The Efficiency of A Dynamic Decentralized Two-sided Matching Market

Tracy Xiao Liu
Tsinghua University

Zhixi Wan
DiDi Chuxing
University of Oregon

Chenyu Yang*
Simon Business School
University of Rochester

June 2018

Abstract

This paper empirically studies a decentralized dynamic matching market. We use data from a leading ride-sharing platform in China to estimate a continuous time dynamic model of search and match between drivers and passengers. In counterfactual simulations, we assess the efficiency of the decentralized market and examine how centralized algorithms may improve welfare. We find that the strategic waiting incentive in the decentralized market increases market thickness and average match quality but decreases the number of matches. Compared with the equilibrium in the decentralized market, centralized algorithms can increase both the match quality and the number of matches by making matches less frequently and matching agents more assortatively.

*Correspondence to: Simon Business School, CS3.219, University of Rochester, Rochester, NY 14620. Email address: chny.yang@gmail.com

1 Introduction

Decentralized two-sided matching markets serve millions of people every year across the world with the rise of the sharing economy. Prominent examples include the accommodation platform Airbnb and ride-sharing platforms such as DiDi Hitch in China and BlaBlaCar in Europe. These markets have several features: (1) agents enter and leave the market over time, (2) agents on one side decide whether to match with agents from the other side[1] and (3) agents have heterogeneous preferences for partners. We use data from DiDi Hitch to empirically study the strategic incentive in a decentralized market and quantify the gains from implementing centralized matching mechanisms. We use the number of matches and average match quality to measure market efficiency. Because centralized algorithms require information that may not be known to agents in a decentralized market, we discuss what information is valuable and quantify the value of information in different centralized algorithms. We consider algorithms that are easily implementable. These algorithms improve the market efficiency in two ways. First, the centralized algorithms can match more agents by keeping agents in the market longer and creating additional match opportunities. Secondly, the centralized algorithms can improve match qualities by taking into account the externalities. We show that the efficiency gains from implementing centralized algorithms depend critically on the level of competition and the heterogeneity of agent preferences.

We use proprietary ridership data from DiDi Chuxing to study our research questions. DiDi Chuxing is the leading firm in the enormous Chinese ride-sharing market. DiDi offers tiered ride-sharing platforms. According to the *2017 DiDi Factsheet*, the company's platforms served 450 million passengers in 2017, and 25 million trips were completed every day. The main operation of the company, DiDi Express, is similar to Uber or Lyft, where a central dispatch algorithm matches drivers to passengers. Our empirical context is the smaller and decentralized platform of DiDi Hitch. In 2017, 2.23 million rides occurred on this platform

[1]Service providers usually make the final decision. On Airbnb, hosts have the ultimate decision right on whether to accept a guest. Drivers on DiDi Hitch and BlaBlaCar decide whether to accept passengers.

during the peak day.

The matching rule on the DiDi Hitch platform during our sample period provides a unique setting to study the effect of strategic waiting. To receive a ride, a passenger sends a request to the platform. The request consists of a pickup and a dropoff location and a departure time. If a request is answered, the answering driver will deliver the passenger according to the conditions specified in the request. The passenger cannot see available drivers or request a particular driver. The only actions available to the passenger are either to wait or cancel the request. To find and answer the most suitable passenger request, a driver can input a specified route into the ride-sharing app that sorts all ride requests according to a compatibility index. Drivers choose between waiting for a better match and answering a request now. Throughout the rest of the paper, we specifically refer to the additional driver waiting (compared with a myopic driver) induced by the desire to wait for a more compatible passenger as strategic waiting. We conceptualize the implication of strategic waiting in a simple framework in Section 3. In particular, we show that strategic waiting and preference heterogeneity play key roles in determining the equilibrium number of matches and the efficiency gains from implementing centralized algorithms.

To address our research questions and exploit the real time ridership data, we develop a continuous time dynamic model of search and match. Passengers and drivers with preferences for different routes arrive at the market stochastically. A route is defined as a pair of pickup and dropoff locations. Passengers send out their trip requests consistent with their true preferences immediately after arriving. Passengers and drivers stochastically leave the market without a match. The driver receives a reservation utility in this case. While in the market, a driver also stochastically receives move opportunities to check her phone, and she can choose to wait or answer a request closest to her ideal route. The level of mismatch between the driver's preferred route and a requested route is measured by a scalar. If a driver answers the request, the driver receives utility as a function of the mismatch, and both the driver and the passenger leave the market. When answering a request, a

driver compares the utility from picking up the current most compatible passenger with the option value of waiting for a potentially better future match. To evaluate the option value of waiting, the driver is assumed to know the stationary distribution of the minimal mismatch. We define a stationary Markov Perfect Equilibrium based on this driver dynamic optimization. The distribution of the minimal mismatch is endogenously determined by the entry rates, the no-match exit rates and the equilibrium driver strategies. In particular, the distribution of the minimal mismatch depends on how fast passengers leave the market, which is composed of the exogenous no-match exits and the endogenous exits due to drivers answering requests. Drivers effectively play against a stationary distribution of the minimal mismatch in this equilibrium. In robustness checks, we also consider allowing drivers to condition their strategies on non-payoff relevant variables.

Driver app usage was not available when we collected the data, which forces us to work with just the data on passenger requests. We use request data to provide evidence that drivers wait and they are selective with requests. We use the intertemporal variations in requests and the structural model to identify (1) driver tolerance of mismatch (preference), (2) how much drivers and passengers wait (the exogenous rate at which agents exit without a match) and (3) driver and passenger arrival rates. These parameters are crucial to our research questions: intuitively, the number of matches in a decentralized market increases in the arrival rates and the tolerance of the mismatch and decreases in the exit rates. In the extreme, if drivers and passengers can stay in the market infinitely long and are indifferent across potential partners, the number of matches in the decentralized market would be close to the social optimum. We can identify passenger arrival rates directly from the data, and we argue that the observed percentage of matched requests help to identify the preference. Variations of answer rates across time and the number of passenger requests help to identify the no-match exit rates and arrival rates. In particular, we show that there exist sufficient variations in data that can support an "identification at infinity" argument. The model is estimated via a two-step indirect inference method.

Using the estimates, we conduct counterfactual simulations to address our research questions. We first simulate the market with myopic drivers. The comparison with the decentralized equilibrium market reveals the implication of strategic waiting. We find that although waiting increases the average number of market participants on both sides of the market at a given moment (higher market thickness) and allows drivers to obtain better matches (higher average driver utility), the option value of waiting also causes drivers to forgo more matches and results in a lower number of matches. Next, we assume that the platform has additional information on agent preferences and when the agents will leave the market without a match, and we simulate three centralized matching algorithms. The first two simulations use the centralized greedy and patient algorithms in Akbarpour, Li and Oveis Gharan (2017), adapted to account for the two-sidedness of the empirical application. These simulations help to quantify the effect of improved sorting and increased market thickness. Finally, we simulate an assortative matching algorithm, where the platform solves a linear programming problem every 5 minutes to maximize the total driver utility. The assortative matching algorithm thus takes into account the externality between matches. With agents ex ante identical in terms of how likely they are compatible with other agents, Akbarpour et al. (2017) shows that the patient algorithm can increase market thickness and the number of matches by keeping agents in the market longer. In our context, drivers have heterogeneous preferences, but we still find that the patient algorithm achieves the highest level of market thickness, number of matches and average driver utility at our estimates. The periodic assortative matching algorithm also improves upon the decentralized equilibrium.

The counterfactual simulations are highly relevant to the operations of the Hitch platform for two reasons. First, the result quantifies the magnitude of the increase in the number of matches by increasing the waiting time of the agents. If the platform were to switch to a centralized system, the platform can quantitatively trade off the additional revenue against the cost of implementation. Secondly, the result also quantifies the value of information. When the platform only collects driver preferences for routes, it can improve market efficiency

by implementing the assortative matching algorithm. When the platform collects (or uses predictive tools to infer) both the driver preferences and the time when an agent would leave the market without a match, it can implement the patient algorithm to achieve an even higher level of market efficiency.

Relations to the Literature and Contributions

This paper is an empirical study on dynamic matching efficiency in a decentralized two-sided market. This paper complements a number of theoretical studies on dynamic matching in economics, computer science and operation research. Some of the recent work (e.g., Arnosti et al. (2015); Baccara et al. (2016); Loertscher et al. (2016); Ashlagi et al. (2016); Akbarpour et al. (2017)) study optimal dynamic matching and thickness. Market thickness is defined as the number of market participants. In particular, Akbarpour et al. (2017) studies algorithms to increase the number of matches between ex ante identical agents. Ashlagi, Burq, Jaillet and Manshadi (2016) studies the algorithms to minimize the waiting time of agents that differ in the difficulty to match. Our environment and objective are more similar to Akbarpour et al. (2017). Based on our empirical context, we assume that the waiting and search costs are low, and ex-ante heterogeneous agents face (stochastic) deadlines to either make a match or leave the market. A key difference from the theoretical framework in Akbarpour et al. (2017) is that the agents in our model are not identical in terms of how likely they are compatible with other agents. In our context, a driver might be compatible with passengers 1 and 2, but another driver might be compatible with only passenger 1. We show by simulation that the patient algorithm may not necessarily generate more matches than the greedy algorithm with heterogeneous compatibility at some parameterization. There is also a theoretical literature on kidney exchanges (e.g., Roth et al. (2005, 2007); Ünver (2010)) that examines properties of centralized dynamic matching algorithms. Finally, a growing number of papers in operation research address the problem of dynamic matching mechanism design in the ride-sharing industry (e.g., Banerjee, Riquelme and Johari (2015); Ozkan and Ward (2016);

Hu and Zhou (2016); Feng, Kong and Wang (2017)).

