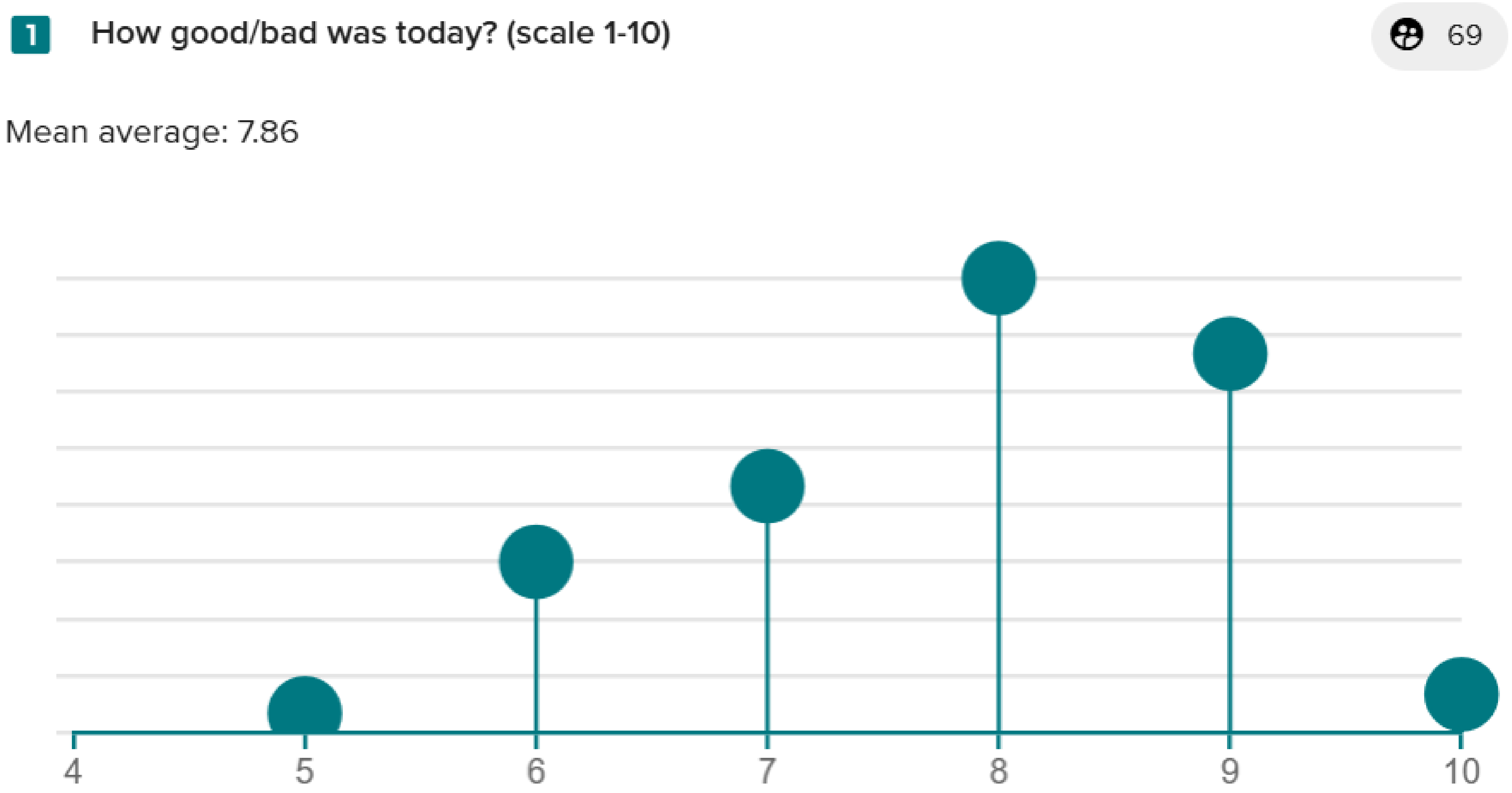
DTU

Decision Making under Uncertainty (02435)

Section for Dynamical Systems, DTU Compute.

DTU Compute
Department of Applied Mathematics and Computer Science

Feedback & Follow-up



What you liked

- Cool quiz
- Finally got somewhat of an idea of what a Markov Decision Process is
- The fact that the class was interactive , to learn doing games and slide through slide it was really nice
- Examples given
- Like that we have possibility to be active
- The overall structure with slides and excercises
- The quizzes helping me realize that I didn't know enough 😅.
- Well structured, clear lecture.
- Very great examples. Very great brick by brick explanations.
- The thorough explanation of the concepts. A clear example containing all concepts.
- Consistency and participation activities
- The quiz! Also the lecture was about 1.5 hour and I like short lectures
- The game
- Relatively quick.
- The game at the end was fun The example (of last year's assignment) was informative
- Overall it was nice to follow and easily explained
- my expectation about the class somewhat fulfilled.
- How the content was explained. Super dynamic and structured, good flow.
- Not too much information but thorough explanation
- The interactive approach
- Approaching the theory with a practical example
- The catching up on theory
- Going through the "old" exam and translating it for us.. Together
- I liked the quizzes during the lecture - it was Easier to understand the difference between...
- Answer a lot of questions about Markov makes it easier to understand
- MDP were new to me, I learnt a new approach to problems
- Active learning
- The approach to explain with a lead example and the quiz in the end
- Everything
- The walkthroughs, the highlighted text pieces
- The quiz was nice
- Seeing an example.
- Nice with a walkthrough of last years exercise and a fun quiz to end the lecture with.
- The connection with the prev assignment
- Explanation about the topic was clear.
- All
- The quiz
- Detailed and good presentation
- The game
- The quiz
- decision approach
- The quiz
- Examples
- Q&A
- Made me think with the interactive quiz
- The example was good, helps explain a lot of the concept
- It's still a fun and interactive class
- Interactive class
- Your explanitions were good
- The fast pace, it was clear but concise.
- pretty much
- Was good.
- Active learning method is the best
- We are engaged in finding the solution
- The game
- I like that it is short and concise
- Active class in which we could participate.
- The quiz
- The explanations were clear, good to see examples of code given as well

What you disliked

Nothing much

Can you present the slides as pdf instead of PowerPoint format

Maybe too fast sometimes , the code also was not on the slide so it was difficult to understand

The programming part could have been a bit more detailed

Very little actual information. I think the interactivity is refreshing but it takes up so much time.

Nothing

Maybe a bit more of tying the subject to the subject of last week. To get a more continuous feel

Vevox doesn't work :(

I couldnt see the code properly, which i consider an important part, so I didnt get some of the commands...

None

I like the quiz format in the end but not that you may take it into account when grading. Top 3 is only selected on...

The speed of explanations, its hard to keep up from time to time. -> talking too fast

The pace. Perhaps a diagram that shows how the elements are connected

It could be nice if the slides had the symbols on them, and not just blank space:)

Some of the coding seems hard and could be explained in more detail. Fx showing some of the functions.

Not much material based on mdp. Maybe find a better case study to explain it for example battery charge

Stress more and spend more time on the most important parts

Confusing quizz

Not being able to prepare for class

Too easy, too much use og technology.

The terms might have been covered a little too fast.

That the slides we had differed from the one at on the projector

not very informative slides. unclear which are the deadlines for tasks/assignments

Try using a kahoot for the quizz

The bugs on vevox

To many questions in the last quiz

That the example was in Julia and we have to code it in pyomo

The quick walkthrough of the environment simulation code

.

I didn’t win the quiz

Maybe you were going to fast

The quiz results

I bit heavy on some areas with only one break. Try dividing into more shorter breaks

hard to follow the code on the slides, maybe upload it next time

A practical numerical simulation of MDP would have been helpful, on top of the qualitative explanation, to better...

Maybe a heads up for a quiz that could enhance your grade beforehand would be nice

Nothing

Having a focus on grading during the lectures (the quiz) feels weird. Takes focus away

Very loose definitions - would not understand eg Markov property if I hadn’t seen it before

nothing

Nothing :) Nice with a shorter lecture and a quiz, I feel like you learn a lot from the quiz.

Could you pls upload the slides a little bit earlier as it a bit of hard for me to catch up directly

Not linking slides with presented code

Nothing

the terms are too complicated to set apart easy. needs more training, but it is not as straight forward as...

NA

Nothing very well formed the whole lecture

Too quick on the code implementation

That I didn't won

I have had a machine learning course before and therefore have done mdp. But today I just got more confused...

n/a

Maybe there could be a bit more math

Nothing

No reading material

Computer bugged out Not ur fault tho

None

All fine

nothing

Nothing :) Nice with a shorter lecture and a quiz, I feel like you learn a lot from the quiz.

Could you pls upload the slides a little bit earlier as it a bit of hard for me to catch up directly

Some comments

Some people like the quiz, some don’t

Math rendering on the slides fixed

Deadline for Assignment A is April 7th

Some people find the material too hard, others too easy

Also check the Q&A section after each lecture.
Some answers & info can be found there.

- Top 3 in the quiz gets you 1,5 <u>bonus</u> points
- Textbook(s): https://algorithmsbook.com/
- Another example on MDP: https://www.youtube.com/watch?v=H_9YQBN45fo

Plan

→ ~~Task 0~~

→ ~~Task 1~~
~~Building an evaluation framework for sequential decision-making methods~~

→ Today: Task 2
Stochastic Programming policy (2-stage)
+ Expected Value policy a.k.a. MPC

→ Next Week: Task 2

Multi-stage Stochastic Programming + caveats

→ Week 5: Assignment Work for Task 2 and Q&A

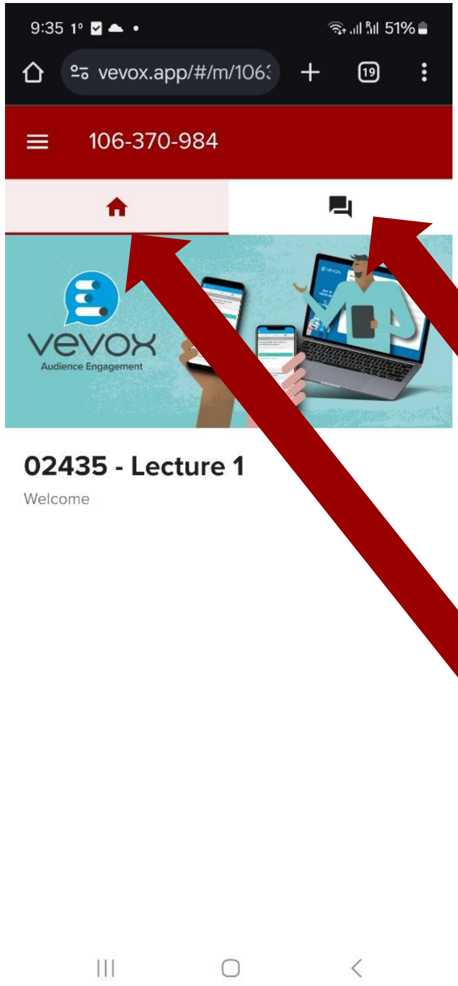
→ Weeks 6-7: Task 3
Approximate Dynamic Programming

→ Week 8: Assignment Work for Task 3 and Q&A

→ Weeks 9-11: Assignment B
Robust Optimization

Task 4 is about reporting the results from Tasks 2 and 3

Scan me:

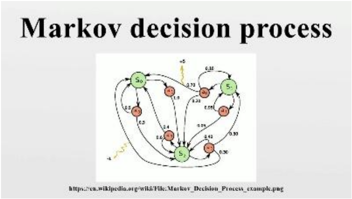


Anonymous Survey (at the end)

Anonymous Questions
(during or after the lecture)

Quizzes

The process of designing "Decision-making" frameworks



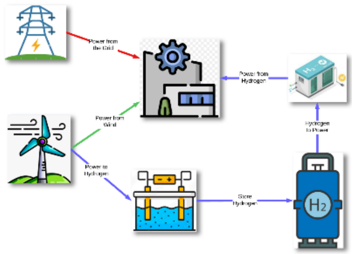
Last Week

Define → Design → Validate

This Week

Last Week

Coding a simulation Environment to evaluate *any* decision-making policy



Agenda for today

1. MDP recap and questions
2. Expected Value Policy
3. Two-stage Stochastic Programming policy

Markov Decision Process

- Stages $t \in \mathcal{T}$
- State $\boldsymbol{x}_t = \{x_{1,t}, x_{2,t}, \dots\}$
- Decision $\boldsymbol{u}_t = \{u_{1,t}, u_{2,t}, \dots\}$
- Dynamics $\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t)$
- Cost $c_t = g(\boldsymbol{x}_t, \boldsymbol{u}_t)$

Markov Decision Process

- Stages $t \in \mathcal{T}$
- State $\boldsymbol{x}_t = \{x_{1,t}, x_{2,t}, \dots\}$
- Decision $\boldsymbol{u}_t = \{u_{1,t}, u_{2,t}, \dots\}$
- Dynamics $\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t)$
- Cost $c_t = g(\boldsymbol{x}_t, \boldsymbol{u}_t)$

The *State* variables $\boldsymbol{x}_t$ enclose the necessary and sufficient information to model the system's behavior from stage t onwards.

