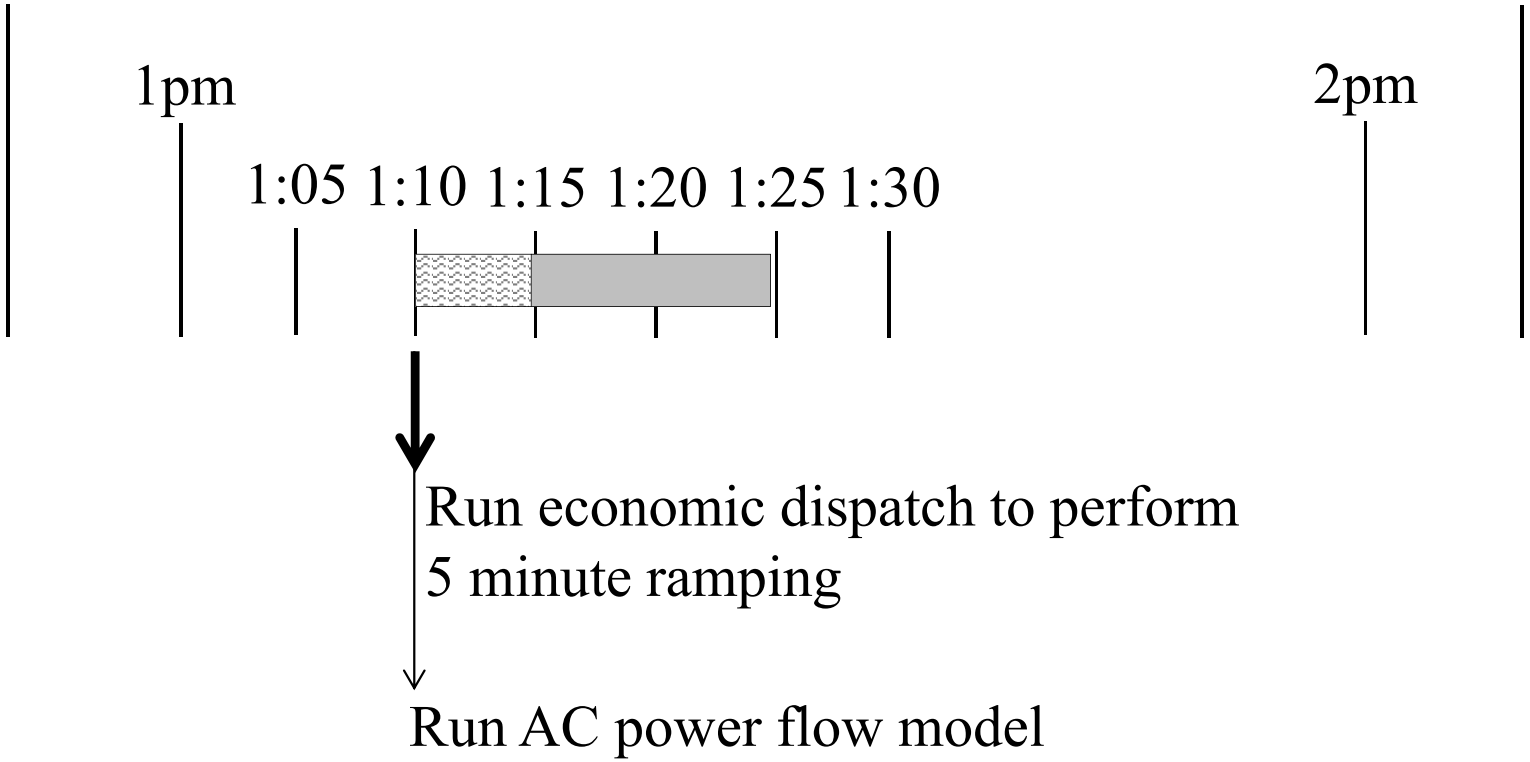
Stochastic unit commitment

- Real-time economic dispatch problem



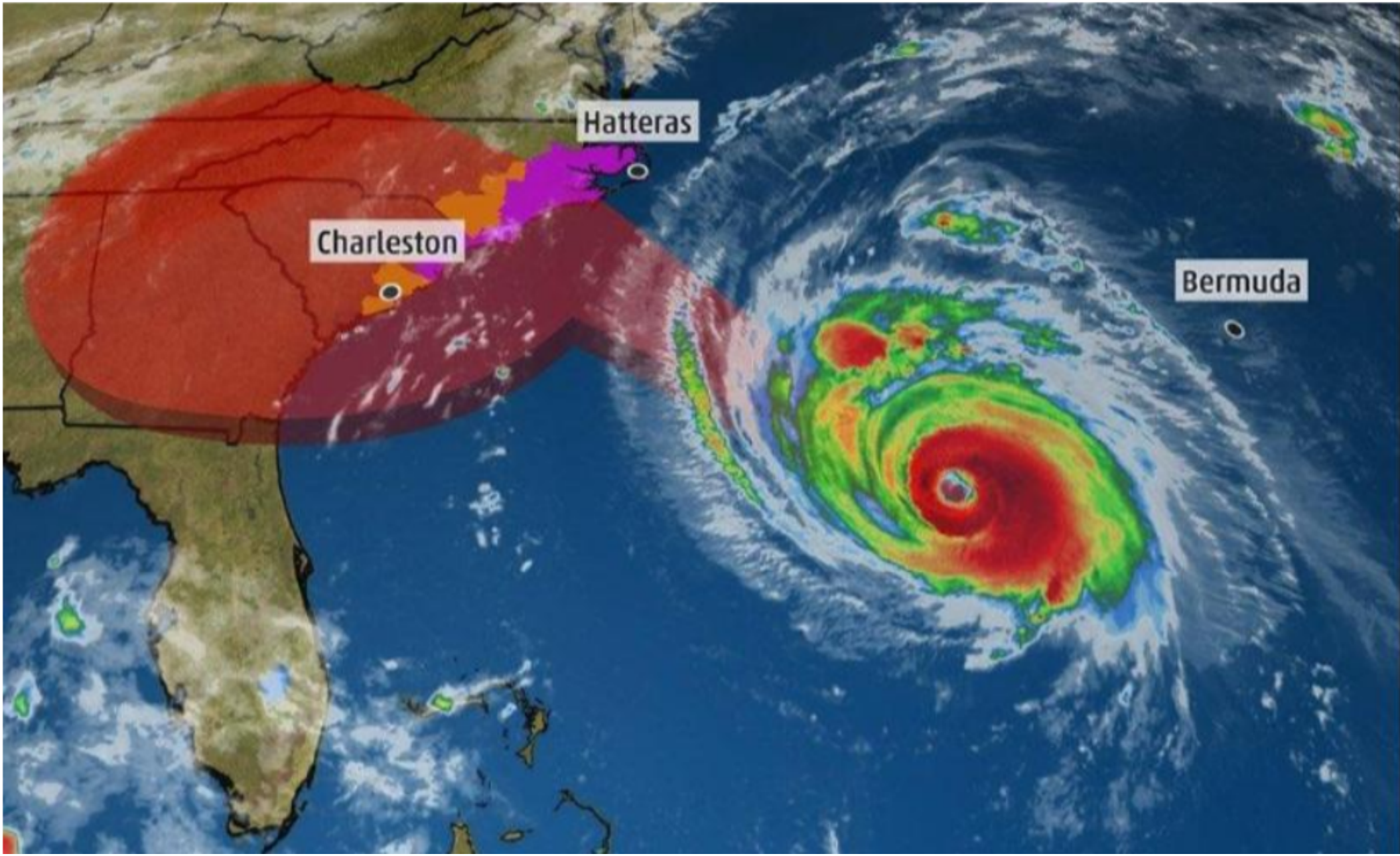
Stochastic unit commitment

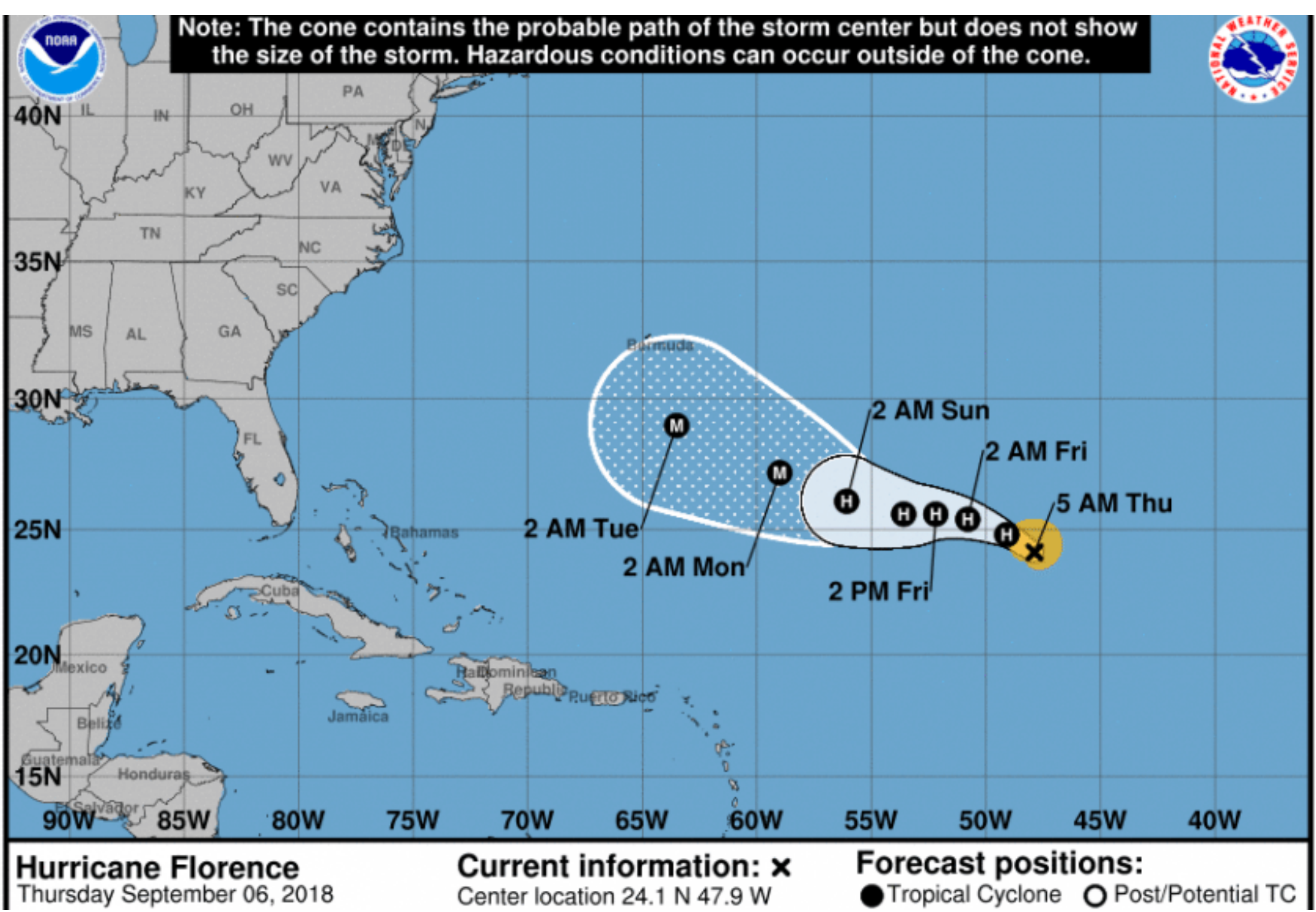
- Real-time economic dispatch problem



Hurricane Florence Targets Carolinas, Appalachians With Potentially Catastrophic Flooding, Destructive Winds; Hurricane Warning Issued

