The whole is greater than the sum of its parts

Using combinatorial optimization to create synergy

Jennifer Nguyen

AnacondaCON 2020

Planning a conference

Speakers

Attendees

Schedule

What is combinatorial optimization?

Process of searching for a maximum/minimum of an objective function whose domain is a discrete configuration space

– *A math textbook*

What is combinatorial optimization?

Process of searching for a maximum/minimum of **an objective function** whose domain is a discrete configuration space

– *A math textbook*

English translation:

A quantity to maximize or minimize (KPI)

E.g., revenue, time spent, productivity, costs, etc.

What is combinatorial optimization?

Process of searching for a maximum/minimum of an objective function whose domain is a **discrete configuration space**

– *A math textbook*

English translation:

A set of countable objects

E.g., products, driving routes, employees, processes, etc.

What is combinatorial optimization?

Process of **searching for a maximum/minimum** of an objective function whose domain is a discrete configuration space

– *A math textbook*

English translation:

Find the set of objects that maximizes/minimizes the KPI

Components of a CO problem

KPI to optimize

Objects that affect KPI

Agenda

- Introduce common CO problems in business
- How to solve these problems using Google OR Tools

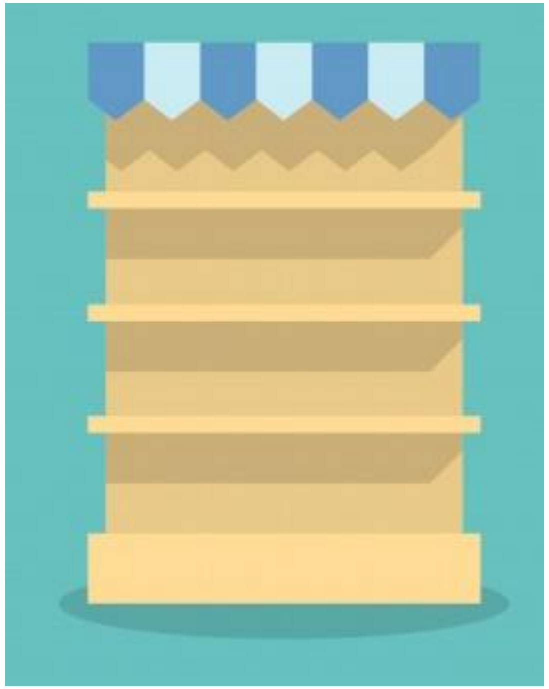
Packing Problems

How to stock a store

You have limited shelf space and can only stock so many products.

Each product has an associated profit margin.

Which items do you display to maximize profit?

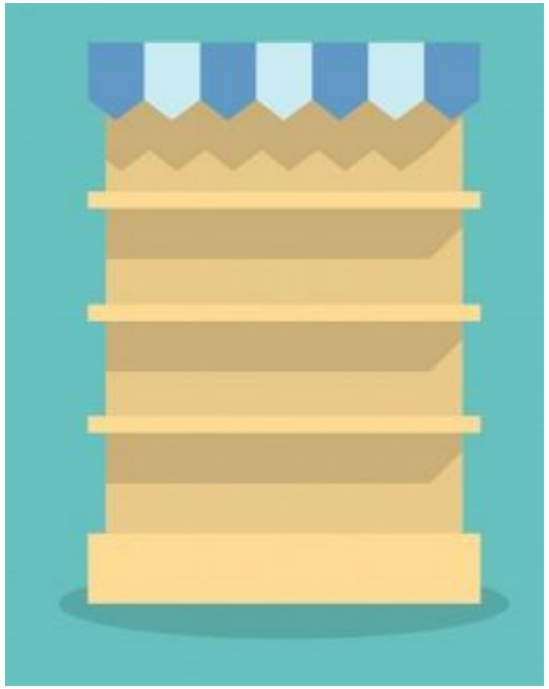


How to stock a store

You have limited shelf space and can only stock so many products.

Each product has an associated **profit margin**.

Which items do you display to maximize profit?