Markov Decision Process

- Stages $t \in \mathcal{T}$
- State $\boldsymbol{x}_t = \{x_{1,t}, x_{2,t}, \dots\}$
- Decision $\boldsymbol{u}_t = \{u_{1,t}, u_{2,t}, \dots\}$
- Dynamics $\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t)$
- Cost $c_t = g(\boldsymbol{x}_t, \boldsymbol{u}_t)$

The *State* variables $\boldsymbol{x}_t$ enclose the necessary and sufficient information to model the system's behavior from stage t onwards.

Solution Concept:

Policy π: $\boldsymbol{u}_t = \pi(\boldsymbol{x}_t)$

Optimal Policy:

we cannot solve it when the algortithm is close to optimal we need to aproxximate

$$\min_{\pi} \left\{ \sum_t E_{\boldsymbol{x}_t \sim \pi}[c_t] \right\}$$

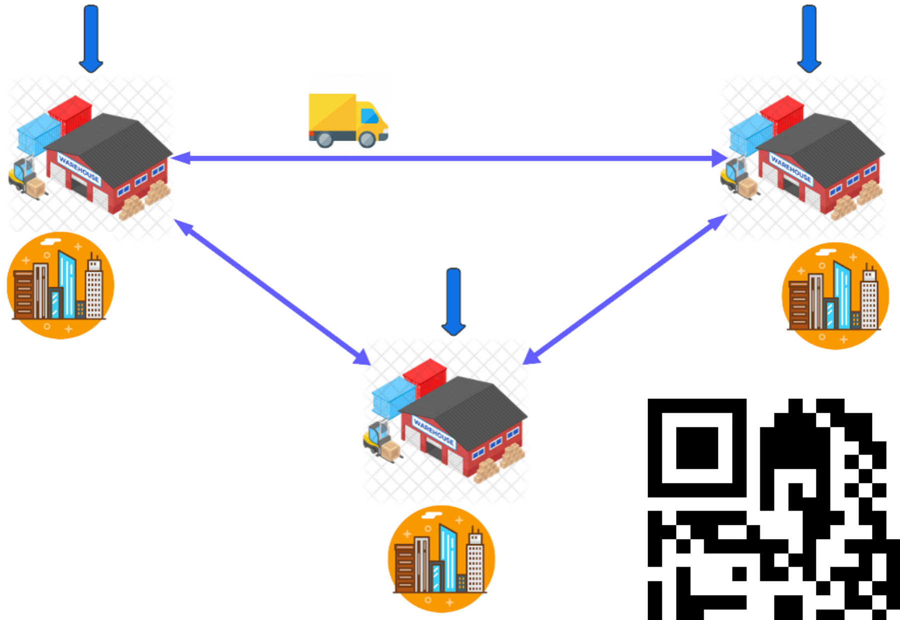
$$\boldsymbol{u}_t = \pi(\boldsymbol{x}_t), \qquad \forall t$$

$$c_t = g(\boldsymbol{x}_t, \boldsymbol{u}_t), \qquad \forall t$$

$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t), \ \forall t$$

For problems with discrete (and small) state/action spaces, we can design optimal policies:
Value Iteration, Policy Iteration (not in this course)

The Lead Example



- Stages $t \in \mathcal{T}$
- State $\boldsymbol{x}_t = \{x_{1,t}, x_{2,t}, \dots\}$
- Decision $\boldsymbol{u}_t = \{u_{1,t}, u_{2,t}, \dots\}$
- Dynamics $\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t)$
- Cost $c_t = g(\boldsymbol{x}_t, \boldsymbol{u}_t)$

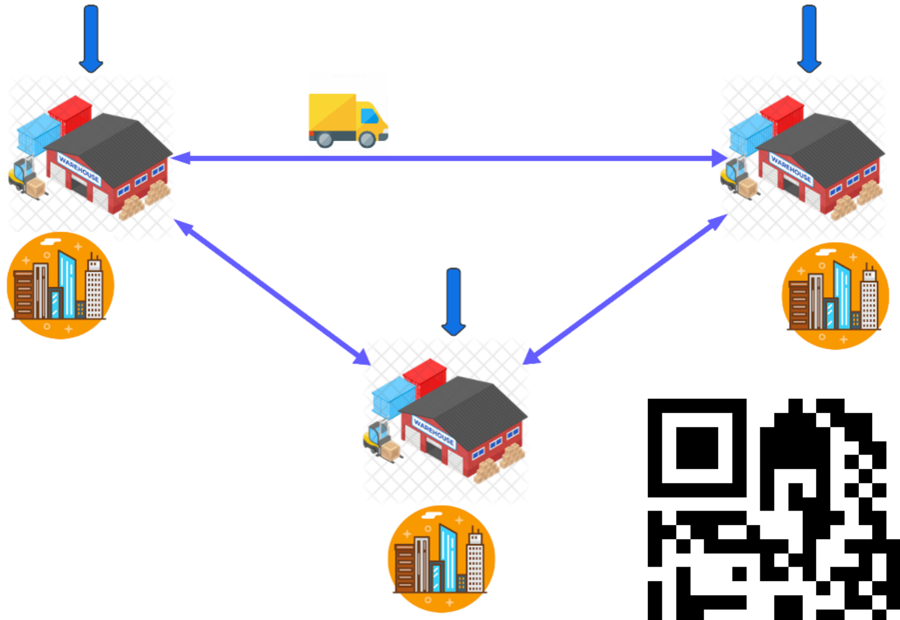
Consider a city divided into three districts.
Each district features a dedicated warehouse $w \in W = \{1,2,3\}$ which serves the district's demand $\mathrm{D}_{w,t}$ for coffee.
The coffee demand for each warehouse and day is known.

Each warehouse can store coffee up to a capacity limit C^{store}.
Denote the storage level of w at t by $z_{w,t}$.
At stage t, each warehouse w can order an amount $o_{w,t}$ of coffee from external suppliers at price $p_{w,t}$.
The price is different for each warehouse and each day.

Neighboring warehouses can also exchange coffee between them.
Let $y^{rcv}_{w,q,t}$ denote the amount received by w from a neighboring warehouse q, at t.
Similarly, $y^{send}_{w,q,t}$ is the amount sent by w to q.
To send an amount $y^{send}_{w,q,t}$, warehouse w must already have this amount previously stored.
The amount sent in one stage is restricted by a transportation limit C^{trnsp}.
Each exchange comes at a per-unit transportation cost $\mathrm{e}_{w,q}$.
Failing to meet a district's demand at any day comes at a per-unit cost of b_w.

Your job is to build a program that makes the day-by-day decisions for the coffee distribution system of the three warehouses, so that the city's coffee needs are met at the minimum expected cost.

The Lead Example



- Stages $t \in \mathcal{T}$
- State $\boldsymbol{x}_t = \{x_{1,t}, x_{2,t}, \dots\}$
- Decision $\boldsymbol{u}_t = \{u_{1,t}, u_{2,t}, \dots\}$
- Dynamics $\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t)$
- Cost $c_t = g(\boldsymbol{x}_t, \boldsymbol{u}_t)$

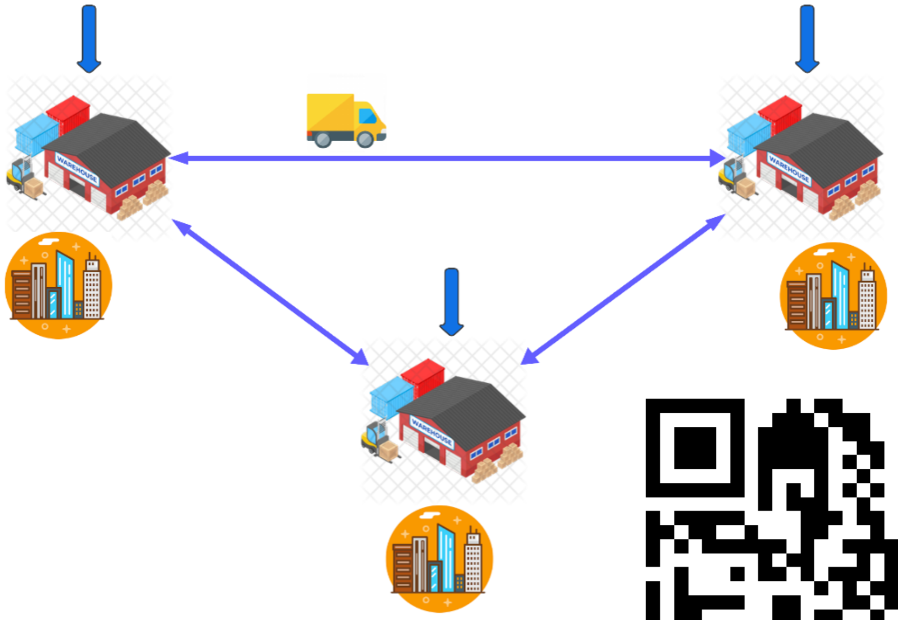
Consider a city divided into three districts.
Each district features a dedicated warehouse $w \in W = \{1,2,3\}$ which serves the district's demand $\mathrm{D}_{w,t}$ for coffee.
The coffee demand for each warehouse and day is known.

Each warehouse can store coffee up to a capacity limit C^{store}.
Denote the storage level of w at t by $z_{w,t}$.
At stage t, each warehouse w can order an amount $o_{w,t}$ of coffee from external suppliers at price $p_{w,t}$.
The price is different for each warehouse and each day.

Neighboring warehouses can also exchange coffee between them.
Let $y^{rcv}_{w,q,t}$ denote the amount received by w from a neighboring warehouse q, at t.
Similarly, $y^{send}_{w,q,t}$ is the amount sent by w to q.
To send an amount $y^{send}_{w,q,t}$, warehouse w must already have this amount previously stored.
The amount sent in one stage is restricted by a transportation limit C^{trnsp}.
Each exchange comes at a per-unit transportation cost $\mathrm{e}_{w,q}$.
Failing to meet a district's demand at any day comes at a per-unit cost of b_w.