By weather.com meteorologists · 10 hours ago · weather.com

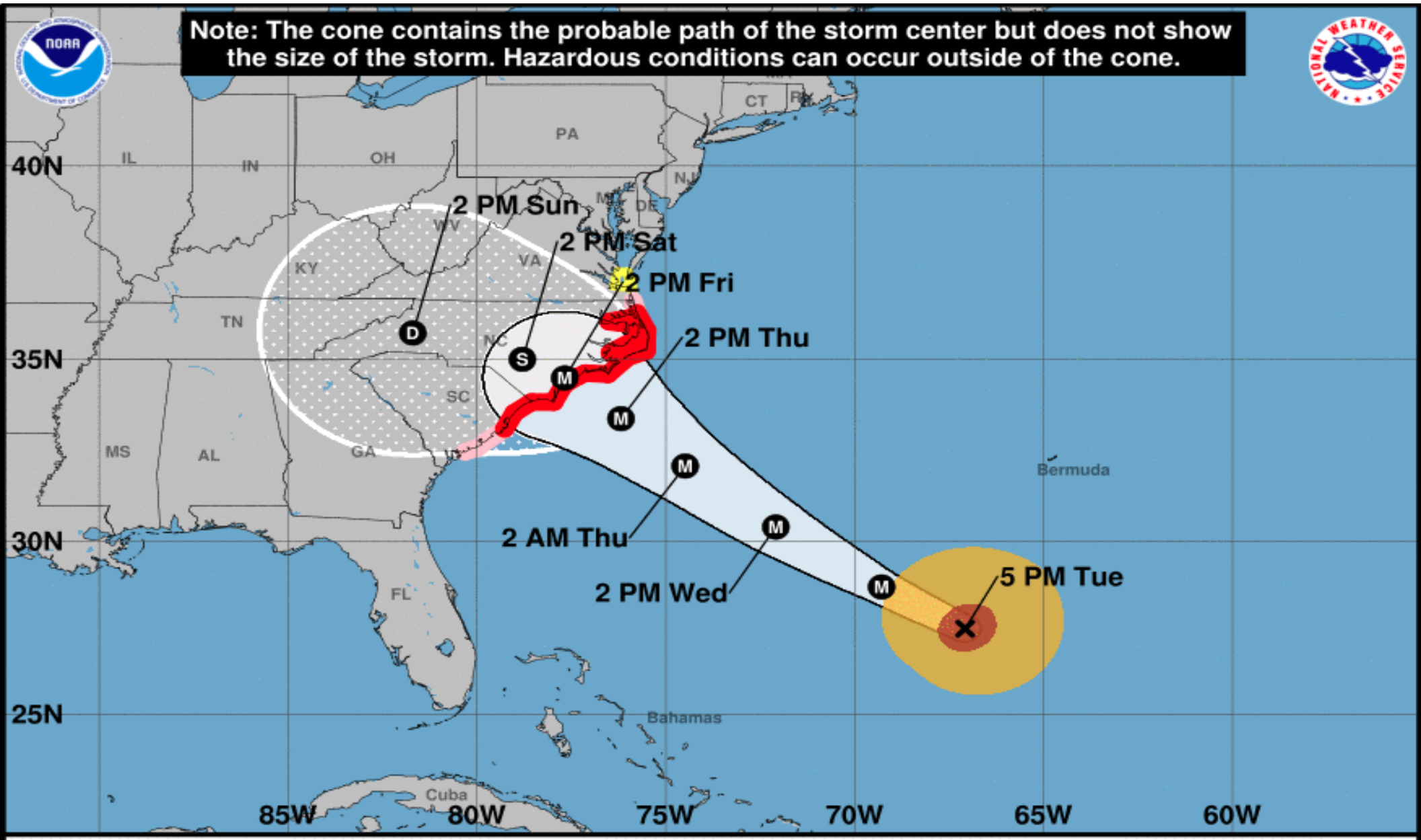




Hurricane Florence
Thursday September 06, 2018

Current information: ×
Center location 24.1 N 47.9 W

Forecast positions:
● Tropical Cyclone ○ Post/Potential TC



Hurricane Florence

Tuesday September 11, 2018
5 PM AST Advisory 50
NWS National Hurricane Center

Current information: ×

Center location 27.5 N 67.1 W
Maximum sustained wind 140 mph
Movement WNW at 17 mph

Forecast positions:

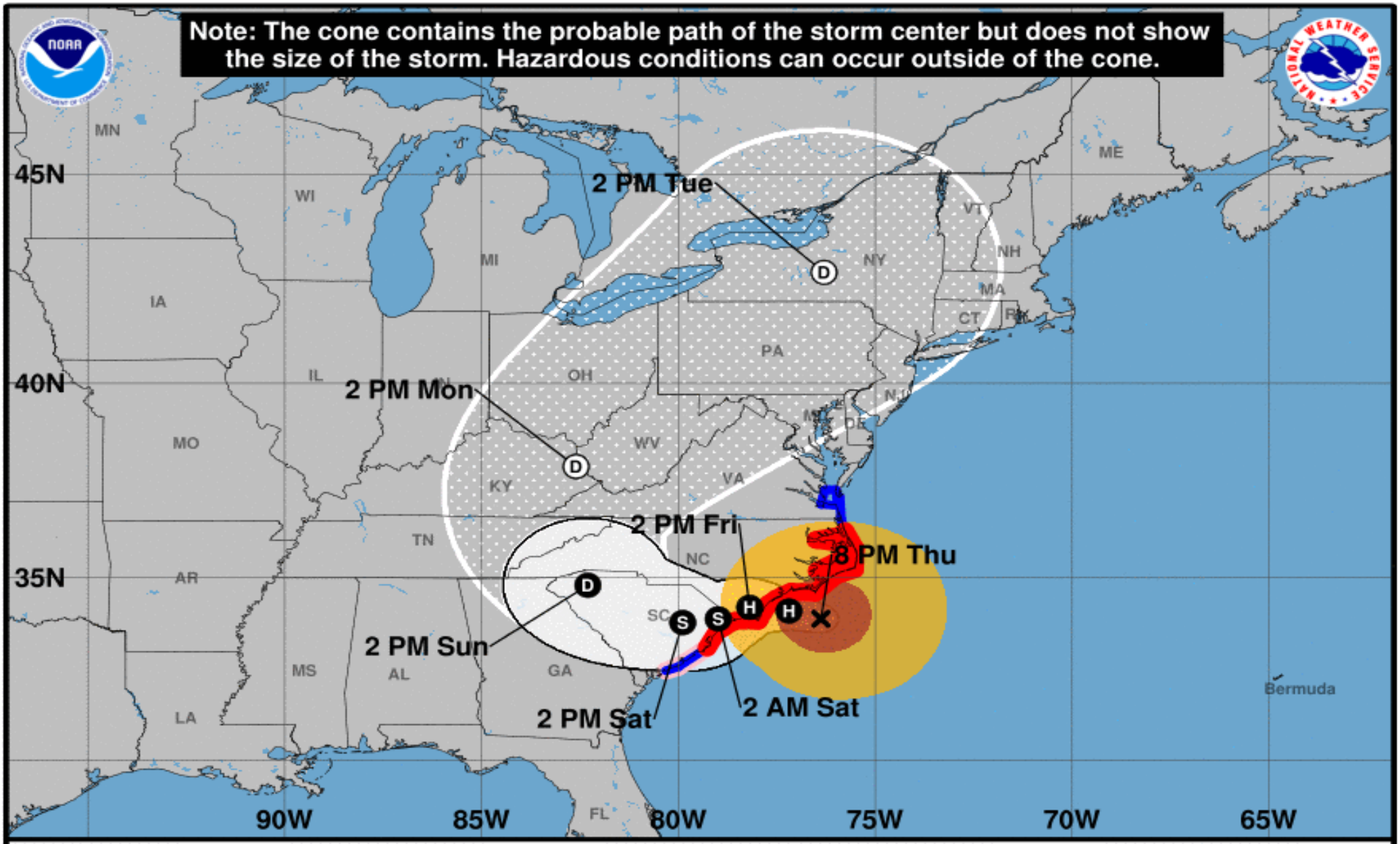
● Tropical Cyclone ○ Post/Potential TC
Sustained winds: D < 39 mph
S 39-73 mph H 74-110 mph M > 110 mph

Potential track area: Day 1-3, Day 4-5

Watches: Hurricane, Trop Stm

Warnings: Hurricane, Trop Stm

Current wind extent: Hurricane, Trop Stm



Hurricane Florence

Thursday September 13, 2018
8 PM EDT Intermediate Advisory 58A
NWS National Hurricane Center

Current information: ×

Center location 33.9 N 76.4 W
Maximum sustained wind 100 mph
Movement NW at 5 mph

Forecast positions:

● Tropical Cyclone ○ Post/Potential TC
Sustained winds: D < 39 mph
S 39-73 mph H 74-110 mph M > 110 mph

Potential track area: Day 1-3, Day 4-5

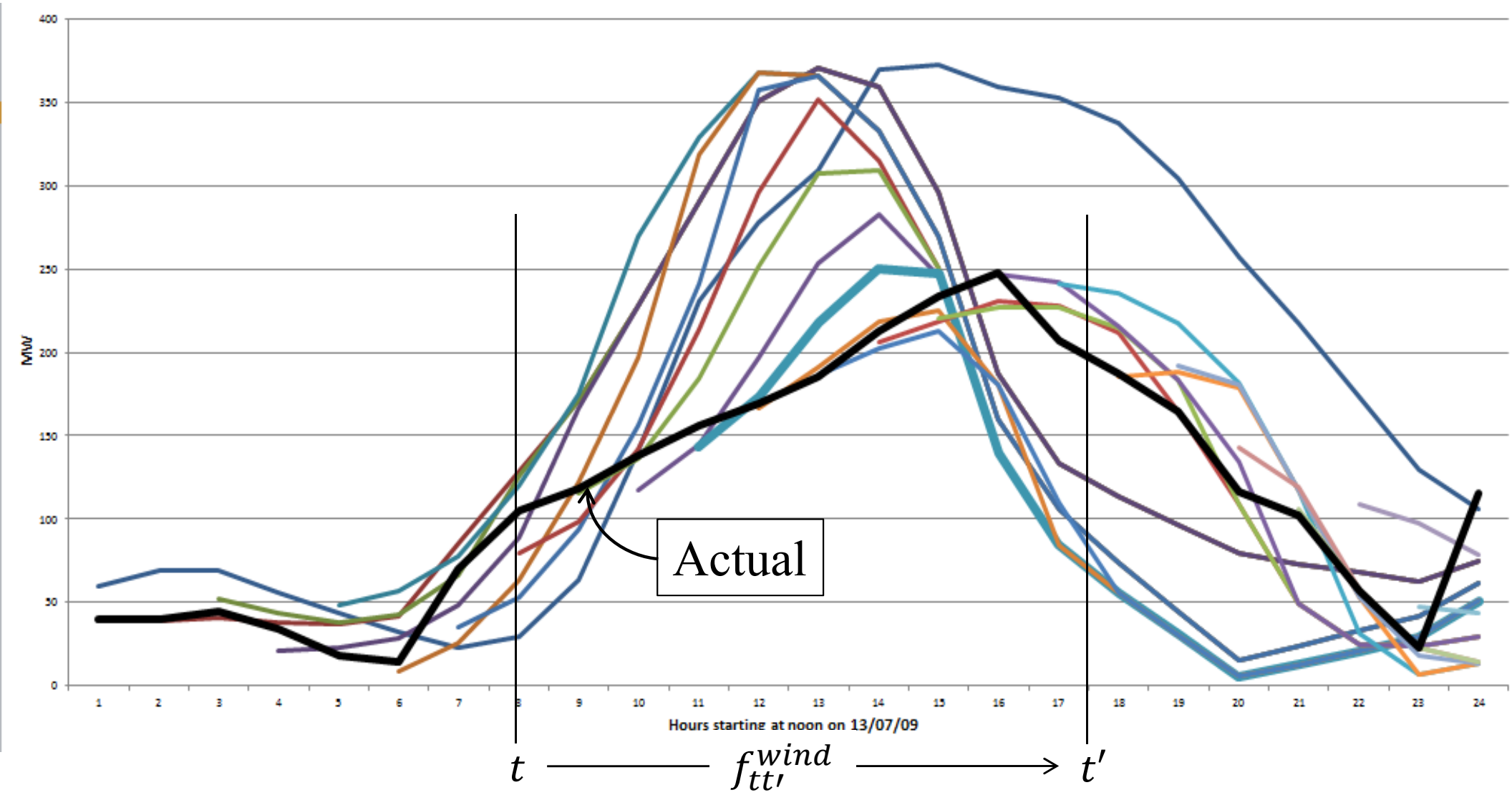
Watches: Hurricane, Trop Stm

Warnings: Hurricane, Trop Stm

Current wind extent: Hurricane, Trop Stm

Forecasting wind power

- Rolling 24-hour forecast of PJM wind farms



Decision variables

Decision variables

- When modeling decision variables:
  - » Model the decision variables
  - » Model the constraints
  - » Introduce the policy $X^{\pi}(S)$, and say that it will be designed later.

Decision variables

- There are three common notational systems for decisions:

  » Computer science

  a_t =Discrete action

  » Control theory

  u_t =Low dimensional continuous vector

  » Operations research

  x_t = Discrete or continuous, scalar or (typically) vector valued decisions.

Decision variables

- Styles of decisions
  - » Binary

    $x \in X = \{0,1\}$
  - » Finite

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

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

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

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

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

Decision variables

- Decisions for multiattribute resources
  - » Let

    $d \in \mathcal{D}$ is a type of decision (buy, sell, move to a location, treat with a drug, …)

    x_{tad} =Number of times we act on a resource with attribute a using decision d .

  - » Constraints:

$$\sum_{d \in \mathcal{D}} x_{tad} = R_{tad}$$

Decision variables

- How do we make decisions?
  - » We use *policies*, which are rules for making decisions. We will use notation such as:
    - $A^{\pi}(S)$ = The policy for determining action a.
    - $U^{\pi}(S)$ = The policy for determining action a.
    - $X^{\pi}(S)$ = The policy for determining action a.

    Here, π is a label that determines the type of function.
- Notes
  - » Model first, then solve. Do not attempt to design the policy while you are modeling the problem.
  - » It is extremely important that making a decision at time t can only use information in the state S_t at time t.

