1cm240: AI for logistics and its interfaces
Lecture 1

07/09/2021

Dr. Albert Schrotenboer, Assistant Professor

School of Engineering, Operations, Planning, Accounting & Control Group

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Lecturers

Albert Schrotenboer
Assistant Professor @ OPAC

- Transportation and Logistics
- Mathematical programming
- Algorithm design
- AI, Machine Learning, Data

Christina Imdahl
Assistant Professor @ OPAC

- Machine Learning
- Operations Management
- Inventory Control
- AI, Machine Learning, Data

We are currently looking to accept new master students as a mentor

Course structure

- Lectures on Tuesday (10:45-12:30)
  - In person
  - Discusses essential knowledge for solving the case

- Instructions on Friday (13:30 - 17:15)
  - In person
  - Presentations in **Week 2, 5, 9** in the first hour.
  - Other weeks, exercises (in Python) to master course content + working on case.

- Office hours Tuesday (09:00 – 10:30)
  - Please make an appointment via email.
  - In-person or online

Schedule

Week	Content
1	Introduction + Framework Stochastic Optimization
2	Dynamic Vehicle Routing I
3	Dynamic Vehicle Routing II
4	**CASE**
5	Data-driven methods in Inventory Management
6	Reinforcement learning and heuristic approaches with applications to Inventory Problems (I)
7	Reinforcement learning and heuristic approaches with applications to Inventory Problems (I)
8	**CASE**

Challenge-based learning

- Two case studies / challenges
  - Week 1 – 4: Transportation
  - Week 5 – 8: Inventory Control

- Each 50% of the total course load

- Teams of 3-4 students.

CASE STUDY 1 (50% of the course)	**Grade share**	**Deadline**
Assignment 1 + Pitch Presentation	15%	Week 2
Final report*	85%	Week 4

CASE STUDY 2 (50% of the course)	**Grade share**	**Deadline**
Final report*	100%	Week 8

Deliverables and grading.

- For each case study a report and a working solution method.
    - Best performing group will gets 0.5 bonus on grade (for each case study).
- Case study 1 contains an additional assignment with a `pitch – presentation’
- Groups need to present their work of a single case study.
    - Assessment is included in report grade
- *Presentations over both case studies need to be balanced. Indicate your preference via email* ***ASAP****.*

CASE STUDY 1 (50% of the course)	**Grade share**	**Deadline**
Assignment 1 + Pitch Presentation	15%	Week 2
Final report*	85%	Week 4

CASE STUDY 2 (50% of the course)	**Grade share**	**Deadline**
Final report*	100%	Week 8

Course goals

- Provide an overview of classic optimization techniques for stochastic sequential decision making in Transportation and Inventory Control.

- Recognize how optimization techniques for stochastic sequential decision making can be improved by making use of recent advances in AI.

- Trade off the usefulness of data-driven optimization versus traditional optimization methods, and recognize problem instances where this is promising.

- Provide an overview of anticipatory methods for planning in transportation and inventory control based on the unified framework for stochastic optimization (Myopic policies, scenario-based sampling and consensus functions, and some potential applications of (deep) reinforcement learning).

- Independently implement problem-specific data-driven optimization techniques (e.g. reinforcement learning) to operational settings and analyze and interpret its output

Course goals in laymen terms.

simple language that anyone can understand

- Think critically about when, where, how, and what AI can be done
- Get some experience in solving optimization problems with uncertainty
- Appreciate what AI can do, and what it cannot do.

Chapter -1

A gentle AI introduction

Artificial Intelligence

artificial

adjective

UK /ˌɑː.tɪˈfɪʃ.əl/ US /ˌɑːr.t̬əˈfɪʃ.əl/

B2

made by people, often as a copy of something natural:

- *clothes made of artificial fibres*
- *an artificial heart*
- *an artificial lake*
- *artificial fur/sweeteners/flowers*

intelligence

noun

UK /ɪnˈtel.ɪ.dʒəns/ US /ɪnˈtel.ə.dʒəns/

intelligence *noun* **(ABILITY)**

B2 [U]

the ability to learn, understand, and make judgments or have opinions that are based on reason:

- *an intelligence test*
- *a child of high/average/low intelligence*
- *It's the intelligence of her writing that impresses me.*

Logistics and its interfaces

We know supply chains: from initial sales, support throughout the life cycle of products, and reverse flows of disposed products. Uniquely, we combine all the required expertise to analyze and optimize supply chain functions from acquisition of raw materials to the satisfaction of ultimate consumer demands. Through this integrative, multidisciplinary approach, advanced methodologies, and our intimate relation with the industry we deliver the highest standards in science and an impressive track record of successful implementations.

Four examples how AI is transforming logistics

- AI unlocks the true potential of Big Data in Logistics
- Usage of robotics can substitute workforce in Logistics
- AI boosts logistic automation of warehouse management and optimizes supply chain planning
- AI in Logistics promotes autonomous vehicles

But..., Really?

- AI unlocks the true potential of Big Data in Logistics
  - Which data !?
- Usage of robotics can substitute workforce in Logistics
  - For our peak capacity in the week before Christmas !?
- AI boosts logistic automation of warehouse management and optimizes supply chain planning
  - Industrial Engineering does not exist ?!
- AI in Logistics promotes autonomous vehicles
  - Learning by accidents !?

Decision making under uncertainty

- Industrial Engineers seek to improve or optimize decision-making in practice, e.g. in supply chain management.

- Typically, we seek a policy or decision rule that acts in a stochastic environment

- Typically, we evaluate the quality of this policy by relying on a model/simulation

- AI / MDP/ SSDP allows to use that simulation to structurally IMPROVE the policy, instead of only evaluating its quality

Actual content of today's lecture

- Introduction to the unified framework for stochastic optimization
  - Stochastic Sequential Decision Problems

- Four classes of policies to solve SSDPs

- Description of the first assignment

- Description of the case-study

Chapter 0:

Communication, mathematical style, time, solution approach strategy

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 interpretable as possible.
- Follow consistent, standard conventions.
- Organize your variables into natural groupings:

Notation of variables

- A basic variable - lower case and script: x
- Use subscripts to identify elements of a vector: x_{ij}, x_{ijt}
- Use superscripts to create different flavors of a variable: x^p, x^s, x^h

- *Never* use variables with more than one letter: CP, TC
  - Might be suitable for a business audience
- *Always* be consistent and follow standard conventions.
  - a, b, c, d parameters (time windows)
  - α, β, γ physical elements (e.g. speeds, ratios)
  - Use hats/bars to indicate estimations or predictions.

- When you have multiple subscripts:
  - Order the subscripts that as they most likely appear in a summation
  - Always model discrete time periods as a subscript
- Use Sets: $\sum_{i \in V} x_{ij}$

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

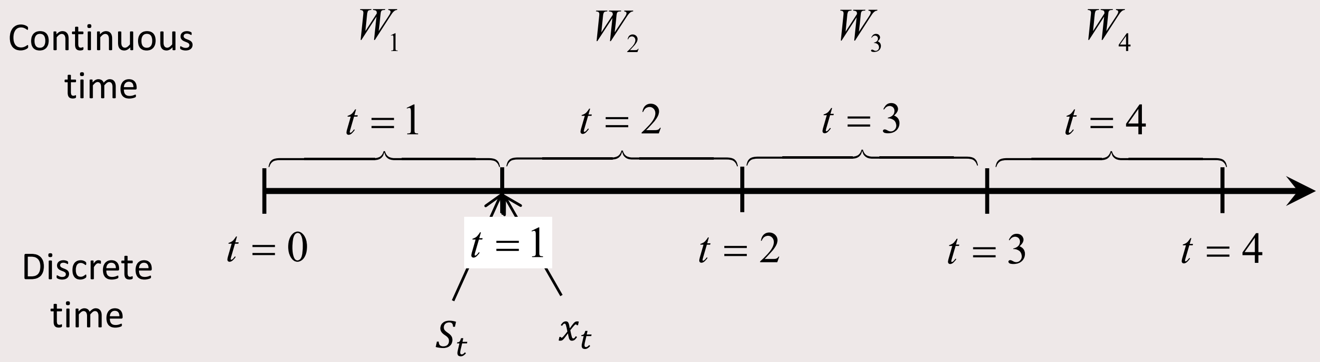
Combinations

- 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.

How to attack a problem under uncertainty

- Model the uncertainty (there is extra material on that on the supplementary materials of this weeks lecture)
  - Next week we focus on predicting uncertainty

- What information do we want to utilize to base our decision on

- The more information, the less computational tractable but the better a decisions quality

    - **Model, Model, Model, Model, Model, Model** solve, solve…

Chapter 1:

Stochastic Sequential Decision Problems and the unified framework for stochastic optimization

What is a Stochastic Sequential Decision Problem

- We consider environments that basically consists of the following sequence:

Information, decision, information, decision, information, decision…..

- Each decision is difficult on its own

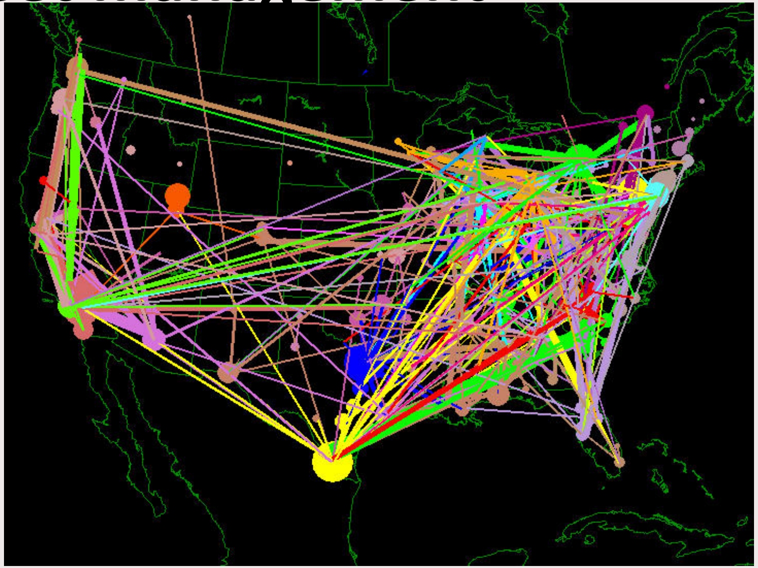
- Each decision impacts future decisions (and possibly information)

Some examples of SSDPs

- At offshore wind farms, how much electricity to buy and sell from the market, and how much to store in a battery or hydrogen tank for later use
- In retail environments, how much goods to send to each store each day; sending more today means sending less tomorrow, but needs coordination among 500 stores (or more)
- In local delivery platforms; stores send products to customers home using cargo-bikes. These bikes need to be routed in real-time.