Your job is to build a program that makes the day-by-day decisions for the coffee distribution system of the three warehouses, so that the city's coffee needs are met at the minimum expected cost.

The Lead Example



- Stages $t \in \mathcal{T}$
- State $\boldsymbol{x}_t = \{x_{1,t}, x_{2,t}, \dots\}$
- Decision $\boldsymbol{u}_t = \{u_{1,t}, u_{2,t}, \dots\}$
- Dynamics $\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t)$
- Cost $c_t = g(\boldsymbol{x}_t, \boldsymbol{u}_t)$

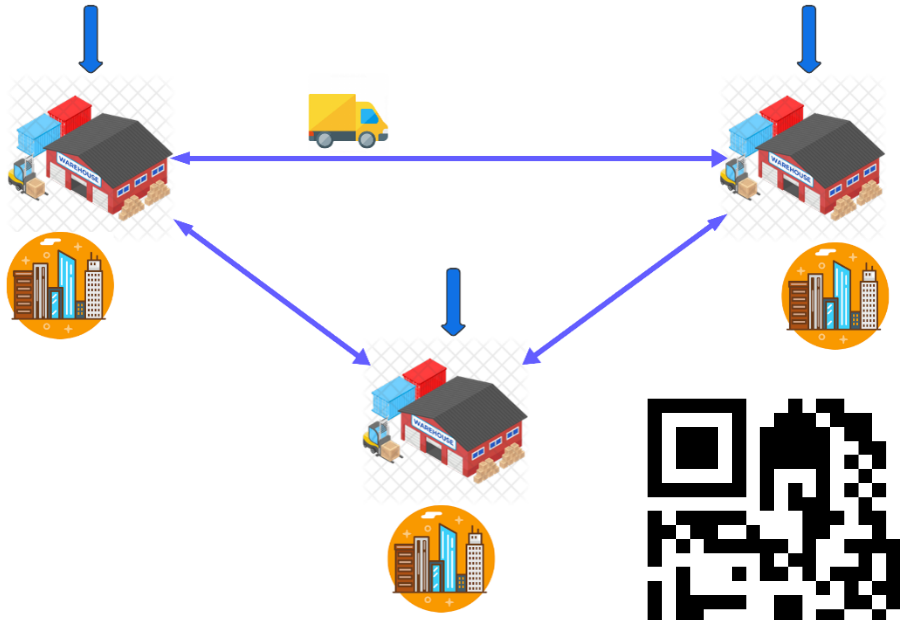
Consider a city divided into three districts.
Each district features a dedicated warehouse $w \in W = \{1,2,3\}$ which serves the district's demand $\mathrm{D}_{w,t}$ for coffee.
The coffee demand for each warehouse and day is known.

Each warehouse can store coffee up to a capacity limit C^{store}.
Denote the storage level of w at t by $z_{w,t}$.
At stage t, each warehouse w can order an amount $o_{w,t}$ of coffee from external suppliers at price $p_{w,t}$.
The price is different for each warehouse and each day.

Neighboring warehouses can also exchange coffee between them.
Let $y^{rcv}_{w,q,t}$ denote the amount received by w from a neighboring warehouse q, at t.
Similarly, $y^{send}_{w,q,t}$ is the amount sent by w to q.
To send an amount $y^{send}_{w,q,t}$, warehouse w must already have this amount previously stored.
The amount sent in one stage is restricted by a transportation limit C^{trnsp}.
Each exchange comes at a per-unit transportation cost $\mathrm{e}_{w,q}$.
Failing to meet a district's demand at any day comes at a per-unit cost of b_w. $d_{w,t}$

Your job is to build a program that makes the day-by-day decisions for the coffee distribution system of the three warehouses, so that the city's coffee needs are met at the minimum expected cost.

The Lead Example



- Stages $t \in \mathcal{T}$
- State $\boldsymbol{x}_t = \{x_{1,t}, x_{2,t}, \ldots\}$
- Decision $\boldsymbol{u}_t = \{u_{1,t}, u_{2,t}, \ldots\}$
- Dynamics $\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t)$
- Cost $c_t = g(\boldsymbol{x}_t, \boldsymbol{u}_t)$

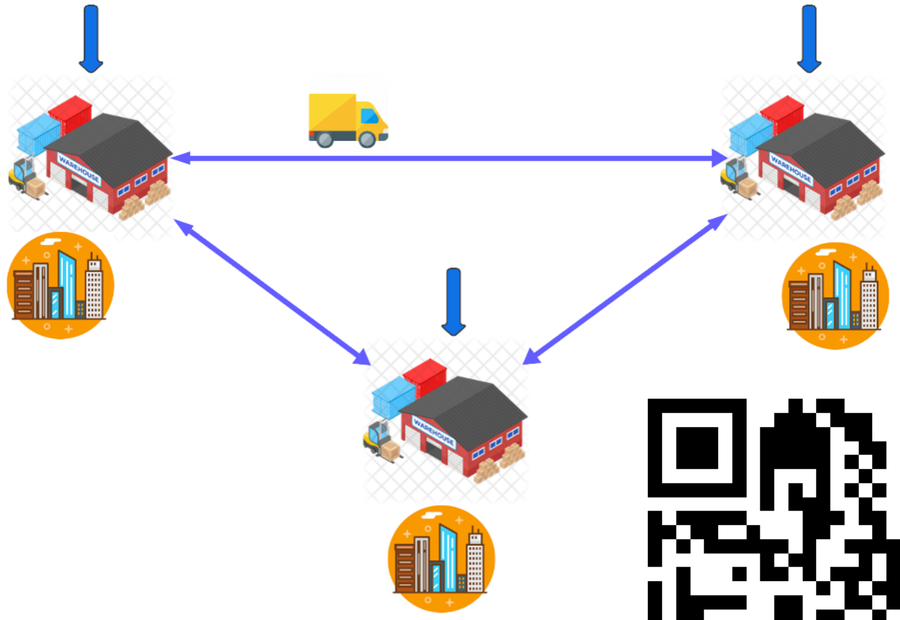
Consider a city divided into three districts.
Each district features a dedicated warehouse $w \in W = \{1,2,3\}$ which serves the district's demand $\mathrm{D}_{w,t}$ for coffee.
The coffee demand for each warehouse and day is known.

Each warehouse can store coffee up to a capacity limit C^{store}.
Denote the storage level of w at t by $z_{w,t}$.
At stage t, each warehouse w can order an amount $o_{w,t}$ of coffee from external suppliers at price $p_{w,t}$.
The price is different for each warehouse and each day.

Neighboring warehouses can also exchange coffee between them.
Let $y^{rcv}_{w,q,t}$ denote the amount received by w from a neighboring warehouse q, at t.
Similarly, $y^{send}_{w,q,t}$ is the amount sent by w to q.
To send an amount $y^{send}_{w,q,t}$, warehouse w must already have this amount previously stored.
The amount sent in one stage is restricted by a transportation limit C^{trnsp}.
Each exchange comes at a per-unit transportation cost $\mathrm{e}_{w,q}$.
Failing to meet a district's demand at any day comes at a per-unit cost of b_w. $d_{w,t}$

Your job is to build a program that makes the day-by-day decisions for the coffee distribution system of the three warehouses, so that the city's coffee needs are met at the minimum expected cost.

The Lead Example



- Stages $t \in \mathcal{T}$
- State $\boldsymbol{x}_t = \{x_{1,t}, x_{2,t}, \dots\}$
- Decision $\boldsymbol{u}_t = \{u_{1,t}, u_{2,t}, \dots\}$
- Dynamics $\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t)$
- Cost $c_t = g(\boldsymbol{x}_t, \boldsymbol{u}_t)$

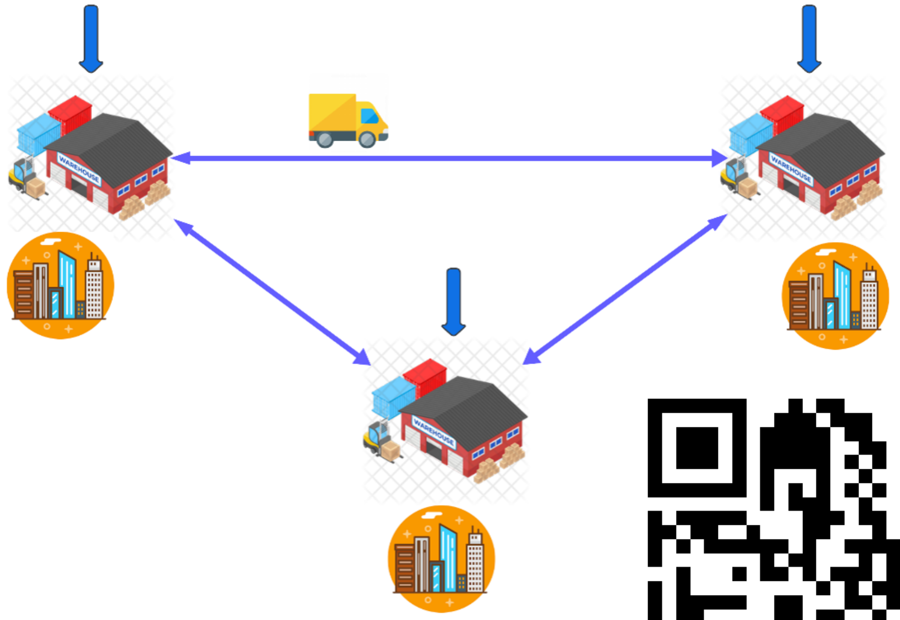
Consider a city divided into three districts.
Each district features a dedicated warehouse $w \in W = \{1,2,3\}$ which serves the district's demand $\mathrm{D}_{w,t}$ for coffee.
The coffee demand for each warehouse and day is known.

Each warehouse can store coffee up to a capacity limit C^{store}.
Denote the storage level of w at t by $z_{w,t}$.
At stage t, each warehouse w can order an amount $o_{w,t}$ of coffee from external suppliers at price $p_{w,t}$.
The price is different for each warehouse and each day.

Neighboring warehouses can also exchange coffee between them.
Let $y^{rcv}_{w,q,t}$ denote the amount received by w from a neighboring warehouse q, at t.
Similarly, $y^{send}_{w,q,t}$ is the amount sent by w to q.
To send an amount $y^{send}_{w,q,t}$, warehouse w must already have this amount previously stored.
The amount sent in one stage is restricted by a transportation limit C^{trnsp}.
Each exchange comes at a per-unit transportation cost $\mathrm{e}_{w,q}$.
Failing to meet a district's demand at any day comes at a per-unit cost of b_w. $d_{w,t}$

Your job is to build a program that makes the day-by-day decisions for the coffee distribution system of the three warehouses, so that the city's coffee needs are met at the minimum expected cost.

The Lead Example



- Stages $t \in \mathcal{T}$
- State $\boldsymbol{x}_t = \{x_{1,t}, x_{2,t}, \dots\}$
- Decision $\boldsymbol{u}_t = \{u_{1,t}, u_{2,t}, \dots\}$
- Dynamics $\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t)$
- Cost $c_t = g(\boldsymbol{x}_t, \boldsymbol{u}_t)$

Consider a city divided into three districts.
Each district features a dedicated warehouse $w \in W = \{1,2,3\}$ which serves the district's demand $\mathrm{D}_{w,t}$ for coffee.
The coffee demand for each warehouse and day is known.

Each warehouse can store coffee up to a capacity limit C^{store}.
Denote the storage level of w at t by $z_{w,t}$.
At stage t, each warehouse w can order an amount $o_{w,t}$ of coffee from external suppliers at price $p_{w,t}$.
The price is different for each warehouse and each day.