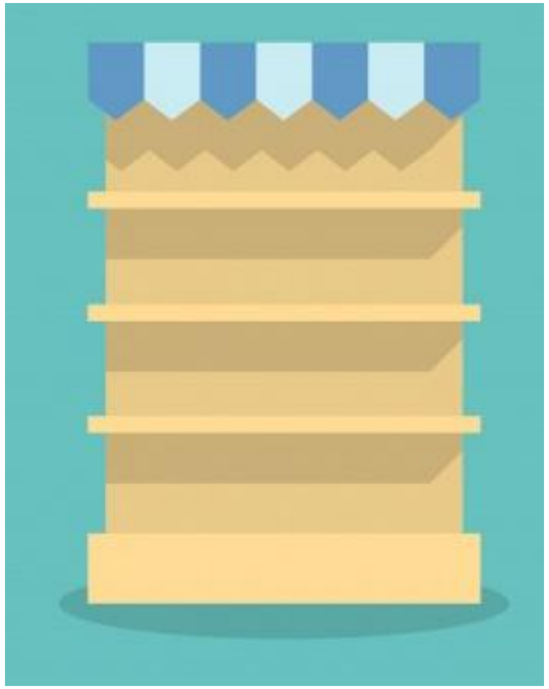


How to stock a store

You have limited shelf space and can only stock so many **products**.

Each product has an associated profit margin.

Which items do you display to maximize profit?



Knapsack problem

Given

1. a knapsack of limited capacity
2. N items of specific weights and user utility

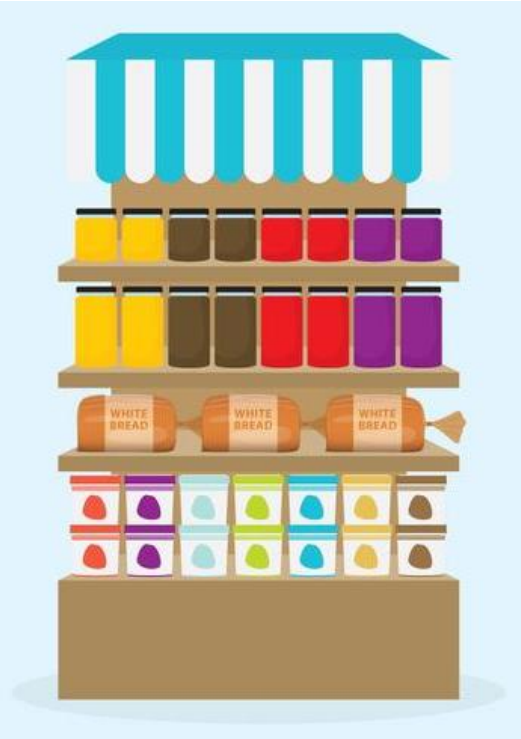
Determine the quantity of each item to pack to maximize overall utility

Knapsack problem: Grocery store

Given

1. shelf space
2. grocery items of varying size and varying profit margins

Determine the quantity of each item to stock to **maximize profit margin**



Knapsack problem: Digital library

Given

1. A book budget
2. Titles with associated costs and customer "enjoyment" (retention)

Determine which titles to include in the catalogue to **maximize user retention**

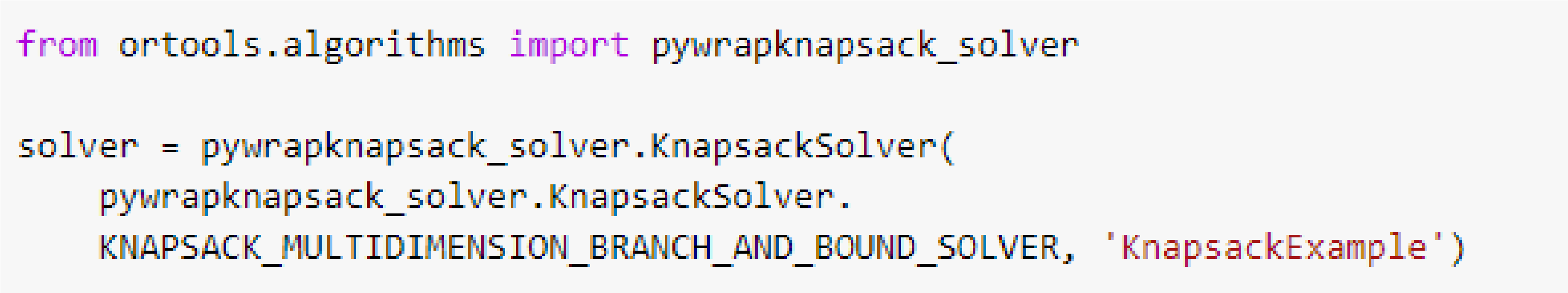
Creating an optimal library with OR Tools

