Basics of optimization

SUPPLY CHAIN ANALYTICS IN PYTHON

Aaren Stubberfield
Supply Chain Analytics Mgr.

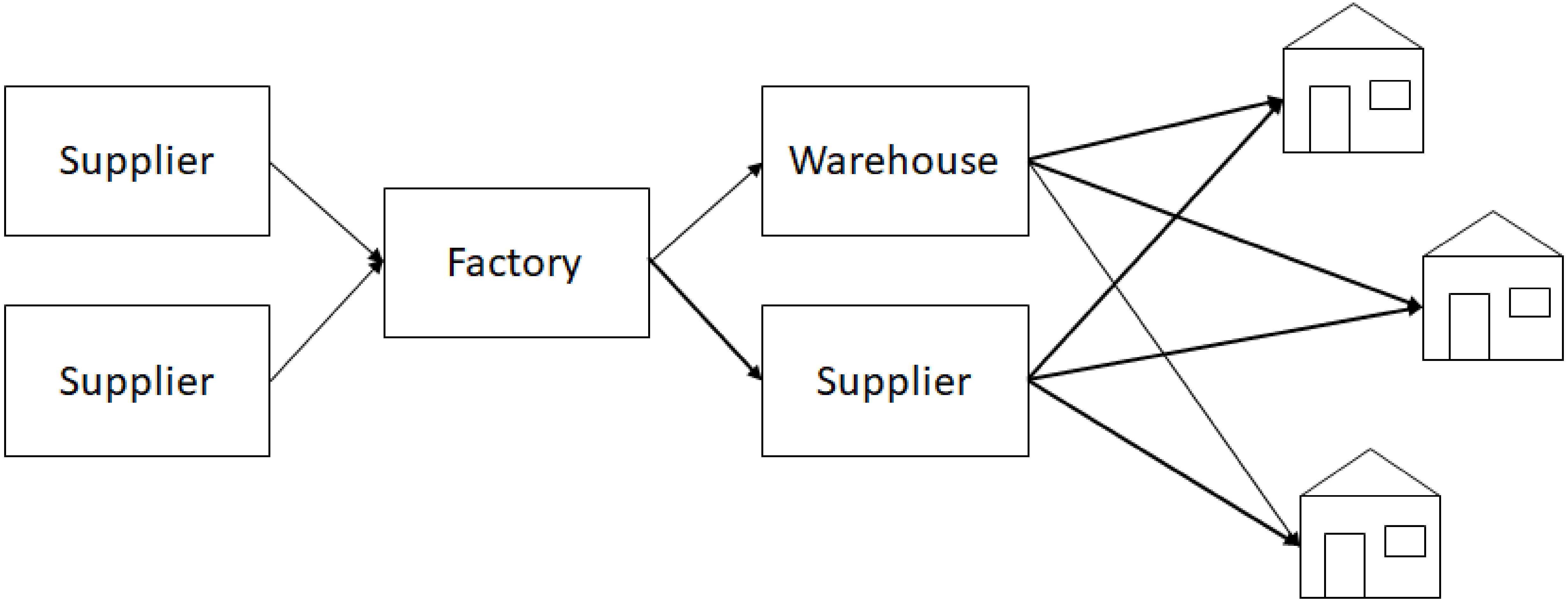
What is a supply chain

- A Supply Chain consist of all parties involved, directly or indirectly, in fulfilling a customer's request.[1]
- Includes:
  - Suppliers
  - Internal Manufacturing
  - Outsourced Logistics Suppliers (i.e. Third Party Suppliers)

[1] Chopra, Sunil, and Peter Meindl. _Supply Chain Management: Strategy, Planning, and Operations._ Pearson Prentice Hall, 2007.

