ISE 533

Integrative Analytics

Project Portfolio

Group 6

Qianran Ma (7798937244)
Wanning Li (5898212888)
Yanan Zhou (4366462176)
Yuxuan Cheng (1278632602)

Overview

In this semester, our group worked on 4 practical integrative analytics projects. In each project, we learned how to transform real-life problems into mathematical problems, how to solve problems with uncertainty, how to design a program with creative ideas, and how to deliver our work to the audience. This portfolio will introduce the reports and code of all of the 4 projects we did.

Project 1

Multi-Location Transshipment Problem

Content

1. Introduction

Shopping is always an important part of people's daily life. However, it sometimes makes them frustrated, since people may go to a store and see clothes they really like, but only to find it do not have their sizes. To address this kind of dissatisfaction, retail stores try to make up the shortage by back-order or transshipment from other stores. This is definitely a happy solution for customers, but it may generate higher cost for the retailers.

Therefore, it is really important for retailers to make a best decision at the beginning on how much goods they would like to order for each store. If they can make an appropriate number of orders that just satisfy the customers' demand, then there will be minimal additional cost. However, they have no way to know the customer demand in advance. The only thing they can do is to collect the history sales data and estimate the possible demand of each store.

The chapter one of this portfolio focuses on a multi-location transshipment model. The chapter first introduces the mathematical optimization model of the multi-location transshipment problem, then uses three methods to solve the model, and finally shows some creative ideas from our group members regarding the improvements of the model and Stochastic Approximation method, one of the three solving methods.

2. Multi-Location Transshipment Model

In this section, we first build the transshipment optimization model and clarify it. Since the demands of each retailer are not constant values, we assume it follows normal distributions with different means and variances. We then use three methods, which are All-in-One, Stochastic Approximation, and Stochastic Decomposition, to solve the model with stochastic elements.

2.1 The Mathematical Model

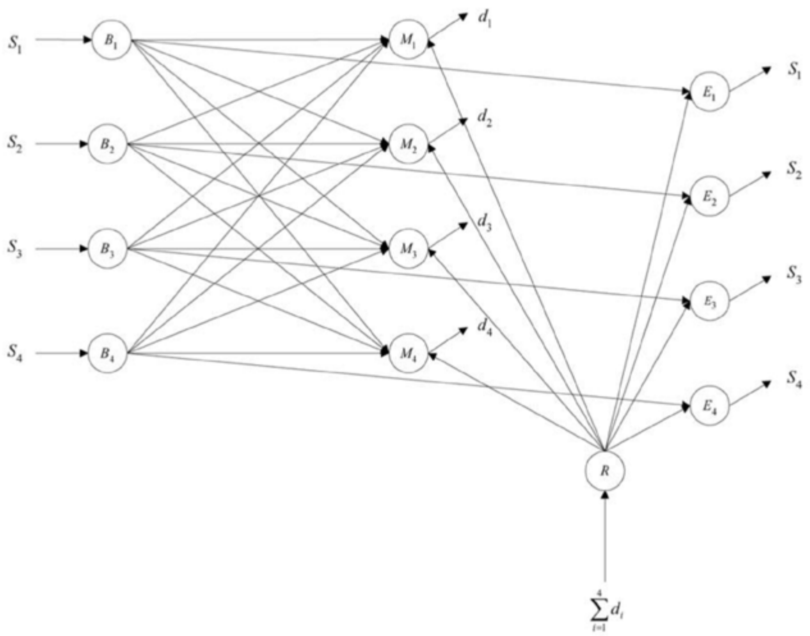


Figure 1 Network flow representation of a single period

The picture above shows the whole process of a single period. At the beginning, the store will order up to a certain level S, which can be considered as the first stage of transshipment problem. During the selling process, we make f denote the stock for demand. When the stock is short for demand, there are two ways to make up for the shortage, transshipment denoted by t, and backorder denoted by r. At the end of this single period, the store will have and ending inventory, denoted by e, and inventory increased through replenishment, denoted by q. The target of the transshipment problem is to minimize the expected long-run average cost. The model of the second stage's transshipment problem is as follows.

$$h(S,D) = \min \sum_i h_i e_i + \sum_{i \neq j} c_{ij} t_{ij} + \sum_i p_i r_i. \tag{1}$$

$$f_i + \sum_{j \neq i} t_{ij} + e_i = s_i, \quad \forall i \tag{1a}$$

$$f_i + \sum_{j \neq i} t_{ji} + r_i = d_i, \quad \forall i \tag{1b}$$

$$\sum_i r_i + \sum_i q_i = \sum_i d_i \tag{1c}$$

$$e_i + q_i = s_i, \quad \forall i \tag{1d}$$

$$e_i, f_i, q_i, r_i, s_i, t_{ij} \geq 0, \quad \forall i, j.$$

The notation we used is listed below.

s_i	Order-up-to quantities of retailer i.
d_i	Demand of retailer i.
h_i	Unit cost of holding inventory at retail i.
c_{ij}	Unit cost of transshipment from retailer i to j.
p_i	Penalty cost for shortage at retailer i.
z_i	Ending inventory held at retail i.
f_i	Stock at retailer i used to satisfy demand at retailer i.
q_i	Inventory at retailer i increased through replenishment
r_i	Amount of shortage met after replenishment at retailer i.
t_{ij}	Stock at retailer i used to meet demand at retailer j, using the transshipment option.

2.2 All-In-One Model

All-in-one model combines several scenarios into one function to simulate the real world situation.

$$\min c^T x + \frac{1}{|s|}\sum_s g_s^T y_s \tag{2}$$

$$\text{s.t.} \quad Ax = b \tag{2a}$$

$$T_s x + W_s y = r_s \tag{2b}$$

$$x \geq 0, y_s \geq 0$$

We assume that all the scenarios are uniformly distributed, which means each scenario has the same probability to appear. Therefore, each scenario has the same probability in the function. We pick up 3 possible values of each store's demand and combine them together to get 2187 scenarios, and assign each scenario the same probability of 1/2187 to calculate the optimal solution of supply.

	Objective Value	S(i)	Running Time
All-In-One	100.8126	[105, 208, 158, 180, 188, 173, 180]	6.3 minutes

Table 1 Results of All-In-One Model

The final objective value using All-In-One model is 100.8126, and the supply levels of each retailer are [105, 208, 158, 180, 188, 173, 180]. The running time is 6 minutes and 20 seconds.

Given the supply levels, we input different demands into the optimization function, repeat 1000 times, to get the distribution of the total transshipment cost. The mean is 458.04. The lower bound of 95% confidence interval is 446.332, the upper bound is 469.9696.

	Mean	95% Confidence Interval
All-In-One	458.04	[446.332, 469.9696]

Table 2 Evaluation for All-In-One method

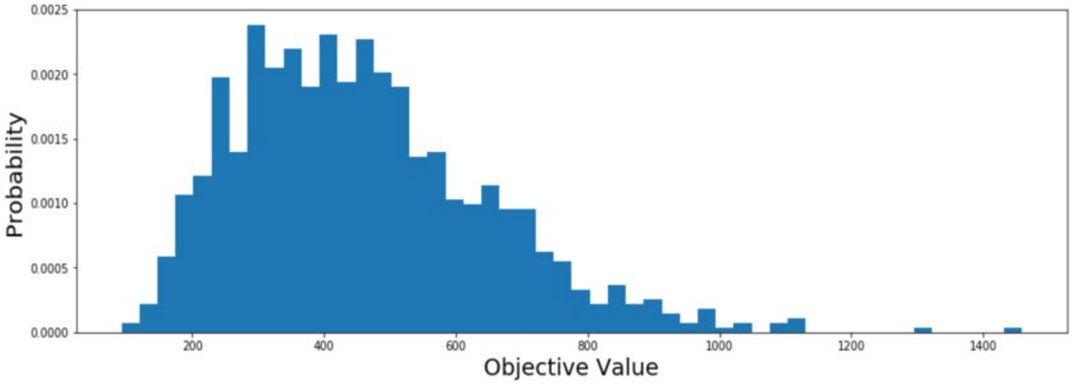


Figure 2 Distribution of the Evaluation for All-In-One method

2.3 Stochastic Approximation Method

Stochastic approximation is also known as SGD (Stochastic Gradient Descent). The transshipment model is not smooth, so technically we cannot calculate the gradient. Therefore, we use dual problem to work out the sub-gradient to represent the "gradient" in SA method. The main idea is to first generate demands with normal distribution. Second, given S and demands, we repeatedly solve optimization dual problem to get the results, where we calculate the average results to represent sub-gradient. Finally, based on the sub-gradient, we can update order-up-to S.

	Objective Value	S(i)	Running Time
SA	133.2476	[106, 206, 156, 176, 186, 176, 176]	31 minutes

Table 3 Results of SA method

We set K=100 (the maximal iteration number of gradient descent), U=500 (the iteration number to calculate average subgradient), size_c=10 (step size for gradient descent) to get the result. The mean of the final objective value is 133.2476. The supplies for seven retailers are [106, 206, 156, 176, 186, 176, 176]. It takes 31 minutes to finish.

During the evaluation part, the mean of the SA method is 193.04. And the lower bound of 95% confidence interval is 183.7315, the upper bound is 203.0226.

	Mean	95% Confidence Interval
SA	193.04	[183.7315, 203.0226]

Table 4 Evaluation for SA method

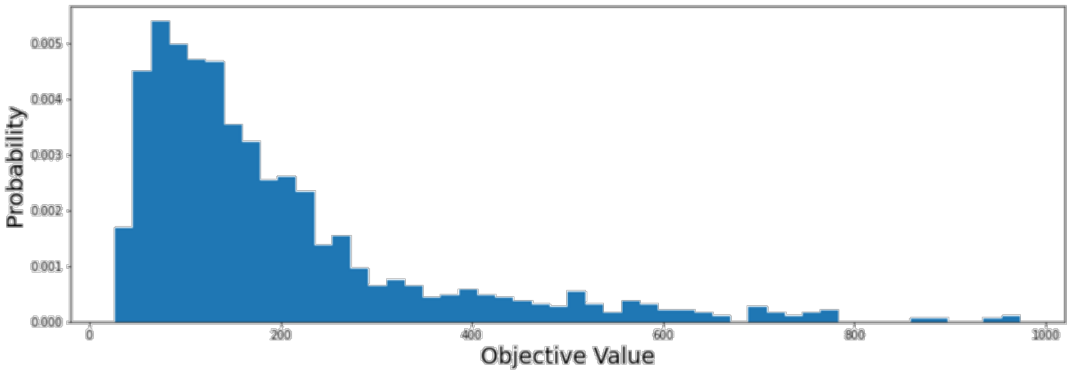


Figure 3 Distribution of the Evaluation for SA method

2.4 Stochastic Decomposition

In general, the idea of SD method is to combine many "cuts" together to compose the shape of the transshipment model approximately.

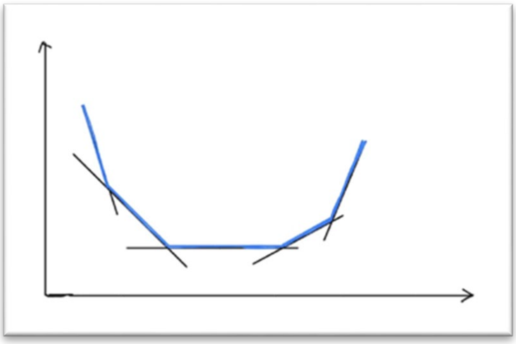


Figure 4 SD Method

The master program is as follows:

$$\begin{aligned} & \min f_k(x) \\ s.t. \quad & Ax \leq b \end{aligned} \qquad (M^k)$$

We define x^{k+1} as a candidate solution, which is the solution to M^k, and $\bar{x}^k$ as incumbent solution, a solution that is potentially the best point generated by the algorithm during the first k iteration. Next, we can update x according to the following condition:

$$x^{k+1} \text{ better than } \bar{x}^k : \quad \bar{x}^{k+1} = x^{k+1}$$
$$\text{Otherwise}: \quad \bar{x}^{k+1} = \bar{x}^k$$

The final lower bound estimate is 122. The supplies for seven retailers are [93, 197, 145, 164, 175, 174, 187]. It takes 2.4 minutes to finish.

	Objective Value	S(i)	Running Time
SD	122.0374	[93, 197, 145, 164, 175, 174, 187]	2.4 minutes

Table 5 Results for SD method

For the evaluation, the mean is 120.6390, and the confidence interval is between 120.1328 and 121.1453.

	Mean	95% Confidence Interval
SD	120.6390	[120.1328, 121.1453]

Table 6 Evaluation for SD method

2.5 Comparison of three methods

	Mean	95% CI	Running Time
All-In-One	458.04	[446.332, 469.9696]	6.3 minutes
SA	193.04	[183.7315, 203.0226]	31 minutes
SD	120.64	[120.1328, 121.1453]	2.4 minutes

Table 7 Comparison of three methods

We can see that All-In-One method has the worst result, since the mean of evaluation is greatly larger than the other two methods, while SA and SD model have relatively lower objective value. However, the running time of SA method is longer than SD method partly because of the different setting of SA's learning rate. Therefore, we try to improve the SA method in the next part to shorten its running time.

3. Creative Ideas

This part discusses some improvements on the transshipment problem from two aspects: the improvement of the optimization model and improvements on the SA model.

3.1 Improvement of the optimization model

For the improvement on the function, we add a package lost condition. Shipping failure may happen during transshipment, so we add a parameter a_{ij} to denote the result of transshipment from retailer i to j. If the transshipment from i to j is successful, a_{ij}=1; otherwise, a_{ij}=0. The updated dual problem is as follows:

$$\begin{aligned} \textit{maximize} \quad & \textstyle\sum_i s_i B_i + \sum_i d_i M_i + \sum_i d_i R + \sum_i s_i E_i \\ \textit{subject to} \quad & B_i + E_i \le h_i \qquad \forall i && \text{(3a)} \\ & B_i + M_i \le 0 \qquad \forall i && \text{(3b)} \\ & B_i + a_{ij} M_j \le c_{ij} \qquad \forall i, j, i \ne j && \text{(3c)} \\ & M_i + R \le p_i \qquad \forall i && \text{(3d)} \\ & R + E_i \le 0 \qquad \forall i && \text{(3e)} \end{aligned}$$

We set different lost rates ranging from 0.2 to 1, the lost cost as 4 or 10, to get the result.

p=4	0	0.2	0.5	0.8	1
S1	111.79	109.92	108.04	103.69	105.11
S2	214.65	212.95	207.71	227.43	220.44
S3	163.19	161.34	155.5	164.78	165.71
S4	184.46	182.05	176.66	192.34	192.32
S5	193.88	191.19	185.96	179	202.67
S6	183.09	181.92	180.74	205.08	189.91
S7	184.37	181.75	183.46	185.61	212.02

p=10	0	0.2	0.5	0.8	1
S1	119.99	118.06	114.67	123.58	111.84
S2	223.76	221.02	214.38	240.17	226.81
S3	171.32	168.98	164.21	174.46	172.43
S4	193.52	191.2	185.26	195.29	204.01
S5	202.7	200.36	196.17	202.61	211.75
S6	191.38	188.98	188.96	189.7	198.5
S7	193.48	191.02	193.34	195.55	238.57

Table 8 Results of lost package condition

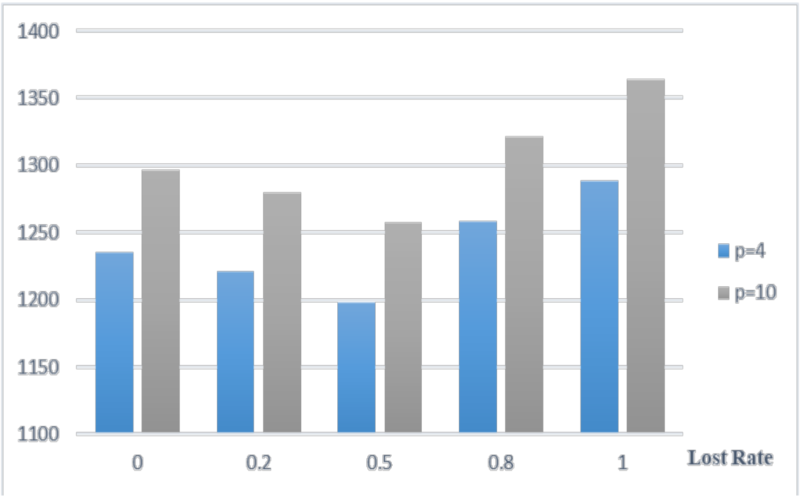


Figure 4 Lost Package Condition

The picture above shows the comparison between the S level when lost penalty equals to 4 or 10. In situation of each lost rate, S will be greater as the lost penalty goes higher.

