**Abs**

Real-time ride-sharing, which enables on-the-fly matching between riders and drivers (even en-route), is an important problem due to its environmental and societal benefits. With the emergence of many ride sharing platforms (e.g., Uber and Lyft), the design of a scalable framework to match riders and drivers based on their various constraints while maximizing the overall profit of the platform becomes a distinguishing business strategy.\\

A key challenge of such framework is to satisfy both types of the users in the system, e.g., reducing both riders' and drivers' travel distances. However, the majority of the existing approaches focus only on minimizing the total traveled distance of drivers which is not always equivalent to shorter trips for riders. Hence, we propose a fair pricing model that simultaneously satisfies both the riders' and drivers' constraints and desires (formulated as their profiles). In particular, we introduce a distributed auction-based framework where each driver's mobile app automatically bids on every nearby request taking into account many factors such as both the driver's and the riders' profiles, their itineraries, the pricing model, and the current number of riders in the vehicle. Subsequently, the server determines the highest bidder and assigns the rider to that driver. We show that this framework is scalable and efficient, processing hundreds of tasks per second in the presence of thousands of drivers. We compare our framework with the state-of-the-art approaches in both industry and academia through experiments on New York City's taxi dataset. Our results show that our framework can simultaneously match more riders to drivers (i.e., higher service rate) by engaging the drivers more effectively. Moreover, our framework schedules shorter trips for riders (i.e., better service quality). Finally, as a consequence of higher service rate and shorter trips, our framework increases the overall profit of the ride-sharing platforms.

**Intro**

Real-time ride-sharing is an important problem because it can ease traffic congestion, decrease auto emissions, while at the same time serve the transportation needs. With the emergence of many commercial platforms (e.g., Uber and Lyft) which automatically match drivers and riders on-the-fly, real-time ride-sharing becomes more and more popular. According to~\cite{uberpool}, millions of trips have been taken on UberPool since its launch at August 2014, and thousands people take it five times a week during commuting hours. Enabled by the development and technology advances of smart phones and location-based services, ride sharing platforms typically operate as follows: (1) Riders and drivers can join the platform via their smart phones, (2) a rider can submit a request, which consists of the origin and destination points, to the platform. (3) Once a new request is received, the platform determines a driver (even en-route) to pick up the rider, (4) When the trip is completed, the platform calculates the riders' fare and the drivers' income. With these platforms, riders can share a vehicle with reduced trip cost while enjoying fast and convenient transportation.

Many challenges exist to enable such real-time ride-sharing platforms. From a business point of view, the platform provider (e.g., Uber) seeks to maximize its own revenue. However, higher profits should not be earned by either charging passengers more or paying drivers less than what will make participating and retaining in the system monetarily incentive for both parties. Consequently, the design of a fair pricing model becomes an essential business strategy. This is particularly important in cases of carpooling where riders share their ride with another rider. Even though carpooling reduces the riders' cost, it incurs extra distance (i.e., detour) for users. While the rider's fare should be reduced based on the amount of the detour incurred, the driver should be rewarded more as the total travel distance is increased due to detours. Furthermore, different users (i.e., riders and drivers) might value their time differently. Therefore, a fair pricing model should be provided to both riders and drivers to express, for a certain amount of detour, how much compensation they expect. Finally, in addition to fair pricing scheme for riders and drivers, the model should account for the provider's revenue as well.

The second challenge of a ride-sharing platform is to process incoming requests in real-time. This involves two different tasks: i) checking which drivers can add the new requests (new pick-up and drop-off locations) to their current trip without violating the constraints that trip (i.e., scheduling) and ii) selecting the best driver among those who can serve the new request (i.e., matching). Processing the schedule of potentially thousands of drivers to check if they can accommodate a new request, is a compute-intensive process. Therefore, the design of an efficient and scalable algorithm that assigns riders to drivers with a fair pricing model while maximizing the provider's revenue, arises as a new problem to be addressed.

The majority of previous studies~\cite{Ota15, Cici15, Cao15, PelzerITS15} focus on improving the efficiency of on-the-fly assignment with the objective of minimizing the total travel distance of drivers. In particular, in existing studies a new request is assigned to a driver who can fit the request in his schedule with the least amount of increase in the total traveled distance. However, minimizing drivers' total travel distance is not always equivalent to overall shorter trips for riders. Consequently, when assigning a new request, the driver with minimum increase in total travel distance is not necessarily the most cost effective option. For example, assume driver $a$ has two riders on board and driver $b$ has only one. To service an incoming request, \textit{a}'s incurred detour is 2 miles while for $b$ is 3 miles. Even though $a$'s detour is shorter, the framework has to compensate both riders of $a$ for 2 miles (a total of 4 miles) while in the case of $b$ it has to compensate only one rider for a total of $3$ miles. Few studies~\cite{Ma13,Ma15} consider a pricing model by defining monetary incentives for riders and drivers. In \cite{Ma13}, a pricing model is introduced where instead of being compensated, a rider can potentially end up being penalized for longer detours by paying a higher fare. Ma et. al.~\cite{Ma15} overcomes the unfairness issue in \cite{Ma13} to some extent. Even though, a new rider can incur detour in his trip, their model only compensates riders that are already on board. In addition, since this model is targeted for a different application, the notion of revenue is left failing to provide any incentive for the platform provider. Furthermore, in all previous studies~\cite{Ma13,Huang14,Ma15}, a centralized server is responsible for matching and scheduling incoming requests. Most of these studies utilize a spatiotemporal index to enable matching, i.e., narrowing down the number of potential drivers who can service the request. With thousands of drivers in the system, even after applying the spatial index, the centralized server still needs to perform scheduling for all the candidate drivers. We observe that with large number of drivers, these frameworks fail to process new requests in real-time (\cref{subsec:exprp}).

