Exploring your True North in Undergraduate Study

AIE 1901 - AI Exploration - LLM for Optimization

Jie Wang, 2025/12/02

Outline

- Part I: Course in Review
- Part II: Sharing my own journey
- Part III: Strategies for Your Journey

Outline

- Part I: Course in Review
- Part II: Sharing my own journey
- Part III: Strategies for Your Journey

General Procedures of optimization modeling

An Expected Training Example

Input - Natural Language Problem:
A company has three transportation options to choose from to transport 25 tons of cargo, namely trucks, airplanes, and ships with costs $\$100, \$120, \$130$ per ton and capacities of $10, 20, 30$ tons respectively. The company can't choose trucks and ships together. How should the company optimize the selection and allocation of these methods to minimize overall costs?

Target - Mathematical Model and Program:

Mathematical Model:

Variables:

- x_1, x_2, x_3: 0-1 variables indicating whether trucks, airplanes, and ships are selected, respectively.
- y_1, y_2, y_3: Non-negative continuous variables indicating the volume of cargo transported by three modes.

Objectives and Constraints:

$$\text{Minimize}_{x_i \in \{0,1\}, y_i \geq 0} \quad 100y_1 + 120y_2 + 80y_3$$

$$\text{subject to}$$

$$x_1 + x_2 + x_3 \geq 1 \qquad \text{(At least one transportation)}$$

$$y_1 \leq 10x_1, \quad y_2 \leq 20x_2, \quad y_3 \leq 30x_3 \qquad \text{(Capacity limits)}$$

$$x_1 + x_3 \leq 1 \qquad \text{(Mutual exclusion constraint)}$$

$$y_1 + y_2 + y_3 \geq 25 \qquad \text{(Transportation requirements)}$$

Program using COPT solver:

```python
import coptpy as cp
from coptpy import COPT
# Initialize the optimization environment
env = cp.Envr()
model = env.createModel("TransportationOptimization")
# Define costs and capacities for each mode of transportation
costs = {'trucks': 100, 'airplanes': 120, 'ships': 130}
capacities = {'trucks': 10, 'airplanes': 20, 'ships': 30}
# Define binary and continuous variables for each transportation mode
x = {mode: model.addVar(vtype=COPT.BINARY, name=f"x_{mode}") for mode in costs}
y = {mode: model.addVar(vtype=COPT.CONTINUOUS, lb=0, name=f"y_{mode}") for mode in costs}
# Set the objective function to minimize the total transportation cost
model.setObjective(sum(costs[mode] * y[mode] for mode in costs), sense=COPT.MINIMIZE)
# Add constraints
model.addConstr(x['trucks'] + x['airplanes'] + x['ships'] >= 1, name="AtLeastOneMode")
for mode in costs:
    model.addConstr(y[mode] <= capacities[mode] * x[mode], name=f"Capacity_{mode}")
model.addConstr(x['trucks'] + x['ships'] <= 1, name="ModeExclusivity")
model.addConstr(sum(y[mode] for mode in capacities) >= 25, name="Volume Requirement")
# Solve the model
model.solve()
# Check the solution status and print the optimal values of the variables
if model.status == COPT.OPTIMAL:
    print("Optimal solution found:")
    for mode in costs:
        print(f"{mode}: x = {x[mode].x}, y = {y[mode].x}")
```

General Procedures of optimization modeling

1. take a problem (p),
2. find the right mathematical model (m),
3. apply the solver (c).

The goal of **ORLM** is to learn how to go from a problem to its correct model and solution method.

Newsvendor Problem (Homework 2 & 4)

$$g(D, y) = \underbrace{p \cdot \min(D, y)}_{\text{Revenue}} - \underbrace{c_v \cdot y}_{\text{Ordering Cost}} - \underbrace{h \cdot \max(y - D, 0)}_{\text{Holding Cost}} - \underbrace{b \cdot \max(D - y, 0)}_{\text{Backorder Cost}}$$

$$\text{Objective} = \mathbb{E}_D[g(D, y)]$$

Newsvendor Problem (Homework 2 & 4)

$$g(D, y) = p \cdot \min(D, y) - c_v \cdot y - h \cdot \max(y - D, 0) - b \cdot \max(D - y, 0)$$