3.2 Improvement on the SA model

The improvements on the SA method include adding an early stopping condition and taking advantage of Stochastic Gradient Descent with Momentum (SGDM) to make the convergence faster.

First of all, we add an early stopping condition to shorten the running time. We set the parameter of early stopping as 10, which means if the objective value doesn't get better (total cost keeps increasing) for 10 consecutive iterations, we stop the iteration and return the n-11 result.

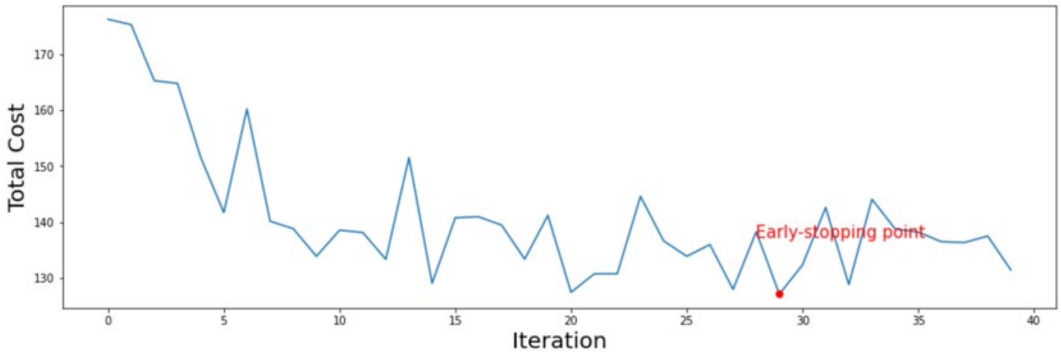


Figure 5 SA with early stopping

The table shows that the process stops early with early stopping method. The total iteration round is 38, much less than 200 as previous. It only takes 6 minutes to finish. It reaches the optimal solution 127 at 28th iteration.

Second, we also try to use SGDM, short for stochastic gradient descent with momentum, to make the converge process faster. Momentum takes into account the past gradients to smooth out the update. The next movement depends on the gradient at this turn and previous movement. It is wisely to consider it as the "velocity" of a ball rolling downhill, building up speed (and momentum) according to the direction of the gradient. Therefore, the converge process is faster.

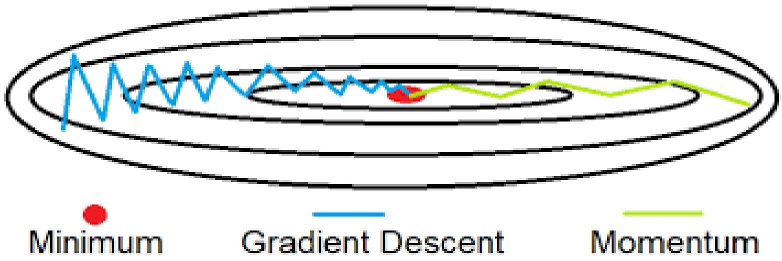


Figure 6 SGDM

The picture above shows the ideal more clearly. Usually, not all directions are equally useful to get to the minimal. That's why the movements are actually going ups and downs. However, the process is smoother and faster with momentum, as the directions of ups and downs are balanced.

Therefore, the algorithm of SGDM is as follows:

$$v = \beta * v + (1 - \beta) * subgradient$$
$$s = s - \eta * v$$

We set the parameters K=200, U=500, size_c=1 (step_wise=size_c/k), and then make a comparison of different strategies.

	Iteration	Order-up-to S	Cost	Running time
Original	200	[106.2864 206.4097 156.3451 176.441 186.4037 176.3483 176.4172]	133.2376	31 minutes
Early-stopping	38	[104.8211 204.9023 154.8526 174.9094 184.8933 174.8651 174.9057]	127.0696	6 minutes
Early-stopping with Momentum (beta=0.2)	20	[103.5913 203.6266 153.6225 173.6311 183.6296 173.6293 173.6311]	124.8491	3 minutes
Early-stopping with Momentum (beta=0.5)	44	[104.8179 204.8678 154.8575 174.88 184.8891 174.864 174.8886]	125.0205	7 minutes

Table 9 Results of SA method with early stopping and momentum strategy

We can see that with the early stopping and with momentum strategy, the convergence rate increases a lot, resulting in the great decrease of the running time. We also try to change the parameter beta. We find that "β = 0.2" is better, since the objective value is 124.8491 and its running time is 3 minutes.

4. Conclusion

In conclusion, multi-location transshipment is a common situation in our daily life, while the stochastic element in the transshipment model makes the problem different to solve. We use three methods, which is All-In-One, SA, and SD, to solve the problem. Comparing the results from these three methods, we find that SD method has the best performance with the objective value of 120.64, and the supply levels of each retailer of [120.1328, 121.1453]. Furthermore, we make some improvements both on the transshipment model and SA method.

5. Creative Contribution

Qianran Ma:
I raised up the idea to use the Gradient Descent with Momentum strategy to improve the SA model to speed up the convergence process.

Wanning Li:
For SA model, I achieved the code of gradient descent with momentum and compared it with plain gradient descent. It turned out with momentum, the algorithm converges faster and shortens the running time used.

Yanan Zhou:
As for All-in-One model, I creatively take the weighted probability of each scenario into account. Since in the real situation, the probability of different scenarios will vary instead of remaining the same.

Yuxuan Cheng:
I creatively introduced “Package Lost” as a possible situation during the transshipment. By adding stochastic variable “result of transshipment” to the function, the model simulate package lost as a normal distribution centered on a certain package lost rate. To achieve this goal, I changed the primal function, constraints, dual problem function accordingly, and did the random generation.
I did experimental study on the “Lost Package” model on SA model with two levels of penalty (4,10) in 5 package lost situations (lost rate=0, 0.2, 0.5, 0.8, 1). We found out S will be greater as the lost penalty goes higher.

Project 2

Meal Plan Problem

Content

1. Background

Health is always a hit problem in young people's life. Nowadays, it is common to see students busy with their classes and have no time to keep a regular diet. However, regular meals can give us enough energy to carry out our daily activities and perform better in both school and work. Thus, to encourage students keep a healthy diet, we designed an optimization model for a group of students to help them make plans for weekly diet. This model needs to consider students eating preference, food allergy, food nutrition, students time availability, students cooking ability, cooking time of different food, limited budget, etc.

2. Assumptions

To make the model more interpretable, we make assumptions before building the model:

- We only make plan for 5 days (Monday - Friday) and one meal per day
- Everyone can acquire the same amount of nutrition from the same dish in a meal
- Everyone needs the same amount of nutrition every day
- A dish costs the same amount of money every time
- We can get all the ingredients we want

3. Basic Model

3.1 Variable Definition

The purpose of this model is to determine the dishes cooked in each meal and the students cooking for each meal. Thus, we first denote the footer letter's meaning:

$\boldsymbol{i}$ stands for students $\boldsymbol{i \in \{1,\ 2,\ 3,\ 4\},\ n_s = 4}$

$\boldsymbol{j}$ stands for dishes: $\boldsymbol{j \in \{1,\ 2,\ \dots,\ 99\},\ n_d = 99}$

$\boldsymbol{k}$ stands for meals: $\boldsymbol{k \in \{1,\ 2,\ \dots,\ 5\},\ n_m = 5}$

With the footnotes above, we denote two decision variable matrixes:

$$d(n_d \times n_m) = \begin{bmatrix} d_{11} & \cdots & d_{1,5} \\ d_{21} & \cdots & d_{2,5} \\ \vdots & \ddots & \vdots \\ d_{98,1} & \cdots & d_{98,5} \\ d_{99,1} & \cdots & d_{99,5} \end{bmatrix}$$

$$d_{jk} = 1 \ \ means\ dish\ j\ is\ cooked\ in\ meal\ k;$$

$$d_{jk} = 0 \ \ means\ dish\ j\ is\ not\ cooked\ in\ meal\ k.$$

$$y(n_s \times n_m) = \begin{bmatrix} y_{11} & y_{12} & \dots & y_{1,99} \\ y_{21} & y_{22} & \dots & y_{2,99} \\ y_{31} & y_{32} & \dots & y_{3,99} \\ y_{41} & y_{42} & \dots & y_{4,99} \end{bmatrix}$$

$$y_{ik} = 1 \text{ } means\ student\ i\ will\ cook\ meal\ k;$$

$$y_{ik} = 0 \text{ } means\ student\ i\ will\ not\ cook\ meal\ k.$$

3.2 Parameter Definition

With the decision variables, we set some parameter matrixes to stands for different constraints.

3.2.1 Rating Matrix

$$r(n_s \times n_d) = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1,99} \\ r_{21} & r_{22} & \dots & r_{2,99} \\ r_{31} & r_{32} & \dots & r_{3,99} \\ r_{41} & r_{42} & \dots & r_{4,99} \end{bmatrix}$$

$$r_{ij}\ means\ the\ rating\ score\ of\ dish\ j\ from\ student\ i$$

3.2.2 Time Availability Matrix

$$Z(n_s \times n_m) = \begin{bmatrix} Z_{11} & Z_{12} & \dots & Z_{1,99} \\ Z_{21} & Z_{22} & \dots & Z_{2,99} \\ Z_{31} & Z_{32} & \dots & Z_{3,99} \\ Z_{41} & Z_{42} & \dots & Z_{4,99} \end{bmatrix}$$

$$Z_{ik}\ means\ the\ total\ available\ cooking\ time(minute)\ of\ student\ i\ during\ meal\ k$$

3.2.3 Cooking Time Vector

$$T(1 \times n_d\ vector) = [T_1 \quad T_2 \quad \cdots \quad T_{99}]$$

$$T_j\ means\ the\ required\ cooking\ time\ of\ dish\ j$$

3.2.4 Nutrition Vectors and Requirements

$N^s\ (1 \times n_d\ vector)\text{:}\ N_j^s\ means\ the\ Sodium\ level\ of\ dish\ j$

$N^c\ (1 \times n_d\ vector)\text{:}\ N_j^c\ means\ the\ Calory\ level\ of\ dish\ j$

$N^f\ (1 \times n_d\ vector)\text{:}\ N_j^f\ means\ the\ Fat\ level\ of\ dish\ j$

$N^p\ (1 \times n_d\ vector)\text{:}\ N_j^p\ means\ the\ Protein\ level\ of\ dish\ j$

$U^s\ ,\ L^s\text{:}\ everyday\ upper\ bound\ and\ lower\ bound\ of\ Sodium\ for\ all\ students$

$U^c\ ,\ L^c\text{:}\ veeryday\ upper\ bound\ and\ lower\ bound\ of\ Calory\ for\ all\ students$

U^f , L^f: $everyday\ upper\ bound\ and\ lower\ bound\ of\ Fat\ for\ all\ students$
U^p , L^p: $everyday\ upper\ bound\ and\ lower\ bound\ of\ Protein\ for\ all\ students$

3.2.5 Price Vectors and Budget

$p\ (1 \times n_d\ vector)$: $p_j\ means\ the\ cost\ of\ dish\ j$

P: $the\ budget\ of\ all\ the\ meals$

3.2.6 Allergy Matrix

$$a\ (n_d \times n_d\ matix) = \begin{bmatrix} a_{11} & \cdots & 0 \\ 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \\ 0 & \cdots & a_{99,99} \end{bmatrix}$$

$$All\ the\ elements\ not\ on\ diagonal\ are\ 0\ \left(a_{ij} = 0\ \ \forall i \neq j\right);$$
$$a_{jj} = 0\ if\ there\ exists\ someone\ who\ is\ allergic\ to\ dish\ j;$$
$$a_{jj} = 1\ if\ no\ one\ is\ allergic\ to\ dish\ j;$$

The purpose of this matrix is to make sure the selected dishes are not the ones some students are allergic to. Although we can simply set $d_{jk} = 0$ if someone is allergic to dish j. To make our model more dynamic and flexible, we choose to use this parameter matrix. For example, we want our model to be suitable for different groups of students and different food allergies. With $d_{jk} = 0$, we need to add/subtract constraints in our model to adapt to different situations, which means we need to change our model when situations vary. However, with the allergy matrix, we only need to change parameters every time and the model remains the same. We think the allergy matrix is a more flexible and dynamic solution.

3.3 Basic Model

$$\textbf{Maximize} \quad \sum_{i=1}^{n_s}\sum_{k=1}^{n_m}\sum_{i=1}^{n_d} r_{ij} d_{jk} \tag{1}$$

Subject to

$$\sum_{j=1}^{n_d} d_{jk} \le 3\,, \quad for\ \forall k \in \{1,\ 2,\ \dots,\ 5\} \tag{2}$$

$$\sum_{k=1}^{n_m} d_{jk} \le 2\,, \quad for\ \forall j \in \{1,\ 2,\ \dots,\ 99\} \tag{3}$$

$$\sum_{j=1}^{n_d} N_j^s d_{jk} \le U^s, \quad \sum_{j=1}^{n_d} N_j^s d_{jk} \ge L^s \qquad for\ \forall k \in \{1,\ 2,\ 3,\ 4, 5\} \tag{4}$$

$$\sum_{j=1}^{n_d} N_j^c d_{jk} \le U^c, \quad \sum_{j=1}^{n_d} N_j^c d_{jk} \ge L^c \qquad for\ \forall k \in \{1,\ 2, 3, 4, 5\} \tag{5}$$

$$\sum_{j=1}^{n_d} N_j^f d_{jk} \le U^f, \quad \sum_{j=1}^{n_d} N_j^f d_{jk} \ge L^f \qquad for\ \forall k \in \{1,\ 2, 3, 4, 5\} \tag{6}$$

$$\sum_{j=1}^{n_d} N_j^p d_{jk} \le U^p, \quad \sum_{j=1}^{n_d} N_j^p d_{jk} \ge L^p \qquad for\ \forall k \in \{1,\ 2, 3, 4, 5\} \tag{7}$$

$$\sum_{k=1}^{n_m}\sum_{j=1}^{n_d} p_j d_{jk} \le P \tag{8}$$

$$a_{jj} d_{jk} = d_{jk} \qquad \forall k \in \{1,\ 2,\ \dots,\ 5\},\ \forall j \in \{1,\ 2,\ \dots,\ 99\} \tag{9}$$

$$y_{ik} \le Z_{ik} \qquad \forall i \in \{1,\ 2,\ 3, 4\} \quad \forall k \in \{1,\ 2,\ \dots,\ 5\} \tag{10}$$

$$\sum_{i=1}^{n_s} y_{ik} \le 2, \qquad \forall k \in \{1,\ 2,\ \dots,\ 5\} \tag{11}$$

$$\sum_{k=1}^{n_m} y_{ik} \le 3, \qquad \sum_{k=1}^{n_m} y_{ik} \ge 2, \qquad \forall i \in \{1,\ 2, 3,\ 4\} \tag{12}$$

$$T_j \times d_{jk} \le \sum_{i=1}^{n_s} y_{ik} \times Z_{ik}\,, \quad \forall j \in \{1,\ 2,\ \dots, 99\},\ \forall k \in \{1,\ 2, \dots,\ 5\} \tag{13}$$

$$y_{ik} \in \{0,\ 1\} \qquad \forall i \in \{1,\ 2,\ 3, 4\}\,, \forall k \in \{1,\ 2,\ \dots,\ 5\} \tag{14}$$

$$d_{jk} \in \{0,\ 1\} \qquad \forall j \in \{1,\ 2,\ \dots, 99\}\,, \forall k \in \{1,\ 2,\ \dots,\ 5\} \tag{15}$$

3.3.1 Objective function

(1) maximize the sum of all students' ratings of all the selected dishes:

$$\sum_{i=1}^{n_s}\sum_{k=1}^{n_m}\sum_{j=1}^{n_d} r_{ij}d_{jk}$$

$$\begin{bmatrix} r_{11} & r_{12} & \dots & r_{1,99} \\ r_{21} & r_{22} & \dots & r_{2,99} \\ r_{31} & r_{32} & \dots & r_{3,99} \\ r_{41} & r_{42} & \dots & r_{4,99} \end{bmatrix} \times \begin{bmatrix} d_{11} & \cdots & d_{1,5} \\ d_{21} & \cdots & d_{2,5} \\ \vdots & \ddots & \vdots \\ d_{98,1} & \cdots & d_{98,5} \\ d_{99,1} & \cdots & d_{99,5} \end{bmatrix} = \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1,5} \\ s_{21} & s_{22} & \dots & s_{2,5} \\ s_{31} & s_{32} & \dots & s_{3,5} \\ s_{41} & s_{42} & \dots & s_{4,5} \end{bmatrix}$$

s_{ik}: *The total rating score of meal k from Teletubby i*

$sum(s_{ik})$: *The total rating score of all meals from all Teletubbies*

3.3.2 Constraints

(2) We cook no more than 3 dishes in each meal
(3) We eat the same dish at most twice
(4) ~ (7) The sum of nutrition of everyday meals satisfies requirement
(8) The sum of total cost of all the meals does not exceed budget
(9) No one is allergic to the selected dishes
(10) The students who cook at this meal must be available
(11) Each meal will be cooked by at most 2 students
(12) Each student will cook 2~3 times a week
(13) The minimum required cooking time of all dishes in meal k should be no more than total available cooking time of all students who cook for meal k
(14) Decision variable y_{ik} should be 0 or 1
(15) Decision variable d_{jk} should be 0 or 1

3.4 Data Collection and Parameter Calculation

3.4.1 Rating Matrix: Matrix Factorization

One of our main goals of this meal plan problem is to recommend dishes according to students' eating preference. However, we have dozens of dishes in our dish list and not all the dishes have been eaten by each student. Thus, to make our recommendation more perfect, we need to estimate the unknown ratings according to the known ratings.