Neighboring warehouses can also exchange coffee between them.
Let $y^{rcv}_{w,q,t}$ denote the amount received by w from a neighboring warehouse q, at t.
Similarly, $y^{send}_{w,q,t}$ is the amount sent by w to q.
To send an amount $y^{send}_{w,q,t}$, warehouse w must already have this amount previously stored.
The amount sent in one stage is restricted by a transportation limit C^{trnsp}.
Each exchange comes at a per-unit transportation cost $\mathrm{e}_{w,q}$.
Failing to meet a district's demand at any day comes at a per-unit cost of b_w. $d_{w,t}$

Your job is to build a program that makes the day-by-day decisions for the coffee distribution system of the three warehouses, so that the city's coffee needs are met at the minimum expected cost.

The Lead Example

$$\cancel{z_{w,t+1} = z_{w,t} - D_{w,t} + o_{w,t} + y^{rcv}_{w,q,t}}$$

$$\cancel{z_{w,t+1} = z_{w,t} - D_{w,t} + o_{w,t} + \sum_{q \in W} y^{rcv}_{w,q,t}}$$

$$z_{w,t+1} = z_{w,t} - d_{w,t} + o_{w,t} + \sum_{q \in W} y^{rcv}_{w,q,t} - \sum_{q \in W} y^{send}_{w,q,t}$$

- Stages $t \in \mathcal{T}$
- State $\boldsymbol{x}_t = \{x_{1,t}, x_{2,t}, \ldots\}$
- Decision $\boldsymbol{u}_t = \{u_{1,t}, u_{2,t}, \ldots\}$
- Dynamics $\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t)$
- Cost $c_t = g(\boldsymbol{x}_t, \boldsymbol{u}_t)$

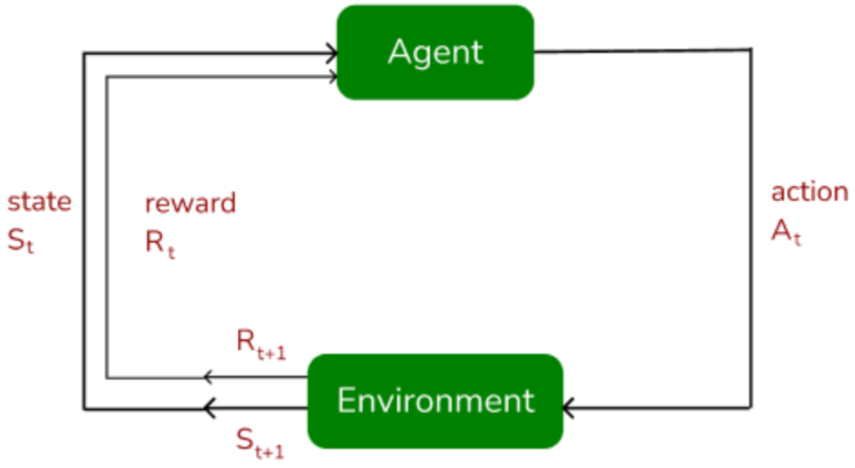
Consider a city divided into three districts.
Each district features a dedicated warehouse $w \in W = \{1,2,3\}$ which serves the district's demand $D_{w,t}$ for coffee.
The coffee demand for each warehouse and day is known.

Each warehouse can store coffee up to a capacity limit C^{store}.
Denote the storage level of w at t by $z_{w,t}$.
At stage t, each warehouse w can order an amount $o_{w,t}$ of coffee from external suppliers at price $p_{w,t}$.
The price is different for each warehouse and each day.

Neighboring warehouses can also exchange coffee between them.
Let $y^{rcv}_{w,q,t}$ denote the amount received by w from a neighboring warehouse q, at t.
Similarly, $y^{send}_{w,q,t}$ is the amount sent by w to q.
To send an amount $y^{send}_{w,q,t}$, warehouse w must already have this amount previously stored.
The amount sent in one stage is restricted by a transportation limit C^{trnsp}.
Each exchange comes at a per-unit transportation cost $e_{w,q}$.
Failing to meet a district's demand at any day comes at a per-unit cost of b_w.

Your job is to build a program that makes the day-by-day decisions for the coffee distribution system of the three warehouses, so that the city's coffee needs are met at the minimum expected cost.

Simulation Environment



Consider a city divided into three districts.
Each district features a dedicated warehouse $w \in W = \{1,2,3\}$ which serves the district's demand $\mathrm{D}_{w,t}$ for coffee.
The coffee demand for each warehouse and day is known.

Each warehouse can store coffee up to a capacity limit C^{store}.
Denote the storage level of w at t by $z_{w,t}$.
At stage t, each warehouse w can order an amount $o_{w,t}$ of coffee from external suppliers at price $p_{w,t}$.
The price is different for each warehouse and each day.

Neighboring warehouses can also exchange coffee.
Let $y^{rcv}_{w,q,t}$ denote the amount received by w from a neighboring warehouse q, at t.
Similarly, $y^{send}_{w,q,t}$ is the amount sent by w to q.
To send an amount $y^{send}_{w,q,t}$, warehouse w must already have this amount previously stored.
The amount sent in one stage is restricted by a transportation limit C^{trnsp}.
Each exchange comes at a per-unit transportation cost $\mathrm{e}_{w,q}$.
Failing to meet a district's demand at any day comes at a per-unit cost b_w.

Your job is to build a program that makes the day-by-day decisions for the coffee distribution system of the three warehouses, so that the city's coffee needs are met at the minimum expected cost.

Simulation Environment

$$c_t = \sum_{w \in W} \left(p_{w,t} o_{w,t} + \mathrm{e}_{w,q} y^{send}_{w,q,t} + \mathrm{b}_w \left(D_{w,t} - d_{w,t} \right) \right)$$

if $d_{w,t} < 0$:
  then $d_{w,t} = 0$

$$Z = z_{w,t} - d_{w,t} + o_{w,t} + \sum_{q \in W} y^{rcv}_{w,q,t} - \sum_{q \in W} y^{send}_{w,q,t} \ldots$$

if $Z > \mathrm{C}^{store}$:
  then $z_{w,t+1} = \mathrm{C}^{store}$
if $Z < 0$:
  then $z_{w,t+1} = 0$, $d_{w,t} = d_{w,t} - |Z|$
else: $z_{w,t+1} = Z$

if $y^{send}_{w,q,t} > \min\{z_{w,t}, \mathrm{C}^{trsnp}\}$:
  then $y^{send}_{w,q,t} = \min\{z_{w,t}, \mathrm{C}^{trsnp}\}$

if $y^{rcv}_{w,q,t} \neq y^{send}_{w,q,t}$:
  then $y^{rcv}_{w,q,t} = y^{send}_{w,q,t}$

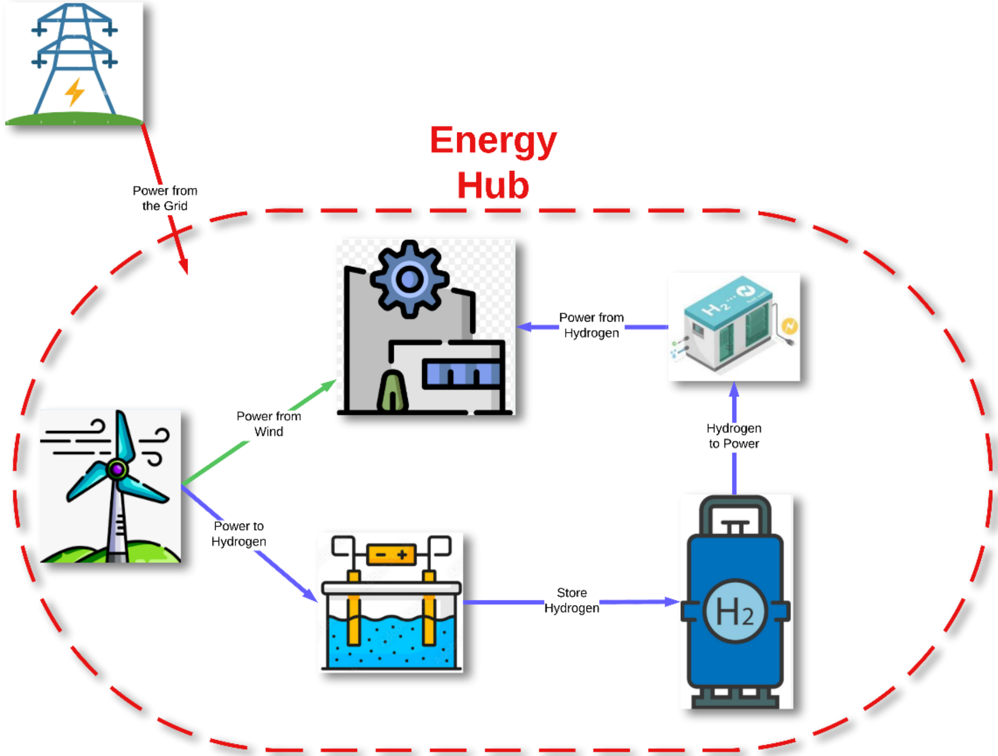
Consider a city divided into three districts.
Each district features a dedicated warehouse $w \in W = \{1,2,3\}$ which serves the district's demand $\mathrm{D}_{w,t}$ for coffee.
The coffee demand for each warehouse and day is known.

Each warehouse can store coffee up to a capacity limit C^{store}.
Denote the storage level of w at t by $z_{w,t}$.
At stage t, each warehouse w can order an amount $o_{w,t}$ of coffee from external suppliers at price $p_{w,t}$.
The price is different for each warehouse and each day.

Neighboring warehouses can also exchange coffee.
Let $y^{rcv}_{w,q,t}$ denote the amount received by w from a neighboring warehouse q, at t.
Similarly, $y^{send}_{w,q,t}$ is the amount sent by w to q.
To send an amount $y^{send}_{w,q,t}$, warehouse w must already have this amount previously stored.
The amount sent in one stage is restricted by a transportation limit C^{trnsp}.
Each exchange comes at a per-unit transportation cost $\mathrm{e}_{w,q}$.
Failing to meet a district's demand at any day comes at a per-unit cost b_w.

Your job is to build a program that makes the day-by-day decisions for the coffee distribution system of the three warehouses, so that the city's coffee needs are met at the minimum expected cost.

