Contents

Week 11: Abstract Formulation . . . 1
Session 21: Expressing Patterns using Mathematical Notation . . . 1
Understanding Summation Notation . . . 1
Examples of Abstract Formulation . . . 3
Exercise 11.1: Abstract Formulation for Supply Chain Planning . . . 7
Exercise 11.2: Abstract Formulation for Box Selection . . . 8
Exercise 11.3: Optimal Debt Repayment . . . 9
Session 22: Creating Abstract Formulations . . . 12
Tips for Creating Abstract Formulations . . . 12
Examples from Last Class . . . 12
Exercise 11.4: Assigning Consultants to Projects . . . 13
Exercise 11.5: Classroom Seating under Social Distancing . . . 14
Exercise 11.6: Supply Chain Planning Revisited . . . 16
Instructions for Quiz 4 . . . 17
Sample Quiz 4 . . . 18

Week 11: Abstract Formulation

By the end of this week, you will be able to use correct mathematical notation to represent large linear optimization models in a succinct and precise way. Learning to read and write abstract mathematical notation is an invaluable skill in all branches of analytics.

Session 21: Expressing Patterns using Mathematical Notation

Understanding Summation Notation

In mathematics, sums can be denoted using $\sum$.

Basic Example

```
[1]: from gurobipy import Model, GRB
     mod=Model()
     x=mod.addVars(range(1,11),name='x')
     mod.update()
     B_L=[1,4,5,9]
     sum(x[b] for b in B_L)
```

```
<gurobi.LinExpr: x[1] + x[4] + x[5] + x[9]>
```

Corresponding Math Notation:

Define B_L to be the set of literary books.

$$\sum_{b \in B_L} x_b$$

Latex code:

```
$$ \sum_{b \in B_L} x_b $$
```

Summing consecutive indices

```
[2]: sum(x[b] for b in range(3,10))
```

```
<gurobi.LinExpr: x[3] + x[4] + x[5] + x[6] + x[7] + x[8] + x[9]>
```

Corresponding Math Notation:

$$\sum_{b=3}^9 x_b$$

Latex code:

`$$\sum_{b=3}^9 x_b$$`

Summing multiple indices

[3]:
```
import pandas as pd
cost=pd.DataFrame([[20,18,21,8],[8,23,24,8],[25,8,8,19]],\
                  index=[1,2,3],columns=['A','B','C','D'])
FCs=cost.index
regions=cost.columns
y=mod.addVars(FCs,regions,name='y')
mod.update()
sum(cost.loc[f,r]*y[f,r] for f in FCs for r in regions )
```

```
<gurobi.LinExpr: 20.0 y[1,A] + 18.0 y[1,B] + 21.0 y[1,C] + 8.0 y[1,D] + 8.0 y[2,A] + 23.
↪0 y[2,B] + 24.0 y[2,C] + 8.0 y[2,D] + 25.0 y[3,A] + 8.0 y[3,B] + 8.0 y[3,C] + 19.0␣
↪y[3,D]>
```

Corresponding Math Notation:

Define F to be the set of FCs and R to be the set of regions.

$$\sum_{f \in F, r \in R} c_{fr}y_{fr}$$

Latex code:

`$$ \sum_{f \in F, r \in R} c_{fr}y_{fr} $$`

In-Class Exercise: Writing Out the Sum Explicitly

Expand the following summations into explicit sum. You can write directly on the handout and you do not need to submit anything on Blackboard.

Example:

$S = \{1, 3, 6, 8\}$

Expand:

$$\sum_{i \in S} x_i$$

Answer:

$$\sum_{i \in S} x_i = x_1 + x_3 + x_6 + x_8$$

a)

$B = \{1, 2, 3, 4, 5, 6\}$

Expand:

$$\sum_{j \in B} q_j x_j$$

Answer:

b)

$i = 3,\ J = \{2, 5, 8, 9\}$

Expand:

$$\sum_{j \in J} c_{ij} y_j$$

Answer:

c)

$j = 5$

Expand:

$$\sum_{i=2}^{j} a_{ij} x_i$$

Answer:

d)

$I = \{1, 2, 3\},\ J = \{2, 4\},\ k = 5.$

Expand:

$$\sum_{i \in I, j \in J} a_{ijk} x_{ij} y_{jk}$$

Answer:

Examples of Abstract Formulation

Example 1: Assortment Planning

[4]:
```
# Gurobi Code from Week 10
from gurobipy import Model, GRB
mod=Model()
books=range(1,11)
booksInGenre={'Literary':[1,4,5,9],\
              'Sci-Fi':[2,7,9],\
              'Romance':[3,4,6,10],\
```

```
                   'Thriller':[2,3,8]}
requirement={'Literary':2,'Sci-Fi':2,'Romance':2,'Thriller':2}
x=mod.addVars(books, vtype=GRB.BINARY, name='x')
mod.setObjective(sum(x[b] for b in books))
for genre in booksInGenre:
    mod.addConstr(sum(x[b] for b in booksInGenre[genre])>=requirement[genre],␣
↪name=genre)
mod.write('10-books.lp')
%cat 10-books.lp
```

```
\ LP format - for model browsing. Use MPS format to capture full model detail.
Minimize
  x[1] + x[2] + x[3] + x[4] + x[5] + x[6] + x[7] + x[8] + x[9] + x[10]
Subject To
 Literary: x[1] + x[4] + x[5] + x[9] >= 2
 Sci-Fi: x[2] + x[7] + x[9] >= 2
 Romance: x[3] + x[4] + x[6] + x[10] >= 2
 Thriller: x[2] + x[3] + x[8] >= 2
Bounds
Binaries
 x[1] x[2] x[3] x[4] x[5] x[6] x[7] x[8] x[9] x[10]
End
```

Abstract Formulation   Data:

- B: the set of books.
- G: the set of genres.
- B_g: the set of books of genre g.
- r_g: the number of books required for genre g.

Decision Variables:

- x_b: whether to carry book b. (Binary)

Objective and constraints:

$$\begin{aligned}
\text{Minimize:} && \sum_{b \in B} x_b \\
\text{subject to:} \\
\text{(Enough books in genre)} && \sum_{b \in B_g} x_b & \ge r_g & \text{ for each genre $g \in G$.}
\end{aligned}$$

Corresponding Latex

```
$$\begin{aligned}
\text{Minimize:} && \sum_{b \in B} x_b \\
\text{subject to:} \\
\text{(Enough books in genre)} && \sum_{b \in B_g} x_b & \ge r_g & \text{ for each genre $g \in G$.}
\end{aligned}$$
```

Example 2: Food Production

```
[5]: # Gurobi Code from Week 10
     import pandas as pd
     n=6
     s=1000
```

```python
months=range(1,n+1)
price=pd.Series([150,160,180,170,180,160],index=months)
demand=pd.Series([2000]*n, index=months)
mod=Model()
X=mod.addVars(months)
Y=mod.addVars(months,ub=s)
mod.setObjective(sum(price.loc[i]*X[i] for i in months))
for t in months:
    if t==1:
        mod.addConstr(Y[1]==X[1]-demand[1])
    else:
        mod.addConstr(Y[t]==X[t]+Y[t-1]-demand[t])
```

Abstract Formulation **Data:**

- n: number of months.
- T: set of months. $T = \{1, 2, 3, \cdots, n\}$.
- p_t: price of oil in month t.
- d_t: demand in month t.
- s: amount of oil that can be stored at any time.

Decision Variables:

- x_t: amount of oil to buy in month t. (Continuous)
- y_t: amount of oil stored at the end of month t. (Continuous)

Objective and Constraints:

$$\begin{aligned}
\text{Minimize: } && \sum_{t \in T} p_tx_t \\
\text{s.t. } && \\
&& y_1 &= x_1 - d_1 \\
&& y_t &= x_t + y_{t-1} - d_t && \text{for each month $t \in \{2,3,\cdots,n\}$.} \\
&& y_t &\le s && \text{for each month $t \in T$.} \\
&& x_t, y_t & \ge 0
\end{aligned}$$

Corresponding Latex

```
$$\begin{aligned}
\text{Minimize: } && \sum_{t \in T} p_tx_t \\
\text{s.t. } && \\
&& y_1 &= x_1 - d_1 \\
&& y_t &= x_t + y_{t-1} - d_t && \text{for each month $t \in \{2,3,\cdots,n\}$.} \\
&& y_t &\le s && \text{for each month $t \in T$.}  \\
&& x_t, y_t & \ge 0
\end{aligned}$$
```