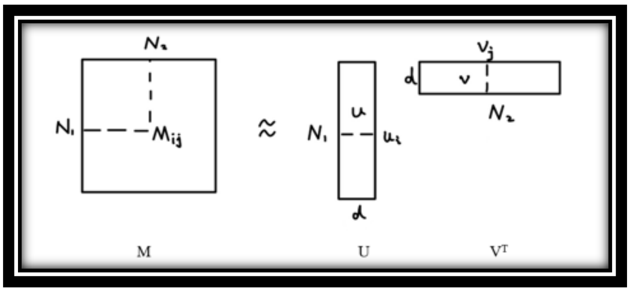
Sparse Matrix:

- **No one rate all dishes**
- **No dishes receive ratings from everyone**

Similarity:

- **Many people with similar tastes**
- **Many dishes with similar attributes**

(1) Low-rank Factorization



$$\boldsymbol{M = UV^T}$$
$$\boldsymbol{u_i = (u_{i1}, u_{i2}, \cdots, u_{id})} \;\; \textit{Features of the user i}$$
$$\boldsymbol{v_j = (v_{j1}, v_{j2}, \cdots, v_{jd})} \;\; \textit{Features of the dish j}$$
$$\boldsymbol{d \ll \min\{N_1, N_2\}}$$
$$\boldsymbol{M_{ij} \approx u_i v_j^T = u_{i1}v_{j1} + u_{i2}v_{j2} + \cdots + u_{id}v_{jd}}$$

Low-rank algorithm decomposes a matrix into two matrix U and V. In our problem, we can consider U is the student matrix, V is the dish matrix. Similar, we can think the d attributes in U and V here as the information/features about student/dish (i.e., students: age/sex/country…, dishes: type/flavor/ingredients…). In this way, each element in M matrix is a combination of the features of students and dishes.

Objective Function:

$$\min_{u_i, v_j} \underbrace{\boxed{\sum_{(i,j)\in\Omega} \frac{\lambda}{2} ||M_{ij} - u_i^T v_j||^2}}_{\text{make predicted rating as close as possible to the real rating}} + \underbrace{\boxed{\sum_{i=1}^{N_1} \frac{\mu}{2} ||u_i||^2 + \sum_{j=1}^{N_2} \frac{\mu}{2} ||v_j||^2}}_{\text{Regularization to avoid overfitting}} \triangleq L$$

$\Omega = \{(i,j): M_{ij} \textit{ is measured}$
$(i,j) \in \Omega \;\; \textit{if person i has rated dish j}$
$\Omega_{u_i} \textit{ be the index set of dishes rated by user i}$
$\Omega_{v_j} \textit{ be the index set of users who rated dish j}$

The first part of objective function is to make the difference between predicted value and true value in M as small as possible. The last two parts are the penalty to avoid overfitting.

Partial Derivative:

$$D_{u_i} L = \sum_{j\in\Omega_{u_i}} \lambda(M_{ij} - u_i^T v_j) v_j - \mu u_i = 0 \quad \forall i = 1, 2, \cdots, N_1$$
$$D_{v_j} L = \sum_{i\in\Omega_{v_j}} \lambda(M_{ij} - u_i^T v_j) u_i - \mu v_j = 0 \quad \forall j = 1, 2, \cdots, N_2$$

$$u_i = \left(\frac{\mu}{\lambda} I + \sum_{j\in\Omega_{u_i}} v_j v_j^T\right)^{-1} \left(\sum_{j\in\Omega_{u_i}} M_{ij} v_j\right) \quad \forall i = 1, 2, \cdots, N_1$$
$$v_j = \left(\frac{\mu}{\lambda} I + \sum_{i\in\Omega_{v_j}} u_i u_i^T\right)^{-1} \left(\sum_{i\in\Omega_{v_j}} M_{ij} u_i\right) \quad \forall j = 1, 2, \cdots, N_2$$

To solve this problem, we calculate the partial derivatives of $\boldsymbol{u_i}$ and $\boldsymbol{v_j}$. It is obvious that the function of $\boldsymbol{u_i}$'s gradient includes $\boldsymbol{v_j}$ while the function of $\boldsymbol{v_j}$'s gradient includes $\boldsymbol{u_i}$. Thus, we can solve this non-linear optimization program by coordinate descent algorithm.

Low-rank Algorithm with Coordinate Descent

1. $Initialize\ v_j \sim N(0,\ \lambda^{-1}I),\ u_i\ as\ vector\ 0.0$
2. $For\ each\ iteration:$

$for\ i = 1,\ 2,\ \cdots,\ N_1:$

$$u_i = \left(\frac{\mu}{\lambda}I + \sum_{j\in\Omega_{u_i}} v_j v_j^T\right)^{-1}\left(\sum_{j\in\Omega_{u_i}} M_{ij}v_j\right)$$

$for\ j = 1,\ 2,\ \cdots,\ N_2:$

$$v_j = \left(\frac{\mu}{\lambda}I + \sum_{i\in\Omega_{v_j}} u_i u_i^T\right)^{-1}\left(\sum_{i\in\Omega_{v_j}} M_{ij}u_i\right)$$

3. $Until\ converge$

4. $M_{ij} = u_i^T v_j\ \ Round\ it\ to\ the\ closest\ rating$

According to the algorithm above, there are several parameters to be chosen:

- $coeffieient\ \lambda$
- $coefficient\ \mu$
- $matrix\ shape\ d$
- $iteration\ round\ K$

According to Yang’s paper, the most common type of regularization is L2, which is also called simply “weight decay”, with values often on a logarithmic scale between 0 and 0.1, such as 0.1, 0.001, 0.0001, etc. Thus, we tried these values for parameter λ and μ. As for parameter d, we set d equals to 1-5, to see which performed better.

(2) Matrix Completion Result

Compare results of different d:

	Objective Value	**Running Time**
d=1	**2239.36**	**54 seconds**
d=2	**128.48**	**1 minute and 22 seconds**
d=3	**12.14**	**1 minute and 5 seconds**
d=4	**9.95**	**1 minute and 18 seconds**
d=5	**9.11**	**1 minute and 25 seconds**
λ **= 0.999,** μ **= 0.001,** K **= 200**		

Table 10 Low-rank results with different d

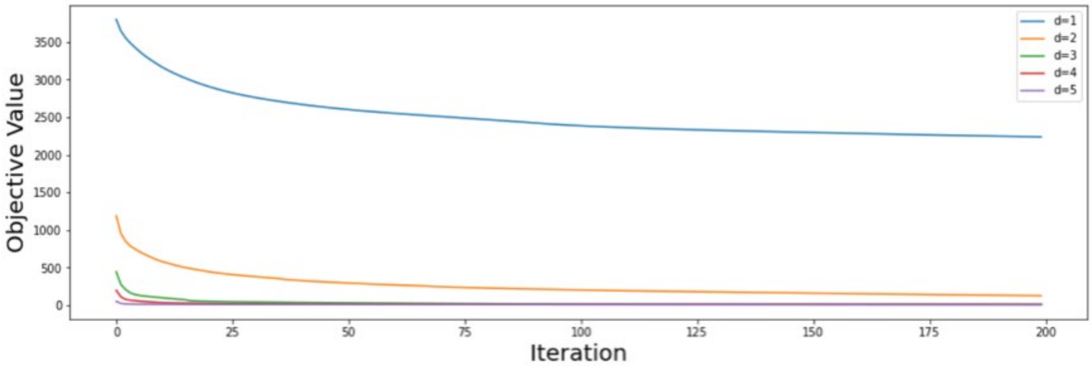


Figure 7 Trend of objective values with different d (1)

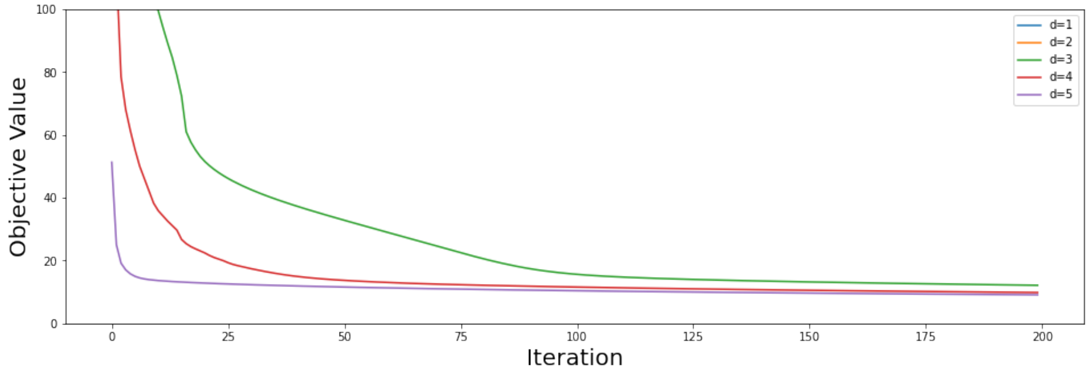


Figure 8 Trend of objective values with different d (2)

The figures are the trend of the objective values with d = 1/2/3/4/5. Obviously, in Figure 7, the result of d = 1 is bad. In Figure 8, we zoom in the y axis and limit it to [1:100]. Although in Table 1, d = 5 has slightly longer running time, it converges very fast, which means it is the earliest one to stop deceasing. Thus, if we add an early stopping condition to the algorithm, d = 5 will be the best choice.

With λ **= 0.999,** μ **= 0.001,** K **= 200,** d = 5, the predicted matrix is shown below:

title	Curried Lentil, Tomato, and Coconut Soup	Roasted Butternut Squash with Herb Oil and Goat Cheese	Pumpkin Muffins	Chopped Salad with Shallot Vinaigrette, Feta, and Dill	Grain Salad with Olives and Whole-Lemon Vinaigrette	Chilled Coconut Corn Soup	Vietnamese-Style Spaghetti Squash “Noodle” Bowls with Skirt Steak	Roasted Squash with Mint and Toasted Pumpkin Seeds	Butternut Squash Steaks with Brown Butter–Sage Sauce	Freeform Chicken Meatballs with Carrots and Yogurt Sauce	...	Twice-Roasted Squash with Parmesan Butter and Grains	Kale Salad with Butternut Squash, Pomegranate, and Pumpkin Seeds	Kabocha Squash Pilaf with Coconut
Wanning	1	5	1	2	3	1	1	4	4	2	...	1	1	3
Yuxuan	1	4	1	4	2	2	1	4	2	1	...	1	3	2
Qianran	1	2	1	2	3	1	1	4	3	1	...	1	1	2
Yanan	1	1	1	1	1	1	1	3	1	1	...	1	1	1

Table 11 The predicted rating matrix

3.4.2 Time Availability Matrix

Collect each student's available time and store them in minute level.

3.4.3 Cooking Time Vector

Collect each dish's cooking time from the dataset.

3.4.4 Nutrition Vectors and Requirements

Collect $\boldsymbol{N^s}, \boldsymbol{N^c}, \boldsymbol{N^f}, \boldsymbol{N^p}$ from the dataset and collect $(\boldsymbol{U^s}, \boldsymbol{L^s})$, $(\boldsymbol{U^c}, \boldsymbol{L^c})$, $(\boldsymbol{U^f}, \boldsymbol{L^f})$, $(\boldsymbol{L^p}, \boldsymbol{L^p})$ from the website: https://health.gov/

3.4.5 Price Vectors and Budget

Collect each dish's estimated price from the dataset and set the budget manually

3.4.6 Allergy Matrix

Collect each student's allergy food and build the allergy matrix

3.5 Basic Model Results

Total Rating Score = 197.0		
Weekday	**Dishes**	**Cooking Students**
Monday	Easy Green Curry with Chicken, Bell Pepper, and Sugar Snap Peas Grilled Cheese Tacos Veggie Burgers with Zucchini and Corn	Dipsy Tinky-Winky
Tuesday	Easy Green Curry with Chicken, Bell Pepper, and Sugar Snap Peas Quick Sesame Chicken with Broccoli Veggie Burgers with Zucchini and Corn	Po
Wednesday	French Spiced Bread Grain Bowls with Chicken, Spiced Chickpeas, and Avocado Orange Sweet Rolls	Dipsy Po
Thursday	Butternut Squash Sandwich with Cheddar Cheese and Pickled Red Onion French Spiced Bread Roasted Garlic Herb Sauce	Laa-Laa Tinky-Winky

Friday	Butternut Squash Sandwich with Cheddar Cheese and Pickled Red Onion Grilled Cheese Tacos Quick Sesame Chicken with Broccoli	Laa-Laa

Table 12 The Result of Basic Model

4. Creative Ideas

4.1 Cooking Ability

Beside from the requirements considered in the basic model, it is also important to consider each student's cooking ability. In real situation, we cannot make sure all the students can cook all the dishes in the dish list. Thus, we add some new constraints to the basic model to ensure all the dishes we select can be cooked by at least one cooking student.

4.1.1 Cooking Ability Matrix

$$M(n_s \times n_d) = \begin{bmatrix} M_{11} & M_{12} & \dots & M_{1,99} \\ M_{21} & M_{22} & \dots & M_{2,99} \\ M_{31} & M_{32} & \dots & M_{3,99} \\ M_{41} & M_{42} & \dots & M_{4,99} \end{bmatrix}$$

M_{ij} *means whether the student i can cook dish j*

We collected the dishes each student can or cannot cook manually and build the matrix. To illustrate this constraint do take effect, we set all students cannot cook some dishes appearing in previous results.

- Tinky-Winky can only cook ['Pasta', 'Noodle', 'Noodles', 'Burger', 'Salad']
- Dipsy cannot cook ['Beef', 'Lamb', 'Chicken', 'Bread']
- Laa-Laa cannot cook ['cake', 'Pie', 'Chicken', 'Bread']
- Po cannot cook ['Shrimp', 'Shell', 'Fish', 'Chicken', 'Bread']

According to the requirements above, the expected result of this advanced model should not include "Easy Green Curry with Chicken, Bell Pepper, and Sugar Snap Peas", "Quick Sesame Chicken with Broccoli", "French Spiced Bread", and "Grain Bowls with Chicken, Spiced Chickpeas, and Avocado".