Exogenous information

Exogenous information

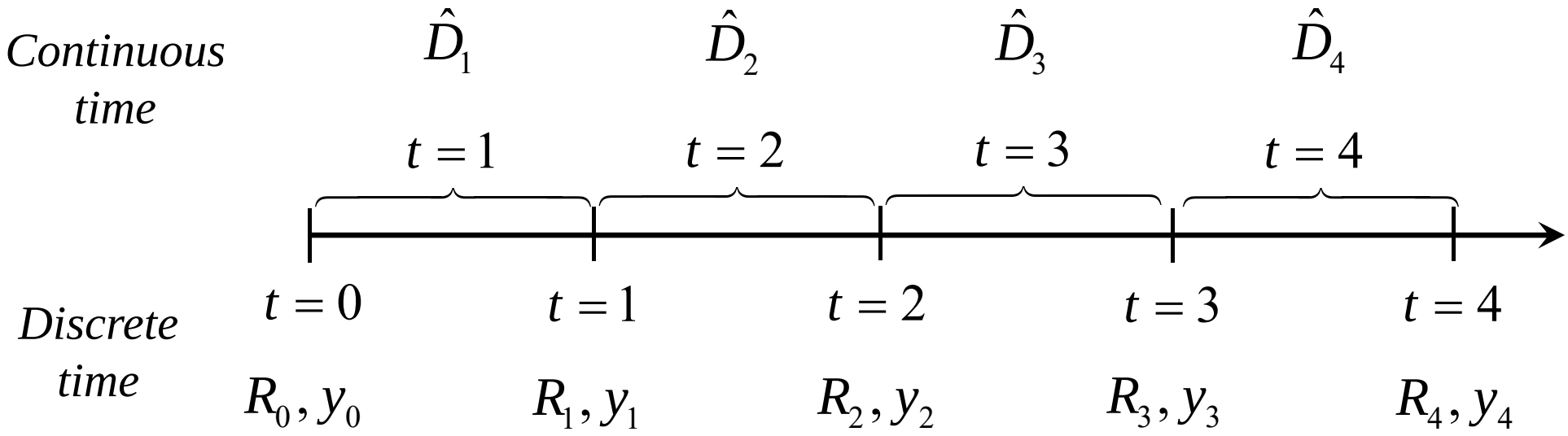
- Notation:
  - » W_t is information that first becomes available from outside the system by time t (or between $t-1$ and t).
  - » We are going to need to model different sources of exogenous information such as prices, temperature, equipment failures, and markets. We need to indicate the variables whose values come from outside our system. My notation is to put hats on exogenous information.
  - » We can represent exogenous information as the most recent value of a random variable, such as the energy from wind:

    $\hat{E}_{t+1}$ = energy from wind as of time $t+1$.
  - » … or we can write the exogenous information as the change in the state variable:

$$E_{t+1} = E_t + \hat{E}_{t+1}$$

Exogenous information

- We need a system for indexing time. In particular, it is important to know the mapping between discrete and continuous time.



It is useful to think of information as arriving continuously over time. Functions (states, decisions) are measured at a point in time.
At time t, anything $t' \leq t$ is known, anything $t' > t$ is unknown.

Exogenous information

Often, we need to model what will happen in the future to make a decision now.

Let:

D_t = The customer demand in time period $(t-1,t)$.

p_t = The market price at time t.

The future looks like:

$$\{(D_1, p_1),(D_2, p_2),...,(D_t, p_t)\}$$

We say that (D_t, p_t) is the *information* arriving at time t. It is sometimes useful to have a single variable to represent the new information arriving to the system at time t. There is not standard notation for modeling information. We let:

W_t = The information arriving in time $(t-1,t)$.

W_t represents a realization of the information that arrives in time period t.

Exogenous information

- Modeling sample paths:
  - » We are often simulating our process, which means we need to be able to represent a particular realization of our random information.
  - » We let

$$\omega = \text{A sample realization of } W_1, W_2, \ldots, W_T.$$

    where $\omega \in \Omega$ is the set of all realizations.
  - » In theory, Ω can be some infinite set of all possible outcomes (esp. if W_t is continuous), but in our work, Ω will *always* be a sample:

$$\Omega = \{\omega^1, \omega^2, \ldots, \omega^N\}$$

Exogenous information

For our example, we would write:

$$W_t = (D_t, p_t)$$

In the future, we do not know what might happen. Assume that the only type of new information arriving is customer demands. That is, $W_t = (D_t)$.

Assume that there are only 10 possible sets of demands that might happen in the future:

ω	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9
1	18	16	13	10	17	6	4	15	16
2	12	7	17	15	5	3	4	14	8
3	6	18	7	9	1	13	4	4	7
4	2	11	16	16	1	2	13	0	13
5	18	5	0	6	10	17	8	3	2
6	3	18	5	20	13	16	18	11	10
7	12	14	4	11	19	3	20	19	18
8	6	15	15	14	2	7	14	1	11
9	19	10	5	19	13	14	16	11	17
10	18	15	14	4	6	17	16	10	9

It is absolutely standard notation to index these outcomes by ω (don't ask why). The set of outcomes (the sample space) is referred to as Ω. So an element $\omega \in \Omega$ refers to a particular set of potential outcomes.

Exogenous information

We would say that D_t is a random variable because we do not know the demand D_t right now. We might, for example, assume that D_t follows some probability distribution so that we can describe the range of possible outcomes.

Sometimes we need to refer to a particular realization. For this, we let:

$D_t(\omega) =$ A sample realization of the demand at time t.

$D_t(\omega)$ is not a random variable. Let's say ω=6. Then,

$D_t(6) =$

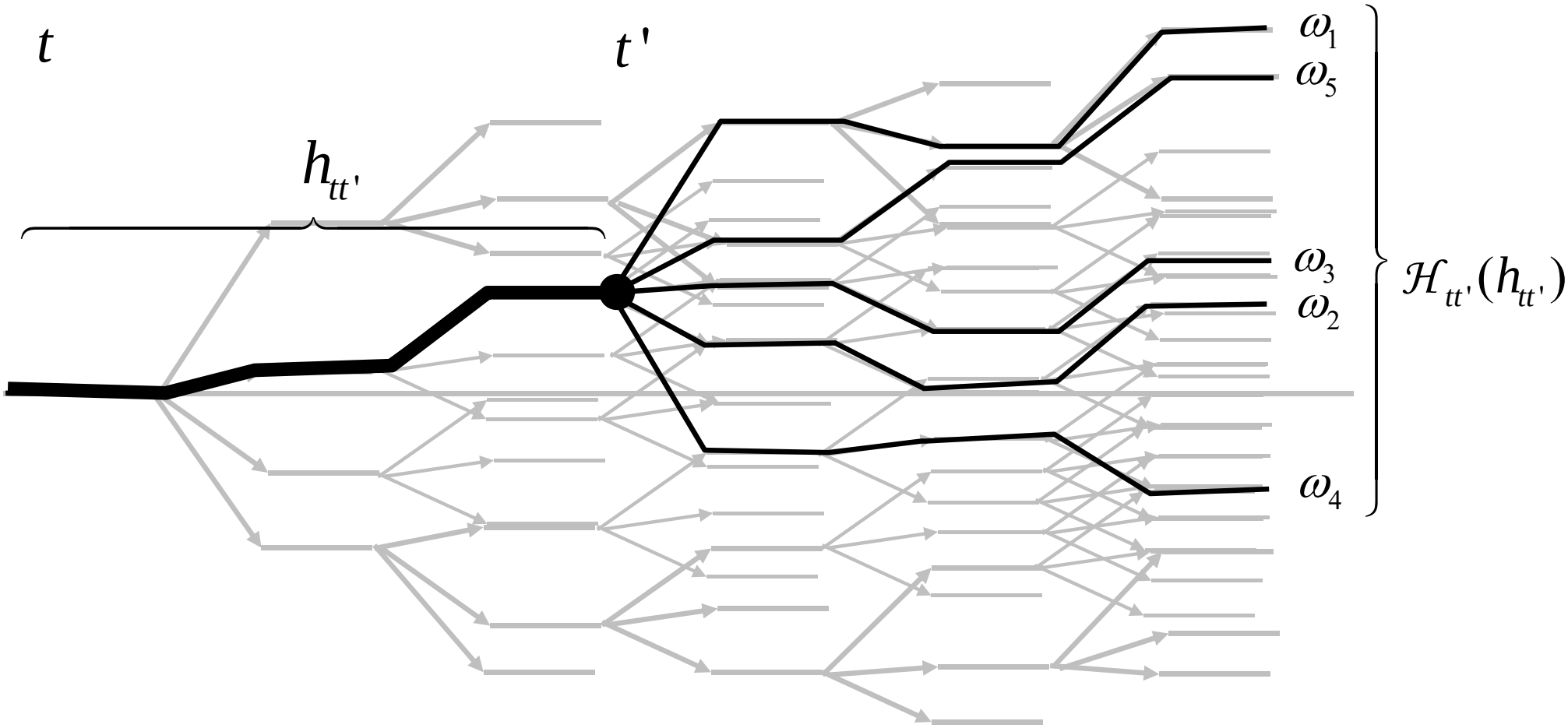
ω	D_1	D_2	D_3	D_4					
6	3	18	5	20	13	16	18	11	10

Exogenous information

- Notes:
  - » You need to be careful when using the "omega" notation.
  - » The index ω refers to an entire sequence of information events $W_1, W_2, \ldots, W_T$.
  - » If you make a decision $x_t(\omega)$, this means the decision you make at time t while following sample path ω, but this is like telling the decision x_t both the history and the future (since this is all contained in the sample path ω).
  - » We have to take additional steps to make sure that a decision x_t made while following sample path ω is not allowed to peek into the future.

Exogenous information

- From sample paths to histories:
  - » A node in the scenario tree is equivalent to a history



Exogenous information

- The complete exogenous information process consists of

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

- Notes:
  - » The initial state S_0 is where we input:
    - Deterministic parameters
      - The loss from converting energy from AC to DC and back
      - The maximum speed of a vehicle
    - Initial values of dynamic parameters
      - Initial inventory or price
    - Initial beliefs about unknown parameters
      - Prior belief about demand as a function of price
      - Prior belief about how a patient responds to a drug
  - » The dynamic information process $W_1, W_2, \dots, W_t$
    - W_{t+1} is any new information that becomes available after making decision x_t
    - W_{t+1} may depend on S_t and/or x_t.

Transition function

The transition function

- The transition function captures the evolution over time:

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

  - » The transition function goes by many names:
    - System model
    - Plant model
    - Plant equation
    - Law of motion
    - State equation
    - Transition law
    - Transfer function
    - "Model"

The transition function

- The transition function captures the evolution over time:

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

  - » At time t:

    S_t is known (deterministic)

    x_t is a deterministic function of S_t

    W_{t+1} is random

- The controls community (where this was invented) writes this as:

$$x_{t+1} = f\left(x_t, u_t, w_t\right)$$

where w_t is random at time t.

The transition function

- Physical resources:
  - » An inventory problem

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

  - » General resource allocation problems
    - Let

$$\delta_{a'}(a,d) = \begin{cases} 1 & If\ decision\ d\ turns\ a\ resource\ with\ attribute\ a\ to\ one\ with\ attribute\ a' \\ 0 & Otherwise \end{cases}$$

$$R_{t+1,a'} = \sum_{a \in A} \sum_{d \in D} x_{tad}\ \delta_{a'}(a,d)$$

The transition function

- Information processes

  » Exogenous information: Price process:

$$p_{t+1} = \theta_0 p_t + \theta_1 p_{t-1} + \theta_2 p_{t-2} + \varepsilon_{t+1}$$

  » Energy from wind:

$$E_{t+1} = E_t + \hat{E}_{t+1}$$

The transition function

- Updating belief models
  - » Lookup table frequentist:

$$\mu^n = \left(1-\frac{1}{n}\right)\mu^{n-1} + \frac{1}{n}W^n, \tag{3.6}$$

$$\hat{\sigma}^{2,n} = \begin{cases} \frac{1}{n}(W^n-\mu^{n-1})^2 & n=2, \\ \frac{n-2}{n-1}\hat{\sigma}^{2,n-1} + \frac{1}{n}(W^n-\mu^{n-1})^2 & n>2. \end{cases} \tag{3.7}$$

  - » Lookup table Bayesian
    - Independent beliefs

$$\mu^{n+1} = \frac{\beta^n\mu^n + \beta^W W^{n+1}}{\beta^n + \beta^W}, \tag{3.8}$$

$$\beta^{n+1} = \beta^n + \beta^W. \tag{3.9}$$

    - Correlated beliefs:

$$\mu^{n+1}(x) = \mu^n + \frac{W^{n+1}-\mu_x^n}{\lambda^W + \Sigma_{xx}^n}\Sigma^n e_x, \tag{3.20}$$

$$\Sigma^{n+1}(x) = \Sigma^n - \frac{\Sigma^n e_x (e_x)^T \Sigma^n}{\lambda^W + \Sigma_{xx}^n}. \tag{3.21}$$

The transition function

- Two frameworks:
  - » Model-based
    - This is where we have a set of equations that describe the transition.
    - In computer science, it refers to problems where the one-step transition matrix is known.
  - » Model-free
    - Here, we do not know the transition function.
    - Typical for complex problems (describing human behavior, the economy, climate, a complex physical problem)
    - In this case, we simply observe the next state without knowing how we got there.

Objective functions

Objective functions

- Performance metrics:
  - » Rewards, profits, revenues, costs (business)
  - » Gains, losses (engineering)
  - » Strength, conductivity, diffusivity (materials science)
  - » Tolerance, toxicity, effectiveness (health)
  - » Speed, stability, reliability (engineering)
  - » Risk, volatility (finance)
  - » Utility (economics)
  - » Errors (machine learning)
  - » Time (to complete a task)

Objective functions

- Styles of writing the performance metric:
  - » State-independent problems

$F(x, W)$ = A general performance metric (to be minimized or maximized) that depends only on the decision x and information W that is revealed after we choose x.

  - » State-dependent problems:

$C(S_t, x_t)$ = A cost/contribution function that depends on the state S_t and decision x_t.