Revenue

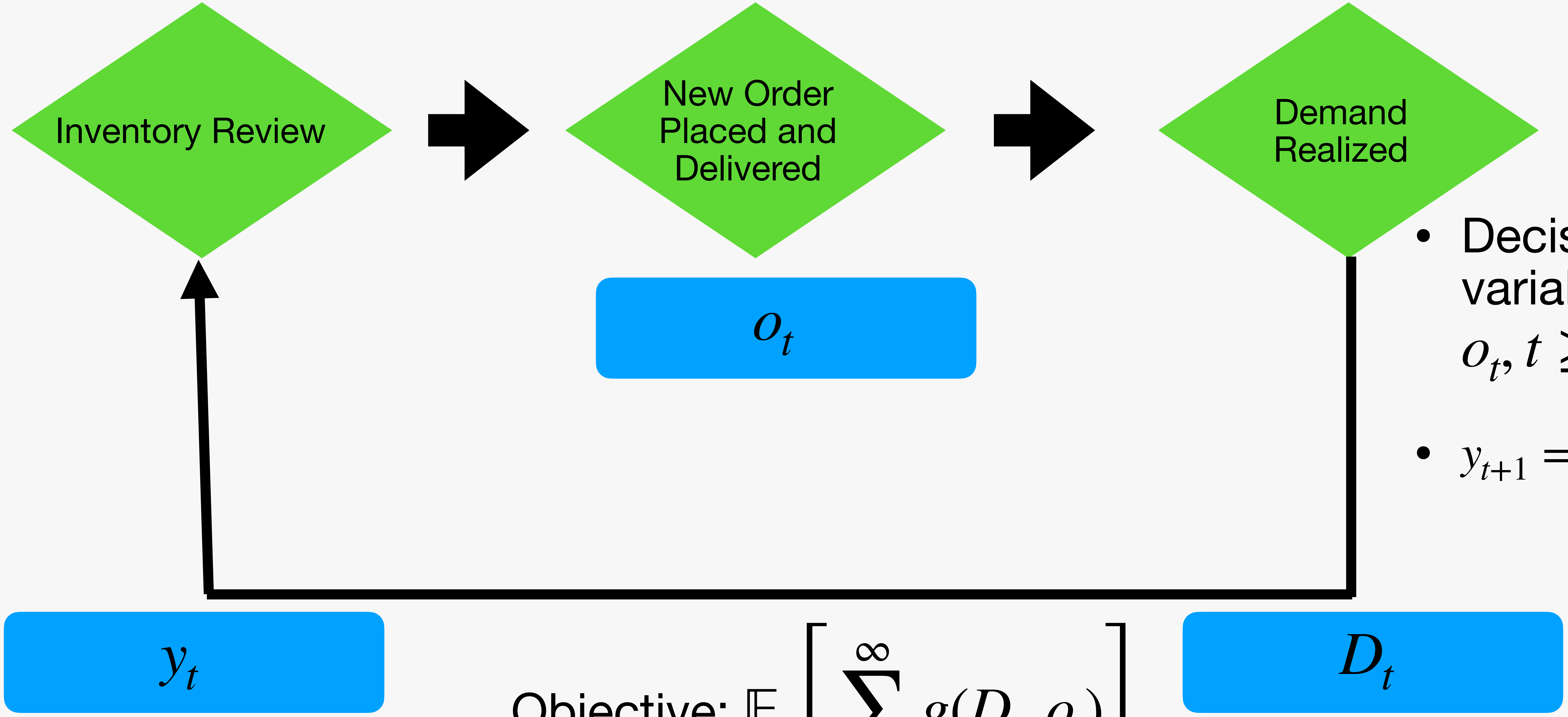
Ordering Cost

Holding Cost

Backorder Cost

Objective $= \mathbb{E}_D[g(D, y)]$

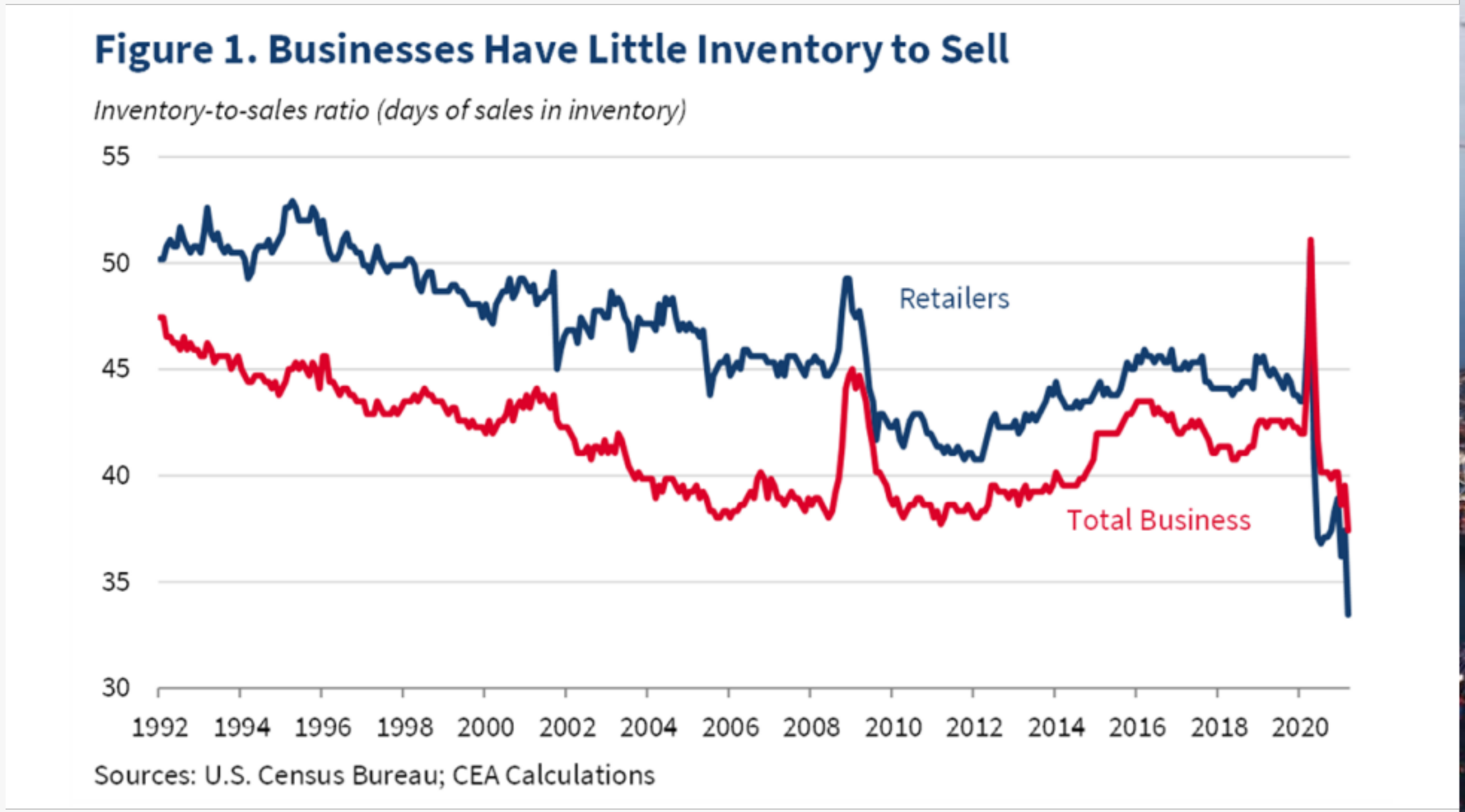
Newsvendor Problem in Dynamic Setting



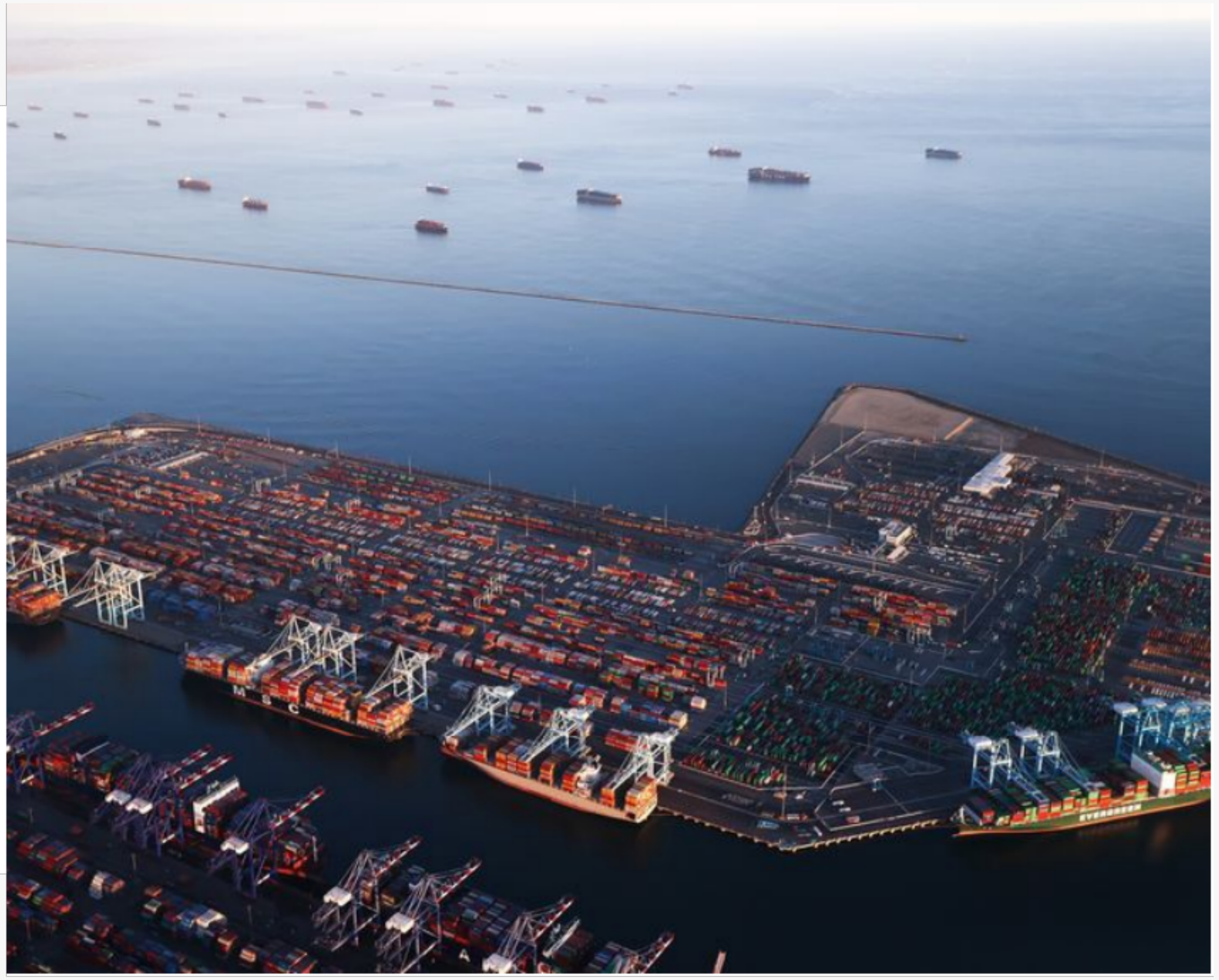
- Decision variable: $o_t, t \geq 1$
- $y_{t+1} = y_t + o_t - D_t$

Objective: $\mathbb{E}\left[\sum_{t=0}^{\infty} g(D_t, o_t)\right]$

Pandemic Has Disrupted Supply Chains



https://www.whitehouse.gov/cea/blog/2021/06/17/why-the-pandemic-has-disrupted-supply-chains/

