

# The matching problem of empty vehicle redistribution in autonomous taxi systems

**Tatiana Babicheva<sup>a,d,\*</sup>, Wilco Burghout<sup>a,b</sup>,  
Ingmar Andreasson<sup>c</sup>, Nadege Faul<sup>a</sup>**

<sup>a</sup> VEDECOM, 23 bis Allée des Marronniers, 78000 Versailles, France

<sup>b</sup> KTH-Royal Institute of Technology, 100 44 Stockholm, Sweden

<sup>c</sup> LogistikCentrum AB, Osbergsgatan 4 A, 42677 V.Frolunda, Sweden

<sup>d</sup> DAVID, Université de Versailles Saint-Quentin-en-Yvelines, 55 Avenue de Paris, 78000 Versailles, France

---

## Abstract

This article discusses empty vehicle redistribution algorithms for PRT and autonomous taxi services from a passenger service perspective. In modern literature reactive methods such as nearest neighbours are commonly used. In this article we first formulate the general matching problem on a bipartite graph of available vehicles and stations. In addition, we propose an index-based proactive redistribution (IBR) <sup>18,19</sup> algorithm based on predicted near-future demand at stations. The results of different redistribution methods implemented on a simple line test case show that none of the proposed methods are optimal in all cases. Test results of six variations of combined proactive and reactive strategies on a test case in Paris Saclay, France with 20 stations and 100 vehicles are given. The combined Nearest Neighbour / IBR provides a promising solution for both peak and off-peak demand, significantly outperforming all other methods considered, in terms of passenger waiting time (both average and maximum) as well as in terms of station queue lengths.

**Keywords:** *Empty Vehicle Redistribution, Fleet-size, Autonomous Taxi, Matching Problem*

---

## 1. Introduction

Personal Rapid Transit (PRT) is an on-demand transportation system that uses a set of small driver-less (and often electric) vehicles to transport people on-demand from point to point (or station to station). In contrast to traditional (mass) transportation services which serve a fixed route at fixed times and where passenger trips often require transfers, PRT offers a direct non-stop transit service between off-line PRT stations, similar to taxi services. Research within PRT<sup>1</sup>, has traditionally assumed guide-ways (elevated or not), exclusively used by PRT vehicles, e.g. Morgantown PRT<sup>2</sup>, Heathrow<sup>3</sup> etc.

Given the recent rapid development of self-driving (also called autonomous) vehicles, the need for guideways is reduced, as autonomous taxis will be able to use existing roadways (when the traffic is not too congested), and thus provide door-to-door public transport services.

In addition, the current trend towards "Transport as a service" (TaaS) or "Mobility as a Service" (MaaS) where cars (and bicycles, etc.) are shared using increasingly convenient on-line platforms, revolutionises the market for transportation of people, and provides new ways to match planned and unplanned travel demand with transport services. In the remainder of this paper we consider PRT and autonomous taxi services to be equivalent with respect to routing strategies.

An essential prerequisite for any such service is the efficient operation of the vehicle fleet, maximising the service to the passengers while minimising the cost to the operator(s).

One component is the way in which vehicles are assigned to passengers and how they are redistributed in the network to optimally provide this service. In both PRT and autonomous taxi systems, the service is demand-responsive, where most passengers request immediate service. Perfect information about future requests is therefore not available, but statistical information about future requests may be available from historical data. In addition, the passenger demand is often asymmetric across the network and over time, causing imbalances between vehicle supply and passenger demand in the system. Therefore, it is necessary to redistribute the empty vehicles based on demand and supply attributes to trim the system.

The empty vehicle redistribution problem in PRT systems can be formulated as a dynamic version of the Vehicle Routing Problem (VRP), which is a generalization of the Travelling Salesman Problem (TSP). Both problems are NP hard, and especially for the dynamic (time-varying) versions, heuristics are usually proposed for their solution.

The methods of empty vehicle redistribution can be divided into two main methods: reactive redistribution (when the call is made at the moment of passenger arrival) and proactive redistribution using to meet demand (redistribution based on future predicted demand).

---

\* Corresponding author. Tel.: +33-754-437-467.

E-mail address: babicheva.t.s@gmail.com

© 2019 International Association for Sharing Knowledge and Sustainability.

DOI: 10.5383/JTTM.01.01.001

Most reactive methods are based on nearest neighbour heuristics. Such methods of calling the nearest available vehicle to a passenger are used by Fagnant<sup>4</sup>, Kockelman<sup>5</sup>, Bell and Wong<sup>8</sup>, Andreasson<sup>1, 13, 14, 15</sup>, Lees-Miller<sup>3, 9, 10</sup>, Fatnassi et al<sup>12, 11</sup> and Kek<sup>16</sup>. These methods may be modified to send the vehicle to the longest waiting passenger in the system (it can be both current waiting time and predicted waiting time at the moment the vehicle will arrive).