$C(S_t, x_t, W_{t+1})$ = A cost/contribution function that depends on the state S_t and the decision x_t, and the information W_{t+1} that is revealed after x_t is determined.

$C(S_t, x_t, S_{t+1})$ = A cost/contribution function that depends on the state S_t and the decision x_t, after which we observe the subsequent state S_{t+1}.

Objective functions

- Offline (final reward)
  - » Asymptotic formulation:

$$\max_x \mathbb{E}F(x,W)$$

  - » Finite horizon formulation:

$$\max_\pi \mathbb{E}F(x^{\pi,N},W)$$

"ranking and selection"
or
"stochastic search"

- Online (cumulative reward)
  - » We have to learn as we go

$$\max_\pi \mathbb{E}\sum_{n=0}^{N-1} F(X^\pi(S^n),W^{n+1})$$

Objective functions

- Evaluating a policy:
  - » State independent, final reward

$$F^{\pi}(S_0) = \mathbb{E}_{S_0}\mathbb{E}_{W_1,\ldots,W_T|S_0}\mathbb{E}_{\widehat{W}|S_0}\{F(x^{\pi,T},\widehat{W})|S_0\},$$

    - Outer $\mathbb{E}_{S_0}$ ... is only used when S_0 contains a prior distribution of belief over some unknown parameter.
  - » State independent, cumulative reward (bandit problems):

$$F^{\pi}(S_0) = \mathbb{E}_{S_0}\mathbb{E}_{W_1,\ldots,W_T|S_0}\left\{\sum_{t=0}^{T} F(X^{\pi}(S_t),W_{t+1})|S_0\right\}.$$

  - » State dependent, cumulative reward:

$$\max_{\pi\in\Pi}\mathbb{E}_{S_0}\mathbb{E}_{W_1,\ldots,W_{T+1}|S_0}\left\{\sum_{t=0}^{T} C_t(S_t,X_t^{\pi}(S_t),W_{t+1})|S_0\right\}.$$

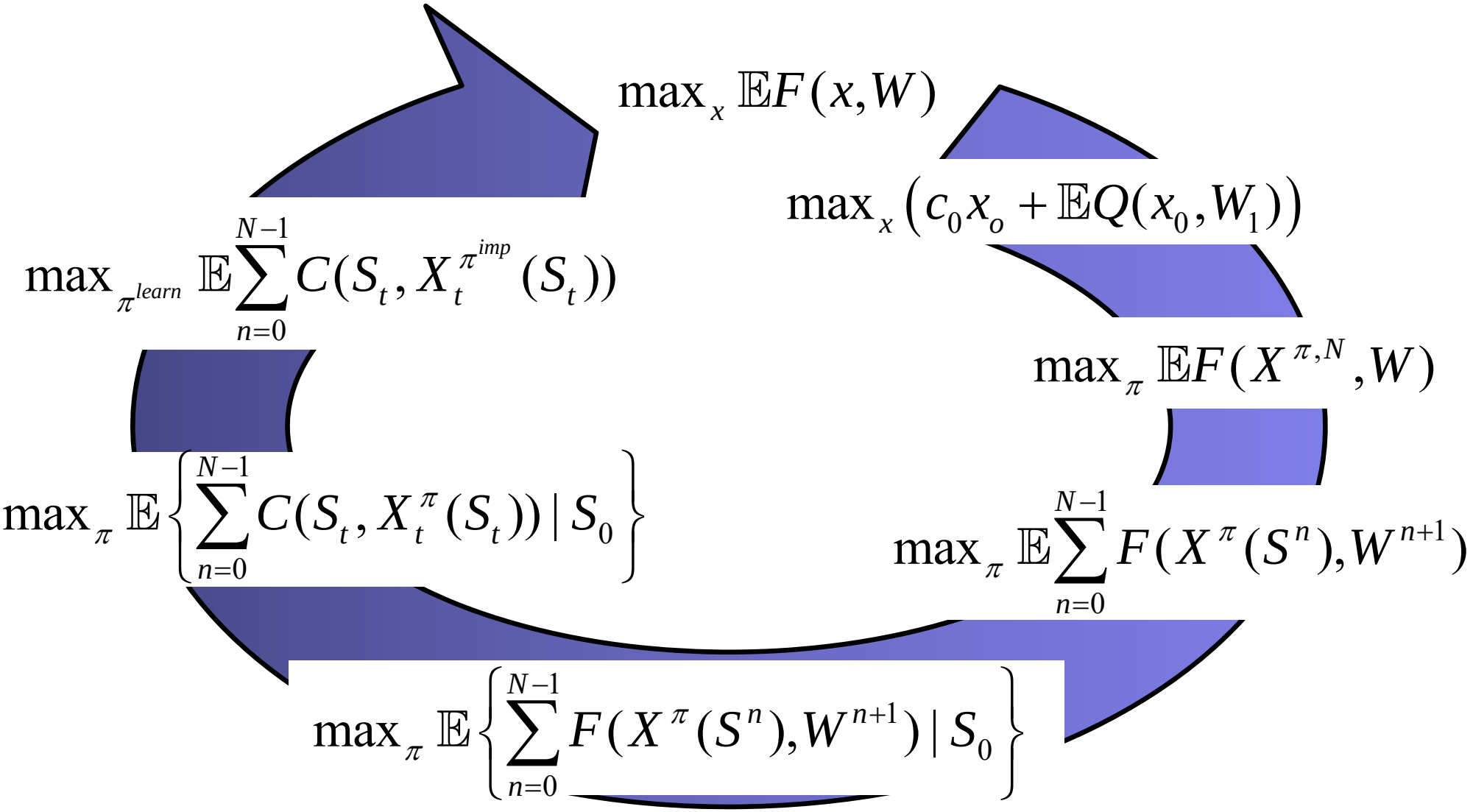
Problem classes

Problem classes

- Sequencing of decisions and information
  - » Decision, information
    - Classical stochastic search: $\max_x \mathbb{E}F(x,W)$
    - Finite time stoch. opt – final reward: $\max_\pi \mathbb{E}F(x^{\pi,N},\widehat{W})$
  - » Information, decision, information
    - State dependent: $\max_x \mathbb{E}\{F(x,W)|S_0\}$.
    - Contextual bandit: $\max_x \mathbb{E}\{F(x,W)|S_t^c\}$. S_t^c is the "context" that may change each time I have to solve the problem
  - » Decision, information, decision
    - Two-stage stochastic programming
      $max_{x_0}\, c_0x_0 + \mathbb{E}_W F(x_1(\omega),W_1(\omega))$
      where $F(x_1(\omega),W_1(\omega)) = \min_{x_1} c_1(\omega)x_1(\omega)$
  - » Decision, information, decision, information, …
    - Sequential dynamic programming, optimal control, …cumulative reward:
      - $\max_\pi \mathbb{E}\{\sum_{t=0}^{T} C(S_t,X^\pi(S_t)|S_0\}$

Problem classes

- The circle of stochastic optimization



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 $\mathbb{E}F(x,W)$, called the belief state B_t, so $S_t = B_t$.
- State dependent problems
  - » Now the *problem* may depend on what we know at time *t*:

$$\max_{0\le x\le R_t} \mathbb{E}C(S,x,W) = \mathbb{E}\left\{p_t \min(x,W) - cx\right\}$$

  - » Now the state is $S_t = (R_t, p_t, B_t)$

Problem classes

- State-dependent problems
  - » Why does it matter if a problem depends on dynamic information or not?
  - » For a state-independent problem, we are looking for a best alternative x^* that we call $x^{\pi,N}$ to represent the solution we found using policy (algorithm) π after N experiments.
  - » For a state-dependent problem, assume we can write our state as $S^n = (I^n, B^n)$ where I^n is information that the problem depends on, while B^n is our belief about the problem.
  - » Now, instead of finding $x^{\pi,N}$, we want to find $x^{\pi,N}(I^n)$, which means instead of finding a number (or vector of numbers), we are looking to find a *function* $x^{\pi,N}(I^n)$.
  - » This is … harder.

Problem classes

- Objective functions for:
  - » State independent and state dependent *problems*.
  - » Final reward and cumulative reward

	Offline Terminal reward	Online Cumulative reward
State independent problems	$\max_\pi \mathbb{E}\{F(x^{\pi,N}, W)\|S_0\}$ Stochastic search (1)	$\max_\pi \mathbb{E}\{\sum_{n=0}^{N-1} F(X^\pi(S^n), W^{n+1})\|S_0\}$ Multiarmed bandit problem (2)
State dependent problems	$\max_{\pi^{lrn}} \mathbb{E}\{C(S, X^{\pi^{impl}}(S\|\theta^{imp}), W)\|S_0\}$ Offline dynamic programming (4)	$\max_\pi \mathbb{E}\{\sum_{t=0}^{T} C(S_t, X^\pi(S_t), W_{t+1})\|S_0\}$ Online dynamic programming (3)

Problem classes

- Problem class (1): $\max_\pi \mathbb{E}F(x^{\pi,N}, W)$
  - » This is our basic stochastic search problem with a finite learning budget.
  - » The "policy" π may be a stochastic gradient algorithm, or a learning policy for derivative-free problems, where the policy would include the choice of stepsize rule.
  - » Either way, we can view this as looking for the best *algorithm* to solve this problem.

Problem classes

- Problem class (2): $\max_\pi \mathbb{E} \sum_{n=0}^{N-1} F(x^n, W^{n+1})$
  - » This is the same as (1), but now we are optimizing the cumulative reward where $x^n = X^\pi(S^n)$.
  - » The derivative-free version is the original multiarmed bandit problem, but the policy could also be a stochastic gradient algorithm with a particular stepsize.
  - » Both (1) and (2) are familiar to us.

Problem classes

- Problem class (3): $\max_\pi \mathbb{E} \sum_{t=0}^{T} C(S_t, X^\pi(S_t))$
  - » This is the same as (2), but now we are optimizing the state-dependent version.
  - » This is the form typically used to write down a classical dynamic program (which we will see later).
  - » In the state independent version, we want policies that will strike a balance between learning and doing. In state dependent problems, the state variable typically does not include a belief state (so no explicit learning), but clearly there has to be a learning component.
  - » If there was a belief state, this would be a form of sequential, contextual bandit problem. As of this writing, there is virtually no literature addressing this problem, even though this is how dynamic programs are traditionally written.

Problem classes

- Problem class (4): $\max_{\pi^{lrn}} \mathbb{E}C(S, X^{\pi^{imp}}(S))$
  - » The way to think of this is to compare to (1), and recognize that the policy π in (1) is a policy to *learn* $x^{\pi,N}$ which is the decision that we *implement*.
  - » So, we view this as a problem to find the best *learning policy* π^{lrn} to find the best policy π^{imp} that will be implemented.
  - » The hard part is – how do we take the expectation over the state S. The distribution of S will depend on the implementation policy.
  - » We can approximate the distribution of S by simulating the implementation policy over T time periods:

$$\max_{\pi^{lrn}} \mathbb{E}_{S^0} \mathbb{E}^{\pi^{imp}}_{(W_t^n)_{t=0}^T, n=1,\ldots,N|S^0} \left(\mathbb{E}^{\pi^{imp}}_{(\widehat{W}_t)_{t=0}^T|S^0} \frac{1}{T} \sum_{t=0}^{T-1} C(S_t, X^{\pi^{imp}}(S_t|\theta^{imp}), \widehat{W}_{t+1}) \right)$$

  - » We will see later that the "learning policy" describes any adaptive learning algorithm for solving dynamic programs. As with (1), we want a "good" algorithm, but finding an optimal learning algorithm is aspirational.

Problem classes

- Simulating objective functions
  - » It is important to know how to simulate objective functions.

	Offline Terminal reward	Online Cumulative reward
State independent problems	$\max_\pi \mathbb{E}\{F(x^{\pi,N}, W)\|S_0\}$ Stochastic search (1)	$\max_\pi \mathbb{E}\{\sum_{n=0}^{N-1} F(X^\pi(S^n), W^{n+1})\|S_0\}$ Multiarmed bandit problem (2)
State dependent problems	$\max_{\pi^{lrn}} \mathbb{E}\{C(S, X^{\pi^{impl}}(S\|\theta^{imp}), W)\|S_0\}$ Offline dynamic programming (4)	$\max_\pi \mathbb{E}\{\sum_{t=0}^{T} C(S_t, X^\pi(S_t), W_{t+1})\|S_0\}$ Online dynamic programming (3)

$$\max_{\pi^{lrn}} \mathbb{E}_{S^0} \mathbb{E}^{\pi^{imp}}_{(W_t^n)_{t=0}^T, n=1,\ldots,N|S^0} \left(\mathbb{E}^{\pi^{imp}}_{(\widehat{W}_t)_{t=0}^T|S^0} \frac{1}{T} \sum_{t=0}^{T-1} C(S_t, X^{\pi^{imp}}(S_t|\theta^{imp}), \widehat{W}_{t+1}) \right)$$

Problem classes

- Objective functions

	Offline Terminal reward	Online Cumulative reward
State independent problems	$\max_\pi \mathbb{E}\{F(x^{\pi,N}, W)\|S_0\}$ Stochastic search (1)	$\max_\pi \mathbb{E}\{\sum_{n=0}^{N-1} F(X^\pi(S^n), W^{n+1})\|S_0\}$ Multiarmed bandit problem (2)
State dependent problems	$\max_{\pi^{lrn}} \mathbb{E}\{C(S, X^{\pi^{impl}}(S\|\theta^{imp}), W)\|S_0\}$ Offline dynamic programming (4)	$\max_\pi \mathbb{E}\{\sum_{t=0}^{T} C(S_t, X^\pi(S_t), W_{t+1})\|S_0\}$ Online dynamic programming (3)

Learning policies:
- Approximate dynamic programming
- Q-learning
- SDDP
- …

We will focus on these in the second half of the course.

