



**HAL**  
open science

# Fleet management for autonomous vehicles using flows in time-expanded networks

Sahar Bsaybes, Alain Quilliot, Annegret K Wagler

► **To cite this version:**

Sahar Bsaybes, Alain Quilliot, Annegret K Wagler. Fleet management for autonomous vehicles using flows in time-expanded networks. *Electronic Notes in Discrete Mathematics*, 2017, 62, pp.255-260. hal-02045780

**HAL Id: hal-02045780**

**<https://hal.science/hal-02045780>**

Submitted on 22 Feb 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Fleet management for autonomous vehicles using flows in time-expanded networks

Sahar Bsaybes, Alain Quilliot, Annegret K. Wagler<sup>1,2,3</sup>

*University Clermont Auvergne (LIMOS, UMR 6158 CNRS), Clermont-Ferrand,  
France*

---

## Abstract

The VIPAFLEET project aims at developing a framework to manage a fleet of Individual Public Autonomous Vehicles (VIPA). We consider a fleet of such cars distributed at specified stations in an industrial area to supply internal transportation, where the cars can be used in different modes of circulation (tram mode, elevator mode, taxi mode). We treat in this paper the pickup and delivery problem related to the taxi mode by means of flows in time expanded networks. We compute optimal offline solutions, propose a replan strategy for the online situation, and evaluate its performance in comparison with the optimal offline solution.

*Keywords:* fleet management, offline and online pickup and delivery problem

---

The project VIPAFLEET [5] 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 closed sites like industrial areas, medical complexes or airports.

---

<sup>1</sup> Email: [sahar.bsaybes@uca.fr](mailto:sahar.bsaybes@uca.fr)

<sup>2</sup> Email: [alain.quilliot@uca.fr](mailto:alain.quilliot@uca.fr)

<sup>3</sup> Email: [annegret.wagler@uca.fr](mailto:annegret.wagler@uca.fr)

This work was funded by the French National Research Agency, the European Commission (Feder funds) and the Région Auvergne in the Framework of the LabEx IMobS3.

A VIPA is an “autonomous vehicle” that does not require a driver nor an infrastructure to operate, developed by Easymile and Ligier thanks to innovative computer vision guidance technologies [2,3].

A fleet of VIPAs shall be used in an industrial site to transport employees and visitors e.g. between parkings, buildings and from or to a restaurant. The fleet is distributed at specified stations to supply internal transportation, and a VIPA can operate in three different circulation modes:

- *Tram mode*: VIPAs continuously run on predefined cycles in a predefined direction and stop at a station if requested to let users enter or leave.
- *Elevator mode*: VIPAs run on lines and react to requests by changing their driving direction and moving to a station to let users enter or leave.
- *Taxi mode*: VIPAs run on a connected network to serve transport requests (from any start to any destination station within given time windows).

This leads to an Online Pickup-and-Delivery Problem in a metric space encoding the considered site, since transport requests are released over time. In [1], tram mode and elevator mode are treated, here we focus on the taxi mode as the most advanced circulation mode for VIPAs.

As proposed in [1], we embed the VIPAFLEET management problem in the framework of a metric task system. We encode the studied site as a *metric space*  $M = (V, d)$  induced by a connected network  $G = (V, E)$ , whose nodes correspond to stations, edges to their physical links in the site, and distances  $d$  between two nodes  $v_i, v_j \in V$  to 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.

An operator manages a fleet of  $k$  VIPAs each with a capacity for  $\text{Cap}$  passengers. The fleet management shall allow the operator to decide when and how to move the VIPAs in the network, and to assign requests to VIPAs. Any request  $r_j$  is defined as a 6-tuple  $r_j = (t_j, x_j, y_j, p_j, q_j, z_j)$  where

- $t_j$  is the release date (i.e., the time when  $r_j$  becomes known),
- $x_j \in V$  is the origin node,  $y_j \in V$  the destination node,
- $p_j$  is the earliest start time,  $q_j$  the latest possible arrival time,
- $z_j$  specifies the number of passengers,

where  $t_j \leq p_j$ ,  $p_j + d(x_j, y_j) \leq q_j$ , as well as  $z_j \leq \text{Cap}$  needs to be satisfied.

The operator decides which requests can be accepted (note that some requests may have to be rejected if, e.g., more requests are specified for a same time window than VIPAs are available in the fleet), and creates tasks  $\tau_j = (t_j, x_j, t_j^{pick}, y_j, t_j^{drop}, z_j)$  to serve accepted requests  $r_j$ , indicating

that  $z_j$  passengers have to be picked up at station  $x_j$  at time  $t_j^{pick}$  and delivered at station  $y_j$  at time  $t_j^{drop}$ , where  $p_j \leq t_j^{pick} \leq q_j - d(x_j, y_j)$  and  $p_j + d(x_j, y_j) \leq t_j^{drop} \leq q_j$  must hold. In order to fulfill tasks, the operator creates tours  $\Gamma^1, \dots, \Gamma^k$  for the VIPAs to circulate in the network  $G$  and to pickup, transport and deliver users s.t.