Jupyter Notebook

https://bit.ly/36pZWkU

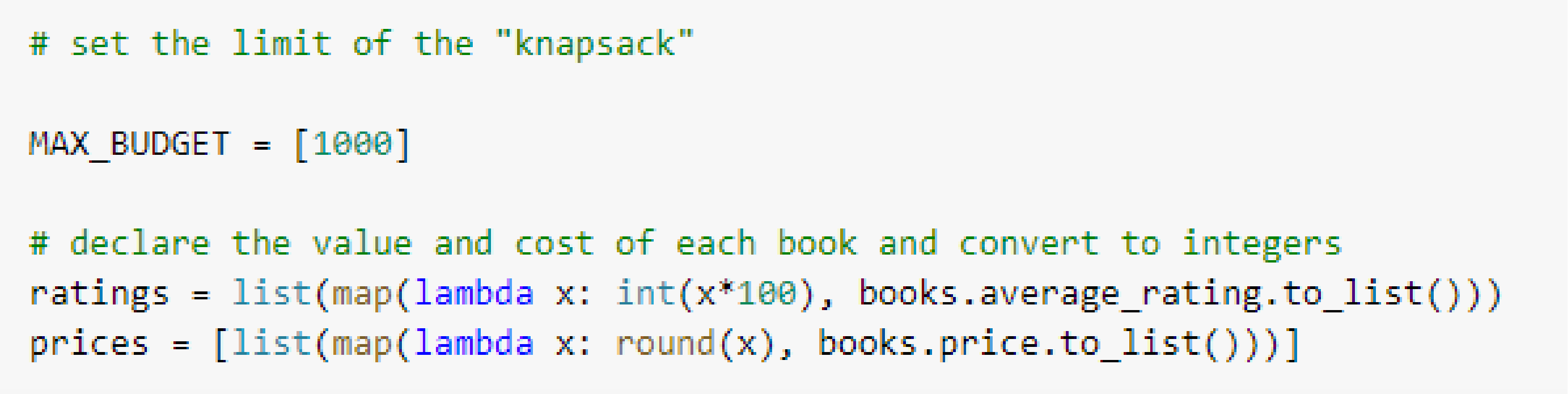
1. Initialize the solver
2. Specify the budget, costs and ratings
3. Run the solver

Step 1: Initialize the solver



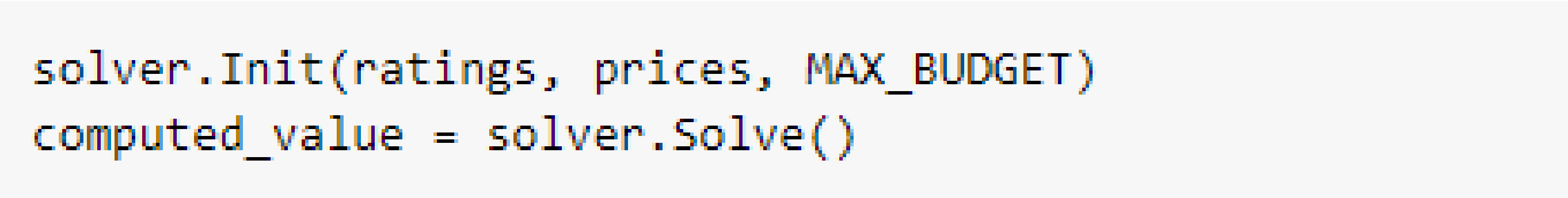
```
from ortools.algorithms import pywrapknapsack_solver

solver = pywrapknapsack_solver.KnapsackSolver(
    pywrapknapsack_solver.KnapsackSolver.
    KNAPSACK_MULTIDIMENSION_BRANCH_AND_BOUND_SOLVER, 'KnapsackExample')
```