The proactive methods use the statistics of supply and demand in different stations or regions of the city. These are used to predict the necessary number of the vehicles in each station (Fatnassi et al<sup>12, 11</sup>, Anderson<sup>7</sup>). Such methods may sort the stations by some predicted criteria such as available vehicles, waiting times etc. (Kek<sup>16</sup>), by sending vehicles from stations with vehicle surplus to stations with vehicle deficit (Andreasson<sup>1, 13, 14, 15</sup>, Hanna et al<sup>6</sup>), or by indexing of the stations<sup>18, 19</sup>.

In this article we focus on reactive redistribution strategies as well as proactive redistributions.

The remainder of the paper is organised as follows. First we formulate the dynamic vehicle redistribution problem, then we describe an index-based redistribution algorithm (IBR) as well as the reference algorithms we compare it with. After that we formulate the matching problem between empty vehicles and serving stations and show on line network, that there are no optimum in pure strategies exist. In the following section we compare the algorithms (and combinations of algorithms) on a proposed autonomous taxi network in Paris Saclay, France and discuss the results.

The last section discusses limitations of this work, future research directions and conclusions.

## 2. Vehicle redistribution methods

### 2.1. Problem formulation

- Given a set of stations  $\langle S \rangle$ , for each station  $s \in S$  we have some statistics about future arrivals (variable expectancy of number of arrival passengers  $N_s(T_1, T_2)$  in time periods  $(T_1, T_2)$ ).
- For  $s_1, s_2 \in S$  in every start time moment  $t$  we have the travel time  $T[s_1, s_2](t)$  between the stations. We assume the travel times to be constant for each optimization time interval.
- For the set of vehicles  $\langle V \rangle$ , each vehicle  $v \in V$  has two trips, “current” and preliminary “next”, based on waiting passengers, with attributes  $\langle \text{time of arrival, destination station, passenger} \rangle$ .
- We have a set of passengers  $\langle P \rangle$ . For each passenger  $p \in P$  we have the current waiting time  $t_p$ .
- In each station  $s \in S$  there is a set of passengers  $P_s$  and the vehicle association set  $\langle V_s^a \rangle$ , which contains the vehicles moving towards this station to take passengers. The passenger surplus/deficit parameter  $P_s^a$  can be calculated as  $P_s^a = |\langle P_s \rangle| - |\langle V_s^a \rangle|$ . The station does not have any constraints on the number of passengers that can wait in the queue or on the number of vehicles it can accommodate.

In this paper we focus mostly on the minimization of the average and maximum passenger waiting times.

### 2.2. Index-based redistribution algorithm

In order to take into account the non-linear passenger preferences for the waiting time, as well as to simplify the

calculations for potential passenger heuristics (for example, to introduce risk aversion), we introduce a passenger (dis-)utility function  $u(t)$ , depending on the passenger waiting time. In this article we use the function  $u(t) = e^{-t}$ . The passenger dis-utility increases exponentially with the waiting time, meaning that the algorithm will strongly prioritise passengers with long waiting times.

The passenger arrival process is modelled as a stationary Poisson point process. The demand data provides the passenger arrival rates in every station during different time periods. The probability of at least one passenger arrival during the time period  $(t_1; t_2)$  is defined as  $P_{arr} > 0(t_1; t_2)$ .

For every station we introduce the “station index”, which is calculated as a measure of expected maximum passenger dis-utility at the time of pick-up with the nearest vehicle, including the time it takes for the vehicle to arrive at the station. The station index calculation is based on passengers already arrived as well as on “virtual” passengers will arrive in certain time periods with Poisson arrivals. For every station  $s \in S$  we determine the nearest available vehicle  $v_s^{nearest} \in V$  (given its current or next trip). Given the current time moment  $t_1$  we calculate that  $v_s^{nearest}$  will arrive to the station at the time moment  $t_2$ . For every station the “station index” is then calculated as follows<sup>18, 19</sup>:

1. If  $P_s^a > 0$  (vehicle deficit). Let the maximum waiting time among the passengers be  $T = \max_{p \in P_s} t_p$ . The station index is defined as  $u(t_2 - t_1 + T)$ . In other words, the station index is defined as the disutility of the longest waiting passenger at the station, at the time of his predicted departure.