4.1.2 Model with Cooking Ability

$$\textbf{Maximize} \quad \sum_{i=1}^{n_s}\sum_{k=1}^{n_m}\sum_{j=1}^{n_d} r_{ij}d_{jk} \tag{1}$$

Subject to

$$\sum_{j=1}^{n_d} d_{jk} \le 3, \quad for\ \forall k \in \{1, 2, \dots, 5\} \tag{2}$$

$$\sum_{k=1}^{n_m} d_{jk} \le 2, \quad for\ \forall j \in \{1, 2, \dots, 99\} \tag{3}$$

$$\sum_{j=1}^{n_d} N_j^s d_{jk} \le U^s, \quad \sum_{j=1}^{n_d} N_j^s d_{jk} \ge L^s \quad for\ \forall k \in \{1, 2, 3, 4, 5\} \tag{4}$$

$$\sum_{j=1}^{n_d} N_j^c d_{jk} \le U^c, \quad \sum_{j=1}^{n_d} N_j^c d_{jk} \ge L^c \quad for\ \forall k \in \{1, 2, 3, 4, 5\} \tag{5}$$

$$\sum_{j=1}^{n_d} N_j^f d_{jk} \le U^f, \quad \sum_{j=1}^{n_d} N_j^f d_{jk} \ge L^f \quad for\ \forall k \in \{1, 2, 3, 4, 5\} \tag{6}$$

$$\sum_{j=1}^{n_d} N_j^p d_{jk} \le U^p, \quad \sum_{j=1}^{n_d} N_j^p d_{jk} \ge L^p \quad for\ \forall k \in \{1, 2, 3, 4, 5\} \tag{7}$$

$$\sum_{k=1}^{n_m}\sum_{j=1}^{n_d} p_j d_{jk} \le P \tag{8}$$

$$a_{jj}d_{jk} = d_{jk} \quad \forall k \in \{1, 2, \dots, 5\}, \forall j \in \{1, 2, \dots, 99\} \tag{9}$$

$$y_{ik} \le Z_{ik} \quad \forall i \in \{1, 2, 3, 4\} \quad \forall k \in \{1, 2, \dots, 5\} \tag{10}$$

$$\sum_{i=1}^{n_s} y_{ik} \le 2, \quad \forall k \in \{1, 2, \dots, 5\} \tag{11}$$

$$\sum_{k=1}^{n_m} y_{ik} \le 3, \quad \sum_{k=1}^{n_m} y_{ik} \ge 2, \quad \forall i \in \{1, 2, 3, 4\} \tag{12}$$

$$T_j \times d_{jk} \le \sum_{i=1}^{n_s} y_{ik} \times Z_{ik}, \quad \forall j \in \{1, 2, \dots, 99\}, \forall k \in \{1, 2, \dots, 5\} \tag{13}$$

$$d_{jk} \le \sum_{i=1}^{n_s} M_{ij}y_{ik} \quad \forall k \in \{1, 2, \dots, 5\}, \forall j \in \{1, 2, \dots, 99\} \tag{14}$$

$$y_{ik} \in \{0, 1\} \quad \forall i \in \{1, 2, 3, 4\}, \forall k \in \{1, 2, \dots, 5\} \tag{15}$$

$$d_{jk} \in \{0, 1\} \quad \forall j \in \{1, 2, \dots, 99\}, \forall k \in \{1, 2, \dots, 5\} \tag{16}$$

All the other constraints are the same with basic model. As for the constraints (14), it is designed to make sure if a dish is selected in meal k, then there must be at least one cooking students can cook this dish.

$$\underbrace{\begin{bmatrix} M_{11} & M_{21} & M_{31} & M_{41} \\ M_{12} & M_{22} & M_{32} & M_{42} \\ \vdots & \vdots & \vdots & \vdots \\ M_{1,99} & M_{2,99} & M_{3,99} & M_{4,99} \end{bmatrix}}_{M^T} \times \underbrace{\begin{bmatrix} y_{11} & y_{12} & \dots & y_{1,99} \\ y_{21} & y_{22} & \dots & y_{2,99} \\ y_{31} & y_{32} & \dots & y_{3,99} \\ y_{41} & y_{42} & \dots & y_{4,99} \end{bmatrix}}_{y} = \underbrace{\begin{bmatrix} D_{11} & D_{1,2} & \cdots & D_{1,5} \\ D_{21} & D_{22} & \cdots & D_{2,5} \\ \vdots & \vdots & \ddots & \vdots \\ D_{99,1} & D_{99,2} & \cdots & D_{99,5} \end{bmatrix}}_{D}$$

$$D_{jk}\text{: The number of Teletubbies who: } \begin{cases} \text{cook for meal } k \\ \text{can cook dish } j \end{cases} \quad \rightarrow d_{jk} \le D_{jk} = \sum_{i=1}^{n_s} M_{ij} y_{ik}$$

4.1.3 Result

Total Rating Score = 179.0		
Weekday	**Dishes**	**Cooking Students**
Monday	Instant Pot Beef and Sweet Potato Chili Orange Sweet Rolls Roasted Garlic Herb Sauce	Po Tinky-Winky
Tuesday	Butternut Squash Sandwich with Cheddar Cheese and Pickled Red Onion Orange Sweet Rolls Roasted Garlic Herb Sauce	Po Tinky-Winky
Wednesday	Garlic Mojo Sauce Grilled Cheese Tacos Veggie Burgers with Zucchin and Corn	Dipsy
Thursday	Butternut Squash Sandwich with Cheddar Cheese and Pickled Red Onion Grilled Cheese Tacos Stuffed Sweet Potatoes with Curried Chickpeas and Mushrooms	Laa-Laa Tinky-Winky
Friday	Instant Pot Beef and Sweet Potato Chili Stuffed Sweet Potatoes with Curried Chickpeas and Mushrooms Veggie Burgers with Zucchini and Corn	Dipsy Laa-Laa

Table 13 The Result of Model with Cooking Ability

4.2 Eating Outside

In real life situation, it is common to see all of the students in the group have the same class at the same time and no one is available during the cooking time of one meal. However, no one is available for cooking does not mean students cannot eat anything. They still can eat a healthy diet outside. Although the food list in the restaurant must be more diverse than the food students can cook at home, we still can recommend some food according to our cooking dish list to provide students a reference when choosing food in the restaurant.

In our basic model, if no one is available during one meal, the model will be infeasible. The reason is that we set constraints to make sure the cooking time of selected dishes is no more than the total available time of students, but no one is available means the total available time is 0. Thus, there will be no dishes selected for this meal so the nutrition requirement cannot be met. To solve this potential problem, we can add some assistant parameters and build a two-stage optimization model set to figure out on which days students should eat outside and which food we recommend students to buy on these days.

4.2.1 Stage - 1 Model

Get the cooking plan for other available days and output "Eat Outside" for the day no one is available.

New Parameters: Assistant Nutrition Vectors

If no one is available during one meal, then the previous basic model will be infeasible since there will be no dish on that day and the nutrition requirement cannot be met. Thus, we define for Assistant Nutrition Vectors to solve this problem:

$$sodium = [s_1 \quad s_2 \quad s_3 \quad s_4 \quad s_5]$$
$$calory = [c_1 \quad c_2 \quad c_3 \quad c_4 \quad c_5]$$
$$fat = [f_1 \quad f_2 \quad f_3 \quad f_4 \quad f_5]$$
$$protein = [pr_1 \quad pr_2 \quad pr_3 \quad pr_4 \quad pr_5]$$

$$If \sum_{i=1}^{n_s} Z_{ik} = 0, \begin{cases} then \;\; s_k = L^s, & \forall k \in \{1, 2, \dots, 5\} \\ then \;\; c_k = L^c, & \forall k \in \{1, 2, \dots, 5\} \\ then \;\; f_k = L^f, & \forall k \in \{1, 2, \dots, 5\} \\ then \;\; pr_k = L^p, & \forall k \in \{1, 2, \dots, 5\} \end{cases}$$

$$If \sum_{i=1}^{n_s} Z_{ik} \geq 1, \begin{cases} then \;\; s_k = 0, & \forall k \in \{1, 2, \dots, 5\} \\ then \;\; c_k = 0, & \forall k \in \{1, 2, \dots, 5\} \\ then \;\; f_k = 0, & \forall k \in \{1, 2, \dots, 5\} \\ then \;\; pr_k = 0, & \forall k \in \{1, 2, \dots, 5\} \end{cases}$$

With these assistant vectors, we can modify the original nutrition constraints by adding each of the elements in the assistant vectors to the sum of everyday nutrition we get from the dishes. For example, we can change the sodium constraints from [$\boldsymbol{L^s \leq \sum_{j=1}^{n_d} N_j^s d_{jk} \leq U^s}$] to $[\boldsymbol{L^s \leq s_k + \sum_{j=1}^{n_d} N_j^s d_{jk} \leq U^s}]$. In this way, when no one is available, $\sum_{i=1}^{n_s} Z_{ik} = 0$, but $s_k = L^s$. Thus, $s_k + \sum_{j=1}^{n_d} N_j^s d_{jk}$ will still remain within $L^s \; and \; U^s$, which will keep the model feasible.

Output Display: "No one is available! Eat Outside!"

We set up the model to output "No one is available! Eat outside!" when no one is available on a day.

Stage - 1 Model:

$$\textbf{Maximize} \quad \sum^{n_s}\sum^{n_m}\sum^{n_d} r_{ij}d_{jk} \tag{1}$$

$$\textbf{Subject to} \quad \sum_{j=1}^{n_d} d_{jk} \le 3, \quad for\ \forall k \in \{1, 2, \dots, 5\} \tag{2}$$

$$\sum_{k=1}^{n_m} d_{jk} \le 2, \quad for\ \forall j \in \{1, 2, \dots, 99\} \tag{3}$$

$$s_k + \sum_{j=1}^{n_d} N_j^s d_{jk} \le U^s, \ s_k + \sum_{j=1}^{n_d} N_j^s d_{jk} \ge L^s \quad for\ \forall k \in \{1, 2, 3, 4, 5\} \tag{4}$$

$$c_k + \sum_{j=1}^{n_d} N_j^c d_{jk} \le U^c, \ c_k + \sum_{j=1}^{n_d} N_j^c d_{jk} \ge L^c \quad for\ \forall k \in \{1, 2, 3, 4, 5\} \tag{5}$$

$$f_k + \sum_{j=1}^{n_d} N_j^f d_{jk} \le U^f, \ f_k + \sum_{j=1}^{n_d} N_j^f d_{jk} \ge L^f \quad for\ \forall k \in \{1, 2, 3, 4, 5\} \tag{6}$$

$$pr_k + \sum_{j=1}^{n_d} N_j^p d_{jk} \le U^p, \ pr_k + \sum_{j=1}^{n_d} N_j^p d_{jk} \ge L^p \quad for\ \forall k \in \{1, 2, 3, 4, 5\} \tag{7}$$

$$\sum_{k=1}^{n_m}\sum_{j=1}^{n_d} p_j d_{jk} \le P \tag{8}$$

$$a_{jj}d_{jk} = d_{jk} \quad \forall k \in \{1, 2, \dots, 5\}, \forall j \in \{1, 2, \dots, 99\} \tag{9}$$

$$y_{ik} \le Z_{ik} \quad \forall i \in \{1, 2, 3, 4\} \quad \forall k \in \{1, 2, \dots, 5\} \tag{10}$$

$$\sum_{i=1}^{n_s} y_{ik} \le 2, \quad \forall k \in \{1, 2, \dots, 5\} \tag{11}$$

$$\sum_{k=1}^{n_m} y_{ik} \le 3, \quad \sum_{k=1}^{n_m} y_{ik} \ge 2, \quad \forall i \in \{1, 2, 3, 4\} \tag{12}$$

$$T_j \times d_{jk} \le \sum_{i=1}^{n_s} y_{ik} \times Z_{ik}, \ \forall j \in \{1, 2, \dots, 99\}, \forall k \in \{1, 2, \dots, 5\} \tag{13}$$

$$d_{jk} \le \sum_{i=1}^{n_s} M_{ij} y_{ik} \quad \forall k \in \{1, 2, \dots, 5\}, \forall j \in \{1, 2, \dots, 99\} \tag{14}$$

$$y_{ik} \in \{0, 1\} \quad \forall i \in \{1, 2, 3, 4\}, \forall k \in \{1, 2, \dots, 5\} \tag{15}$$

$$d_{jk} \in \{0, 1\} \quad \forall j \in \{1, 2, \dots, 99\}, \forall k \in \{1, 2, \dots, 5\} \tag{16}$$

All the other constraints are the same with basic model. As for constraints (4)-(7), we include the Assistant Nutrition Vectors together to deal with the infeasibility when no one is available during that meal. Also, we calculate this assistant matrix according to the time availability matrix as stated above.

4.2.2 Stage - 2 Model

Make recommendations for eating outside.

Decision Variable:

$$o(n_d \times n_m) = \begin{bmatrix} o_{11} & \cdots & o_{1,5} \\ o_{21} & \cdots & o_{2,5} \\ \vdots & \ddots & \vdots \\ o_{98,1} & \cdots & o_{98,5} \\ o_{99,1} & \cdots & o_{99,5} \end{bmatrix}$$

$$o_{jk} = 1 \ \ means\ dish\ j\ is\ recommended\ to\ buy\ in\ meal\ k;$$

$$o_{jk} = 0 \ \ means\ dish\ j\ is\ not\ recommended\ to\ buy\ in\ meal\ k.$$

No Time List: A list reflecting whether no one is available during each meal

$$B = [B_1 \quad B_2 \quad B_3 \quad B_4 \quad B_5] \qquad \begin{cases} \boldsymbol{if \sum_{i=1}^{4} y^*_{ik} = 0, \qquad B_k = 0} \\ \boldsymbol{if \sum_{i=1}^{4} y^*_{ik} > 0, \qquad B_k = 1} \end{cases}$$

We generate this assistant vector according to the output of stage-1 model. With this vector, we can set some constraints in the stage-2 model to make sure we recommend 2~3 dishes when no one is available for cooking and recommend nothing if there exists someone to cook.

Cook Dish List: A list reflecting whether one dish is selected to be cooked in Stage – 1 Model

$$C = [C_1 \quad C_2 \quad C_3 \quad \dots \quad C_{99}] \qquad \begin{cases} \boldsymbol{if \sum_{k=1}^{5} d^*_{jk} = 0, \qquad C_j = 1} \\ \boldsymbol{if \sum_{k=1}^{5} d^*_{jk} > 0, \qquad C_j = 0} \end{cases}$$

We generate this assistant vector according to the output of stage-1 model. With this vector, we can set some constraints in the stage-2 model to make sure we do not recommend dishes which have already been selected to cook this week, which keeps the eating diversity.

Stage - 2 Model:

$$\textit{Maximize} \quad \sum_{i=1}^{n_s}\sum_{k=1}^{n_m}\sum_{j=1}^{n_d} r_{ij} o_{jk} \tag{1}$$

$$\textit{Subject to} \quad \sum_{j=1}^{n_d} o_{jk} \le 3, \quad for\ \forall k \in \{1, 2, \ldots, 5\} \tag{2}$$

$$\sum_{k=1}^{n_m} o_{jk} \le 1, \quad for\ \forall j \in \{1, 2, \ldots, 99\} \tag{3}$$

$$s_k + \sum_{j=1}^{n_d} N_j^s o_{jk} \le U^s,\ s_k + \sum_{j=1}^{n_d} N_j^s o_{jk} \ge L^s \quad for\ \forall k \in \{1, 2, 3, 4, 5\} \tag{4}$$

$$c_k + \sum_{j=1}^{n_d} N_j^c o_{jk} \le U^c,\ c_k + \sum_{j=1}^{n_d} N_j^c o_{jk} \ge L^c \quad for\ \forall k \in \{1, 2, 3, 4, 5\} \tag{5}$$

$$f_k + \sum_{j=1}^{n_d} N_j^f o_{jk} \le U^f,\ f_k + \sum_{j=1}^{n_d} N_j^f o_{jk} \ge L^f \quad for\ \forall k \in \{1, 2, 3, 4, 5\} \tag{6}$$

$$pr_k + \sum_{j=1}^{n_d} N_j^p o_{jk} \le U^p,\ pr_k + \sum_{j=1}^{n_d} N_j^p o_{jk} \ge L^p \quad for\ \forall k \in \{1, 2, 3, 4, 5\} \tag{7}$$

$$\sum_{k=1}^{n_m}\sum_{j=1}^{n_d} p_j o_{jk} \le P' \tag{8}$$

$$a_{jj} o_{jk} = o_{jk} \quad \forall k \in \{1, 2, \ldots, 5\}, \forall j \in \{1, 2, \ldots, 99\} \tag{9}$$

$$B_k \times \sum_{j=1}^{99} o_{jk} = 0 \quad for\ k = 1, 2, \ldots, 5 \tag{10}$$

$$2 \times B_k + \sum_{j=1}^{99} o_{jk} \ge 2 \quad for\ k = 1, 2, \ldots, 5 \tag{11}$$

$$2 \times B_k + \sum_{j=1}^{99} o_{jk} \le 3 \quad for\ k = 1, 2, \ldots, 5 \tag{12}$$

$$o_{jk} \le C_j \quad \forall k \in \{1, 2, \ldots, 5\}, \forall j \in \{1, 2, \ldots, 99\} \tag{13}$$

$$o_{jk} \in \{0, 1\} \quad \forall j \in \{1, 2, \ldots, 99\}, \forall k \in \{1, 2, \ldots, 5\} \tag{14}$$

In this model, we still need to take the number of dishes we recommend, nutrition requirement, food allergy, and budget into consideration. The good news is we do not need to consider the cooking time of dishes, time availability and cooking ability of each student:

(2) We recommend no more than 3 dishes in each meal
(3) We recommend the same dish at most once
(4) ~ (7) The sum of nutrition of everyday meals satisfies requirement
(8) The sum of total cost of all the meals does not exceed remaining budget
(9) No one is allergic to the recommended dishes

However, there are several extra constraints we need to take into account. According to

the results of stage-1 model, we need to make sure to recommend dishes on the day that no one is available and do not recommend anything on the day we can cook. Also, the dishes we recommend should be different from the dishes we plan to cook, which ensures the eating diversity:
(10) ~ (12) If there exists someone who can cook meal k, then do not buy dishes for this meal. If no one is available for cooking meal k, then recommend 2-3 dishes and buy it in the restaurant.
(13) If dish j is selected to be cooked in other meals, then do not recommend dish j for this meal.