Example 3: Project Selection

```python
[6]: # Gurobi Code from Week 10
projects=['A','B','C','D','E','F','G']
conflicts=[['A','B'],['B','C'],['A','C'],['A','D'],\
           ['D','E'],['E','F'],['F','G'],['E','G']]
prereqs=[['A','F'],['B','G']]
mod=Model()
```

```
x=mod.addVars(projects,vtype=GRB.BINARY)
mod.setObjective(sum(x[p] for p in projects),sense=GRB.MAXIMIZE)
for p1,p2 in conflicts:
    mod.addConstr(x[p1]+x[p2]<=1)
for p1,p2 in prereqs:
    mod.addConstr(x[p1]>=x[p2])
```

Abstract Formulation **Data:**

- P: set of projects
- C: set of conflicts. Each $(p_1, p_2) \in C$ is a pair of projects that conflicts with one another.
- R: set of prerequisite pairs. Each $(p_1, p_2) \in R$ is a pair such that project p_1 is a prerequisite to project p_2.

Decision Variables: x_p: whether to pursue project p. (Binary)

Objective and Constraints:

$$\begin{aligned}
\text{Maximize} && \sum_{p \in P} x_p \\
\text{s.t.} && \\
&& x_{p_1}+x_{p_2} &\le 1 && \text{For each conflicting pairs $(p_1,p_2) \in C$.} \\
&& x_{p_1} & \ge x_{p_2} && \text{For each pair (p_1,p_2) such that p_1 is a prereq to p_2.}
\end{aligned}$$

Corresponding Latex

```
$$\begin{aligned}
\text{Maximize} && \sum_{p \in P} x_p \\
\text{s.t.} && \\
&& x_{p_1}+x_{p_2} &\le 1 && \text{For each conflicting pairs $(p_1,p_2) \in C$.} \\
&& x_{p_2} & \le x_{p_1} && \text{For each pair $(p_1,p_2)$ such that $p_1$ is a prereq to $p_2$.}
\end{aligned}$$
```

Notes about Mathematical Notation

The following table compares Python code with the correponding mathematical notation.

Python code	Math notation
DataFrame entry: `a.loc[i,j]`	a_{ij}
Series entry: `a.loc[i]`	a_i
List/Dict entry: `a[i]`	a_i
List `L=[3,5]`	Set $S = \{3, 5\}$ or Tuple $t = (3, 5)$

In math, a set $\{\cdot\}$ does not encode the order of elements, so it does not make sense to refer to the position of the element (i.e. first element/second element). To refer to an element in the set, we use the notation $s \in S$ (i.e. element s in the set S).

A typle $(\cdot)$ does encode the order, so one can refer to the first element in the tuple t as t_1 and the second as t_2, etc.

Latex code: `$S=\{3,5\}$` `$s \in S$` `$t=(3,5)$` `$t_1$`

In Example 3 above, each pair (p_1, p_2) is a tuple, whereas P, C and R are sets.