Much of the following material is adapted and/or taken from the excellent free resources by © 2019 Warren Powell Google his name, he is a famous professor at Princeton University. His webpages provide many more resources then we can cover in this course

Fleet management



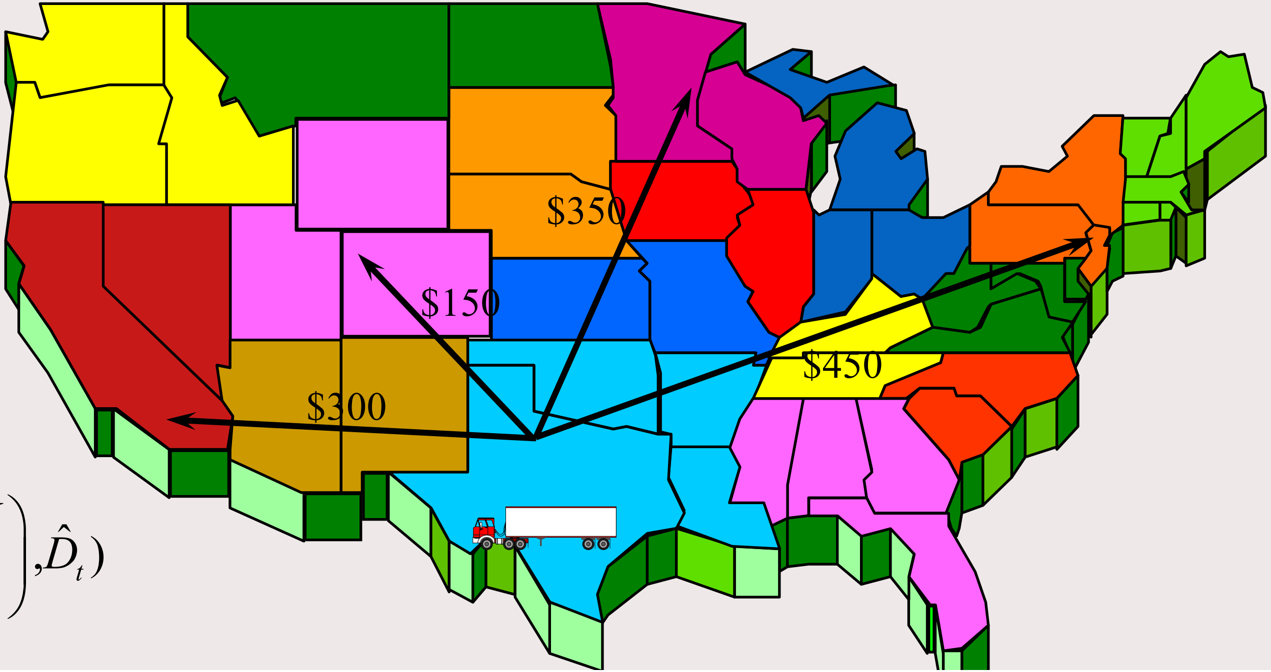
Fleet management problem

Optimize the assignment of drivers to loads over time.

Tremendous uncertainty in loads being called in

Nomadic trucker illustration

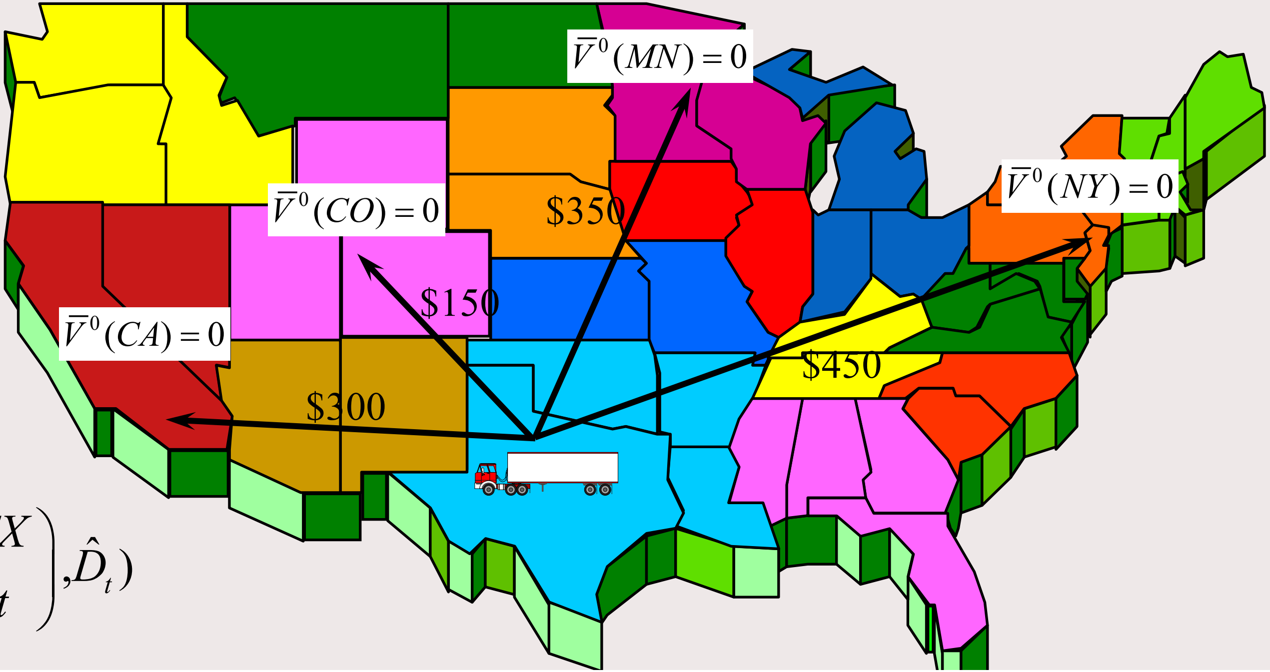
Pre-decision state: we see the demands



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

Nomadic trucker illustration

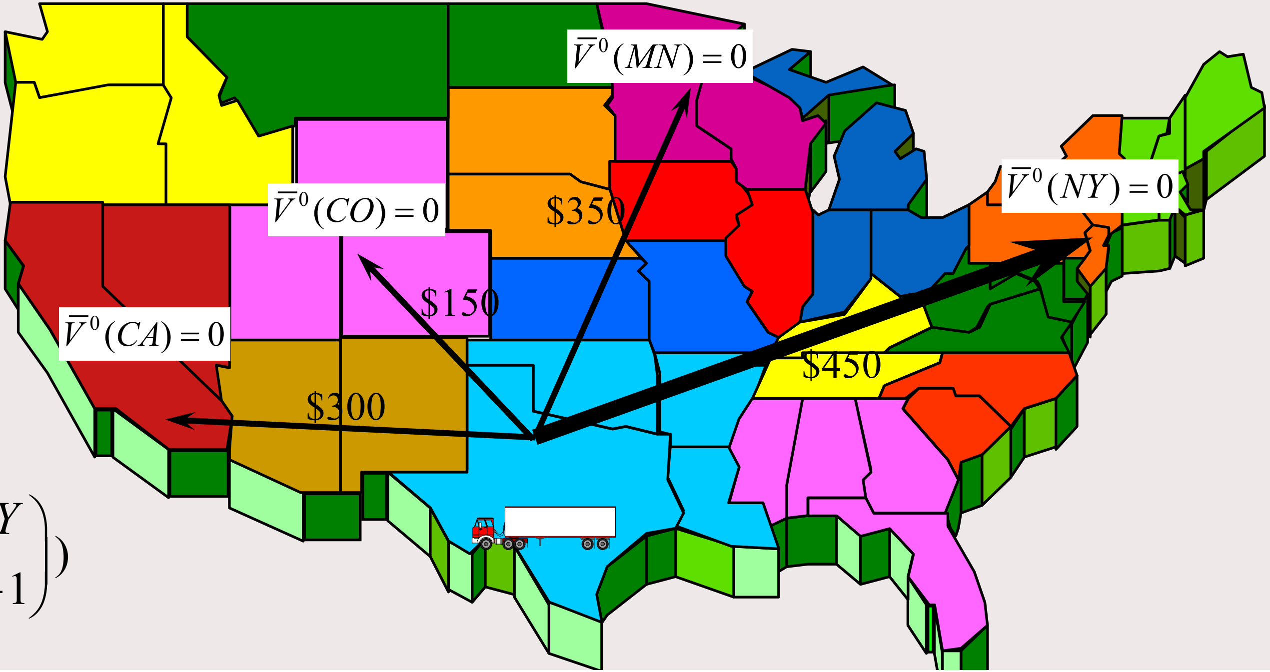
We use initial value function approximations…



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

Nomadic trucker illustration

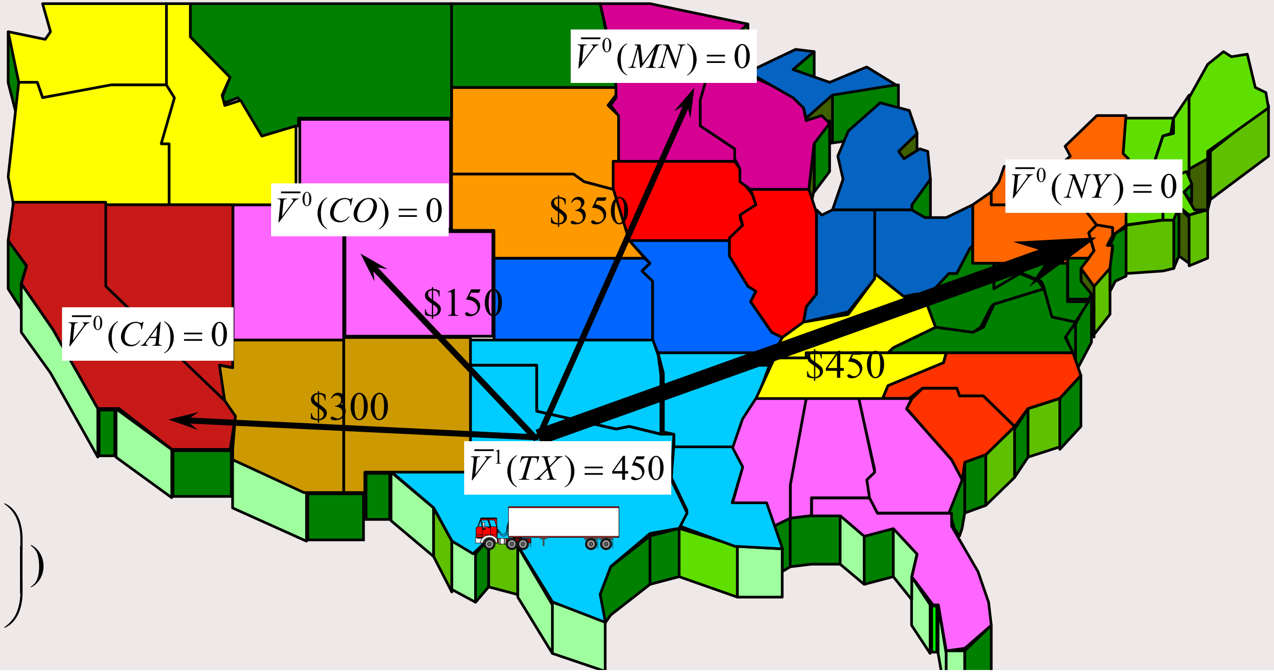
... and make our first choice: x^1



$$S_t^x = \left(\begin{pmatrix} NY \\ t+1 \end{pmatrix}\right)$$

Nomadic trucker illustration

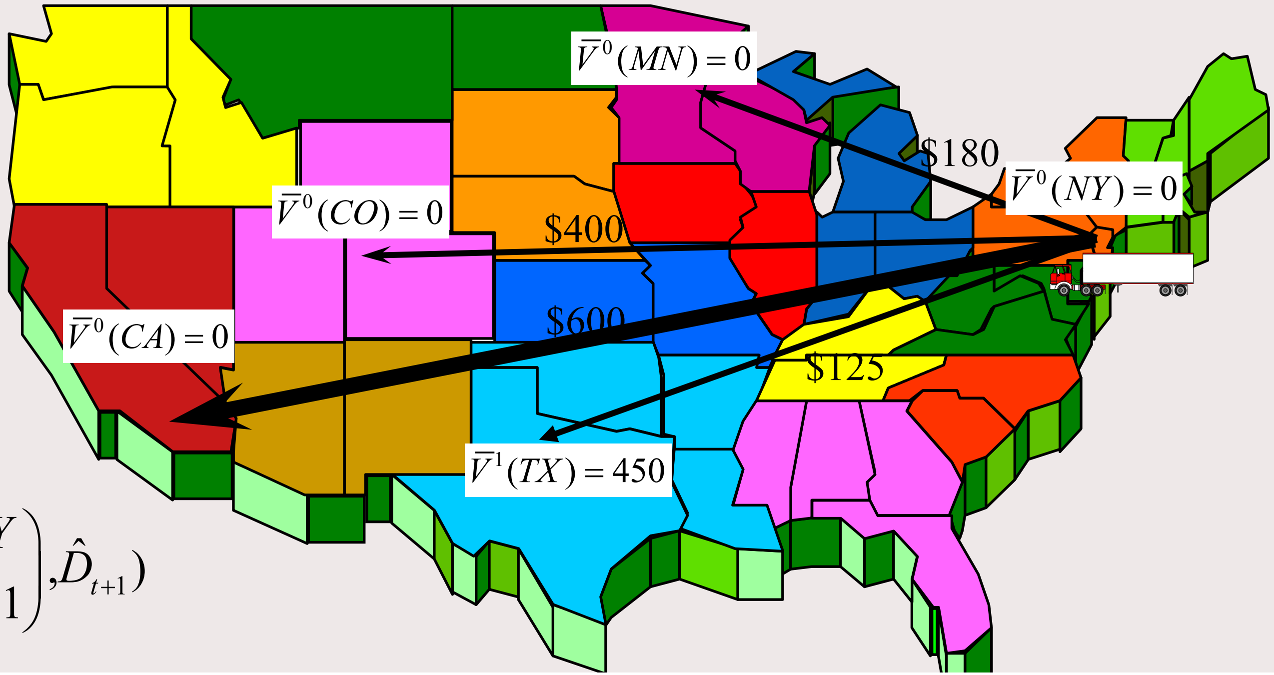
Update the value of being in Texas.



