Dynamic Pricing and Matching in Ride-Hailing Platforms

Naval Research Logistics, Forthcoming

Chiwei Yan
Uber Technologies, Inc. & University of Washington Seattle, chiwei@uw.edu
Helin Zhu, Nikita Koroko, Dawn Woodard
Uber Technologies, Inc.

Ride-hailing platforms such as Uber, Lyft, and DiDi have achieved explosive growthand reshaped urban transportation. The theory and technologies behind these plat-forms have become one of the most active research topics in the fields of economics,operations research, computer science, and transportation engineering. In particular, advanced matching and dynamic pricing (DP) algorithms — the two key levers inride-hailing — have received tremendous attention from the research community andare continuously being designed and implemented at industrial scales by ride-hailing platforms. In this paper, we discuss essential models for matching and DP techniques in ride-hailing, complemented with Uber data, and show that they are critical for providing an experience with low waiting time for both riders and drivers. Then we link the two levers together by studying and analyzing a pool-matching mechanism called *dynamic waiting* (DW) that varies rider waiting and walking before dispatch, which is inspired by a recent carpooling product Express Pool from Uber. We show using data from Uber that by jointly optimizing DP and DW, price variability can be mitigated, while increasing capacity utilization, trip throughput,and welfare.

The central piece of our analysis is a steady-state closed spatial queueing model for pricing and matching in ride-hailing, perhaps first introduced in Arnott (1996) and then Castillo, Knoepfle, and Weyl (2017), but significantly new insights and analysis have been provided. In the simplest setting, drivers participating on the platform cycle through three states sequentially: "open" — waiting to be dispatched, "en route" — on the way to the pickup location, and "on-trip" — driving riders to their destination, as shown below.

$$\cdots \to \text{open} \to \text{en route} \to \text{on-trip} \to \text{open} \to \cdots$$

A request can be matched with a dispatchable driver using very simple algorithms, such as the *first-dispatch protocol*. In the first-dispatch protocol only open drivers are considered as dispatchable. Each request is immediately assigned to the open driver who is predicted to have the shortest en route time.

Consider a single-geo setting, meaning that all trips begin and end within the geographic region. We initially consider a fixed number, denoted by L, of drivers on the platform, who do not relocate into or out of the geolocation. Recall that drivers cycle through three states: open, en route, and on-trip. Using the first-dispatch protocol, only open drivers are dispatchable, and all requests are immediately dispatched. In this case, en route time is equal to rider waiting time. Further suppose each trip (from pickup to dropoff) takes a given constant time T (units of time) to complete. The steady-state trip throughput, i.e., the average number of trips completed per unit of time, is denoted by Y. We then have the flow balance equation

$$L = \overbrace{O}^{\text{open}} + \overbrace{\eta \cdot Y}^{\text{en route}} + \overbrace{T \cdot Y}^{\text{on-trip}}, \tag{1}$$

where O represents the number of open drivers, η denotes the en route time, $\eta \cdot Y$ represents the number of en route drivers, and $T \cdot Y$ represents the number of on-trip drivers. The en route time η is determined by the number of open drivers: a lower number of open drivers O leads to a higher en route time. This relationship is captured in Proposition 1.

PROPOSITION 1 **(Larson and Odoni (1981))**. *Suppose the open drivers are distributed uniformly in a n-dimensional ($n > 1$) Euclidean space. Further assume a constant travel speed on the straight line between any two points in that space. Then the expected en route time, denoted by $\eta(O)$, satisfies*

$$\eta(O) = O^{-\frac{1}{n}}, \tag{2}$$

where O is the number of open drivers.

This model also fits well empirically with data from Uber if we relax the exponent in the en route time function (2) from $-\frac{1}{n}$ to a general $\alpha \in (-1, 0)$, as demonstrated in the left plot in Figure 1.

From Proposition 1, we can rearrange the flow balance equation (1), and substitute in the expected value $\eta(O)$ for η (noting that we are approximating a random variable with its mean), to obtain $Y = (L - O)/(\eta(O) + T)$. One way to interpret this equation is through Little's Law. In this case Y represents the long-run average trip throughput, which equals the long-run average number of busy drivers in the system $(L - O)$ divided by the average time required for a driver to complete a trip. The latter is equal to the sum of en route time $\eta(O)$ and trip duration T. To illustrate

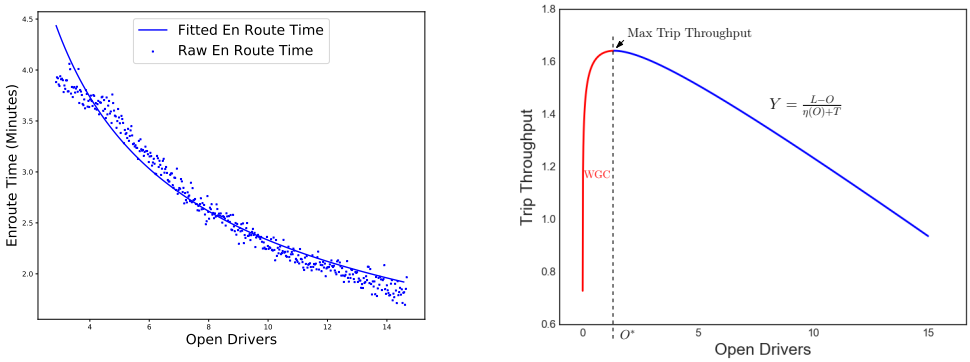


Figure 1 **The en route time and trip throughput functions. Left: the mean en route time in minutes from UberX data, and the fitted curve $\eta(O) = 7.63 \cdot O^{-0.52}$, as a function of the nearby open drivers per km^2; Right: the trip throughput Y as a function of open drivers. The model parameters are calibrated based on UberX data in San Francisco downtown during rush hour: $L = 30$ per km^2 and $T = 15.0$ minutes.**