Exercise 11.1: Abstract Formulation for Supply Chain Planning

Write the abstract formulation corresponding to the following Gurobi code from Week 10.

```
[7]: import pandas as pd
     cost=pd.DataFrame([[20,18,21,8],[8,23,24,8],[25,8,8,19]],\
                       index=[1,2,3],columns=['A','B','C','D'])
     demand=pd.Series([30,50,10,20],index=['A','B','C','D'])
     capacity=pd.Series([40]*3, index=[1,2,3])
     FCs=cost.index
     regions=cost.columns
     mod=Model()
     x=mod.addVars(FCs,regions,name='x')
     mod.setObjective(sum(cost.loc[f,r]*x[f,r] for f in FCs for r in regions))
     for f in FCs:
         mod.addConstr(sum(x[f,r] for r in regions)<=capacity[f],name=f'Capacity_{f}')
     for r in regions:
         mod.addConstr(sum(x[f,r] for f in FCs)>=demand[r],name=f'Demand_{r}')
     mod.write('10-supplyChain.lp')
     %cat 10-supplyChain.lp
```

```
\ LP format - for model browsing. Use MPS format to capture full model detail.
Minimize
  20 x[1,A] + 18 x[1,B] + 21 x[1,C] + 8 x[1,D] + 8 x[2,A] + 23 x[2,B]
   + 24 x[2,C] + 8 x[2,D] + 25 x[3,A] + 8 x[3,B] + 8 x[3,C] + 19 x[3,D]
Subject To
 Capacity_1: x[1,A] + x[1,B] + x[1,C] + x[1,D] <= 40
 Capacity_2: x[2,A] + x[2,B] + x[2,C] + x[2,D] <= 40
 Capacity_3: x[3,A] + x[3,B] + x[3,C] + x[3,D] <= 40
 Demand_A: x[1,A] + x[2,A] + x[3,A] >= 30
 Demand_B: x[1,B] + x[2,B] + x[3,B] >= 50
 Demand_C: x[1,C] + x[2,C] + x[3,C] >= 10
 Demand_D: x[1,D] + x[2,D] + x[3,D] >= 20
Bounds
End
```

Abstract Formulation

Data:

Decision Variables:

Objective:

Constraints:

Exercise 11.2: Abstract Formulation for Box Selection

Complete the abstract formulation corresponding to the following concrete formulation from Week 9.

Item type	1	2	3
Minimum box size (in cubit feet)	1.5	1.7	2.0
Demand	400	500	200

Concrete Formulation

Decision Variables:

- Y_1, Y_2, Y_3: how many boxes to make of each box type. (Integer)
- Z_1, Z_2, Z_3: whether to make the mold for each box type. (Binary)

Objective and Constraints:

$$
\begin{aligned}
\text{Minimize:}\quad & 1.5Y_1 + 1.7Y_2 + 2.0Y_3 + 1000(Z_1 + Z_2 + Z_3) \\
\text{s.t.}\quad & \\
\text{(Demand 1)}\quad & Y_1 + Y_2 + Y_3 \geq 1100 \\
\text{(Demand 2)}\quad & Y_2 + Y_3 \geq 700 \\
\text{(Demand 3)}\quad & Y_3 \geq 200 \\
\text{(S boxes on/off)}\quad & Y_1 \leq 1100Z_1 \\
\text{(M boxes on/off)}\quad & Y_2 \leq 1100Z_2 \\
\text{(L boxes on/off)}\quad & Y_3 \leq 1100Z_3 \\
& Y_1, Y_2, Y_3 \geq 0
\end{aligned}
$$

Abstract Formulation

Data:

Decision Variables:

Objective and Constraints:

Exercise 11.3: Optimal Debt Repayment

In this question, you practice everything learnt in the second half of the course thus far, as you will write an English description, a concrete formulation, an abstract formulation, as well as Python code.

Paris has come to you because she needs help paying off her credit card bills. Her statement at the beginning of month 1 shows the following balances:

Credit Card	Balance	Monthly Rate
Saks Fifth Avenue	\$20,000	0.5%
Bloomingdale's	\$50,000	1.0%
Macy's	\$40,000	1.5%

Paris has agreed not to shop at any of these stores anymore, and she is willing to allocate up to 5,000 dollars per month to pay off these credit cards. All cards must be paid off within 36 months (meaning that her statement at the beginning of month 37 must be zero for all card). Paris' goal is to minimize the total of all her payments.

For this problem, assume that the interest for the month is applied after the payment for that month. For example, suppose Paris pays 5,000 on Saks for month 1. Then her Saks balance at the beginning of month 2 is $(1.005) \times (20000 - 5000) = 15075$.

Help Paris solve her problem by formulating it into a linear optimization model, then generalize it to be able to handle arbitrary number of credit cards, balances, monthly rates, monthly budget, and time required for full payment.

Step 1. English Description

Describe the decision, objective and constraints in English using precise, complete and succinct language.

Decision:

Objective:

Constraints:

Step 2. Concrete Formulation

Write a linear optimization formulation in which the only variables are decision variables, and all input data are represented as numbers.

Decision variables:

Objective and constraints:

Step 3. Abstract Formulation

Generalize the concrete formulation to arbitrarily many credit cards, number of months, initial card balance, cash available in each month, and monthly interest rate. You may assume that the available cash is the same in each month, and that the interest rate for each card does not change over time.

