

Mobility and Smart Cities

Carpooling

Presentation

Dynamic carpooling is a concept that has been gaining popularity in recent years as a solution to the problems of traffic congestion, pollution, and high fuel costs. It involves the sharing of private vehicles by multiple passengers who are traveling in the same direction, thereby reducing the number of cars on the road and the associated negative impacts.

The traditional carpooling model involves a group of people who agree to share a ride to work or other destinations on a regular basis. However, this model has limitations, such as the need for a fixed schedule and route, and the difficulty of finding compatible carpool partners.

Dynamic carpooling, on the other hand, is a more flexible and efficient model that allows users to find and join carpool trips in real-time, based on their current location and destination. It also allows for the possibility of transshipment, which means changing vehicles during a trip to reach a final destination.

Problem

Dynamic Carpooling Intra-Modal with Transshipment is a multi-constraints and multi-objectives problem.

Objectives:

- Maximization of contractualized requests number
- Maximization of Occupancy Rate
- Minimization of vehicles number (VC)

Constraints:

- Normalization constraint of time windows
- Capacity constraint
- Constraint of modal shift's authorization: role variability (only driver, only passenger, rather driver, rather passenger)
- Identification constraint of a return itinerary
- Dynamic constraint of deposited vehicle's recuperation
- Constraints of inclusion between organization's trips

Conception

The Carpooling algorithm allows the user to share a ride with the owner of a car. It can take into account the number of seats available as well as the choices of path to take (for example, with detours) and to switch from one car to another. Therefore it must manage the trip, pick up locations as well as the delivery and the time constraints of each user.

The algorithm is based on a combination of normalization of itinerary/trip/journey , local validation of time windows , and propagation of time windows and trip generation . The algorithm aims to match potential carpoolers based on their origin, destination, and time windows, while also taking into account the constraints of the system, such as the maximum detour time and the maximum number of passengers per vehicle. The algorithm is designed to be scalable and efficient, and it has been tested and evaluated using real-world data to demonstrate its effectiveness.

Inputs

- Trip origin and destination: The system needs to know the starting and ending points of each trip.
- Time windows: The system needs to know the time windows during which each trip is available.
- Vehicle capacity: The system needs to know the maximum number of passengers that can be accommodated in each vehicle.
- Maximum detour time: The system needs to know the maximum amount of time that a driver is willing to deviate from their original route to pick up or drop off a passenger.
- Geographic data: The system uses geographic data from OpenStreetMap to calculate the distances and travel times between different locations.

Outputs

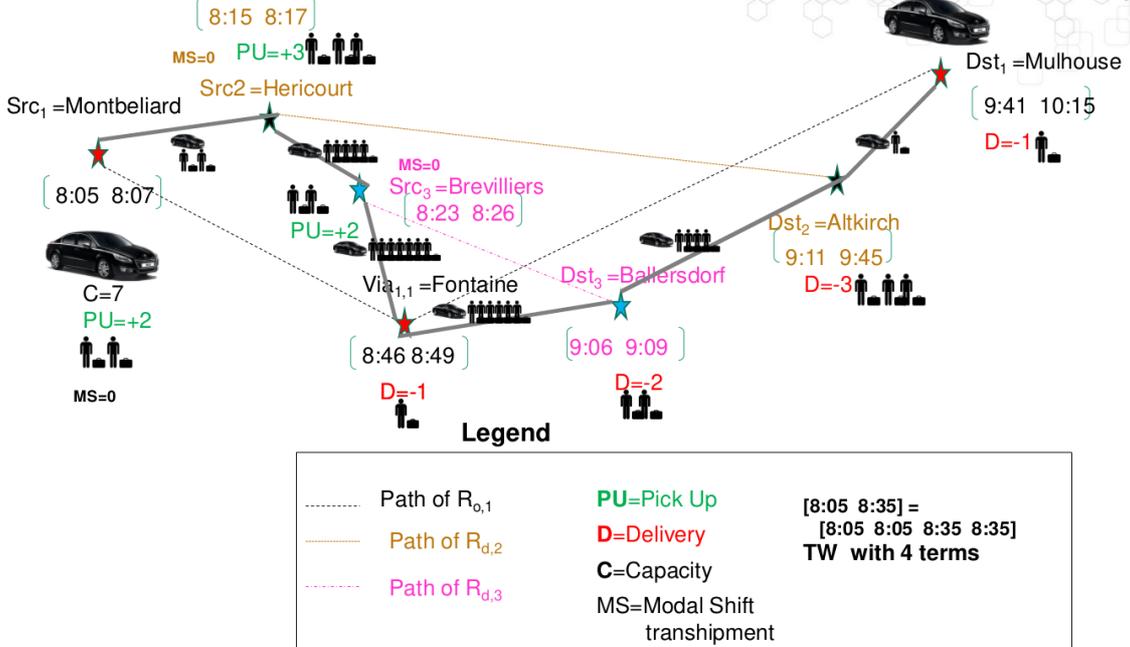
The system generates the following output:

- Matched trips: The system generates a list of matched trips, indicating which trips have been successfully matched with other trips based on their origin, destination, and time windows.
- Matched passengers: The system generates a list of matched passengers, indicating which passengers have been successfully matched with other drivers based on their trip requirements and the constraints of the system.
- Routing information: The system generates routing information for each matched trip, indicating the optimal route that the driver should take to pick up and drop off passengers.
- Cost information: The system generates cost information for each matched trip, indicating the cost savings that each participant will achieve by carpooling instead of driving alone.