$$S_t^x = \left(\begin{pmatrix} NY \\ t+1 \end{pmatrix}\right)$$

Nomadic trucker illustration

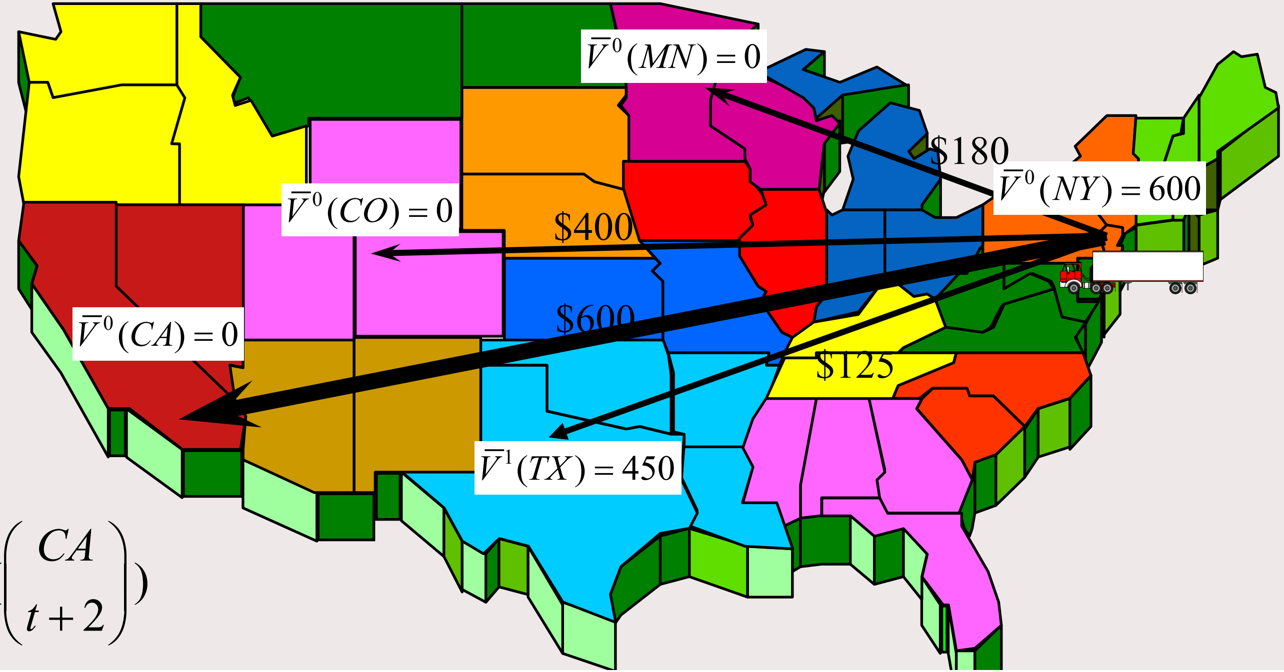
Now move to the next state, sample new demands and make a new decision



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

Nomadic trucker illustration

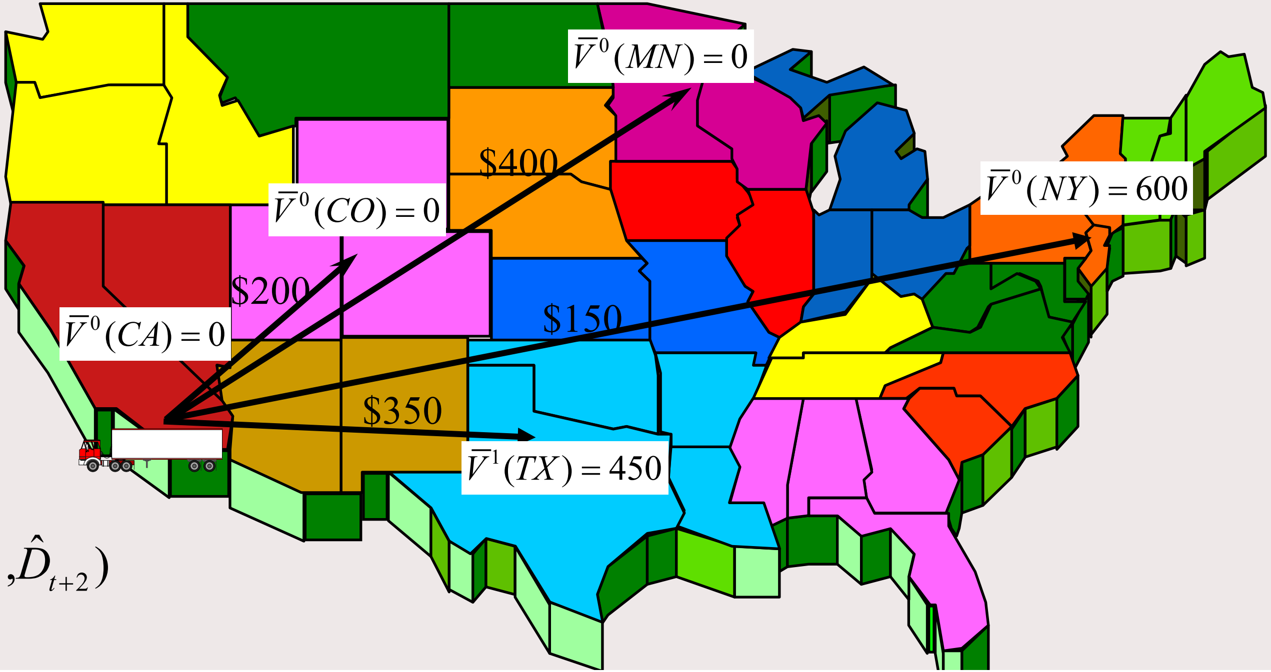
Update value of being in NY



$$S^x_{t+1} = \left(\begin{pmatrix} CA \\ t+2 \end{pmatrix}\right)$$

Nomadic trucker illustration

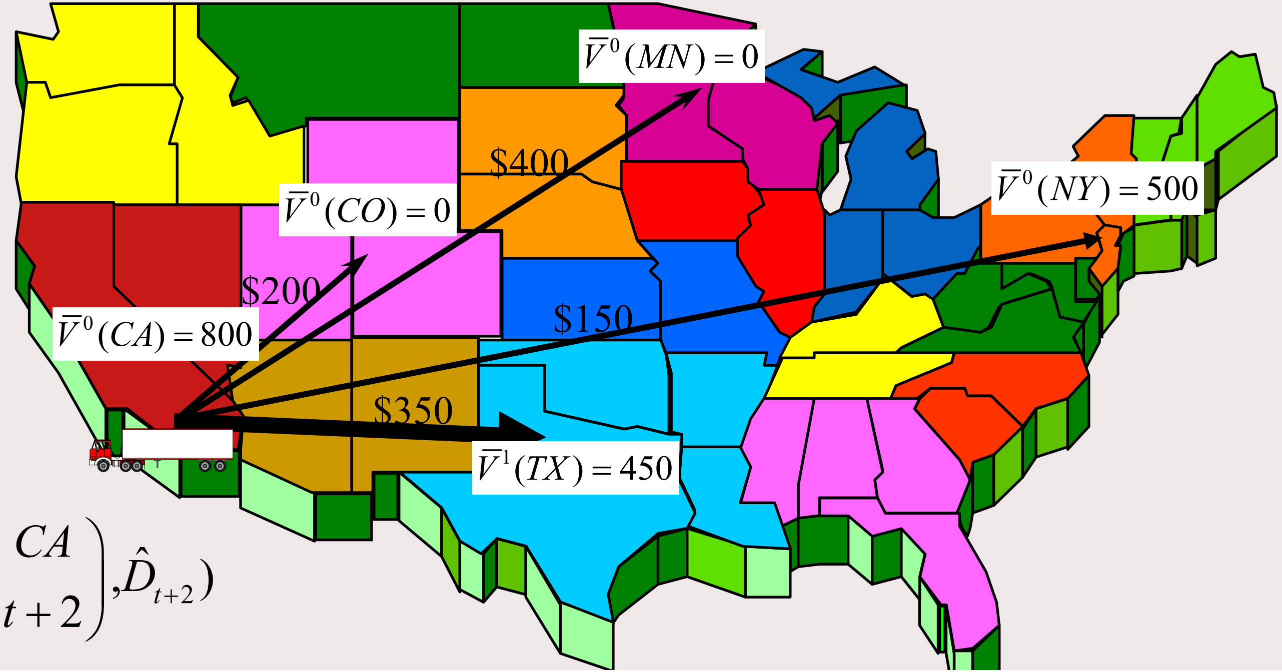
Move to California.



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

Nomadic trucker illustration

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



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

TU/e

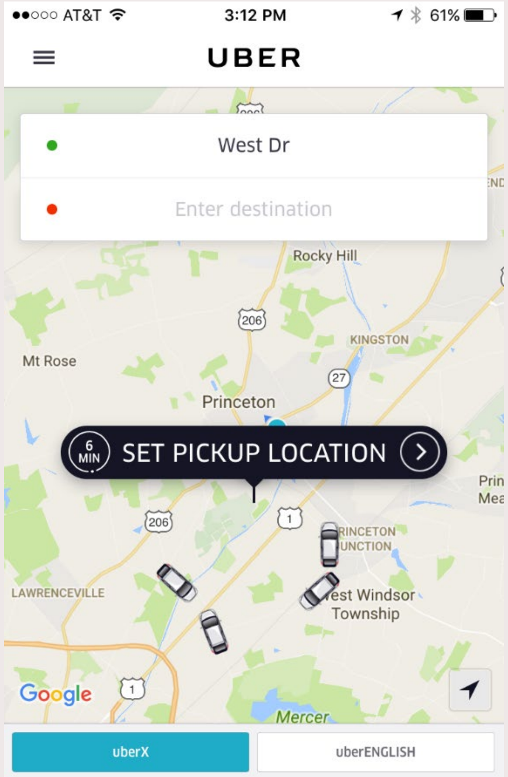
Fleet management

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, …)



Stochastic Sequential Decision Problems

- General name and framework for making decision under uncertainty, abbreviated to SSDP

- Each SSDP is modeled along **five** fundamental directions:

- State variables $S_k \in S$
- Decision variables $x_k(S_k) \in X(S_k), X^{\pi}(S_t) := x_k(S_k), \pi \in \Pi,$
- Exogenous information $W_{k+1}\,(S_k, x_k)$
- Transition function $S^M(S_k, X^{\pi}(S_t), W_{k+1}) \; := S_{\{k+1\}}$
- Objective function reward: $C(S_k, x_k(S_k), W_{k+1}$

  objective: $\min_{\pi \in \Pi} E \; \sum_{t \in T} C\big(S_t, X^{\pi}(S_t)\big) |\; S_0$

The state variable

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.

The decision variable

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 $\theta \in \Theta^f$.

Exogenous information:

$$W_t = \text{New information that first became known at time } t$$

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

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

$$\text{New drivers being hired to the network}$$

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

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

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

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.

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"
"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.