Step 2: Specify the budget, costs and ratings



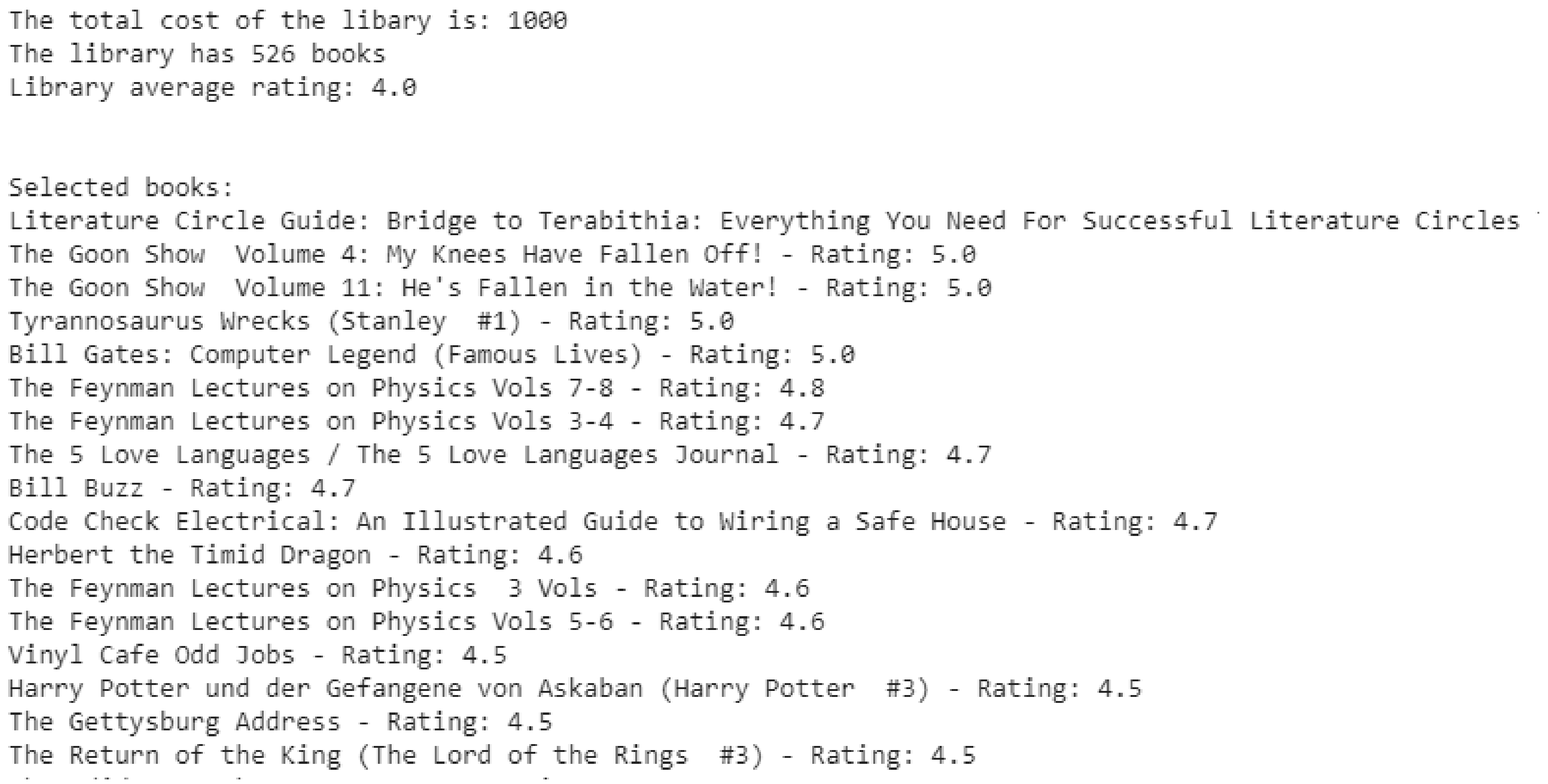
```
# set the limit of the "knapsack"

MAX_BUDGET = [1000]

# declare the value and cost of each book and convert to integers
ratings = list(map(lambda x: int(x*100), books.average_rating.to_list()))
prices = [list(map(lambda x: round(x), books.price.to_list()))]
```