To address aforementioned challenges, in this paper, we introduce an Auction-based Price-Aware Real-time (\textit{APART}) ride-sharing framework. We propose a general and versatile pricing model that allows both riders and drivers to set their monetary expectations for participating in ride sharing based on their predefined profiles. Specifically, each rider's profile defines the expected discount ratio for the detours incurred by ride-sharing. For example, one rider can express he's willing to accept a 10 mile detour for 30\% discount. On the other hand, each driver's profile defines the expected cost in terms of his total travel distance and time. The model also accounts for the revenue of the platform provider. Consequently, our objective is to maximize the revenue of the ride sharing framework while satisfying various temporal and monetary constraints of all users. APART is price-aware because a new request is assigned to a driver which generates the highest profit. Since our pricing model is designed to compensate riders for detours, the most profitable choice is the one where \textit{riders} incur the least amount of detour. Finally, APART maximizes the revenue of the provider with increased service rate in the system by engaging available drivers more effectively to serve more ride requests.

To efficiently assign riders to the candidate drivers, we introduce a distributed auction-based algorithm. With the algorithm, the server broadcasts a new request to a set of candidate drivers and each candidate driver\footnote{Hereafter we use the term \qoutes{driver} to refer to both the human driver and the software running on his mobile device.} submits a bid based on his current schedule, his and other riders' profiles and other spatiotemporal constraints. Subsequently, the server collects all the bids from candidate drivers and assigns the rider the highest bidding driver. To guarantee high service quality, each driver runs a branch-and-bound algorithm which performs an exhaustive search to find out whether it can fit a new request into its current scheduling. Each driver carries a small number of riders so even an exhaustive search can be performed in real-time. Due to the distributed nature of APART, all candidate drivers perform the search in parallel. Once each driver finds its own best schedule, the server has to select the driver that generates the highest profit. Consequently, APART is able to find the most profitable drivers in real-time.

We conduct extensive experiments on a large scale New York City taxi dataset and show that APART is scalable and efficient, capable of processing hundreds of tasks per second in the presence of thousands of drivers.

By comparing our framework with state-of-the-art approaches~\cite{Huang14}, we show that our framework can simultaneously match up to 25\% more riders to drivers (i.e. higher service rate), while the total travel distance of riders are xxx\% less (i.e., better service quality)\dingedit{filling this number here}, hence our framework can earn more profit than other approaches. On the other hand, we show that in a framework where riders are assigned to drives with least increase in driver's travel distance, up to 25\% of the requests do not get assigned to the most profitable driver.

The remainder of this paper is organized as follows. We define our problem in \cref{sec:problem\_def}, and explain our price model in \cref{sec:pricing}. We present our APART framework, and discuss its auction-based approach in \cref{sec:framework}. In \cref{sec:exp}, we report the experiment results. We discuss the related work in \cref{sec:related} and conclude the paper in \cref{sec:conclusion}.

**Conclusion**

In this paper, we studied the problem of real-time ride sharing. Unlike existing studies, our objective is to maximize the revenue of the platform provider, hence we introduced APART with a fair pricing model so that higher profits cannot be gained by scamming the riders. With APART, riders were able to set their monetary preferences through their profiles, and consequently the trips were generated according to our price-aware assignment mechanism. We validated our framework with a large scale New York City's taxi trips data. The experiment results demonstrated the effectiveness and efficiency of our framework in terms of both service rate and quality. Also, the distributed nature of APART allowed for real-time processing of ride requests.

In the future, we plan to expand the framework by considering a time-dependent road network, where pre-computing the shortest paths between different nodes of the road network becomes impossible. Another interesting direction is to batch the process in a short amount of time (e.g., 5 seconds). This is because in large markets where the requests are submitted at a high rate, the server can wait for this short time to bundle multiple requests and process them together. It is expected to generate better assignments if multiple requests are processed at the same time. However, this introduces new challenges such as the one-request-to-many-drivers matching in APART will become a many-requests-to-many-drivers, hence matching will be more complex.