4.2.3 Result

Stage-1:

Total Rating Score = 148.0		
Weekday	**Dishes**	**Cooking Students**
Monday	Butternut Squash Sandwich with Cheddar Cheese and Pickled Red Onion Grilled Cheese Tacos Stuffed Sweet Potatoes with Curried Chickpeas and Mushrooms	Laa-Laa Tinky-Winky
Tuesday	No one is available! Eat outside!	
Wednesday	Grilled Cheese Tacos Instant Pot Beef and Sweet Potato Chili Veggie Burgers with Zucchini and Corn	Po Tinky-Winky
Thursday	Butternut Squash Sandwich with Cheddar Cheese and Pickled Red Onion Orange Sweet Rolls Roasted Garlic Herb Sauce	Dipsy Laa-Laa
Friday	Orange Sweet Rolls Roasted Garlic Herb Sauce Veggie Burgers with Zucchini and Corn	Dipsy Po

Table 14 The Result of Eating Outside Model – Stage 1

Stage-2:

Total Rating Score = 41.0	
Weekday	**Recommended Dishes for Eating Outside**
Tuesday	French Spiced Bread Quick Sesame Chicken with Broccoli

Table 15 The Result of Eating Outside Model – Stage 2

5. Conclusion

In conclusion, keeping a healthy diet is important to nowadays young people. Our model helps to build a simple cooking and eating recommendation for a group of students according to their personal preference and nutrition requirements. In addition, this model helps a group of students monitor each other to keep eating healthy. In the future, we will keep perfecting our model and help more and more young people to keep a healthy diet.

6. Creative Contribution

Qianran Ma:
I suggested that we can restrict the number of each dish to make our meal plan more balanced and diverse. I also advised to set the constraints about the range of the cooking times of one person per week to make the cooking plan more reasonable and equitable.

Wanning Li:
My group member found out the basic model was infeasible sometimes. We talked about the feasibility of eating outside for recommendations.

Yanan Zhou:
I came up with the idea about cooking ability matrix and allergy matrix to realize these two constraints in a more flexible way.
Also, after discussing with my team members, I added constraints to the number of dishes we eat every meal and the total times we eat the same dish.
I noticed if no one is available during one meal, the model will be infeasible. To solve this problem, I discussed with my team members about a two-stage model and implemented it.

Yuxuan Cheng:
I suggested that, in the real-life meal planning problem, people's interests in dishes may change over time and they could have a dynamic rating system. Teammates are good at different type of cooking. Texture of a same dish may vary when it comes from different cook or in different season. The rating should evolve overtime and refer to teammates' skills. I suggested we could also design a self-adaption model to evolve the menu over time.

Project 3

Wyndor Advertising Problem

Content

1. Wyndor Problem

In the Wyndor problem, there are two types of doors produced by three plants with time limits in Figure 9. The doors have either aluminum frames (A) or wood frames (B). The plants are named 1, 2 and 3.

Plant	Prod. time A (Hours/Batch)	Prod. time B (Hours/Batch)	Total Hours
1	1	0	4
2	0	2	12
3	3	2	18
Profit	\$3,000	\$5,000	

Figure 9 Production data

The company now wants to optimize the profit by using advertising. The sales are needed when making advertising strategy. However, sales information is uncertain and also depends on the advertising strategy. The company uses TV and radio as two different advertising media. In this condition, sales are predicted based on the time slots spent on TV and radio.

	TV	Radio	Sales
1	230.1	37.8	22.1
2	44.5	39.3	10.4
3	17.2	45.9	9.3
...	...	...	...
200	232.1	8.6	13.4

Figure 10 Advertising data

The dataset has 200 data points. The total budget of advertising is \$200,000. Let x_1 and x_2 denote the TV and Radio expenditures. The total expenditures should be less than \$200 (in thousand). One more policy is that the TV expenditures should be as least half of the Radio expenditures. Finally, we have lower and upper limits for x_1 and x_2, observed from training set. Assuming the adverting costs are $c_1 = \$100$ and $c_2 = \$500$, the first stage model can be written as following.

$$Max - 0.1x_1 - 0.5x_2 + E[Profit(\tilde{w})]$$

$$s.t. x_1 + x_2 \leq 200$$
$$x_1 - 0.5x_2 \geq 0$$
$$L_1 \leq x_1 \leq U_1$$
$$L_2 \leq x_2 \leq U_2$$

The sales $\widetilde{w}$ are random variables influenced by the advertising strategies x_1 and x_2. We choose linear regression to measure it in this project. Therefore, the total sales are denoted as $\widetilde{w} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \tilde{\varepsilon}$ and this equality is added as a constraint in the second-stage production problem. In this equation, we can say the profit is determined by the choice x and the uncertainty captured by the random variable $\tilde{\varepsilon}$.

For each specific w, we have following second stage model. y_A and y_B mean the batch of door A and door B produced. The first three constraints are the time limits in Figure 1. In this illustrative example, the "Total Hours" are multiplied by two to allow more options for product mix decisions. One more important change is that the product needs to satisfy an additional constraint about the potential future sales. The production should be less or equal to the sales.

$$Profit(w) = Max\ 3y_A + 5y_B$$
$$s.t. y_A \leq 8$$
$$2y_B \leq 24$$
$$3y_A + 2y_B \leq 36$$
$$y_A + y_B \leq w$$
$$y_A \geq 0$$
$$y_B \geq 0$$

Since the interface between two stages is $\widetilde{w}$, it's important to reconcile the assumptions of linear regression underlying statistical estimation and optimization.

1. Linearity: Linear model is sufficient.
2. Independence: The error terms in regression model are independent random variables. Each data points represents independent clients.
3. Normality: The error is approximately normally distributed.
4. Homoscedasticity: The variance of errors doesn't change with the choice of x.

2. Learning Enabled Optimization

To solve two-stage problem, this project integrates optimization with statistical learning, which uses training and validation sets. The entire workflow is referred as Learning Enabled Optimization (LEO) in Figure 3. We would utilize this model for Wyndor advertising problem.

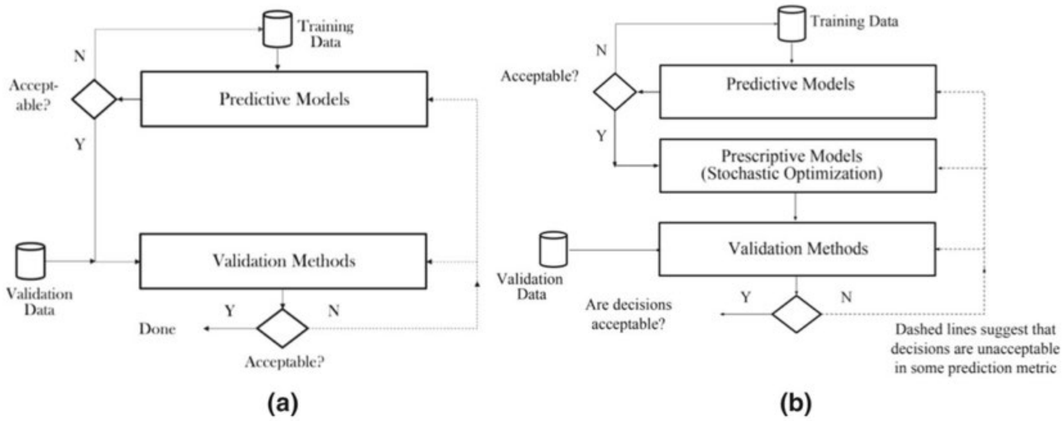


Figure 11 (a) statistical learning and (b) learning enabled optimization

Based on LEO, we implement three steps to solve Wyndor problem.

1. Descriptive Analytics:
    a. Split data into 50% training and 50% validation set.
    b. Build linear regression model with training set and obtain the linear coefficients.
    c. Calculate training error ε_{ti} and validation error ε_{vi} where $\varepsilon_{ti} = w_i - \beta_0 - \beta_1 x_{1i} - \beta_2 x_{2i}$ in training set and $\varepsilon_{vi} = w_i - \beta_0 - \beta_1 x_{1i} - \beta_2 x_{2i}$ in validation set.
2. Prescriptive Analytics:
    a. Build the all-in-one Sample Average Approximation (SAA) or Stochastic Decomposition (SD) with training error as the uncertainty.
    b. Solve the model and report $\hat{x}_1$, $\hat{x}_2$ and Model Predicted Objective (MPO).
3. Validation:
    a. Do F test between ε_{ti} and ε_{vi} to detect if two variables come from the same distribution.
    b. Plot the Q-Q graph of ε_{ti} and ε_{vi} to check if they are normally distributed.
    c. Use validation errors,
        i. Calculate $w_i = \beta_0 + \beta_1 \hat{x}_1 + \beta_2 \hat{x}_2 + \varepsilon_{vi}$ for each validation error.
        ii. Calculate $Profit(w_i)$ for each w_i by second stage production model.
        iii. Calculate $-0.1\hat{x}_1 - 0.5\hat{x}_2 + Profit(w_i)$ for each $Profit(w_i)$.
        iv. Calculate middle point and 95% confidence interval to report Model Sample Average Estimate (MVSAE) and confidence interval.

Below is the workflow of analyzing process.

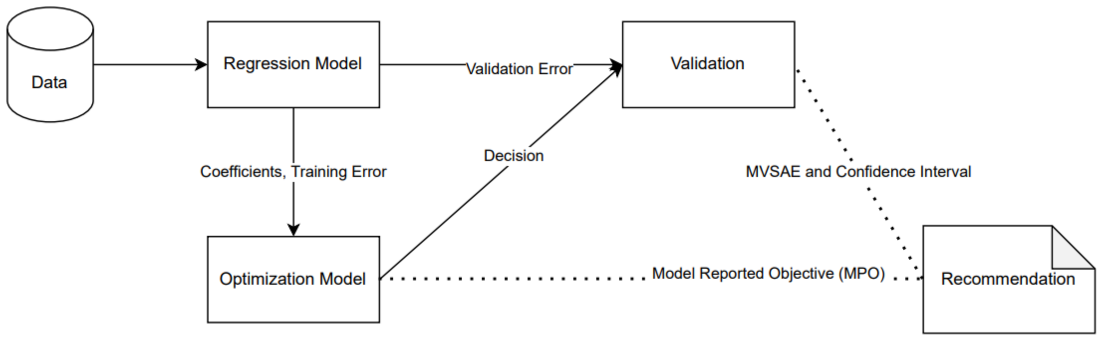


Figure 12 Workflow

3. Descriptive Analytics

Specifically, we use first 100 data points for training and remaining 100 data points for validation. We build linear regression by Python only with training set. The coefficients are $\beta_0 = 3.15$, $\beta_1 = 0.046$ and $\beta_2 = 0.184$. The training errors and validation errors are then calculated. The related code is referenced at the end.

4. Prescriptive Analytics

4.1 All-in-one SAA

Indeed, the model stated before can be formally classified as a Stochastic Linear Programming (SLP) problem. SLP allows discrete random variables to represent data uncertainty in an optimization model. The distribution is measured by the empirical distribution. We replace the expectation operator E with a finite representation of the average objective $\sum_{i=1}^{N} p_i Profit(w_i)$ where $0 \leq p_i \leq 1$ and $\sum_{i=1}^{N} p_i = 1$. We set $p_i = \frac{1}{N}$ where N denotes the number of training data points.

β_0, β_1 and β_2 are measured from the training set, we can denote w as following for each data point. ε_{ti} is the training error.

$$w_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \varepsilon_{ti}$$

A statistical approximation of the SLP, named as Sample Average Approximation, can be stated as following.

$$Max - 0.1x_1 - 0.5x_2 + \frac{1}{N}\sum_{i=1}^{N} Profit(w_i)$$
$$s.t. x_1 + x_2 \le 200$$
$$x_1 - 0.5x_2 \ge 0$$
$$L_1 \le x_1 \le U_1$$
$$L_2 \le x_2 \le U_2$$

$$Profit(w_i) = Max\ 3y_A + 5y_B$$
$$s.t. y_A \le 8$$
$$2y_B \le 24$$
$$3y_A + 2y_B \le 36$$
$$y_A + y_B \le \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \varepsilon_{ti}, \forall i$$
$$y_A \ge 0$$
$$y_B \ge 0$$

Every data point i can choose its own production level (y_{Ai}, y_{Bi}) because $Profit(w_i)$ is solved for each i. Now we can integrate two model as following.

$$Max - 0.1x_1 - 0.5x_2 + \frac{1}{N}\sum_{i=1}^{N} Profit(3y_{Ai} + 5y_{Bi})$$
$$s.t. x_1 + x_2 \le 200$$
$$x_1 - 0.5x_2 \ge 0$$
$$y_{Ai} \le 8, i = 1, \cdots, N$$
$$2y_{Bi} \le 24, i = 1, \cdots, N$$
$$3y_{Ai} + 2y_{Bi} \le 36, i = 1, \cdots, N$$
$$-\beta_1 x_1 - \beta_2 x_2 + y_{Ai} + y_{Bi} \le \beta_0 + \varepsilon_{ti}, i = 1, \cdots, N$$
$$L_1 \le x_1 \le U_1$$
$$L_2 \le x_2 \le U_2$$
$$y_{Ai} \ge 0$$
$$y_{Bi} \ge 0$$

With $N = 100$, we have 202 variables and hundreds of constraints, which is within the capacity of LP software. After solving the model, we get $\hat{x}_1 = 184.83$, $\hat{x}_2 = 15.17$, $MPO = 40.855$ (all in thousands). We will illustrate with validation results later. The code is referenced at the end.

4.2 Stochastic Decomposition

Another way is Stochastic Decomposition (SD). In statistical learning, using replications is a good way to reduce variance. SD automatically uses the replications in SLP. There are some assumptions needed to be clarified about SD.

1. Boundedness: In the advertising problem, we obtained the lower and upper limits from training and validation set and set those as boundaries.
2. Fixed Recourse: Random elements of the recourse problem only appear on the right-hand side.
3. Relatively Complete Recourse: Recourse problem is feasible for all values of (x, ε).
4. Optimistic Lower Bound: In minimization problems, the second stage is bounded below 0. And "Max" problem can be transposed to "Min" problem.

SD is a sequential sampling algorithm based on sampling from the training data set and drawing data points repeatedly. We solve the model using Julia with TwoSD. Here is the code.