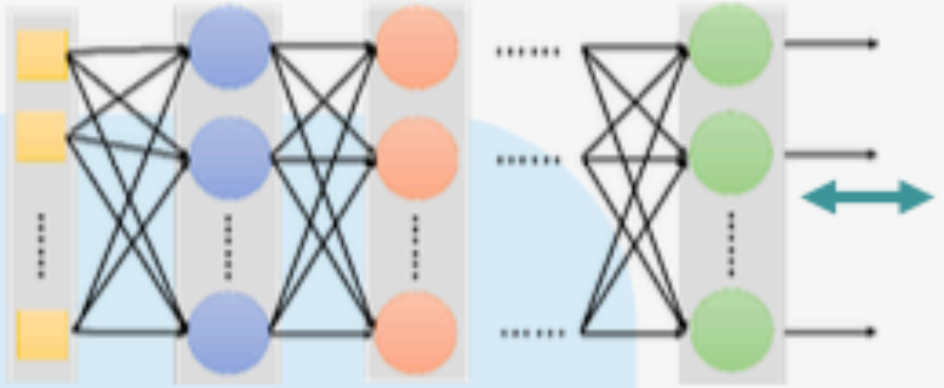


Bloomberg, 8/28/2021, U.S. Port Problems Reach Worst of Pandemic Amid Crush of Imports

Inventory dynamics with lead time

- On-hand inventory I_t
- Pipeline vector $\boldsymbol{x}_t = (x_{0,t}, x_{1,t}, \ldots, x_{L-1,t})$
  - Orders already placed but not yet received
- Decision variables o_t: new order placed

Linear Regression and Variable Selection (Homework 3)

1) Given a function, $\text{Cat or Dog} = f($  $) =$ 

↔ Neural network

2) 