2. If  $P_s^a = 0$  (balanced) The index equals the maximal expected disutility of the first arriving passenger

$$\max_{t_1 < \tau < t_2} [P_{arr} > 0(t_1, \tau) u(t_2 - \tau)].$$

3. If  $P_s^a = -X$  (vehicle surplus of  $X$  vehicles). Considering the waiting time expectancy  $T_X$  of the  $X$  passengers arriving at this station for the Poisson process. If  $t_1 + T_X \geq t_2$  the *index* = 0, else it is calculated as in  $P_s^a = 0$  (balanced) case, changing the start time from  $t_1$  to  $t_1 + T_X$ . In other words, in case of non-positive passenger surplus, the index equals the probable disutility of the first arriving passenger at the time of his departure

To generalize the three cases above, the station index is a measure of expected maximum passenger disutility at the time of pick-up, including the time it takes for the vehicle to arrive at the station.

### 2.3. Algorithms of redistribution

The methods of empty vehicle redistribution used in this article are described in the following list

- Sending/calling algorithms (reactive redistribution methods)
  - Basic allocation (BA). If there are passengers and empty vehicles in the same station, the vehicle will be assigned to the longest waiting passenger. (No redistribution of empty vehicles to other stations)
  - Simple Nearest Neighbours (SNN). Calling the nearest empty vehicles based on longest waiting passenger time in the current moment.
  - Heuristic Nearest Neighbours (HNN). Reallocation of nearest empty vehicles is based on longest waiting passenger time in the vehicle arriving moment. This method attempts to improve upon SNN by including the time it takes for a vehicle to move to the waiting passenger.
  - Send The Nearest (STN). Reallocation of nearest empty vehicles is based on the station with waiting passengers

ranging on the time from the nearest available vehicle to the station.

- Redistribution algorithms (proactive redistribution methods)
  - Index-based Redistribution (IBR). The redistribution based on maximum station index described and implemented by Babicheva et al.<sup>18,19</sup>.
  - Surplus/Deficit vehicle redistribution (SDR). The redistribution from the station with the maximum vehicle surplus to the station with the maximum vehicle deficit.

#### 2.4. The SNN, IBR and STN algorithms evaluation

In this section we first evaluate the SNN, IBR and STN algorithms mathematically on a simple linear network, to see if any of the methods dominates the other in terms of average and maximum waiting time. Let us consider a single line network with two stations and two vehicles (Figure 1). Each station has one passenger waiting for a vehicle. The driving time between the two stations is 5 minutes. We will now evaluate a number of cases with different placement of vehicles and waiting times for the passengers.

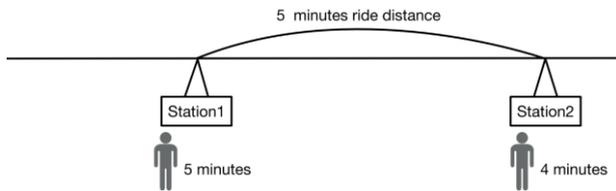


Fig. 1. The sample line network

##### 2.4.1. The first case of vehicle placement

The first case is shown in the Figure 2. The first vehicle is placed at 3 minutes ride time from station 1 and the second vehicle is placed between station 1 and station 2, with a ride time of 1 minute to station 1 and 4 minutes to station 2. The three algorithms are now evaluated and compared to see which algorithm produces the best maximum waiting time, and the best average waiting time for the passengers.

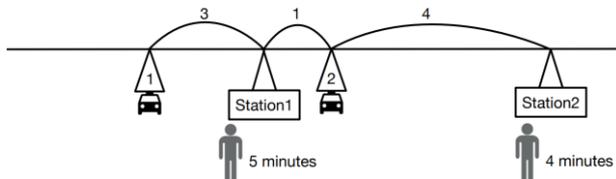


Fig. 2. The sample line network. Case 1

The SNN algorithm will in each optimisation step serve the longest waiting passenger first with the nearest empty vehicle. The longest waiting passenger is in station 1, and the nearest vehicle is vehicle 2, which will thus be assigned to the passenger in station 1. Then there is one waiting passenger in station 2 that has not yet been assigned. To this passenger the remaining unassigned empty vehicle 1 will be assigned.

Therefore, the waiting times for the passengers, at time of departure will be:

- For the passenger in station 1: 6 minutes (5 min + 1 min drive time)

- For the passenger in station 2: 12 minutes (4 min + 8 min drive time)

Thus for SNN the maximum waiting time would be 12 minutes and the average waiting time 9 minutes.

The STN algorithm will for each optimisation step evaluate all the vehicle-station distances (in drive time) and assign the nearest vehicle-station pair. In this case,

- the V1-St1 distance is 3 minutes,
- the V1-St2 distance is 8 minutes,
- the V2-St1 distance is 1 minute,
- the V2-St2 distance is 4 minutes.