Assignment A, Task 1



Deliverable 1: MDP

State variables $\boldsymbol{x}_t = \{x_{1,t}, x_{2,t}, \ldots\}$

Decision variables $\boldsymbol{u}_t = \{u_{1,t}, u_{2,t}, \ldots\}$

Dynamics $\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t)$

Cost function $c_t = g(\boldsymbol{x}_t, \boldsymbol{u}_t)$

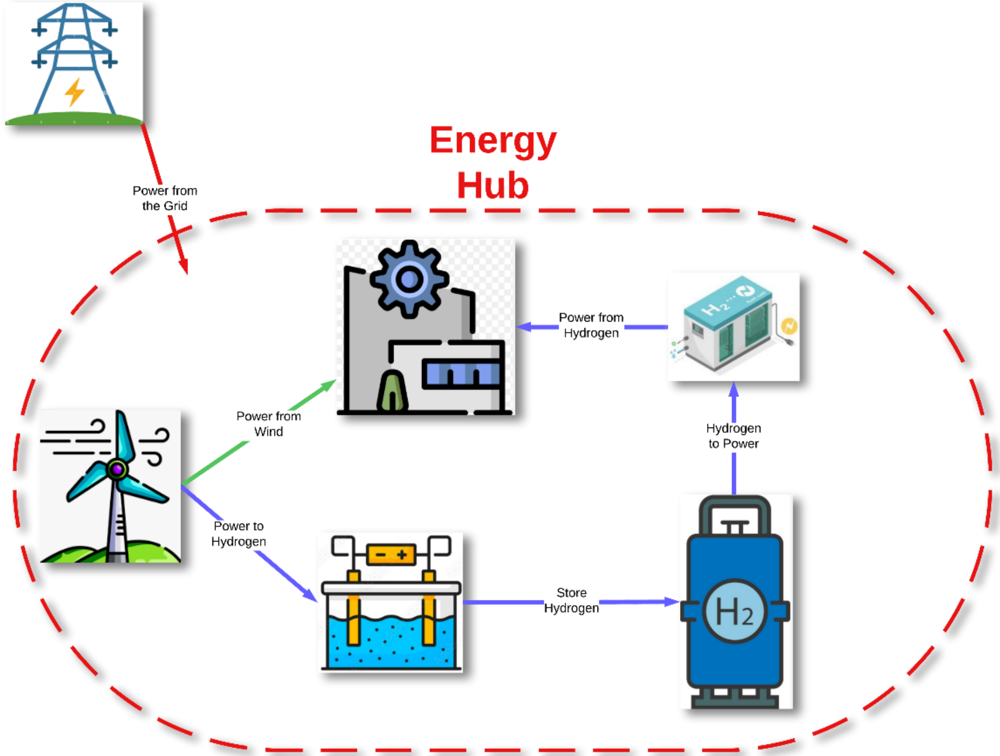
Deliverable 2: Policy Evaluation Framework

Input: *policy* (python function that returns decisions)
Initialize state variables
For experiment 1 to E:
    For stage 1 to H:
        decisions = *policy*(state)
        check/correct decisions if inconsistent
        calculate cost for this stage and experiment
        calculate state at next stage
    calculate total cost of policy for this experiment
Return: expected policy cost (average over experiments)

the environment need to check if the action are feasible, environment implements this, it claculates the cost and the next state by some random process and it return for the next state

Assignment A, Task 1

Constraints from Task 0



Deliverable 1: MDP

State variables $\boldsymbol{x}_t = \{x_{1,t}, x_{2,t}, \dots\}$

Decision variables $\boldsymbol{u}_t = \{u_{1,t}, u_{2,t}, \dots\}$

Dynamics $\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t)$

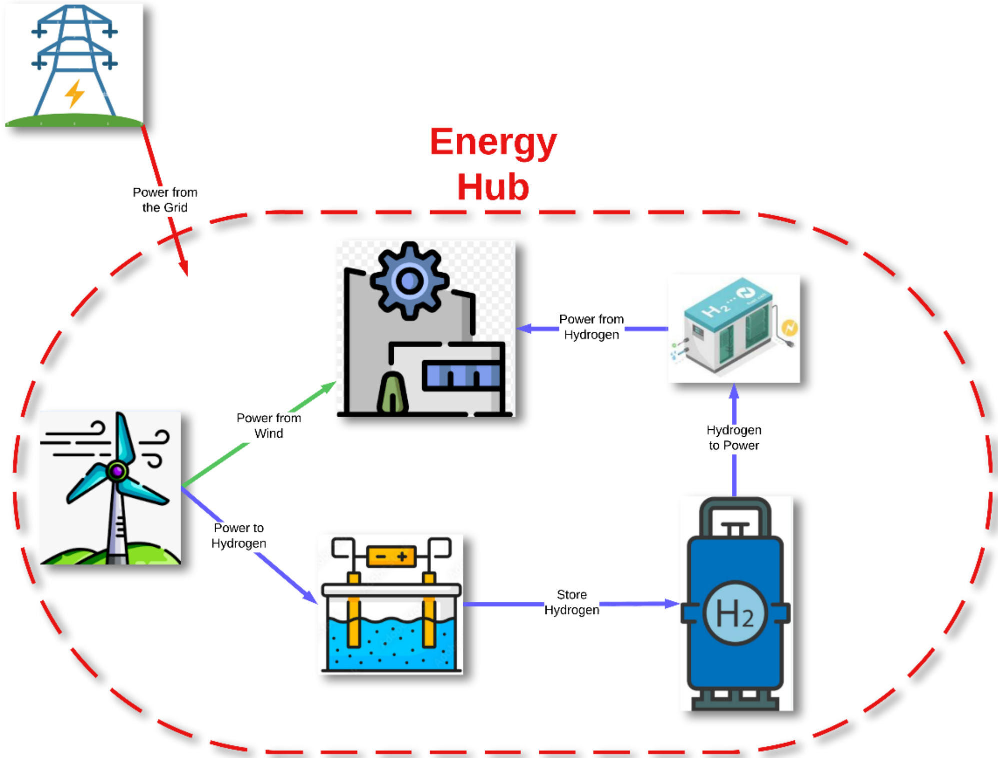
Cost function $c_t = g(\boldsymbol{x}_t, \boldsymbol{u}_t)$

Deliverable 2: Policy Evaluation Framework

Input: *policy* (python function that returns decisions)
Initialize state variables
For experiment 1 to E:
  For stage 1 to H:
    decisions = *policy*(state)
    check/correct decisions if inconsistent
    calculate cost for this stage and experiment
    calculate state at next stage
  calculate total cost of policy for this experiment
Return: expected policy cost (average over experiments)

this is linked with task 0, implements everything in the constratins, we should find all the conditions that should be cpmplied task 0 server as bench mark to compare with if we new everything before

Assignment A, Task 1



Constraints from Task 0

Deliverable 1: MDP

State variables $\boldsymbol{x}_t = \{x_{1,t}, x_{2,t}, \dots\}$

Decision variables $\boldsymbol{u}_t = \{u_{1,t}, u_{2,t}, \dots\}$

Dynamics $\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t)$

Cost function $c_t = g(\boldsymbol{x}_t, \boldsymbol{u}_t)$

Deliverable 2: Policy Evaluation Framework

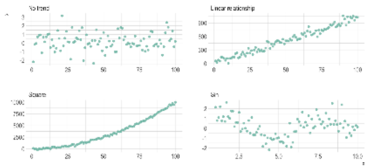
Input: *policy* (python function that returns decisions)
Initialize state variables
For experiment 1 to E:
    For stage 1 to H:
        decisions = *policy*(state)
        check/correct decisions if inconsistent
        calculate cost for this stage and experiment
        calculate state at next stage
    calculate total cost of policy for this experiment
Return: expected policy cost (average over experiments)

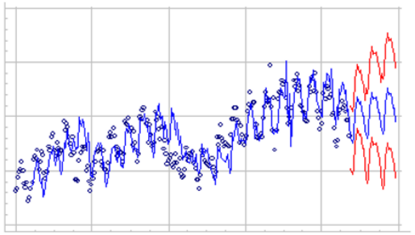
Some Perspective

Not in this course

we had already the forecasting, usually we do not have this stochastic problems that give us the next wind generation, usually we just have historicall data

Data → Forecasting Model → wind_process, price_process





we assume that someone did that for us

we can use samllples and we calcualte recursevily, we fit the current state and we can have the next wind, and son, we can generate all trajectories

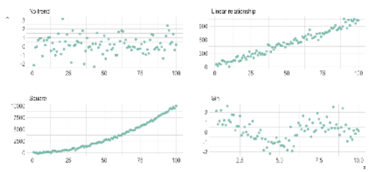
easieast policy, meet the demand
we can use linnear programming
we can solve optimization problem, we can generate point forecast and plug into the optimization, and we assume this is reallity and we solver a linear problem, linear, deterministic

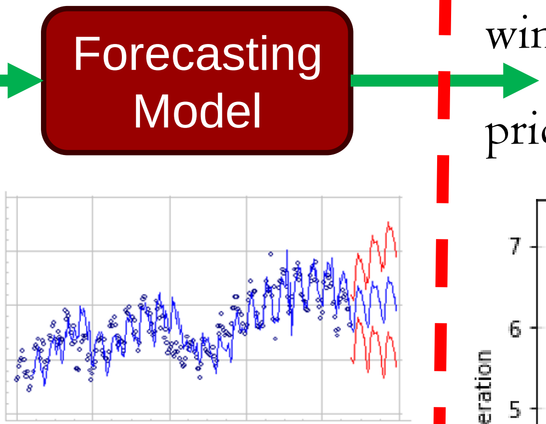
The Expected-Value Policy

Not in this course

At each stage:

1. Forecast uncertainties for a lookahead horizon
2. Solve a deterministic optimization program (similar to the OiH) for the lookahead horizon



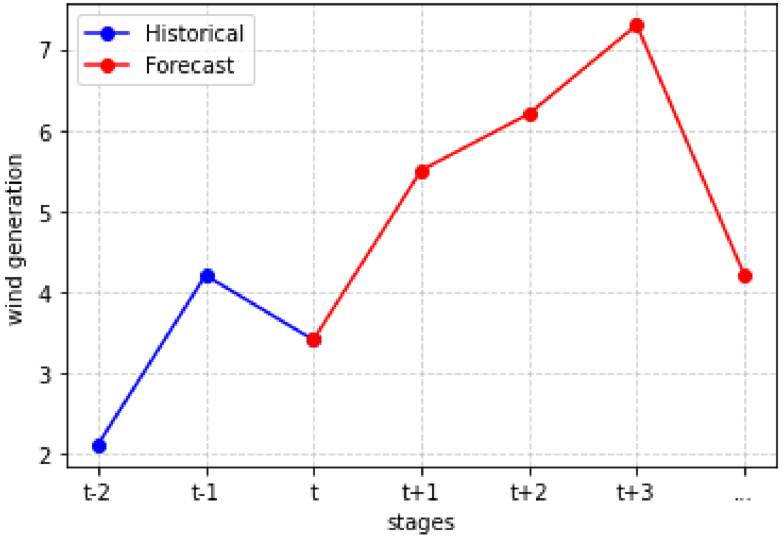


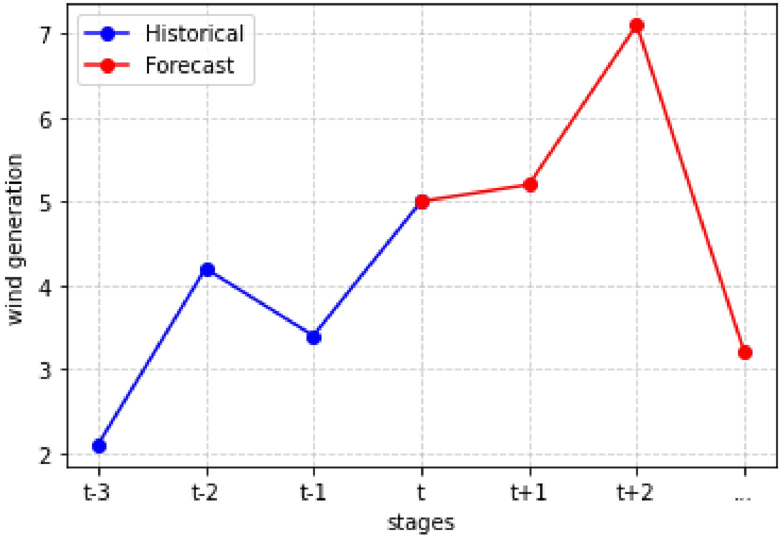
wind_process

price_process

i will look 10 days ahead and solve the forecastig, future values

once i do the decison the policy gave and i am going to the next statge, most proably my forecast will be wrogng, i will have a new wind generation, i take the new uncertainiy that is reavealed, this i my new state, i prodce new forecast, i solve the optimization as every state





