Experimental Comparison of Autonomous Vehicles Scheduling Algorithms

With Battery Management in a Fixed Line Service

Prisca Aeby

July 28, 2017

Master Thesis, Computer Science

Table of contents

The Problem: Autonomous Vehicles in a Fixed Line Transportation Service

Goals

The Problem

Scheduling autonomous vehicles' activities in a circular line transportation service with dynamic bookings

Goals

- Run simulations as close as possible to real world conditions
- Get insights about strategical choices to improve the:
  - passengers level of service
  - vehicles operating costs

How

- Propose a dynamic fleet size scheduling strategy making use of demand forecasting
- Test how the system's performance metrics react to different scenarios and scheduling strategies

Goals

The Problem

Scheduling autonomous vehicles' activities in a circular line transportation service with dynamic bookings

Goals

- Run simulations as close as possible to real world conditions
- Get insights about strategical choices to improve the:
    - passengers level of service
    - vehicles operating costs

How

- Propose a dynamic fleet size scheduling strategy making use of demand forecasting
- Test how the system's performance metrics react to different scenarios and scheduling strategies

The Problem

Scheduling autonomous vehicles' activities in a circular line transportation service with dynamic bookings

Goals

- Run simulations as close as possible to real world conditions
- Get insights about strategical choices to improve the:
  - passengers level of service
  - vehicles operating costs

How

- Propose a dynamic fleet size scheduling strategy making use of demand forecasting
- Test how the system's performance metrics react to different scenarios and scheduling strategies

Model Outline

Loop

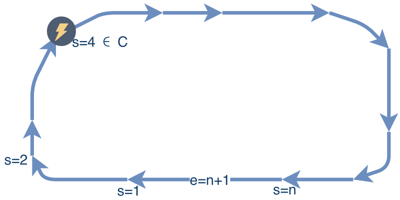


Figure 1: Example of a circular route

- Set of stations $S = 1, ..., n$
- Set of charging stations $C \in S$
- Stop s is mandatory or optional
- Speed of edge e at time t: $speed_t^e$

Vehicles

- Vehicles fleet: V
- Vehicle $v \in V$:
  - Capacity: c^v
  - Maximum battery: q^v Recharging rate: α^v Consumption rate: β^v
  - Maximum speed: s^v

Dynamic Information at time t

- Battery level: l_t^v
- Number of passengers in vehicle: o_t^v
- Distance to the next vehicle on the loop: $next_t^v$
- Stopping duration: $wait_t^v$
- Indicator if vehicle is charging: x_t^v

Vehicles

- Vehicles fleet: V
- Vehicle $v \in V$:
  - Capacity: c^v
  - Maximum battery: q^v Recharging rate: α^v Consumption rate: β^v
  - Maximum speed: s^v

Dynamic Information at time t

- Battery level: l_t^v
- Number of passengers in vehicle: o_t^v
- Distance to the next vehicle on the loop: $next_t^v$
- Stopping duration: $wait_t^v$
- Indicator if vehicle is charging: x_t^v

Bookings

- Booking from stop i to j: $b_{i,j}$, made at time t^b for n^b passengers
- If it has been satisfied $D_b = 1$:
    - Pickup time: p^d
    - Drop-off time: d^d
    - Satisfied by vehicle v^d
    - Waiting time: $w^b = p^b - t^b$
    - Journey time: $\rho^b = d^b - p^b$
- Set of bookings which have been satisfied: $B^* \in B$

Demand

dem_{t^*}: the number of bookings which have been made between t^* and $t^* + 1\text{h}$, so the number of bookings with $t^* \leq t^{*b} < (t^* + 1\text{h})$

t^* is a discrete hourly timestamp

Bookings

- Booking from stop i to j: $b_{i,j}$, made at time t^b for n^b passengers
- If it has been satisfied $D_b = 1$:
  - Pickup time: p^d
  - Drop-off time: d^d
  - Satisfied by vehicle v^d
  - Waiting time: $w^b = p^b - t^b$
  - Journey time: $\rho^b = d^b - p^b$
- Set of bookings which have been satisfied: $B^* \in B$