Step 3: Run the solver



```
solver.Init(ratings, prices, MAX_BUDGET)
computed_value = solver.Solve()
```

The optimal library collection



```
The total cost of the libary is: 1000
The library has 526 books
Library average rating: 4.0


Selected books:
Literature Circle Guide: Bridge to Terabithia: Everything You Need For Successful Literature Circles
The Goon Show  Volume 4: My Knees Have Fallen Off! - Rating: 5.0
The Goon Show  Volume 11: He's Fallen in the Water! - Rating: 5.0
Tyrannosaurus Wrecks (Stanley  #1) - Rating: 5.0
Bill Gates: Computer Legend (Famous Lives) - Rating: 5.0
The Feynman Lectures on Physics Vols 7-8 - Rating: 4.8
The Feynman Lectures on Physics Vols 3-4 - Rating: 4.7
The 5 Love Languages / The 5 Love Languages Journal - Rating: 4.7
Bill Buzz - Rating: 4.7
Code Check Electrical: An Illustrated Guide to Wiring a Safe House - Rating: 4.7
Herbert the Timid Dragon - Rating: 4.6
The Feynman Lectures on Physics  3 Vols - Rating: 4.6
The Feynman Lectures on Physics Vols 5-6 - Rating: 4.6
Vinyl Cafe Odd Jobs - Rating: 4.5
Harry Potter und der Gefangene von Askaban (Harry Potter  #3) - Rating: 4.5
The Gettysburg Address - Rating: 4.5
The Return of the King (The Lord of the Rings  #3) - Rating: 4.5
```

Constraint Optimization

Constraint optimization

This is constraint optimization where the goal is to find a **feasible** solution under a **set of conditions.**

Optimization is a secondary goal

Constraint optimization: Planning a conference

Schedule the **speakers** so that:

A. All attendees are able to attend their top two "must attend" speakers

B. Accommodate speakers' availability

To **maximize participant satisfaction**

Constraint optimization: Forming teams

Financial advisors working in teams are known to be more productive than working alone.

How can we form teams of advisors so that overall **productivity is maximized**?

Constraint optimization: Forming teams

Conditions:

1. Are located < 10km of each other

2a. Each team has one senior member and one junior member, or

2b. At least one member is a specialist in area A and at least one member is a specialist in area B