Thus, the algorithm will assign vehicle 2 to the station 1.

In the next optimisation step, the algorithm will assign the remaining vehicle 1 to the remaining passenger in station 2.

The dispatching scheme in this case is therefore the same as for SNN. Thus, the waiting times for the passengers, at time of departure will be:

- For the passenger in station 1: 6 minutes (5 min + 1 min drive time)
- For the passenger in station 2: 12 minutes (4 min + 8 min drive time)

Thus for STN as in SNN the maximum waiting time would be 12 minutes and the average waiting time 9 minutes.

The IBR algorithm will sort all the stations by their index as defined in section 2.2, and assign the nearest vehicle to the station with the highest index.

The index for station 1 is the (dis-)utility of the longest waiting passenger in this station including the time it takes for the nearest vehicle to get to the station. It will take 1 minute for the nearest vehicle to get to station 1 (vehicle 2), thus the index will be equal  $u(5 + 1) = u(6)$ . The second station index is  $u(4 + 4) = u(8)$ .

The utility function is monotonously increasing thus  $u(8) > u(6)$ . Based on the algorithm, the station with the maximal index (station 2) will be served with the nearest vehicle (vehicle 2).

In the next optimisation step, the station 1 will be served with vehicle 1.

Thus, the waiting times for the passengers, at time of departure will be:

- For the passenger in the station 1: 8 minutes (5 min + 3 min drive time)
- For the passenger in the station 2: 8 minutes (4 min + 4 min drive time)

Thus for IBR the maximum waiting time would be 8 min and the average waiting time 8 min.

Therefore, in case 1 the IBR method shows the best results for both maximal and average passenger waiting times.

##### 2.4.2. The second case of vehicle placement

The second case is shown in the Figure3. The second vehicle is placed at 7 minutes ride time from station 1 and the first vehicle is placed at 1 minute ride time from station 1.

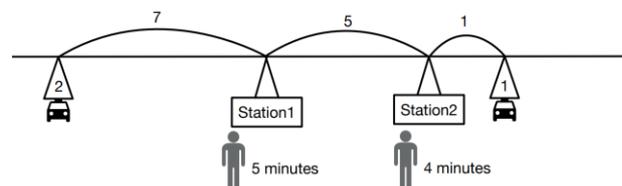


Fig. 3. The sample line network. Case 2

The SNN algorithm will assign the vehicle 1 to the station 1, and the vehicle 2 to the station 2.

Therefore, the waiting times for the passengers, at time of departure will be:

- For the passenger in station 1:  
11 minutes (5 min + 6 min drive time)
- For the passenger in station 2:  
16 minutes (4 min + 12 min drive time)

Thus for SNN the maximum waiting time would be 16 minutes and the average waiting time 13.5 minutes.

The STN algorithm will assign vehicle 1 to the station 2 and the vehicle 2 to the station 1.

Thus, the waiting times for the passengers, at time of departure will be:

- For the passenger in station 1: 12 minutes (5 min + 7 min drive time)
- For the passenger in station 2: 5 minutes (4 min + 1 min drive time)

Thus for STN the maximum waiting time would be 12 minutes and the average waiting time 8.5 minutes.

The IBR algorithm will send the vehicle 1 to the station 1 and the vehicle 2 to the station 2. The dispatching scheme in this case is therefore the same as for SNN. Thus, the waiting times for the passengers, at time of departure will be:

- For the passenger in station 1:  
11 minutes (5 min + 6 min drive time)
- For the passenger in station 2:  
16 minutes (4 min + 12 min drive time)

Thus for SNN and IBR the maximum waiting time would be 16 minutes and the average waiting time 13.5 minutes. Therefore, in case 2 the STN method shows the best results both in maximal and average passenger waiting times.

#### 2.4.3. The third case of vehicle placement

The third case is shown in the Figure4. The first vehicle is placed between station 1 and station 2, with a ride time of 1 minute to station 1 and 4 minutes to station 2. The second vehicle is placed at 5 minutes ride time from station 2.

The SNN algorithm will send vehicle 1 to the station 1 and vehicle 2 to the station 2.

Therefore, the waiting times for the passengers, at time of departure will be:

- For the passenger in station 1:  
6 minutes (5 min + 1 min drive time)
- For the passenger in station 2:  
9 minutes (4 min + 5 min drive time)

Thus for SNN the maximum waiting time would be 9 minutes and the average waiting time 7.5 minutes.

The STN algorithm will send vehicle 1 to the station 1 and vehicle 2 to the station 2.