cat *dog* *cat* *dog*

X

Y

↔ Training data

3) Determine the parameters: w and b

↔ Learning/ training process

4) Based on the determined parameters, please predict the y when x = 

↔ Prediction/ inference

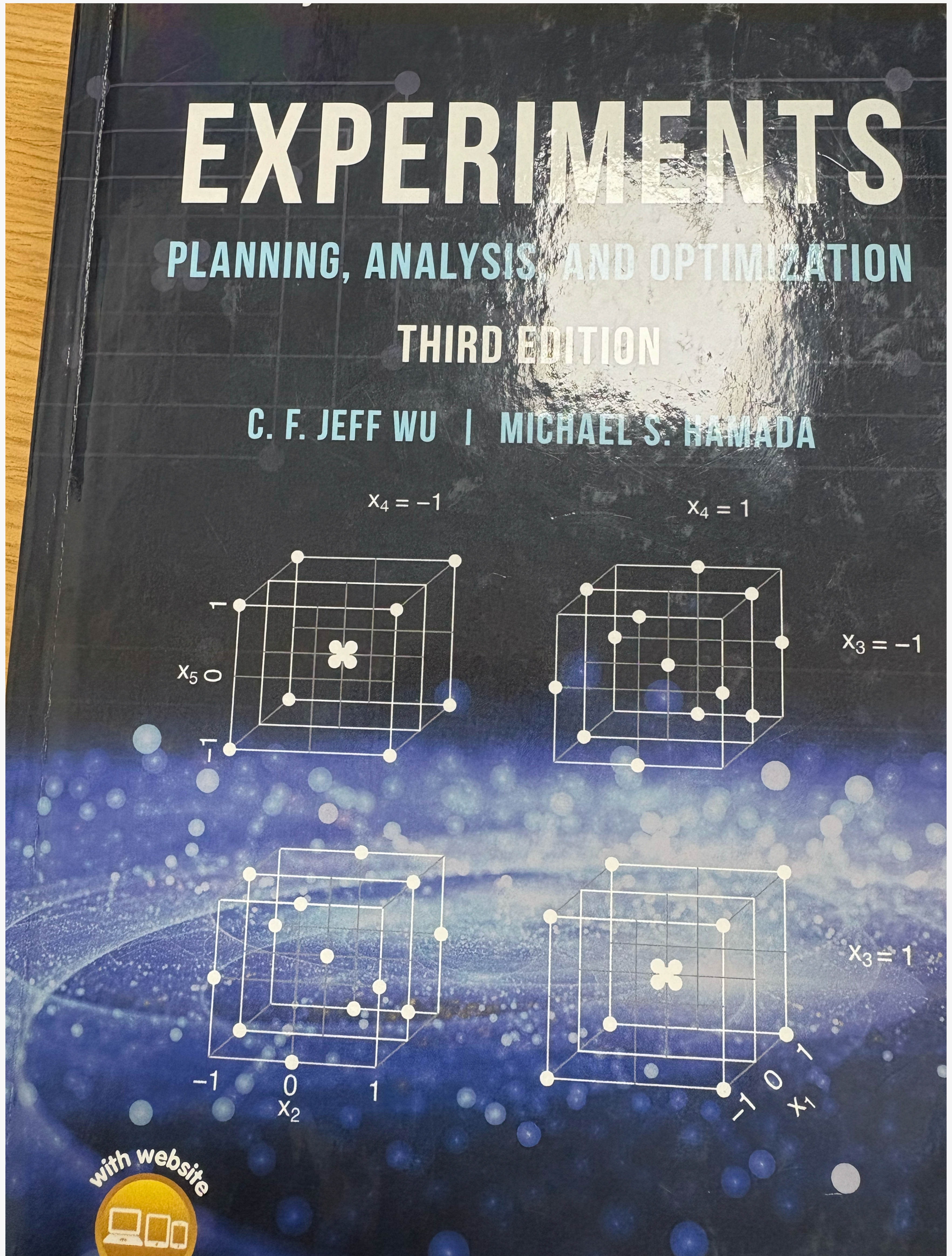
$$\min_{\beta} \ \|y - X\beta\|_2^2$$

Linear Regression and Variable Selection

Table 3.52 Corrosion Resistances of Steel Bars, Steel Experiment

Whole Plot	Temperature (T)	Coatings (C)	Replication (Rep)	Resistance
1	1	2	1	73
1	1	3	1	83
1	1	1	1	67
1	1	4	1	89
2	2	1	1	65
2	2	3	1	87
2	2	4	1	86
2	2	2	1	91
3	3	3	1	147
3	3	1	1	155
3	3	2	1	127
3	3	4	1	212
4	1	4	2	153
4	1	3	2	90
4	1	2	2	100
4	1	1	2	108
5	2	4	2	150
5	2	1	2	140
5	2	3	2	121
5	2	2	2	142
6	3	1	2	33
6	3	4	2	54
6	3	2	2	8
6	3	3	2	46

39. The data (Roth, 1992) in Table 3.53 were obtained from a digital thickness gauge for measuring the wall thickness (in inches) of an unidentified part. Ten parts were selected randomly from a production lot. Three operators were selected from those who normally use the gauge. Each of the three operators measured the 10 parts that were randomly ordered. The 10 parts were again randomly ordered and each of the three operators again measured the



When should we use AI?

THE CHINESE UNIVERSITY OF HONG KONG, SHENZHEN

Use of Artificial Intelligence Tools in Teaching, Learning and Assessments

A Guide for Students

The Risk (Short-Term)	**The Cost** (Long-Term)
Academic integrity violation.	**Erosion of Core Skills:** Problem-solving, critical thinking, and depth of understanding atrophy without practice.
Loss of instructor trust.	**The "Black Box" Dependency:** You risk not knowing what you don't know, creating fragile knowledge.
A grade penalty.	**Missed "Struggle" Benefits:** The cognitive wrestling is where true mastery and insight are forged.

When should we use AI?

✅ DO: Use AI for…

- Explaining concepts in different analogies
- Finding bugs in code you've written.
- Brainstorming research angles or project ideas

小猿搜题
中小学拍照搜题利器

❌ DON'T: Use AI to…

- Generate final, uncritiqued answers for submission but do not understand it at all
- Complete core tasks meant to assess your skill

Outline

- Part I: Course in Review
- Part II: Sharing my own journey
- Part III: Strategies for Your Journey

Can you guess what top percentage I graduated in during my UG?