Data:

Decision variable:

Objective and constraints:

Step 4. Python Code

Implement your abstract formulation and find the numerical solution. The input data is given below.

You should print the minimum total payments and display a DataFrame listing the payments to each card each month. (See the expected output below.)

[8]:
```python
# Input data
I=['S','B','M']
n=36
T=range(1,n+1)
b={'S':20000,'B':50000,'M':40000}
c=5000
r={'S':0.005,'B':0.01,'M':0.015}
```

[10]:
```python
# Expected output
```

```
Minimum total payments: 121797.50485862953

       1       2       3       4       5       6       7       8   \
S     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
B     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
M  5000.0  5000.0  5000.0  5000.0  5000.0  5000.0  5000.0  5000.0

            9       10      11      12      13      14      15      16  \
S          0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
B  2736.955001  5000.0  5000.0  5000.0  5000.0  5000.0  5000.0  5000.0
M  2263.044999     0.0     0.0     0.0     0.0     0.0     0.0     0.0

       17      18      19           20      21      22      23      24  \
S     0.0     0.0     0.0   482.259871  5000.0  5000.0  5000.0  5000.0
B  5000.0  5000.0  5000.0  4517.740129     0.0     0.0     0.0     0.0
M     0.0     0.0     0.0          0.0     0.0     0.0     0.0     0.0

            25   26   27   28   29   30   31   32   33   34   35   36
S  1797.504859  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
B          0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
M          0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
```

Session 22: Creating Abstract Formulations

Tips for Creating Abstract Formulations

1. First create a concrete formulation or at least fragments of a concrete formulation. Write the abstract formulation by generalizing this, and afterward, manually expand the abstract formulation using made-up data and ensure that you get back the concrete formulation you started with.

2. Be familiar with **commonly used patterns**: capacity constraints, demand constraints, flow conservation constraints, conflicts, pre-requisites, conditional right hand sides, big-M, ...

3. Watch for **red flags**:

i) undefined indices or variables;

ii) clashing definitions of indices or variables;

iii) hard-coded numbers that are not inherent in the logic;

iv) summation signs with nothing under it.

v) the objective or constraints contain non-linear terms.

Examples from Last Class

Example 1: Assortment Planning

Data:

- B: the set of books.
- G: the set of genres.
- B_g: the set of books of genre g.
- r_g: the number of books required for genre g.

Decision Variables:

- x_b: whether to carry book b. (Binary)

Objective and constraints:

$$
\begin{aligned}
\text{Minimize:} \quad & \sum_{b \in B} x_b \\
\text{subject to:} \quad & \\
\text{(Enough books in genre)} \quad & \sum_{b \in B_g} x_b \geq r_g \qquad \text{for each genre } g \in G.
\end{aligned}
$$

Example 2: Food Production

Data:

- n: number of months.
- T: set of months. $T = \{1, 2, 3, \cdots, n\}$.
- p_t: price of oil in month t.
- d_t: demand in month t.
- s: amount of oil that can be stored at any time.

Decision Variables:

- x_t: amount of oil to buy in month t. (Continuous)
- y_t: amount of oil stored at the end of month t. (Continuous)

Objective and Constraints:

$$\begin{aligned}
\text{Minimize:} \quad & \sum_{t \in T} p_t x_t \\
\text{s.t.} \quad & y_1 = x_1 - d_1 \\
& y_t = x_t + y_{t-1} - d_t && \text{for each month } t \in \{2, 3, \cdots, n\}. \\
& y_t \leq s && \text{for each month } t \in T. \\
& x_t, y_t \geq 0
\end{aligned}$$

Example 3: Project Selection

Data:

- P: set of projects
- C: set of conflicts. Each $(p_1, p_2) \in C$ is a pair of projects that conflicts with one another.
- R: set of prerequisite pairs. Each $(p_1, p_2) \in R$ is a pair such that project p_1 is a prerequisite to project p_2.

