# 1 Healthcare Routing Optimization Assignment

Following your successful design of healthcare facility locations in Assignment 1, the organization now needs to plan the delivery of healthcare resources to residential areas. These resources will be distributed from one central healthcare facility (the depot), and efficient routing is critical to minimize costs and ensure timely service to all areas.

The problem can be modeled as a Traveling Salesman Problem (TSP), where the objective is to minimize the total distance traveled while visiting each residential area exactly once. The distance between sites is calculated using the Manhattan distance:

$$d_{ij} = |x_i - x_j| + |y_i - y_j|.$$

**Objective and Requirements**

The goal is to find the optimal route, starting and ending at the single depot, minimizing the total travel distance. To solve the problem, we consider two alternative formulations:

1. Edge formulation:

$$\min \quad \sum_{(i,j)\in E} d_{ij}x_{ij} \tag{1a}$$

$$\text{s.t.} \quad \sum_{i\in V:(i,j)\in E} x_{ij} + \sum_{i\in V:(j,i)\in E} x_{ji} = 2 \qquad \forall j \in V \tag{1b}$$

$$\sum_{(i,j)\in E: i\in S, j\notin S} x_{ij} + \sum_{(j,i)\in E: i\in S, j\notin S} x_{ji} \geq 2 \qquad \forall S \subset V : 2 \leq |S| \leq \left\lceil \frac{|V|}{2} \right\rceil \tag{1c}$$

$$x_{ij} \in \mathbb{B} \qquad \forall (i,j) \in E \tag{1d}$$

where $V = \{1, \ldots, n\}$ is the set of sites, $E = \{(i,j) \in V \times V : i < j\}$ the set of undirected edges, $d_{ij}$ the distance between two sites $i$ and $j$ and $x_{ij}$ a decision variable indicating whether we travel between those sites or not. Although this model has a strong LP relaxation, such that only a limited number of branching nodes are needed to solve to optimality, it includes an exponential number of constraints (1c) to eliminate subtours.

2. Alternative formulation:

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in A} d_{ij} z_{ij} && && \text{(2a)}\\
\text{s.t.} \quad & \sum_{i\in V:(i,j)\in A} z_{ij} = 1 && \forall j \in V && \text{(2b)}\\
& \sum_{i\in V:(j,i)\in A} z_{ji} = 1 && \forall j \in V && \text{(2c)}\\
u_i - u_j + (n-1) & z_{ij} \le n-2 && \forall (i,j) \in A : i \neq 1, j \neq 1 && \text{(2d)}\\
& z_{ij} \in \mathbb{B} && \forall (i,j) \in A && \text{(2e)}\\
& u_i \in \{1, \dots, n-1\} && \forall i \in V \setminus \{1\} && \text{(2f)}
\end{aligned}
$$

where $A = \{(i,j) \in V \times V : i \neq j\}$ the set of directed arcs, $z_{ij}$ a decision variable whether we travel from site $i$ to site $j$ and $u_i$ some dummy variable. This formulation has a polynomial number of constraints, but at the cost of more integer variables and (quite possibly) weaker lower bounds.

**Exercise 1.** Explain the correctness of Formulation (2) for the TSP, particularly the pivotal role of the dummy variables $u_i$ in eliminating subtours. (1 point)

**Exercise 2.** Implement both TSP models and (try to) solve them. For both models, report what happens. That is, if it solves, report the optimal objective value and plot the solution route. Also provide the time it takes Gurobi to solve the problem to optimality. If it does not solve, report why. (2 points)

To enhance efficiency for the first formulation, you can use a branch-and-cut algorithm.

**Exercise 3.** Before delving into the branch-and-cut algorithm, elaborate on how you detect violated subtours (cycle detection) in the context of the TSP. (1 point)

For adding subtour elimination constraints, there are several possibilities. One approach is to add a cut for each subtour in the current optimal solution. Another possibility is to add a cut only for the first subtour that you encounter (i.e. adding a single cut per iteration).

**Exercise 4.** Explain the advantages and disadvantages of adding a cut for each subtour in the current optimal solution versus adding a cut only for the first subtour encountered. (0.5 point)

**Exercise 5.** Implement branch-and-cut for the edge formulation for the TSP as follows:

1. Solve model (1a)-(1d) to integer optimality without any cuts (1c).
2. Identify all violated subtours (using the cycle detection you described in Exercise 2) and add the corresponding cuts to the model.[1]
3. Re-solve the model and keep adding cuts until the found solution does not contain subtours.

Report the time it takes to solve the model to optimality. (2 points)

Another way to possibly speed up the process is to implement a warm start.