The dynamic search and match problem has been studied both theoretically and empirically in a large body of the labor literature (surveyed in, e.g., Mortensen (1986); Pissarides (2000); Shimer and Smith (2001); Canals and Stern (2002); Rogerson, Shimer and Wright (2005); Eckstein and Van den Berg (2007)). A main goal of this literature is to examine the implications of search frictions. The model for our empirical context allows us to unpack the various sources of search frictions and discuss their implications separately. In particular, we show that strategic waiting does not necessarily increase the number of matches, even though it increases market thickness and decreases the search friction that results from an inter-temporal mismatch (compatible drivers and passengers not arriving at the market at the same time). Like many papers in this literature, we model the search as a dynamic discrete choice problem.

Our paper is related to the broad empirical literature on trade frictions/market design failures and allocative efficiency (e.g., Allen, Clark and Houde (2014); Gavazza (2011, 2016); dynamic auctions: Adachi (2016); Hendricks and Sorensen (2016); Bimpikis, Elmaghraby, Moon and Zhang (2017); Bodoh-Creed, Boehnke and Hickman (2017); taxi market: Lagos (2003); Frechette, Lizzeri and Salz (2016); Buchholz (2017); kidney exchange: Agarwal, Ashlagi, Azevedo, Featherstone and Karaduman (2017)). We quantify how and to what extent heuristic algorithms provably close to the optimal matching algorithm under certain conditions (Akbarpour et al. (2017)) mitigate the inefficiency from non-assortative matching as well as the inter-temporal mismatch. A key difference from the cited papers on taxi search is that we highlight the role of market thickness when agents on both sides of the market have heterogeneous preferences across matches. In this case, the ratio of matched passengers increases when the numbers of drivers and passengers both increase, even if the driver-passenger ratio is held fixed. In contrast, if the (expected) match value for a driver is the same across matches (as is the case of professional taxi drivers who cannot discriminate among passengers), the ratio of matched passengers is the driver-passenger ratio (if less than

1) in the absence of other search frictions.

A number of recent papers study pricing on the ride-sharing network. Many of these papers focus on the use of dynamic pricing to increase professional drivers' labor supply and passenger-driver matching on Uber (e.g., Hall et al. (2015, 2017); Castillo et al. (2017)). Ostrovsky and Schwarz (2018) study how proper road pricing can implement the efficient allocation of passenger to the road capacity in a carpooling context with autonomous cars and non-professional drivers whose waiting cost is low. The pricing rule on the Hitch platform has a simple two-part tariff structure strictly based on the trip distance during our sample period and we do not study pricing in this paper. We focus on how to improve the allocative efficiency in a market of passengers and non-professional drivers through non-pricing channels.

Many papers on empirical matching models precede us (surveyed in e.g., Choo and Seitz (2013); Chiappori and Salanié (2016); Fox (2017)). Many of these papers use data on who matches with whom and an equilibrium model of matching to estimate preference primitives. These papers typically examine long term relationships or relationships formed under particular market designs, and the matches can reasonably be assumed to satisfy a notion of stability (Roth and Sotomayor (1992); Hatfield, Kominers, Nichifor, Ostrovsky and Westkamp (2013)). Choo (2015) studies the gains of marriage in a frictionless dynamic matching market. Fox (2008) studies repeated matching between forward-looking workers and firms. Relationships in our empirical contexts are typically short-term and agents are unlikely to coordinate to swap partners after the initial match. We use this setting to study the frictions that may prevent matches from being stable as defined in the cited work.

Our model is based on the continuous time dynamic game framework (Doraszelski and Judd (2012); Arcidiacono, Bayer, Blevins and Ellickson (2016)). We focus on a stationary equilibrium where drivers who cannot observe their competition (by the design of the platform) make decisions based on the stationary distribution of the payoff-relevant variable. This equilibrium concept is related to the oblivious equilibrium (Weintraub, Benkard and

Van Roy (2008)), the mean field equilibrium (Iyer et al. (2014)) and the equilibrium concepts used in Krusell and Smith (1998), Backus and Lewis (2016), Bodoh-Creed et al. (2017), Buchholz (2017) and others.

Road Map In the rest of the paper, we first discuss the empirical context and the data. We next present a simple theoretical model of search and match to highlight when the decentralized market may produce a fewer number of matches than a social planner. We then describe our empirical structural model in detail, followed by the estimation, counterfactual experiments and robustness checks.

2 Empirical Context

We use 20 weekdays of detailed request-level data on the DiDi Hitch platform during 7:30AM and 8:30AM in a prefecture-level city in China in the summer of 2016. The first subsection discusses institutional details that inform our data choice and motivate the empirical model later. The second subsection presents features of the data that support modeling assumptions. In Section 5.1, we present additional patterns in data that help to identify model primitives.

2.1 Institutional Details

On the DiDi Hitch platform, a prospective passenger sends a request to the platform to receive a ride. The request consists of the desired pickup location, dropoff location and the pickup time. The passenger does not observe the driver and cannot choose the driver. A prospective driver can view all requests via the platform's app. The first driver that answers a request "wins" the trip. A driver can input her own desired pickup and dropoff locations and sort trip requests by a notion of compatibility. For each request, the driver observes a single percentage rate calculated by the system, in addition to the detailed specifications

of the request. We do not know exactly how the system calculates the mismatch displayed to the drivers, but the mismatch is largely based on distance. In addition to the level of compatibility, many random factors such as cellphone network speeds and how often a driver checks the phone play a role in which driver gets a ride. The app can push messages to drivers whose compatibility index with a ride request is above a certain level, but multiple drivers can still get the push messages simultaneously, and which one of these drivers gets the ride is still rather random.[2] A driver does not have to reveal its route preference to the platform. We do not observe the locations of waiting drivers or their preferences. A passenger can cancel the ride at any time before any driver answers the request without penalty. The platform charges passengers \$0.8 for the first 2KM and \$0.15/KM for the rest of the trip. The platform collects 10% of the total charge and the driver earns the rest. In comparison, a taxi costs \$1.5 for the first 3KM and charges \$0.3/KM for the additional distance. DiDi ran promotions during our sample period and often awarded drivers by a lump sum subsidy for completing a trip. The actual amount varies. The average is about \$1.

There are several institutional features that guide our data and modeling choices. First, when we collected the data, the DiDi platform was introduced about one year ago and the user base was growing. We therefore chose days from four consecutive weeks so that the growth trend is not substantial. Secondly, we use the data during the morning rush hour period to reduce the heterogeneity of driver profiles in the sample and focus on the decision of non-professional drivers. Professional (DiDi Express) drivers may use the platform to move locations, and therefore multiple destinations may be attractive. Non-professional drivers may only be interested in going to one location. It is hard to identify the intentions of the drivers directly from data. Identifying the types of drivers is of its own separate interest but a second order issue to our main research questions. We argue that our data choice minimizes the concern of driver heterogeneity. During the morning rush hour, DiDi Express is in high demand, and therefore the need for professional drivers to use the DiDi Hitch

[2]A new project with DiDi currently looks at how to improve the sorting with alternative messaging systems.

platform to move locations is low, but their opportunity cost of waiting for a ride on the DiDi Hitch platform is high. In addition, the platform instituted rules to differentiate itself from the DiDi Express and appeal to nonprofessional drivers. For example, a driver must wait at least 30 minutes after the dropoff time of the last ride to answer a new request, and an algorithm unknown to us sets limits on how far apart the origins of the two rides must be. These rules made the platform less attractive to professional drivers and reduce the concern of cannibalization with DiDi Express.

When a driver answers a ride, a match is formed and there are penalties for the party that cancels the ride. Multiple cancellations can lead to suspension of the account. About 8% of the matches are canceled in our data. According to conversations with DiDi, many of the cancellations appear to be "random", such as the passenger no longer needing the ride, instead of "strategic", such as a driver noticing a better fit.[3] We view increasing the rate at which drivers answer rides as a first order issue and leave post-match cancellations for future research.

2.2 Data Summary

In our data, we observe all the requests sent by the passenger during the sample period. Each record of the request consists of the passenger-specified trip origin, destination, pickup time, the time the request was created, the time the request was answered or canceled. We only use requests where agents ask to leave within 40 minutes, which accounts for 80% of the total number of requests. In the subsequent analysis, we do not use pickup time when comparing two requests, because the time dimension seems to be a "soft" constraint: passengers often change departure time after some waiting. As shown in Figure 5, the number of waiting requests per minute, averaged across 20 days, is relatively stable during the sampled hour. The pickup and dropoff locations are across the city. We show a map of the pickup locations in Figure 1. We re-scaled the coordinates to avoid revealing the identity of the city. Our

[3]The matched driver would have to use a different phone to see the requests in this case.

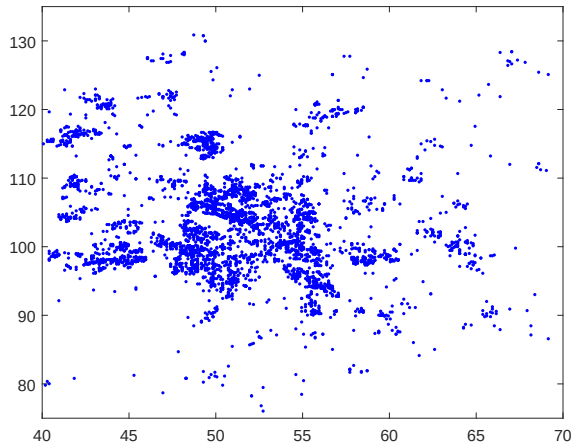


Figure 1: Pickup Locations

Each dot represents a pickup location. To avoid revealing the identify of the city, we rescaled the coordinates and do not show the units on the axis deliberately. The more popular pickup locations are both population and business centers and are also the more popular dropoff locations.

data show that the more popular pickup locations are both population and business centers and also more popular dropoff locations. Finally, we ensured that the weather was similar throughout the sample period (mild rain) and there was no major construction project in the chosen city.

