[Smart Cities | Free Full-Text | A Predictive Vehicle Ride Sharing Recommendation System for Smart Cities Commuting (mdpi.com)](https://www.mdpi.com/2624-6511/4/1/10)

The use of a private vehicle per single passenger transportation is no longer viable in sustainable Smart Cities (SC) because of the vehicles’ resource allocation and urban pollution. The current research on car ride sharing systems is widely expanding in a range of contemporary technologies, however, without covering a multidisciplinary approach. The adopted system also provides a recommendation to citizens to select the persons they would like to commute with. An Artificial Intelligence (AI)-enabled weighted pattern matching model is used to assess user movement behavior in SC and provide the best predicted recommendation list of commuting users. Citizens are then able to engage a current trip to next destination with the more suitable user provided by the list. Software-Defined Networking (SDN) provides the necessary technology to achieve a vehicle routing protocol for improving rush hour delay in SC environments. Dealing with taxi routing trips incorporates certain citizen mobility distribution rules of pick-up and drop-off SC locations, which provide accurate information of the origin and the destination of the commute. A sustainable shared mobility car riding system is proposed for commute, aiming to reduce traffic at rush hours in SCs. Uber 🡪 batch matching policy for best serving riders and drivers.

Input 🡪 will take daily GPS then depending on that predict next destination then suggest people for ride share.

current location of a pivot user invoke the stochastic historic places (i.e., historic window size 𝑚) she has visited in the near past and predict the future location of the user (i.e., prediction window size 𝑙). Time is encoded implicitly with regards to the sequentially modeling of the represented ordered historic places the user has visited, thus a user cannot get to work early in the morning if she doesn’t leave her house first.

<https://www.sciencedirect.com/science/article/pii/S0360835220307506>

This paper provides an updated review on car-sharing optimization studies (including ride-sharing and carpooling), compares different analytical approaches in this research area, and discusses the emerging concept of ‘**agile’** algorithms as one of the approaches that might contribute to deal with the requirements of large-scale and dynamic car-sharing optimization problems. This is the **goal of carpooling and ride-sharing strategies**, which, apart from generating substantial economic impact to users, aim at reducing the number of vehicles on the road and, as a consequence, contribute to diminishing traffic and pollution. Carpooling and ridesharing are two of the main peer-to-peer (P2P) services in car-sharing. PSP services followed the diffusion of smartphone technology and social networking websites (Prieto, Baltas, & Stan, 2017), transforming car-sharing services into an international transportation trend. Such services rely on sharing privately owned vehicles for a particular trip in the surrounding area on an hourly or daily basis. Different types of ride-sharing can be identified in the literature: (i) ride-sharing with static requests –in which all requests are known before the trip starts (Yu, Wu, & Wang, 2019); (ii) ride-sharing with dynamic requests –where new requests can be added during the execution of the transport service (Simonetto, Monteil, & Gambella, 2019); and (iii) ride-sharing with either deterministic or stochastic requests (Long, Tan, Szeto, & Li, 2018). Agile optimization algorithms are able to provide high-quality solutions in real-time by combining biased-randomized algorithms ([Grasas, Juan, Faulin, de Armas, & Ramalhinho, 2017](https://www.sciencedirect.com/science/article/pii/S0360835220307506" \l "b0200)) with parallel computing ([Malapert, Régin, & Rezgui, 2016](https://www.sciencedirect.com/science/article/pii/S0360835220307506#b0365)). By taking advantage of these two approaches, the resulting methodology is capable of efficiently responding to every piece of new information that is being continuously incorporated into the system.

**Ridesharing vs Carpooling:**

Ridesharing complexity relies mainly on matching individuals subject to spatiotemporal constraints, which must be specified from both parties –i.e., drivers and users– before the desired ride is established and executed. Unlike ride-sharing activities, carpooling rides are less flexible activities that aim to transport simultaneously several people from a common starting point to a common end point ([Nechita, Crişan, Obreja, & Damian, 2016](https://www.sciencedirect.com/science/article/pii/S0360835220307506#b0410)), with the main goal of saving money. These services encourage commuters who are moving in the same direction to share private vehicles. **CPLEX** commercial optimization software for ride sharing. Likewise, [Hosni, Naoum-Sawaya, and Artail (2014)](https://www.sciencedirect.com/science/article/pii/S0360835220307506#b0240) proposed a [Lagrangian](https://www.sciencedirect.com/topics/mathematics/lagrangian" \o "Learn more about Lagrangian from ScienceDirect's AI-generated Topic Pages) decomposition approach to maximize the total profit in a ride-sharing problem –i.e., their goal was to minimize the vacant seats, taxi fares to passenger, and number of vehicles. mixed-integer [linear program](https://www.sciencedirect.com/topics/engineering/linear-program) (MILP) to solve a ride-sharing problem, resolved by the CPLEX, and to validate a metaheuristic approach. In **Carpooling** CPLEX was shown to solve the integer programs –i.e., the matching process– in a few seconds in all tested settings, which suggests that the algorithm is appropriate for use in practice.  The authors used an insertion heuristic to insert new passengers into live rides and developed a local neighborhood search (LNS) to solve this complex variant. Two real-life data-sets are used in order to test their LNS: the New York City taxi data-set and the Melbourne metropolitan area data-set. Apart from proposing an exact approach for solving the ride-sharing problem, [Hosni et al. (2014)](https://www.sciencedirect.com/science/article/pii/S0360835220307506#b0240) also introduced an [incremental cost](https://www.sciencedirect.com/topics/engineering/incremental-cost) heuristic to solve the dynamic version of the problem. In this version, the location of the seekers appears in real-time. For each taxi vehicle, whenever a new request arrives, a [minimization problem](https://www.sciencedirect.com/topics/engineering/minimization-problem) is solved. This allows to compute the additional cost when including it into the route. The objective is to maximize the total profit –i.e., minimizing the vacant seats, taxi fares to passengers, and the number of vehicles. three different algorithms to solve the dynamic shared-taxi-dispatch problem: a nearest vehicle dispatch (NVD) algorithm, an insertion heuristic (IS), and a hybrid-simulated annealing (HSA).

**Challenges related to synchronization & coordination:**The objectives when solving the ride-sharing problem are: *(i)* to minimize the driving distance, detour distance, commute costs, vacant seats, taxi fares to passengers, and the number of vehicles; and *(ii)* to maximize the total profit obtained from serving the involved riders –possibly including parcel requests (e.g., [Li, Krushinsky, Reijers, & Van Woensel (2014)](https://www.sciencedirect.com/science/article/pii/S0360835220307506#b0305))– while indirectly protecting the environment and reducing fuel consumption as well as traffic in urban areas.

When considering a real-world application, AVs-related issues are affected by a lot of external factors that change the standard and expected behavior of the involved variables ([Levin et al., 2017](https://www.sciencedirect.com/science/article/pii/S0360835220307506#b0300)). For instance, when immersed in a realistic scenario, such as the city centers –in which pedestrians and vehicles share a shared space– decisions must be taken in real-time and dynamically. It might be the case when a pedestrian crosses the road at the wrong time, or when a [traffic accident](https://www.sciencedirect.com/topics/engineering/highway-accidents) happens. Another example can be described as a road that is blocked off, or even when obstacles are found in the roads. Therefore, in order to solve the resulting problem dynamically and efficiently, the solving methodology must be able to deal with uncertainty, dynamism, and unexpected events during the execution of the planning routes.