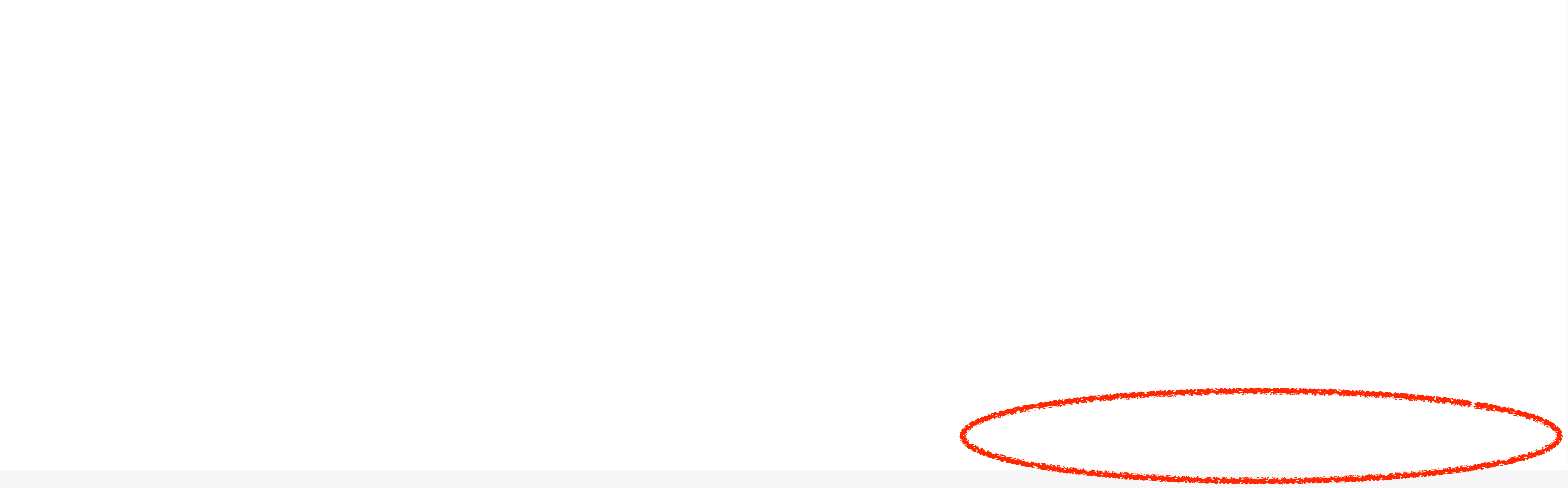
My UG Academic Performance in Freshman

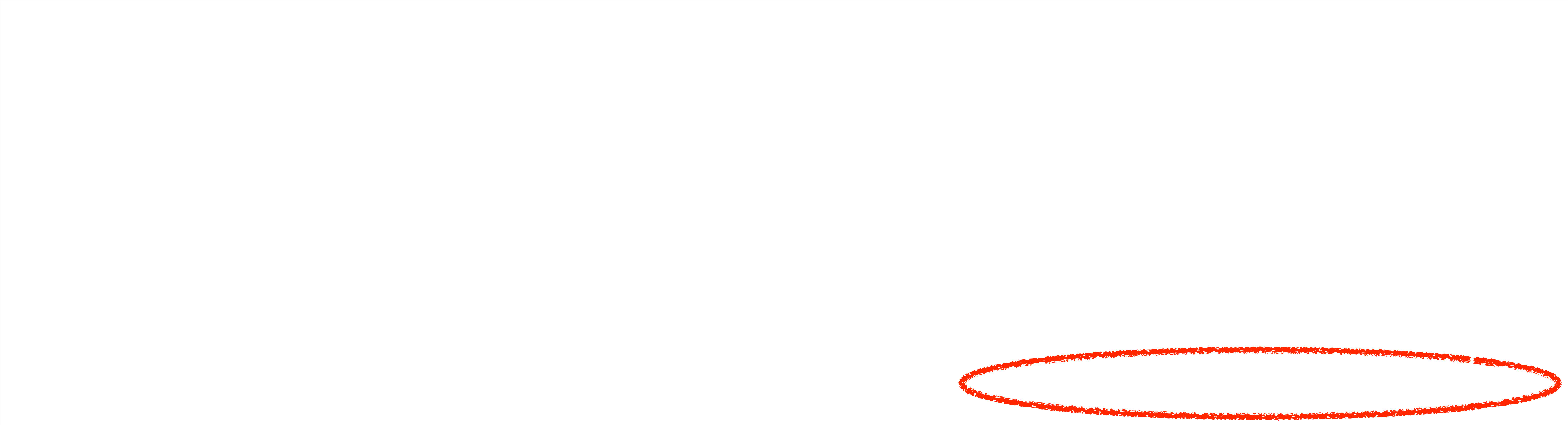
2016-17 Term 1

My UG Academic Performance in Second Year

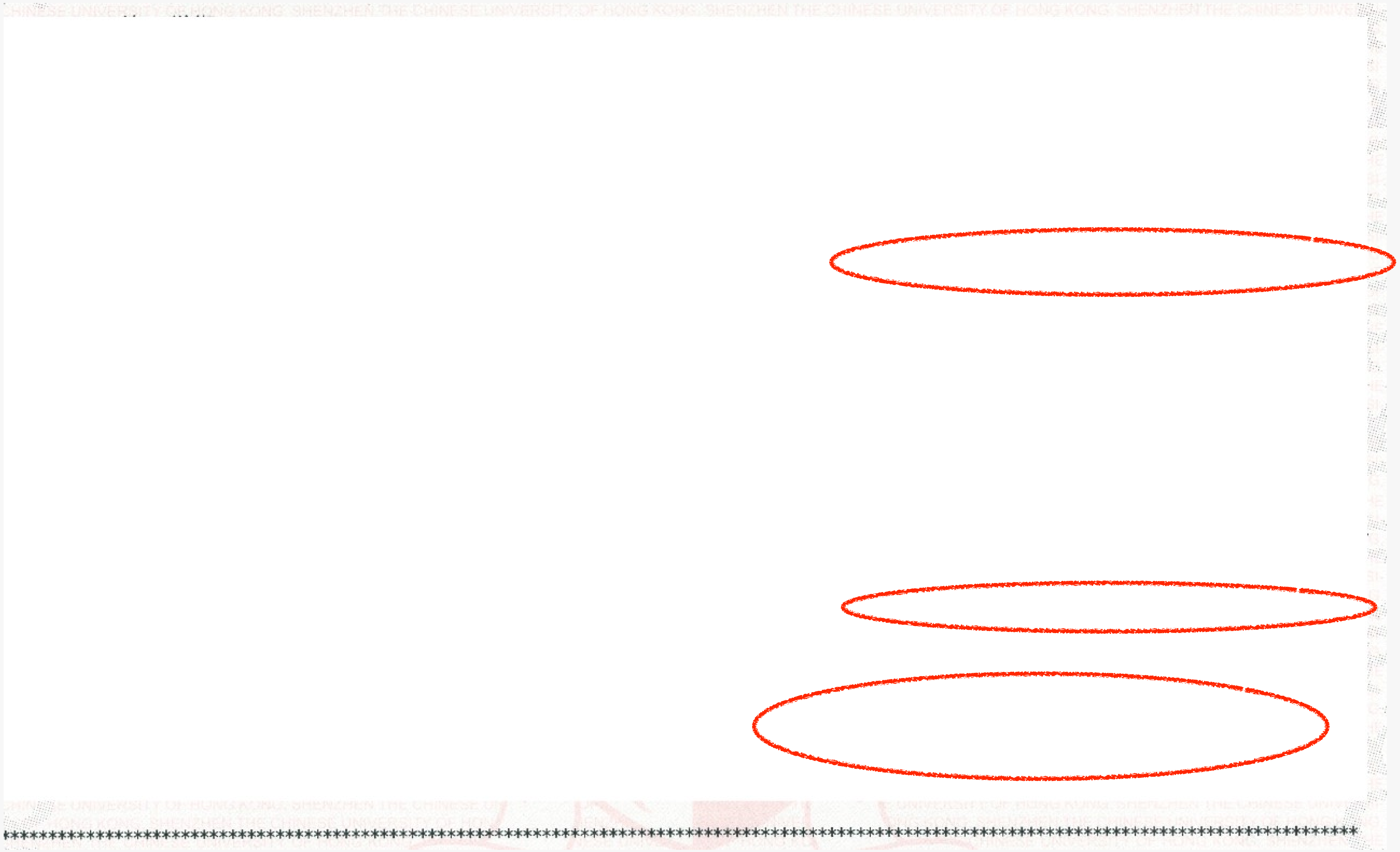
2017-18 Term 1

My UG Academic Performance in Third Year





My UG Academic Performance in Fourth Year



Besides GPA, What do I explore?

https://www.cuhk.edu.cn/en/article/3413

Sophomore's Paper Accepted by IEEE International Symposium on Information Theory 2018