What is a supply chain optimization

- Involves finding the best path to achieve an objective based on constraints



Crash course in LP

- Linear Programing (LP) is a Powerful Modeling Tool for Optimization
- Optimization method using a mathematical model whose requirements are linear relationships
- There are 3 Basic Components in LP:
    - Decision Variables - ***what you can control***
    - Objective Function - ***math expression that uses variables to express goal***
    - Constraints - ***math expression that describe the limits of a solutions***

Introductory example

Use LP to decide on an exercise routine to burn as many calories as possible.

	Pushup	Running
Minutes	0.2 per pushup	10 per mile
Calories	3 per pushup	130 per mile

Constraint - only 10 minutes to exercise

Basic components of an LP

Decision Variables - What we can control:

- Number of Pushups & Number of Miles Ran

Objective Function - Math expression that uses variables to express goal:

- Max (3 * Number of Pushups + 130 * Number of Miles)

Constraints - Math expression that describe the limits of a solutions:

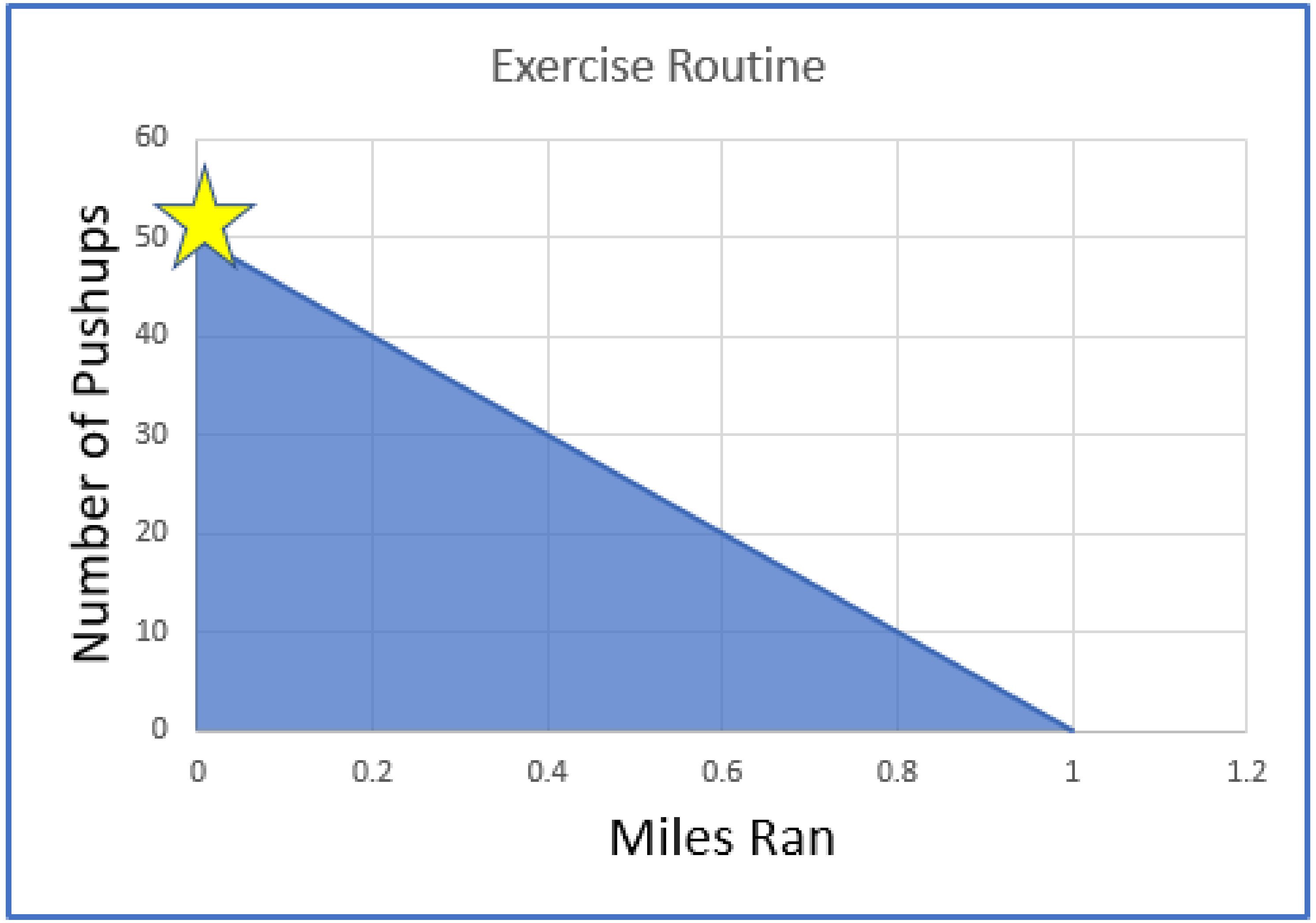
- 0.2 * Number of Pushups + 10 * Number of Miles $\leq$ 10
- Number of Pushups $\geq$ 0
- Number of Miles $\geq$ 0