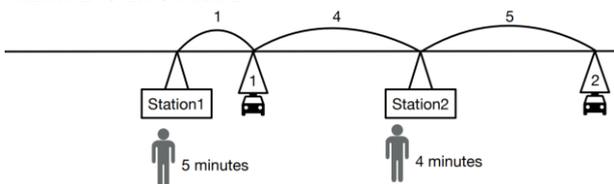


Fig. 4. The sample line network. Case 3

The dispatching scheme in this case is therefore the same as for SNN.

Thus, the waiting times for the passengers, at time of departure will be:

- For the passenger in station 1: 6 minutes (5 min + 1 min drive time)
- For the passenger in station 2:  
9 minutes (4 min + 5 min drive time)

Thus for SNN and STN the maximum waiting time would be 9 minutes and the average waiting time 7.5 minutes.

The IBR algorithm will send vehicle 1 to the station 2 and the vehicle 2 to the station 1.

Thus, the waiting times for the passengers, at time of departure will be:

- For the passenger in station 1:  
15 minutes (5 min + 10 min drive time)
- For the passenger in station 2:  
8 minutes (4 min + 4 min drive time)

Thus for IBR the maximum waiting time would be 15 minutes and the average waiting time 11.5 minutes.

So, in case 3 the STN and SNN methods shows best results both in maximal and average passenger waiting times.

#### 2.4.4. The fourth case of vehicle placement

The fourth case is shown in the Figure5. The first vehicle is placed between station 1 and station 2, with a ride time of 4 minutes to station 1 and 1 minute to station 2. The second vehicle is placed at 2 minutes ride time from station 2.

The SNN algorithm will assign vehicle 1 to the station 1 and vehicle 2 to the station 2. Therefore, the waiting times for the passengers, at time of departure will be:

- For the passenger in station 1:  
6 minutes (5 min + 4 min drive time)
- For the passenger in station 2:  
9 minutes (4 min + 2 min drive time)

Thus for SNN the maximum waiting time would be 9 min and the average waiting time 7.5 min.

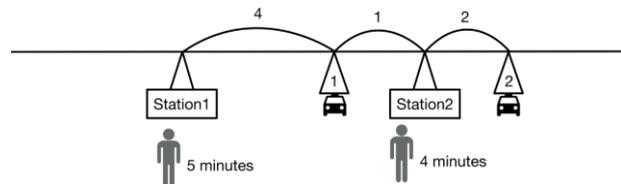


Fig. 5. The sample line network. Case 4

The STN algorithm will assign vehicle 1 to the station 2 and the vehicle 2 to the station 1.

Thus, the waiting times for the passengers, at time of departure will be:

- For the passenger in station 1:  
12 minutes (5 min + 7 min drive time)
- For the passenger in station 2:  
5 minutes (4 min + 1 min drive time)

Thus for STN the maximum waiting time would be 12 minutes and the average waiting time 8.5 minutes.

The IBR algorithm will assign vehicle 1 to the station 1 and vehicle 2 to the station 2.

The dispatching scheme in this case is therefore the same as for SNN. Thus, the waiting times for the passengers, at time of departure will be:

- For the passenger in station 1:  
6 minutes (5 min + 4 min drive time)
- For the passenger in station 2:  
9 minutes (4 min + 2 min drive time)

Thus for SNN and IBR the maximum waiting time would be 9 minutes and the average waiting time 7.5 minutes.

So, in case 4 the IBR and SNN methods shows best results both in maximal and average passenger waiting times.

2.4.5. Summary

As shown in table 1, there is no single best algorithm which suits all evaluated cases. For example, the IBR algorithm was the best in cases 1 and 4, the SNN algorithm — in cases 3 and 4 and STN algorithm — in cases 2 and 3. This demonstrates that even in a very simple evaluation case, different situations need different algorithms. We therefore conclude that a mix of algorithms may provide a more robust strategy, less sensitive to particular demand characteristics and vehicle distributions on the network. Moreover, we note that the greedy type of assigning vehicle-passenger pairs sequentially often leads to suboptimal solutions.

Table 1. Average and maximal waiting times, in min

	Case 1		Case 2		Case 3		Case 4	
	Avg	Max	Avg	Max	Avg	Max	Avg	Max
IBR	8	8	13,5	16	11,5	15	7,5	9
SNN	9	12	13,5	16	7,5	9	7,5	9
STN	9	12	8,5	12	7,5	9	8,5	12

3. The matching problem

After determining for each station a parameter that depends on the chosen redistribution strategy (station index, number of passengers, waiting time of the maximum passenger, or the minimum distance to an available vehicle with the presence of passengers at a given station) and the set of vehicles available for redistribution, the "greedy" methods based on step-by-step matching between the first-sorted elements can be considered, for example, to send the nearest available vehicle to the station with the largest index.

