Final Project

Discrete optimization and decision making A.A. 2021/22

Matteo D'Alessandro

January 30, 2023

1 Introduction

The following report aims to illustrate the developed solution to a planning and multimodal transportation problem. The project work consists of the formulation of the problem as a mathematical Mixed Integer Linear Programming model, its variables, relative constraints and objective function, its implementation in Gurobi and the inclusion of a family of inequalities for the development of a Branch-and-Cut.

The first part of the report will contain a precise description of the model and the rationale behind the choices for its implementation. Secondly, we include a section presenting the model's results while tested on 3 provided instances: the resolution process is analysed with respect to solving time, gap, number of instances optimally solved, etc.. The answers to some specific question surrounding the problem in the case of specific changes are also included.

The problem's specifics, together with the Python source code for the model and the test instances can be found at `https://github.com/matteodales/DiscreteOptimization_FinalProject`.

2 Mathematical model

Before presenting the model, some remarks over the used notation are needed. In order to simplify the formulation of the model's constraints, some changes with respect to the notation used in the problem's description have been introduced. This changes also allow a very close similarity between the mathematical model and its following implementation in Gurobi.

Firstly, the developed model doesn't contain some of the aspects introduced in the problem's description. The bus mode of transportation is not included: for this reason, the set of all bus stops S is not included in the set of points describing the city. Furthermore, the total number of bikes in each bike stop is not considered: this aspect could possibly be introduced in a similar way to how the number of people in a place at the same time was, but its modeling turned out to be very complex and has therefore not been included.

Another difference concerns the tasks and the places where they are completed: we call T the set of all tasks to be completed, and we include a node in the multigraph for each task. The total set of nodes V then becomes

$V = H \cup T \cup B$. This is done to ensure that every task node will have to be visited by at most a single person a single time and removes the problem of regulating time when a person has multiple tasks to be done in the same place. We indicate by P_i the place of task i.

In some constraints, the person that has to do task $i \in T$ is indicated with n^*.

The time horizon [mintime,maxtime] has been adjusted for simplicity to the equivalent [0,tottime], with tottime = maxtime - mintime. All time windows for tasks have been adapted to this change.

Finally, in the formulated model, each bike stop is "duplicated" into a node for people taking a bike and a node for people leaving a bike ($B = B_{out} \cup B_{in}$). This is done to allow for a simpler implementation and to better regulate the evolution of the number of bikes at a stop. We consider that each bike stop is composed of a bike taking place with index i and a bike-leaving place with index $i + |B_{out}|$

A final remark: with the described setup, if we only consider walking as a mode of transportation it is immediate that, since the graph is complete and the triangle inequality is valid, it doesn't make sense for a person to visit a node more than once. This is not equally immediate in the case of bike stops: while the most common solution would be to take a bike at the beginning of the day to speed up the tour and leaving it at the end of the day, costraints on the budget and on the number of bikes at each stop could make the option of taking and leaving a bike at the same bike stop multiple times in the day optimal. This would mean that each person could visit the same bike stop multiple times, making the regulation of time more complicated.
To avoid this complication, we make the assumption that each person can take and leave a bike from the same bike stop at most one time during the day.

2.1 Variables

The model contains several binary and continuous variables that model different requested aspects of the problem.

- $w_{n,i,j}$ for $n \in N$ and $i, v \in V$: binary, 1 if person n walks from node i to node j and 0 otherwise.
- $c_{n,i,j}$ for $n \in N$ and $i, v \in V$: binary, 1 if person n bikes from node i to node j and 0 otherwise.

- $z_{n,i}$ for $n \in N$ and $i \in V$: binary, 1 if person n visits node i and 0 otherwise.
- s_t for $t \in T$: binary, 1 if task t is done as special and 0 otherwise.

The following continuous variables are introduced to describe the evolution of time for the people in the city. u_i and v_i don't have the person n as an index because they are visited by at most a single person.

- u_i for $i \in H \cup P$: continuous, time place i is reached by the person visiting it.
- v_i for $n \in N$ and $i \in H \cup P$: continuous, time place i is left by the person visiting it.
- $t_{n,i}$ for $n \in N$ and $i \in B$: continuous, time person n visits bike stop i.