Example solution

Optimal Solution:

- 50 Pushups
- 0 Miles Ran

Calories Burned: 150



LP vs IP vs MIP

Terms	Decision Variables
Linear Programing (LP)	Only Continuous
Integer Programing (IP)	Only Discrete or Integers
Mixed Integer Programing (MIP)	Mix of Continuous and Discrete

Summary

- Defined Supply Chain Optimization
- Defined Linear Programing and Basic Components
  - Decision Variables
  - Objective Function
  - Constraints
- Defined LP vs IP vs MIP

Let's practice!

SUPPLY CHAIN ANALYTICS IN PYTHON

Basics of PuLP modeling

SUPPLY CHAIN ANALYTICS IN PYTHON

Aaren Stubberfield
Supply Chain Analytics Mgr.

What is PuLP

- `PuLP` is a modeling framework for Linear (LP) and Integer Programing (IP) problems written in Python
- Maintained by COIN-OR Foundation (Computational Infrastructure for Operations Research)
- `PuLP` interfaces with Solvers
  - `CPLEX`
  - `COIN`
  - `Gurobi`
  - etc...

PuLP example – resource scheduling

- Consultant for boutique cake bakery that sell 2 types of cakes
- 30 day month
- There is:
  - 1 oven
  - 2 bakers
  - 1 packaging packer – only works 22 days

PuLP example – resource scheduling

- Different resource needs for the 2 types of cakes:

	Cake A	Cake B
Oven	0.5 days	1 day
Bakers	1 day	2.5 days
Packers	1 day	2 days

.

	Cake A	Cake B
Profit	$20.00	$40.00

PuLP example – resource scheduling

- Objective is to Maximize Profit
  - Profit = 20*A + 40*B
- Subject to:
  - $A \geq 0$
  - $B \geq 0$
  - $0.5A + 1B \leq 30$
  - $1A + 2.5B \leq 60$
  - $1A + 2B \leq 22$

Common modeling process for PuLP

1. Initialize Model
2. Define Decision Variables
3. Define the Objective Function
4. Define the Constraints
5. Solve Model

Initializing model - LpProblem()

```
LpProblem(name='NoName', sense=LpMinimize)
```

- `name` = Name of the problem used in the output .lp file, *i.e. "My LP Problem"*
- `sense` = Maximize or minimize the objective function
  - Minimize = `LpMinimize` *(default)*
  - Maximize = `LpMaximize`

PuLP example – resource scheduling

1. **Initialize Model**

```
from pulp import *


# Initialize Class
model = LpProblem("Maximize Bakery Profits", LpMaximize)
```

Define decision variables - LpVariable()

```
LpVariable(name, lowBound=None, upBound=None, cat='Continuous', e=None)
```

- `name` = Name of the variable used in the output .lp file
- `lowBound` = Lower bound
- `upBound` = Upper bound
- `cat` = The type of variable this is
  - Integer
  - Binary
  - Continuous *(default)*
- `e` = Used for column based modeling

PuLP example – resource scheduling

1. Initialize Class
2. **Define Variables**

```
# Define Decision Variables
A = LpVariable('A', lowBound=0, cat='Integer')
B = LpVariable('B', lowBound=0, cat='Integer')
```