Problem classes

- Objective functions

	Offline Terminal reward	Online Cumulative reward
State independent problems	$\max_\pi \mathbb{E}\{F(x^{\pi,N}, W)\|S_0\}$ Stochastic search (1)	$\max_\pi \mathbb{E}\{\sum_{n=0}^{N-1} F(X^\pi(S^n), W^{n+1})\|S_0\}$ Multiarmed bandit problem (2)
State dependent problems	$\max_{\pi^{lrn}} \mathbb{E}\{C(S, X^{\pi^{impl}}(S\|\theta^{imp}), W)\|S_0\}$ Offline dynamic programming (4)	$\max_\pi \mathbb{E}\{\sum_{t=0}^{T} C(S_t, X^\pi(S_t), W_{t+1})\|S_0\}$ Online dynamic programming (3)

"Online" (cumulative reward) dynamic programming is recognized as the "dynamic programming problem," but the entire literature on solving dynamic programs describes class (4) problems. Class (3) appears to be an open problem class.

From machine learning to stochastic optimization

Stochastic opt. and machine learning

- Optimization in machine learning

  » Fitting them model to data
    - Offline-batch
    - Online

  » The decision of what information to collect
    - Active learning – origins in the machine learning community
    - Multiarmed bandit problems – origins in the probability community

Canonical problems

- Statistical model fitting
  - » Given dataset:

$$x^n = (x_1^n, x_2^n, ..., x_K^n)$$

$$= \text{Independent variables, explanatory variables, covariates, ...}$$

$$y^n = \text{Dependent variable}$$

  - » General optimization problem:

$$\min_{f \in F, \theta \in \Theta^f} \sum_{n=1}^{N} \left(y^n - f(x^n \mid \theta) \right)^2$$

  - » Popular strategy
    - Linear models:

$$f(x^n \mid \theta) = \sum_{f \in F} \theta_f \phi_f(x^n)$$

    - Neural networks
    - Etc. etc. etc. (many approaches from stat/ML)

Stochastic opt. and machine learning

- Statistics/machine learning
  - » Data

    x_t or x_n or x^n
  - » "Decision" (parameters)

    function $f \in F$ and $\theta \in \Theta^f$
  - » Predicting:

    y_n
  - » Objective

$$MSE = \min_\theta \frac{1}{N} \sum_{n=1}^{N} \left(y_n - f(x_n \mid \theta) \right)^2$$

- Stochastic optimization
  - » Data

    State S_t or S^n
  - » Decision

    x_t or x^n
  - » "Predicting"

    Decision x_t or x^n

    Value of state $V(S_t)$
  - » Objective

$$\max_\pi \mathbb{E} F(X^{\pi,N}, W)$$

or:

$$\max_\pi E^\pi \left\{ \sum_t C\left(S_t, X_t^\pi(S_t)\right) \right\}$$

Stochastic opt. and machine learning

- Statistics/machine learning
  - » Classical batch learning

(1) $$\min_\theta \frac{1}{N}\sum_{n=1}^{N}\left(y_n - f(x_n \mid \theta)\right)^2$$

  - » Online (recursive) statistics

(2) $$\min_\theta \mathbb{E}\left(Y - f(x \mid \theta)\right)^2$$

  - » Searching over functions

(3) $$\min_{f\in\mathcal{F},\theta\in\Theta^f} \mathbb{E}\left(Y - f(x \mid \theta)\right)^2$$

  - » Matching data over time

(4) $$\min_{f\in\mathcal{F},\theta\in\Theta^f} \mathbb{E}\sum_t \left(\hat{y}_{t+1} - f(x_t \mid \theta)\right)^2$$

- Stochastic optimization
  - » Sample average approximation

$$\max_x \frac{1}{N}\sum_{n=1}^{N} F(x,\omega^n)$$

  - » Stochastic search

$$\max_x \mathbb{E}F(x,W)$$

  - » Policy search

$$\max_\pi E^\pi\left\{\sum_t C\left(S_t, X_t^\pi(S_t)\right)\right\}$$

  - » Matching posterior decisions

$$\max_{f\in F,\theta\in\Theta^f} E^\pi\left\{\sum_t \left(\hat{X}_{t+1}(\omega) - X^\pi(S_t \mid \theta)\right)^2\right\}$$

An energy storage example

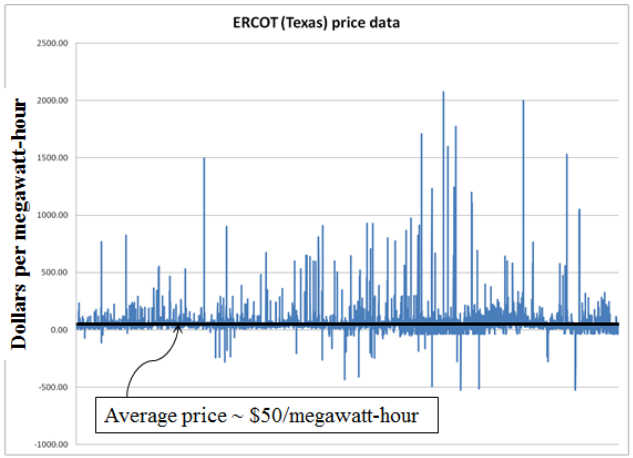
Energy arbitrage

An energy storage problem

- A model of our problem
  - » State variables
  - » Decision variables
  - » Exogenous information
  - » Transition function
  - » Objective function

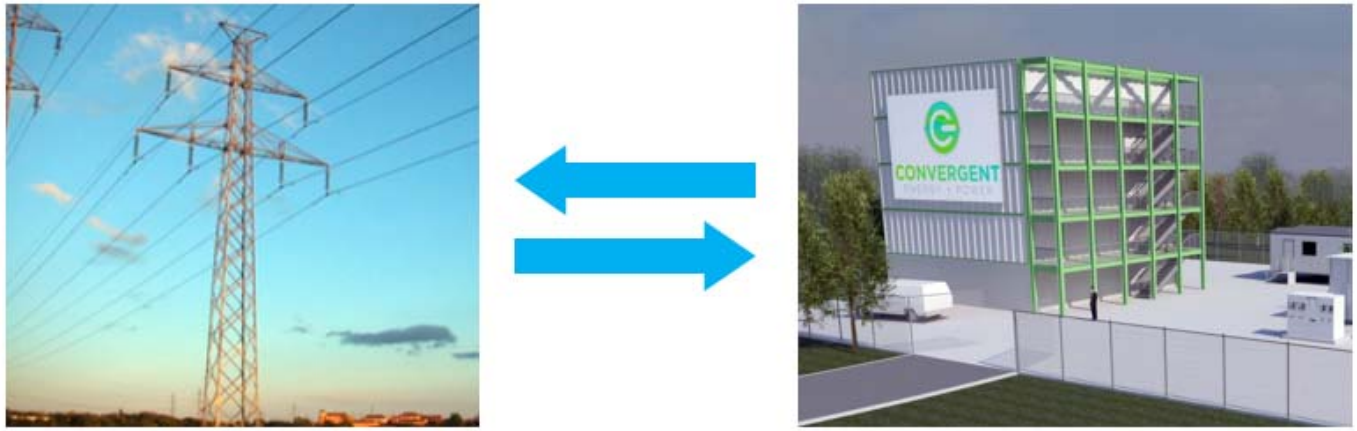
Policy function approximations

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



Energy arbitrage

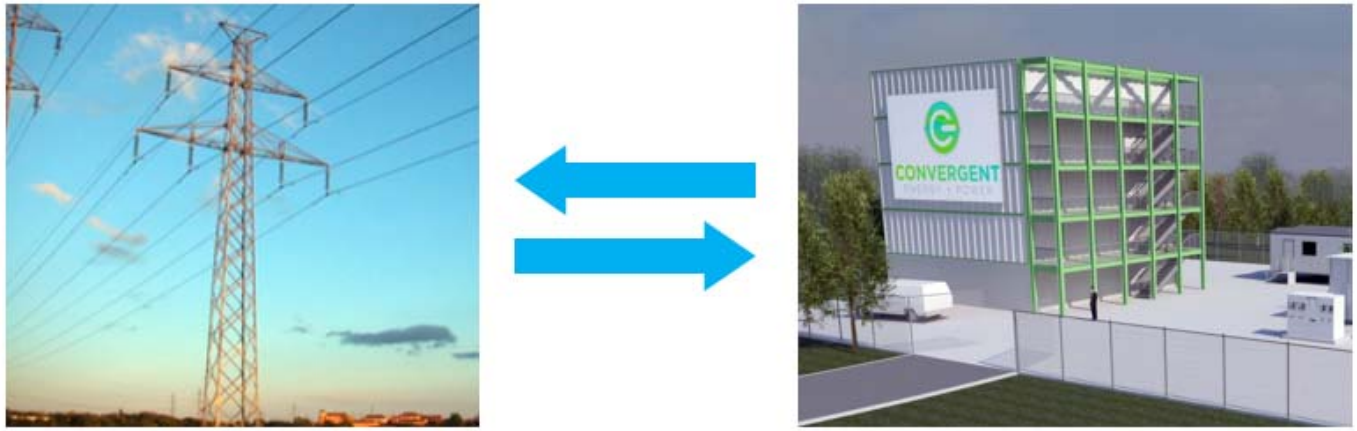
- State variables



  - » R_t =Energy stored in the battery
  - » p_t =Current grid price

Energy arbitrage

- Decision variables



» x_t =How much to sell ($x_t > 0$) or buy ($x_t < 0$) to/from the grid.

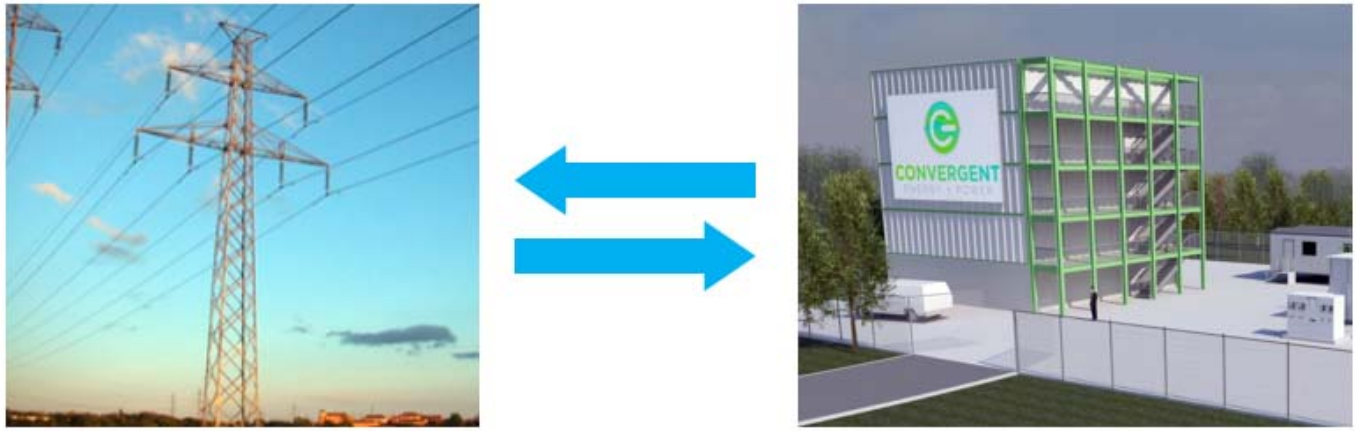
» Constraints:

- $x_t \leq R_t$
- $-x_t \leq R^{max} - R_t$

» Policy: $X^{\pi}(S_t)$

Energy arbitrage

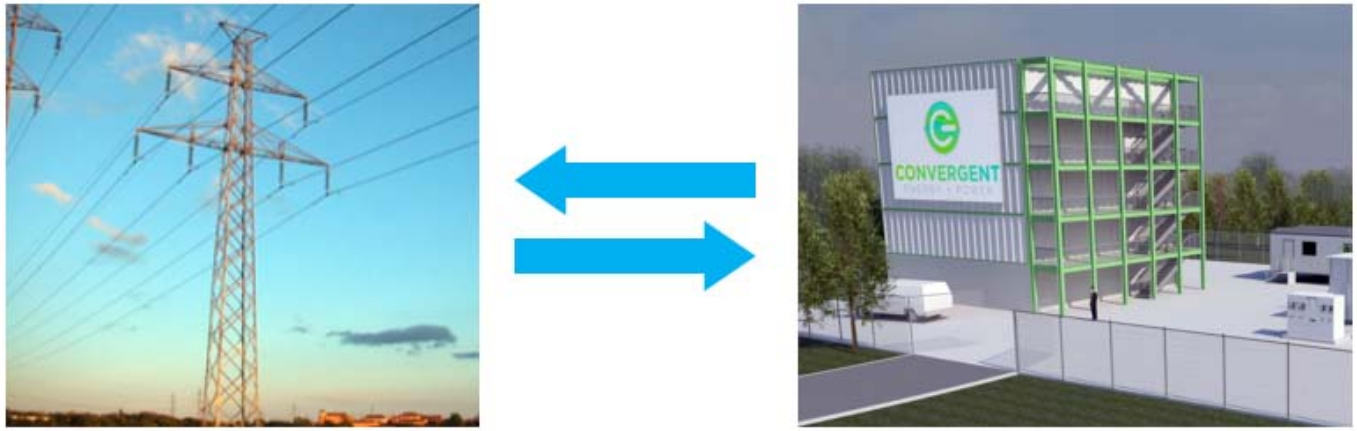
- Exogenous information



» $W_t = \hat{p}_t$ =Change in the electricity price p_t between $(t-1)$ and t.