We introduce binary variables that allows us to constrain the number of people simultaneously in the same place:

- $y1_{s,t}$ for $s, t \in T$: binary, 1 if task s starts before task t starts.
- $y2_{s,t}$ for $s, t \in T$: binary, 1 if task s ends before task t starts.

2.2 Constraints

We now introduce and describe the logic behind the constraints that define the model. The symbol $\mathbb{1}$ represents the indicator function.

- Walking or biking from a place to itself is not allowed.

$$w_{n,i,i} = 0 \ \forall n \in N \ \forall i \in V$$

$$c_{n,i,i} = 0 \ \forall n \in N \ \forall i \in V$$

- Each person has to go to their house and can't visit other houses.

$$z_{n,i} = \mathbb{1}_{\{i=n\}} \ \forall n \in N \ \forall i \in H$$

- Each person can only enter and exit thier house once walking.

$$\sum_{j\in V} w_{n,i,j} = z_{n,i} \ \forall n \in N \ \forall i \in H$$

$$\sum_{j\in V} w_{n,j,i} = z_{n,i} \ \forall n \in N \ \forall i \in H$$

- No one can visit a house by bike.

$$\sum_{j\in V} c_{n,i,j} = 0 \ \forall n \in N \ \forall i \in H$$

$$\sum_{j\in V} c_{n,j,i} = 0 \ \forall n \in N \ \forall i \in H$$

- A person can only visit a task place if the task is theirs.

$$z_{n,i} \leq \mathbb{1}_{\{i\in T_n\}} \ \forall n \in N \ \forall i \in T$$

- If a person visits a task place, they only do it once.

$$\sum_{j\in V} w_{n,i,j} + c_{n,i,j} = z_{n,i} \ \forall n \in N \ \forall i \in T$$

$$\sum_{j\in V} w_{n,j,i} + c_{n,j,i} = z_{n,i} \ \forall n \in N \ \forall i \in T$$

- If a person enters a task place biking they leave biking, if they enter walking they leave walking.

$$\sum_{j\in V} w_{n,i,j} = \sum_{j\in V} w_{n,j,i} \ \forall n \in N \ \forall i \in T$$

$$\sum_{j\in V} c_{n,i,j} = \sum_{j\in V} c_{n,j,i} \ \forall n \in N \ \forall i \in T$$

- If a person visits a bike-taking place, they enter walking and exit biking. If they visit a bike-leaving place the enter biking and leave walking.

$$\sum_{j\in V} c_{n,j,i} = z_{n,i} \ \forall n \in N \ \forall i \in B_{out}$$

$$\sum_{j\in V} w_{n,i,j} = z_{n,i} \ \forall n \in N \ \forall i \in B_{out}$$

$$\sum_{j \in V} w_{n,j,i} = z_{n,i} \ \forall n \in N \ \forall i \in B_{in}$$

$$\sum_{j \in V} c_{n,i,j} = z_{n,i} \ \forall n \in N \ \forall i \in B_{in}$$

- No one can enter a bike-taking place biking or a bike-leaving place walking.

$$c_{n,j,i} = 0 \ \forall n \in N \ \forall j \in V \ \forall i \in B_{out}$$
$$w_{n,i,j} = 0 \ \forall n \in N \ \forall j \in V \ \forall i \in B_{out}$$
$$w_{n,j,i} = 0 \ \forall n \in N \ \forall j \in V \ \forall i \in B_{in}$$
$$c_{n,i,j} = 0 \ \forall n \in N \ \forall j \in V \ \forall i \in B_{in}$$

- A task can't be done as special if it isn't special.

$$s_i \leq \mathbb{1}_{\{i \in T^*\}}$$

- A task can't be done as special if the person doesn't do it.

$$s_i \leq z_{i,n^*}$$

- Constraint on the number of mode of transportation changes for each person. The number of changes is computed as the number of bike-leaving and bike-taking places visited in total.

$$\sum_{i \in B} z_{n,i} \leq K_n \ \forall n \in N$$

The following constraints refer to variables u_i, v_i and regulate the evolution of time for houses and task places. They make use of the the time window [0,tottime] that defines the full day in the city. These constraints also regulate the completion of tasks in the related time-window and their duration.