	Mean	Std	# Observation
Time Till Cancellation (Min)	6.9810	7.8223	2738
Time Till Answer (Min)	5.0965	6.5533	2828
Percentage Answered	0.5081	-	5566
Trip Length (KM)	12.9181	9.0106	5566
# Waiting Requests/min	17.1156	9.5931	1200

Table 1: Summary Statistics

Table 1 reports the summary statistics. The table shows that these passengers wait a significant amount of time for a ride on DiDi Hitch. Conditional on a passenger's request not being answered, the passenger cancels the request at about the 7th minute. Conditional on a request being answered, the request is answered at the 5th minute. In comparison, getting a ride on DiDi Express, Uber or Lyft requires a passenger to wait no more than a

few seconds. Furthermore, only half of all requests are answered. We call the percentage of answered requests the "match rate". In comparison, the match rate on DiDi Express in most Chinese cities was above 99% during our sample period. This comparatively low match rate on the decentralized market is not uncommon: the occupancy rate on Airbnb ranges from 50% to 60% in most major American cities (Andreevska (2016)). According to conversations with DiDi, many passengers use the Hitch platform as a first choice, and if the requests are not answered by certain time, the passengers can always count on getting a DiDi Express car, a taxi or taking the public transit. We can compute a back-of-the-envelope number for the benefit of waiting: by waiting 5 minutes, with probability 0.5, a passenger going on a 13-KM trip could save

$$\underbrace{\$1.5 + \$0.3 \times 10}_{taxi} - \underbrace{(\$0.8 + \$0.15 \times 11)}_{Hitch} = \$2.05,$$

or 46% of the taxi fare. We thus think there is substantial benefit to waiting. It is harder to quantify the cost of passenger waiting. The waiting cost likely is psychological, because passengers do not actively choose drivers and waiting does not prevent the passenger from engaging in otherwise productive activities. Drivers can choose to frequently check available passengers, but can also have the system send messages to alert the presence of a nearby passenger request. We also learned from DiDi that drivers typically wait longer than passengers. This assertion is based on their personal experiences and interactions with DiDi Hitch drivers. Overall, we think waiting per se does not impose substantial cost, and agents leave the market mostly because they need to head to their destinations by a certain time.

We use the city-block distance measure[4] to calculate the length of trips, and on average these trips are about 13 KM in length. In a given minute, there are on average about 17 waiting requests.

To aid further analysis, we group requests by how similar they are. Each request is

[4]For a trip from (x_1, y_1) to (x_2, y_2), the distance is $|x_1 - x_2| + |y_1 - y_2|$.

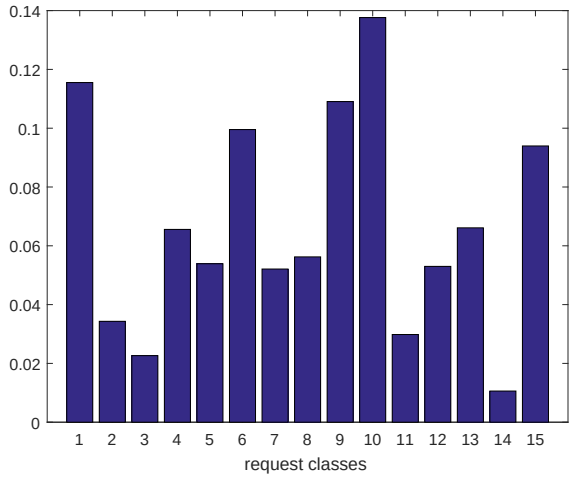


Figure 2: Frequency of Requests By Classes, Given by a K-means Algorithm

associated with its pickup and dropoff locations, and we use a K-means algorithm to cluster requests based on the difference between two routes. We define the distance ℓ_{ij} between route i going from $a_i = (a_i^x, a_i^y)$ to $b_i = (b_i^x, b_i^y)$ and route j using a city-block distance measure:

$$\ell_{ij} = \left|a_i^x - a_j^x\right| + \left|a_i^y - a_j^y\right| + \left|b_i^x - b_j^x\right| + \left|b_i^y - b_j^y\right|. \tag{1}$$

We cluster requests into 15 classes. Figure 2 presents the frequency of each class. Figure 6 shows the average match rates (percentage of requests answered) across classes. Figure 3 presents the average and standard deviation of trip lengths. The most popular routes are largely around the population centers of the city. We plot the origins and destinations of the most popular class, Class 10, in Figure 4.

We next discuss the properties of the empirical passenger arrival and exit process. The passenger arrival process can be approximated by a constant hazard model reasonably accurately. Figure 7 plots the model-predicted and actual frequency of the arrival time intervals of the two most popular classes. We also show that the classes of new passengers are at most weakly correlated across time. We test whether the classes of two consecutive requests are

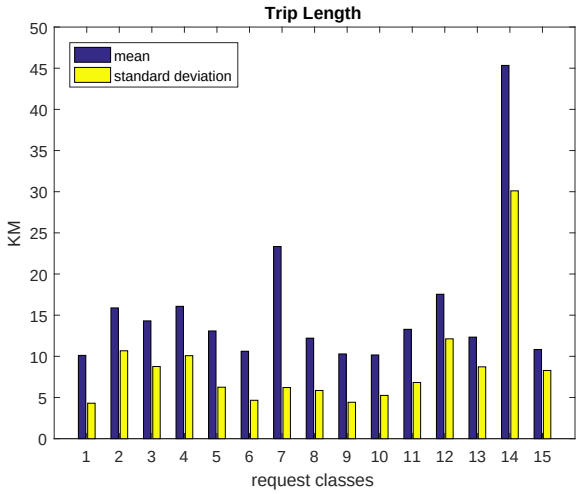


Figure 3: Trip Lengths by Classes

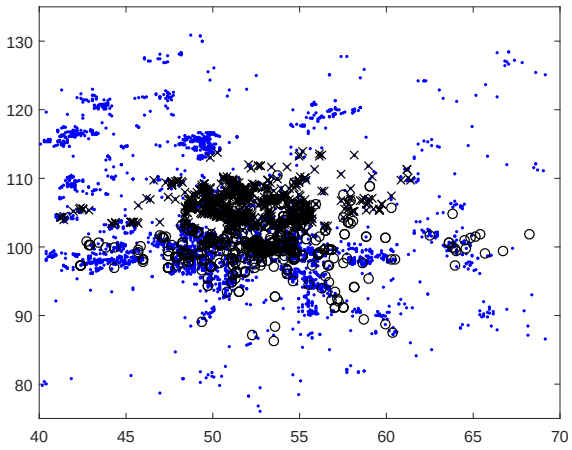


Figure 4: Pickup and Drop-off Locations of the Class 10 Requests

The blue dots represents the pickup locations of all classes across the city. Class 10 is the most frequently requested class of requests. The black crosses represent the pickup locations of the Class 10 requests. The black circles represent the dropoff locations of the Class 10 requests.

correlated using a χ^2 test. The p-value of the test is 0.7855, and we cannot reject the null that consecutive classes are independent. A similar exercise can also test whether driver and passenger arrivals are independent. The platform can classify the driver who opens her app immediately after a passenger request is sent and test if passenger classes are independent of the driver classes. Ideally, we would like to use data on what requests drivers see on the phone and how much drivers wait before answering a request. The data were not available during our sample period. In fact, one of the goals of the paper is to demonstrate that the driver app usage is important to understand the workings of the market and shed light on how to improve efficiency.[5] An internal test conducted by the platform using the more recent data shows that the p-value is 0.4704.

How quickly passengers exit the market is not directly observed, because drivers answer some of the requests and the passengers who voluntarily exit are a selected sample. On this select sample, we show that how long a request stays in the market are only weakly correlated with match rates. The left panel of Figure 8 shows the average time a passenger cancels a request (conditional on the request never being answered). The right panel of Figure 8 shows the average time a request stays in the market until the request is answered or canceled. Figure 8 shows that there is substantial variation in how long a request is in the market. The mean time also does not strongly correspond with the match rates in Figure 6. Figure 9 presents the frequency of the exit time for the two most popular classes.

We also find that repeated interactions between drivers and passengers are quite rare. We have reliable ID trackers of drivers and passengers for about 80% of the sample. 81% of passengers use the platform only once and 11.3% twice during our sample period. On the driver side, 76% of drivers have successfully completed a ride once, and 15% twice. Of all the answered requests on the sample with ID trackers, only 10% are repeated between the same driver and passenger. Conditional on a driver completing two or more rides during our sample, few carry the same passenger. Figure 10 shows that for drivers who have completed

[5]The platform started collecting these data later, but they were restricted for internal use only.

two requests, only 10% carry the same passenger. At same time, the number of rides on the platform is stable across the sampled days. After reading the feedback of many Hitch platform users, we think that the passenger base mainly consists of people who are not in a hurry and have alternative means of transportation available. The passengers are aware that there is a good chance they may not get a driver going in the same direction, but the low fares make "trying their luck" on the platform worthwhile.

Finally, we provide suggestive evidence that drivers are selective with requests. We document that giving drivers the right to choose passengers generates different patterns from what one would expect in the case of street-hailing taxis. In the latter case, although drivers may have preferences for routes, they in general do not know the preferences of the passengers until after the trip is initiated. The taxi drivers therefore pickup the first passenger they meet, regardless of the passenger request. Specifically, we consider the behavioral implication of two different types of behavior:

1. a driver randomly answers a request;

2. a driver only answers a request sufficiently close to her preference.

