Fleet management of autonomous vehicles: VIPAFLEET

Sahar Bsaybes, Alain Quilliot and Annegret K. Wagler
Université Blaise Pascal (Clermont-Ferrand II)
Laboratoire d’Informatique, de Modélisation et d’Optimisation des Systèmes
(LIMOS, UMR 6158 CNRS)
BP 10125, 63173 Aubière Cedex, France
{bsaybes,quilliot,wagler}@isima.fr

keywords: Online transportation problem, fleet management, autonomous vehicles.

1 Introduction

The project VIPAFLEET aims at contributing to sustainable mobility through the development of innovative urban mobility solutions by means of fleets of Individual Public Autonomous Vehicles (VIPA) allowing passenger transport in complex environments and in closed sites [4]. Note that “autonomous vehicles” means that neither a driver nor an infrastructure is required to operate a VIPA, which reflects the innovative property of the project from the technical side.

Currently, a fleet of VIPAs is used in the industrial site of Michelin to transport employees and visitors between parkings, buildings and from or to the restaurant at lunch breaks (but this service can also be exploited in many other contexts: medical complexes, campuses, business centers, airports etc.). The fleet is distributed at specified stations in an (industrial) area to supply internal transportation, and a VIPA can operate in three transportation modes:

- Tram mode: VIPAs continuously run on predefined lines or circuits in a predefined direction and stop at a station if requested to let users enter or leave.
- Elevator mode: VIPAs run on predefined lines or circuits, but react on customer requests (to enter or leave a car at a station). The elevator mode can be interpreted as an Online Traveling Salesman Problem in a specific network.
- Taxi mode: users book their transport requests (from a start to a destination station with a start and an arrival time) in advance or in real time, which can be interpreted as an Online Dial-a-Ride Problem with time windows in a specific network.

Users can call a VIPA from a station by the help of a call-box installed at a station or book their request in advance or in real time by mobile or web applications. Our aim is to develop and install a Dynamic Fleet Management System that allows the operator to switch between different network topologies, transportation modes and according online algorithms within the different periods of the day in order to react to changing demands evolving during the day, with the objective to satisfy all demands in a best possible way. For that we:

- develop suitable algorithms for each mode and subnetwork;
- analyse best-case and worst-case behavior of these algorithms for different demands;
- cluster the demands into subproblems in such a way that, for each subproblem, a suitable subnetwork and a suitable algorithm can be proposed leading to a globally good solution (transportation schedule).
Note that the latter is the innovative idea and challenging problem in the project: in many existing approaches, clustering in subproblems has been proposed (e.g. in [1]), but the same solution technique is applied to each subproblem. Here, we intend the subproblems to be treated by different techniques to improve the global quality of the solution.

2 Problem description and model

We embed the VIPAFLEET management problem in the framework of a metric task system. We encode the closed site where the VIPAFLEET system is running as a metric space $M = (V, d)$ induced by a connected network $G = (V, E)$, where the nodes correspond to stations, arcs to their physical links in the closed site, and the distance $d$ between two nodes $v_i, v_j \in V$ is the length of a shortest path from $v_i$ to $v_j$. In $V$, we have a distinguished origin $v_o \in V$, the depot of the system where all VIPAs are parked when the system is not running, i.e., outside a certain time horizon $[0, T]$. The task is to satisfy the sequence $R$ of customer requests by transporting people within the metric space. Any request $r_j$ is defined as a 6-tuple $r_j = (x_j, y_j, p_j, q_j, t_j, \text{load})$ where $x_j$ is the origin node, $y_j$ is the destination node, $t_j$ is the release date, $p_j$ is the earliest time $r_j$ can start being processed, $q_j$ is the latest possible arrival time, and $\text{load}$ specifies the number of passengers. Note that a request $r_j$ can sometimes have missing information according to the mode wherein a VIPA is circulating. In order to fulfill these tasks, we let a fleet of VIPAs (one or many, each having a capacity for $L$ passengers) circulate in the network inducing the metric space. More precisely we determine a feasible transportation schedule consisting of one tour per VIPA. Hereby, a tour can be interpreted as

- an Eulerian path in the network $G$ or a directed path in the time-expanded version of $G$ over the whole time horizon $[0, T]$ if the VIPA is running all day long;
- a collection of such paths if the VIPA is running only during some periods in $[0, T]$ with high volume of requests;
- an empty tour if the VIPA stays in the depot.

This will be addressed by dividing $[0, T]$ in different periods according to the volume and kind of the requests, and by providing specific solutions within each period. The global goal is to provide a feasible transportation schedule over the whole time horizon satisfying all requests by minimizing the waiting times for the users or minimizing the total tour length (Dynamic Fleet Management Problem).

In addition, depending on the policy of the operator of such a system, different side constraints have to be obeyed in order to construct a feasible transportation schedule. If call-boxes only allow to call a VIPA to the station, but not to indicate immediately the destination, we are in the situation that each request is split into two (one pickup request from a call-box, one delivery request from a VIPA after the passengers have entered). The choice of the call-box has an impact on the request type and therefore on the online transportation problem behind. If two or many VIPAs circulate on the same subnetwork we have to handle many constraints as: meeting of two vehicles on a station or an arc, blocking the route of a VIPA by another one waiting at a station (if no two VIPAs are allowed to enter the same node or arc). In addition, we need to consider the events of breakdown or discharge of a vehicle.