The objective function

Dimensions of objective functions

» Type of performance metric

» Final cost vs. cumulative cost

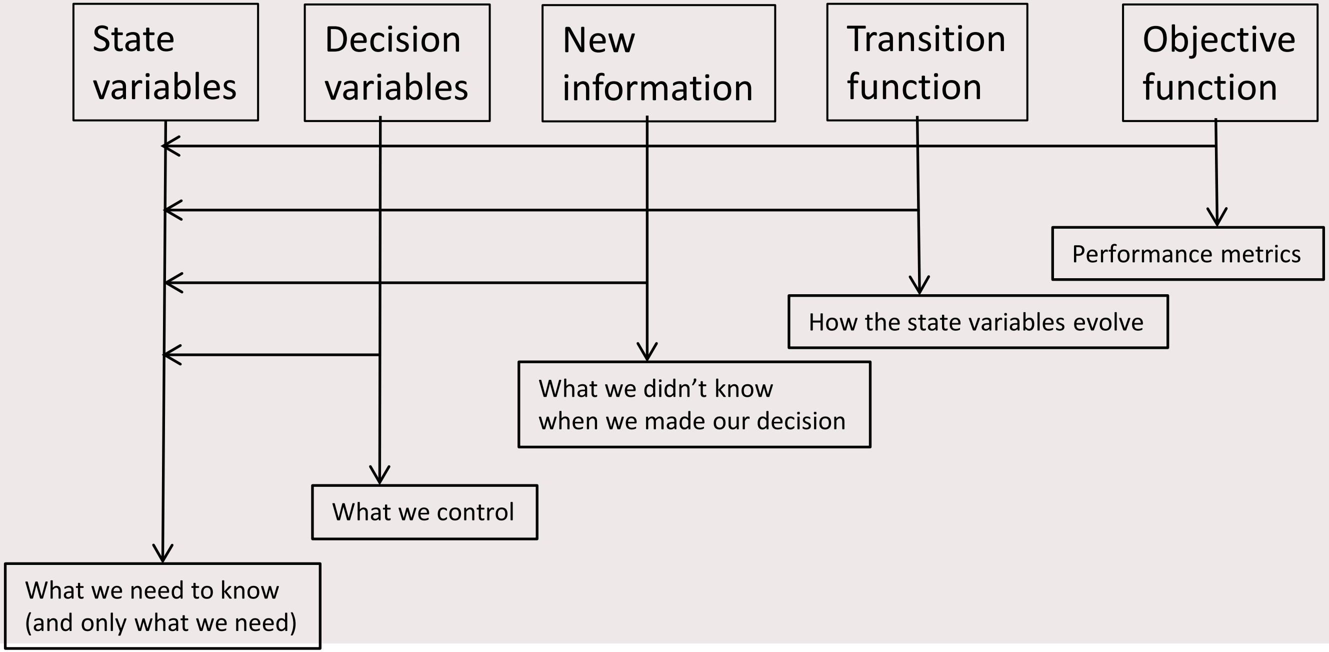
» Expectation or risk measures

» Mathematical properties (convexity, monotonicity, continuity, unimodularity, …)

» Time to compute (fractions of seconds to minutes, to hours, to days or months)

Elements of an SSDP

- Conduct a conversation with a **domain expert** to fill in the elements of a problem:



MAJOR OBSERVATION

WE DO NOT NEED DATA, WE ONLY NEED A GOOD MODEL OF REALITY

- In some applications, using data is actually easier than a model of reality

- In supply chain management, operations management and business, we mostly work with a model or simulation of reality (think e.g. about digital twins)

BREAK

It is very much possible this slide is either timed too early or too late.

In case it is timed too early: I hope that you still understand where this lecture is moving towards

In case it is too late: I hope you are still with me in the second part of the lecture.

Chapter 1.1:

Examples of the elements of SSDPs

The state variable

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 are determined for $t = t_0$ 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 $t > t_0$ such that the output [cost] *y(t)* for all $t \geq t_0$ 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* 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:

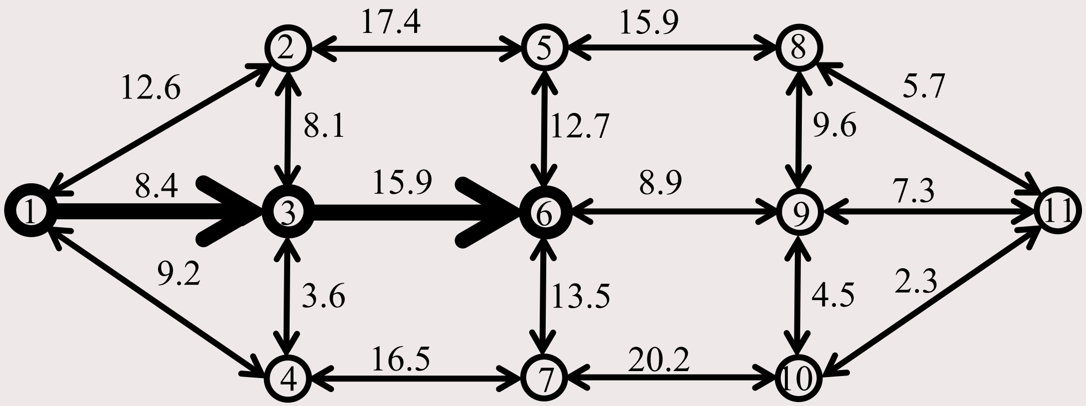
The mathematician's view – The state variable is a given, at which point the mathematician will characterize its properties ("Markovian," "history-dependent," …)

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

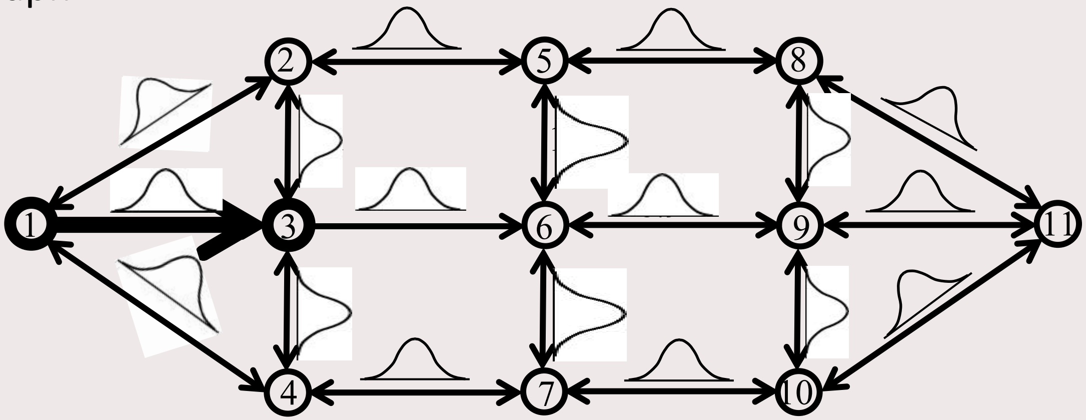


$$S_t = (N_t) = 6$$

The state variable

Illustrating state variables

A stochastic graph

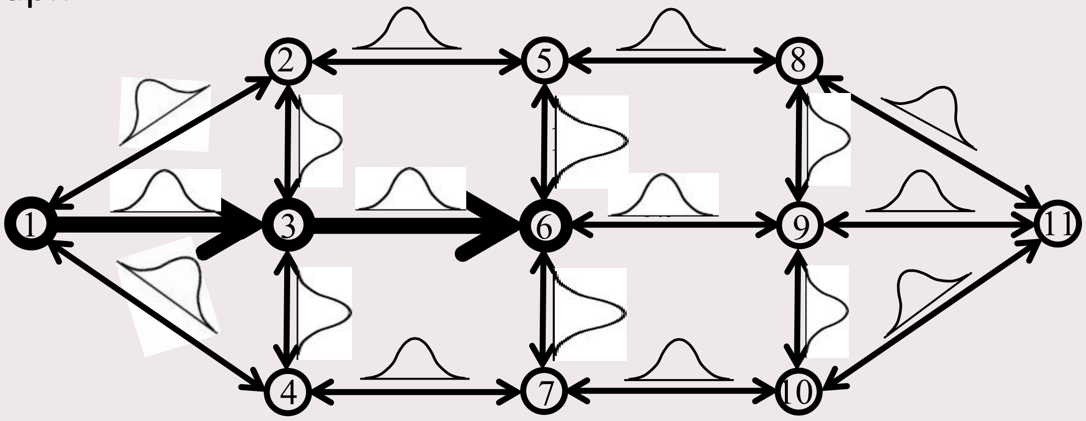


$$S_t = ?$$

The state variable

Illustrating state variables

A stochastic graph

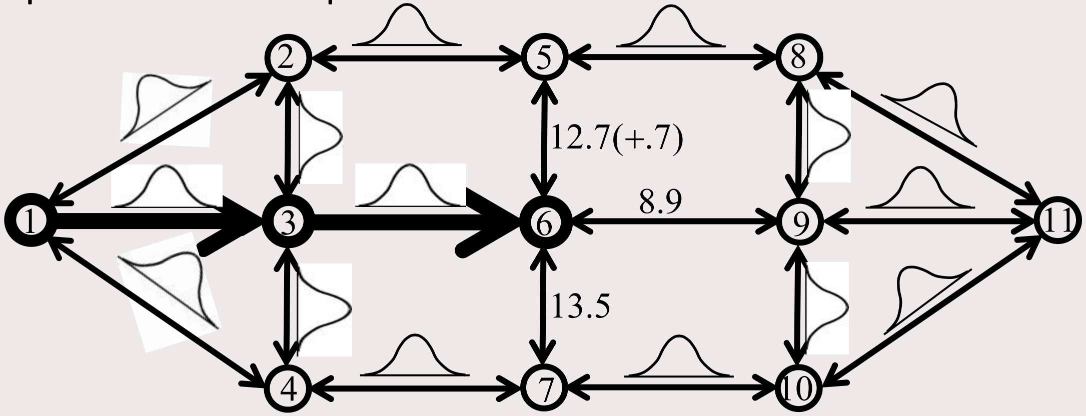


$$S_t = \left(\underbrace{N_t}_{R_t}, \underbrace{\left(c_{t,N_t,j} \right)_j}_{I_t} \right) = \left(6, \left(12.7, 8.9, 13.5 \right) \right)$$

The state variable

Illustrating state variables

A stochastic graph with left turn penalties

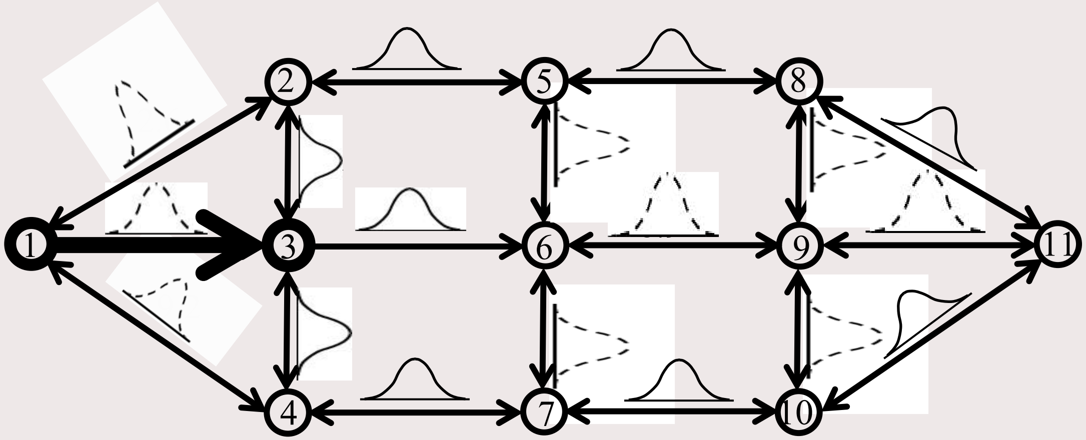


$$S_t = \Big(\underbrace{N_t}_{R_t}, \underbrace{\left(c_{t,N_t,j}\right)_j, N_{t-1}}_{I_t}\Big) = \big(6,(12,7,8.9,13.5),3\big)$$

