Course Project:
A rich vehicle routing problem to assist local governments during the COVID-19 pandemic

Discrete Optimization and Decision Making: a.y. 2020/2021

1 Introduction

The spread of the SARS-CoV-2 virus and its rapid transformation into a pandemic have led several governments, including the Italian one, to adopt increasingly restrictive measures on the free movement of citizens and on all forms of social aggregation, until the inevitable total lockdown. Mitigation and containment policies, needed to overcome the medical-health crisis, have had catastrophic social and economic consequences. In Italy, the GDP has shrunk by more than 6 percentage points compared to January estimates. The forced closure of many commercial and service activities has caused impoverishment and failures, whereas it has favored alternative distribution channels (e-commerce, home deliveries). From a social point of view, forced isolation has particularly affected the most vulnerable family units such as those composed of elderly or disabled people, single parents with children, or destitute ones.

The second surge in the COVID-19 cases has shown that the pandemic was far from being over in the second half of 2020, but Governments across the world tried to avoid a total lockdown situation, given that it would have been much harder to sustain at that time rather than it had been months before. A critical aspect that a local government

(*DECISION MAKER*) has to face is the situation of *CRITICAL UNITS*, i.e., vulnerable family units that are in mandatory isolation due to a positive COVID-19 test or simply limited in their movements due to age or other illnesses. These units require assistance for basic tasks, such as picking up groceries, medical appointments, etc., that needs to be performed in local shops, commercial activities, or public buildings (ACTIVITIES). We assume that the DECISION MAKER provides a team (PROXY) that is tasked with taking care of these units, by performing the actions they need, or by just doing regular check-ups.

The goal of the project is the development of a mathematical model and a solution algorithm (matheuristic/metaheuristic approach) for the problem.

2 Problem Description and Notation

The problem is defined over a complete directed graph $G = \{V, H\}$ with node set $V = N \cup \{0, n+1\}$ and arc set $H = \{(i, j) : i, j \in V, i \neq j\}$. $N = \{1, \dots, n\}$ represents the nodes corresponding to UNITS and ACTIVITIES, whereas 0 and $n+1$ represent initial and final depots in which proxies are located and where they have to go at the end of the day. A travel time t_{ij} is associated with each arc $(i, j) \in H$. Let $K = \{1, \dots, e\}$ be a fleet of homogeneous vehicles, corresponding to the workers available to perform tasks for the UNITS as PROXIES. Let $T = \{1, \dots, p\}$ be the set of time slots into which the day is partitioned and let $R = \{1, \dots, v\}$ be the set of tasks that have to be performed.In general, each task is defined by a node $a \in N$ in which the task has to be performed (usually called *pickup node*) and a node $b \in N$ in which the result of the task have to be delivered (*delivery node*). In certain cases, a task can be defined by an additional node (e.g., a person that needs to go to a medical appointment has to be picked up in a node, brought to the appointment node, and brought back) or by a single node (periodic checkups). Additionally, a time slot $t \in T$ and a service time ψ are associated with each request. A time limit θ_r for the duration of the task (time between the first and the last node of a task) can be imposed. ψ is the time that a vehicle has to spend in the pickup node to perform the task. ψ is constituted by two parts:

a fixed part α and a variable part β. When a single task is performed by a vehicle in a node, the time required is $\psi = \alpha + \beta$. In certain cases, the nodes associated with a task need to be visited consecutively (i.e., without visiting any other node in between) **(A)** or they need to be visited without having any other person onboard **(B)**. When a vehicle performs more than one task in the same node, the time needed for that macro task becomes $\psi = \alpha + f(\beta, w)$, where w is the number of tasks performed in the node by the vehicle, and f is a function that determines the time saving with respect to the time $w\beta$ required to serve each task independently. As already mentioned, when a single person performs multiple tasks in the same ACTIVITY, he/she usually requires less time than several people performing the same total number of tasks in the same ACTIVITY. We assume that a part of the time to perform a task cannot be reduced, even if several tasks are performed by the same vehicle. Going back to our groceries example, the time required to checkout is not much different if the same amount of groceries is bought by two people or by one. A part of the time required to perform a task can instead be reduced if multiple tasks are performed by the same vehicle. Considering once again our groceries example, you only pick up a shopping cart once, even if you buy groceries for multiple people. Each task is associated with a time SLOT where the task has to be executed. A penalty is paid each time the task is scheduled outside the selected time slot, and it is equivalent to ρ multiplied by the number of minutes that the starting time of the task exceeds or precedes the ending time or starting time of the time slot, respectively. A penalty is also paid when a task cannot be performed at all. The goal of the problem is to identify a set of routes, one for each vehicle, that minimizes the total travel time and the total penalties.

3 Requirements

You have to achieve the following project requirements:

- develop a mathematical model for the problem that accurately encompasses all the defined constraints and features of the problem (only one of **A** and **B** is needed).

- introduce and implement a metaheuristic or matheuristic method choosing between the ones presented during the course for the problem. Combinations of different approaches are more than welcome. The developed approach can make use of Gurobi for solving subproblems. The Java code for the basic version of Kernel Search will be provided, and you can adapt it to the current problem, but it is not a requirement.

- Test the proposed model and algorithm on benchmark instances that will be provided.

At the end of the project, the group should provide an executable jar file that takes a time limit (expressed in seconds) (*-t "timeLimit"*) and the name of the instance file to solve (*-f "fileName"*) as arguments. Once the time limit is reached, the executable has to write to disk the best solution available (with the objective function value in the first line) and terminate.
The executable will be tested on all the provided benchmark instances and on a set of new benchmark instances that has not been made available.

4 Project evaluation

The oral exam consists in a presentation made by the group, followed (or interrupted) by some questions. It is strongly suggested to prepare some slides to describe the model, the algorithm, and 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. Variants that are not included in the final algorithm can be described as well, if deemed interesting. 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 the analysis of the results. Finally, the identification of particular problem properties will be considered as a plus in the project evaluation.

You can freely use any information you might obtain from the web and that can help you in achieving the project requirements.