Demand

dem_{t*}: the number of bookings which have been made between t^* and $t^* + 1\text{h}$, so the number of bookings with $t^* \leq t^{*b} < (t^* + 1\text{h})$

t^* is a discrete hourly timestamp

Vehicles Output Metrics

Metrics at the end of the scheduling horizon of one day (midnight to midnight)

Battery

- $battery^v$: sum of the battery changes over the scheduling horizon when the vehicle is not charging
- $batteryCost = \sum_{v \in V} battery^v$

Occupancy

- $avgOccupancy^v$: average occupancy of vehicle v
- $loadFactor^v$: doesn't consider time a vehicle is going to charge. Passenger-battery over seat-battery:

$$\frac{\sum_{b \in B^*} battery^b * n^b}{battery^v * c^v}$$

Customers' Level of Service Output Metrics

Waiting Time

- *avgWaitingTime* $= \frac{\sum_{b \in B*} w^b}{|B*|}$
- *stabilityWaitingTime* $= \max_{b^+ \in B*} w^{b^+} - \min_{b^- \in B*} w^{b^-}$

Journey Time

- *avgJourneyTime* $= \frac{\sum_{b \in B*} \rho^b}{|B*|}$

Completed Bookings Percentage

- *completedBookings* $= \frac{|B*|}{|B|}$

Numerical Experiments

Parameters Tuning for Stable Results

Testing Environment

- Scheduling Horizon $H = 1$ hour
- Demands $dem_0 = 27$
- Fleet size $|V| = 5$
- Heterogeneous fleet $\forall v \in V$:
  - $c^v = 10$
  - battery $q^v = 1$, charging/consumption rates = 1
  - Maximum speed $s^v = 20$km/h
- Delphi Graph: $S = 1, ..., 32$
- Each simulation run 3 times, average reported
- `Reporting` application fetches vehicles' logs every second

Simulation Settings

Vehicles

- Simulated concurrently following an actor model, each vehicle being an actor
- Actor step: position + battery updated every 1000/*rate* [ms]
- Trade-off between:
  - complexity of computation: if *rate* is too high => actor system has too many unhandled messages
  - precision: if *rate* is too low => vehicles skip stops
- *rate* = 3

Simulation Settings

Time Synchronization

- Synchronize simulated time which is accelerated r times in the different applications, using the system clock
- $simTime = startTimestamp + (currentSystemTime - startTimestamp) * r + shiftDuration$

Figure 2: Performance metrics at growing simulation ratio

Table 1: Difference to $r = 1$

	$r = 24$	$r = 144$
batteryCost	0.5%	21%
completedBookings	4%	52%

=> r = 24 is a reasonable choice!

Time Consistency Between Applications

Difference between $simTime^b$ at which `World Simulator` sends booking b and the reported time in the logs from `Reporting` is not more than 55 seconds and on average 5 seconds

Time Synchronization

- Synchronize simulated time which is accelerated r times in the different applications, using the system clock
- $simTime = startTimestamp + (currentSystemTime - startTimestamp) * r + shiftDuration$

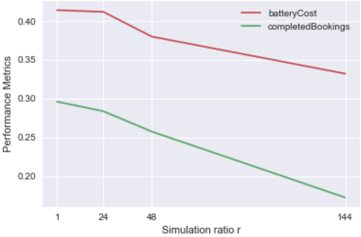


Figure 2: Performance metrics at growing simulation r

Table 1: Difference to $r = 1$

	$r = 24$	$r = 144$
batteryCost	0.5%	21%
completedBookings	4%	52%

=> r = 24 is a reasonable choice!