April 17, 2018 News

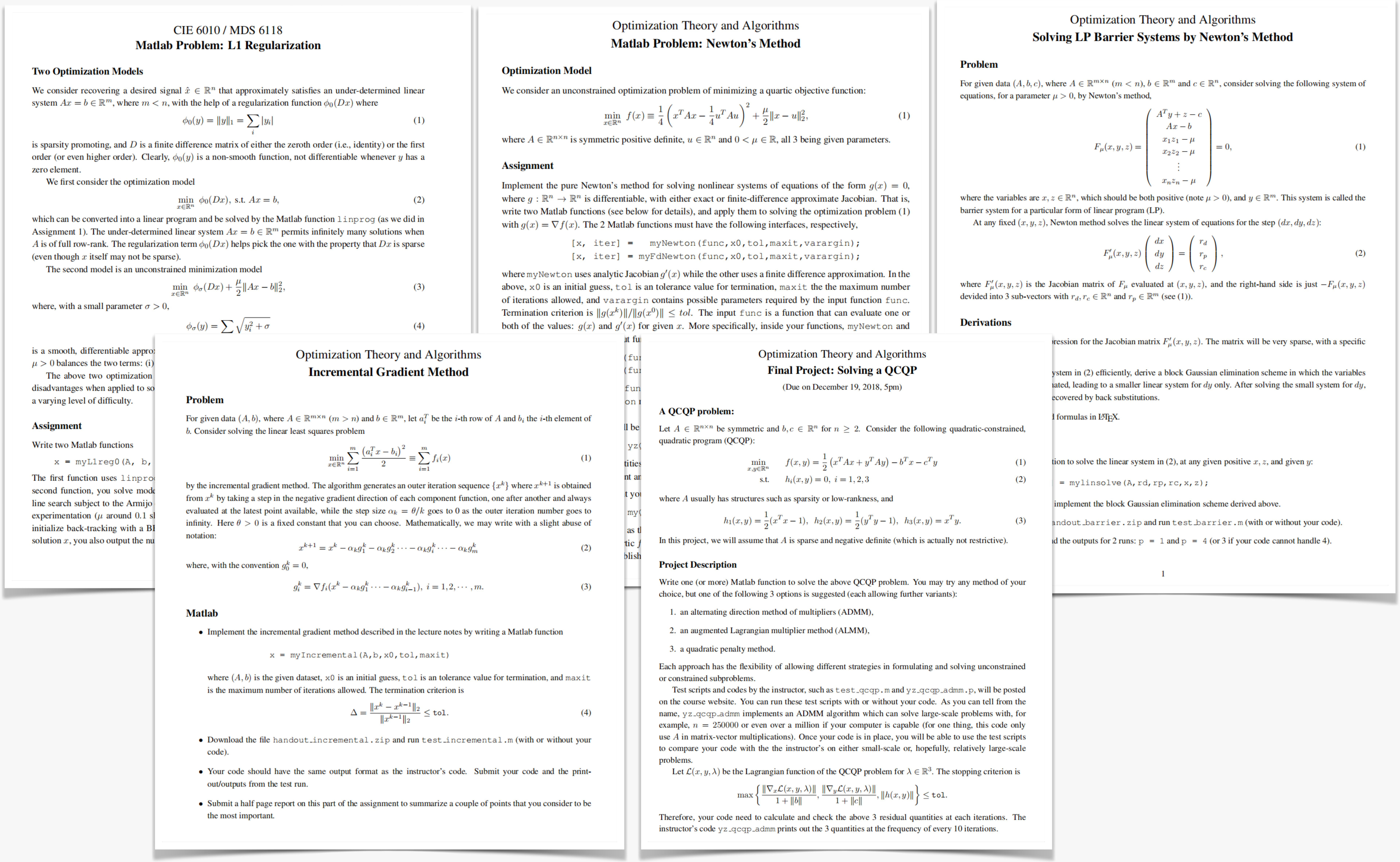
The academic paper of Wang Jie, a sophomore majoring in Pure Mathematics, and Yang Shenghao, an assistant professor of School of Science and Engineering, was accepted by the International Symposium on Information Theory 2018 (ISIT 2018). The paper proves that the upper bounds of the computational capabilities proposed by the former studies are not always compact by studying the computational capabilities of a network. Wang Jie, the first author of this paper, graduated from Shenzhen Hongling Middle School.

Besides GPA, What do I explore?

https://www.bilibili.com/video/BV1EZ4y1377j/

- Claude Shannon worked with ease, and played with seriousness—and never saw a difference between the two.
- He happily followed his curiosity into whatever grabbed his attention.
- He transformed the dry technicalities of science into masses of compelling puzzles

I also discovered my career interest: Operations Research



CIE 6010 / MDS 6118
Matlab Problem: L1 Regularization

Two Optimization Models

We consider recovering a desired signal $\hat{x} \in \mathbb{R}^n$ that approximately satisfies an under-determined linear system $Ax = b \in \mathbb{R}^m$, where $m < n$, with the help of a regularization function $\phi_0(Dx)$ where

$$\phi_0(y) = \|y\|_1 = \sum_i |y_i| \quad (1)$$

is sparsity promoting, and D is a finite difference matrix of either the zeroth order (i.e., identity) or the first order (or even higher order). Clearly, $\phi_0(y)$ is a non-smooth function, not differentiable whenever y has a zero element.

We first consider the optimization model

$$\min_{x \in \mathbb{R}^n} \phi_0(Dx), \text{ s.t. } Ax = b, \quad (2)$$

which can be converted into a linear program and be solved by the Matlab function `linprog` (as we did in Assignment 1). The under-determined linear system $Ax = b \in \mathbb{R}^m$ permits infinitely many solutions when A is of full row-rank. The regularization term $\phi_0(Dx)$ helps pick the one with the property that Dx is sparse (even though x itself may not be sparse).

The second model is an unconstrained minimization model

$$\min_{x \in \mathbb{R}^n} \phi_\sigma(Dx) + \frac{\mu}{2}\|Ax - b\|_2^2, \quad (3)$$

where, with a small parameter $\sigma > 0$,

$$\phi_\sigma(y) = \sum \sqrt{y_i^2 + \sigma} \quad (4)$$

is a smooth, differentiable appro
$\mu > 0$ balances the two terms: (i)
The above two optimization
disadvantages when applied to so
a varying level of difficulty.

Assignment

Write two Matlab functions

```
x = myL1reg0(A, b,
```

The first function uses `linpro`
second function, you solve mode
line search subject to the Armijo
experimentation (μ around 0.1 sl
initialize back-tracking with a BI
solution x, you also output the nu

Optimization Theory and Algorithms
Matlab Problem: Newton's Method

Optimization Model

We consider an unconstrained optimization problem of minimizing a quartic objective function:

$$\min_{x \in \mathbb{R}^n} f(x) \equiv \frac{1}{4}\left(x^T Ax - \frac{1}{4}u^T Au\right)^2 + \frac{\mu}{2}\|x - u\|_2^2, \quad (1)$$

where $A \in \mathbb{R}^{n \times n}$ is symmetric positive definite, $u \in \mathbb{R}^n$ and $0 < \mu \in \mathbb{R}$, all 3 being given parameters.