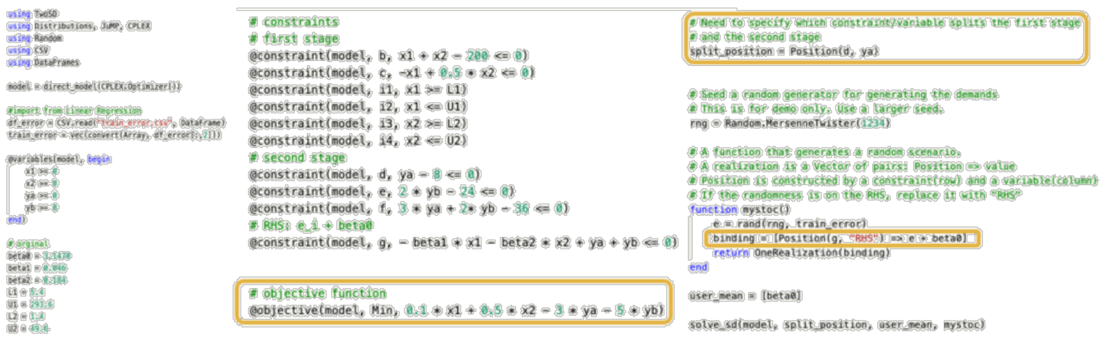


Figure 13 SD code

There are three parts needed to be emphasized.

1. Convert 'Max' problem to 'Min' problem.
2. Specify the position to split the first and second stages constraints.
3. Replace β_0 with $\beta_0 + e_{ti}$ each time to introduce randomness.

After solving the model, we get $\hat{x}_1 = 185.02$, $\hat{x}_2 = 14.98$, $MPO = 41.024$ (all in thousands). Since SD has replications, we could also report the confidence interval of MPO, which is [40.743, 41.305] (in thousands). We will illustrate with validation results later.

5. Validation

5.1 F test

The p-value of F test is 0.93, larger than 0.05. We do not reject the null hypothesis. Training errors and validation errors are from the same distribution.

5.2 Q-Q plot

Training errors and validation errors are not normally distributed based on the Q-Q plot. We will further delete the outliers to solve this issue.

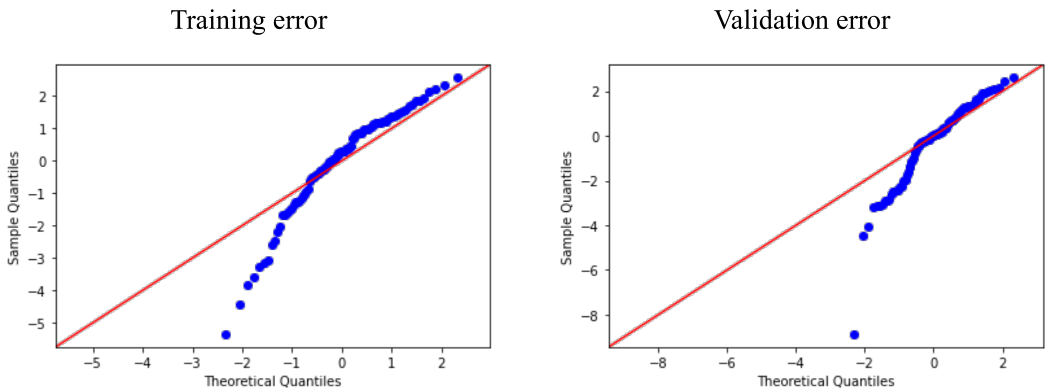


Figure 14 Q-Q plot with outliers

5.3 Results of validation errors

From the steps of workflow, we calculate $-0.1\hat{x}_1 - 0.5\hat{x}_2 + Profit(w_i)$ for each validation error and report MVSAE and confidence interval. In the end, the whole optimization results are as following.

	x_1	x_2	MPO ($)	MVSAE ($)
SAA	184.83	15.17	40855	39840 [38604, 41076]
SD	185.02	14.98	41024 [40743, 41305]	40681 [38598, 41078]

Table 16 Linear regression results with outliers

SD and SAA have similar advertising choice. Both algorithms predict accurately to make MPO fall within the MVSAE. The code is referenced at the end. However, since the errors are not normally distributed, we need to refine the algorithms and then compare the MPO and MVSAE.

6. Creative Ideas

6.1 Delete Outliers

One problem we encountered is that the dataset has outliers. Therefore, we train the regression model first with the **whole** data set and calculate errors. We then calculate the inter quantile range (IQR), lower and upper bounds. The error falling outside of the lower-upper range is deleted. Then we split the new dataset into 50% training and 50% validation sets, **retrain** linear regression with training set and calculate the training and validation errors with the linear coefficients.

$$IQR = Q3 - Q1$$
$$Lower = Q1 - 1.5 * IQR$$
$$Upper = Q3 + 1.5 * IQR$$

After deleting the outliers, the new training and validation sets both have 85 data points. We repeat the LEO workflow. The new p-value is 0.53, larger than 0.05. Training and validation errors are from the same distribution. New Q-Q plots are as following. The errors are now normally distributed.

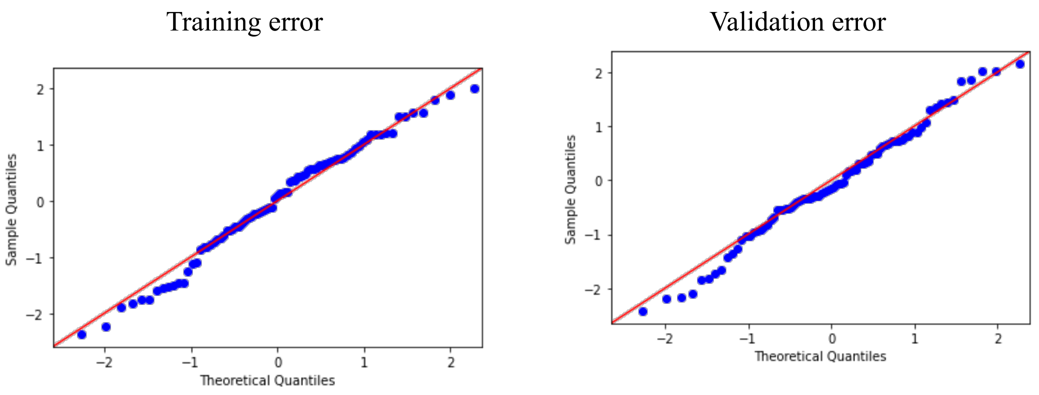


Figure 15 Q-Q plot without outliers

The optimization results are as following.

	x_1	x_2	MPO ($)	MVSAE ($)
SAA with outliers	184.83	15.17	40855	39840 [38604, 41076]
SAA without outliers	186.94	13.06	42060	41819 [41183, 42456]
SD with outliers	185.02	14.98	41024 [40743, 41305]	40681 [38598, 41078]
SD without outliers	186.62	13.38	42001 [41719, 42283]	41819 [41186, 42453]

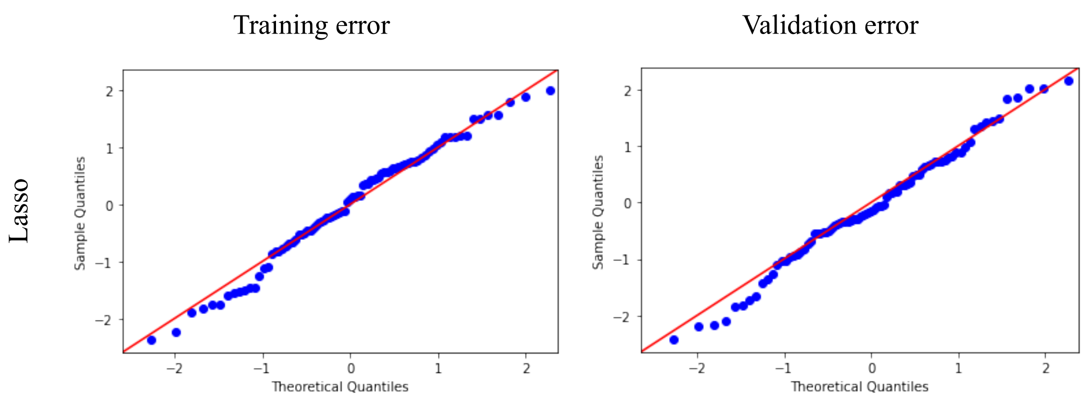
Table 17 Linear regression results without outliers

Either with or without outliers, MPO both reside within MVSAE. The important change is that the difference between MPO and MVSAE is smaller, which indicates the optimal result of training set is more reasonable. After deleting outliers, SAA and SD choose more TV advertising and have higher MPO and MVSAE. With less variance, the algorithm is more positive about the total profit.

6.2 Lasso and Ridge

We add two more regression methods to capture the relationship between demands and advertising choice. They are ridge and lasso regression. The mean difference is that they add penalties when minimizing the squared error. We set the range of penalty parameter as $(10^{-8}, 10^{5})$ and use the cross validation to choose the optimal parameter.

The p-value of lasso and ridge are 0.53 and 0.55. The Q-Q plots are as following. Training errors and validation errors are from the same normal distribution.



Ridge

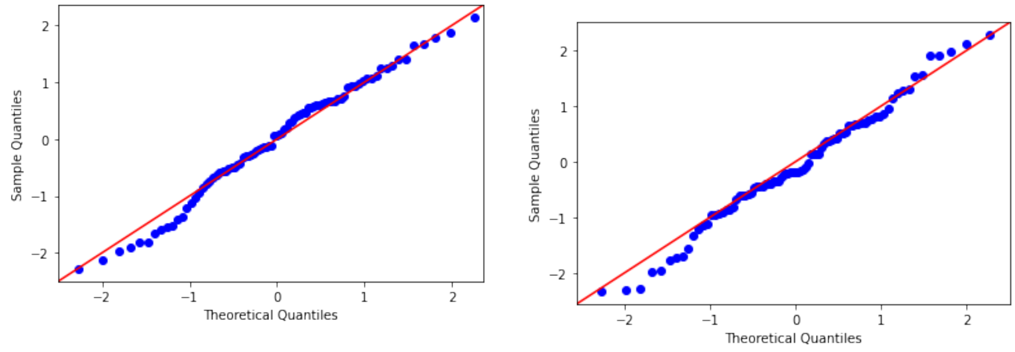


Figure 16 Lasso and ridge Q-Q plot without outliers

The new results are as following.

	x_1	x_2	MPO ($)	MVSAE ($)
SAA Linear Regression	186.94	13.06	42060	41819 [41183, 42456]
SAA Lasso	186.94	13.06	42060	41819 [41183, 42456]
SAA Ridge	189.19	10.81	42309	42034 [41377, 42691]
SD Linear Regression	186.62	13.38	42001 [41719, 42283]	41819 [41186, 42453]
SD Lasso	186.62	13.38	42004 [41721, 42286]	41819 [41186, 42453]
SD Ridge	189.82	10.18	42293 [41015, 42572]	42039 [41373, 42705]

Table 18 Lasso and ridge results without outliers

For SAA and SD, the results of linear regression and lasso are similar. The result of ridge is slightly larger. Therefore, adding L1 penalty is not effective but adding L2 penalty influences the advertising choice.

6.3 Different Split

The last comparison is the results of different split methods. We add an additional random split method to split the data set.

The p-value of random split are 0.22. The Q-Q plots are as following. Training errors and validation errors are from the same normal distribution.

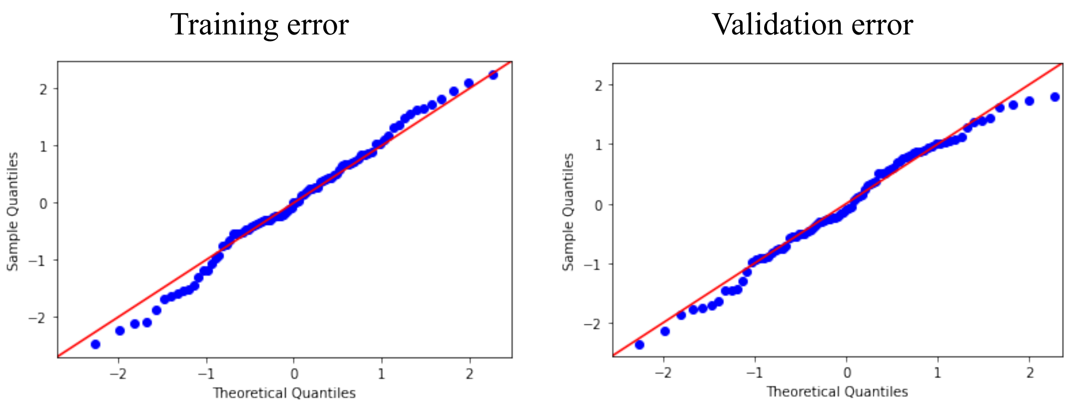


Figure 17 Random split Q-Q plot without outliers

The new results are as following.

	x_1	x_2	MPO ($)	MVSAE ($)
SAA original	186.94	13.06	42060	41819 [41183, 42456]
SAA random	186.01	13.99	41829	41804 [41181, 42427]
SD original	186.62	13.38	42001 [41719, 42283]	41819 [41186, 42453]
SD Random	186.54	13.45	41580 [41299, 41860]	41792 [41165, 42419]

Table 19 Random split method without outliers (linear regression)

The influence of choosing different split is not large. In this case, the estimation of profit with random split is smaller.

7. Conclusion

Without creative results, SAA and SD spend around $185,000 and $15,000 on TV and radio advertising. The training profit is around $41,000 and the validation profit is around $40,000. But the training errors and validation errors are not normally distributed. The performance of algorithms can be improved.

Compared with creative results, deleting outliers and ridge regression influence the advertising choice and profit. Lasso regression and different split does not have obvious influence. The new results show that SAA and SD spend around $186,000-$187,000 and $13,000-$14,000 on TV and radio advertising. The training and validation profit are around $42,000.

There are two changes. First, the profit increases. The algorithm is more positive when data has less variance. Second, the difference between MPO and MVSAE decreases. The algorithm is more reasonable because training result represents the testing result.

8. Creative Contribution

Qianran Ma:
Based on Stochastic Decomposition algorithm, I wrote the code and input the data without outliers and the parameters we got from Lasso and Ridge model to get the final result.

Wanning Li:
The data has outliers. I utilized inter quantile range to delete the outliers. With TA's advice, I used the whole dataset to fit the linear model instead of fitting with training and validation sets separately. I also compared the results with lasso/ridge and different splits. I prepared the relative data, and the SAA/SD results were finished with group members together.

Yanan Zhou:
I helped finished the implementation of Stochastic Decomposition Algorithm by generating training error randomly and distinguishing SD with Bender. Also, I put forward the idea that we could run a loop to try different train-test-splitting methods and compared the differences between groups.

Yuxuan Cheng:
To improve the accuracy, I tuned parameters of Lasso/Ridge regression models to best fit the data, and delete outliers based on the Lasso and ridge regression results. I did the study of among regressions to see the difference in model performance. I also helped my teammate to realize the primal model deleting outliers.

Project 4

Towards a sustainable power grid:

Stochastic hierarchical planning for high renewable integration

Content

1. Introduction

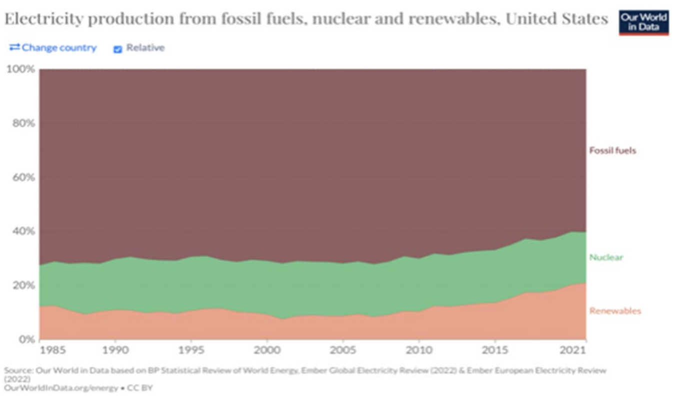
In Project 4, we studied stochastic hierarchical planning for considering high renewable integration in electricity generation. Different from the project we have done before, in project 4, we studied how layers of model make different decisions and higher layer decisions, lower layer decisions, former decisions influence each other to approach the best solution.

During the project, we firstly studied the background of the paper: planning for high renewable integration and found out the real time application of electricity generation decisions in the real-life problem. We followed up the former studies in this area and understand the considerations of author to choose stochastic hierarchical planning. Then, we studied the stochastic hierarchical planning model and figured out the difference between deterministic hierarchical planning and stochastic hierarchical planning. We followed the experimental study result of the paper and find the benefits of using stochastic hierarchical planning model. Finally, we discussed the future creative ideas to improve the model.