The state variable

Illustrating state variables

A stochastic graph with generalized learning

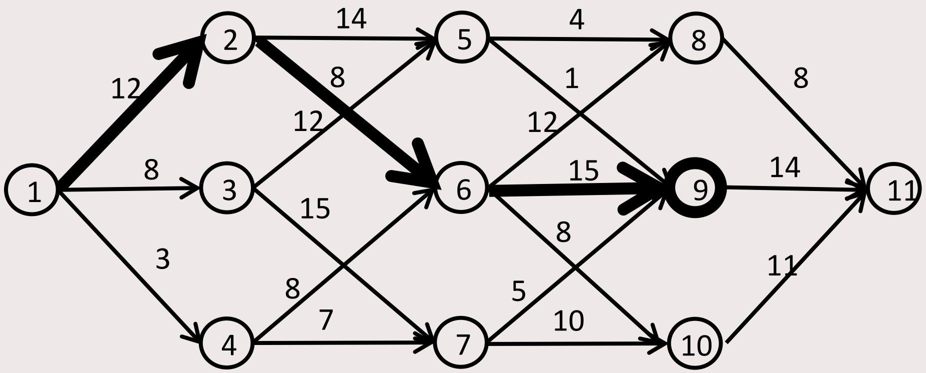


$$S_t = \Big(\underbrace{N_t}_{R_t}, \underbrace{\left(c_{t,N_t,j}\right)_j}_{I_t}, \underbrace{\qquad\qquad}_{B_t} \Big)$$

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:



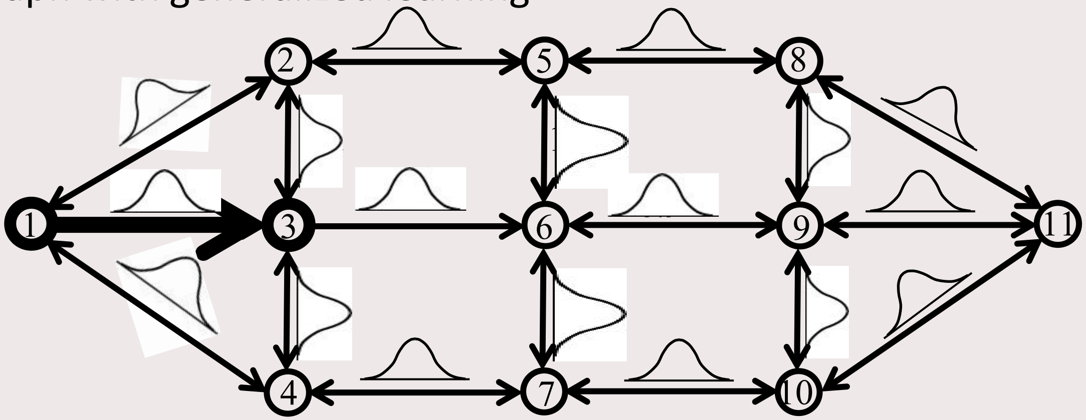
If the traveler is at node 9, what is her state?

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

The state variable

Illustrating state variables

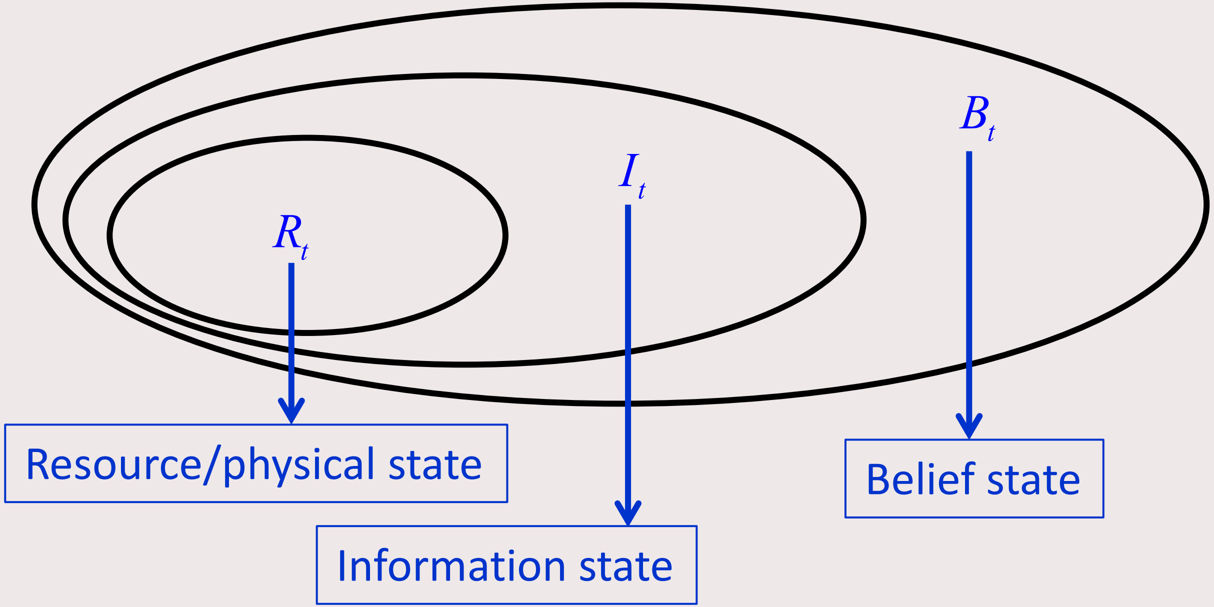
A stochastic graph with generalized learning



$$S_t = ?$$

The state variable

Classes of state variables - we will not make a formal distinction in this course



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!***

TU/e

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

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

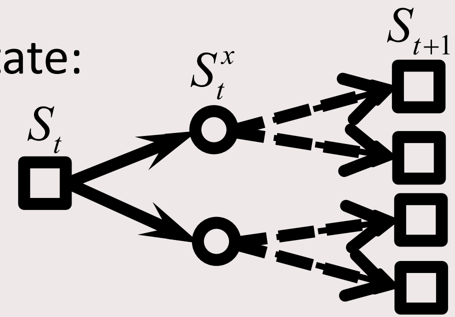
The state variable

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:

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

Modeling resources

The evolution of attributes:

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

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

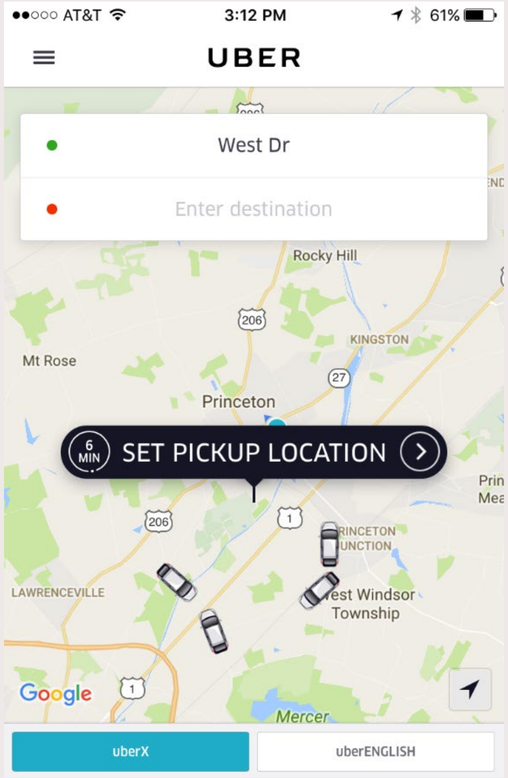
Fleet management

Uber

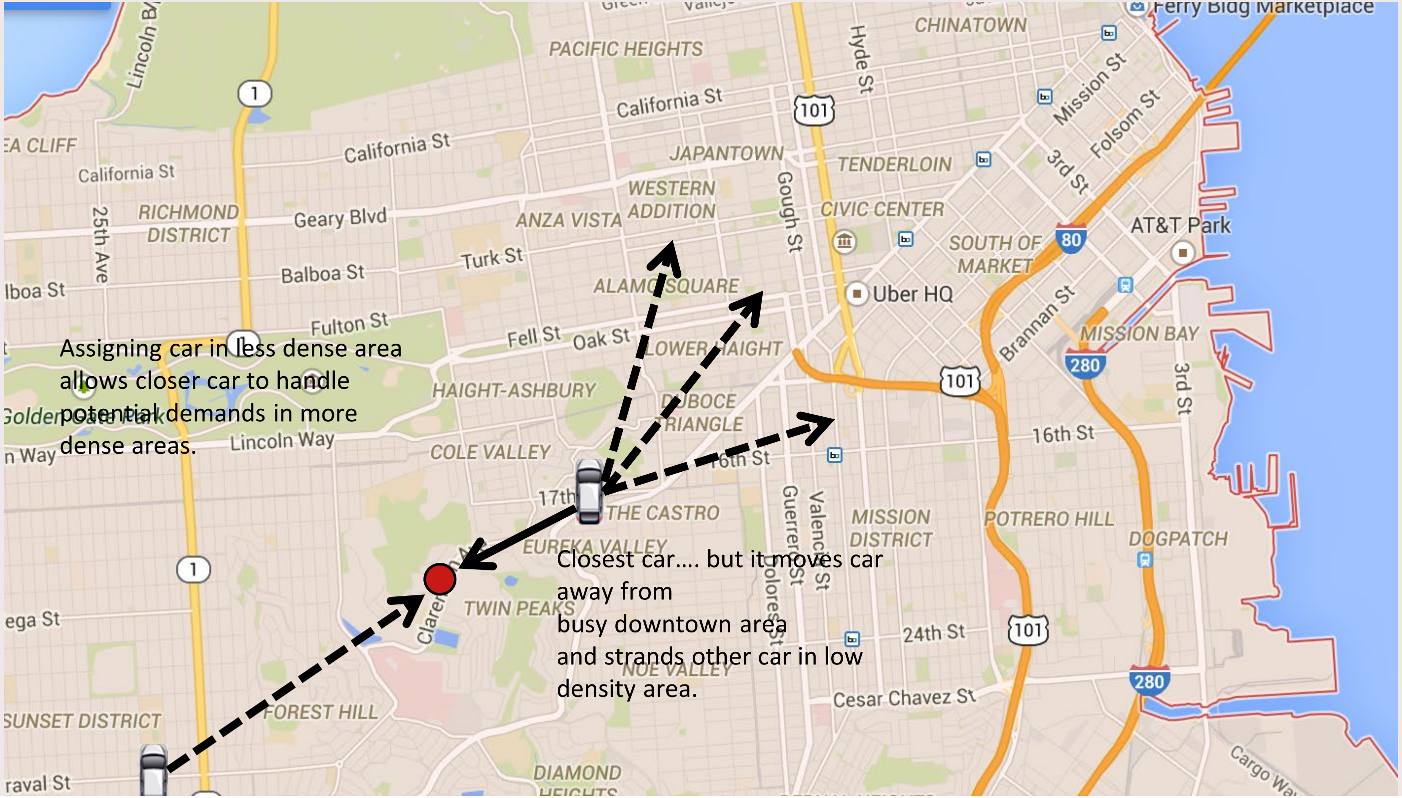
- Provides real-time, on-demand transportation.
- Drivers are encouraged to enter or leave the system using pricing signals and informational guidance.