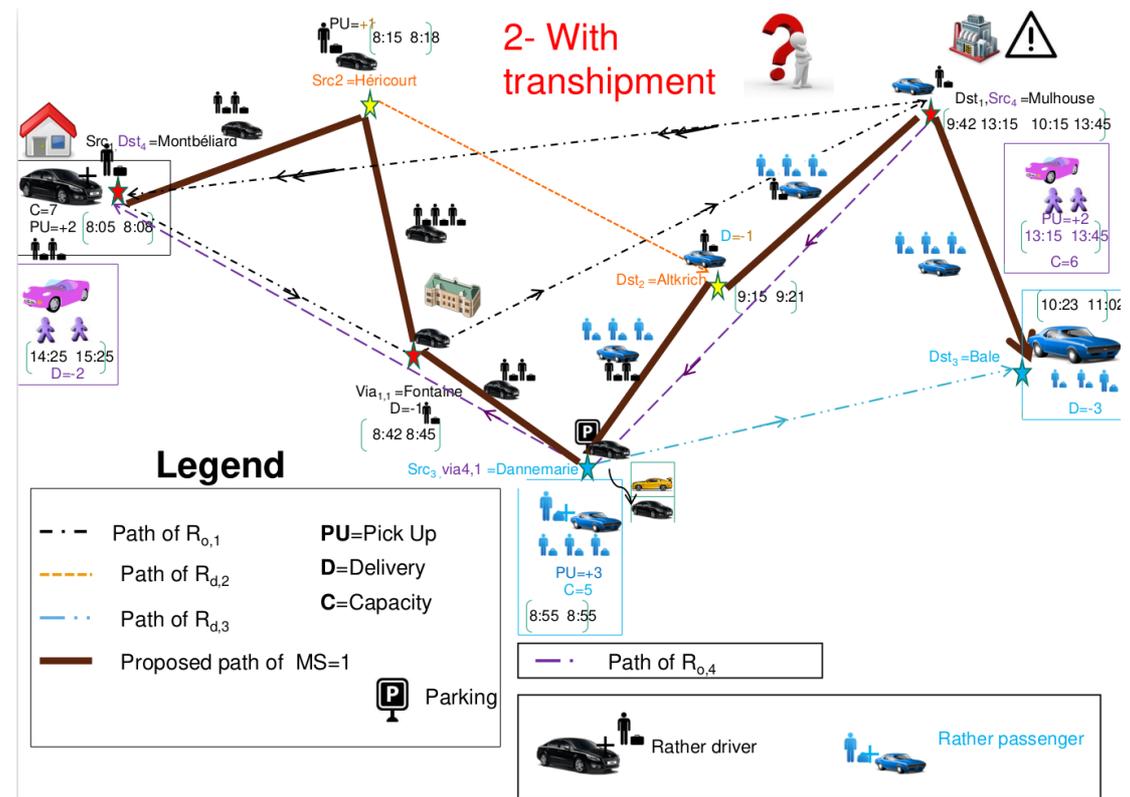
Examples

With Detours Without Transshipment

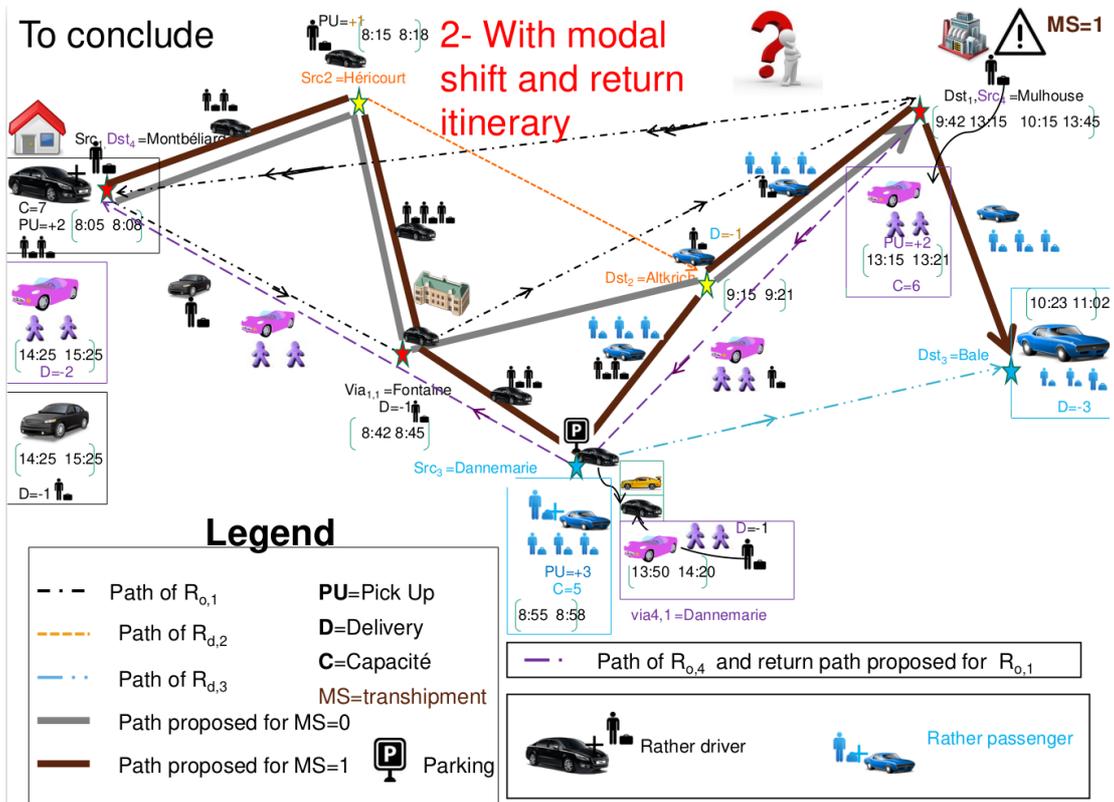
1- with detours without transshipment



With transshipment



With transshipment



2- With modal shift and return itinerary