3. Everyone must be on a team

4. Teams have a maximum size of 8 people

4 steps to creating synergistic teams

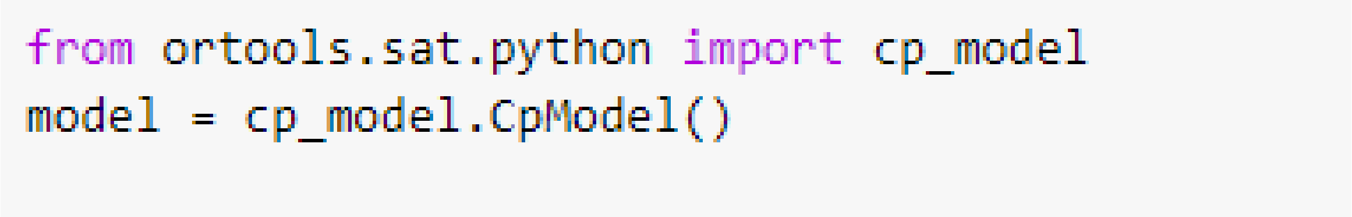
Jupyter Notebook

https://bit.ly/2A67GMJ

1. Initialize the solver
2. Declare decision variables
3. Add the constraints
4. Run the solver

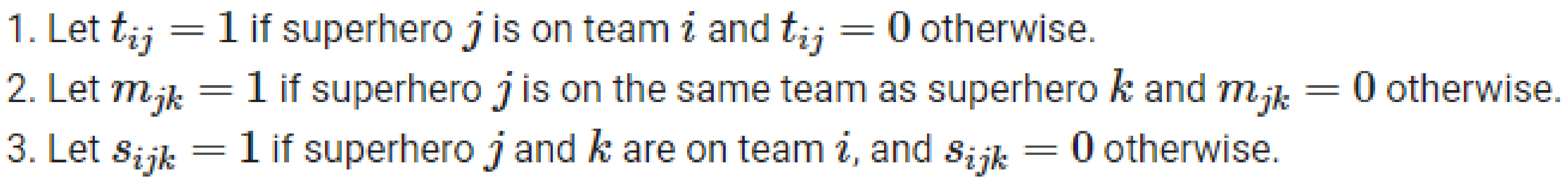
Teaming superheroes

Step 1: Initialize the solver



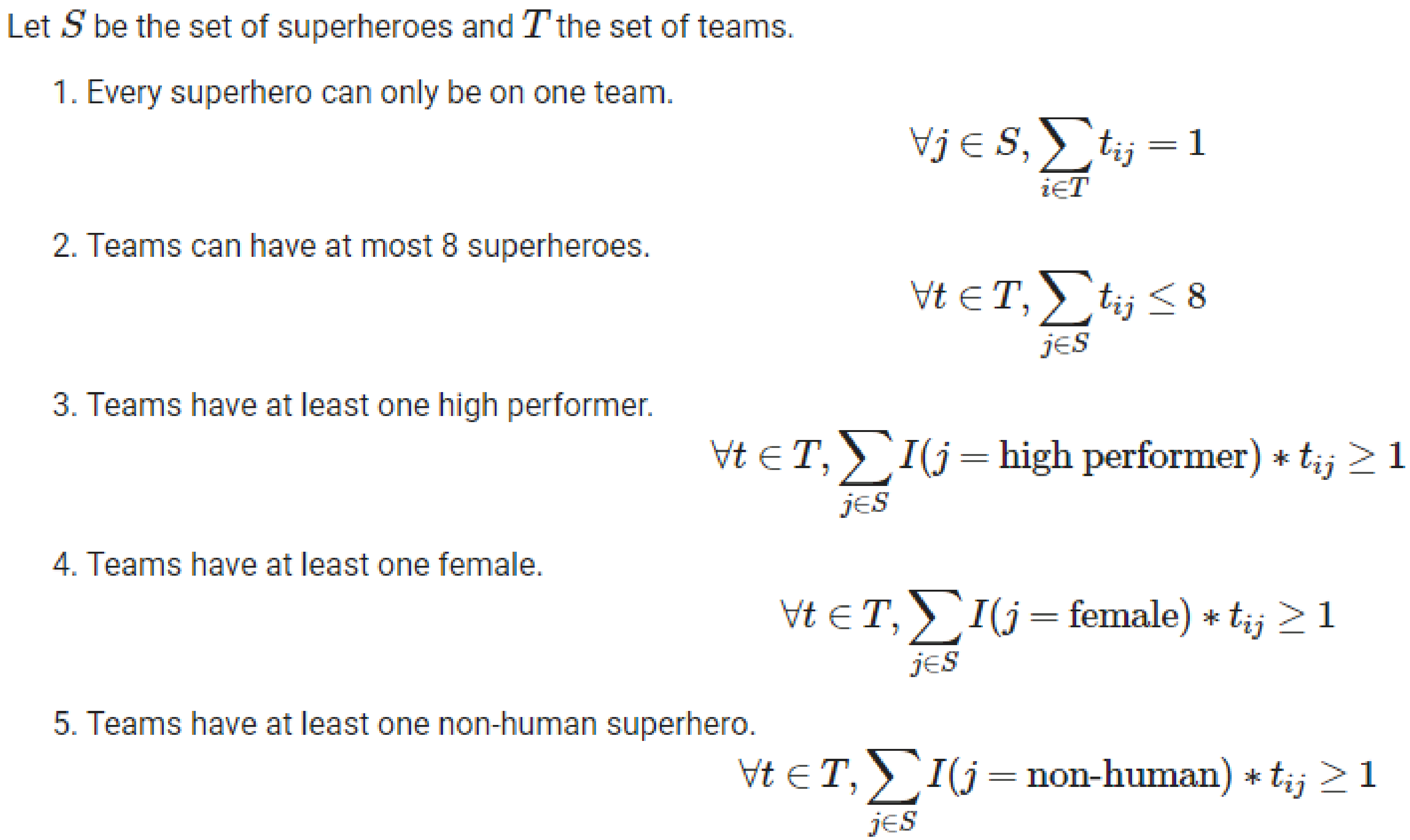
```
from ortools.sat.python import cp_model
model = cp_model.CpModel()
```