- each accepted request  $r_j$  is served (within the time window  $[p_j, q_j]$ ),
- each tour starts and ends in the depot and respects the VIPA capacity  $Cap$ .

This leads to the following problem:

**Problem 1 (Online Taxi Mode Problem  $(M, \sigma, p, T, k, Cap)$  (OTMP))**  
*Given a metric space  $M = (V, d)$  induced by a connected network  $G = (V, E)$ , a sequence of requests  $\sigma$ , profits  $p$  for accepted requests, a time horizon  $[0, T]$  and  $k$  VIPAs of capacity  $Cap$ , determine a maximum subset  $\sigma_A$  of accepted requests and find tours  $\Gamma^1, \dots, \Gamma^k$  of minimum total tour length to serve all requests in  $\sigma_A$ .*

In order to solve the OTMP, we propose a replan strategy that considers at each moment in time  $t'$  the subsequence  $\sigma(t')$  of currently waiting requests (i.e., already released but not yet served requests), determines which requests from  $\sigma(t')$  can be accepted, and computes optimal (partial) tours to serve them, performs these tours until new requests are released and recomputes  $\sigma(t')$  and the tours (keeping already accepted requests). Hereby, finding optimal (partial) tours corresponds to solving, in each replanning step, an optimal offline solution on the subsequence  $\sigma(t')$  (i.e. an optimal solution under the condition that the whole sequence of requests is known in advance).

<b>Algorithm 1 (REPLAN)</b>
<b>Input:</b> $(M, \sigma, p, T, k, Cap)$
<b>Output:</b> $\sigma_A$ and tours $\Gamma^1, \dots, \Gamma^k$
1: initialize $\sigma_A = \emptyset$ , $\sigma(t') = \{r_j \in \sigma : t_j = 0\}$ , and $\Gamma^i = (v_0, 0)$ for $1 \leq i \leq k$
2: WHILE $\sigma \neq \emptyset$ DO: call OFFLINE( $\sigma_A, \sigma(t'), \Gamma^1, \dots, \Gamma^k$ ) perform the (modified) tours until new requests become known, update $\sigma(t')$
3: return $\sigma_A$ and $\Gamma^1, \dots, \Gamma^k$

To compute those optimal solutions for the subsequences  $\sigma(t')$ , we build a time-expanded request network  $G(t') = (V', A')$  based on  $\sigma(t')$  and the original network  $G$  and consider a flow in  $G(t')$  that corresponds to the studied (partial) tours.

**Algorithm 2 (OFFLINE)****Input:**  $\sigma_A, \sigma(t'), \Gamma^1, \dots, \Gamma^k$ **Output:** modified  $\sigma_A$  and tours  $\Gamma^1, \dots, \Gamma^k$ 

- 1: determine VIPA start positions  $P(t')$  and start times  $S(t')$  from  $\Gamma^1, \dots, \Gamma^k$
- 2: create the request network  $G(t')$
- 3: solve the max profit flow problem (1) on  $G(t')$
- 4: update  $\sigma_A$  and  $\Gamma^1, \dots, \Gamma^k$  accordingly and return them

To construct  $G(t') = (V', A')$ , we extract the possible start positions  $P(t')$  and start times  $S(t')$  for the VIPAs from the current tours  $\Gamma^1, \dots, \Gamma^k$ : if VIPA  $i$  is currently serving a request  $r_j$ , then we have  $P(t')_i = y_j$  and  $S(t')_i = t_j^{drop}$ ; otherwise,  $P(t')_i$  is the current position  $v$  of VIPA  $i$  and  $S(t')_i = t'$ .

The node set  $V' = V_+ \cup V_x \cup V_y \cup (v_0, T')$  is composed of

- the VIPAs start positions and times  $(P(t')_i, S(t')_i)$  for  $1 \leq i \leq k$  as sources in  $V_+$ ,
- all possible origins  $(x_j, t_j^{pick})$  of all  $r_j \in \sigma(t')$  and all  $p_j \leq t_j^{pick} \leq q_j - d(x_j, y_j)$  in  $V_x$ ,
- all possible destinations  $(y_j, t_j^{drop})$  of all  $r_j \in \sigma(t')$  and all  $p_j + d(x_j, y_j) \leq t_j^{drop} \leq q_j$  in  $V_y$ ,
- a sink node  $(v_0, T')$  with  $T' = \max\{t_j^{drop}, r_j \in \sigma(t')\}$ .