Assignment

Implement the pure Newton's method for solving nonlinear systems of equations of the form $g(x) = 0$, where $g : \mathbb{R}^n \to \mathbb{R}^n$ is differentiable, with either exact or finite-difference approximate Jacobian. That is, write two Matlab functions (see below for details), and apply them to solving the optimization problem (1) with $g(x) = \nabla f(x)$. The 2 Matlab functions must have the following interfaces, respectively,

```
[x, iter] =   myNewton(func,x0,tol,maxit,varargin);
[x, iter] = myFdNewton(func,x0,tol,maxit,varargin);
```

where `myNewton` uses analytic Jacobian $g'(x)$ while the other uses a finite difference approximation. In the above, `x0` is an initial guess, `tol` is an tolerance value for termination, `maxit` the the maximum number of iterations allowed, and `varargin` contains possible parameters required by the input function `func`. Termination criterion is $\|g(x^k)\|/\|g(x^0)\| \le tol$. The input `func` is a function that can evaluate one or both of the values: $g(x)$ and $g'(x)$ for given x. More specifically, inside your functions, `myNewton` and

Optimization Theory and Algorithms
Solving LP Barrier Systems by Newton's Method

Problem

For given data (A, b, c), where $A \in \mathbb{R}^{m \times n}$ $(m < n)$, $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$, consider solving the following system of equations, for a parameter $\mu > 0$, by Newton's method,

$$F_\mu(x, y, z) = \begin{pmatrix} A^T y + z - c \\ Ax - b \\ x_1 z_1 - \mu \\ x_2 z_2 - \mu \\ \vdots \\ x_n z_n - \mu \end{pmatrix} = 0, \quad (1)$$

where the variables are $x, z \in \mathbb{R}^n$, which should be both positive (note $\mu > 0$), and $y \in \mathbb{R}^m$. This system is called the barrier system for a particular form of linear program (LP).

At any fixed (x, y, z), Newton method solves the linear system of equations for the step (dx, dy, dz):

$$F'_\mu(x, y, z)\begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} = \begin{pmatrix} r_d \\ r_p \\ r_c \end{pmatrix}, \quad (2)$$

where $F'_\mu(x, y, z)$ is the Jacobian matrix of F_μ evaluated at (x, y, z), and the right-hand side is just $-F_\mu(x, y, z)$ devided into 3 sub-vectors with $r_d, r_c \in \mathbb{R}^n$ and $r_p \in \mathbb{R}^m$ (see (1)).

Derivations

ression for the Jacobian matrix $F'_\mu(x, y, z)$. The matrix will be very sparse, with a specific

ystem in (2) efficiently, derive a block Gaussian elimination scheme in which the variables
ated, leading to a smaller linear system for dy only. After solving the small system for dy,
ecovered by back substitutions.

l formulas in LaTeX.

tion to solve the linear system in (2), at any given positive x, z, and given y:

```
= mylinsolve(A,rd,rp,rc,x,z);
```

implement the block Gaussian elimination scheme derived above.

andout_barrier.zip and run test_barrier.m (with or without your code).

d the outputs for 2 runs: p = 1 and p = 4 (or 3 if your code cannot handle 4).

1

Optimization Theory and Algorithms
Incremental Gradient Method

Problem

For given data (A, b), where $A \in \mathbb{R}^{m \times n}$ $(m > n)$ and $b \in \mathbb{R}^m$, let a_i^T be the i-th row of A and b_i the i-th element of b. Consider solving the linear least squares problem

$$\min_{x \in \mathbb{R}^n} \sum_{i=1}^m \frac{(a_i^T x - b_i)^2}{2} \equiv \sum_{i=1}^m f_i(x) \quad (1)$$

by the incremental gradient method. The algorithm generates an outer iteration sequence $\{x^k\}$ where x^{k+1} is obtained from x^k by taking a step in the negative gradient direction of each component function, one after another and always evaluated at the latest point available, while the step size $\alpha_k = \theta/k$ goes to 0 as the outer iteration number goes to infinity. Here $\theta > 0$ is a fixed constant that you can choose. Mathematically, we may write with a slight abuse of notation:

$$x^{k+1} = x^k - \alpha_k g_1^k - \alpha_k g_2^k \cdots - \alpha_k g_i^k \cdots - \alpha_k g_m^k \quad (2)$$

where, with the convention $g_0^k = 0$,

$$g_i^k = \nabla f_i(x^k - \alpha_k g_1^k \cdots - \alpha_k g_{i-1}^k), \ i = 1, 2, \cdots, m. \quad (3)$$

Matlab

- Implement the incremental gradient method described in the lecture notes by writing a Matlab function

```
x = myIncremental(A,b,x0,tol,maxit)
```

  where (A, b) is the given dataset, `x0` is an initial guess, `tol` is an tolerance value for termination, and `maxit` is the maximum number of iterations allowed. The termination criterion is

$$\Delta = \frac{\|x^k - x^{k-1}\|_2}{\|x^{k-1}\|_2} \le \texttt{tol}. \quad (4)$$

- Download the file `handout_incremental.zip` and run `test_incremental.m` (with or without your code).
- Your code should have the same output format as the instructor's code. Submit your code and the print-out/outputs from the test run.
- Submit a half page report on this part of the assignment to summarize a couple of points that you consider to be the most important.

Optimization Theory and Algorithms
Final Project: Solving a QCQP
(Due on December 19, 2018, 5pm)

A QCQP problem:

Let $A \in \mathbb{R}^{n \times n}$ be symmetric and $b, c \in \mathbb{R}^n$ for $n \ge 2$. Consider the following quadratic-constrained, quadratic program (QCQP):

$$\min_{x,y \in \mathbb{R}^n} \ f(x, y) = \frac{1}{2}(x^T Ax + y^T Ay) - b^T x - c^T y \quad (1)$$
$$\text{s.t.} \ h_i(x, y) = 0, \ i = 1, 2, 3 \quad (2)$$

where A usually has structures such as sparsity or low-rankness, and

$$h_1(x, y) = \frac{1}{2}(x^T x - 1), \ h_2(x, y) = \frac{1}{2}(y^T y - 1), \ h_3(x, y) = x^T y. \quad (3)$$

In this project, we will assume that A is sparse and negative definite (which is actually not restrictive).

Project Description

Write one (or more) Matlab function to solve the above QCQP problem. You may try any method of your choice, but one of the following 3 options is suggested (each allowing further variants):

1. an alternating direction method of multipliers (ADMM),
2. an augmented Lagrangian multiplier method (ALMM),
3. a quadratic penalty method.

Each approach has the flexibility of allowing different strategies in formulating and solving unconstrained or constrained subproblems.

Test scripts and codes by the instructor, such as `test_qcqp.m` and `yz_qcqp_admm.p`, will be posted on the course website. You can run these test scripts with or without your code. As you can tell from the name, `yz_qcqp_admm` implements an ADMM algorithm which can solve large-scale problems with, for example, $n = 250000$ or even over a million if your computer is capable (for one thing, this code only use A in matrix-vector multiplications). Once your code is in place, you will be able to use the test scripts to compare your code with the the instructor's on either small-scale or, hopefully, relatively large-scale problems.