Energy arbitrage

- Transition function

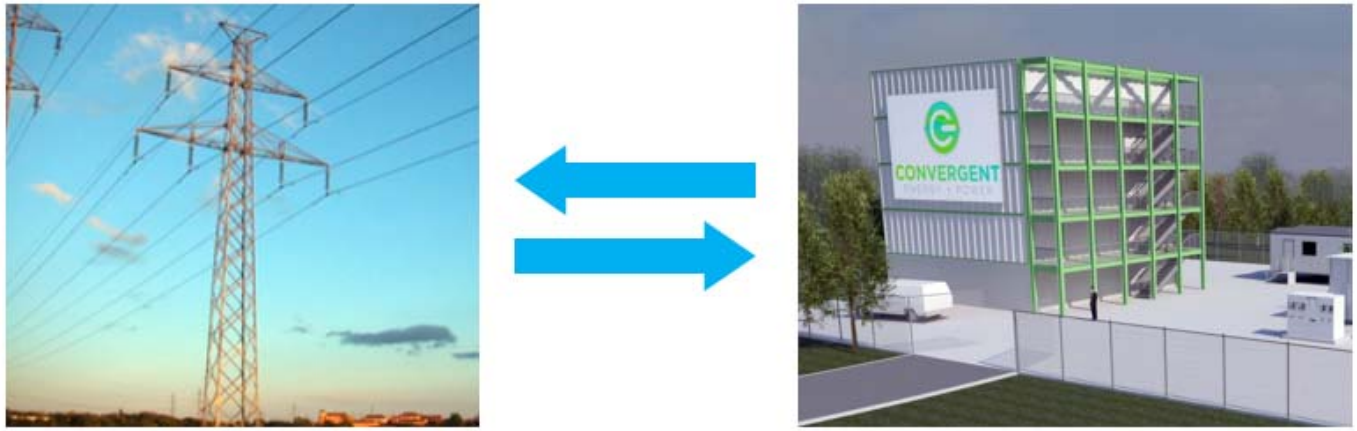


» $p_{t+1} = p_t + \hat{p}_{t+1}$

» $R_{t+1} = R_t - \eta x_t$

Energy arbitrage

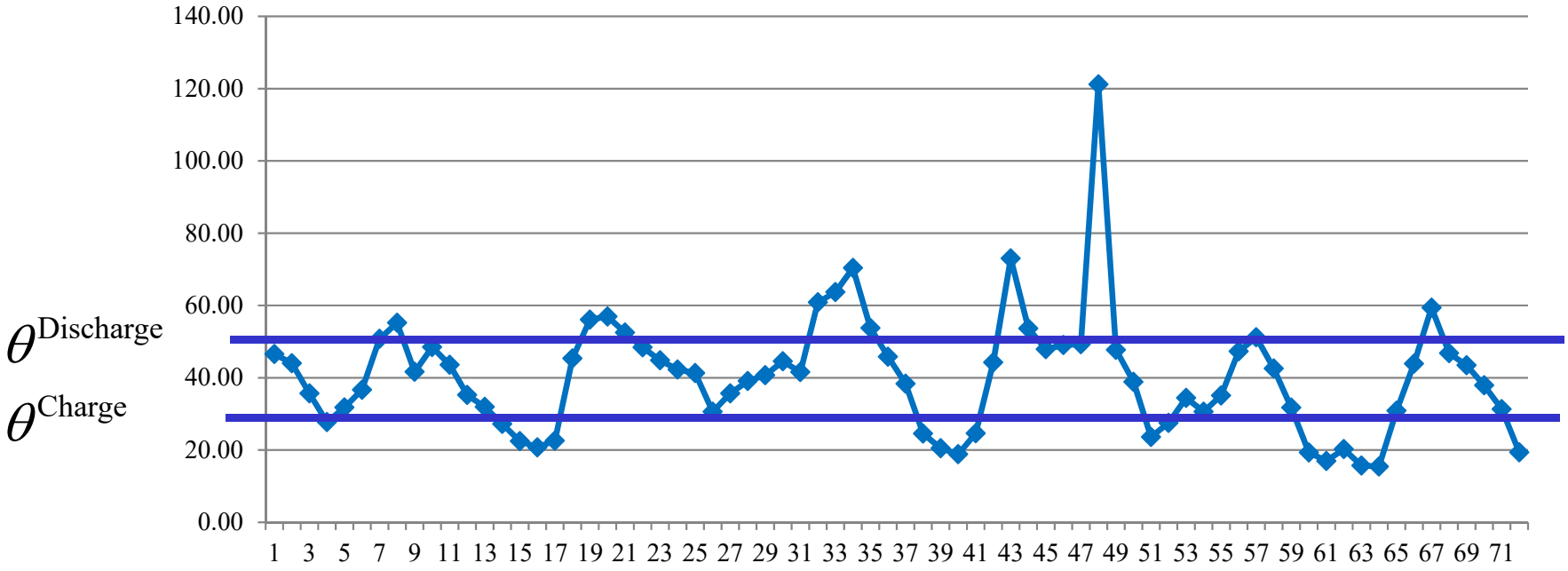
- Objective function



  - » $C(S_t, x_t) = p_t x_t$
  - » $\max_\pi \mathbb{E} \sum_{t=0}^{T} C(S_t, x_t)$

Energy arbitrage

- Buy low, sell high policy



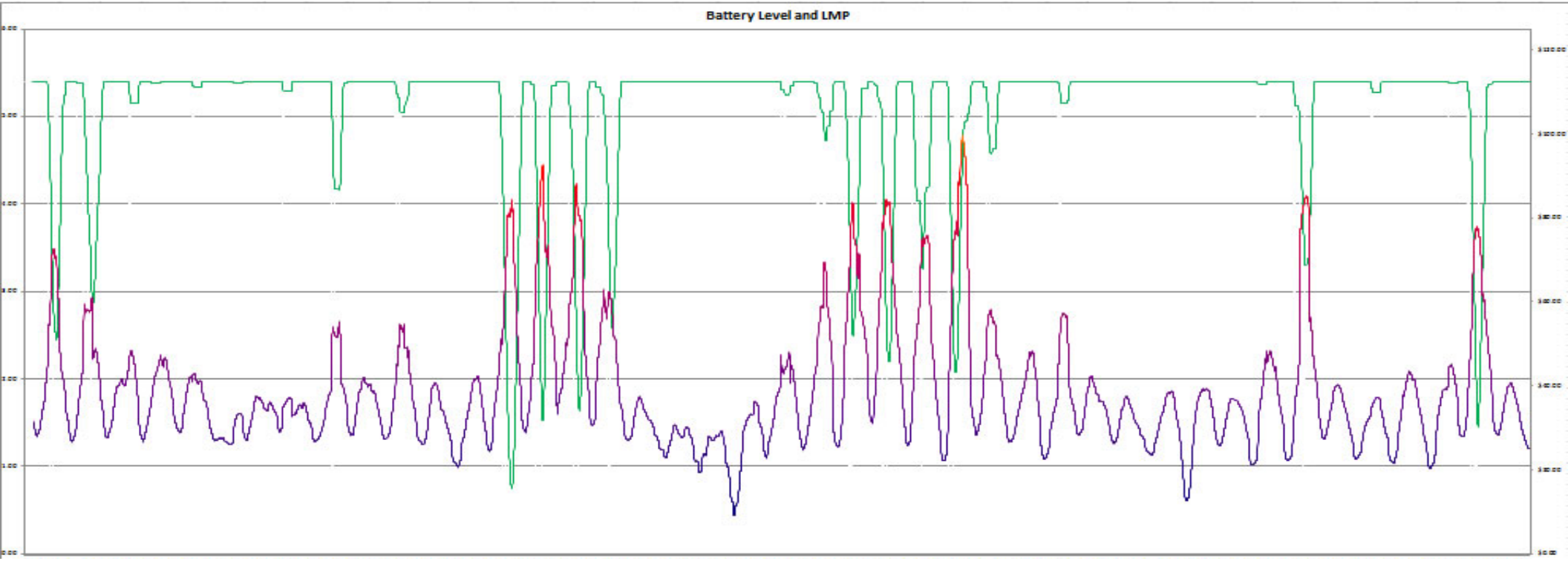
- We have to search for the best values for the policy parameters

$$\theta^{Charge} \text{ and } \theta^{Discharge}.$$

Energy arbitrage

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

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

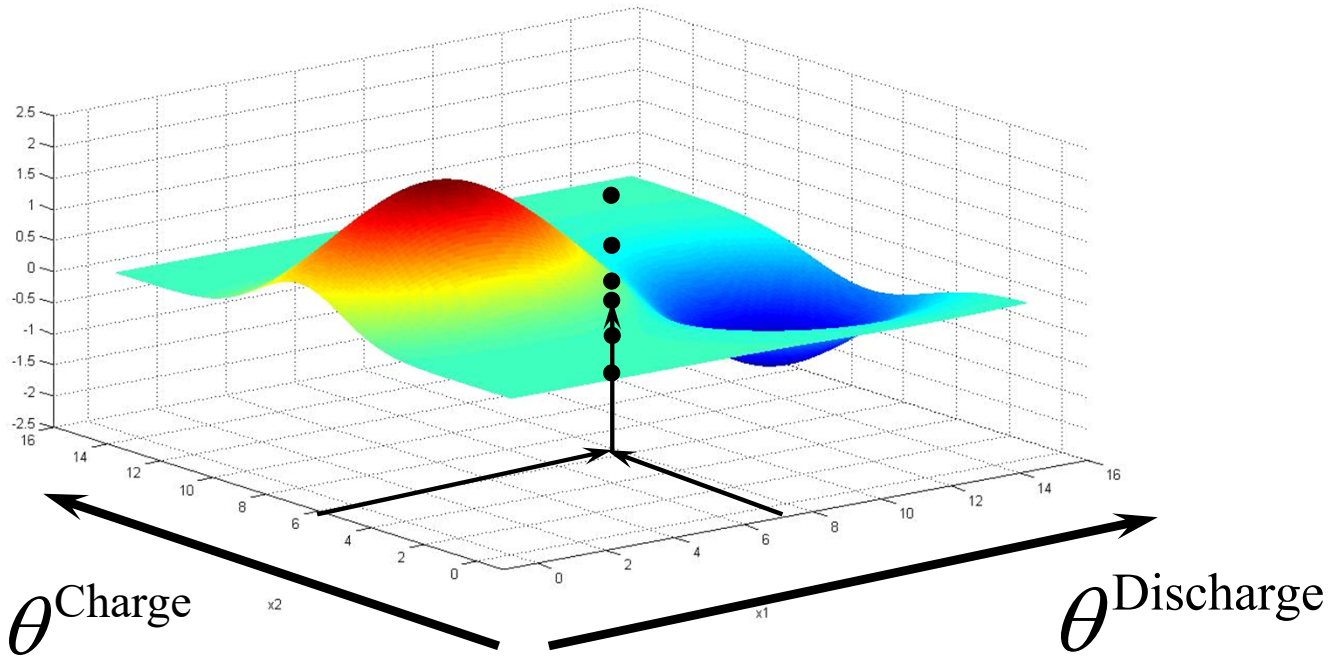


Energy arbitrage

- Finding the best policy
  - » We need to maximize

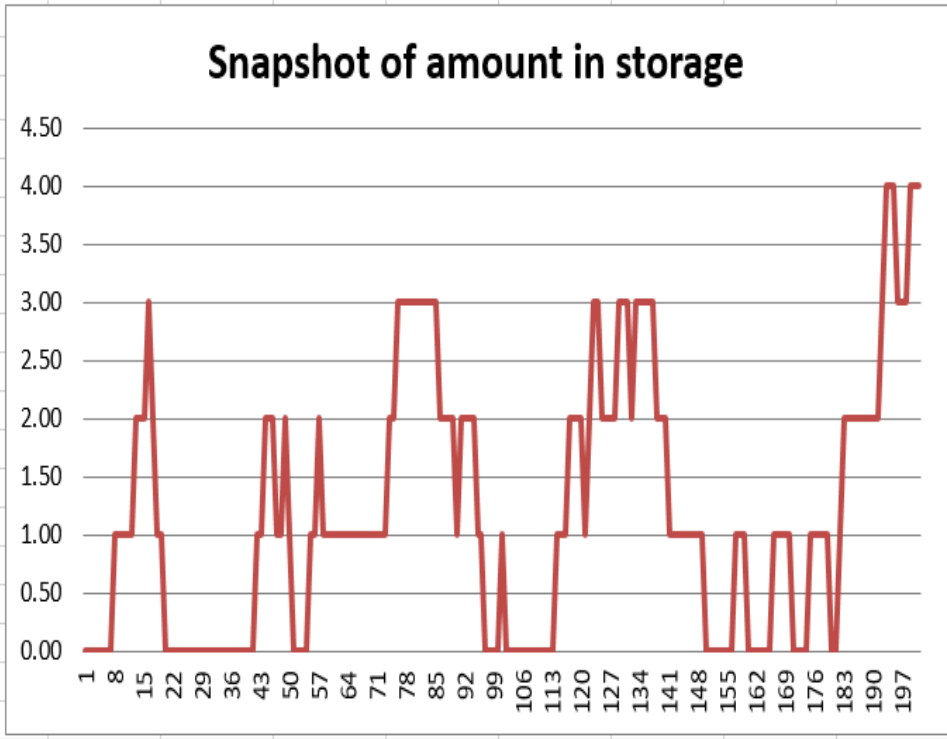
$$\max_\theta F(\theta) = \mathbb{E}\sum_{t=0}^{T} \gamma^t C\left(S_t, X_t^\pi(S_t \mid \theta)\right)$$