Note that we have to take also into account that there might be technical problems with the server, database or the communication network between the stations, VIPAs and the central server. Therefore, we also need to ensure that the system can operate in a stand-alone manner.

Based on all the above technical features and properties that have an impact on the feasibility of the transportation schedule, we can cluster the requests into subproblems, apply to each subproblem a certain algorithm, and check the results in terms of feasibility and performance. The choice of the topology in the industrial site where the VIPAs will operate is dynamic and will change over time according to the technical features and properties. Hereby, for a certain period $[t, t'] \subseteq [0, T]$, we define a metric subspace $M' = (V', d')$ induced by a subnetwork
\( G' = (V', E') \) of \( G \), where a subset of nodes and arcs of the network is active. Let us consider four typical scenarios that can happen while operating a fleet in an industrial site, based on some preliminary studies of the transport requests within the site:

- **Morning/evening:** main requests are between parkings and buildings, thus we partition the network into disjoint unidirected cycles as subnetworks such that each subnetwork contains at least one parking. The cycles take the size of parkings and number of employees in the served buildings into account; accordingly assign one or several VIPAs to every cycle operating in tram mode.

- **Lunch time:** main requests are between buildings and the restaurant, thus we consider, e.g., a spanning star of bidirected lines all meeting in a central station, one VIPA on each line operating in elevator mode. here, all requests have a common destination: the center of the star.

- **Other periods:** unspecified requests without common origins or common destinations. For that, we consider, e.g., two unidirected Hamilton cycles with opposite directions where several VIPAs operate in tram mode, a collection of bidirected lines and unidirected cycles, all meeting in a central station, where several VIPAs operate in elevator or tram mode, or all VIPAs operating in taxi mode on the whole network (leading to a general k-server Pickup and Delivery Problem.

- **Emergency case:** we consider two unidirected Hamilton cycles with opposite directions, half of the fleet operating on each cycle in tram mode.

## 3 Algorithms and competitive analysis

We propose the following combinations of modes, subnetworks and algorithms:

**Tram mode:** VIPAs operate on a unidirected cycle using the algorithms “Stop If Requested” (SIR) or “Start if fully loaded” (SIF) in case that all requests have the same origin.

**Elevator mode:** one VIPA operates on a bidirected path using the algorithm “Move-Right-If-Necessary” (MRIN) [2]. If a new request is released and the request is to the right of the current position of the VIPA operated by MRIN, then it starts to move right and continues as long as there are yet unserved requests to the right of the VIPA. If there are no more unserved requests to the right, then the VIPA moves towards the origin.

**Taxi mode:** VIPAs operate on a connected subnetwork using an algorithm for resolving DARP, e.g., based on insertion techniques [3] or column generation [6].

It is standard to evaluate the quality of online algorithms by competitive analysis, that can be viewed as a game between an online algorithm and a malicious adversary who tries to select a worst-case request sequence which maximizes the ratio between the online and the offline cost. We are interested in analyzing the performance of each scenario.

While it is known that MRIN is 3/2-competitive for Online TSP [2] and that there does not exist a strictly competitive deterministic online algorithm for the Online DARP (taxi mode) w.r.t minimizing the maximum waiting time [5], the scenarios using tram mode are new and subject of our analysis w.r.t minimizing the total tour length.

**Theorem 1** For one or several VIPAs operating in tram mode on a unidirected cycle, SIF yields the optimal solution if all requests have a common origin (morning scenario).

This provides a globally optimal solution if the cycles are designed such that each building can be reached via a shortest path from the nearest parking and some further technical conditions hold.

**Theorem 2** SIR is \( L \cdot C \)-competitive in general on a unidirected cycle and 2-competitive with one VIPA (only allowed to wait at the origin), and the waiting time of any request is bounded by \( C \), where \( L \) and \( C \) stand for the capacity of a VIPA and the length of the cycle.
4 Concluding Remarks

The vehicle routing problems integrating constraints on autonomy are new in the field of operational research but they are the future mobility. These vehicles, which are intended to be used as a fleet in order to form a transport demand service, need to be effective also considering to their management. The future works are to develop a simulation tool for the master problem that encompasses all possible scenarios and subproblems that change over time, so that we can test them in practice. After that we can analyse the scenarios in terms of competitive analysis and get an idea about what strategies are promising for real experimentations. Competitive analysis has been one of the main tools for deriving worst-case bounds on the performance of algorithms but an online algorithm having the best competitive ratio in theory may reach the worst case more frequently in practice with a certain topology. That’s why we are not only interested in the “worst-case” but also in the “best-case” performance of the algorithms, thus we need to determine properties which govern the behavior of each chosen algorithm and define the cases where it can be applied and give the best results in terms of performance.

References