Decision Variables: x_p: whether to pursue project p. (Binary)

Objective and Constraints:

$$\begin{aligned}
\text{Maximize} \quad & \sum_{p \in P} x_p \\
\text{s.t.} \quad & x_{p_1} + x_{p_2} \leq 1 && \text{For each conflicting pairs } (p_1, p_2) \in C. \\
& x_{p_1} \geq x_{p_2} && \text{For each pair } (p_1, p_2) \text{ such that } p_1 \text{ is a prereq to } p_2.
\end{aligned}$$

Exercise 11.4: Assigning Consultants to Projects

Trojan Consulting would like to assign consultants to projects in a way that minimizes total travel costs while satisfying the skill requirements of each project and avoiding assigning the same consultant to two projects with overlapping time frames.

In the following example, there are four consultants, each of whom may possess one or more of two possible skills. (A checkmark indicates whether the person has the skill.) Each project requires at least a certain number of consultants of each skill. If a consultant has both skills, he/she can count toward the number required for both skills, and the travel cost may potentially be less as one less person would be needed. Projects 1 and 2 have conflicting timelines, so the same consultant cannot be assigned to both. Similarly, projects 2 and 3 are also in conflict. But the same consultant may be assigned to projects 1 and 3.

Consultant	Accounting	Operations
Alice	✓	
Bob	✓	✓
Charlie		✓
Daphne	✓	✓

Project	Accounting	Operations
P1	2	1
P2	1	1
P3	0	2

Costs	P1	P2	P3
Alice	10	0	5
Bob	8	15	13
Charlie	0	5	10
Daphne	10	3	0

Write an abstract formulation of a linear optimization model to find a cost-minimizing assignment that satisfies all of the above constraints. Your formulation needs to be general enough to handle arbitrarily many consultants, projects, skills, as well as arbitrary information on skills of consultants, requirements of projects, conflicts between projects, and travel costs.

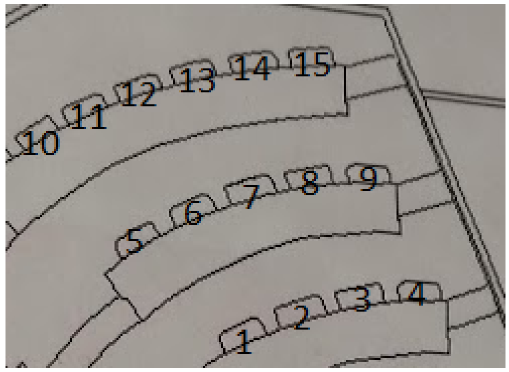
Data:

Decision Variables:

Objective and Constraints:

Exercise 11.5: Classroom Seating under Social Distancing

A committee at USC Marshall is exploring the viability of in-person instruction while observing social distancing guidelines. One challenge is that certain classrooms have tables and seats bolted to the floor, and the seats cannot be moved unless the rooms undergo significant remodeling. As an illustration, the following image is a portion of the floor plan for JKP204, and the numbers in the image correspond to the individual seats. As you can see, the distance between adjacent seats can be quite close, and the room would be overly dense if every seat is used. Since the seats cannot be moved, only a subset of them can be used to seat students.



Your task is to formulate an optimization problem to maximize the number of students that can be safely seated in the current classrooms without remodeling. The committee has access to the detailed floor plans of every classroom, and they have labelled every seat as above and measured its exactly position in terms of x-y coordinates, so they can easily compute the distance between any two seats. (For simplicity, the distance between two seated students is defined to be the straight-line distance between the center of the two seats.) Based on discussions with public health officials, the committee has summarized the requirements for safely seating students as follows:

1. The minimum distance between any two seated students is at least 6 feet.
2. For every seated student, the number of other students seated within a 9 feet radius is at most 3, so the density of the room is kept low. (In other words, if we draw a circle centered at a seated student with a radius of 9 feet, then there are at most 4 students seated strictly inside this circle, including the first student.)

Write an abstract formulation of a LP/MIP to solve the above problem, by listing all the data variables, decision variables, objective, and constraints. You may define any data variables that can be straightforwardly calculated based on the information the commitee has access to, but your definition must be completely clear and without ambiguities. Your formulation must be flexible enough to handle an arbitrary floor plan, not only the one shown above, and your objective and constraints must all be linear.

