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

Sahar Bsaybes, Allain Quilliot, **Annegret K. Wagler**

Université Clermont Auvergne
LIMOS UMR 6158 CNRS
Clermont-Ferrand, France

LAGOS 2017, Marseille, France
September 12, 2017

L'Europe s'engage en Auvergne avec le Fonds européen de développement régional (FEDER)

Projet co-financé par l'Union européenne

Outline

Outline

VIPAfleet Project: The Partners

A **VIPA** is an **Individual Public Autonomous Vehicle** that

- allows passenger transport in closed sites, but
- does neither require a driver nor an infrastructure to operate.

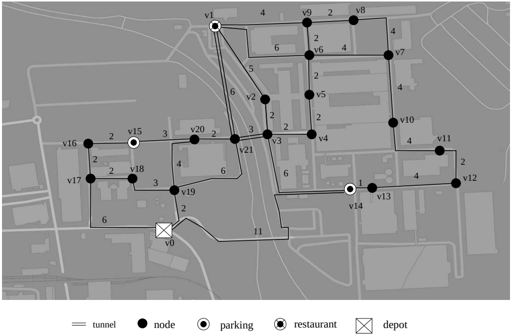
The following partners are involved in the project:

- construction of VIPAs (Easymile and Ligier)
- computer vision guidance technologies (Institut Pascal, UCA)
- fleet management (LIMOS, UCA)

VIPAfleet Project: The Aim

The project VIPAfleet aims at

- contributing to innovative urban mobility solutions by
- transporting passenger in closed sites (e.g. industrial areas, airports).



Example: industrial site of Michelin at Clermont-Ferrand (experimentation from October 2015 until February 2016)

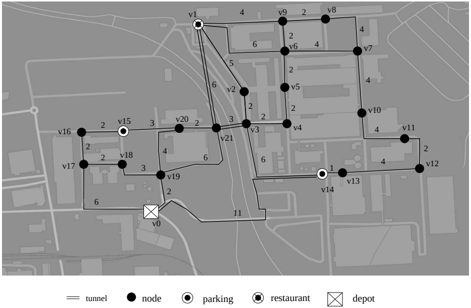
Outline

Modeling the System

The closed site can be encoded by a **metric space** $M = (V, d)$ induced by a weighted graph $G = (V, E, w)$ where

- nodes in V stand for the stations (including an origin v_0, the depot),
- edges in E represent the road connections between them,
- edge weights $w : E \to \mathbf{R}^+$ reflect the road lengths,

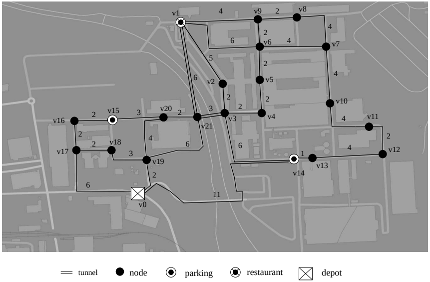
and $d : V \times V \to \mathbf{R}_0^+$ is a metric (induced by shortest path distances).



An Online Pickup-and-Delivery Problem

The task is to transport employees and visitors e.g.

- between parkings and buildings (morning, evening)
- to or from a restaurant (lunch time)
- between buildings (other periods of the day).

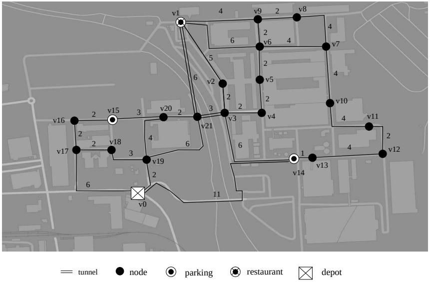


Hereby, the transport requests are released over time (**online situation**), leading to an Online Pickup-and-Delivery Problem.

An Online Pickup-and-Delivery Problem

The task is to transport employees and visitors e.g.

- between parkings and buildings (morning, evening)
- to or from a restaurant (lunch time)
- between buildings (other periods of the day).

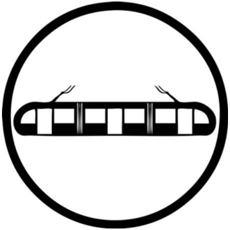


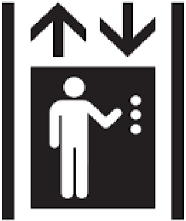
Hereby, the transport requests are released over time (**online situation**), leading to an Online Pickup-and-Delivery Problem.