The Expected-Value Policy

Input: current state x_0

Set a lookahead horizon length L

Calculate the expected scenario for the exogenous state variables for the horizon, i.e., $\tilde{x}_1 \tilde{x}_2, \dots, \tilde{x}_L$

Solve a deterministic optimization program (similar to the OiH) for the horizon:

$$\min_{u_0, u_1, \dots, u_L} \left\{ \sum_{t \in [0,L]} c_t \right\}$$

$$s.t. \quad constraints$$

we go for the horizon, samll optimiztons

Return: u_0 (and not $u_1, \dots, u_L$)

only the decisiion fro now, the decisoiins for tall the horizon are just to guide
policy is presented with the state, whaat will we do,

At each stage:

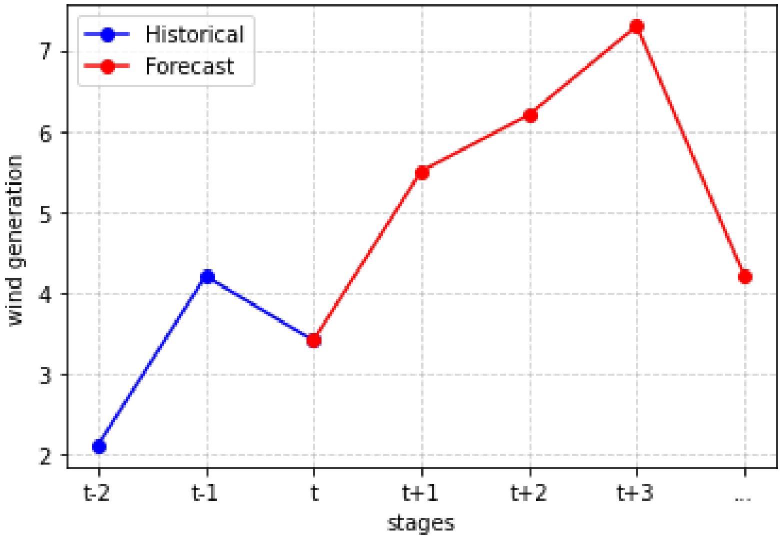
how long will i look ahead, with this horizon, i will create random samples, random wind, monte carlo for example, we simulate many samples and take the average to take the expected scenaario
i can do it when they exogeneus wehn they ddo not depend of my decisions - like weather, but if it is chess i cannot predict the next 10 plays, it will ddepend on my moves, it is depedent

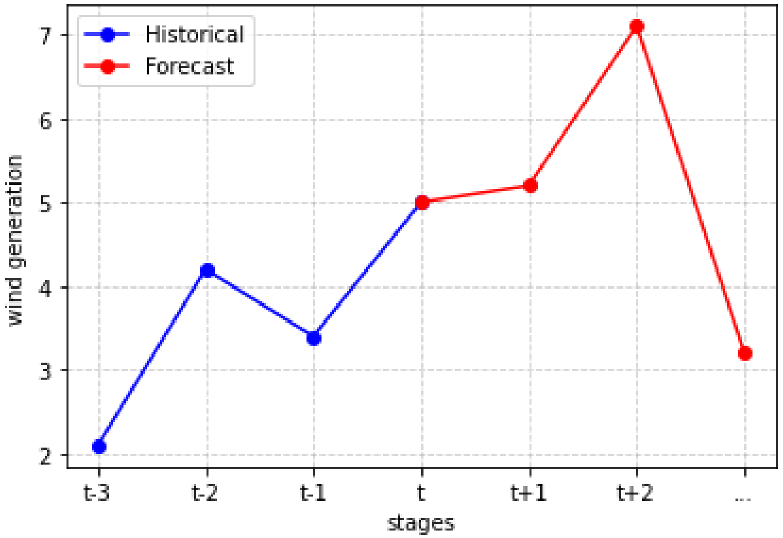
1. Forecast
2. Solve a deterministic optimization program (similar to the OiH) for a lookahead horizon

wind_process

price_process

Simulate (Monte Carlo) many samples and calculate the average





u_0 effective decsion, u_!, uL are prespective decsioson,s, they are not biding, we do not stick with them

Two-stage Stochastic Programming

Why two-stage?

a) Easier to understand and sometimes it’s a good enough policy

b) Some problems are just naturally two-stage problems (e.g. investment planning)

invstment, in first state iwant to buikd, second, operacional, and then investing how much to invest considereing now this new inputs

Remember Carl?

Apart from having cows and sheep, Farmer Carl grows wheat, corn and sugar beets on his 500 acres of farmland. He has to decide at the beginning of the season how much space he wants to use for each of the crops. To plant one acre with wheat, corn or sugar beets costs him 150 €, 230 € or 260 €, respectively.

He needs at least 200 tons of wheat and 240 tons of corn to feed his cattle. If he does not produce enough wheat and corn himself, he can buy the missing amount on the market for 238 € and 210 € per ton, respectively. In the case of overproduction, he can sell his potential overproduction at a price of 170 € per ton wheat and 150 € per ton corn.

His sugar beet yield is meant for selling only and brings 36 € per ton. However, due to a given quota only for a maximal amount of 6000 tons. Every additional ton can be sold at a price of 10 € per ton only.

The average yield per acre of wheat, corn and sugar beets is 2.5, 3 and 20 tons, respectively.

Remember Carl?

Apart from having cows and sheep, Farmer Carl grows wheat, corn and sugar beets on his 500 acres of farmland. He has to decide at the beginning of the season how much space he wants to use for each of the crops. To plant one acre with wheat, corn or sugar beets costs him 150 €, 230 € or 260 €, respectively.
He needs at least 200 tons of wheat and 240 tons of corn to feed his cattle. If he does not produce enough wheat and corn himself, he can buy the missing amount on the market for 238 € and 210 € per ton, respectively. In the case of overproduction, he can sell his potential overproduction at a price of 170 € per ton wheat and 150 € per ton corn.
His sugar beet yield is meant for selling only and brings 36 € per ton. However, due to a given quota only for a maximal amount of 6000 tons. Every additional ton can be sold at a price of 10 € per ton only.
The average yield per acre of wheat, corn and sugar beets is 2.5, 3 and 20 tons, respectively.

x^W, x^C, x^S	Acres used for wheat, corn, sugar beets
y^W, y^C	Tons of wheat and corn purchased
z^W, z^C	Tons of wheat and corn sold
z^S	Tons of sugar beets sold at price 36 €
v^S	Tons of sugar beets sold at price 10 €

$$
\begin{aligned}
\text{Max } Profit = & 170z^W + 150z^C + 36z^S + 10v^S \\
& - 150x^W - 230x^C - 260x^S - 238y^W - 210y^C \\
\text{s.t. } & x^W + x^C + x^S \leq 500 \\
& 2.5x^W + y^W - z^W \geq 200 \\
& 3.0x^C + y^C - z^C \geq 240 \\
& z^S \leq 6000 \\
& z^S + v^S \leq 20x^S \\
& x^W, x^C, x^S, y^W, y^C, z^W, z^C, z^S, v^S \geq 0
\end{aligned}
$$

if we short we need to buy, if we have more we need to sell

Linear program formulation

x^W, x^C, x^S	Acres used for wheat, corn, sugar beets
y^W, y^C	Tons of wheat and corn purchased
z^W, z^C	Tons of wheat and corn sold
z^S	Tons of sugar beets sold at price 36 €
v^S	Tons of sugar beets sold at price 10 €

the varaibles will be the same and constraits as well

$$
\begin{aligned}
\text{Max } Profit = \; & 170z^W + 150z^C + 36z^S + 10v^S \\
& - 150x^W - 230x^C - 260x^S - 238y^W - 210y^C \\
\text{s.t. } \; & x^W + x^C + x^S \leq 500 \\
& \underline{2.5}x^W + y^W - z^W \geq 200 \\
& \underline{3.0}x^C + y^C - z^C \geq 240 \\
& z^S \leq 6000 \\
& z^S + v^S \leq 20\underline{x^S} \\
& x^W, x^C, x^S, y^W, y^C, z^W, z^C, z^S, v^S \geq 0
\end{aligned}
$$

Considering Uncertainty

The average yield per acre of wheat, corn and sugar beets is 2.5, 3 and 20 tons, respectively.

What if, the yield depends on weather and other circumstances in each year. In a good year, the yield can be higher while in a bad year, the yield can be lower. This table gives the yields in the different situations.

Yield	Wheat	Corn	Sugar beets	Probability
High (+20%)	3.0	3.6	24.0	0.2
Average	2.5	3.0	20.0	0.5
Low (-20%)	2.0	2.4	16.0	0.3

We solve the optimization model also for a year with high yield and for a year with low yield.

Stochastic Program Formulation

s.t. $x^W + x^C + x^S \leq 500 \qquad x^W, x^C, x^S \geq 0$

Stochastic Program Formulation

for each scenario we chage the yiel, we have 3 version, good, bad and average year

we need to include all the constraits we dont know each will be real

$$\text{s.t. } x^W + x^C + x^S \leq 500 \qquad x^W, x^C, x^S \geq 0$$

Scenario 1

$$\boxed{3.0x^W} + y_1^W - z_1^W \geq 200$$
$$3.6x^C + y_1^C - z_1^C \geq 240$$
$$z_1^S \leq 6000$$
$$z_1^S + v_1^S \leq 24x^S$$
$$y_1^W, y_1^C, z_1^W, z_1^C, z_1^S, v_1^S \geq 0$$

Scenario 2

$$\boxed{2.5x^W} + y_2^W - z_2^W \geq 200$$
$$3.0x^C + y_2^C - z_2^C \geq 240$$
$$z_2^S \leq 6000$$
$$z_2^S + v_2^S \leq 20x^S$$
$$y_2^W, y_2^C, z_2^W, z_2^C, z_2^S, v_2^S \geq 0$$

Scenario 3

$$\boxed{2.0x^W} + y_3^W - z_3^W \geq 200$$
$$2.4x^C + y_3^C - z_3^C \geq 240$$
$$z_3^S \leq 6000$$
$$z_3^S + v_3^S \leq 16x^S$$
$$y_3^W, y_3^C, z_3^W, z_3^C, z_3^S, v_3^S \geq 0$$

Stochastic Program Formulation

We optimize for the expected value over all scenarios

$$\text{Max } Profit = -150x^W - 230x^C - 260x^S$$
$$+ 0.2(170z_1^W + 150z_1^C + 36z_1^S + 10v_1^S - 238y_1^W - 210y_1^C)$$
$$+ 0.5(170z_2^W + 150z_2^C + 36z_2^S + 10v_2^S - 238y_2^W - 210y_2^C)$$
$$+ 0.3(170z_3^W + 150z_3^C + 36z_3^S + 10v_3^S - 238y_3^W - 210y_3^C)$$

they look the same but have a probability
we have varaibles definied for each state, if we have a good state we will sell thius mych z_!, if we dont seel that much z_3...

$$\text{s.t. } x^W + x^C + x^S \le 500 \qquad x^W, x^C, x^S \ge 0$$

what changes is the availability i ahve to seel or buy

Scenario 1

$$\boxed{3.0x^W} + y_1^W - z_1^W \ge 200$$
$$3.6x^C + y_1^C - z_1^C \ge 240$$
$$z_1^S \le 6000$$
$$z_1^S + v_1^S \le 24x^S$$
$$y_1^W, y_1^C, z_1^W, z_1^C, z_1^S, v_1^S \ge 0$$

Scenario 2

they are indexed by scenario

$$\boxed{2.5x^W} + y_2^W - z_2^W \ge 200$$
$$3.0x^C + y_2^C - z_2^C \ge 240$$
$$z_2^S \le 6000$$
$$z_2^S + v_2^S \le 20x^S$$
$$y_2^W, y_2^C, z_2^W, z_2^C, z_2^S, v_2^S \ge 0$$

Scenario 3

$$\boxed{2.0x^W} + y_3^W - z_3^W \ge 200$$
$$2.4x^C + y_3^C - z_3^C \ge 240$$
$$z_3^S \le 6000$$
$$z_3^S + v_3^S \le 16x^S$$
$$y_3^W, y_3^C, z_3^W, z_3^C, z_3^S, v_3^S \ge 0$$

General Formulation

x first-stage decision variables

y second-stage decision variables

Let $\omega \in \Omega$ be the **finite set of scenarios** of uncertainty and $\pi(\omega)$ the probability of scenario ω.