The two types of behavior are distinguishale under the following assumptions:

1. *new* passenger requests are not correlated across time

2. *new* driver preferences are not correlated with new passenger classes across time

In the above, we have shown that there is no strong evidence against either assumption. Use t to denote the time a driver answers a request. Under these assumptions, we should see no statistically meaningful difference in the numbers of requests that are "similar" to the driver preference in a period before t compared with a randomly selected period in scenario 1. In scenario 2, sufficiently many requests must exist for the driver to have a high quality partner before the request is answered. Specifically, for a request answered at time t, we show that

there are more requests "similar" to the answered one appear within the interval $t-10$ and t than in a random 10-minute interval.[6]

We say routes i and j are similar if $\ell_{ij} \leq 10$KM. We use the route of the answered request as a proxy for driver preference. In Table 2, we report the estimated probabilities of 0, 1, 2 or 3 requests that are (1) similar to the answered route, (2) *created* in the 10-minute intervals before the answered request was answered and (3) created in a random 10-minute interval. The estimates strongly suggest that drivers are selective with requests: before a request is answered, there are a lot more requests similar to the answered route *created* than in a random 10-minute interval. We postpone the discussion of driver waiting until Section 5.1.

# New Requests	Before	Random
1	35.25%	22.88%
2	21.22%	16.27%
3	13.93%	8.73%
4	10.33%	6.44%

Table 2: Probability of Similar New Requests in A 10-Minute Interval

"Before" column: the percentages are estimated probabilities of 0, 1, 2 or 3 requests that are (1) similar to the answered route ($\ell_{ij} \leq 10$) and (2) created in the 10-minute intervals before the time the answered request was answered. "Random" column: for each answered request, we sample a random 10 minute interval from data and calculate the number of similar requests to that request. We then calculate the probability across all answered requests.

[6]To see this effect, consider the following simple scenario with a Bernoulli random variable X ($\Pr(X=0)=p$) and an independent positive integer-valued random variable N that takes on the value of $0,1,2$ with probability $1-p_1-p_2$, p_1 and p_2. Suppose a driver waits for a realization of X equal to 0, and N realizations occur per minute. Then the conditional expectation of the number of realizations given the driver obtains $X=0$ after waiting for 1 minute is

$$\begin{aligned} E(N \mid \text{answered}) &= \frac{p_1 p}{p_1 p + p_2\left(1-(1-p)^2\right)} + \frac{2p_2\left(1-(1-p)^2\right)}{p_1 p + p_2\left(1-(1-p)^2\right)} \\ &= 1 + \frac{p_2(2-p)}{p_1 + p_2(2-p)} \end{aligned}$$

The previous line is minimized when $p \to 1$, and thus

$$\begin{aligned} E(N \mid \text{answered}) &\geq 1 + \frac{p_2}{p_1+p_2} \\ &\geq 1 + p_2 \\ &\geq p_1 + 2p_2 = E(N). \end{aligned}$$

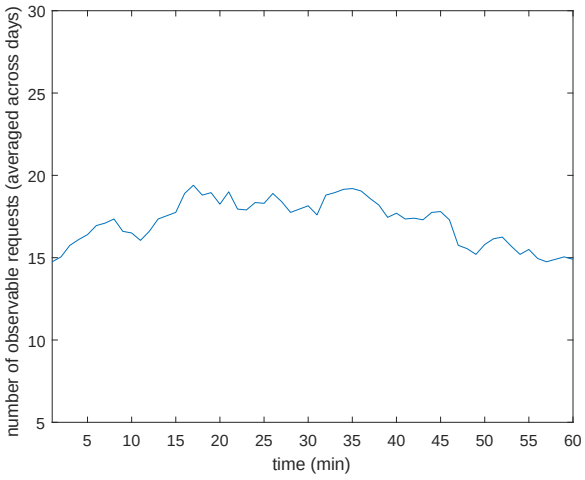


Figure 5: Average number of requests per minute

The time period is 7:30AM to 8:30AM. We average the number of requests across 20 days. A request is counted in a minute t if it is created in or before t and answered or canceled in or after t.

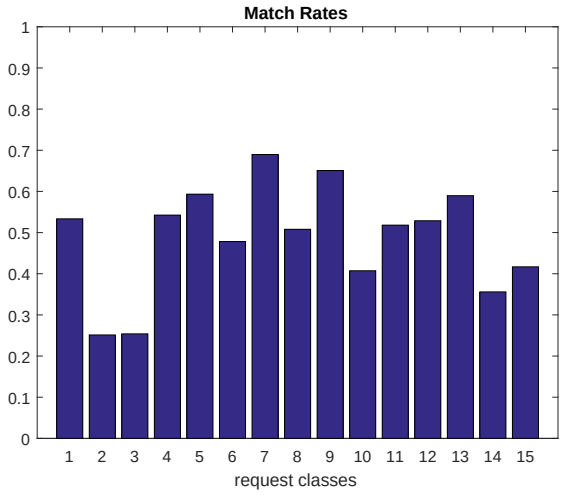


Figure 6: Match Rates by Classes

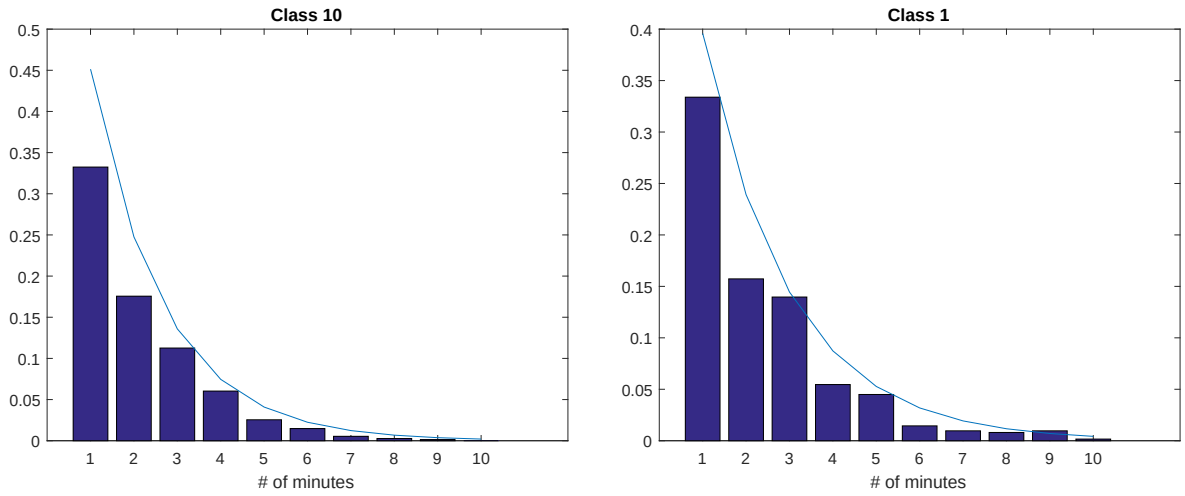


Figure 7: Arrival Frequency of Class 10 and Class 1 Requests

The bar graph represents the empirical frequency of the time between two arrivals within the same class. The solid line represents the predicted probability of arriving between the tth and $t+1$th minute from the estimated exponential distribution.

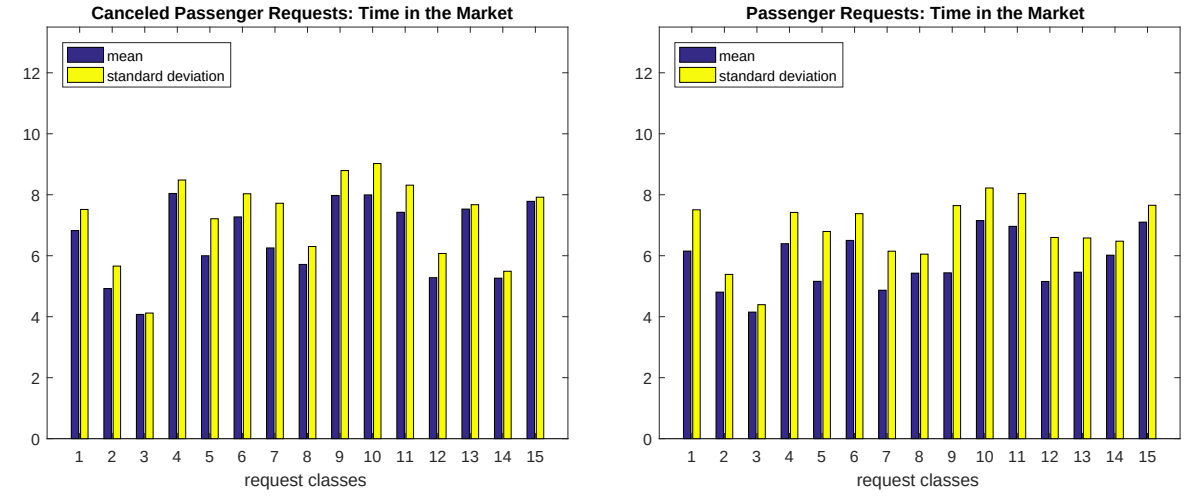


Figure 8: Left: Time Till Cancellation; Right: Time Till Being Canceled or Being Answered

The left panel estimates the mean and standard deviation of how long a request is in the market until being canceled for each class, on the select sample of canceled requests. The right panel estimates the same quantities but on all requests. The blue bars represent the mean and the yellow bar represents the standard deviation. The unit is in minutes.