- The time exiting the house precedes the time returning to it.

$$u_i \geq v_i \ \forall i \in H$$

- The enter and exit time for the task place are both 0 if the place is not visited.

$$u_t \leq tottime * z_{n^*,t} \ \forall t \in T$$
$$v_t \leq tottime * z_{n^*,t} \ \forall t \in T$$

- These constraints regulate the evolution of time when travelling between houses and task places. We only consider this constraints for places that person n can go to, so the places of their tasks (T_n) and their house (n).

$$v_i - u_j + t^w_{i,j} \leq tottime(1 - w_{n,i,j}) \ \forall n \in N \ \forall i, j \in T_n \cup \{n\}$$

$$v_i - u_j + t^c_{i,j} \leq tottime(1 - c_{n,i,j}) \ \forall n \in N \ \forall i, j \in T_n$$

- These constraints impose that the task has to be started in the time window and that the person stays in the task place enough time to complete it.

$$u_t \geq a_t * z_{n^*,t} - tottime * s_t \ \forall t \in T$$

$$u_t \leq b_t * z_{n^*,t} + tottime * s_t \ \forall t \in T$$

$$v_t \geq u - t + \tau_t * z_{n^*,t} + \tau^+_t * s_t$$

We also go on to include the constraints that regulate the evolution of time for bike stops. The constraints are similar but also deal with the fact that each bike stop can be visited by multiple people.

- The visiting time is zero if the place is not visited.

$$t_{n,i} \leq tottime * z_{n,i} \ \forall n \in N \ \forall i \in B$$

- Regulating time when going from a place to a bike stop.

$$v_i - t_{n,j} + t^w_{i,j} \leq tottime * (1 - w_{n,i,j}) \ \forall n \in N \ \forall i \in H \cup T \ \forall j \in B_{out}$$

$$v_i - t_{n,j} + t^c_{i,j} \leq tottime * (1 - c_{n,i,j}) \ \forall n \in N \ \forall i \in H \cup T \ \forall j \in B_{in}$$

- Regulating time when going from a bike stop to a place.

$$t_{n,i} - u_j + t^w_{i,j} \leq tottime * (1 - w_{n,i,j}) \ \forall n \in N \ \forall j \in H \cup T \ \forall i \in B_{in}$$

$$t_{n,i} - u_j + t^c_{i,j} \leq tottime * (1 - c_{n,i,j}) \ \forall n \in N \ \forall j \in H \cup T \ \forall i \in B_{out}$$

- Regulating time when going from a bike stop to a bike stop.

$$t_{n,i} - t_{n,j} + t^w_{i,j} \leq tottime * (1 - w_{n,i,j}) \ \forall n \in N \ \forall i \in B_{in} \ \forall j \in B_{out}$$

$$t_{n,i} - t_{n,j} + t^c_{i,j} \leq tottime * (1 - c_{n,i,j}) \ \forall n \in N \ \forall i \in B_{out} \ \forall j \in B_{in}$$

The following constraints regulate variables $y1$ and $y2$ and the number of people in the same place at the same time.

- Every task doesn't start or end before itself.

$$y1_{i,i} = 0 \ \forall i \in T$$
$$y2_{i,i} = 0 \ \forall i \in T$$

- If task i and j don't have to be completed in the same place then the variables are imposed equal to zero.

$$y1_{i,j} = 0 \ \forall i \in T \ \forall j \in T \text{ if } P_i \neq P_j$$
$$y2_{i,j} = 0 \ \forall i \in T \ \forall j \in T \text{ if } P_i \neq P_j$$

- If task i ends before task j then it has to start before task j.

$$y2_{i,j} \leq y1_{i,j} \forall i \in T \ \forall j \in T$$

- If $i \neq j$ then either i starts before j or j before i.

$$y1_{i,j} + y1_{j,i} = 1 \forall i \in T \ \forall j \in T \text{ if } P_i = P_j$$

- The following constraints enforce the definition of $y1$ and $y2$: $y1_{i,j}$ is 1 if and only if task i starts before task j starts, $y2_{i,j}$ is 1 if and only if task i ends before task j starts.