Time Consistency Between Applications

Difference between $simTime^b$ at which `World Simulator` sends booking b and the reported time in the logs from `Reporting` is not more than 55 seconds and on average 5 seconds

Headway

Goal

Maintain equal distance between vehicles to avoid bus bunching effect

Strategy

- *default*: default waiting time in seconds at a station
- When a vehicle v arrives at a station it waits $wait^{v}_{simTime} = default * ratio$, with $ratio$ computed as follows:
    - Ideal distance: $ideal = len/|V*|$, $V*$ being the set of active vehicles on the line
    - $ratio = next^{v}_{timTime}/ideal$
    - Bounded by *waitMin* and *waitMax*

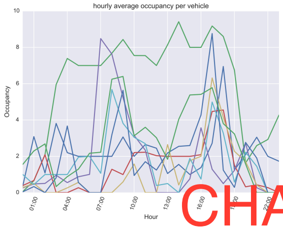
Testing Scenario

- Delphi Site: $S = 1, ..., 32$
- Scheduling horizon: $H = 1$ day (midnight to midnight)
- Week-day bookings from Samira (generated analyzing traffic info from HERE map)
- Heterogeneous fleet size $|V| = 8$
- Speed of edges at any time $speed^{e}_{t} = 20\text{km/h}$
- Charging stations: 10 at the same place on the loop

HEADWAY

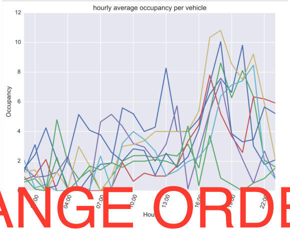
Constant $wait_t^V$

$default$ = 20 [s], $ratio$ = 1



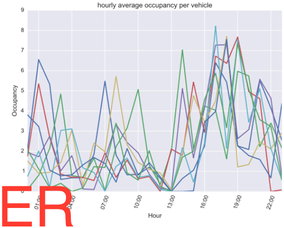
Optional Stops

$default$ = 20 [s], average: 24 [s]



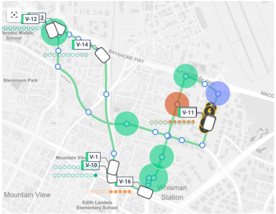
Adapting $wait_t^V$

$default$ = 20 [s], average: 35 [s]

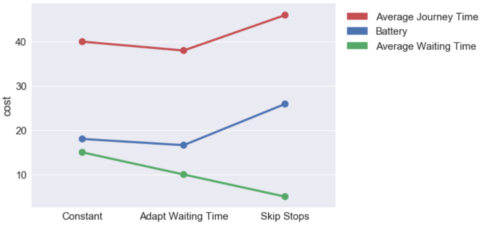


CHANGE ORDER

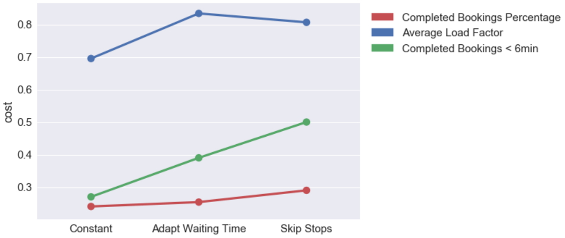
$simTime$ = 6:00 am



Performance Metrics Comparison for Headway



AVG LOAD FACTOR



Dynamic Fleet Size

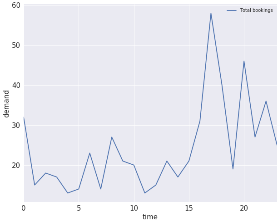
Incentives

- **Demand varies** through the scheduling horizon
- System gathers real-life data and build prediction models through a machine learning pipeline
  => reasonable to assume that demand can be predicted
- Demand high and not enough vehicles => passengers can't board in and increase waiting time
- Demand low and too many vehicles => vehicles running empty (not efficient)
- Need to find the right moment to send vehicles to charge

Strategy

- Decide how many vehicles need to be active at each hour of the scheduling horizon based on the demand.
- Need to decide:
  - How the fleet is rescheduled when the number of vehicles needed on the line changes
    => send the vehicles with the lowest battery level to charge, reactivate the ones with the highest battery level
  - What is the demand
    =>number of bookings made within an hour
  - What is the balance between quality of service VS cost of the running vehicles
    => define cost function