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, …)



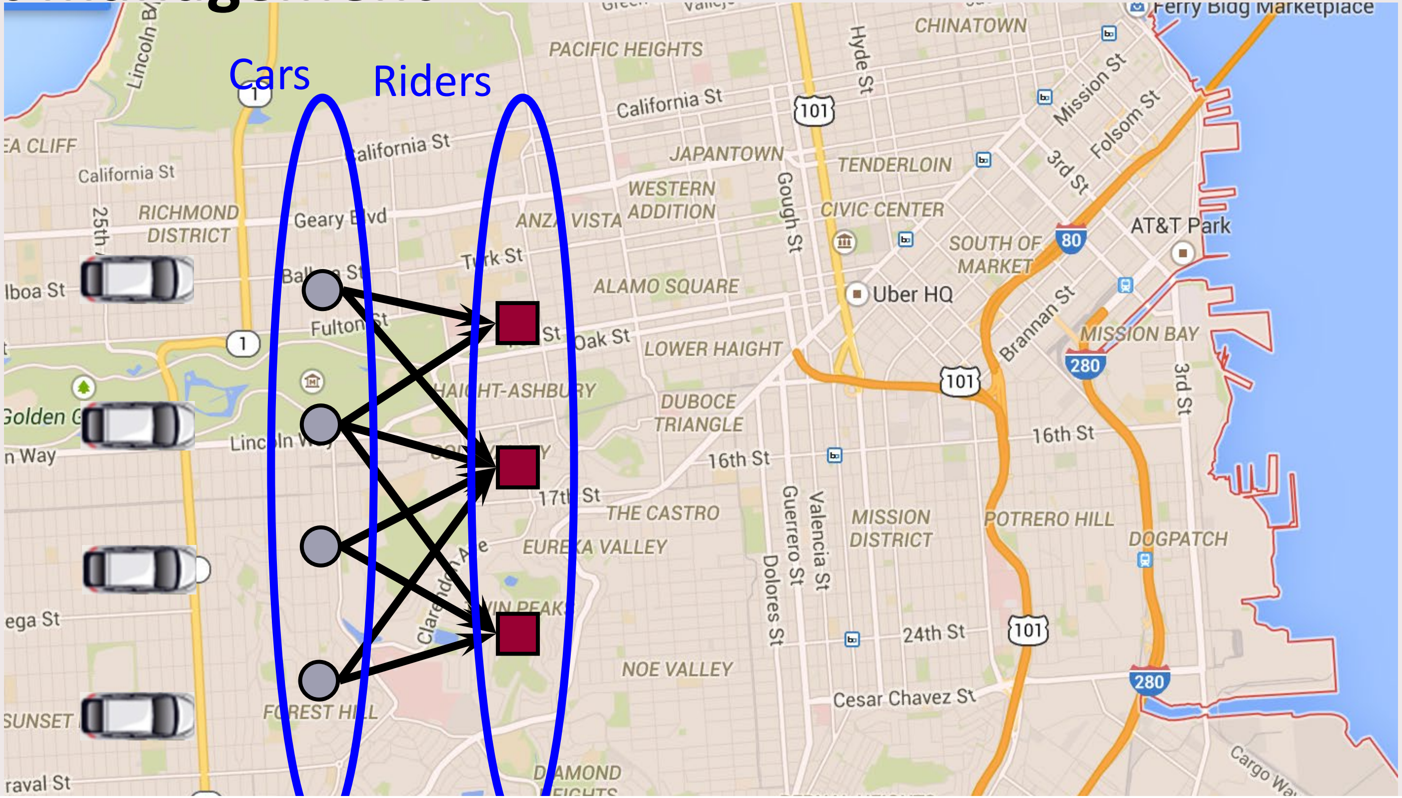
Fleet management



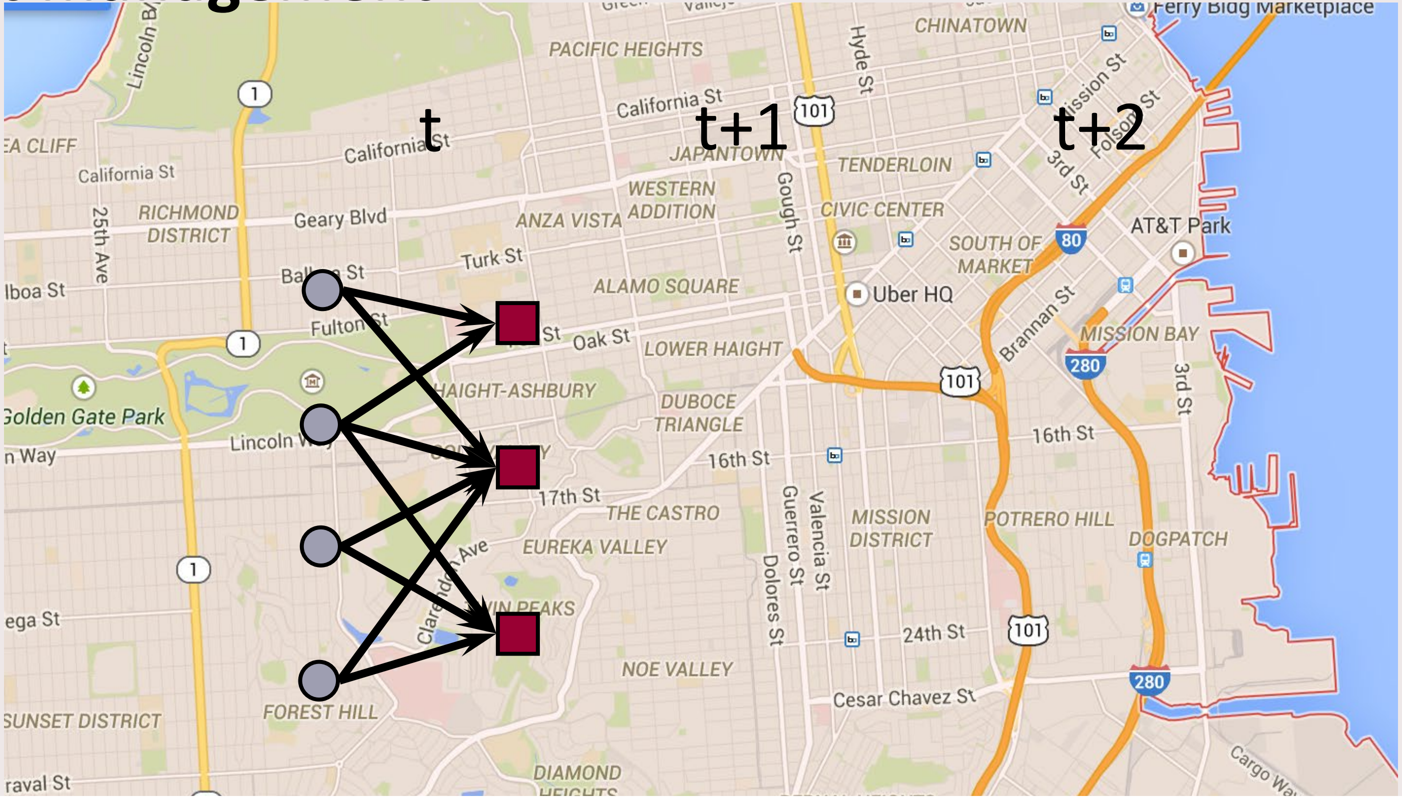
Assigning car in less dense area allows closer car to handle potential demands in more dense areas.

Closest car…. but it moves car away from busy downtown area and strands other car in low density area.

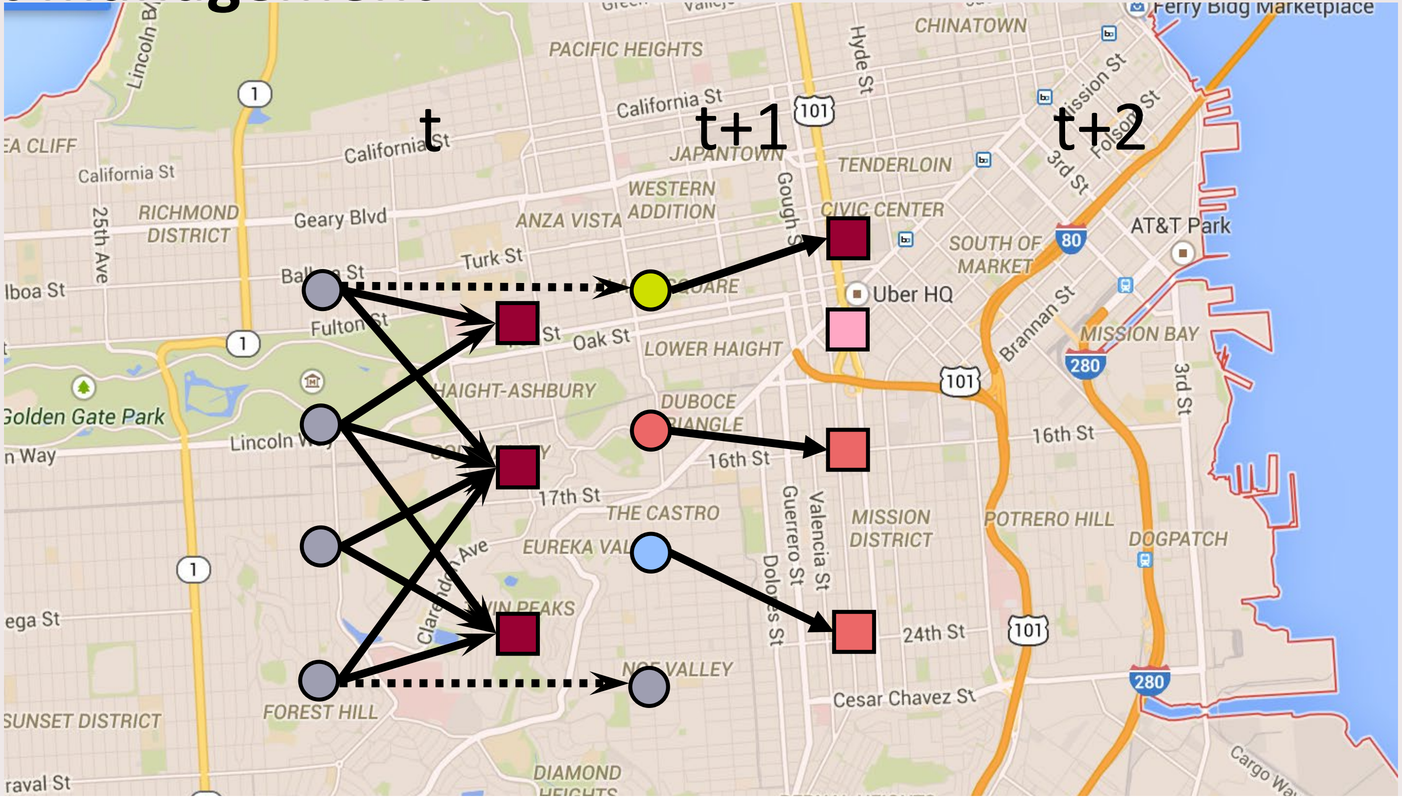
Fleet management



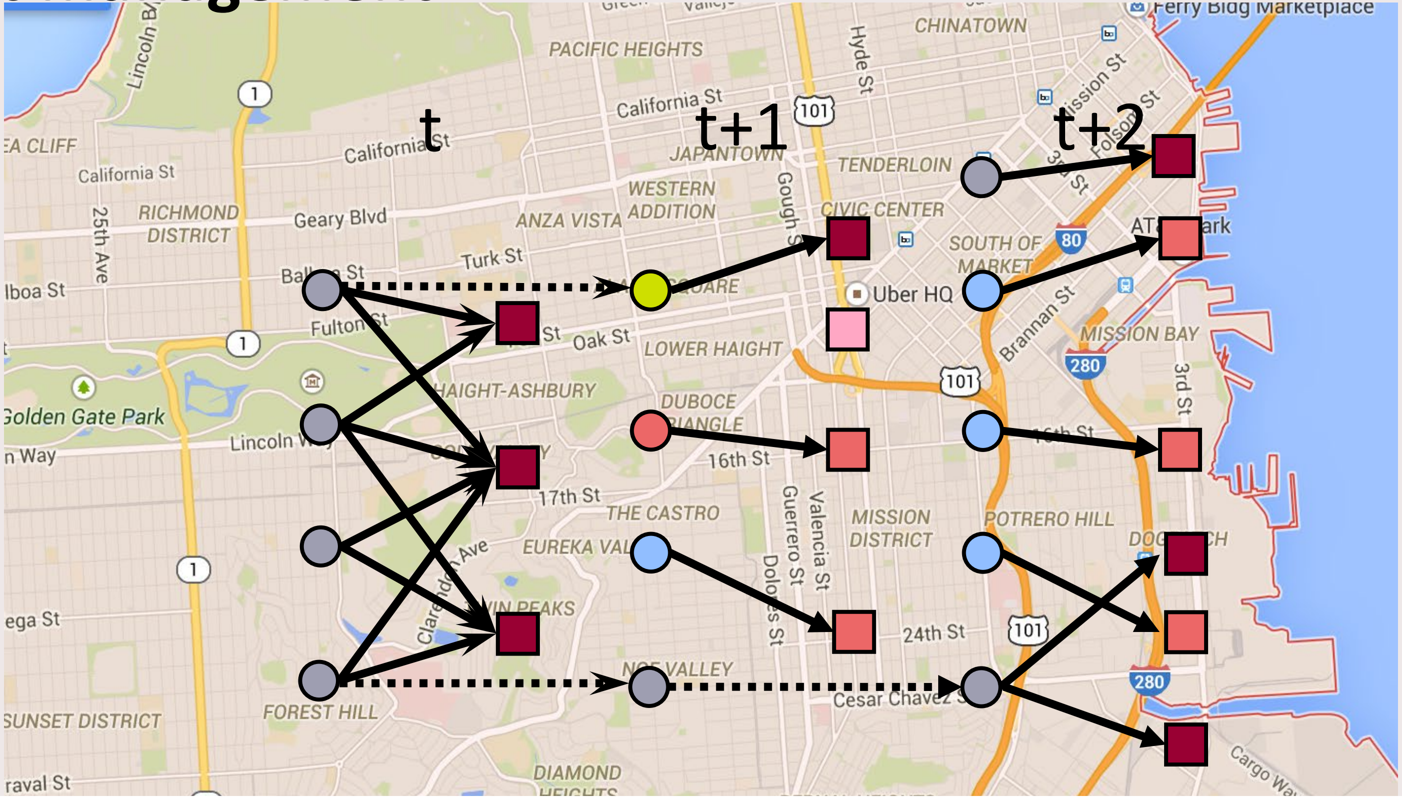
Fleet management



Fleet management



Fleet management



TU/e

Decision variables

When modeling decision variables:

Model the decision variables

This can be extremely complex

Model the constraints

Introduce the policy $X^\pi(S)$, and say that it will be designed later.

Decision variables

Styles of decisions

Binary $x \in X = \{0,1\}$

Finite $x \in X = \{1,2,\ldots,M\}$

Continuous scalar $x \in X = [a,b]$

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

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

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

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 x.
- $U^{\pi}(S)$ = The policy for determining action x.
- **$\boldsymbol{X^{\pi}(S)}$ = The policy for determining action $\boldsymbol{x}$.**

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.

To give an idea of the complexity I

- Dynamic vehicle routing problem:
  - Set of vehicles, set of customers to visit that appear dynamically with time windows

  - At each decision point – we need to solve a vehicle routing problem
  - Vehicle routing problems are combinatorial optimization problems that are NP-hard
  - Actually, such type of decision problems are (until somebody makes a real AI breakthrough) hard to solve with fancy Reinforcement Learning techniques
  - We focus on such a setting in the first half of this course.

To give an idea of the complexity II

- Bidding on the day-ahead electricity market:
  - At 12:00 each day, we need to determine for each hour the next day, how much energy we will produce.
  - This implies 24 time periods, each having continuous amount of variables
  - The number of possibilities is in the order of gazillions
  - Much better suited for Reinforcement Learning (less of a combinatorial/graph-based structure)

Thus: Selecting a decision is in operations management already a complex optimization problem by itself.

Exogenous information

Notation:

$\boldsymbol{W_t}$ 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. We can represent exogenous information as the most recent value of a random variable, such as the energy from wind:

$$\hat{E}_{t+1} = \text{energy from wind as of time } t+1.$$

… 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

- 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 ω be 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

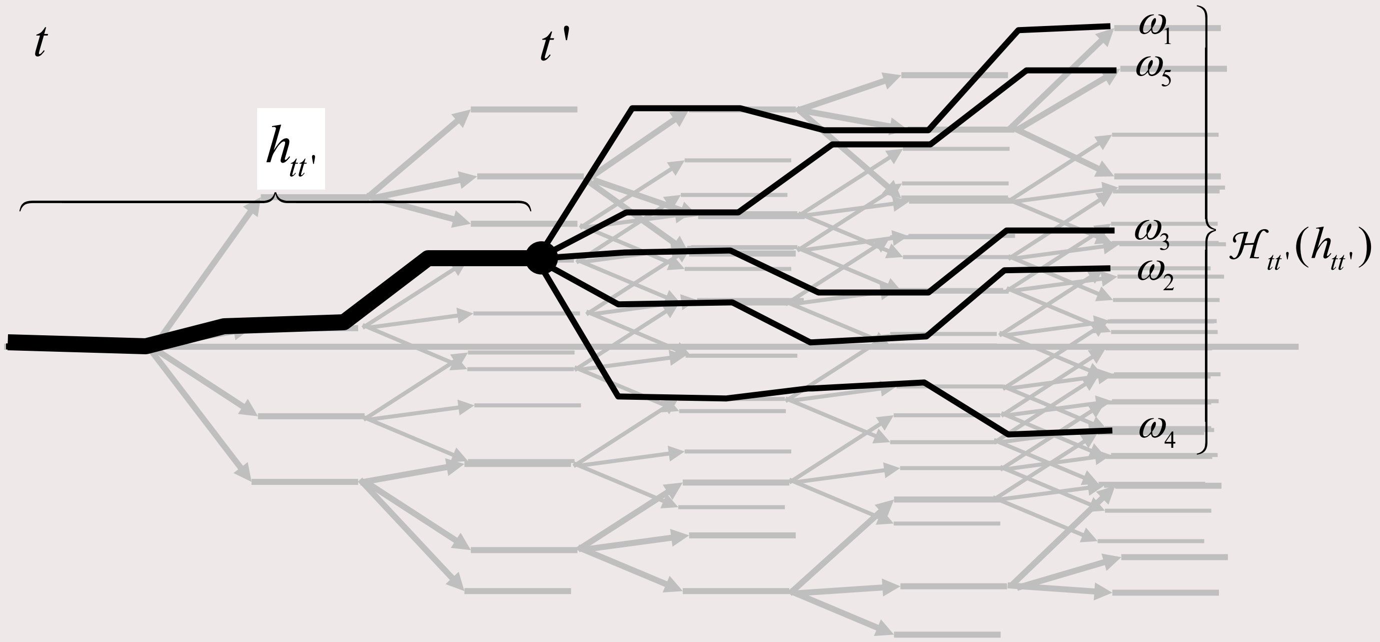
- Consider demand realizations over 9 periods.

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

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

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

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.

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 transition function

Physical resources:

An inventory problem

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

General resource allocation problems

- Let

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

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

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

Two frameworks:

Model-based

- This is where we have a set of equations that describe the transition.

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

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}.

In the end, the goal of any SSDP is:

$$\min_{\pi \in \Pi} E \sum_{t \in T} C\big(S_t, X^{\pi}(S_t)\big) | \, S_0$$

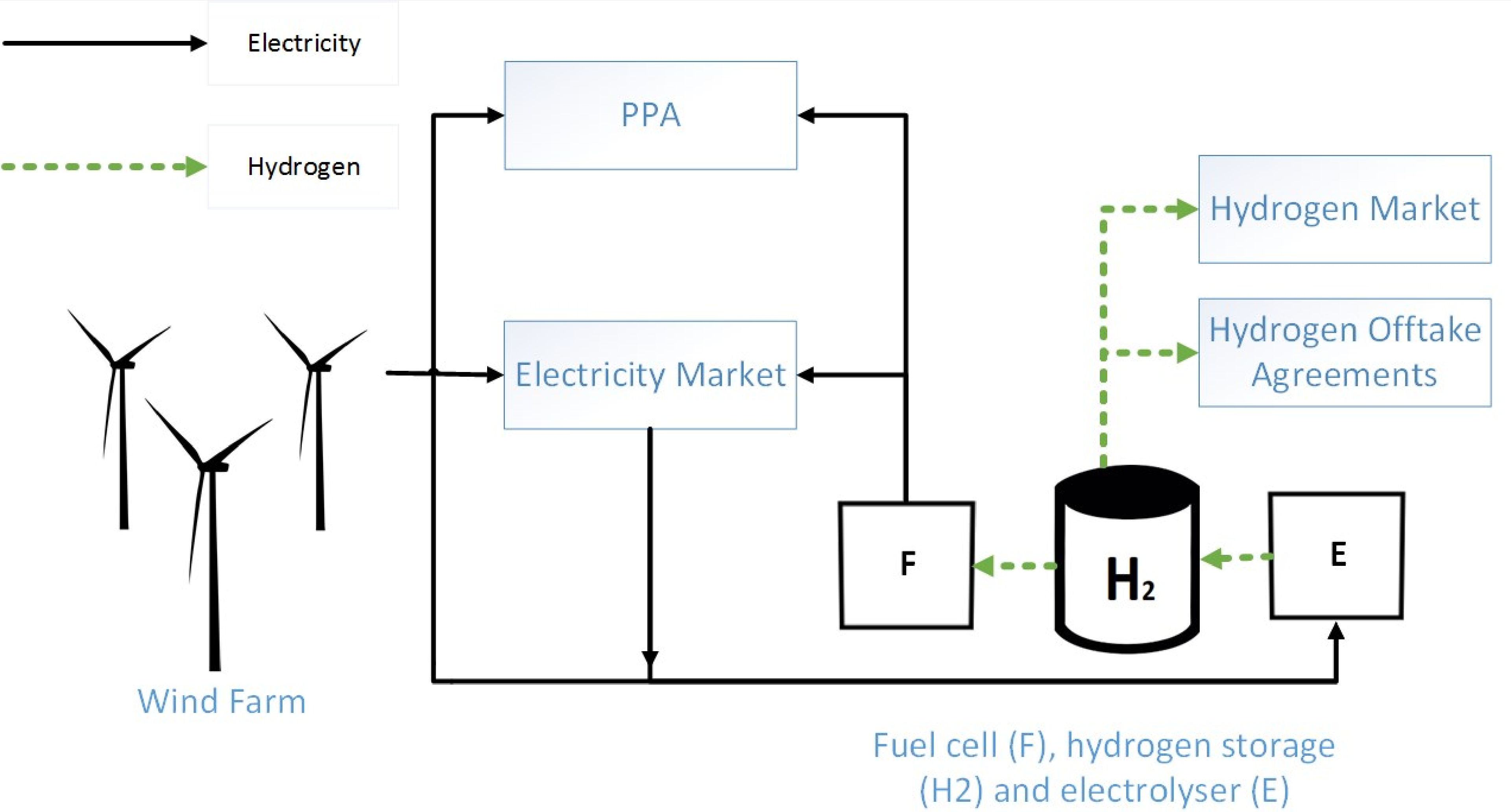
Where

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

And for some initial state S_0

A straightforward example in Hydrogen Plants

- Offshore wind farm that produces electricity
- Can store it in a hydrogen tank, with conversion los
- Can sell / buy elektricity from the market (to store in hydrogen e.g.)
- Uncertain price processes
- Maximize expected daily profit


Electricity
Hydrogen
PPA
Hydrogen Market
Hydrogen Offtake Agreements
Electricity Market
F
H_2
E
Wind Farm
Fuel cell (F), hydrogen storage (H2) and electrolyser (E)

1. Global system parameters

- Planning horizon $\mathcal{T} = \{1, \dots, T\}$ of one year.
- Power purchase agreement: Sell q^{ppa} electricity units for a fixed price p^{ppa} every interval of n^{ppa} periods. If the PPA target is not met, a penalty c^{penalty} is incurred for each unit of electricity that is short.
- Hydrogen is converted with efficiency α
- The storage tank has size Q (max in k^e, max out k^f)
- Maximum transmission capacity k^c
- Maximum hydrogen that can be sold k^h

2. State variable

- We define $St, t \in T$ as the state observed at period t
- $S\_t = (t, p_t^e, p_t^h, y_t, I_t, v_t)$.
- p_t^e denotes the electricity price
- p_t^h the hydrogen market price
- y_t the wind energy production
- I_t the inventory level (hydrogen in storage)
- v_t denotes the amount of electricity that still has to be sold according to the PPA.

3. Decision variable:

The decision variable $x_t(S_t) \in X(S_t)$ is described by a vector of four variables that indicate how much electricity is sold to the market, how much electricity is bought from the market, how much electricity is sold according to the PPA (in addition to the market interaction), and how much hydrogen is sold.