Extensive form / deterministic equivalent linear program:

cost to go

$$\min\ c^T x + \overbrace{\sum_{\omega \in \Omega} \pi(\omega) q(\omega)^T y(\omega)}^{\text{Expected value}}$$

expected cost to go we will talk later, coeffieciets, for the corn it was the prices to buy seel, q(w)

each constraint need t be sted to each scenario

$$\begin{aligned} \text{s.t.}\ & Ax = b \\ & T(\omega)x + W(\omega)y(\omega) = h(\omega) && \forall \omega \in \Omega \\ & x \geq 0 \\ & y(\omega) \geq 0 && \forall \omega \in \Omega \end{aligned}$$

Stochastic Program Solution

Stage	Scenario	Variable	Wheat	Corn	Sugar beets	Profit
1st	all	Acres	120	80	300	
2nd	High yield	Purchase	0	0	-	148000
		Sale	160	48	6000 (36 €)	
					1200 (10 €)	
	Average yield	Purchase	0	0	-	118600
		Sale	100	0	6000 (36 €)	
					0 (10 €)	
	Low yield	Purchase	0	48	-	55120
		Sale	40	0	4800 (36 €)	
					0 (10 €)	

Expected profit = 105436 €

i will need to purchase in a bad year

sum them with the given probabilities we have the provabilityif we dont know if it will be good or bad year

Stochastic Program vs Expected Value Program

Expected profit = 105436 €

$$\text{Max } Profit = -150x^W - 230x^C - 260x^S$$
$$+0.2(170z_1^W + 150z_1^C + 36z_1^S + 10v_1^S - 238y_1^W - 210y_1^C)$$
$$+0.5(170z_2^W + 150z_2^C + 36z_2^S + 10v_2^S - 238y_2^W - 210y_2^C)$$
$$+0.3(170z_3^W + 150z_3^C + 36z_3^S + 10v_3^S - 238y_3^W - 210y_3^C)$$

$$\text{s.t. } x^W + x^C + x^S \leq 500 \qquad x^W, x^C, x^S \geq 0$$

Scenario 1	Scenario 2	Scenario 3
$\boxed{3.0x^W} + y_1^W - z_1^W \geq 200$	$\boxed{2.5x^W} + y_2^W - z_2^W \geq 200$	$\boxed{2.0x^W} + y_3^W - z_3^W \geq 200$
$3.6x^C + y_1^C - z_1^C \geq 240$	$3.0x^C + y_2^C - z_2^C \geq 240$	$2.4x^C + y_3^C - z_3^C \geq 240$
$z_1^S \leq 6000$	$z_2^S \leq 6000$	$z_3^S \leq 6000$
$z_1^S + v_1^S \leq 24x^S$	$z_2^S + v_2^S \leq 20x^S$	$z_3^S + v_3^S \leq 16x^S$
$y_1^W, y_1^C, z_1^W, z_1^C, z_1^S, v_1^S \geq 0$	$y_2^W, y_2^C, z_2^W, z_2^C, z_2^S, v_2^S \geq 0$	$y_3^W, y_3^C, z_3^W, z_3^C, z_3^S, v_3^S \geq 0$

Yield	Wheat	Corn	Sugar beets	Probability
High (+20%)	3.0	3.6	24.0	0.2
Average	2.5	3.0	20.0	0.5
Low (-20%)	2.0	2.4	16.0	0.3
Expected	2.45	2.94	19.6	

Expected profit = 113545.92 €

i dont bother myself wit texoected scenarios, i summe them up and i get the expected scneario
i only have one scnerio i o the average witht the probablility

$$\text{Max } Profit = 170z^W + 150z^C + 36z^S + 10v^S$$
$$-150x^W - 230x^C - 260x^S - 238y^W - 210y^C$$
$$\text{s.t. } x^W + x^C + x^S \leq 500$$
$$2.45x^W + y^W - z^W \geq 200$$
$$2.94x^C + y^C - z^C \geq 240$$
$$z^S \leq 6000$$
$$z^S + v^S \leq 19.6x^S$$
$$x^W, x^C, x^S, y^W, y^C, z^W, z^C, z^S, v^S \geq 0$$

we need to validate we need to simulate, first we apply and then reallity kicks in and after the uncertainity is reavealed i need to decide,

Stochastic Program vs Expected Value Program

Expected profit $= 105436$ €

$$\begin{aligned} \text{Max } Profit = & -150x^W - 230x^C - 260x^S \\ & + 0.2(170z_1^W + 150z_1^C + 36z_1^S + 10v_1^S - 238y_1^W - 210y_1^C) \\ & + 0.5(170z_2^W + 150z_2^C + 36z_2^S + 10v_2^S - 238y_2^W - 210y_2^C) \\ & + 0.3(170z_3^W + 150z_3^C + 36z_3^S + 10v_3^S - 238y_3^W - 210y_3^C) \end{aligned}$$

s.t. $x^W + x^C + x^S \leq 500 \qquad x^W, x^C, x^S \geq 0$

Scenario 1	Scenario 2	Scenario 3
$\boxed{3.0x^W} + y_1^W - z_1^W \geq 200$	$\boxed{2.5x^W} + y_2^W - z_2^W \geq 200$	$\boxed{2.0x^W} + y_3^W - z_3^W \geq 200$
$3.6x^C + y_1^C - z_1^C \geq 240$	$3.0x^C + y_2^C - z_2^C \geq 240$	$2.4x^C + y_3^C - z_3^C \geq 240$
$z_1^S \leq 6000$	$z_2^S \leq 6000$	$z_3^S \leq 6000$
$z_1^S + v_1^S \leq 24x^S$	$z_2^S + v_2^S \leq 20x^S$	$z_3^S + v_3^S \leq 16x^S$
$y_1^W, y_1^C, z_1^W, z_1^C, z_1^S, v_1^S \geq 0$	$y_2^W, y_2^C, z_2^W, z_2^C, z_2^S, v_2^S \geq 0$	$y_3^W, y_3^C, z_3^W, z_3^C, z_3^S, v_3^S \geq 0$

Simulate second stage uncertainty realization (experiments) and evaluate the "actual" profit…

Expected profit $= 104156.7$ € So, actually less than the SP

~~Expected profit $= 113545.92$ €~~

$$\begin{aligned} \text{Max } Profit = & 170z^W + 150z^C + 36z^S + 10v^S \\ & - 150x^W - 230x^C - 260x^S - 238y^W - 210y^C \end{aligned}$$

$$\begin{aligned} \text{s.t. } & x^W + x^C + x^S \leq 500 \\ & 2.45x^W + y^W - z^W \geq 200 \\ & 2.94x^C + y^C - z^C \geq 240 \\ & z^S \leq 6000 \\ & z^S + v^S \leq 19.6x^S \\ & x^W, x^C, x^S, y^W, y^C, z^W, z^C, z^S, v^S \geq 0 \end{aligned}$$

we dont stick what solvers predicts, we need to only take first decsions and we implement these and then after observing reality, only then we decidde what we will actually sell in the market, the recall decisions, and we simulate with thenvironment then

Connection to the Assignment

1. Optimal in Hindsight solution (Task 0)
2. Expected Value policy (Task 2)
3. 2-stage Stochastic Programming policy (Task 2)

we can use the code to simulate one scenario

Develop a simulation environment to evaluate a policy (Task 1)

Evaluate each of the above over the same experiments
to evaluate each one's expected cost (Task 4)

stochastic performs better then expected

Which of the three will have the highest, and
which the lowest average cost?

Literature

Ruszczynski A, Shapiro A (2003): *Chapter 1 - Stochastic Programming Models*, Handbooks in Operations Research and Management Science, Vol. 10

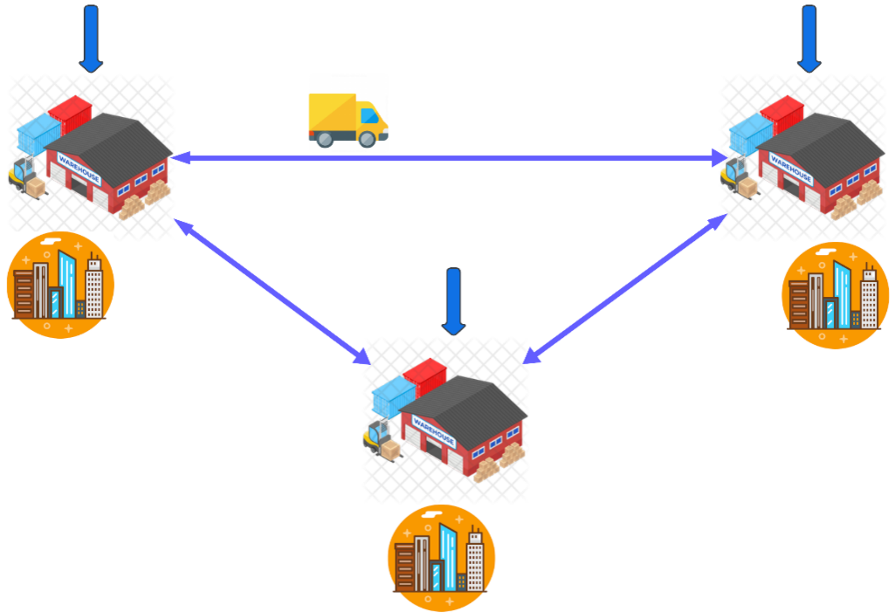
Relevant pages: 1-22

The farmer example is based on:

Birge J R, Louveaux F (1997): *Introduction to Stochastic Programming*, Springer

Relevant pages: 4-10

Back to Coffee



Consider a city divided into three districts.
Each district features a dedicated warehouse $w \in W = \{1,2,3\}$ which serves the district's demand $\mathrm{D}_{w,t}$ for coffee.
The coffee demand for each warehouse and day is known.

Each warehouse can store coffee up to a capacity limit C^{store}.
Denote the storage level of w at t by $z_{w,t}$.
At stage t, each warehouse w can order an amount $o_{w,t}$ of coffee from external suppliers at price $p_{w,t}$.
The price is different for each warehouse and each day.

Neighboring warehouses can also exchange coffee between them.
Let $y^{rcv}_{w,q,t}$ denote the amount received by w from a neighboring warehouse q, at t.
Similarly, $y^{send}_{w,q,t}$ is the amount sent by w to q.
To send an amount $y^{send}_{w,q,t}$, warehouse w must already have this amount previously stored.
The amount sent in one stage is restricted by a transportation limit C^{trnsp}.
Each exchange comes at a per-unit transportation cost $\mathrm{e}_{w,q}$.
Failing to meet a district's demand at any day comes at a per-unit cost of b_w.

Your job is to build a program that makes the day-by-day decisions for the coffee distribution system of the three warehouses, so that the city's coffee needs are met at the minimum expected cost.