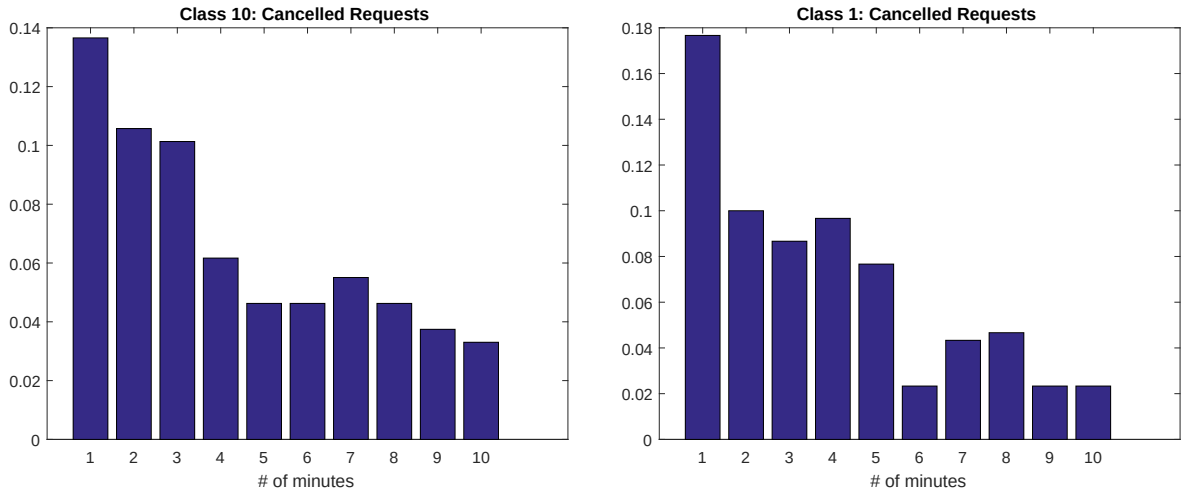


Figure 9: Time Till Cancellation; Class 10 and Class 1 Requests

The bar graph represents the empirical frequency of the time till cancellation for the two classes, estimated on the sample of canceled requests.

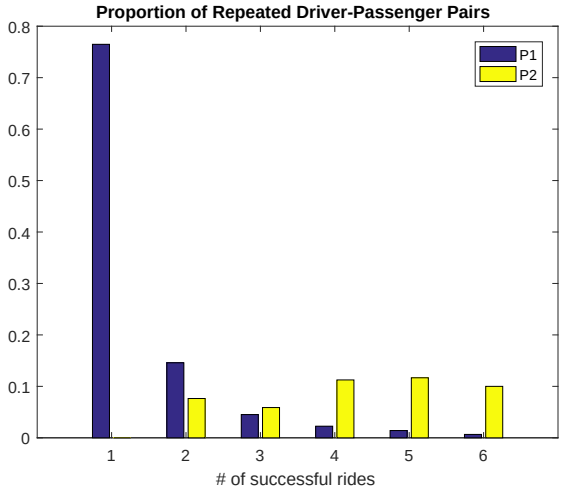


Figure 10: Percentage of Drivers Carrying the Same Passenger More Than Once

The blue bars (P1) represent the proportion of drivers who have completed a certain number of rides. The yellow bar (P2) represents the conditional probability that, given a driver having completed k rides, at least two of the k rides are with the same passenger. For example, about 14% of drivers completed two trips, and 10% of these drivers, or 1.4% of all drivers during the sample, carry the same passengers on these two trips.

3 A Simple Model of Driver and Passenger Match

We consider a simple model of match formation between drivers and passengers in an environment similar to our empirical application. We use this model to show the intuition that underlies our more complicated structural model and why the efficiency gain from a centralized algorithm is an empirical question. We note that the driver strategic waiting can thicken the market (compared with a market of myopic drivers), thus allowing more passengers and drivers to meet each other and creating more and higher quality matches. However, a driver does not consider the benefit of a match for the passenger and thus can wait too much, forgoing matches with passengers that would be formed in a planner's solution. A driver also does not consider the effect of waiting on other drivers, and competition causes drivers to wait too little.

Consider a continuous time, infinite horizon matching market of four potential entrants: two drivers (A, B) and two passengers (a, b). Drivers arrive at rate ρ^0 and passengers arrive at κ^0. At any moment, a driver in the market receive opportunities to move (that arrive independently at rate $\gamma > 0$) and choose whether to wait or pick up a passenger in the market. Only a driver can actively pick up a passenger and form a match. Upon the formation of a match with passenger i, the driver j receives utility u_{ij}, and the matched driver and passenger leave the market. An unmatched driver or passenger randomly exit the market at rate ρ and κ. If a driver leaves the market without a match, she receives a reservation utility 0. The entry and (unmatched) exit processes are independent across agents and over time.

While the passengers are not "strategic" in the sense that they do not actively choose partners or how much they wait, the model captures a key feature of a matching market: one driver's action limits the choice of another driver by changing the availability of the matching partners. The simplification allows us to characterize the key economic trade-offs. This bare bone model also represents the main features of our empirical application.

To solve for driver strategies, we assume that drivers have full knowledge of the past

history once they enter the market. An agent (a driver or a passenger) has three possible states: the agent has not entered the market, the agent is in the market or the agent has exited (either by a match or by a random exit). The state space of a driver problem consists of the Cartesian product of other agents' states and thus has $3^3 = 27$ elements. Use S_{jt} to denote the state of a driver at t. Given an opportunity to move, the driver observes the set of available passengers R_t and the utility of matching with the most compatible passenger $s_{jt}(R_t) = \max_{i \in R_t} u_{ij}$. Driver j decides between matching with $i^\star = \arg\max_{i \in R_t} u_{ij}$ or waiting. Without loss of generality, we consider A's problem. The value of waiting $V(S_{At}) \geq 0$ depends on S_{At}. A forward-looking A compares $\max_{i \in R_t} u_{iA}$ against $V(S_t)$ and a myopic A compares $\max_{i \in R_t} u_{iA}$ against the reservation value 0, and given preference ordering, a forward-looking agent will pass over some match opportunities to wait for a better partner. In Appendix A, we present the parameterization of two scenarios where strategic waiting has different implications for the number of matches in equilibrium and solve for a pure strategy Bayesian equilibrium explicitly. Below, we discuss the intuition of the two cases. The ex post realized number of matches depends on the order in which agents enter and leave the market. We focus the discussion on the scenarios where the market of forward-looking agents generates different match outcomes compared with a market of myopic agents.

1. Preference ordering:

$$A : a \succ b \succ \text{unmatched}$$

$$B : b \succ \text{unmatched} \succ a$$

When ρ and κ are sufficiently small, forward-looking A would always wait for a and B always for b. In this case, both drivers will wait for their most compatible partners and all drivers and passengers will be matched. In contrast, if b and A arrive at the market first, a myopic A would match with b, leaving the late arrivals (B, a) unmatched. In this case, strategic waiting increases the expected number of matches by increasing the market thickness so that A will not leave until meeting a. In fact, the strategic waiting

incentives allow the decentralized market equilibrium to achieve the social optimum.

2. Preference ordering:

$$A : a \succ b \succ \text{unmatched}$$

$$B : a \succ \text{unmatched} \succ b$$

When ρ and κ are small, and A sufficiently prefers a to b, A would wait for a when the market contains (A, b). In Appendix A, we provide numerical examples for how much A needs to prefer a to b in order for this assertion to be true. If a arrives before B, or when a, A and B are in the market but A gets to move before B, A will match with a, which leads to B and b unmatched. In contrast, when the market consists of only (A, b), a myopic A would match b and both would leave the market, and B can always match with a when they enter the market. If agents appear in different orders (and the event of exits before all agents have entered is close to 0), the same number of agents will be matched with either forward-looking or myopic agents. In this case, strategic waiting decreases the expected number of matches. The expected number of matches is lower than the social optimum regardless of whether the agents are myopic or forward-looking.

We will assume that the drivers in the empirical model introduced next are forward-looking. This assumption is useful for two reasons. First, our data and the platform's internal data both suggest that drivers stay in the market and look for passengers multiple times before they leave. This assumption is consistent with the behavior of rational drivers. Secondly, our main goal is to investigate whether centralized algorithms can improve efficiency even if drivers have rational beliefs. If drivers are myopic, a sophisticated platform can always strategically exploit how much an agent will wait to increase market thickness and create socially optimum matches. In contrast, implementing a centralized algorithm may be less worthwhile if most drivers have preferences like case 1. In this sense, our analysis below provides a framework to examine the effectiveness of alternative market designs in the "worst-

case" scenario, and the assumption of forward-looking agents is an integral part of the research design.

4 Empirical Model

We use an infinite horizon continuous time dynamic model to study the matching between passengers and drivers. Each agent (driver or passenger) has an ideal route (type), meaning that an agent of type i wants to travel from location a_i to location b_i. We discuss how this preference is specified in the model below. Drivers and passengers arrive to the market at hazard rates ρ^0 and κ^0. Use F_I and F_J to denote the type distribution of the drivers and passengers. An agent's type is drawn independently from the respective distribution and persistent throughout her stay in the market.

Drivers are forward-looking. A driver faces two "competing risks", one from a stochastic risk of leaving the market without a match, which occurs at rate κ, and one from a stochastic move opportunity with a hazard rate of γ. If the departure risk is realized first, then the driver i leaves the market without a match and receives utility $\underline{u}_i$. Alternatively, conditional on receiving a move opportunity first, a driver in the market views the available requests and chooses between answering a request or waiting. Use R_t to denote the set of available requests at instant t. If a driver chooses to answer a request j, the driver and the chosen passenger leave the market and the driver receives utility

$$u_{ij} = u_{i0} - \ell_{ij}, \tag{2}$$

where u_{i0} is the level of the driver utility if there is no mismatch and ℓ_{ij} is a scalar measure of the mismatch between routes i and j defined in (1). The trip fare is strictly based on distance, not dynamically calculated and does not include idle time in congestion. Depending on the assumptions on how drivers make the traveling decisions, the fares may or may not have a functionally different effect from the mismatch. Appendix B.1 discusses a few cases.

Here we use the simplest functional form to capture the first-order issue and do not include fares separately.