Step 2: Declare decision variables



1. Let $t_{ij} = 1$ if superhero j is on team i and $t_{ij} = 0$ otherwise.
2. Let $m_{jk} = 1$ if superhero j is on the same team as superhero k and $m_{jk} = 0$ otherwise.
3. Let $s_{ijk} = 1$ if superhero j and k are on team i, and $s_{ijk} = 0$ otherwise.

Step 3: Add the constraints



Let S be the set of superheroes and T the set of teams.

1. Every superhero can only be on one team.

$$\forall j \in S, \sum_{i \in T} t_{ij} = 1$$

2. Teams can have at most 8 superheroes.

$$\forall t \in T, \sum_{j \in S} t_{ij} \leq 8$$

3. Teams have at least one high performer.

$$\forall t \in T, \sum_{j \in S} I(j = \text{high performer}) * t_{ij} \geq 1$$

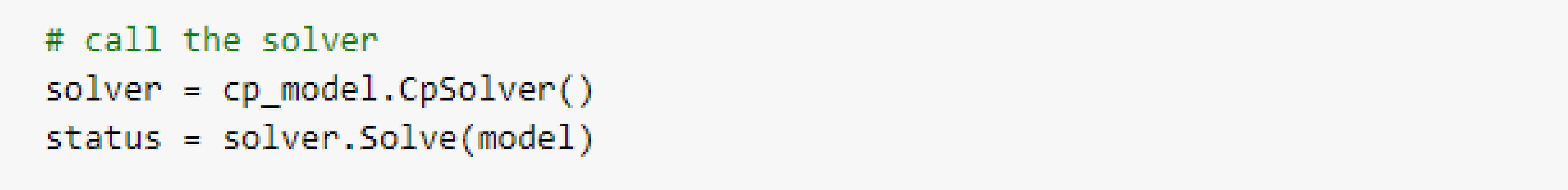
4. Teams have at least one female.