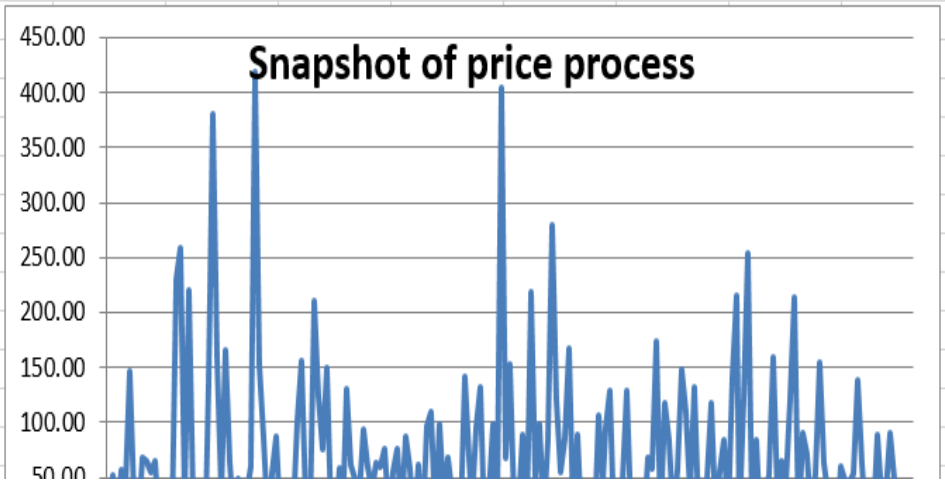
  - » We cannot compute the expectation, so we run simulations:



Energy arbitrage

loss	0.70			Battery size	8.00		
Smoothing	1.00		Buy	10.00			
			Sell	120.00		Total profit/hr	$12.64
					Amt in		
	Time period	Hour of day	Price	Buy-sell	storage	Bought/sold	Revenue
					0.00		
1/1/05	1	1	15.40	0.00	0.00	0.00	0.00
1/1/05	2	2	52.22	0.00	0.00	0.00	0.00
1/1/05	3	3	10.54	0.00	0.00	0.00	0.00
1/1/05	4	4	56.67	0.00	0.00	0.00	0.00
1/1/05	5	5	42.41	0.00	0.00	0.00	0.00
1/1/05	6	6	146.25	-1.00	0.00	0.00	0.00
1/1/05	7	7	31.60	0.00	0.00	0.00	0.00
1/1/05	8	8	6.44	1.00	1.00	1.00	-6.44
1/1/05	9	9	68.47	0.00	1.00	0.00	0.00
1/1/05	10	10	64.59	0.00	1.00	0.00	0.00
1/1/05	11	11	54.43	0.00	1.00	0.00	0.00
1/1/05	12	12	64.21	0.00	1.00	0.00	0.00
1/1/05	13	13	4.64	1.00	2.00	1.00	-4.64
1/1/05	14	14	21.80	0.00	2.00	0.00	0.00
1/1/05	15	15	22.13	0.00	2.00	0.00	0.00
1/1/05	16	16	6.69	1.00	3.00	1.00	-6.69
1/1/05	17	17	229.40	-1.00	2.00	-0.70	160.58
1/1/05	18	18	258.78	-1.00	1.00	-0.70	181.15
1/1/05	19	19	30.19	0.00	1.00	0.00	0.00
1/1/05	20	20	219.78	-1.00	0.00	0.00	0.00
1/1/05	21	21	27.94	0.00	0.00	0.00	0.00
1/1/05	22	22	18.53	0.00	0.00	0.00	0.00
1/1/05	23	23	21.54	0.00	0.00	0.00	0.00
1/1/05	24	24	21.67	0.00	0.00	0.00	0.00





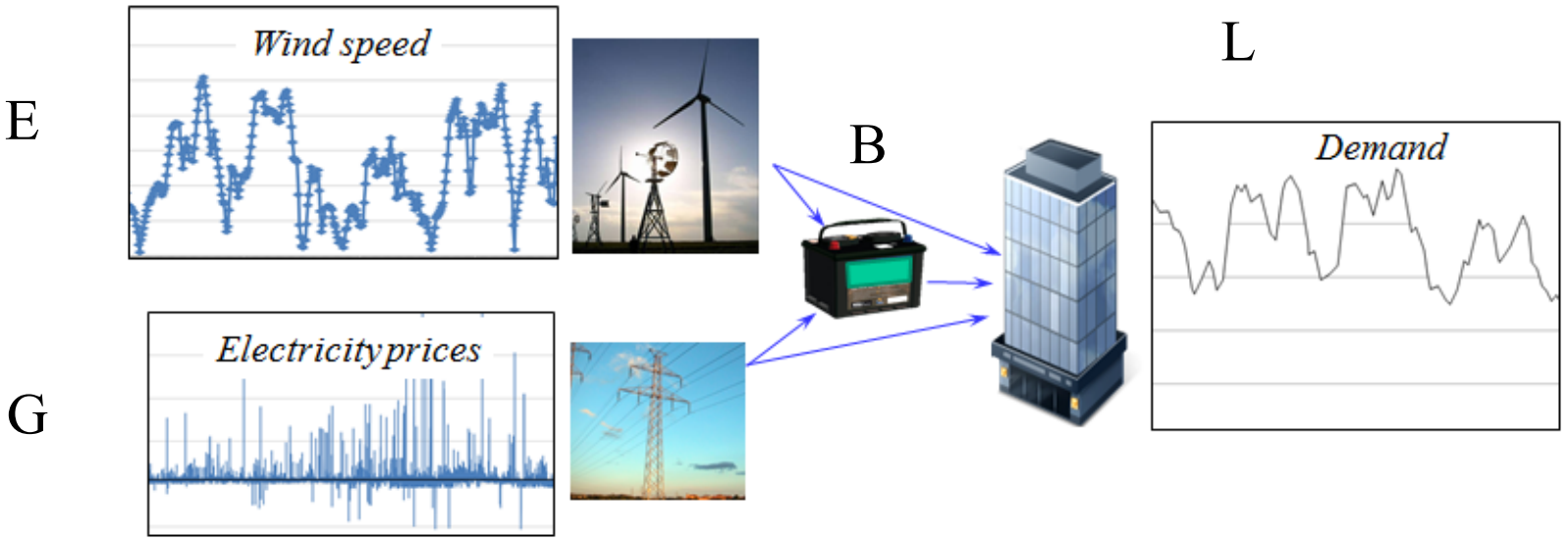
Click on sheet for spreadsheet

An energy storage example

A storage system

An energy storage problem

- State variables

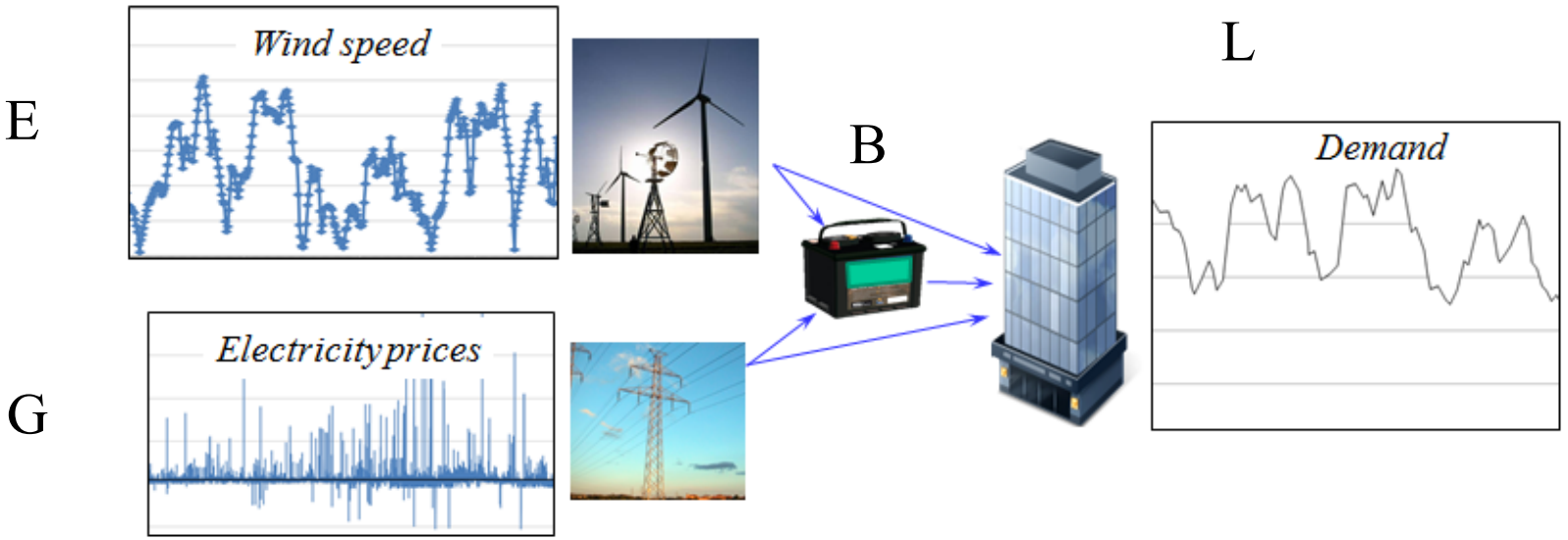


» We will present the full model, accumulating the information we need in the state variable.

» We will highlight information we need as we proceed. This information will make up our state variable.

An energy storage problem

- Decision variables



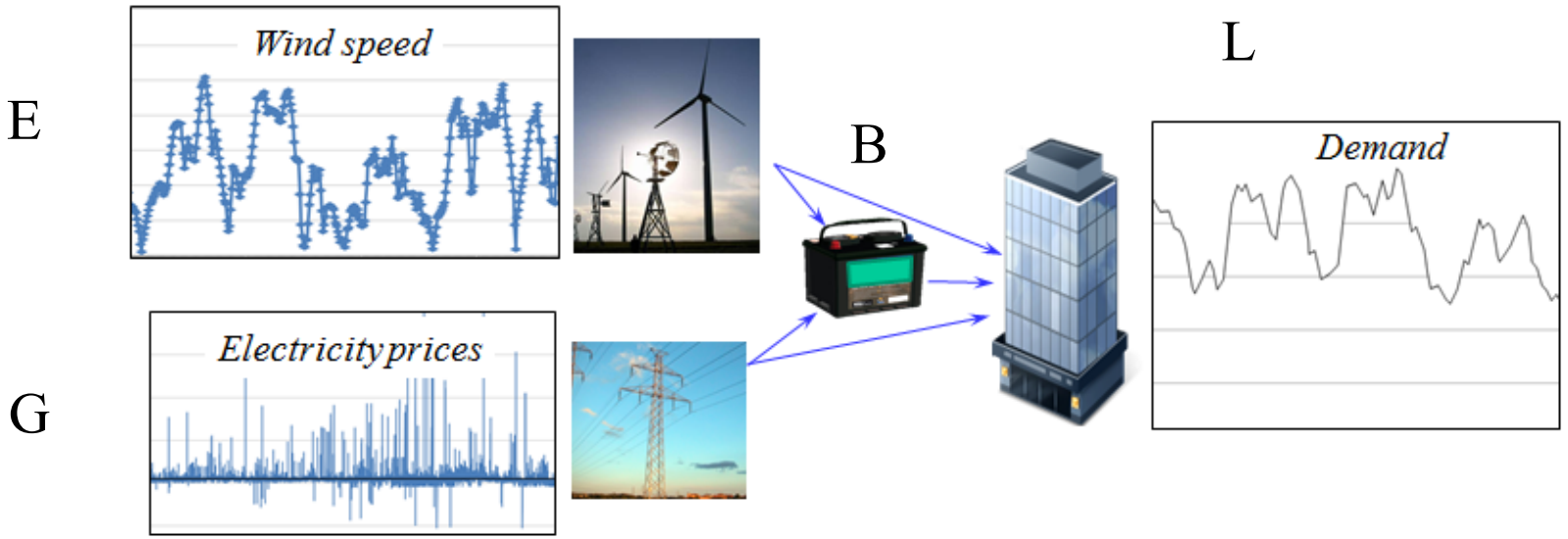
$$x_t = \left(x_t^{EL}, x_t^{EB}, x_t^{GL}, x_t^{GB}, x_t^{BL}, \right)$$

