MiniProject 2: Vehicle Routing Problem

IP Model and Python Implementation

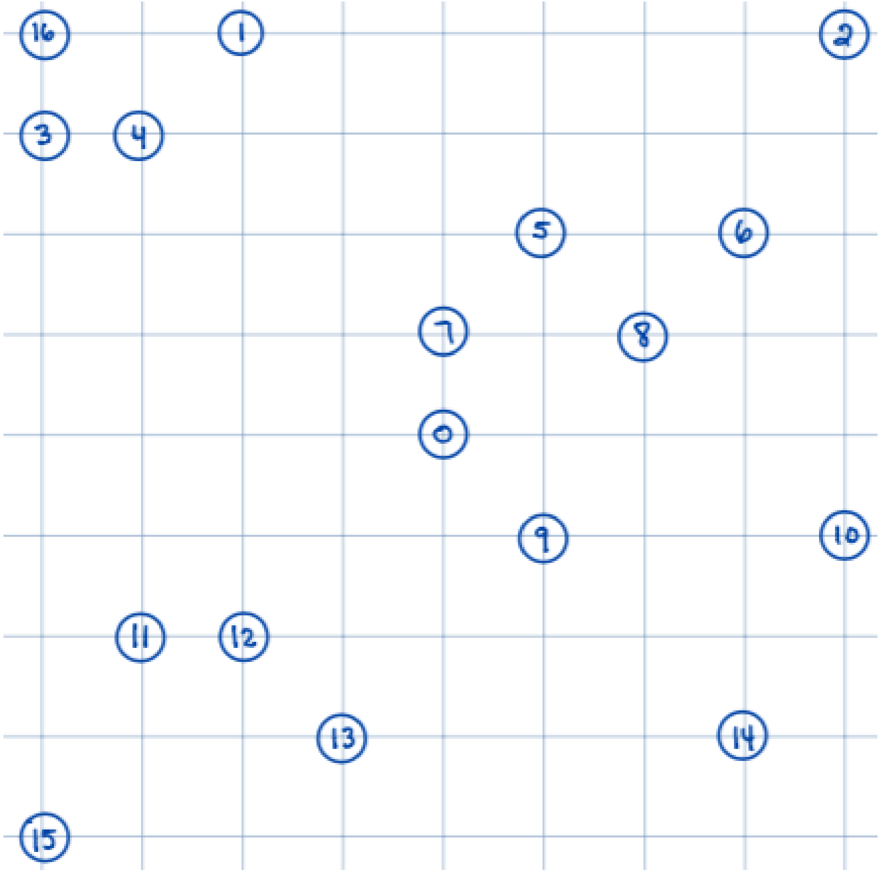
- Formulation and python implementation due **Tuesday 2 November at 2359**

Ground Rules:

- This is an individual project
- You may ask me for help, your textbook/notes, or Google but cite your sources.

1 The Problem

The USNA coffee company has decided to start delivering (iced) coffee around the yard. The diagram below shows the location of the coffee company (location 0) and the coordinates of each customer on the yard. Assume that each customer has the same demand. The USNA coffee company has four (4) people who are available to deliver coffee and each of which can deliver to up to five customers. Distances are the direct, straight-line, distance between points.



Part 1: Formulation. Formulate this model as a vehicle routing problem.

a. Write a concrete model for this VRP model. Do **not** include subtour elimination constraints.

b. Convert your model to a parameterized VRP model with generalized subtour elimination constraints.

Part 2: Python implementation with manual subtour elimination constraints.

a. Using the project 2 starter file, complete the Jupyter notebook to implement your VRP model.

b. For iteratively adding subtour elimination constraints, at each step, explain why your current solution is wrong and what you need to change before you solve the problem again.

c. To make sure you're on the right track, your initial solution, with no subtour elimination constraints, will have an objective function value of `44.59856358193051`

Grading: For grading, submit both your formulation and Jupyter notebook. The formulation will count as 25% of your project 2 grade while the python code will count for 75% of your project 2 grade.

OPTIONAL BONUS: In modern solvers for VRP/TSP, the subtour elimination constraints are automated. As an optional bonus, modify your Jupyter notebook to automate the cycle elimination constraints. That is, if I run your notebook, I run it once and it will produce the final solution while eliminating cycles as it goes without user input. To do so, you need to determine how to detect a cycle based on a given solution and how to automate the cycle elimination constraints. This is completely **optional**, but if you do it correctly bonus points will be added to your project 2 grade.

Hint for the bonus: In order to automate adding constraints, you need to add multiple constraints of similar names. One easy way to do this is to use the function `model.add_component`. For example, if I want to add my favorite constraint iteratively, one way to do it is:

```
model.add_component(f'my_const_{counter}',pyo.Constraint(rule=my_const_rule))
```

Where counter is the index of the constraint I'd like to add.