$$u_j - u_i \leq tottime * y1_{i,j} \forall i \in T \ \forall j \in T \text{ if } P_i = P_j$$
$$u_i - u_j \leq tottime * (1 - y1_{i,j}) \forall i \in T \ \forall j \in T \text{ if } P_i = P_j$$
$$u_j - v_i \leq tottime * y2_{i,j} \forall i \in T \ \forall j \in T \text{ if } P_i = P_j$$
$$v_i - u_j \leq tottime * (1 - y2_{i,j}) \forall i \in T \ \forall j \in T \text{ if } P_i = P_j$$

- The total number of people that are already in place P when task j is started can be obtained by the sum of $y1_{i,j} - y2_{i,j}$ between the tasks i that are done in the same place. The following constraint imposes that the number of people in the same place at the same time doesn't go over the maximum.

$$\sum_{i|P_i=P_j} y1_{i,j} - y2_{i,j} \leq N_{P_j} - 1 \ \forall j \in T$$

The final constraint deals with the budget for each person and imposes that, considering the cost of biking and the expense for each completed task, the maximum budget is not surpassed.

$$\sum_{t \in T} c_t z_{n,t} + C_{bike}(\sum_{i \in B_{in}} t_{n,i} - \sum_{i \in B_{out}} t_{n,i}) \leq W_n \ \forall n \in N$$

2.3 Objective function

The model's objective function is introduced:

$$min \;\; R\sum_{n\in H}(u_n - v_n) + P\sum_{t\in T}Q_t(1-z_t) - F\sum_{n\in N}\sum_{i\in V}\sum_{j\in V}(f^W_{i,j}w_{n,i,j} + f^C_{i,j}c_{n,i,j})$$

The first term of the quantity to minimize is the total time that the people in the city take to complete their journey, so the time they get back minus the time they leave. The second term represents the sum of penalties accumulated by the people for not completing their tasks. The last term is the total sum of the fitness associated to each arc that is travelled through. Parameters R, P and F regulate the relative importance given to the three terms and, unless otherwise specified, are considered fixed to 1.

3 Gurobi implementation

To develop an implementation for the model, a Python code was produced. All of the code can once again be found at `https://github.com/matteodales/DiscreteOptimization_FinalProject`.

The first step was creating the code to import the relevant data from an instance. The instances were provided as a `txt` file, which was imported and read by the code in order to extract the information about the vertices of the multigraph, their respective features and the traveling times and related fitness between them.

The mathematical model was then implemented using the Gurobi Optimizer package. The variables and constraints introduced in the model reflect the ones described in Section 2.

3.1 Branch-and-Cut

The project description also called for the development of a Branch-and-Cut through the introduction of a valid family of inequalities. This was achieved by relaxing the constraints related to the number of people simultaneously in a same place. The problem is solved without these constraints: when a MIP solution is found, we check if it violates any of these constraints and, in case it does, the model is solved again after their introduction.

This procedure is implemented by developing a code that detects wheter at any time point there are more people than allowed in a single place P: in that case the following constraints are introduced.

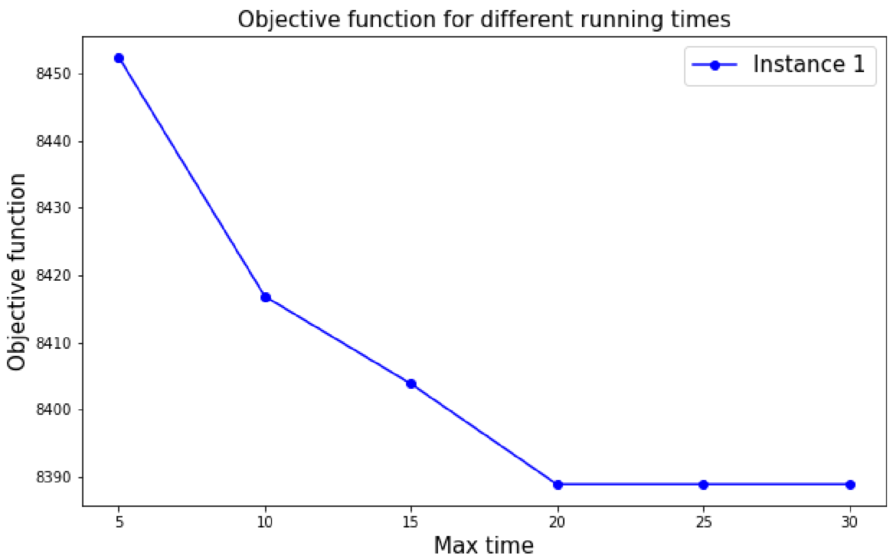
$$u_j - u_i \leq tottime * y1_{i,j} \forall i \in P \; \forall j \in P$$