**Exercise 6.** In this question, you are asked to implement a MIP start for the TSP. The question consists of three parts:

1. Describe and implement a nearest-neighbour-like heuristic for the TSP. You should start from the first site in the data file. Provide a plot of the heuristic solution route and report the heuristic solution value. (1 point)
2. Let the heuristic solution serve as a warm start for formulation (2) of the TSP model. Describe how to set the values of $u_i$ given the heuristic solution. Report the time it takes Gurobi to solve the model to optimality when using the heuristic solution as a MIP start, distinguishing between the time it takes to generate the heuristic solution and the time it takes to solve the model with the heuristic solution as the MIP start. Compare with the results from Exercise 2. Discuss whether it performs well in this case or not. (1.5 points)
3. Instead of taking the first site in the data file as starting point for your heuristic, now repeat for all other sites in the data file. Report the objective function and starting point of the best found heuristic solution. Use this as a warm start for formulation (2) of the TSP model. Again, describe how to set the values of $u_i$ given the heuristic solution in this case. Report the time it takes Gurobi to solve the model to optimality when using the heuristic solution as a MIP start, distinguishing between the time it takes to generate the heuristic solution and the time it takes to solve the model with the heuristic solution as the MIP start. Compare with the results from Step 2. (1 point)

**Exercise 7.** As an alternative (or addition) to using a warm start, the following modifications can be made to make model (2) solve faster:

- Replacing constraints (2d) with $u_i - u_j + (n-1)z_{ij} + (n-3)z_{ji} \leq n-2, \forall i, j \in A : i \neq 1, j \neq 1$.
- Adding the constraints $u_i \geq 1 + (n-3)z_{i1} + \sum_{j \neq 1:(j,i) \in A} z_{ji}, \forall i \in V \setminus \{1\}$.

[1] Note that the idea of this assignment is that you try out different features of Gurobi. Hence, you are supposed to modify your Gurobi model in each iteration. Creating a new model after each iteration will not give you any points.

- Adding the constraints $u_i \neq n - 1 - (n-3)z_{1i} + \sum_{j \neq 1:(i,j)\in A} z_{ij}, \forall i \in V \setminus \{1\}$.

Program the modifications, report the solution, its objective value and solving time. Does adding the warm start from Part 2 of Exercise 6 help the model? (2 points)

**Exercise 8.** Usually, practitioners like to see more than one solution. For both models, describe a strategy to determine an alternative solution. Implement it and provide the new objective value and plot the solution route. If there are multiple optimal solutions, your strategy should identify a solution with the same objective value. If the optimal solution is unique, your strategy should find the second-best solution. Clearly indicate the distinction between the earlier optimal tour you discovered and this newly identified tour. (1 point)

To make the problem more difficult, the vehicle providing healthcare resources is powered by electricity and has limited battery capacity. Charging, in this case, is only possible at specific recharging stations scattered across the city (in particular, at locations $R \subset V$, which will be provided in a separate file)[2]. The vehicle can travel a maximum distance of $D = 100$ km before it needs to recharge. Consequently, the route must be optimized to ensure that the vehicle can complete the replenishment without running out of battery.

**Exercise 9.** Write down an extension to the MILP (2a)-(2f) to incorporate the new recharging restriction. Clearly introduce any new variables or parameters that you use. (2 points)

**Bonus Exercise.** Implement and solve the model. Provide the optimal objective value and plot the solution. Make sure that the distinction between locations with and without recharging possibility is visible. Provide the time it takes Gurobi to solve the model. (2 points)

## 2 Data

The file containing the coordinates of all the residential areas and their corresponding population is available in `Residential_areas_2.txt` on the Canvas website. Every row in this file corresponds to one residential area, presented in the form $x_i, y_i$ with $x_i$ the x-coordinate and $y_i$ the y-coordinate of residential area $i$. The file containing the fuel locations is `Fuel_Locations.txt`. All other data is provided in the exercise itself.

## 3 Instructions

The assignment can be carried out individually or in groups of two persons.

[2] You may assume that the depot is located at one of the recharging locations.

- In case of a group of two, both members must participate in assignment preparation.
- You are not allowed to work together with other groups or individuals.
- Mention your names and student numbers clearly at the start of the report.
- Write your code in a way that it can be used with different sets of instances
- Document your code. Add comments in your code that makes them readable.
- You will need to hand in the assignment on Canvas, under "Assignments". Reports need to be in PDF format (LaTeX preferred), adhering to the naming scheme "*<studentnr>.pdf*" or "*<studentnr1><studentnr2>.pdf*". You will also need to hand in any source code combined archived in a .zip file. An example submission is
  1. "123456 789012.pdf"
  2. "123456 789012.zip"
- The assignment contains 15% of the final grade
- All models must be linear. Additionally, the efficiency of your models will be considered during grading. Introducing unnecessary variables and constraints may result in a deduction of points.
- The submission deadline is on **Tuesday 11th of February 2025** at 23:59 in the night. Submissions after this date and hour will not be corrected.
- Achieving two bonus points is possible, and (part of) the bonus points from Assignment 1 can be carried over to the second assignment if you scored more than 13 regular points in the first assignment. However, there's a cap on the total points you can attain over the two assignments. To clarify, suppose you managed to secure 15 points in the first assignment and earned the two bonus points, such that you have a total of 17 points for that assignment.
  1. If you obtain 12 points for the second assignment, your total will be 29 out of 30 points.
  2. If you obtain a full score of 15 points for the second assignment, your overall score will be capped at 30 points.

  Hence, bonus points cannot be carried over to the final exam.