Circulation modes of VIPAs

To serve transport requests, a VIPA can operate in three modes:

- **Tram mode:** VIPAs contineously run on predefined circuits in a predefined direction and stop at a station if requested to let users enter/leave.
- **Elevator mode:** VIPAs run on predefined lines, but react on customer requests (to enter or leave a car at a station).
- **Taxi mode:** users book their transport requests (from a start to a destination station within a time window) in real time





Circulation modes of VIPAs

To serve transport requests, a VIPA can operate in three modes:

- **Tram mode:** VIPAs contineously run on predefined circuits in a predefined direction and stop at a station if requested to let users enter/leave.
- **Elevator mode:** VIPAs run on predefined lines, but react on customer requests (to enter or leave a car at a station).
- **Taxi mode:** users book their transport requests (from a start to a destination station within a time window) in real time

Proposition

- define inside G circuits/lines and use VIPAs in tram/elevator mode during morning and evening/lunch time
- use the whole network G and VIPAs in taxi mode during the other periods of the day

The latter leads to the Online Taxi Mode Problem!

Outline

Online Taxi Mode Problem: Input

The input for the Online Taxi Mode Problem consists of

- an edge-weighted graph $G = (V, E, w)$ with depot $v_0 \in V$,
- the total number k and the capacity Cap of the VIPAs,
- a time horizon $[0, T]$,
- a sequence $\sigma = \{r_1, \ldots, r_h\}$ of requests $r_j = (t_j, x_j, y_j, p_j, q_j, z_j)$ with
  - 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 possible pickup time,
  - q_j the latest possible delivery time,
  - z_j specifies the number of passengers,

  satisfying $0 \leq t_j \leq p_j$, $p_j + d(x_j, y_j) \leq q_j \leq T$, as well as $z_j \leq \text{Cap}$,
- per request a profit $p(r_j)$ for serving the request r_j.

Online Taxi Mode Problem: Output and Objective

In order to serve requests, the operator creates tours $\Gamma^1, \ldots, \Gamma^k$ for the VIPAs to circulate in the network G and to pickup, transport and deliver users s.t.

- each tour starts and ends in the depot,
- each VIPA carries at most one request at the same time (thus the capacity Cap is always respected),
- each accepted request r_j is served within $[p_j, q_j]$ s.t. all z_j passengers are transported by the same VIPA on a shortest path from x_j to y_j.

Output:

- subset $\sigma_A \subseteq \sigma$ of accepted requests and
- tours $\Gamma^1, \ldots, \Gamma^k$ to serve all requests in σ_A

Objective:

- maximize $|\sigma_A|$ and
- minimize the total tour length of $\Gamma^1, \ldots, \Gamma^k$

Online Taxi Mode Problem: REPLAN strategy

REPLAN

Consider at each moment in time $t' \in [0, T]$ the subsequence $\sigma(t')$ of currently waiting requests and

(1) determine which requests from $\sigma(t')$ can be accepted,

(2) compute optimal (partial) tours to serve them,

(3) perform these tours until new requests are released,

(4) recompute $\sigma(t')$ and the tours (keeping already accepted requests).

Remark:

- Steps (1) and (2) correspond to finding an optimal offline solution for $\sigma(t')$. For that, we use flows in time-expanded request networks $G(t')$.
- To keep previously accepted requests in steps (4), partition $\sigma(t')$ into
  - $\sigma_A(t')$ of previously accepted but until time t' not yet served requests and
  - $\sigma_N(t') = \{r_j \in \sigma : t_j = t'\}$ of requests that are newly released at time t'.

The time-expanded request network $G(t') = (V', A')$

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

- the current VIPA positions as sources in V_+,
- all possible origins (x_j, t_j^{pick}) of all $r_j \in \sigma(t')$ in V_x,
- all possible destinations (y_j, t_j^{drop}) of all $r_j \in \sigma(t')$ in V_y,
- a sink node (v_0, T') with $T' = \max\{t_j^{drop}, r_j \in \sigma(t')\}$.