Data:

Decision Variables:

Objective and Constraints:

Exercise 11.6: Supply Chain Planning Revisited

This question modifies the supply chain planning problem of Exercise 8.2 by for both production and transportation. You company produces and sells a certain product. There is a certain set of production plants, each with a given capacity, which is the maximum number of units that can be produced there in a given month. There is a certain set of demand regions, each with a given estimated demand, which is the maximum number of units you can sell in that region per month. It is possible to not fulfill all the demand, as that corresponds to having the item being stocked out for some customers. The price that you charge when sell the item may be different in each region, but the prices are determined by another department, so is outside of your control.

As an illustration, suppose there are 3 plants, with the following production costs and capacities:

Plant	1	2	3
Cost per unit	60	60	55
Capacity	35	40	35

Suppose there are 4 regions, with the following selling prices and demand estimates:

Region	A	B	C	D
Price	80	85	70	75
Demand	30	50	10	20

The following table provides the transportation cost of shipping each unit from each plant to each region.

Region	Plant	1	2	3
A		20	8	25
B		18	23	8
C		21	24	8
D		8	8	19

Create an abstract formulation of a linear optimization problem to determine a plan for producing and shipping items, so as to maximize total profit, which is the revenue from the selling the product minus the production and transportation costs. You formulation must be flexible enough to handle arbitrary sets of production plants and regions, and none of the numbers in the above description can be hardcoded.

(In this problem, fractional units are allowed, since we are working with rates.)

Data:

Decision Variables:

Objective:

Constraints:

Instructions for Quiz 4

Quiz 4 will take place on Thursday next week, in the first fifteen minutes of class. The quiz will be open-notes but closed-computer. **You will be asked to create an abstract formulation to represent a given concrete formulation.** Learning to correctly write abstract formulations enables you to succinctly communicate mathematical patterns and increases your proficiency in mathematical modeling.

As always, the quiz is worth 4 percent of your final grade. If you miss the quiz, the weight will automatically be transferred to the final exam. After you are done, do not share the quiz questions or your answers with other students, including those in other sections. There will be many versions of the quiz, so don't be tempted to look at your neighbors' answers.

If you finish the quiz early, you can quietly stay in your seats or leave the room. **Do not use a cell phone, tablet or computer in the classroom before all the quizzes are handed in.**

Sample Quiz 4

Translate the below concrete formulation into an abstract formulation. You do not have to re-define the decision variables, but you need to define appropriate data variables. (Your abstract formulation should not hard-code any concrete data, but all data should be encoded by the data variables.)

Decision Variables:

- $X_1, X_2, \cdots, X_6$: amount of oil to buy in each month. (continuous)
- $Y_1, Y_2, \cdots, Y_6$: amount of oil stored at the end of each month to be used in the next month. (continuous)
- $Z_1, Z_2, \cdots, Z_6$: amount of oil stored at the end of each month to be used two months later. (continuous)

Objective and Constraints (Concrete):

$$
\begin{aligned}
\text{Min. } & 150X_1 + 160X_2 + 180X_3 + 170X_4 + 180X_5 + 160X_6 \\
& X_1 - Y_1 - Z_1 = 2000 \\
& X_2 - Y_2 - Z_2 = 2500 - Y_1 \geq 0 \\
& X_3 - Y_3 - Z_3 = 3000 - Y_2 - Z_1 \geq 0 \\
& X_4 - Y_4 - Z_4 = 3000 - Y_3 - Z_2 \geq 0 \\
& X_5 - Y_5 - Z_5 = 2500 - Y_4 - Z_3 \geq 0 \\
& X_6 - Y_6 - Z_6 = 2000 - Y_5 - Z_4 \geq 0 \\
& Y_1 + Z_1 \leq 8000 \\
& Y_2 + Z_2 + Z_1 \leq 8000 \\
& Y_3 + Z_3 + Z_2 \leq 8000 \\
& Y_4 + Z_4 + Z_3 \leq 8000 \\
& Y_5 + Z_5 + Z_4 \leq 8000 \\
& Y_6 + Z_6 + Z_5 \leq 8000 \\
& X_t, Y_t, Z_t \geq 0 \qquad \text{for each month } t \in \{1, 2, 3, 4, 5, 6\}.
\end{aligned}
$$

Data:

Objective and Constraints (Abstract):