Coffee Problem Nomenclature

Sets

- $w, q \in W$: set of warehouses, where w and q belong to $\{1, 2, 3\}$
- $t \in T$: set of timeslots (daily), in $\{1, 2, ...\}$

Parameters

- $D_{w,t}$: coffee demand for warehouse w in period t, being constant at 4 units per day
- $C_w^{storage}$: storage capacity limit for warehouse w
- $C_{w,q}^{transp}$: daily transportation capacity limit for what warehouse w can send to warehouse q, where if $w = q$ then the capacity is zero
- $p_{w,t}$: external coffee price for warehouse w at time t, continuous in $[0, 10]$
- $e_{w,q}$: transportation per-unit cost between warehouse w and q, where if $w = q$ then the cost is zero
- b_w: per-unit cost of missing daily demand for warehouse w

Variables

- $x_{w,t}$: continuous, amount of coffee ordered in timeslot t by warehouse w
- $z_{w,t}$: continuous, amount of coffee stored by the end of timeslot t in warehouse w
- $m_{w,t}$: continuous, amount of coffee missing when daily demand is not met in warehouse w in t
- $y_{w,q,t}^{send}$: continuous, amount of coffee sent by warehouse w to q in timeslot t
- $y_{w,q,t}^{receive}$: continuous, amount of coffee received by warehouse w from q in timeslot t

Coffee Problem OiH formulation

Minimize the total cost of the coffee distribution system (orders, transfers and missed) so that the demands are met. The objective function includes all 3 costs: orders placed, transport between warehouses and failing to meet the demand.

$$\min \quad \sum_{w\in W}\sum_{t\in T} x_{w,t}\cdot p_{w,t} + \sum_{w\in W}\sum_{q\in W}\sum_{t\in T} y^{send}_{w,q,t}\cdot e_{w,q} + \sum_{w\in W}\sum_{t\in T} m_{w,t}\cdot b_w \tag{1}$$

Constraints

- To always respect the truck (transport) capacity between the warehouses

$$y^{send}_{w,q,t} \le C^{transp}_{w,q} \qquad \forall w,q \in W, t\in T \tag{2}$$

- To have the same quantity being received and sent between two warehouses in a specific period

$$y^{send}_{w,q,t} = y^{receive}_{q,w,t} \qquad \forall w,q \in W, t\in T \tag{3}$$

- To always respect the storage capacity in all warehouses

$$z_{w,t} \le C^{storage}_{w} \qquad \forall w \in W, t\in T \tag{4}$$

- To ensure demand fulfillment in all warehouses in $t \in T$

$$x_{w,t} + m_{w,t} + z_{w,t-1} + \sum_{q\in W} y^{receive}_{w,q,t} = D_{w,t} + z_{w,t} + \sum_{q\in W} y^{send}_{w,q,t} \qquad \forall w \in W, t\in T \tag{5}$$

- The amount sent between warehouses can only be determined by the amount stored in the previous time slot

$$\sum_{q\in W} y^{send}_{w,q,t} \le z_{w,t-1} \qquad \forall w \in W, t\in T \tag{6}$$

- Non-negativity for all variables

$$\begin{aligned} x_{w,t} \ge 0, z_{w,t} \ge 0, m_{w,t} \ge 0 \qquad & \forall w \in W, t\in T \\ y^{send}_{w,q,t} \ge 0, y^{receive}_{w,q,t} \ge 0 \qquad & \forall w \in W, t\in T \end{aligned} \tag{7}$$

Coffee Problem 2-stage Stochastic Program

Sets

- $s \in S$: set of scenarios, in $\{1, 2, ...\}$

Parameters

- $p1_w$: external coffee price for warehouse w on day 1, continuous in $[0, 10]$
- $p2_{w,s}$: external coffee price for warehouse w on day 2 in scenario s, continuous in $[0, 10]$
- $prob_s$: probability of occurrence of scenario s.
- $z0_w$: initial stock for each warehouse in $t = 1$.

The previously defined variables had to be separated into 1^{st} stage (equal for all scenarios) and 2^{nd} stage (different for each scenario) in the following way:

Variables : First $(t = 1)$ and Second $(t = 2)$ Stage Decision

- $x1_w$: continuous, amount of coffee ordered in timeslot 1 by warehouse w.
- $x2_{w,s}$: continuous, amount of coffee ordered in timeslot 2 by warehouse w in scenario s.
- $z1_w$: continuous, amount of coffee stored by the end of timeslot 1 in warehouse w.
- $z2_{w,s}$: continuous, amount of coffee stored by the end of timeslot 2 in warehouse w in scenario s.
- $m1_w$: continuous, amount of coffee missing when daily demand is not met in warehouse w in timeslot 1.
- $m2_{w,s}$: continuous, amount of coffee missing when daily demand is not met in warehouse w in timeslot 2 in scenario s.
- $y1^{send}_{w,q}$: continuous, amount of coffee sent by warehouse w to q in timeslot 1.
- $y2^{send}_{w,q,s}$: continuous, amount of coffee sent by warehouse w to q in timeslot 2 in scenario s.
- $y1^{receive}_{w,q}$: continuous, amount of coffee received by warehouse w from q in timeslot 1.
- $y2^{receive}_{w,q,s}$: continuous, amount of coffee received by warehouse w from q in timeslot 2 in scenario s.

1:
X_stage_1
X_stage_2

2:
X_w,s,t
X_w,s,1 = X_w,s',1 fora all s,s' €S

Why do they define different variables for stage 1 and stage 2?

first decisions there are no scenrios, we nkow the here and now proces, we just need for the future

we need to make sure we are just choosing one scneario on the first stage, not several scenarios

if we have 2 stage policy we can go with the first problem and if we have deifferent varaibles, when it is a multi stage program, how much coffee to order, quickly become not easy to handle, but it is more pratically to define it as second ways

Coffee Problem 2-stage SP Objective Function

$$\min \sum_{w\in W} x1_w \cdot p1_w + \sum_{w\in W}\sum_{s\in S} x2_{w,s} \cdot p2_{w,s} \cdot prob_s$$
$$+ \sum_{w\in W}\sum_{q\in W} ys1_{w,q} \cdot e_{w,q} + \sum_{w\in W}\sum_{q\in W}\sum_{s\in S} ys2_{w,q,s} \cdot e_{w,q} \cdot prob_s$$
$$+ \sum_{w\in W} m1_w \cdot b_w + \sum_{w\in W}\sum_{s\in S} m2_{w,s} \cdot b_w \cdot prob_s \quad (8)$$

Coffee Problem 2-stage SP Constraints

Constraints

- To always respect the truck (transport) capacity between the warehouses

$$y1^{send}_{w,q} \leq C^{transp}_{w,q} \qquad \forall w, q \in W \tag{9}$$

$$y2^{send}_{w,q,s} \leq C^{transp}_{w,q} \qquad \forall w, q \in W, \boxed{s \in S} \tag{10}$$

- To have the same quantity being received and sent between two warehouses in a specific period

$$y1^{send}_{w,q} = y1^{receive}_{q,w} \qquad \forall w, q \in W \tag{11}$$

$$y2^{send}_{w,q,s} = y2^{receive}_{q,w,s} \qquad \forall w, q \in W \boxed{s \in S} \tag{12}$$

- To always respect the storage capacity in all warehouses

$$z1_w \leq C^{storage}_w \qquad \forall w \in W \tag{13}$$

$$z2_{w,s} \leq C^{storage}_w \qquad \forall w \in W, \boxed{s \in S} \tag{14}$$

- To ensure demand fulfillment in all warehouses in $t \in T$

$$x1_w + m1_w + z0_w + \sum_{q \in W} y1^{receive}_{w,q} = D_{w,1} + z1_w + \sum_{q \in W} y1^{send}_{w,q} \qquad \forall w \in W \tag{15}$$

$$x2_{w,s} + m2_{w,s} + z1_w + \sum_{q \in W} y2^{receive}_{w,q,s} = D_{w,2} + z2_{w,s} + \sum_{q \in W} y2^{send}_{w,q,s} \qquad \forall w \in W, \boxed{s \in S} \tag{16}$$

- The amount sent between warehouses can only be determined by the amount stored in the previous time slot

$$\sum_{q \in W} y1^{send}_{w,q} \leq z0_w \qquad \forall w \in W \tag{17}$$

$$\sum_{q \in W} y2^{send}_{w,q,s} \leq z1_w \qquad \forall w \in W \boxed{s \in S} \tag{18}$$

- Non-negativity for all variables

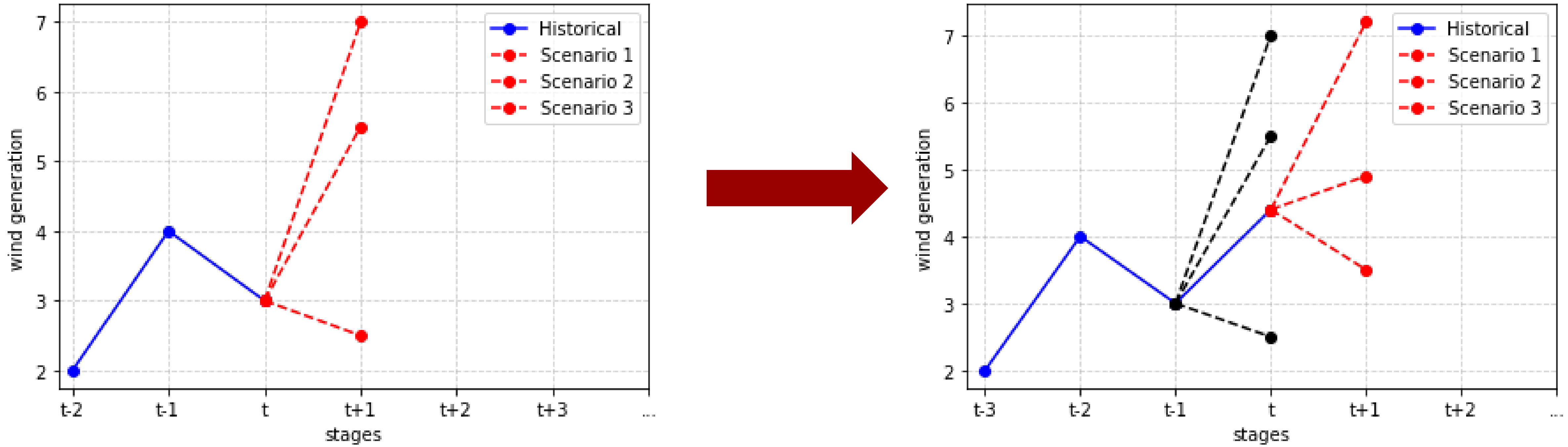
$$x1_w \geq 0, z1_w \geq 0, m1_w \geq 0 \qquad \forall w \in W$$

$$y1^{send}_{w,q} \geq 0, y1^{receive}_{w,q} \geq 0 \qquad \forall w, q \in W \tag{19}$$

$$x2_{w,s} \geq 0, z2_{w,s} \geq 0, m2_{w,s} \geq 0 \qquad \forall w \in W, s \in S$$

$$y2^{send}_{w,q,s} \geq 0, y2^{receive}_{w,q,s} \geq 0 \qquad \forall w, q \in W, s \in S \tag{20}$$

One Last Thing



this is the uncertainity, we create scenarios for the next stage, one ahead. anf we input, we do a decisoin to the outpu, we rreun to the inveriment, oit evaluates our decisons and gives a cost, the enxt realization, the new can be something th at is none of hour scenarios.
this is way, 2 stags, 1st two decision are here and now, anf then the others ae proepctive
take new state, genraate new scenario,....

Questions and Survey

input- takes stategives decision