If the driver chooses to wait, she may leave the market without a match or receives another move opportunity, depending on whether the departure risk or the move opportunity arrives first. The arrival, departure and move processes are independent of each other and across agents and time. The driver decision maximizes her expected total payoff before leaving the market. Conditional on answering a request, the driver then chooses the request from the set of all available requests R to minimizes the mismatch. Let

$$s_i = \min_{j \in R} \ell_{ij}.$$

Then s_i is the only payoff-relevant variable at the instant the driver moves. To define our Markov Perfect Equilibrium, we also assume that the driver only uses s_i to make request-answering and waiting decisions.[7] Lastly, for the simplicity of the exposition, we assume that $\ell_{ij} \neq \ell_{ij'}$ for any $j, j' \in R$ and for any R.

To describe the transition of s_i and the precise decision rule, we introduce some more notation. Use $p_{ij}(R)$ to denote whether driver i answers request j when the set of requests is R. Passenger and driver types are drawn from the set I and J, which are Cartesian products of four intervals for each coordinate of the pickup and dropoff locations. Use D to the denote the set of drivers at an instant. Use $\ell_i^{(2)}(R)$ to denote the second lowest ℓ_{ij} for $j \in R$. Use $F^i_{DR|s}(D, R \mid s_i)$ to denote the joint distribution of the set of drivers and passengers conditional on i's state being s_i. Use $j_i^\star(R)$ to denote the request type such that

[7] See the discussion in Section 7.1 on using additional state variables to construct the strategy.

$\ell_{ij_i^\star} = s_i$. The Bellman equation for driver i in an infinitesimal amount of time Δt is

$$
\begin{aligned}
V_i(s_i) = {} & \underbrace{\Delta\kappa E_{D,R}^{F_{DR|s}^i}\left(V\left(\ell_i^{(2)}(R)\right)\middle|\, s_i\right)}_{\text{best trip cancelled}} + \underbrace{\Delta E_j^{F_J}\kappa_j^0\left(\ell_{ij} < s_i\right) V_i\left(\ell_{ij}\right)}_{\text{new passengers}} \\
& + \underbrace{E_{D,R}^{F_{DR|s}^i}\left(\sum_{i'\in D\setminus i} \Delta\gamma \cdot p_{i'j_i^\star(R)}(R) \cdot V\left(\ell_i^{(2)}(R)\right)\middle|\, s_i\right)}_{\text{some other driver answers the request } j^\star} + \underbrace{\Delta\gamma E_{\varepsilon_{it}}^{F_{\varepsilon_{it}}} \max\left\{u_{i0} - s_i + \varepsilon_{it}, V(s_i)\right\}}_{i\text{'s decision at her turn}} + \underbrace{\Delta\rho\underline{u}_i}_{\text{driver exit}} \\
& + \underbrace{\left(1 - \Delta\kappa E_{D,R}^{F_{DR|s}^i} - \Delta E_j^{F_J}\kappa_j^0\left(\ell_{ij} < s_i\right) - \Delta E_{D,R}^{F_{DR|s}^i}\left(\sum_{i'\in D\setminus i} \Delta\gamma \cdot p_{i'j_i^\star(R)}(R)\middle|\, s_i\right) - \Delta\rho\right) V_i(s_i)}_{\text{nothing happens}}.
\end{aligned}
\tag{3}
$$

The superscript on E denotes the distributions involved in the expectation, and the subscript denotes the corresponding random variables. ε_{it} is a random shock specific to driver i and the move opportunity at t. Define

$$
\begin{aligned}
\delta_i(s_i) = {} & \kappa E_{D,R}^{F_{DR|s}^i} + E_j^{F_J}\kappa_j^0\left(\ell_{ij} < s_i\right) \\
& + E_{D,R}^{F_{DR|s}^i}\left(\sum_{i'\in D\setminus i} \Delta\gamma \cdot p_{i'j_i^\star(R)}(R)\middle|\, s_i\right),
\end{aligned}
\tag{4}
$$

and

$$
\begin{aligned}
G_i(\ell \mid s_i) = {} & \frac{1}{\delta_i(s_i)} E_{D,R}^{F_{DR|s}^i} E_j^{F_J}\left[\kappa\left(\ell_i^{(2)}(R) \le \ell\right) + \kappa_j^0\left(\ell_{ij} < \min\{s_i, \ell\}\right)\right. \\
& \left. + \sum_{i'\in D\setminus i} \Delta\gamma \cdot p_{i'j_i^\star(R)}(R) \cdot \left(\ell_i^{(2)}(R) \le \ell\right)\right].
\end{aligned}
\tag{5}
$$

G_i is a well-defined conditional distribution function. The Bellman equation for the driver

can then be simplified as

$$V_i(s_i) = \frac{1}{\rho + \gamma + \delta_i(s_i)} \times \left\{ \gamma E_{\varepsilon_{it}}^{F_{\varepsilon_{it}}} \max\{u_{i0} - s_i + \varepsilon_{it}, V_i(s_i)\} + \rho \underline{u}_i + \delta_i(s_i) E_{s'}^{G_i}(V_i(s') \mid s_i) \right\}. \quad (6)$$

δ_i and G_i are the "reduced-form" hazard that the state variable changes and the distribution of the new state variable conditional on the state variable change and the current state. The driver knows δ_i and G_i but cannot observe D by the design of the platform or use non-payoff relevant information in R by our assumption. We relax this assumption in Section 7.1.

A passenger is assumed to reveal her true preference to the platform as soon as she arrives. A passenger faces two "competing risks", one also from a stochastic risk of leaving the market without a match, which occurs at rate κ, and one from drivers who might answer her request.

We next define our Markov Perfect Equilibrium.

Definition 1. $\{V_i, \delta_i, G_i, p_{ij}(R)\}_{i \in I, j \in J}$ consists of a stationary Markov Perfect Equilibrium if

- The Bellman equation (6) is satisfied for all $i \in I$;
- $\forall i$, (D, R) has a stationary distribution $F_{DR|s}^i$ consistent with $\{\delta_i, G_i, p_{ij}(R)\}_{i \in I, j \in J}$ and $\{F_I, F_J, \rho^0, \kappa^0, \kappa, \rho\}$.

In Appendix B.2, we provide conditions for the existence of the equilibrium.

In the empirical application, we estimate a model setting $\varepsilon_{it} = 0$. ε_{it} with an absolutely continuous distribution provides smoothness that helps to establish equilibrium existence. However, such errors are particularly hard to interpret in our context. By omitting this term, we thus avoid interpreting the meaning of this shock and later adding up i.i.d errors in the welfare calculation (an issue sometimes also faulted in the logit-based demand models)

or when we simulate the centralized matching algorithms. The trade off is that we lose the theoretical guarantee that the equilibrium exists.[8] We verify that across a large set of parameters, we can always numerically find an equilibrium. Ronald Gallant, Hong and Khwaja (2017) estimate a dynamic game of complete information and similarly rely on direct computation to find equilibria.

We do not explicitly model waiting or search costs for two reasons. First, our model is suited for the behavior of commuters who need to go to their destinations by certain deadlines. Waiting does not prevent drivers from doing other things and search requires minimal efforts: the app can alert the driver if a request nearby shows up. A passenger can also cancel her request with no penalty before it is answered. In contrast, professional drivers likely face a substantial opportunity cost of waiting because it would otherwise be easy for them to receive an order through the central dispatch during the morning rush hours and go on a different trip immediately. In addition, many online forums and blog posts typically advise drivers and passengers to use the service only if their trips are not urgent. Secondly, even when driver choices are observed in data, driver waiting cost is not separately identified from other primitives, as shown in Appendix B.3. In data, we only observe driver choices when a driver answers a request.

5 Identification and Estimation

In this section, we discuss the identification and estimation of the unknown parameters:

1. (F_I, F_J) the distribution of the driver and passenger types upon entry.
2. the driver preference parameters $(u_{i0}, \underline{u}_i)$,
3. (ρ^0, ρ, γ) the rates at which (1) a driver arrives to the market, (2) exits without a match and (3) a driver moves,

[8] In a setting of discrete and finite time, Doval (2018) showed that matching outcomes that satisfy a notion of dynamic stability may not exist for some agent preferences in two-sided markets.

4. (κ^0, κ) the rates at which (1) a passenger arrives to the market and (2) exits without a match.

We use the observed passenger arrival rates for κ^0. We use the observed passenger request distribution as F_J. In data, there are 4642 unique passenger types (routes). We further use a k-means algorithm and divide these routes into 15 classes by how closes the routes are to each other using the measure defined in (1), as discussed in the data section. We use the empirical frequency of each class as the probability of drawing a type from the class. The estimates of the empirical frequency is presented in Figure 2. To identify the distribution of driver types, we need the support of trip requests at least as large as that of driver types:

Assumption 1. $I = J$.

In practice, we set I and J to be the set of all observed routes in the data. Use $w_I(i)$ and $w_J(j)$ to denote the probability weight of each type of drivers and passengers. We assume that within a class, types are distributed uniformly.

Assumption 2. $w_I(j) = w_I(j')$ *and* $w_J(j) = w_J(j')$ *if* j *and* j' *belong in the same class.*

We next assume that driver and passenger preference distributions are the same conditional on entry:

Assumption 3. $F_I = F_J$.

The distribution of the driver types is determined by F_I and the driver entry and exit rates. We fix F_I in order to estimate the rates. We discuss how this assumption can be relaxed at the end of Section 5.1. We stress that the parsimonious representation of driver distribution based on $(u_{i0}, \underline{u}_i, \rho^0, \rho)$ and Assumption 3 captures the main trade-off represented in Section 3: the externality (driver competition) increases in the entry rates ρ^0 and the tolerance of mismatch $u_{i0} - \underline{u}_i$.

