Project for Introduction to Operation Research

Deadline December 9th

Submission requirement. Send (*i*) a runnable Jupyter notebook (`.ipynb`) containing the implemented code and answers to the coding exercise (referred to as **Part 1** bellow); sample source code (`source_code.ipynb`) with both Gurobi and LaTeXcode is available on OLAT. You should directly modify this notebook file. (*ii*) A PDF (written in LaTeX) with your answers to the modeling exercise (referred to as **Part 2** bellow) in LaTeX. Focus on both the modeling and the explanations. Clear reasoning earns points. You do not need to match any "official" answer to receive full points (a LaTeX file of this project description can be used as a template).

Motivation. Swiss Post runs last-mile delivery in Swiss cities and in the mountains. Think of a weekday morning in Zürich. At 07:30 a yellow van leaves the depot near the edge of the city. The driver has about 35-40 stops: a few cafés that want parcels before the morning rush, some apartment buildings with intercoms, two parcel lockers, and one medical parcel that must arrive on time. The streets are narrow in the old town. Many are one-way. Trams have priority. The van is electric, so the team also monitors range and schedules a short break. The van must be back by 12:45 to reload. In this assignment, we optimize the delivery plan for a single day for Swiss Post. To learn the basics, we use the Traveling Salesman Problem (TSP): find the fastest tour that starts and ends at the depot and visits each stop exactly once. This abstract problem gives a clean model and clear intuition about visit order and total travel time.

Part 1 (Coding exercise). As a first abstraction of the problem, we consider the Traveling Salesman Problem. You are given several locations and want a single round trip that visits each location exactly once and returns to where you started. Roads have no direction, so traveling between two locations costs the same in either way. Because every pair of locations is directly connected, the question is simply: in what order should you visit the locations to keep the total travel time as small as possible? In the undirected case, any starting point is equivalent; one may still label a specific node as the "depot" for implementation or presentation.

Let $G = (V, A)$ be a complete directed graph on $n = |V|$ locations. Vertices are $V = \{0, \ldots, n-1\}$ with node 0 denoting the depot. Vertices are $A = \{(i, j) \in V \times V : i \neq j\}$. Travel time on (i, j) is given by $c_{ij} > 0$, for all $i \neq j \in V$.

Question 1 (2 points): MTZ TSP. Formulate the Traveling Salesman Problem as a mixed integer linear program using the Miller–Tucker–Zemlin (MTZ) formulation (clearly indicate your decision variables, objective function, constraints). Then, in at most 150 words, explain why solving your MTZ model returns the shortest round trip (i.e., why the constraints eliminate subtours and the objective selects a minimum–cost tour). Write all parts of your answer in **indicated cell of the Jupyter notebook**.

Question 2 (8 points). Implement and solve two TSP models in Python with Gurobi on the starter instance provided in the Jupyter notebook on OLAT:

(a) **MTZ formulation:** implement it yourself using your model from Question 1.

(b) **DFJ formulation:** use the implementation provided in the Jupyter notebook on OLAT.

For *each* model, report:

- the optimal tour (list the nodes in visiting order, starting and ending at the depot),
- the optimal total travel time,
- the solver runtime (wall-clock seconds).

Compare the results: Are the optimal tours identical? If not, justify. If the runtimes differ, explain why. Write all answers in the same Jupyter notebook.

Part 2 (Modeling exercise). Swiss Post's real planning is far more complex than the TSP you just studied. It is a Vehicle Routing Problem (VRP): multiple vans, each with a load limit; some customers have time windows; drivers must take a break; electric vehicles (EVs) may need a quick top-up charge; there may be pickups on the way back; and parcel lockers might have to be served in a single visit. VRP adds these constraints coming on top of the touring problem. Instead of one tour as in the TSP, we have one per vehicle; we enforce load limits along each route; we keep time windows and the break; and we decide which vehicle serves each stop. In short, TSP determines *order*; VRP adds *who*, *when*, and *under which limits*.

Use case at a glance. To simplify, assume the van visits $n \in [10, 20]$ stops: (*i*) morning *B2B* customers (cafés, pharmacies, print shops); (*ii*) *B2C* apartments with intercom access; (*iii*) two *PickPost* parcel lockers; (*iv*) one *medical* delivery requiring temperature control. The van can drive about 120 km per charge and can carry $Q = 120$ volume units (VU). Service times vary: $\tau_i \in [2, 3]$ minutes for locker drops, $\tau_i \in [5, 7]$ for normal parcels, and $\tau_i \in [9, 12]$ for ID/signature. Intercoms or elevators can add 1–3 minutes.

Simple operational rules. In addition to the constraints above, Swiss Post may incorporate practical rules. A *soft time window* allows arrivals outside the interval but penalises them; a *hard time window* requires arrivals within the interval. For example:

- *Start, end, and break.* The driver must leave at 07:30 and return by 12:45. Each driver takes one 15-minute break between 10:30 and 11:30 (at any stop).
- *Time windows.* B2B: soft window 08:00–11:00 (early/late discouraged). Medical: hard window 09:30–10:15. B2C: flexible, but arrivals after 12:00 are discouraged.
- *Access and parking.* Some old-town streets close to vehicles after 10:00. Unloading are allowed for short stops (about 10 minutes).
- *Direction-dependent travel times.* There may be one-way streets, tram priority, and congestion that make travel times asymmetric, namely time from i to j can differ from j to i. Between 08:00 and 09:00, some streets are about 15% slower due to the traffic.
- *Restricted corridors.* Certain lanes (e.g., bus lanes) can shorten travel.
- *and so on ...*

No coding required. For the next part, you only need to formulate the model. No Python or Excel implementation is required. Please submit a single PDF, written in LaTeX, with your answers to Question 3 below. Respect the length limits.

Question 3 (10 points): Open modeling extension. Choose a few real-world features that matter most for the Zürich delivery tour at Swiss Post (see examples above but feel free to incorporate something). Build a model to help Swiss Post improve its day-to-day routing operations. As a tip, you may start from the TSP model discussed in Part 1. *Your answer must include:*

(a) **Scope (maximum 250 words).** Explain how you construct your model and why it is appropriate for this setting.

(b) **Data & parameters (maximum 150 words).** Introduce only the minimum necessary symbols, with units and defaults as needed. A small table is fine if helpful.

(c) **Model (maximum 1 page).** Formulate your model as a mixed-integer linear program. Clearly state the objective function, decision variables, and constraints. Briefly explain how the MILP follows from part (a).

Creativity is encouraged: pick something you care about. You do not need a perfect model to earn full points. Clear writing, well-stated assumptions, and honest trade-offs earn points. A simple but well-reasoned design can receive full points.