» 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

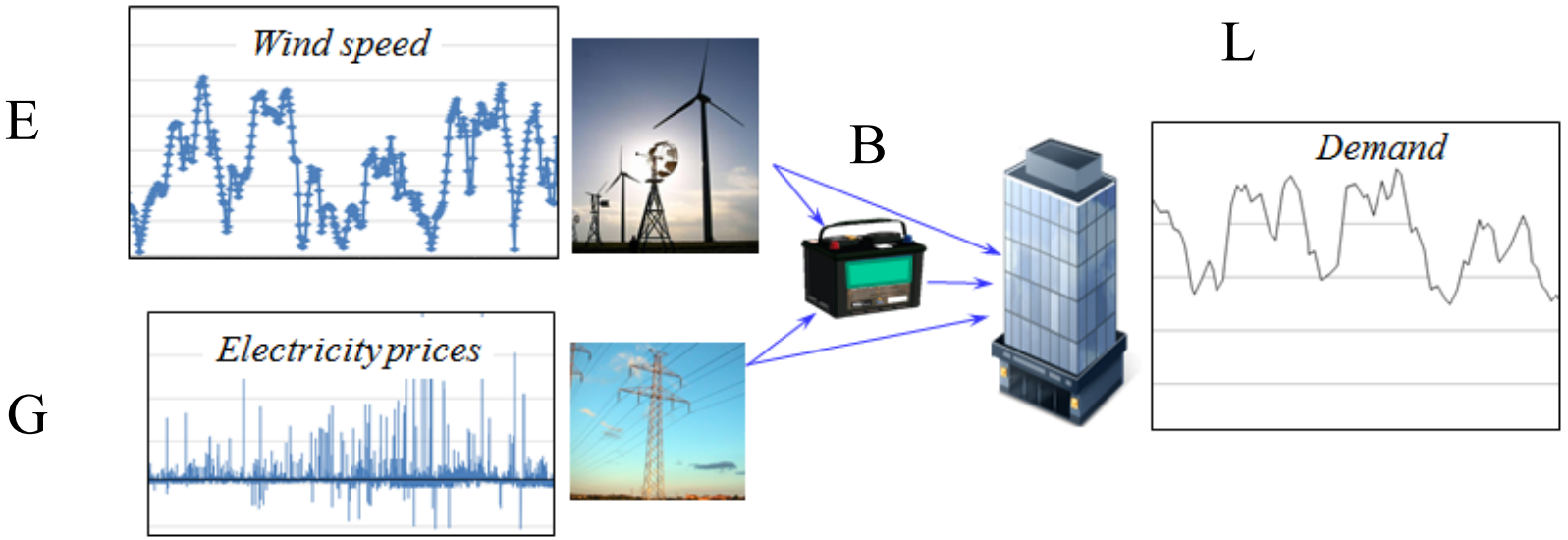
- Exogenous information



$$W_t = \begin{cases} \hat{E}_t = \text{Change in energy from wind between } t-1 \text{ and } t \\ \varepsilon_t^p = \text{Noise in the price process between } t-1 \text{ and } t \\ f_{tt'}^D = \text{Forecast of demand } D_{t'} \text{ provided by vendor at time } t \\ f_t^D = \left(f_{tt'}^D \right)_{t'>t} \quad \text{Provided exogenously} \\ \varepsilon_t^D = \text{Difference between actual demand and forecast} \end{cases}$$

An energy storage problem

- Transition function



$$E_{t+1} = E_t + \hat{E}_{t+1}$$

$$p_{t+1} = \theta_0 p_t + \theta_1 p_{t-1} + \theta_2 p_{t-2} + \varepsilon^p_{t+1}$$

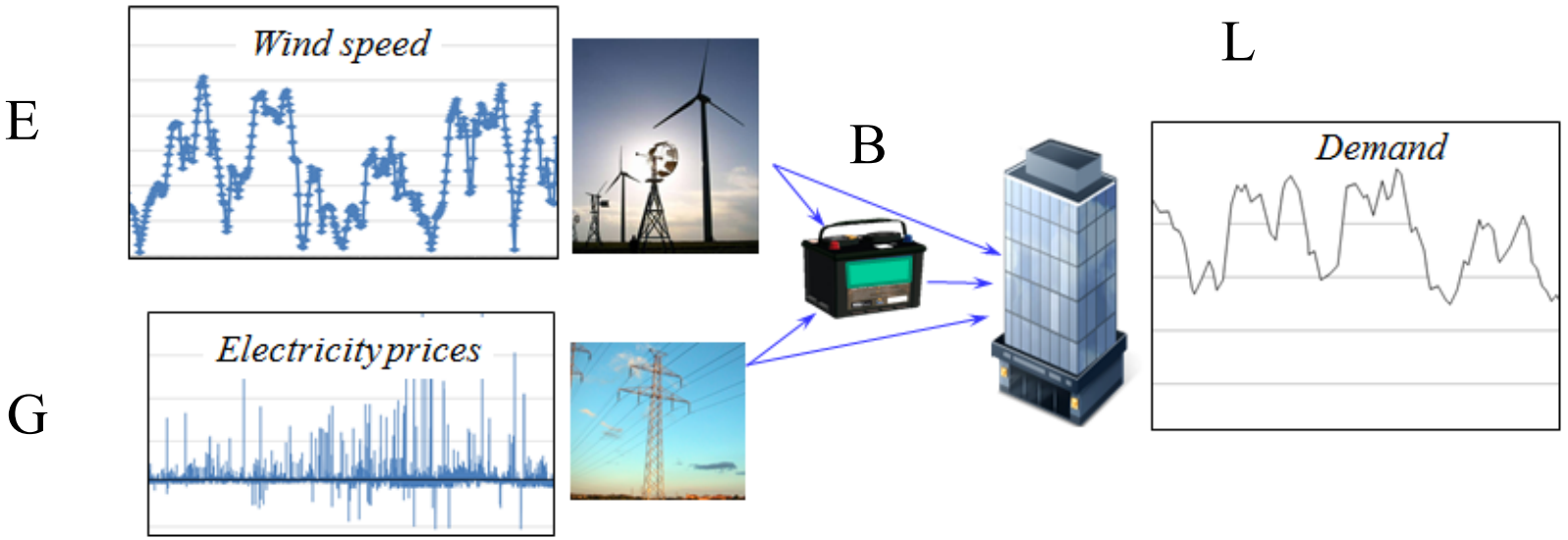
$$D_{t+1} = f^D_{t,t+1} + \hat{D}_{t+1}$$

$$f^D_{t+1} = \text{Provided exogenously}$$

$$R^{battery}_{t+1} = R^{battery}_t + x_t$$

An energy storage problem

- Objective function



$$C(S_t, x_t) = p_t \left(x_t^{GB} + x_t^{GL} \right)$$

$$\min_\pi \mathbb{E} \left\{ \sum_{t=0}^{T} C_t(S_t, X_t^\pi(S_t)) \mid S_0 \right\}$$

An energy storage problem

- State variables
  - » Cost function

    p_t = Price of electricity

  - » Decision function

    Constraints:

$$
\begin{aligned}
x_t^{EL} + x_t^{EB} &\leq E_t \\
(x_t^{GL} + x_t^{EL} + x_t^{BL}) &= L_t \\
x_t^{BL} &\leq R_t
\end{aligned}
$$

    f_t^L = Needed if we use a lookahead policy

  - » Transition function

$$p_{t+1} = \theta_0 p_t + \theta_1 p_{t-1} + \theta_2 p_{t-2} + \varepsilon_{t+1}^p$$

$$S_t = \left(R_t, E_t, L_t, (p_t, p_{t-1}, p_{t-2}), f_t^L\right)$$

An energy storage example

With passive learning

An energy storage problem

- Types of learning:
  - » No learning ($\theta's$ are known)

$$p_{t+1} = \theta_0 p_t + \theta_1 p_{t-1} + \theta_2 p_{t-2} + \varepsilon^p_{t+1}$$

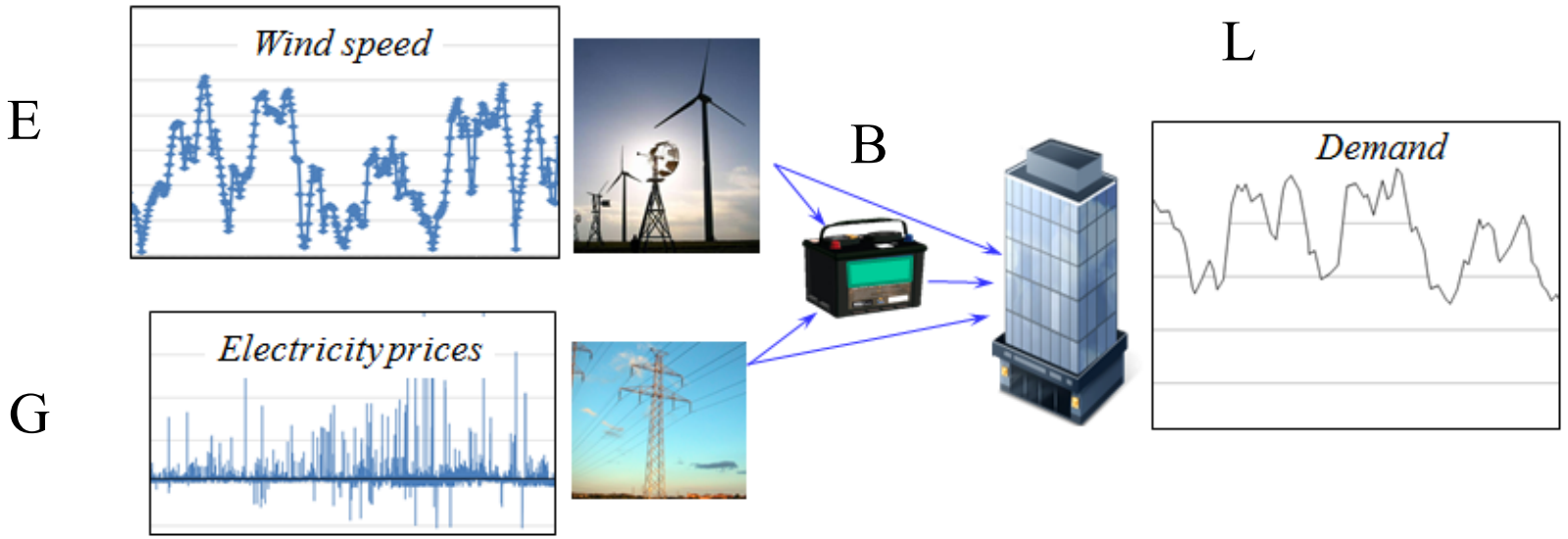
  - » Passive learning (learn θs from price data)

$$p_{t+1} = \bar{\theta}_{t0} p_t + \bar{\theta}_{t1} p_{t-1} + \bar{\theta}_{t2} p_{t-2} + \varepsilon^p_{t+1}$$

This is an example of learning a transition function.

An energy storage problem

- Transition function



$$E_{t+1} = E_t + \hat{E}_{t+1}$$

$$p_{t+1} = \bar{\theta}_{t0} p_t + \bar{\theta}_{t1} p_{t-1} + \bar{\theta}_{t2} p_{t-2} + \varepsilon^p_{t+1} = (\bar{\theta}_t)^T \bar{p}_t + \varepsilon^p_{t+1} \quad \bar{p}_t = \begin{pmatrix} p_t \\ p_{t-1} \\ p_{t-2} \end{pmatrix}$$

$$D_{t+1} = f^D_{t,t+1} + \varepsilon^D_{t+1}$$

$$R^{battery}_{t+1} = R^{battery}_t + x_t$$

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:

$$S_t = \left(\underbrace{E_t, L_t, R_t}, \underbrace{(p_t, p_{t-1}, p_{t-2})}, f_t^D, \underbrace{(\bar{\theta}_t, B_t)} \right)$$

$$\begin{aligned} x_t^{EL} + x_t^{EB} &\leq E_t \\ (x_t^{GL} + x_t^{EL} + x_t^{BL}) &= L_t \\ x_t^{BL} &\leq R_t \end{aligned}$$

  - » Transition function

$$p_{t+1} = \bar{\theta}_{t0} p_t + \bar{\theta}_{t1} p_{t-1} + \bar{\theta}_{t2} p_{t-2} + \varepsilon_{t+1}^p$$

$$D_{t+1} = f_{t,t+1}^D + \varepsilon_{t+1}^D$$

$$\bar{\theta}_{t+1} = \bar{\theta}_t - \frac{1}{\gamma_{t+1}} B_t \phi(x_t) \varepsilon_{t+1}$$

An energy storage example

With active learning

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")

Buy/sell decisions

$$p_{t+1} = \bar{\theta}_{t0} p_t + \bar{\theta}_{t1} p_{t-1} + \bar{\theta}_{t2} p_{t-2} + \bar{\theta}_{t3} x^{GB}_t + \varepsilon^p_{t+1}$$

Our decisions influence the prices we observe, which helps with learning.

An energy storage problem

- Notes:
  - » When we introduce active learning, the state variable is the same as with passive learning.
  - » This becomes a very rich form of continuous bandit problem.
  - » What is likely to change is the policy. Recall from earlier when we were doing pricing while learning a demand function that we designed policies that encouraged broader exploration.
  - » We are going to accomplish this by insisting that x_t is a function of the state S_t rather than the sample path ω.

What is next?

What is next?

- Now that we have our basic modeling framework, we have two major challenges:
  - » Modeling uncertainty
  - » Designing policies