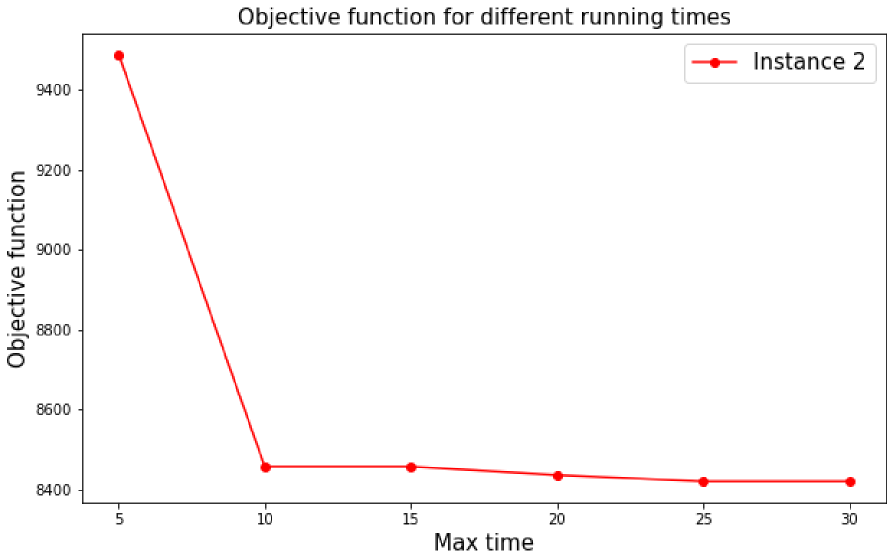
$$u_i - u_j \leq tottime * (1 - y1_{i,j}) \forall i \in P \; \forall j \in P$$
$$u_j - v_i \leq tottime * y2_{i,j} \forall i \in P \; \forall j \in P$$
$$v_i - u_j \leq tottime * (1 - y2_{i,j}) \forall i \in P \; \forall j \in P$$
$$\sum_{i \in P} y1_{i,j} - y2_{i,j} \leq N_P - 1 \; \forall j \in P$$

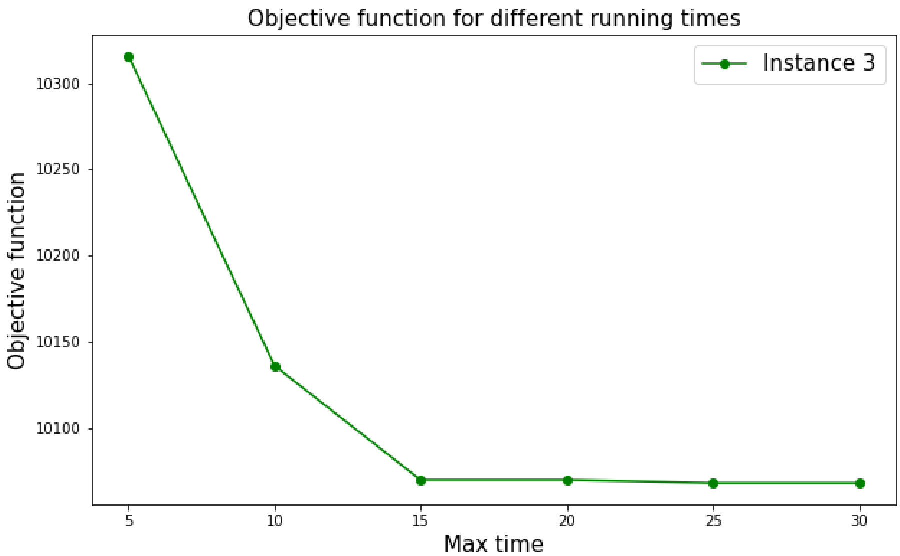
4 Results

This last section reports some details of the resolution process for the three instances provided and considering the model described above. Given the size of the problem in terms of variables and constraints, none of the instances were solved to optimality. To obtain some results on performance, a time limit was introduced on the solver and the best integer solution found up to that point was studied.