This method can show good convergence, but it is unlikely to always reach the optimal solution. In addition it may cause unnecessary runs of empty vehicles. To improve the solution we represent the data of the 2 sets in the form of a bipartite graph, where the edges are the travel times from the available vehicles to the stations that need to be served. First, the stations are ranked according to their priority (index, passenger waiting times or distance for the nearest available vehicle). Let it be required to find matching vehicles to the *N* top ranked stations in the network.

To achieve a minimum average waiting time for passengers, it is required to find matchings minimizing the total sum of the used edges.

In general, this problem is NP-hard, but for some specific cases polynomial solutions exist. For example, if the number of serving stations is equal to the number of available vehicles, the Hungarian algorithm<sup>23</sup> can be used.

In this article we investigate some heuristic methods of the matching problem solution. In every call of the matching function in the network based on the statistics about the stations we will serve the top *N* stations in area of the first-order station. The number of matchings can never exceed the number of available vehicles.

We use the following example to show that the greedy algorithm may not provide the optimal solution. Let us choose the SNN strategy based on maximal passenger waiting time.

Fig. 6 shows the sample network with 2 stations and 2 available vehicles that can be represented as bipartite graph in Fig. 7.



Fig. 6. The sample network

Greedy solutions can be used in two different approaches: in each step we can choose a passenger with maximal waiting time or we can choose the vehicle with shortest path to any of the waiting passengers. In the first case the longest waiting passenger is at station 1, so the nearest vehicle (2) will serve it. The next longest waiting passenger at station 2 will then be served by vehicle 1. Total waiting time is  $(5 + 2) + (4 + 8) = 19$  minutes and total run of empty vehicles are  $8 + 2 = 10$  minutes. In the second case the second vehicle will be chosen for serving a passenger on the station 1, because it has shortest time to reach nearest passenger. The next step the first vehicle serves a passenger on the station 2. This scenario gives us the same result as in the first case.

The optimal solution would assign vehicle 2 to station 2, and vehicle 1 to the station 1, and the departure waiting times will be 8 minutes and 7 minutes for the stations 1 and 2 respectively. Thus, both average waiting time and maximal waiting time will be smaller.

3.1. The Greedy+Hungarian matching algorithm

In order to avoid the NP-complexity of this problem, it is proposed to use the 2-steps method.

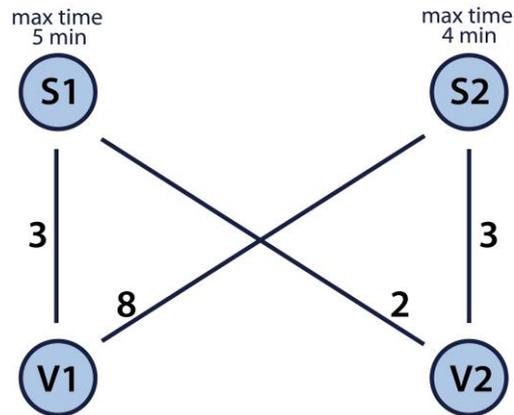


Fig. 7. The bipartite graph

In the first step we call the greedy strategy for pre-assignment of the available vehicles to the station.

In the second step the set obtained with *N* stations and *N* vehicles will be the input of the Hungarian algorithm. The computational complexity of this complex algorithm is estimated as follows. Let the number of requests be *N<sub>R</sub>* and the number of free vehicles be *N<sub>C</sub>*. The first step requires two sorting operations  $T_1 = O(N_R \log N_R + N_C \log N_C)$ . Let  $N_H = \min(N_R; N_C)$ . The second step will require

$T_2 = O(N^3H)$  which is the final computational complexity of algorithm, because  $T_2 \gg T_1$ , i.e. polynomial hard.

#### 4. Evaluation of the greedy mixed algorithms

In this section we compare the algorithms presented in the previous section, using data from a study case in Paris Saclay, France. The area has been selected as the French test bed for testing and evaluating autonomous vehicle services. The proposed autonomous taxi network connects the main campuses to the Massy public transportation hub (SNCF long distance trains, RER commuter trains).

In the remainder of this section we evaluate the mixed vehicle redistribution algorithms on the network, given a fixed fleet size of 100 vehicles. In every decision moment (every time step) we choose between the methods with fixed probabilities,

bigger for reactive and smaller for proactive methods.

The off-peak passenger demand between the 20 stations with the maximal distance around 10 kilometers was obtained from the Ile de France Transport Authority (STIF), using data for current passenger demand for the area. The results presented below were obtained as an average over 100 simulation runs.

In Table 1 the main results for the mixed methods are presented.

The results obtained show that the scenario with mixed SNN + IBR methods produce the lowest values across the table for maximum waiting time 5.1 minutes and average waiting time 0.25 minutes.