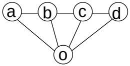
The **arc set** $A' = A_+ \cup A_R \cup A_L \cup A_-$ is composed of

- source arcs from all nodes in V_+ to all reachable origins in V_x in A_+,
- request arcs from each (x_j, t_j^{pick}) to $(y_j, t_j^{pick} + d(x_j, y_j))$ in A_R,
- link arcs from all destinations in V_y to all reachable origins in V_x in A_L,
- sink arcs from all destinations in V_y to (v_0, T') in A_-.

Moreover, define for each request $r_j \in \sigma(t')$ its subset A_R^j of request arcs in A_R.

Applying the REPLAN strategy

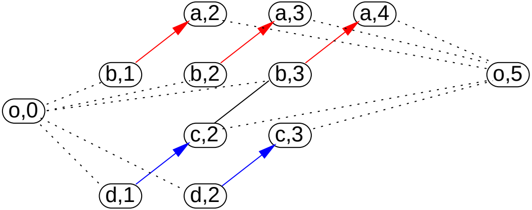
Consider the following graph:



edge weights $x_e = 1$, depot o, 2 VIPAs

$$r_1 = (0, d, c, 1, 3, .) \qquad r_3 = (1, c, d, 2, 3, .)$$
$$r_2 = (0, b, a, 1, 4, .) \qquad r_4 = (1, c, b, 2, 4, .)$$

REPLAN treats this instance as follows:
at time $t = 0$, we have $\sigma_N(0) = \{r_1, r_2\}$ and $G(0)$:



A max profit flow problem on $G(t')$

$$\max \sum_{a \in A_R} p(a)f(a) - \sum_{a \in A' \setminus A_-} d(a)f(a) \tag{1a}$$

$$\text{s.t.} \sum_{a \in \delta^+(v,t)} f(a) = k(v) \qquad \forall (v,t) \in V_+ \tag{1b}$$

$$\sum_{a \in \delta^-(v,t)} f(a) = \sum_{a \in \delta^+(v,t)} f(a) \qquad \forall (v,t) \in V_x \cup V_y \tag{1c}$$

$$\sum_{a \in A_R^j} f(a) = 1 \qquad \forall r^j \in \sigma_A(t') \tag{1d}$$

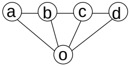
$$\sum_{a \in A_R^j} f(a) \leq 1 \qquad \forall r^j \in \sigma_N(t') \tag{1e}$$

$$f(a) \in \mathbf{Z}_+ \qquad \forall a \in A' \tag{1f}$$

N.B. Constraints (1d) ensure that previously accepted requests are served whereas constraints (1e) allow to reject newly released requests.

Applying the REPLAN strategy

Consider the following graph:

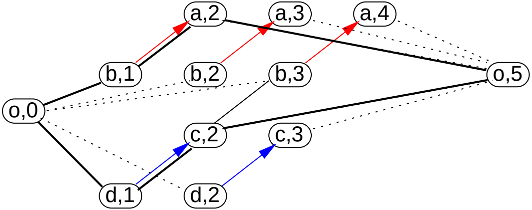


edge weights $x_e = 1$, depot o, 2 VIPAs

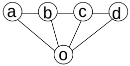
$$r_1 = (0, d, c, 1, 3, .) \qquad r_3 = (1, c, d, 2, 3, .)$$
$$r_2 = (0, b, a, 1, 4, .) \qquad r_4 = (1, c, b, 2, 4, .)$$

REPLAN treats this instance as follows:

r_1, r_2 are accepted and two (partial) tours are planned:



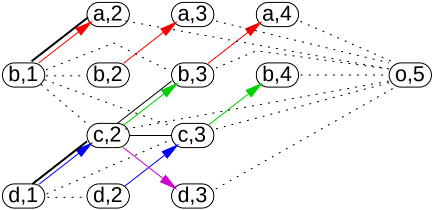
Consider the following graph:



edge weights $x_e = 1$, depot o, 2 VIPAs

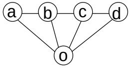
$$r_1 = (0, d, c, 1, 3, .) \qquad r_3 = (1, c, d, 2, 3, .)$$
$$r_2 = (0, b, a, 1, 4, .) \qquad r_4 = (1, c, b, 2, 4, .)$$

REPLAN treats this instance as follows:
at time $t = 1$, we have $\sigma_A(1) = \{r_1, r_2\}$ and $\sigma_N(1) = \{r_3, r_4\}$ and $G(1)$:



Applying the REPLAN strategy

Consider the following graph: edge weights $x_e = 1$, depot o, 2 VIPAs

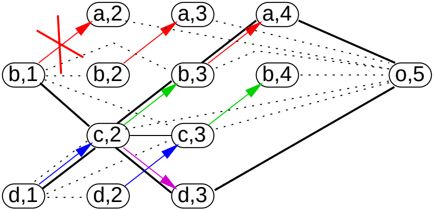


$r_1 = (0, d, c, 1, 3, .)$ $r_3 = (1, c, d, 2, 3, .)$

$r_2 = (0, b, a, 1, 4, .)$ $r_4 = (1, c, b, 2, 4, .)$

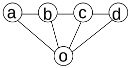
REPLAN treats this instance as follows:

r_3, r_4 are accepted and two (partial) tours are planned:



The optimal offline solution

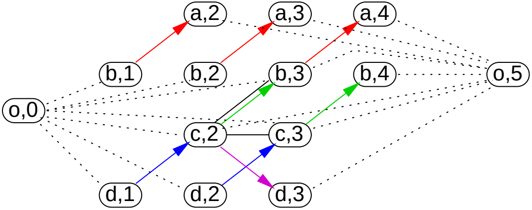
Consider the following graph:



edge weights $x_e = 1$, depot o, 2 VIPAs

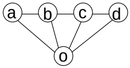
$$r_1 = (0, d, c, 1, 3, .) \qquad r_3 = (1, c, d, 2, 3, .)$$
$$r_2 = (0, b, a, 1, 4, .) \qquad r_4 = (1, c, b, 2, 4, .)$$

In the offline situation, the whole request sequence is known at the beginning; the resulting network is as follows:



The optimal offline solution

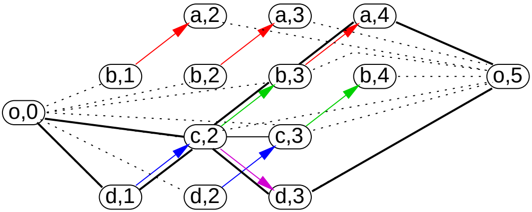
Consider the following graph:



edge weights $x_e = 1$, depot o, 2 VIPAs

$$r_1 = (0, d, c, 1, 3, .) \qquad r_3 = (1, c, d, 2, 3, .)$$
$$r_2 = (0, b, a, 1, 4, .) \qquad r_4 = (1, c, b, 2, 4, .)$$

In the offline situation, the whole request sequence is known at the beginning; the resulting optimal flow is as follows:



Outline

Conclusions

The Online and Offline Taxi Mode Problem are NP-hard in general. However:

Corollary

The Offline Taxi Mode Problem with tight time windows (with $p_j + d(x_j, y_j) = q_j$ and thus $|A^j_R| = 1$ for each request $r_j \in \sigma$) can be solved in polynomial time.

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 σ we have that c REPLAN(σ) $\geq$ OPT(σ),
- in practice, REPLAN provides solutions of reasonable quality within short time for each recomputation step.

Future work: drop the condition that only one request can be served by one VIPA at the same time!

Conclusions

The Online and Offline Taxi Mode Problem are NP-hard in general. However:

Corollary

The Offline Taxi Mode Problem with tight time windows (with $p_j + d(x_j, y_j) = q_j$ and thus $|A_R^j| = 1$ for each request $r_j \in \sigma$) can be solved in polynomial time.

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 σ we have that c REPLAN$(\sigma) \geq$ OPT(σ),
- in practice, REPLAN provides solutions of reasonable quality within short time for each recomputation step.

Future work: drop the condition that only one request can be served by one VIPA at the same time!

Conclusions

The Online and Offline Taxi Mode Problem are NP-hard in general. However:

Corollary

The Offline Taxi Mode Problem with tight time windows (with $p_j + d(x_j, y_j) = q_j$ and thus $|A^j_R| = 1$ for each request $r_j \in \sigma$) can be solved in polynomial time.

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 σ we have that c REPLAN$(\sigma) \geq$ OPT(σ),
- in practice, REPLAN provides solutions of reasonable quality within short time for each recomputation step.

Future work: drop the condition that only one request can be served by one VIPA at the same time!

Thank you very much! Questions?