The arc set  $A' = A_+ \cup A_R \cup A_L \cup A_-$  is composed of

- source arcs from all nodes  $(P(t')_i, S(t')_i) \in V_+$  to all reachable origins  $(x_j, t_j^{pick}) \in V_x$  with  $t' + d(v, x_j) \leq t_j^{pick}$ ,
- request arcs from each  $(x_j, t_j^{pick}) \in V_x$  to  $(y_j, t_j^{pick} + d(x_j, y_j)) \in V_y$  in  $A_R$ ,
- link arcs from all destinations  $(y_j, t_j^{drop}) \in V_y$  to all reachable origins  $(y_i, t_i^{pick}) \in V_x$  with  $t_j^{drop} + d(y_j, x_i) \leq t_i^{pick}$  in  $A_L$ ,
- sink arcs from all destinations  $(y_j, t_j^{drop}) \in V_y$  to  $(v_0, T')$  in  $A_-$ .

To keep previously accepted requests, we partition  $\sigma(t')$  into the subsequences

- $\sigma_A(t')$  of previously accepted but not yet served requests and
- $\sigma_N(t') = \{r_j \in \sigma : t_j = t'\}$  of newly released requests,

and partition the request arcs accordingly in  $A_{R_A}$  and  $A_{R_N}$ . Moreover, let  $A_{R_A}^j$ ,  $A_{R_N}^j$  be the subsets of request arcs of the corresponding previously accepted request  $r_j \in \sigma_A(t')$  resp. newly released request  $r_j \in \sigma_N(t')$ .

In  $G(t')$ , we solve the following max profit flow problem

$$\begin{aligned}
\max \quad & \sum_{a \in A_R} p(a)f(a) - \sum_{a \in A'} d(a)f(a) & (1a) \\
\text{s.t.} \quad & \sum_{a \in \delta^+(v,t)} f(a) = k(v) & \forall (v,t) \in V_+ & (1b) \\
& \sum_{a \in \delta^-(v,t)} f(a) = \sum_{a \in \delta^+(v,t)} f(a) & \forall (v,t) \in V_x \cup V_y & (1c) \\
& \sum_{a \in A_{RA}^j} f(a) = 1 & \forall A_{RA}^j \subseteq A_{RA} & (1d) \\
& \sum_{a \in A_{RN}^j} f(a) \leq 1 & \forall A_{RN}^j \subseteq A_{RN} & (1e) \\
& f(a) \geq 0 & \forall a \in A' & (1f) \\
& f(a) \in \mathbf{Z} & \forall a \in A' & (1g)
\end{aligned}$$

where  $\delta^-(v,t)$  denotes the set of outgoing arcs of  $(v,t)$ ,  $\delta^+(v,t)$  the set of incoming arcs of  $(v,t)$  and  $k(v)$  the number of VIPAs initially situated in  $v$ .

Constraints (1d) ensure that previously accepted requests are served whereas constraints (1e) allow to reject newly released requests. Source, flow conservation and nonnegativity constraints (1b), (1c), (1f) together give rise to a totally unimodular matrix, but due to (1d) and (1e) the entire constraint matrix is not totally unimodular s.t. integrality constraints (1g) are required.

From the computed flow  $f$  in the request network  $G(t')$ , it is straightforward to determine newly accepted requests (corresponding to request arcs  $a \in A_{RN}$  with  $f(a) > 0$ ) and to construct tours  $\Gamma^1, \dots, \Gamma^k$  for the VIPAs by standard flow decomposition techniques, see e.g. [4].

In the special case of tight time windows satisfying  $p_j + d(x_j, y_j) = q_j$  (which clearly results in  $p_j = t_j^{pick}$  and  $q_j = t_j^{drop}$ ) for all  $r_j \in \sigma$ , there is exactly one request arc per request s.t. the entire constraint matrix becomes totally unimodular which implies:

**Corollary 1** *The Offline Taxi Mode Problem with tight time windows can be solved in polynomial time.*

In the general case, this is not true, but experiments show that the running times to solve the Offline Taxi Mode Problem are still reasonable, see Table ???. Regarding the quality of the solutions obtained by REPLAN, we note that

- in theory, REPLAN is *not* competitive since there is no finite  $c$  s.t. for all instances  $\sigma$  we have that  $\text{REPLAN}(\sigma) \leq c \text{OPT}(\sigma)$ ,
- in practice, REPLAN provides solutions of good quality within reasonably short time for each recomputation step, see again Table ??.

Hence, we can conclude that REPLAN is a promising algorithm to handle the OTMP for the taxi mode in the studied VIPAfleet management system.

## References

- [1] S. Bsaybes, A. Quilliot, A.K. Wagler: ...
- [2] E. Royer et al.: Outdoor autonomous navigation using monocular vision, In proc. of Intelligent Robots and Systems (IROS 2005), pp. 1253–1258
- [3] E. Royer et al.: Retour d'expérience après plus de 1000 Km en navette sans conducteur guidée par vision, In proc. of RFIA 2016, to appear
- [4] A. Schrijver, Theory of linear and integer progr., John Wiley & Sons, 1998
- [5] <http://www.viameca.fr>