The rush hour demand data also shows that the IBR+SNN method shows the best results on maximal and average waiting times and the total queue length in the end of the simulation.

**Table 2. Comparison between different redistribution methods for the off-peak demand.**

Strategy	BA	SNN	SNN,SDR	SNN,IBR	HNN	HNN,SDR	HNN,IBR
Maximal waiting time, min	60	8.4	7.9	5.1	10.7	9.6	6.2
Average waiting time, min	20	0.78	0.77	0.25	1.13	1.10	0.36
Average queue, passengers	398	47	46	15	68	66	22
Total vehicle run, min	609	3416	3327	9305	3317	3342	8975

**Table 3. Comparison between different redistribution methods for the rush hour demand.**

Strategy	BA	SNN	SNN,SDR	SNN,IBR	HNN	HNN,SDR	HNN,IBR
Maximal waiting time, min	50	14	13.5	13	30	25	23
Average waiting time, min	31	3.3	3.2	2.4	7	6.2	4.7
Average queue, passengers	872	31	30	21	95	72	57
Total vehicle run, min	262	7376	7649	9451	7331	7385	9922

#### 5. Evaluation of multi-matching algorithms

Based on the results from the previous section, the mixed SNN+IBR method was proposed for the multi-matching method base. In order to evaluate the algorithms, we choose a fleet-size of 60 vehicles. The program base for the modelling and evaluation of the algorithms was the VIPSIM project by VEDECOM, a tool for simulation of mobility service networks, to define, test and evaluate operation strategies embedding all the relevant economic factors, including the quality of service for users.

In this section we consider three algorithm versions to compare. One-step mixing is a redistribution algorithm, according to which, each time a single decision on redistribution of a single vehicle is about to be made, we choose randomly (according to predefined probabilities) one of the redistribution methods described above (which includes itself its own criteria for ranging stations and vehicles). This algorithm is then used to determine the top-rated station, and match this station to one available vehicle, which is then redistributed to this top-rated station. This algorithm is computationally expensive since it only assigns a single vehicle to a station at each step.

In the greedy mixing and greedy-Hungarian algorithms, in each step we choose one redistribution method and assign all

available vehicles to stations. In greedy mixing we do step by step assignment between the top-rated station and the nearest available vehicle to it, in the greedy-Hungarian algorithm we minimize the total run distance in order to serve all the top-rated stations. These two algorithms are less computationally expensive than the one-step mixing as they assign all available vehicles at each algorithm step.

The results obtained show that the Greedy-Hungarian matching strategy improves average waiting time in relation to just Greedy mixing. However, in terms of waiting times and queue lengths the one-step mixing outperforms both the greedy mixing and the greedy-Hungarian method, albeit at a computational cost of having to re-run the redistribution algorithm for each available vehicle.

It is necessary to mention that although the GRID solution (checking of all the possibilities, not considered here due to its NP-complexity) may give a global optimum solution at any step, total simulation still could deviate from optimum. For example, if we apply the SNN strategy to every vehicle, there will not be any vehicles available to employ proactive the IBR strategy, and thus the strategy would fail to meet higher future demand at any station. Balancing applying redistribution strategies to meet current demand versus reserving some empty vehicles to be redistributed later to meet future demand, is an important issue for further research.

**Table 4. The comparison between different redistribution methods for the off-peak demand.**

Matching strategy	One-step mixing	Greedy mixing	Greedy-Hungarian algorithm
Maximal waiting time, min	6.3	7.2	7.1
Average waiting time, min	0.51	0.94	0.59
Average queue at the end, passengers	0	5	6

## 6. Discussion and conclusions

In this article, we investigate empty vehicle redistribution algorithms for PRT or autonomous taxi services, primarily from a passenger service perspective. Using a simple case, we first evaluate three algorithms for four different cases of initial vehicle placements and demonstrate that none of the algorithms dominates, and that greedy types of algorithms by themselves often reach suboptimal outcomes. We then introduce the matching problem and the multi-matching algorithms. We evaluate six variations of greedy algorithm combinations on a test case in Paris Saclay, France. The results show that especially the combination of Simple Nearest Neighbours + Index Based Redistribution provides very promising results. Then we evaluate multi-matching algorithms on the same network. While the Greedy-Hungarian multi-matching algorithm improves on the well-known Hungarian method to get better results both in empty run time and in average waiting time, the one-step mixing shows the best results. All three multi-matching algorithms outperform the single matching algorithms.