$$\forall t \in T, \sum_{j \in S} I(j = \text{female}) * t_{ij} \geq 1$$

5. Teams have at least one non-human superhero.

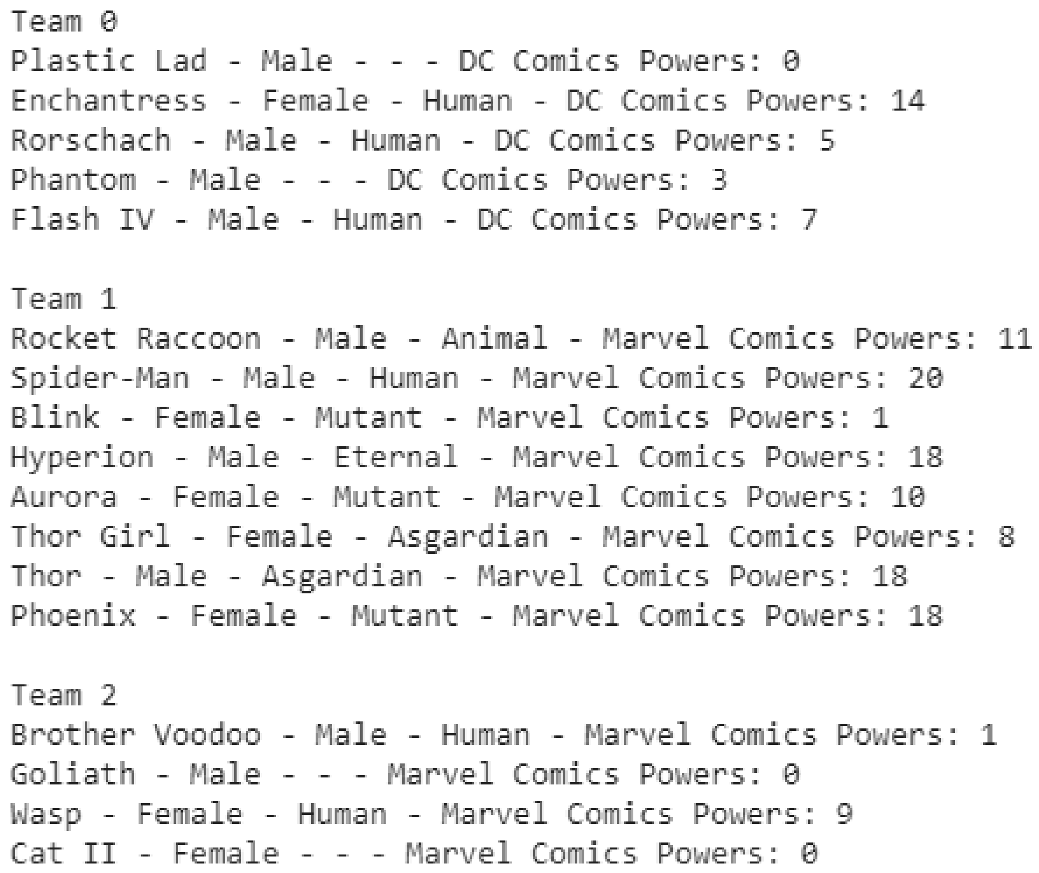
$$\forall t \in T, \sum_{j \in S} I(j = \text{non-human}) * t_{ij} \geq 1$$

Step 4: Run the solver



```
# call the solver
solver = cp_model.CpSolver()
status = solver.Solve(model)
```

Team assignments



```
Team 0
Plastic Lad - Male - - - DC Comics Powers: 0
Enchantress - Female - Human - DC Comics Powers: 14
Rorschach - Male - Human - DC Comics Powers: 5
Phantom - Male - - - DC Comics Powers: 3
Flash IV - Male - Human - DC Comics Powers: 7

Team 1
Rocket Raccoon - Male - Animal - Marvel Comics Powers: 11
Spider-Man - Male - Human - Marvel Comics Powers: 20
Blink - Female - Mutant - Marvel Comics Powers: 1
Hyperion - Male - Eternal - Marvel Comics Powers: 18
Aurora - Female - Mutant - Marvel Comics Powers: 10
Thor Girl - Female - Asgardian - Marvel Comics Powers: 8
Thor - Male - Asgardian - Marvel Comics Powers: 18
Phoenix - Female - Mutant - Marvel Comics Powers: 18

Team 2
Brother Voodoo - Male - Human - Marvel Comics Powers: 1
Goliath - Male - - - Marvel Comics Powers: 0
Wasp - Female - Human - Marvel Comics Powers: 9
Cat II - Female - - - Marvel Comics Powers: 0
```

Resources

Optimizing library Jupyter Notebook – https://bit.ly/36pZWkU

Creating synergistic teams Jupyter Notebook – https://bit.ly/2A67GMJ

Google OR-Tools – https://developers.google.com/optimization

Branch & Bound algorithm (Stanford) – https://stanford.io/2APH4QA

Thank you

Contact Info:
Jennifer@jennguyen.ca
linkedin.com/in/jenguyen