Demand distribution through the day



Demand Frequency

Generating Data

- Simulations with scheduling horizon $H = 1\text{h}$ => no battery management's noise
- $\forall dem \in D = \{13, 15, 21, 23, 25, 27, 31, 36, 40, 46, 58\}$ simulate with fleet size $|V| \in [2, 10]$
- Gather all **performance metrics**
- Time between two consecutive bookings: $1\text{h}/dem$
- **Clean** the data: replace values lower than 5th quantiles and higher than 95th quantiles

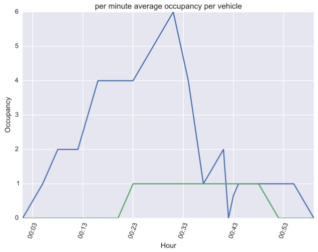
Data: Scheduling horizon H, maximum fleet size V^{max}, optimal fleet size function $optFleet(dem)$
Result: Optimal fleet size at each hour t^* of the scheduling horizon H
$precFleetSize = 0$;
for *each hour* $t^* \in T^*$ **do**
    $optFleet_{t*} = \min(optFleet(dem_{t*}), V^{max})$;
    $\Delta_{t*} = optFleet_{t*} - precFleetSize$;
    if $\Delta_{t*} > 0$ **then**
        create increase fleet size event by Δ_{t*}
    end
    else
        create decrease fleet size event by $-\Delta_{t*}$
    end
    $precFleetSize = optFleet_{t*}$ s;
end

Algorithm 1: Dynamic Fleet Size Algorithm

Extreme Simulation Examples

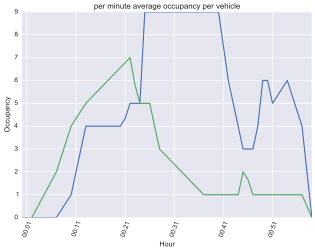
$dem = 13$

$|V|$ = 2, WT = 14, $CBP = 0.61$



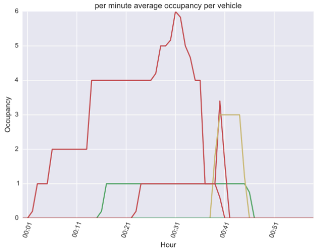
$dem = 58$

$|V|$ = 2, WT = 12, $CBP = 0.31$

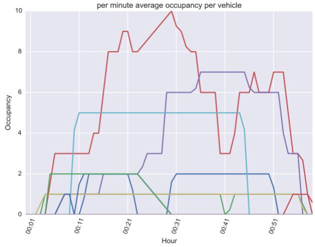


Texte
Texte

$|V|$ = 10, WT = 6, $CBP = 0.69$

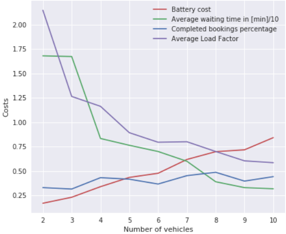


$|V|$ = 10, WT = 5, $CBP = 0.51$



Fleet Size' Impact on Performance Metrics

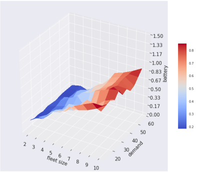
Averages over all demands $\in D$



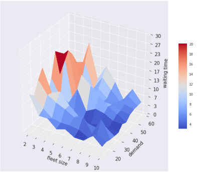
	$\lvert V\rvert$=2	$\lvert V\rvert$=10	Difference (in times)
batteryCost	0.17	0.84	4.95
avgWaitingTime	16.8	3.18	-13.62
avgJourneyTime	27	20	-6.54
completedBookings	0.33	0.44	1.34
avgLoadFactor	0.21	0.06	-3.67

Correlation Coeff	*Dem*	$\lvert V\rvert$
batteryCost	-0.09	0.9
avgWaitingTime	0.24	-0.48
completedBookings	-0.39	0.25
avgLoadFactor	0.5	-0.5