## References

- 49(15), 2016, pp. 254-259. <https://doi.org/10.1016/j.ifacol.2016.07.757>
- [7] Anderson JE. Control of Personal Rapid Transit System. *Elektronikk*, 99(1), 2003, pp. 108-116.
- [8] Bell MG, Wong K. A rolling horizon approach to the optimal dispatching of taxis. *Transportation and Traffic Theory. Flow, Dynamics and Human Interaction. 16th International Symposium on Transportation and Traffic Theory*, 2005. <https://doi.org/10.1016/B978-008044680-6/50034-9>
- [9] Lees-Miller JD, Wilson RE. Proactive empty vehicle redistribution for personal rapid transit and taxis. *Transportation Planning and Technology* 35(1), 2012, pp.17-30. <https://doi.org/10.1080/03081060.2012.635414>
- [10] Lees-Miller J, Hammersley J, Wilson R. Theoretical maximum capacity as benchmark for empty vehicle redistribution in personal rapid transit. *Transportation Research Record: Journal of the Transportation Research Board* (2146), 2010, pp.76-83. <https://doi.org/10.3141/2146-10>
- [11] Fatnassi E, Chebbi O, Chauouchi J. Dealing with the Empty Vehicle Movements in Personal Rapid Transit System with Batteries Constraints in a Dynamic Context. *Journal of Advanced Transportation* 2017, 2017. <https://doi.org/10.1155/2017/8512728>
- [12] Fatnassi E, Chebbi O, Siala JC. Two strategies for real time empty vehicle redistribution for the personal rapid transit system. *Intelligent Transportation Systems-(ITSC), 16th International IEEE Conference on IEEE*, 2013, pp. 1888-1893. <https://doi.org/10.1109/ITSC.2013.6728504>
- [13] Andreasson, I. Quasi-optimum redistribution of empty PRT vehicles. *Sixth International Conference on Automated People Movers (APMs)*, 1997.
- [14] Andreasson, I. Reallocation of empty personal rapid transit vehicles en route. *Transportation Research Record: Journal of the Transportation Research Board* (1838), 2003, pp. 36-41. <https://doi.org/10.3141/1838-05>
- [15] Andreasson, I. Ride-sharing on PRT. *Automated People Movers 2005: Moving to Mainstream*, 2005, pp. 1-7. [https://doi.org/10.1061/40766\(174\)22](https://doi.org/10.1061/40766(174)22)
- [16] Kek A, Cheu R, Chor M. Relocation simulation model for multiple-station shared-use vehicle systems. *Transportation Research Record: Journal of the Transportation Research Board* (1986), 2006, pp. 81-88. <https://doi.org/10.3141/1986-13>
- [17] Kuhn HW. The Hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 2(12), 1955, pp. 83-97. <https://doi.org/10.1002/nav.3800020109>
- [18] Babicheva T, Burghout W, Andreasson I, Faul N. Empty vehicle redistribution and fleet size in autonomous taxi systems. *IET Intelligent Transport Systems*, 2018, pp. 677-682 <https://doi.org/10.1049/iet-its.2018.5260>
- [1] Andreasson, I. Operational strategies from Personal to Mass Transit. *Proceedings of the 15th International Conference on Automated People Movers and Automated Transit Systems*, Toronto, 2016. <https://doi.org/10.1061/9780784479797.027>
- [2] Morgantown personal rapid transit system. System operation description manual. M-PRT-1-1. <http://assets.slate.wvu.edu/resources/1610/1406301617.pdf>, Accessed Jun. 15, 2017.
- [3] Lees-Miller JD. Minimising average passenger waiting time in personal rapid transit systems. *Annals of Operations Research* 236(2), 2016, pp. 405-424. <https://doi.org/10.1007/s10479-013-1492-3>
- [4] Fagnant DJ, Kockelman KM. The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios. *Transportation Research Part C: Emerging Technologies* 40, 2014, 1-13. <https://doi.org/10.1016/j.trc.2013.12.001>
- [5] Kockelman K, Boyles S, Stone P, Fagnant D, Patel R, Levin MW, Sharon G, Simoni M, Albert M, Fritz H, et al. An Assessment of Autonomous Vehicles: Traffic Impacts and Infrastructure Needs Final Report. Tech. rep., 0-6847-1, Retrieved from <http://library.ctr.utexas.edu/ctr-publications/0-6847-1.Pdf>, 2017.
- [6] Hanna JP, Albert M, Chen D, Stone, P. Minimum cost matching for autonomous carsharing. *IFAC-Papers OnLine*

- [19] Babicheva T, Burghout W, Andreasson I, Faul N. The matching problem of empty vehicle redistribution in autonomous taxi systems. *Procedia computer science*, 2018, 130, pp. 119-125.  
<https://doi.org/10.1016/j.procs.2018.04.020>