We studied the meaning of each constraint based on the guide in appendix a, with our understanding towards the paper and some basic knowledge of power generation. We made some more detailed explanation to objective functions and constrains to fully understand all the models.

1.1 Background

Many countries are trying to make renewable resources meet a significant percentage of the electricity supply. Recently the most popular resource for the electricity production in the US is still fossil fuels. There is a growing trend in the electricity production by renewable resources these years.



State and local authorities such us Independent System Operators（ISO）in US and

Transmission System Operators (TSO) in Europe have commissioned studies to assess operational considerations such as system reliability, market design, incorporation of storage technologies, and other avenues.

1.2 Problem

Renewable sources have a high level of uncertainty. Every year as the renewable-integration level increases, this uncertainty become severe. A recent simulation study, commissioned by California ISO(CAISO), suggests that for renewable-integration levels beyond 33, one can expect a fair amount of over-generation and renewable curtailment during the daytime and load-shedding around sun-down. Higher levels of renewable integration exacerbate these issues.

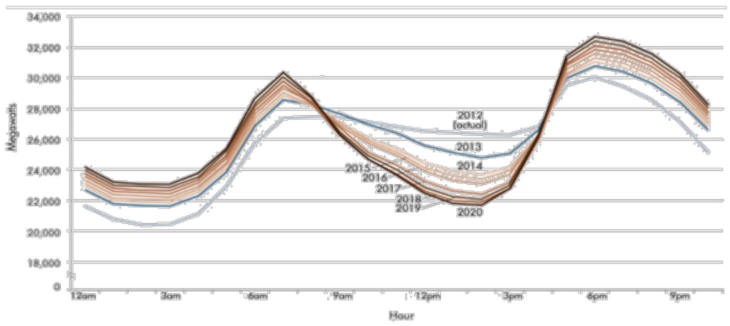


Figure 18 CAISO’s duck chart, predicting four emerging ramping patterns with increased renewable integration California ISO (2016).

The current practice of operation planning with deterministic models may be ill-suited for a future abundant uncertainty. Some ISOs (e.g., New England ISO) have already recognized the shortcomings of deterministic planning within the context of renewable integration.

1.3 Outlook for Solutions

To overcome the challenges of renewable integration, U.S. Department of Energy showed four specific rules:

For the operation planning:

1. Power generation and transmission capacity must be sufficient to meet the peak demand for electricity

2. Power systems must have adequate flexibility to address variability in demand and generation resources

For the operations control:

3. Power systems must be able to maintain a steady frequency
4. Maintain a steady voltage at various points on the grid

Therefore, plans based on deterministic models that use point forecasts as input will result in immense reliability challenges.

2. Literature review and creative contribution

2.1 Literature Review

According to the literature review in the paper, we summarized the topics of former study. Previous papers were focusing on using deterministic Unit Commitment and Economic Dispatch and solve it as a large-scale deterministic problem. There were also studies on the electric market.

2.2 Contributions of this paper

While there are many studies of stochastic optimization within any one layer of the hierarchical system, no other study pits the standard hierarchy of deterministic models against a hierarchy of stochastic ones in a system operated by a centralized planner. It is such a comparison that provides a preview of the potential advantages and disadvantages of alternative hierarchies on electricity production planning.

Compared to others, this paper involved a large collection of scenarios. The overarching goal of this paper is Comparing deterministic and stochastic Hierarchical Planning. A decision evolution architecture that draws conclusions regarding costs, GHG emissions, Load shedding etc.

Author performed a comprehensive computational study of SHP under different models (deterministic v. stochastic) at individual layers of the hierarchy acted as a precursor to an ISO-scale study and found out the potential of SHP when compared to the current DHP strategy. This model also acts as a precursor to an ISO-scale study

3. Hierarchical Planning

3.1 Why hierarchical planning?

Electric power system are very large-scale networks interconnecting many sources of electric power to points of consumption. The overarching goal of the system operators is to minimize costs while ensuring reliable power delivery to customers.

Based on the background of electric generation planning, electric power system are very large-scale networks interconnecting many sources of electric power to points of consumption. The nature of the generator operation is large-scale and interconnecting.

Given this continuous and large-scale nature, the system operators use a reformulation involving a hierarchy of optimization models defined over over-lapping horizons with different time resolutions for decisions and constraints. Therefore, system operators use a reformulation involving a hierarchy of optimization models defined over overlapping horizons with different time resolutions for decisions and constraints. Our hierarchical planning paradigm assumes coordination via a central planning authority (e.g., ISO or TSO). Most ISOs/TSOs currently implement some form of a DHP framework, which divide the daily planning activities into three principal layers:

	Full name	Description	Horizon	Resolution	Solution Frequency
DA-UC	Day-Ahead unit Commitment	Forecast daily load and generation limits leading to a production and transmission plan	24 hours	60 minutes	24 hours
ST-UC	Short-Term unit Commitment	Produce some commitment decisions and update transmission plans	4 hours	15 minutes	3 hours
HA-ED	Real-Time Economic Dispatch	Finalize the production and transmission plans and commit necessary reserve capacities	75 minutes	15 minutes	15 minutes

Table 20 DHP framework

DA-UC layer mainly makes a daily forecast of load and generation limits to provide a general and fixed plan which cannot be adjusted under emergency. It is formulated over a 24-hour planning horizon with decisions and constraints specified over an hourly resolution. ST-UC layer allows some adaptive commitment decisions with a horizon of 4 hours and a resolution of 15 minutes. HA-ED layer finalizes the production and transmission plans with 15-minute resolution units to respond instantaneously to system imbalances.

For each time point of the day, it has a recent DA-UC, ST-UC, and HA-ED results.

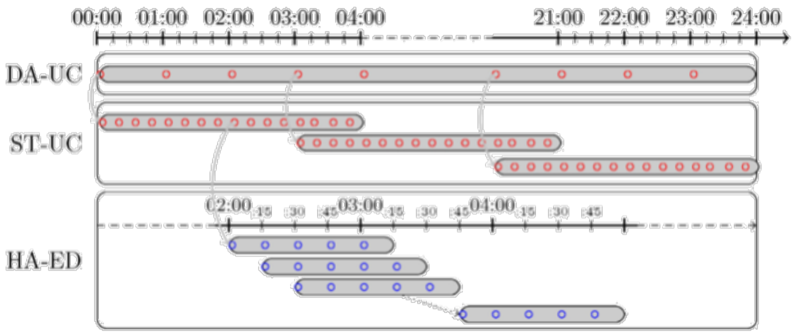


Figure 19 Hierarchical structure timescales of operating framework

3.2 Stochastic Hierarchical Planning

3.2.1 Decision Variables Settings

In this model, x and y are the decision vectors that model commitment and dispatch decisions.

Generator commitment decisions are often modeled using three sets of binary variables that indicates whether generator g is operational, turned on, and turned off in period t respectively. We denote by G the set of all generators in the system. This set includes the both the conventional generators Gc and renewable generators Gr. Furthermore, we assume that only a subset of conventional generators (typically gas-fired) are capable of providing fast-start services. We denote this set of generators by Gsc ⊂ Gc.

$$\mathbf{x}_{[t]} = (x_{g\tau}, s_{g\tau}, z_{g\tau})_{\forall g \in \mathcal{G}, \tau \in [t]}$$

s_{gt}: 1 if $g \in \mathcal{G}$ is turned on in $\tau \in [t]$, 0 otherwise.

x_{gt}: 1 if $g \in \mathcal{G}$ is operational in $\tau \in [t]$, 0 otherwise.

z_{gt}: 1 if $g \in \mathcal{G}$ is turned off in $\tau \in [t]$, 0 otherwise.

For the dispatch decisions. We use two sets of variables to model generation levels. G+ denotes the amount of electricity produced by every generator g in G during period t and consumed by grid. G- have two different meanings, that are overgeneration amount for conventional generator and renewable curtailment.

$$\mathbf{y}_{[t]} = ((G^{\pm}_{g\tau},)_{\forall g\in\mathcal{G}}, (F_{ij\tau})_{\forall(i,j)\in\mathcal{L}}, (D^{\text{shed}}_{i\tau}, R_{i\tau}, \theta_{i\tau})_{\forall i\in\mathcal{B}}))_{\forall\tau\in[t]}.$$

G^{+}_{gt}: generation amount of $g \in \mathcal{G}$, in $\tau \in [t]$, which is consumed by the grid.

G^{-}_{gt}: over-generation amount by $g \in \mathcal{G}_c$, in $\tau \in [t]$.

G^{-}_{gt}: renewable curtailment in $g \in \mathcal{G}_r$, in $\tau \in [t]$.

$F_{ij,t}$: electricity flow through $(i,j) \in \mathcal{L}$, in $\tau \in [t]$.

$\theta_{j\tau}$: voltage angle at $j \in \mathcal{B}$, in $\tau \in [t]$.

$D^{\text{shed}}_{j\tau}$: amount of unmet load at $j \in \mathcal{B}$, in $\tau \in [t]$.

Electricity transmission is modeled using three sets of variables which represents the electricity flow, bus voltage angles and the amount of unmet demand. Their basic relationship is flow-balance equations.

3.2.2 Parameters and Model Settings

Sets	
$\mathcal{B}$: buses.	$\mathcal{G}_j$: generators located in bus $j \in \mathcal{B}$.
$\mathcal{L}$: transmission lines.	$\mathcal{G}^d_c$: Slow-ramp DA generators.
$\mathcal{G}$: generators.	$\mathcal{G}^s_c$: Fast-ramp ST generators.
$\mathcal{G}_r$: solar and wind generators.	$[t]$: time periods.
$\mathcal{G}_c$: conventional generators ($\mathcal{G}_c = \mathcal{G} \setminus \mathcal{G}_r$).	

Parameters	
$G^{\max}_g$: generation capacity of $g \in \mathcal{G}_c$.	c^s_g: start up cost of $g \in \mathcal{G}$.
$G^{\min}_g$: minimum generation requirement for $g \in \mathcal{G}_c$.	c^f_g: no-load cost of $g \in \mathcal{G}$ (i.e., the intercept of the cost curve).
$\Delta G^{\max}_g$: ramp up limit for $g \in \mathcal{G}_c$.	c^p_g: variable generation cost of $g \in \mathcal{G}$ (i.e., the slope of the cost curve).
$\Delta G^{\min}_g$: ramp down limit for $g \in \mathcal{G}_c$.	$\theta^{\max}_j$: upper bound on the voltage-angle at bus $j \in \mathcal{B}$.
UT_g: minimum uptime requirement of $g \in \mathcal{G}_c$.	$\theta^{\min}_j$: lower bound on the voltage-angle at bus $j \in \mathcal{B}$.
DT_g: minimum downtime requirement of $g \in \mathcal{G}_c$.	ϕ^o_g: penalty for over-generation by $g \in \mathcal{G}_c$.
B_{ij}: susceptance of arc $(i,j) \in \mathcal{L}$.	ϕ^c_g: penalty for renewable curtailment in $g \in \mathcal{G}_r$.
$D_{j\tau}$: load in bus $j \in \mathcal{B}$, in period $\tau \in [t]$.	ϕ^u_j: penalty for unmet demand in bus $j \in \mathcal{B}$.
$R_{j\tau}$: reserve requirement in bus $j \in \mathcal{B}$ and period $\tau \in [t]$.	
$F^{\max}_{ij}$: maximum permitted flow through arc $(i,j) \in \mathcal{L}$.	

Table 21 Model and Parameters settings

3.2.3 Stochastic Unit Commitment Model

The objective function of stochastic unit commitment model is:

$$\min \sum_{\tau\in[t]} \sum_{g\in\mathcal{G}^*_c} (c^s_g s_{g\tau} + c^f_g x_{g\tau}) + \mathbb{E}[ED(\mathbf{x}_{[t]}, \tilde{\xi}_{[t]})]$$

The constraints are:

$$x_{g\tau} - x_{g\tau-1} = s_{g\tau} - z_{g\tau}, \qquad \forall g \in \mathcal{G}_c^*,\ \tau \in [t]. \qquad \text{(A.1)}$$

$$\sum_{j=\tau-UT_g+1}^{\tau-1} s_{g\tau} \leq x_{g\tau}, \qquad \forall g \in \mathcal{G}_c^*,\ \tau \in [t], \qquad \text{(A.2)}$$

$$\sum_{j=\tau-DT_g}^{t} s_{g\tau} \leq 1 - x_{g\tau}, \qquad \forall g \in \mathcal{G}_c^*,\ \tau \in [t]. \qquad \text{(A.3)}$$

The objective function includes the startup, no-load cost, and the expectation of the cost of production. This is the overall cost of the whole process.

(A.1) shows how these variables are linked. For every generator g in G during time τ, the difference of operating decision x between this period and last period equals to the difference between the turn-on and turn-off decisions.

There are three possible results of the left-hand side. When in this time, we keep the operational decision same with last one, the left-hand side value equals to zero, which means we did not do any turn-on or turn-off, therefore the right-hand side should be zero. When the generator turns on at time τ, the left-hand side equals to 1. the turn-on decision equals to 1 and turn-off decision equals to 0, therefore the right-hand side equals to 1. When the generator turns off at time τ, the left-hand side equals to -1. the turn-on decision equals to 0 and turn-off decision equals to 1, therefore the right-hand side equals to -1.

(A.2) stands for the minimum uptime requirements of generators. For each generator, the total times of turn-on decisions in the period from uptime to the most recent decision should be less than or equal to x_{gr}. To be more specific, if the operational decision x is 1, which means operating, in the period from a certain uptime to most recent, the generator keeps turning off or only turn on once. If the operational decision x is 0, which means not operating, in the period from a certain uptime to most recent, the generator keeps turning off.

(A.3) stands for the minimum downtime requirements of generators. For each generator, the total times of turn-on decisions in the period from downtime to the most recent decision should be less than or equal to 1- x_{gr}. If the operational decision x is 1, which means operating, in the period from downtime to the end, the generator cannot be turned on anymore. If the operational decision x is zero, which means it is not operating, from a downtime to the end it can be turned on at most once.

3.2.4 Stochastic Economic Dispatch Model

The objective function of stochastic economic dispatch model is：

$$ED(\mathbf{x}_{[t]}, \xi^{\omega}_{[t]}) = \min \sum_{\tau \in [t]} \left(\sum_{g \in \mathcal{G}_c^*} c_g^v G_{g\tau}^+ + \sum_{j \in \mathcal{B}} \left(\sum_{g \in \mathcal{G}_j \cap \mathcal{G}_c^*} \phi_g^o G_{g\tau}^- + \sum_{g \in \mathcal{G}_j \cap \mathcal{G}_r} \phi_g^c G_{g\tau}^- + \phi_j^u D_{j\tau}^{\text{shed}} \right) \right)$$

The second-stage problem captures the dispatch corresponding to solar/wind availability under scenario $\tilde{\xi}$. The objective value stands for the total cost of all buses. The objective value is all the cost of electricity production for grid and total cost of each bus for curtailment, over production and unmet demand.

The constrains are:

$$G_g^{\min} x_{g\tau} \le G_{g\tau} \le G_g^{\max} x_{g\tau}, \qquad \forall g \in \mathcal{G}_c,\ \tau \in [t], \qquad \text{(A.4)}$$

$$-\Delta G_g^{\min} \le G_{g\tau} - G_{g\tau-1} \le \Delta G_g^{\max}, \qquad \forall g \in \mathcal{G}_c,\ \tau \in [t]. \qquad \text{(A.5)}$$

(A.4) is the generator capacities and minimum generation requirements. This ensures all conventional generators obey certain physical requirements for attaining feasible production schedules.