The following graphs represent the performance of the solver in terms of the objective function for growing running time. All three instances are included.

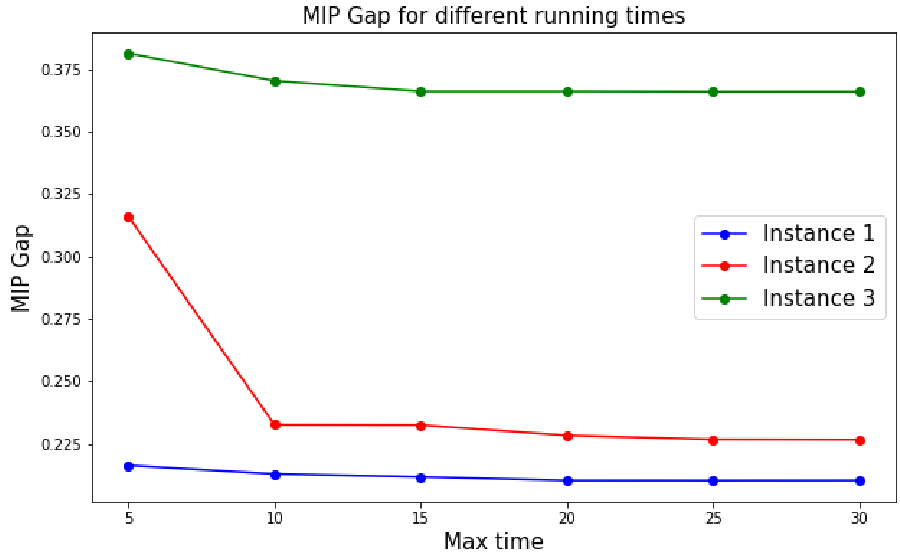






As expected the values for the objective function are decreasing when the model is allowed for a longer optimization. However, for times over 10/15 minutes, the improvement obtained is very small.

We also go on to consider the MIP Gap, which shows how close the incumbent solution is relatively to the MIP objective bound with a relative value between 0 and 1. The results of this analysis for all three instances are presented in the next figure.



4.1 Questions

Finally, some questions on possible modifications of the model are analysed.

1. **How do the solutions change if the maximum number of mode of transportation changes K is set to 2 for every person $n \in N$?**

   For most people the best route would be to take a bike at the beginning of the day, travel with it and leave it at the end of the day. Other people would only walk around or take the bus just once.

   This was studied in practice for the first instance: as expected, 9 out of the 10 people considered take the bike to travel between their task places and leave it at the end of the day.

2. **How do the solutions change when this K increases?**

   The solutions become more complex, with people being able to change their modes of transportation possibly indefinitely, and only constrained

K	Objective Function
0	9100.83
2	8130.62
4	8203.86
6	8333.38

by their budget. Based on the instance, people could take the bus multiple times or change between biking and using the bus to optimize their timing. These changes in solutions would only happen when K reaches an even number (2 to 4, to 6, etc.), since the number of transportation changes could only ever be even: each person has to start the day walking from their home and end it walking back. In general, as K increases, the objective function would reduce, since we are considering a more relaxed constraint and allowing more solutions to be viable.

This was studied in practice in the case of the first given instance, with a time limit of 10 minutes. The results are presented in the following table. As we can see the objective function initially reduces for K going from 0 to 2, but for K going to 4 and 6 unexpectedly increases again. This is probably due to the fixed time in calculations rather than an actual increase in the optimum.

3. **How do the solutions change when soft constraints are not allowed anymore?**

   In the absence of soft constraints, all tasks in a task list would have to be completed: this could create an infeasible problem in case the time windows of different tasks didn't match or the budget couldn't allow for all of them to be completed. In general, the final objective function would increase, since we are restricting the model and reducing the set of allowed solutions. This was not studied in practice.

4. **How do the solutions change if the goal becomes to only maximize the overall fitness score?**

   In this case, the penalties associated with not doing tasks would be ignored. Total time would also not be a factor. If for a person there exists a route starting and terminating at their home that produces positive fitness, this would be repeated until reaching the maximum time. Otherwise, the person wouldn't move at all.

This conjecture on the form of the solution is hard to verify since, given the way our model is formulated, a route can't be repeated more than once.