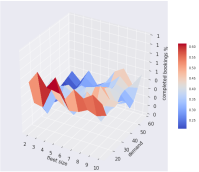
batteryCost



avgWaitingTime



completedBookings



completedbookingsratio

1st strategy: Cost Function

$$cost_{dem,|V|} = \frac{batteryCost + avgWaitingTime}{e^{completedBookings}}$$

batteryCost and *avgWaitingTime* normalized by feature scaling $(x - min)/(max - min)$ for each group of same *dem*

- optimal fleet size for *dem*: $|V|$ which minimizes the cost function: $min_{V\in[2,10]}(cost_{dem,|V|})$

=> For *dem* = 13 we obtain $optFleet = 3$, which grows linearly to $dem = 58$ with $optFleet = 8$ ($R^2 = 0.85$)

Optimal Number of Vehicles: Two Strategies

2^{nd} strategy: Level of Service Constraint

- Find function describing waiting time: $avgWait_{dem,|V|}$
- Optimal fleet size: minimum $|V|$ such that $avgWait_{dem,|V|}$ is under an acceptable threshold (6 minutes)

Function $avgWait_{dem,|V|}$

- Add dummy variables for each group of same *dem* to overcome multicollinearity
- Linear regression: $avgWaitingTime \sim (dem, |V|, dummyVar)$ => $R^2 = 0.53$ (plot of residuals ok)

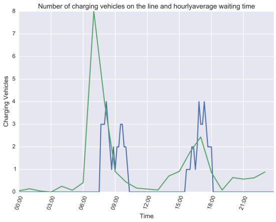
=> *optFleet* between 6 and 8

We compare the simulation results with 8 vehicles always active and adaptive stopping time at stations without skipping stops.

Dynamic Fleet Size Experiments

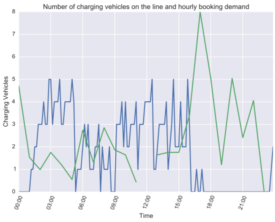
8 Vehicles Active

No dynamic fleet size



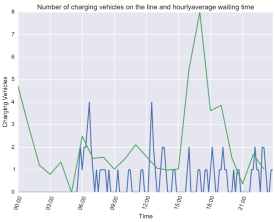
Strategy 1: Cost Function

Fleet size between 3 and 8



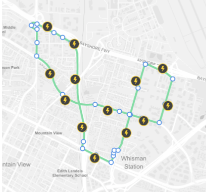
Strategy 2: Max *WT* 6min

Fleet size between 3 and 8



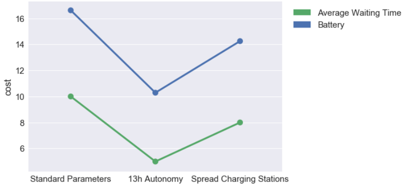
Battery Management Experiments

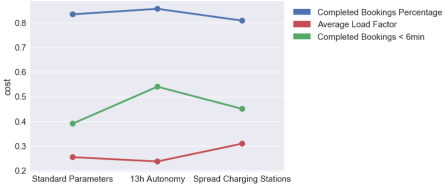
Charging Stations' Position



Better Autonomy

Increase battery autonomy from 8 hours to 13 hours





Improvement of Performance Parameters based on Scheduling Strategies

Difference in percentage to 8 vehicles always active with adaptive stopping times at stations.

Dispatching Strategy	Battery	Waiting Time	Load Factor	Completed Bookings	Completed < 6min
Reference: Constant 8 vehicles	16.61	10	0.25	0.83	0.47
Optional Stops	**43**	**-66**	13.26	-3	28
adaptVehicles	-11	-22	19	**-5**	2
adaptVehicles2	-0.2	-35	**55**	0	4
Spread Charging Stations	-15	-22	19	-3	17
13h Autonomy	**-47**	**-66**	**-7**	**2**	**29**

Thank you!
Questions?