PuLP example – resource scheduling

1. Initialize Class
2. Define Variables
3. **Define Objective Function**

```
# Define Objective Function
model += 20 * A + 40 * B
```

PuLP example – resource scheduling

1. Initialize Class
2. Define Variables
3. Define Objective Function
4. **Define Constraints**

```
# Define Constraints
model += 0.5 * A + 1 * B <= 30
model += 1 * A + 2.5 * B <= 60
model += 1 * A + 2 * B <= 22
```

PuLP example – resource scheduling

1. Initialize Class
2. Define Variables
3. Define Objective Function
4. Define Constraints
5. **Solve Model**

```
# Solve Model
model.solve()
print("Produce {} Cake A".format(A.varValue))
print("Produce {} Cake B".format(B.varValue))
```

PuLP example – resource scheduling

```python
from pulp import *

# Initialize Class
model = LpProblem("Maximize Bakery Profits",
                  LpMaximize)

# Define Decision Variables
A = LpVariable('A', lowBound=0,
               cat='Integer')
B = LpVariable('B', lowBound=0,
               cat='Integer')

# Define Objective Function
model += 20 * A + 40 * B
```

```python
# Define Constraints
model += 0.5 * A + 1 * B <= 30
model += 1 * A + 2.5 * B <= 60
model += 1 * A + 2 * B <= 22

# Solve Model
model.solve()
print("Produce {} Cake A".format(A.varValue))
print("Produce {} Cake B".format(B.varValue))
```

Summary

- PuLP is a Python LP / IP modeler
- Reviewed 5 Steps of PuLP modeling process
  1. Initialize Model
  2. Define Decision Variables
  3. Define the Objective Function
  4. Define the Constraints
  5. Solve Model
- Completed Resource Scheduling Example

Let's practice!

SUPPLY CHAIN ANALYTICS IN PYTHON

Using lpSum

SUPPLY CHAIN ANALYTICS IN PYTHON

Aaren Stubberfield
Supply Chain Analytics Mgr.

Moving from simple to complex

Simple Bakery Example

```
# Define Decision Variables
A = LpVariable('A', lowBound=0, cat='Integer')
B = LpVariable('B', lowBound=0, cat='Integer')
```

More Complex Bakery Example

```
# Define Decision Variables
A = LpVariable('A', lowBound=0, cat='Integer')
B = LpVariable('B', lowBound=0, cat='Integer')
C = LpVariable('C', lowBound=0, cat='Integer')
D = LpVariable('D', lowBound=0, cat='Integer')
E = LpVariable('E', lowBound=0, cat='Integer')
F = LpVariable('F', lowBound=0, cat='Integer')
```

Moving from simple to complex

Objective Function of Complex Bakery Example

```
# Define Objective Function
model += 20*A + 40*B + 33*C + 14*D + 6*E + 60*F
```

Need method to scale

$$z = X1 + X2 + X3 + \cdots + Xk$$

Using lpSum()

```
lpSum(vector)
```

- `vector` = A list of linear expressions

Therefore ...

```
# Define Objective Function
model += 20*A + 40*B + 33*C + 14*D + 6*E + 60*F
```

Equivalent to ...

```
# Define Objective Function
var_list = [20*A, 40*B, 33*C, 14*D, 6*E, 60*F]
model += lpSum(var_list)
```

lpSum with list comprehension

```
# Define Objective Function
cake_types = ["A", "B", "C", "D", "E", "F"]
profit_by_cake = {"A":20, "B":40, "C":33, "D":14, "E":6, "F":60}
var_dict = {"A":A, "B":B, "C":C, "D":D, "E":E, "F":F}

model += lpSum([profit_by_cake[type] * var_dict[type]
                for type in cake_types])
```

Summary

- Need way to sum many variables
- `lpSum()`
- Used in list comprehension

Practice time!

SUPPLY CHAIN ANALYTICS IN PYTHON