5.1 Identification

First, u_{i0} and $\underline{u}_i$ are not separately identified just like in most discrete choice models. We identify the difference $u_{i0} - \underline{u}_i \equiv u_0$. We also report the result where we specify the difference as $u_0 + \beta \ell_i$, where ℓ_i is the city block distance of the trip. Intuitively, u_0 is identified by the spatial variation in the percentage of matched requests under assumptions 1. A lower u_0 means that more requested routes are more likely to be answered. A larger u_0 means the spatial variation of the answer rates would be low.

We next focus on identifying parameters crucial to the question of market thickness: the agent arrival and departure rates. Two features of the data help to pin down these two parameters on the driver side. First, our model implies that every driver will answer a request if there exist sufficiently many requests matching each driver type. Intuitively, the number of answered requests traces out the distribution of the number of arriving drivers when the number of requests is large and the types are finite (Assumption 1). This identification argument relies on the presence of such "extreme" variations. Figure 11 shows that the number of answered requests is indeed concave in the number of requests. We aggregate the number of answered requests and number of requests in 10-minute intervals. The plot shows the average numbers of answered requests if the number of requests is less than 15, between 15 and 25, between 25 and 35, between 35 and 45 and greater than 45. The plot shows that the average number of answered requests "flattens out" when the number of requests is in the last two categories. We view this concavity as evidence that there exist periods in data where the number of answered requests is close to the number of drivers.

Secondly, variations in the number of answered requests across time helps to identify the departure rate ρ. If ρ is high, meaning that drivers only exist for a short amount of time, then drivers who answer requests in minute t could only be drivers that arrive in t. We impose that the arrival of drivers are independent across time. Under this assumption, the number of requests and the number of answered requests in minute $t-1$ are uncorrelated with the number of answered requests in t. On the other hand, if drivers are long-lived, then a large

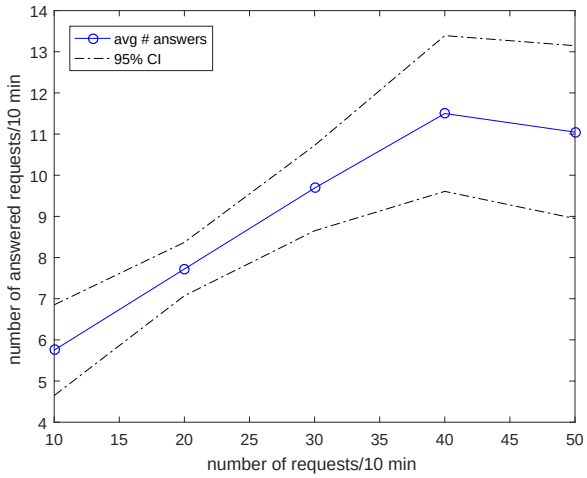


Figure 11: The number of answered requests concave in the number of requests

We aggregate the number of answered requests and the number of requests in 10-minute intervals. The plot shows the average numbers of answered requests if the number of requests is less than 15, between 15 and 25, between 25 and 35, between 35 and 45 and greater than 45. The plot shows that the average number of answered requests "flattens out" when the number of requests is in the last two categories.

number of requests in minute $t-1$ will reduce the stock of drivers and decrease the number of requests in t. The number of requests in $t-1$ may be high for another reason because of long-living drivers: the stock of drivers is low and many requests have gone unanswered. The latter effect also generates a negative relationship between the number of requests in $t-1$ and the number of answered requests in t. We show this relationship in Figure 12. We plot the number of answered requests in t against the number of requests in $t-10$ and $t-6$. To be precise, this negative relationship is not evidence that drivers are forward-looking, but that drivers wait. In the model, we impose that drivers make forward-looking, strategic waiting decisions consistent with how long they will be in the market. A market of myopic drivers would still have the problem of non-assortative matching as well as a sub-optimally low level of market thickness, and both inefficiencies can unambiguously be lifted by the centralized matching algorithms proposed later.

κ^0 and F_J can be directly identified from data. We observe when and what requests appear in the market. κ can conceptually be identified by a deconvolution of the observed exit rates and the model-implied rate at which drivers answer requests, which is given by the model and the parameters governing driver behavior. As discussed in Arcidiacono et al.

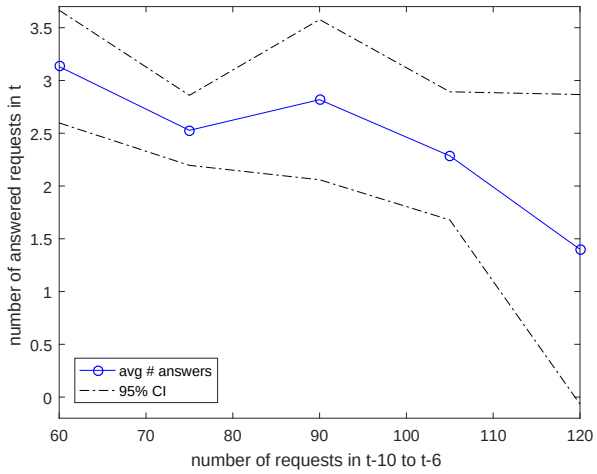


Figure 12: The number of answered requests decrease in the number of past requests

We plot the number of answered requests in t against the number of requests in $t-10$ and $t-6$. The evidence is consistent with a sufficiently small ρ.

(2016), γ usually cannot be identified. We set $\gamma = 10$, a sufficiently high number so that the inefficiency from drivers "randomly" missing available requests is negligible. This value of γ implies that drivers check for requests every 6 seconds on average.

To generalize Assumption 3 and incorporate additional heterogeneity, we can subset routes into K groups by, for example, geographical closeness of origins. We then assume that driver and and passenger type distributions are the same for each group $F_I^k = F_J^k$, but the entry rate and exit rates differ across groups. In theory, we can identify and estimate group-specific hazard rate parameters and u_0. Within-group variations of request and request answer rates across time and locations help to pin down these primitives following the identification arguments above. We also note that our data limit the ability to identify the heterogeneity of driver preferences over the mismatch separately from driver entry rates. Because we do not observe the actual number of waiting drivers, we can add to the market any number of drivers with $u_0 < 0$ and the observed outcome is equivalent. The heterogeneity of driver preference may not be identified even if we restrict $u_0 > 0$. The same answer rate may be explained by either many drivers with low u_0 or a few drivers with high u_0. In the context of dynamic auctions, Backus and Lewis (2016) shows that it is possible to identify the buyer preference conditional on participation when bids are observed and buyers

are observed in multiple auctions. The bids are inverted to identify buyer preferences. In our context, a day is a market. We have reliable ID trackers for about 80% of the drivers on answered requests. There are almost no within-day panel data: only 0.36% of drivers answer more than one request during the sampled hour within a day. It should be noted that we do observe about 30% of drivers answering requests on more than one day in our sample. In principle we could exploit the panel structure across days to identify driver heterogeneity by assuming that driver preferences are persistent not only throughout one hour in a day, but also across days. We find this assumption less attractive in our context. Our data show that 10% of these drivers answer requests that are 30KM apart. We think it is more plausible that drivers answer very different routes because of a change in preference instead of the "right tail" of the tolerance of the mismatch. We therefore do not exploit the panel structure. Section 7.3 discusses the estimation of the dispersion of mismatch tolerance under additional assumptions.

5.2 Estimation

Our empirical setting offers a unique challenge. Because we do not observe the drivers or their exit actions, we cannot directly estimate policy functions and apply the two-step method in Arcidiacono et al. (2016) (which extends the two-step method from Bajari, Benkard and Levin (2007) to the continuous time setting). Nor can we use the full solution method and form likelihood directly on agent action: the set of competing drivers is not directly observed in data. A likelihood function would require a high dimensional integral over all possible sets of waiting drivers.

To overcome these challenges and follow the identification strategy, we use a two-step simulated indirect inference method (Collard-Wexler (2013)). In the first step, for each type of drivers, we estimate the "reduced-form" hazard rate of change of the state δ_i and its conditional distribution G_i if the state changes. The estimates of δ_i and G_i in this step are consistent because the set of drivers and passengers and hence s_i have a stationary and

ergodic distribution. In the second step, we guess a vector of parameters, solve the Bellman equation (6) and obtain driver policy functions for each type, simulate the evolution of the market and construct moments from a set of auxiliary models. An outer optimization routine searches over parameter space to minimize the distance between the simulated and data moments. We describe the set of moments, how they relate to the identification argument and other estimation details in Appendix C. Finally, we only use trip requests that ask to leave in 40 minutes so that the requested time to leave are roughly comparable.

Table 4 reports the estimates. The standard errors are from a 100-reptition bootstrap on the days, preserving the correlations across time within a day. We estimate the model with and without a term controlling for the distance of the driver's ideal trip in the utility function. The estimates show that the driver arrival rates are lower than passenger arrival rates, but could wait substantially longer than the passenger. A surprising finding is that drivers have a high tolerance of the mismatch. The average length of the trips is about 13 KM, but a driver will be worse off than driving without a match only when the level of mismatch is more than 10.3 KM. The high tolerance of mismatch is likely due to the lump sum subsidy. A \$1 subsidy amounts to about 6.7 KM of fares on the Hitch platform.

To examine the model fit, we look at the most important metric to the platform operator, the match rate (percentage of answered requests).[9] The match rate is not part of our moments, because the match rate relies on passenger ID trackers and our moments use data aggregated to the minutes. Still, the model predicts an overall 51% match rate, which is very close to the observed rate of 50.8%. The left panel in Figure 13 reproduces the empirical match rates by each class of passenger requests from Figure 6, and the right panel presents the model predicted match rates. The model overall predicts a larger variation of match rates across types. The model over-predicts the answer rates for Class 10 by 50% but matches the second and third most popular classes reasonably well (7.5% and 9%). Class 2, 3 and 14

[9]In conversations with DiDi, the managers care much more about increasing the match rate than the current revenues. The strategy in part is based on the perception that higher match rates will encourage entry in the long run.

