UNIVERSITÀ di VERONA
Dipartimento di INFORMATICA

Discrete Optimization and Decision Making

Final project

Master's degree in Data Science
Academic Year 2021/2022

Alice Raffaele (University of Verona)
Roberto Zanotti, Ph.D. (University of Brescia)

Planning and multimodal transportation

It is a new day in Verona, and a set $\mathcal{N} := \{1, \dots, N\}$ of people is ready to perform all their errands as efficient as they can be.

Indeed, each person $n \in \mathcal{N}$, who starts from a known point H_n (e.g., their home) with a given amount of money W_n, has a certain set $\mathcal{T}_n$ of tasks to do. Each task t may cost c_t €, has to start during the related time window $[a_t, b_t]$, and generally lasts τ_t minutes. Anyway, for some special tasks $t^* \in \mathcal{T}_n^* \subseteq \mathcal{T}_n$, either t^* has to start in the related time window, or it can be started anytime. However, in this latter case, the duration of t^* will be increased by a certain amount of minutes.

Each task t occurs in a different place p. Let $\mathcal{P} := \{1, \dots, P\}$ be the set of all places of the tasks of all people in $\mathcal{N}$ (e.g., shops, squares, bars, facilities, etc.). Each place $p \in \mathcal{P}$ is described by a latitude value lat_p and a longitude value lon_p. Also, a place p cannot be visited by more than N_p people at the same time.

To reach the places associated to their tasks $\mathcal{T}_n$, each person n is allowed to either walk, or to ride a bike offered by the bike-sharing company in Verona, or also to take a bus, according to the distribution of the bike-sharing stations and the bus lines in the city.

Let $\mathcal{M} := \{Walking, Cycling, Bus\}$ be the set of modes of transportation. Each person $n \in \mathcal{N}$ can change their mode of transportation at most K_n times and, in the end, n must return to their starting point H_n. A penalty Q_t must be paid for a task $t \in \mathcal{T}_n$ not performed.

In particular, let $\mathcal{B} := \{1, \ldots, B\}$ be the set of bike stations. For each bike station $b \in \mathcal{B}$, the number α_b of bikes available at the beginning of the day is known, as well as the number β_b of free spots where to leave a bike (of course, if all spots are available, the maximum capacity of a bike station b is $N_b := \alpha_b + \beta_b$). Similarly to any place, each bike station b is associated with a latitude value lat_b and a longitude value lon_b. Riding a bike costs C_{bike} €/min.

Regarding buses, a set $\mathcal{L} := \{1, \dots, L\}$ of lines is available in the city. Each line ℓ is composed by a set of stops $\mathcal{S}_\ell := \{D_\ell, \dots, S_\ell\}$, and starts from a certain depot D_ℓ at time δ. Then, there is a ride every δ minutes. Each stop $s \in \mathcal{S}_\ell$ is associated with a latitude value lat_s and a longitude value lon_s. The maximum number of people who can be on a bus at the same time is N_ℓ.

No two lines $\mathcal{S}_{\ell_1}$ and $\mathcal{S}_{\ell_2}$ have two identical consecutive stops. Anyway, some lines $\mathcal{S}_{\ell_1}$ and $\mathcal{S}_{\ell_2}$ may intersect at some stops (i.e., $\mathcal{S}_{\ell_1} \cap \mathcal{S}_{\ell_2} \neq \emptyset$).

Every time that a person n chooses the bus to change their mode of transportation, they pay C_{bus} €. If n just chooses to change the bus line in a stop, without changing the mode of transportation, they do not have to pay another ride.

Let $\mathcal{H}$ be the set of all starting points H_n, for every $n \in \mathcal{N}$. Let $\mathcal{S}$ be the set of all bus stops, for every line $\ell \in \mathcal{L}$.

Let $\mathcal{G} := (\mathcal{V}, \mathcal{A})$ the directed weighted multi-graph such that $\mathcal{V} := \mathcal{H} \cup \mathcal{P} \cup \mathcal{B} \cup \mathcal{S}$ is the set of all points in the city. Let $\mathcal{A} := \{(i, j, m) \mid i, j \in V, m \in \mathcal{M}\}$ be the set of all arcs connecting point i to point j by using mode of transportation m. Each arc is associated with a travelling time $t_{i,j}^m$ and with a fitness coefficient $f_{i,j}^m \in [-1, 1]$, which represents a sort of "health gain" or "health loss", according to the mode of transportation chosen.

Determine the tours that all people have to do in order to perform their tasks, by considering the different modes of transportation available and all the constraints, in order to minimize the total travelled time by also taking into account the overall fitness score.

Project requirements

To discuss the final project, **every group** will have to:

1. formulate the problem as a **mathematical model** by applying the right paradigm among Linear Programming, Integer Linear Programming, and Mixed Integer Linear Programming;
2. **implement** the mathematical model in Gurobi;
3. **develop** a Branch-and-Cut by including at least one family of valid inequalities.

Additional requirements

The following task will be mandatory for groups composed by more than one person, and optional for other groups.

4. Either:
   4.1 **adapt a metaheuristic/matheuristic** to the problem (the Java code for the basic version of Kernel Search will be provided),
   4.2 or **add** some **further families** of valid inequalities to the Branch-and-Cut formulation,

   and compare the model developed before with the new implementation.

Instances and computational experiments

Every group will be given 10 benchmark instances (5 small and 5 larger), to test the proposed model and algorithm.

In particular, you will have to perform some computational experiments in order to:

5. **report** details about the resolution process (e.g., solving time, gap, number of instances optimally solved, etc.);
6. **answer** at least three of the following questions:
   - 6.1 how do the solutions change if the maximum number of changes of mode of transportation K is set to 2, for every person $n \in \mathcal{N}$?
   - 6.2 how do the solutions change when this K increases?
   - 6.3 how do the solutions change when soft constraints are not allowed anymore?
   - 6.4 how do the solutions change if the goal becomes to only maximize the overall fitness score?

Reproducibility

7. At the end of the project, every group should provide the source code and, in case of Java, an executable .jar file, that takes as arguments:
    - a time limit, expressed in seconds (`-t "timelimit"`);
    - the name of the instance file to solve (`-f "filename"`).

   Once the time limit is reached, the executable file or the script has to write to disk the best solution available (with the objective function value in the first line), and terminate. The executable or the script will be tested on all the provided benchmark instances and on a set of new benchmark instances that has not been made available.

Project evaluation

The oral exam consists in a **presentation** made by the group, followed (or alternated) by some questions.

It is strongly suggested to prepare some **slides** to describe the model, the Branch-and-Cut approach implemented, if required the matheuristic or the other cuts added, and all the results obtained.

At least **a week before** the presentation, the group should provide the source code and the executable of the algorithm, alongside with a **report** containing a precise description of the implemented model and algorithm (step by step), carefully analyzing the rationale behind the choices made in each step.

Moreover, the report should contain a **detailed analysis** of the computational results obtained. Since a major goal of your research is to study the behavior of the proposed algorithm, a special care has to be devoted to this part.

Recap on the mandatory requirements

Groups composed of more than one person:
requirements 1, 2, 3, 4.1 or 4.2, 5, at least three among 6.1, 6.2, 6.3, and 6.4., and 7.

Groups composed of one person only:
requirements 1, 2, 3, 5, at least three among 6.1, 6.2, 6.3, and 6.4., and 7.

You can freely use any information you might obtain from the web and which can help you achieve the project requirements.