the behavior of the trip throughput Y as a function of the number of open drivers, we show this approximation in the right plot in Figure 1. We can see that there is an open driver level O^* that maximizes the trip throughput. For $O > O^*$, the trip throughput is lower than the one at O^*; however, since the en route time $\eta(O)$ is lower, there is an experience versus throughput trade-off and there may be some cases in which it is desirable to have a value of $O > O^*$. For example, there may be a value $O > O^*$ that leads to a higher welfare than the one at O^*. In contrast, when $O < O^*$, the trip throughput decreases sharply, due to the fact that en route time grows quickly. This regime is undesirable from both the experience perspective and the trip throughput perspective. A key function of dynamic pricing is to keep the price point above the level that would lead the marketplace to enter this regime. An interesting analogy is the "*hypercongestion*" (Walters 1961, Small and Chu 2003) phenomenon in transportation economics, where the travel speed decreases significantly as traffic flow increases. This eventually leads to a sharp decrease in throughput.

While dynamic pricing brings many benefits, it also comes with downsides. The price can fluctuate due to the natural variability in local demand and supply levels. This price variability is an undesirable experience and can cause reduced engagement of both riders and drivers. We address this concern by investigating the benefits of combining DP with a pool-matching mechanism called dynamic waiting (DW), which allows batching of requests and varies rider waiting time before dispatch in order to find a good pool-match. This mechanism is inspired by the recent ride-sharing product Express Pool from Uber. Assume that each car can hold up to two riders, and that all riders opt into pooling. In DW, two riders can be pool-matched if their origin locations as well as destination locations are close; i.e., if the distance between the origins is within walk-able range (*walking radius*), and similarly for their destinations. Additionally, a rider can be asked to wait up

to a certain duration (*waiting window*) before receiving a dispatch. When a rider requests a ride, the system first checks whether there is a matching rider currently waiting. If so, these two riders are matched immediately, and the car with the shortest predicted en route time is dispatched to pick them up simultaneously at the midpoint of their origins, and drop them off at the midpoint of their destinations. If no matching rider is available, the rider is asked to wait. If a pool-match is found within the waiting window, the match and dispatch are made; otherwise the system dispatches an open driver to this rider's origin location and will not attempt to pool-match this rider going forward. The resulting rider waiting time is then the sum of the dispatch waiting time and the en route time. An example is shown in Figure 2.

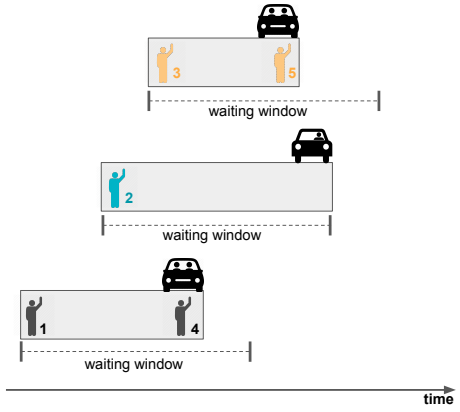


Figure 2 **Dynamic Waiting (DW) for pool-matching. Riders are numbered in order of arrival time and pool-matched riders are shown in the same color (e.g., Riders 1 and 4 are pool-matched, while Rider 2 gets their own car).**

We show that DW could be used as a complement to DP to alleviate the WGC, reduce price volatility, and increase trip throughput and welfare. By jointly optimizing DP and DW, we show how one can explicitly trade off price with rider waiting time. See Figure 3 below.

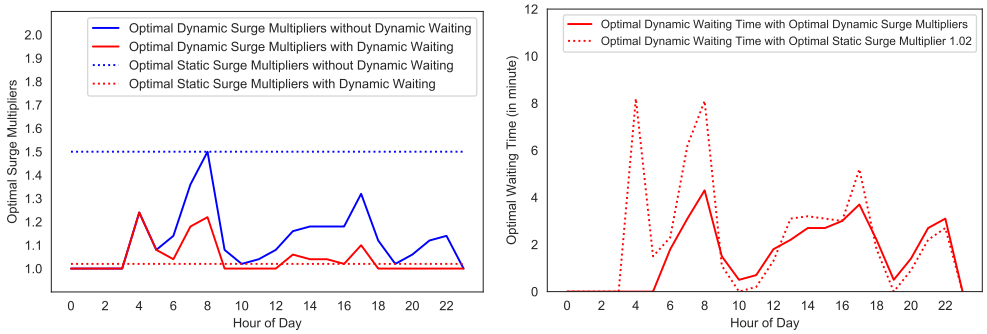


Figure 3 **Optimal prices and waiting windows by hour-of-day. Left: optimal dynamic and static prices; Right: optimal DW windows under DP and static pricing.**

References

Arnott R, 1996 *Taxi travel should be subsidized. Journal of Urban Economics* 40(3):316–333.

Castillo JC, Knoepfle D, Weyl G, 2017 *Surge pricing solves the wild goose chase. Proceedings of the 2017 ACM Conference on Economics and Computation*, 241–242 (ACM).

Larson R, Odoni A, 1981 *Urban operations research* (Prentice Hall PTR), ISBN 9780139394478, URL `https://books.google.com/books?id=5moeAQAAIAAJ`.

Small KA, Chu X, 2003 *Hypercongestion. Journal of Transport Economics and Policy (JTEP)* 37(3):319–352.

Walters AA, 1961 *The theory and measurement of private and social cost of highway congestion. Econometrica: Journal of the Econometric Society* 676–699.