(A.5) is the ramping constraint that can be strengthened with binary variables to enhance the computational performance of MIP solvers.

$$\sum_{i \in \mathcal{B}:(i,j) \in \mathcal{L}} F_{ij,\tau} - \sum_{i \in \mathcal{B}:(j,i) \in \mathcal{L}} F_{ji,\tau} + \sum_{g \in \mathcal{G}_j} G_{g\tau}^+ + D_{j\tau}^{\text{shed}} = D_{j\tau} + R_{j\tau},\ j \in \mathcal{B},\ \tau \in [t]. \qquad \text{(A.6)}$$

(A.6) is the flow-balance equation using electricity flow involves reserve consideration, bus voltage angles, and the amount of unmet demand. For each bus, the sum of difference between inflow and outflow, all feasible demand and unmet demand should equal to the true demand and reserve consideration.

$$F_{ij,\tau} = B_{ij}(\theta_{i\tau} - \theta_{j\tau}), \qquad \forall (i,j) \in \mathcal{L}, \tau \in [t], \qquad \text{(A.7)}$$

$$\theta_j^{\min} \le \theta_{j\tau} \le \theta_j^{\max}, \qquad \forall j \in \mathcal{B},\ \tau \in [t]. \qquad \text{(A.8)}$$

(A.7) and (A.8) describes linear approximations of power-flows in the model in terms of the bus voltage-angles. This is a concept in electricity area which denotes the angle between the angles of the voltages at two different points (bus). The transfer of power between the two points of power system is proportional to the sine of this angle. They are also limited by these two constraints.

$$F_{ij}^{\min} \le F_{ij,\tau} \le F_{ij}^{\max}, \qquad \forall (i,j) \in \mathcal{L},\ \tau \in [t]. \qquad \text{(A.9)}$$

(A.9) stands for the transmission capacities.

$$G^{+}_{g\tau} + G^{-}_{g\tau} = \xi^{\omega}_{g\tau} x_{g\tau}, \qquad \forall g \in \mathcal{G}_r,\ \tau \in [t]. \qquad \text{(A.10)}$$

(A.10) includes the stochastic process corresponding to the solar and wind output. ξ^{ω}_{gr} denote the realization of solar/wind availability under scenario ω, for generator g ∈ Gr, in period τ ∈ [t]. This constraint implies that the amount of available renewable generation is either consumed or curtailed.

3.3 Difference between deterministic model and Stochastic model

The deterministic variants of the model instances defined above use only point forecast and the objective functions are defined as the costs associated with both unit commitment decisions and dispatch under such forecasts. In these models, we replace the expectation-valued objective functions with deterministic counterparts. The second stage is defined only for a single scenario as opposed to a set of scenarios in stochastic models.

3.4 The Multi-layer Framework

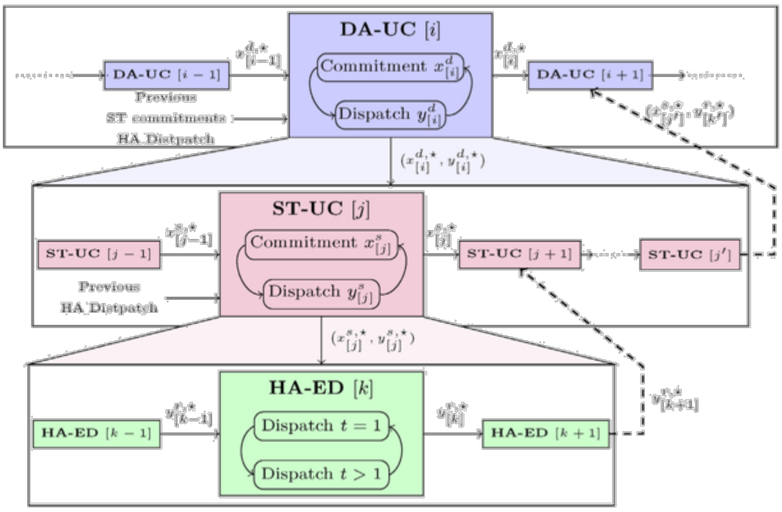


Fig. 3. Flow of decisions between layers of the hierarchical structure.

Figure 20 Multi-layer framework

Higher layer passes decisions to the lower layer, DA-UC commitment and dispatch decisions will be used in ST-UC, ST-UC commitment and dispatch decisions will be used in HA-ED.

Lower layer provides sampled objective values and sub gradients to the upper layer, the dispatch decision of HA-ED will influence next ST-UC, and decisions of ST-UC will influence next DA-UC.

In the model, these are realized by constrains for ST-UC second-stage and HA-ED.

$$x_{g\tau} = x_{g\tau}^{d,\star}; \quad s_{g\tau} = s_{g\tau}^{d,\star}; \quad z_{g\tau} = z_{g\tau}^{d,\star} \qquad \forall g \in \mathcal{G}_c^d, \tau \in [t]. \tag{A.12}$$

$$|G_{g\tau} - G_{g\tau}^{d,\star}| \le \epsilon_j \qquad \forall g \in \mathcal{G}_c^d,\ \tau \in [t]. \tag{A.13}$$

(A.12) Ensures the DA-UC commitments are honored in the ST-UC. (A.13) means the difference in the generation amounts is bounded.

$$x_{g\tau} = x_{g\tau}^{s,\star}; \quad s_{g\tau} = s_{g\tau}^{s,\star}; \quad z_{g\tau} = z_{g\tau}^{s,\star}, \tag{A.16a}$$

$$|G_{g\tau} - G_{g\tau}^{s,\star}| \le \epsilon_k \qquad \forall g \in \mathcal{G}_c, \tau \in [k]. \tag{A.16b}$$

(A.16) shows upstream commitment decisions are honored and the differences in the generation amounts are bounded.

4. Experimental study

The dataset in this paper is the NREL-118 instance includes 327 generators, 118 buses and 186 transmission lines, along with DA forecasts and RT outputs of renewable generators and demand. The experimental study part analyses the impact from three factors (solar and wind integration, reserve requirements, and the planning strategy) on three aspects: reliability, economic, and environmental metrics. And the paper considers a total planning horizon of 7 days.

In each result, we focused on the different types of resources, renewable resources integration, reserve requirements, and the planning setting.

4.1 Reliability Impact

The paper mainly discussed the reliability impact in unmet demand. It is important to reduce the blackouts with damaging economic consequence.

Average and maximum unmet demand amounts (Megawatt).

Planning setting	Reserve req.	Avg. unmet demand			Max. unmet demand		
		Solar & wind integ.			Solar & wind integ.		
		Low	Med.	High	Low	Med.	High
DDD	V. Low	1.9	4.4	17.0	320.0	595.1	956.1
	Low	0.5	3.0	1.5	246.9	336.5	274.3
	Med.	0.0	0.4	1.2	0.0	252.6	552.5
	High	0.0	0.0	0.0	0.0	0.0	0.0
DDS	V. Low	1.2	0.7	3.9	172.0	162.8	494.3
	Low	0.0	0.0	0.5	0.0	0.8	217.8
	Med.	0.0	0.0	0.0	0.0	0.0	0.0
	High	0.0	0.0	0.0	0.0	0.0	0.0
SDS	V. Low	0.0	1.9	0.6	19.2	182.0	356.6
	Low	0.0	0.0	0.0	0.0	0.0	0.0
	Med.	0.0	0.0	0.0	0.0	0.0	0.0
	High	0.0	0.0	0.0	0.0	0.0	0.0

Table 22

In the paper, author chose average and maximum unmet demand to study, and get following results:

1. More introduction of solar and wind resources results in a higher level of unmet demand.
2. More conservative reserve requirements reduce the unmet demand
3. The use of stochastic planning models contributes to zeroing out unmet demand, even at less conservative reserve requirements.

To analysis this result, we firstly notice it is reasonable for with the greater integration of renewable resources, the uncertainty is also greater. Therefore, the prediction might have more variance with the true scenario, which cause a higher level of unmet demand. If there are more reserve requirements, it allows the model to be more flexible and balance the demand through outflow and inflow. The stochastic model in this part performed better in reducing the unmet demand than the deterministic model, it suggests the possibility of a more economic manner of system operations.

We evaluate the reliance on ST-UC problems by looking at the percentage of time that these generators were active from Table 23.

Average percentage of time ST-UC generators were active.

Planning setting	Low SW	Med. SW	High SW
DDD	11.0	11.7	11.8
DDS	9.7	8.3	7.8
SDS	8.7	6.5	5.3

Table 23

We find out that the DDD setting heavily relies on ST-UC problems to maintain reliability comparing to DDS and SDS. All of three situations of solar and wind integration show deterministic model relies on ST-UC problems. In contrast, DDS and SDS settings substantially reduce these requirements even under higher renewable-integration settings.

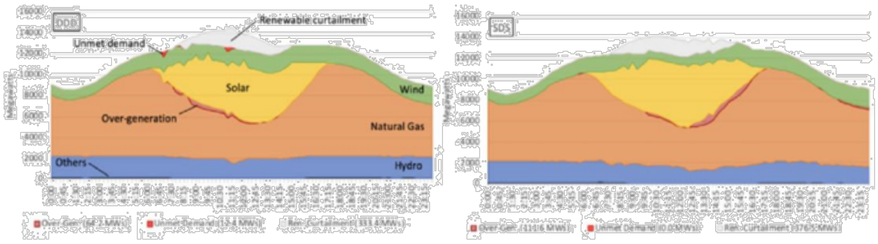


Fig. 6. Generation mix, unmet-demand, and over-generation in DDD and SDS settings (very low reserves, High SW, and a sample day).

Figure 21

Unmet demand, over-generation, and renewable curtailment may all occur simultaneously. On the right side, SDS leads to more over-generation but reduces unmet demand. Under SDS, hydro-based generation has a higher variability, which contributes to its better accommodation to uncertainty.

4.2 Economic Impact

Fig.7 demonstrates the average daily operating costs recorded in our experiments. The reported amounts exclude the potential costs associated with the consequences of operating a grid with low reliability

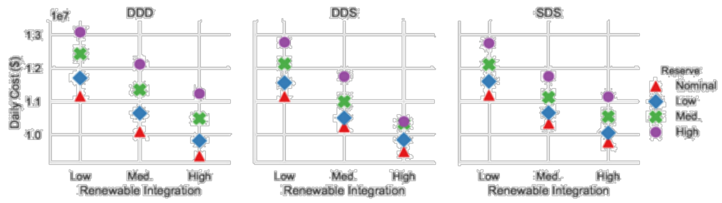


Fig. 7. Average daily generation cost of the power system under different reserve requirements and operations planning strategies.

Figure 22

We get following results:

1. Increased renewable integration leads to lower costs as these resources have negligible generation costs
2. Increased reserve requirements lead to higher costs since more resources need to be committed to maintain these requirements.

Average number of committed generators in each framework across different reserve requirement levels.

Reserve req:	V Low	Low	Med.	High
DDD	162.9	170.7	180.3	186.4
DDS	157.6	162.8	168.0	174.2
SDS	154.5	159.8	166.3	174.6

Table 24

The number of committed generators under DDS and SDS settings are consistently lower than that in DDD. Overall, the stochastic models can mitigate concerns over reliability without committing an abundance of resources.

4.3 Environmental Impact

Avg. daily operating cost of the system corresponding to the minimum reserve requirements leading to zero unmet demand (in million $; Reserve requirements in parenthesis).

	DDD	DDS	SDS
Low SW	12.42 (Med.)	11.56 (Low)	11.60 (Low)
Med. SW	12.11 (High)	11.00 (Med.)	10.66 (Low)
High SW	11.23 (High)	10.34 (Med.)	10.06 (Low)

Table 25

With stochastic optimization, reserve requirements can be relaxed as the models can dynamically adjust production levels by accounting for uncertainty in the future.

However, the operating costs of DDS is better than that of SDS under medium and high reserve requirements, and high renewable integration.

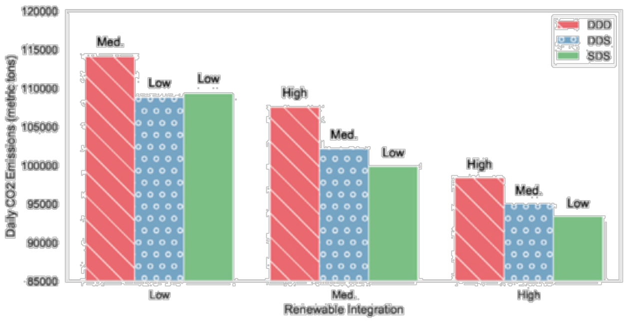


Figure 23

Higher renewable integration leads to lower levels of CO2 emissions. While stochastic modelling (i.e. DDS, SDS) also leads to over generation and renewable curtailment, their impact can largely be reversed by the lower reserve requirements necessary to achieve the same level of reliability.

5. Creative Ideas

Our group regards that it is plenitude in the model and considerations. The model can improve at two aspects.

5.1 Forecasts

The accuracy of weather information is important for the prediction and decision making. Reasonably accurate probabilistic forecasts of wind energy over short intervals of time.

5.2 Algorithm

We notice that, in the experimental study part, tests are made for three types of planning: DDD, DDS, SDS. The study does not include the following combinations: (•, S, •). Specialized (fast) algorithms for large-scale ST-UC problems which can deliver near optimal solutions for real-scale models with binary (start-up/shut- down) variables.

6. Creative Contribution

Qianran Ma:
Based on the paper, I discovered that there is a direction for the further improvement for the renewable integration in the model, which is the forecasts of wind energy over short intervals of time.

Wanning Li:
Based on paper, I suggested implement SSS model in the future if possible. Also, the model is highly related to the California situation. It can be transferred to implement in more places.

Yanan Zhou:
I came up with the idea that we can try other different combinations of stochastic and deterministic models to make the system more dynamic. Also, I put forward the idea that we could consider other different type of renewable energy.

Yuxuan Cheng:
I considered the possible aspects of model improvements in forecasts and algorithms. I also tried to explain some meanings of constraints that are not shown in paper.

References & Code

Reference

Project 1:

[1] Herer, Yale & Tzur, Michal & Ucesan, Enver. (2006). The multi-location transshipment problem. IIE Transactions. 38. 185-200.10.1080/07408170500434539.

[2] Higle, Julia & Sen, Suvrajeet. (1994). Finite master programs in regularized stochastic decomposition. Math. Program.. 67. 143-168.10.1007/BF01582219.

[3] Ruder, Sebastian. (2016). An overview of gradient descent optimization algorithms.

[4] Sutskever, I. & Martens, J. & Dahl, G. & Hinton, G.. (2013). On the importance of initialization and momentum in deep learning. 30th International Conference on Machine Learning, ICML 2013. 1139-1147.

[5] Warren B. Powell. 2007. Approximate Dynamic Programming: Solving the Curses of Dimensionality (Wiley Series in Probability and Statistics). Wiley-Interscience, USA.

[6] Yao, Yuan & Rosasco, Lorenzo & Caponnetto, Andrea. (2007). On Early Stopping in Gradient Descent Learning. Constructive Approximation. 26. 289-315. 10.1007/s00365-006-0663-2.

[7] Zhao, Lei & Sen, Suvrajeet. (2006). A Comparison of Sample-Path-Based Simulation-Optimization and Stochastic Decomposition for Multi-Location Transshipment Problems. Proceedings - Winter Simulation Conference. 238-245. 10.1145/1218112.1218161.

Project 2:

[1] Sheen, Spencer. (2016). A Coordinate Descent Method for Robust Matrix Factorization and Applications. SIAM Undergraduate Research Online. 9. 10.1137/15S014472.

[2] Yang, J. (2015). Notes on Low-rank Matrix Factorization. ArXiv, abs/1507.00333.

Project 3:

[1] Yunxiao Deng & Suvrajeet Sen, 2022. "Predictive stochastic programming," Computational Management Science, Springer, vol. 19(1), pages 65-98, January.

Project 4:

[1] Semih Atakan, Harsha Gangammanavar, Suvrajeet Sen,
Towards a sustainable power grid: Stochastic hierarchical planning for high renewable integration, European Journal of Operational Research, 2022, ISSN 0377-2217, https://doi.org/10.1016/j.ejor.2021.12.042.

Code

Project 1:

[https://colab.research.google.com/drive/19tZaQO3ovk6w-tYkCDiXZDkxYzj67WOh?authuser=1#scrollTo=RpSu8kZJ_QMV]

Project 2:

Data Preparation:

- Data Cleaning for Rating Matrix Factorization: [https://colab.research.google.com/drive/10JMQtIXo0Am7OxUF-tMn8auRw-tNczfm]
- Data Cleaning for Optimization Model: [https://drive.google.com/file/d/1OJIZLd6_r7oK2fQFwEcx3Ixm6B-FIuyz/view?usp=sharing]

Rating Matrix Factorization: [https://colab.research.google.com/drive/1u8NJRyN4ntNDLhTjUTLgiysuqlrzLLWb]

Optimization Model: [https://colab.research.google.com/drive/1btmTQc8HMjzdeAAbT3pf-OwYpG3olGnA#scrollTo=koc6MlNI424k]

Project 3:

[https://colab.research.google.com/drive/1Pizkg4ynN4SwY-pELzQGv_Ds4wZFTwLz?usp=sharing