Let $\mathcal{L}(x, y, \lambda)$ be the Lagrangian function of the QCQP problem for $\lambda \in \mathbb{R}^3$. The stopping criterion is

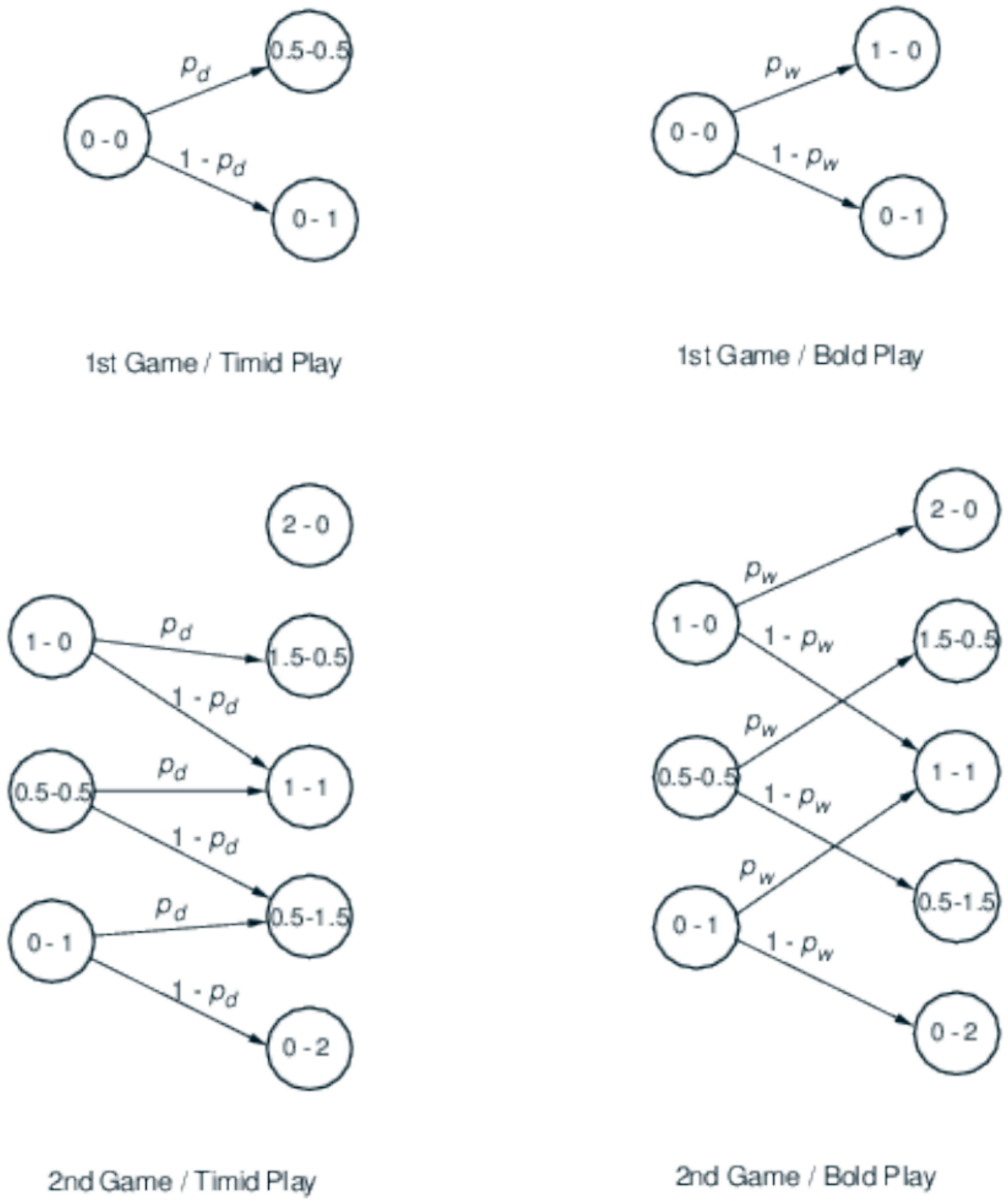
$$\max\left\{\frac{\|\nabla_x \mathcal{L}(x, y, \lambda)\|}{1 + \|b\|}, \frac{\|\nabla_y \mathcal{L}(x, y, \lambda)\|}{1 + \|c\|}, \|h(x, y)\|\right\} \le \texttt{tol}.$$

Therefore, your code need to calculate and check the above 3 residual quantities at each iterations. The instructor's code `yz_qcqp_admm` prints out the 3 quantities at the frequency of every 10 iterations.

I also discovered my career interest: Operations Research

Find two-game chess match strategy.

- *Timid* play draws with prob. $p_d > 0$ and loses with prob. $1 - p_d$.
- *Bold* play wins with prob. $p_w < 1/2$ and loses with prob. $1 - p_w$.
- Sudden-death rematch if a tie at the end of two games.



Outline

- Part I: Course in Review
- Part II: Sharing my own journey
- Part III: Strategies for Your Journey

Main Takeaways

- GPA is a data point, **not a destiny**. It measures a specific type of performance under specific conditions, but it is an ***incomplete picture*** of your abilities, potential, and intelligence.
- **Learning is more than the pursuit of the grades (求学不是求分数)**
- It is important to maintain a high GPA. It roughly measures how well you learn for the course. Matters for first jobs, graduate school applications, scholarships, etc.
- But do not blame yourself if you make mistakes, or if you really try hard but not earn a perfect GPA.
- Don’t get stressed by peer pressure.

Which components matter more than GPA?

- **Knowledge & Skills:** Can you apply what you have learned?
- **Experience:** Internships, research, part-time jobs, course projects. This is where theory meets practice.
- **Network & Relationships:** Connections with professors, mentors, and peers. These lead to recommendations, collaborations, and opportunities.
- **Self-Awareness & Resilience:** Knowing your strengths, interests, and how you handle setbacks. This is the pillar of long-term career satisfaction.

Which components will not be replaced by AI?

- Subjective Experience, Emotional Depth
- Creativity and Breakthrough Inspiration
- Ethical Decision-Making
- Curiosity, Passion, and the Meaning of Life
- Leadership, mentorship, collaboration

To cultivate these skills, what is the general procedure?

- **Part I: Discover Yourself**
    - When you are in trouble and all your defenses get stripped away, you realize what matters and who matters. That’s when you need to get back to your roots and to your values.
- **Part II: Develop Yourself**
    - Follow your compass and not your clock.
- **Part III: Lead People and Navigate Today’s Challenges**