We write $x_t = (x_t^{sell}, x_t^{buy}, x_t^{PPA}, x_t^{h})$

But, we cannot take ANY decision

1. We can sell to or buy from the market but not both, hence $x_t^{\text{sell}} x_t^{\text{buy}} = 0$.
2. If $x_t^{\text{sell}} > 0$ and $x_t^{\text{buy}} = 0$, then the following constraints must hold:
   (a) The total amount of energy sold should satisfy the transmission capacity: $x_t^{\text{sell}} + x_t^{\text{PPA}} \leq k^{\text{c}}$,
   (b) As there is no electricity bought from the market, we convert all the production that is left after market interaction: $x_t^{\text{in}} = \max(0, y_t - x_t^{\text{sell}} - x_t^{\text{PPA}})\alpha$, where $\alpha = \alpha^{\text{e}}\alpha^{\text{f}}$,
   (c) For the same reason, the amount of energy leaving the storage equals the production in excess of the electricity sold to the market: $x_t^{\text{out}} = \max(0, x_t^{\text{sell}} + x_t^{\text{PPA}} - y_t)$.

3. If $x_t^{\text{sell}} = 0$ and $x_t^{\text{buy}} > 0$, then the following constraints hold:

   (a) The total amount of energy bought should fit in storage: $x_t^{\text{buy}} < \min\{(Q - I_t)/\alpha, k^{\text{c}}\}$,

   (b) We cannot simultaneously buy and sell, thus we convert the electricity bought from the market and the production not used for satisfying the PPA: $x_t^{\text{in}} = \alpha x_t^{\text{buy}} + \alpha \max\{0, Y_t - x_t^{\text{PPA}}\}$,

   (c) For the same reason, energy that leaves the storage is only used for meeting PPA obligations: $x_t^{\text{out}} = \max(0, x_t^{\text{PPA}} - Y_t)$.

4. Electrolyzer, fuel cell, and inventory capacity should be respected: $x_t^{\text{in}} \leq k^{\text{e}}$, $x_t^{\text{out}} \leq k^{\text{f}}$, and $0 \leq I_t + x_t^{\text{in}} - x_t^{\text{out}} \leq Q$

5. The maximum amount of hydrogen that can be sold equals: $x^{\text{h}} = \min\{I_t + x_t^{\text{in}} - x_t^{\text{out}}, k^{\text{h}}\}$.

Reward of taking an action

- The reward of taking an action consists of four parts

- The profit from selling electricity on the power market
- The costs of buying electricity on the power market
- The profit from selling to the contractual PPA
- The profit from selling hydrogen on the hydrogen market

4. Exogenous Information Function

After each decision point t, we observe new market electricity prices, new hydrogen prices, and wind-energy production. The exogenous information variable is denoted by $W_{t+1}(S_t)$ and does not depend on the action taken in period t. The prices are stochastic and follow an AR(1) process

- $p_{t+1}^e = \mu^e + \theta^e p_t^e + \epsilon, \epsilon \sim N(0, \sigma^e)$
- $p_{t+1}^h = \mu^h + \theta^h p_t^h + \epsilon, \epsilon \sim N\left(0, \sigma^h\right)$

- The production I_{t+1} follows a Weibull distribution with period dependent shape and scale parameters

5. Transition Function

The transition function $S^M(S_t, x_t, W_{t+1}) = S_{t+1}$ describes how we transition towards the state in period $t+1$. It first applies the feasible action x_t to reach a post-decision state S_t^*, after which the exogenous information determines the transition from S_t^* into S_{t+1}.

Let $S_t^* = (t^*, p_{t^*}^{\mathrm{e}}, p_{t^*}^{\mathrm{h}}, I_{t^*}, v_{t^*})$, where $t^* = t$, $p_{t^*}^{\mathrm{e}} = p_t^{\mathrm{e}}$, and $p_{t^*}^{\mathrm{h}} = p_t^{\mathrm{h}}$.

1. The post-decision inventory level $I_{t^*} = I_t + x_t^{\mathrm{in}} - x_t^{\mathrm{out}} - x_t^{\mathrm{h}}$.

2. The post-decision target $v_{t^*} = v_t - x_t^{\mathrm{ppa}}$ in case $t \mod n^{\mathrm{PPA}} \neq 0$, else $v_t^* = q^{\mathrm{PPA}}$.

Then, the transition to S_{t+1} follows trivially by including the new information from W_{t+1}. That is, $S_{t+1} = (t+1, p_{t+1}^{\mathrm{e}}, p_{t+1}^{\mathrm{h}}, I_{t+1}, v_{t+1})$ with $I_{t+1} = I_{t^*}$ and $v_{t+1} = v_{t^*}$.

6. Objective Function

The objective of a profit-maximizing GHP operator is then given by a decision policy $\pi \in \Pi$ and a decision rule $X^\pi : \mathcal{S}_t \to \mathcal{X}(\mathcal{S}_t)$. Thus, we write $x_t = X^\pi(S_t)$ as the decision x_t under decision policy π. The objective of the GHP operator is then to maximize its total expected profit:

$$\max_{\pi \in \Pi} \mathbb{E}\Big[\sum_{t \in \mathcal{T}} R(S_t, X^\pi(S_t)) \mid S_0\Big], \tag{1}$$

where $S_{t+1} = S^M(S_t, X^\pi(S_t), W_{t+1})$, and S_0 denotes the state at the start of the year.

Chapter 2:

Four classes of policies for SSDPs

To attack the three curses of dimensionality
- States, Decisions, Transitions

The four fundamental policies

- Policy Function Approximations (PFAs)
- Cost Function Approximations (CFAs)

- Value Function Approximations (VFAs)
- Direct lookahead approximations (DLAs)

- Recal that at time t, we observe state S_t, face action space $X(S_t)$, and need a decision rule π or policy to take decision $x_t \in X(S_t)$

Policy Function Approximations (PFAs)

These are analytical functions that map a state (which includes all the information available to us) to a decision.

- Lookup tables
- Parametric functions (OLS, regression)
- Nonparametric functions (deep neural nets)

PFAs in the Hydrogen Plant

- In reality, electricity needs to be bid on the day-ahead market
- Neural net that uses information of today (weather, prices, weather forecasts) for tomorrows prices so that we can strategically bid
- Lookup Table: <If price is above a threshold sell, otherwise buy>

Cost Function Approximations (CFAs)

We rely on solving a parameterized optimization problems for each state. This optimization problem gives us a decision. The parameters steer which decision is taken

Example: Google Maps – You enter a destination, it says 1h20m. You leave 1h30m because you buffer against uncertainty. This is a very simple cost function approximation, you just add a certain amount of time to the problem.

CFAs in the Hydrogen Plant

- We need to fulfill PPA obligations – scale the time until deadline with some parameter

- Past variability in prices, parameterize and see how we can act upon that (direct sell, later sell/buy)

Value Function Approximations (VFAs)

- We define the value of being in a state as $V_t(S_t)$, and we want to solve:
$$V_t(S_t) = \max_{x_t \in X(S_t)} \left(C(S_t, x_t) + E_{W_{t+1}}\{V_{t+1}(S_{t+1})|S_t\} \right)$$
- We can do this to optimality by dynamic programming and Markov decision process theory.
  - We will discuss the details of this later in this course.

- A value function approximation provides an approximation of $V_{t+1}(S_{t+1})$. In its most simplest form (basically a simulation) it is called Q-learning. All more efficient forms are typically some form of Deep Reinforcement Learning or use techniques from Approximate Dynamic Programming.

VFA for the Hydrogen market

- Only look two periods ahead;
- Simulate future events to update value functions
- Stochastic Dual Dynamic Integer Programming (google it, it is AMAZING)
- Solve to optimality

Solving an SSDP to optimality

- (Stochastic) Dynamic programming
- Based on the Value function:
  - $V_t(S_t) = \max_{x_t \in X(S_t)} \left(C(S_t, x_t) + E_{W_{t+1}}\{V_{t+1}(S_{t+1}) | S_t\} \right)$
- This can be done by conditioning on time step t, backwards in time.

Focus of this course is on:

State-dependent problems without learning - These are problems that depend on a dynamically changing state. These include any of a vast range of dynamic resource allocation problems (managing blood, energy, money, people, equipment, logistics) in the presence of different sources of uncertainty (demands, prices, costs, times, ...), but which do not require learning any exogenous parameters.

We challenge you to make the step towards learning !

Further course information

Assignment 1

- Find a practical and useful example of an SSDP in the field of logistics/operations management on the internet, news, newspapers, radio, etc etc.
- Define the problem properly in terms of the unified framework for stochastic optimization (State variable, Decision variable, Exogenous Information Function, Transition Function, Objective Function)
- This is probably a formulation that faces the curses of dimensionality, so indicate what the most crucial elements are to anticipate future decisions.
- In other words, why is it crucial that your example is being solved better then some simple myopic policy

Assignment 1

- Deliverable is a **Pitch presentation of 5 minutes by your team.**
- This will be a fun exercise, because you will not have 1 second more than 5 minutes.

- Goal of the pitch presentation is to convince the audience that your problem should be solved using AI and or ADP and or RL and or ML.

- Presentations take place in week 2 in the first hour of instructions.

- Grading: ***15% of case-study grade*** and is based
  - 1) quality of presentation, 2) novelty of your problem, 3) relevance of SSDP/AI for your problem, 4) correctness of the description of your SSDP, 5) link to practice and interpretation of business problem

Assignment 1: Rules of the game

- No double topics: Please share your topic at Canvas Discussion Board

- A max 10 slide presentation (see assignment on canvas) to be handed in via Canvas – this is not necessarily the pitch presentation

- Grading will be done based on both the handed in slides and the presentation.

- NOTE: This is ``only'' a small part of your final assessment; see it as a fun exercise that you should tackle in an afternoon of work.

Case-study : Bike-sharing systems and rebalancing operations

We are given a set of data (1 mln transactions) from the Washington DC bike-sharing system in April 2014. This entails a number of bike-sharing stations at which customers take a bike. They can bring their bikes back at any other bike-sharing station. The problem that arises is that at some point the number of bikes at each station is becoming very imbalanced. There are trucks driving around that rebalance this system in real-time. The case entails around three questions.

1. What are suitable bike-sharing capacities for each station ?
2. Find good real-time demand prediction for new customers
3. Optimize, in real-time, the operations of the trucks that rebalance the network.

Goal is simple: You need to maximize the expected number of customers being served.

Case-study : Bike-sharing systems and rebalancing operations

Basic requirements are to:

0) Being able to structurally assign capacities to the system and test its impact with the next points.
1) Being able to formally define the associated SSDP of the operational/real-time problem.
2) Being able to make prediction of future demand (in real-time) that help to make better decisions
3) Being able to make better decisions based on a) the better predictions of future demand and b) better decision rules or policies in isolation of the demand prediction
4) To achieve the latter, a basic scenario sampling method using consensus functions (Lecture week 2) have to be implemented. In this method, how scenarios are sampled can be done by the fanciest deep neural nets or by simply relying on empirical data.
5) There are **MANY** potential refinements possible to the case-study, we **URGE** you to consider such refinements and discuss these with the lecturers.

Case-study : Bike-sharing systems and rebalancing operations

Rules of the game:

- Much can be found on the internet, there are gazillion good sources
- We want you to **learn**, so if we suspect you have no clue what is going on in the code/algorithms you delivered, we will question you. In case we deem your team does not understand the details of what you use you automatically fail this course.

- In the end, new data is provided on which the performance need to be reported.

Case-study : Bike-sharing systems and rebalancing operations

Comparison among teams

- At the start of week 2, a gym and trainer environment will be provided.
- **The performance of the trainer on this fairly simple environment MUST be reported, and serves as a comparison among the teams.**
- Extensions to the gym/trainer are all possible! As long as you use the old dataset for training and the new dataset for testing/evaluation, and adhere to the same objective function and global system semantics
- Extensions to the global system semantics are allowed too (e.g., incorporating customer behavior choosing more than a single station), but this should be done in addition to the base-line model.