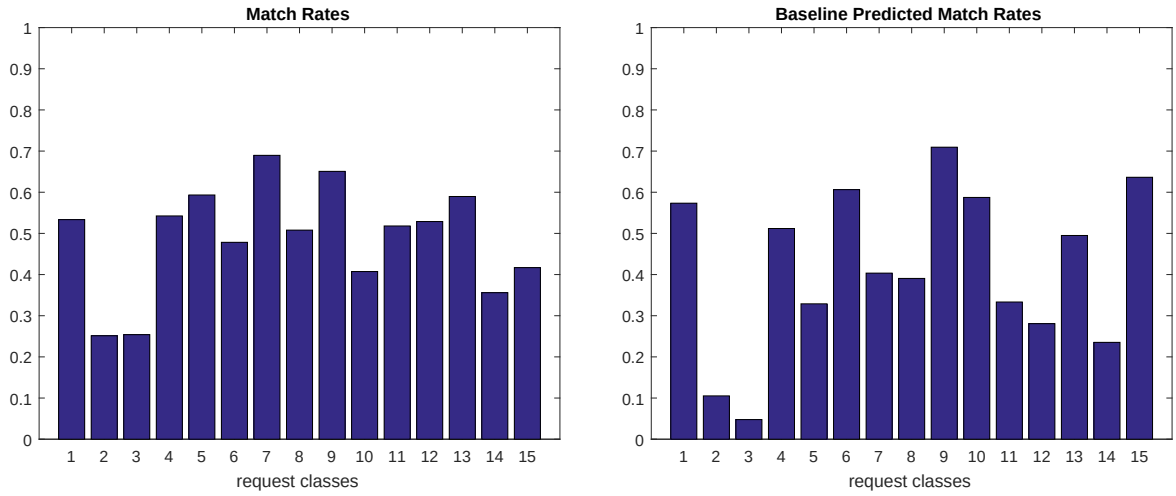


Figure 13: Left: Empirical Match Rates; Right: Model-Predicted Match Rates

Compatible Probability	Percent of Drivers
>10%	5.9133%
>5%,≤10%	33.9286%
>1%,≤5%	34.4555%
>0.1%,≤1%	19.9649%

Table 3: Heterogeneous Compatibility of Drivers

The left column represents the probability that a driver is compatible ($u_0 - \ell_{ij} > 0$) with a randomly drawn passenger i. The right column represents the percentage of such drivers.

have the lowest match rates both in data and the model.

Given the estimates, we can answer the following question: how heterogeneous is driver compatibility with passengers? The heterogeneity among drives comes from their different route preferences. A driver that travels between more popular destinations has more choices. Using the model estimates, we can simulate the probability a driver i is compatible with a randomly drawn passenger j, where compatibility is defined as $u_0 - \ell_{ij} > 0$. In Table 3, we present the distribution of driver compatibility. The results show substantial differences in compatibility. Given a random draw of a passenger, greater than 20% of drivers have less than 0.1% probability of meeting a compatible passenger, while about 6% of drivers have more than 10% probability of meeting a compatible passenger.

To estimate the alternative specification where $u_{i0} - \underline{u}_i$ is given by $u_0 + \beta\ell_i - \ell_{ij}$, we include

		Est	Se	Est	Se
κ^0	Passenger Entry	4.4193	0.2649	4.4193	0.2649
ρ^0	Driver Entry	3.742	0.116	3.7836	0.1136
κ	Passenger Exit	0.093	0.0028	0.0935	0.0025
ρ	Driver Exit	0.0158	0.003	0.0163	0.0023
u_0	Max Driver Utility	10.3219	0.3999	9.3587	0.3652
β	Distance Utility			0.0434	0.0605

Table 4: Parameter Estimates

The left panel estimates the model where the utility function for driver i from answering a request j is $u_0 - d_{ij}$, and the right panel's utility function is $u_0 + \beta\ell_i - \ell_{ij}$. The standard errors are from a 100-reptition bootstrap on the days, preserving the correlations across time within a day.

an additional moment: the average distance of answered requests. If β has a large positive effect, we should find that longer distance trips are more frequently answered. The estimates show that the actual distance of a driver's ideal route seems to have a small, statistically insignificant (and positive) effect on driver utility.

6 Counterfactual

We conduct five simulations to answer our research questions. We first use the estimates to solve the game and simulate the "factual" market evolution. To solve the game, we use simulations to approximate δ_i and G_i for each driver in each iteration and solve for the value function until convergence. We do not presume that the equilibrium is unique. To select an equilibrium, we start the value function iteration with the estimated evolution of the state variables. In the second counterfactual simulation, we assume that drivers still enter and leave the market without a match following the process given by the estimates, but only make static optimal decisions (or myopic decisions): when given the move opportunity, a driver i answers the request $j \in R$ if

$$u_{ij} = u_0 - \ell_{ij} > 0, j = \arg\min_{j' \in R} \ell_{ij'}.$$

In contrast, a dynamically optimizing i answers the request j if

$$u_{ij} = u_0 - \ell_{ij} - V_i(\ell_{ij}) > 0, j = \arg\min_{j' \in R} \ell_{ij'}.$$

The comparison between the first two simulations reveals the effect of strategic waiting.

We next examine three sets of counterfactual centralized algorithms. The platform is assumed to know the preference of the drivers. We first simulate the "greedy" algorithm in Akbarpour et al. (2017) adapted to the two-sided market and adjusted for sorting. The platform matches a new driver or a passenger subject to the driver's incentive constraint (the driver utility must be non-negative) immediately after the agent shows up , and the platform chooses the matching partner to maximize the driver utility. If an agent is not matched, she stays in the market until she leaves without a match or gets matched with another new agent. Between the decentralized myopic matching market and the greedy algorithm, the difference shows the effect of sorting: in every instance of matching, the matched pair in the greedy algorithm minimizes the mismatch conditional on the identity of the arriving agent. In the steady state, such a match also minimizes the mismatch across all possible matches, because there do not exist compatible matches among waiting agents. We next simulate the "patient" algorithm. In this case, the platform is assumed to also know when agents leave without a match. The platform matches an agent immediately before she leaves the market to an agent that maximizes the driver utility subject to the driver incentive constraint. A motivation for this algorithm in the context of the ride-sharing platform is that the platform may have the technology to predict the departure timing of the agents. Finally, we consider an assortative matching algorithm. The platform matches agents every 5 minutes, solving a linear programming problem that satisfies the driver incentive constraint and the one-to-one

capacity constraint:

$$\max_{x_{ij}} x_{ij} u_{ij}$$

$$s.t. 0 \leq x_{ij} \leq 1$$

$$0 \leq \sum_i x_{ij} \leq 1, 0 \leq \sum_j x_{ij} \leq 1$$

Roth and Sotomayor (1992) for example shows that this linear programming problem always has a unique solution, and the solution (x_{ij}) solves the agent allocation problem: x_{ij} is either 0 or 1, and i is matched to j if $x_{ij} = 1$. This algorithm allows the platform to take into account the externality among drivers given a set of passenger and drivers. In addition, this algorithm only requires the platform to know driver route preferences, and we hope that this algorithm might strike a balance between efficiency improvement and implementation feasibility.

We report the results in Table 5. Because we do not specify the utility of the passengers, we separately report the match rates (percentage of requests answered) as a proxy for passenger participation and utility. We also report the average number of waiting passengers and drivers at each minute mark in equilibrium. We summarize the key observations below. All percentage differences are statistically significant at the 95% level.

1. The comparison between the "Factual" and "Myopic" results show that removing strategic waiting can increase the match rate by 17% , but the driver utility decreases by 20%. Strategic waiting increases equilibrium numbers of waiting drivers and passengers. The results show that although this strategic incentive allows more drivers to meet more passengers and thus reduces the inefficiency from compatible drivers and passengers not arriving at the same time, it only helps the drivers.

2. The comparison between "Myopic" and "Greedy" shows that sorting substantially increases driver utility (22%) and also has a small positive impact on the match rate.

3. The comparison between "Greedy" and "Patient" shows that additional market thickness increases both the match rate (33%) and driver utility (15%). The improvement quantifies the gains from knowing when agents leave the market without a match.

4. Even if the platform does not have the information on when agents leave but knows the driver preference, the last column shows that it can still use an assortative matching algorithm to improve upon both the match rate (12%) and driver utility (4%).

To understand the positive effect on the number of matches by forcing agents to be myopic, we re-examine how the heterogeneity of driver preferences drives our results. An insight from the example in Section 3 is that the negative externality may be more prominent when more drivers prefer the same passenger. Our simulation suggests substantial competition: among the answered requests in the factual simulation, 20.35% of the requests are the most preferred (and compatible) requests for at least two waiting drivers when these requests are answered. As the driver entry rates decrease or exit rates increase, the competitive effect is diminished and the number of matches gained when drivers are forced to be myopic is smaller (the simulation results for this comparative statics are omitted for brevity).

The counterfactual result also illustrates the role of market thickness when drivers have heterogeneous preferences across matches. In all five simulations, the number of drivers is higher than the number of passengers. Had drivers been indifferent across matches, the match rate (for the passengers) should be 100% in each case. Interestingly, the "Patient" column has the lowest driver-passenger ratio, or fewest drivers per passenger, but a higher percentage of drivers and passengers are matched as a result of the thicker market. The result would be reversed if both sides have homogeneous preferences (as would be the case if the drivers only care about picking up a passenger but not the destination).

As a benchmark, we also consider an ex post optimal outcome where agents that appear at different time points can be matched. A motivation is a one-day-ahead market if agents know their commuting needs one day ahead, and a centralized algorithm makes the match after all agents submit their preferences. Because the passenger arrival rate is greater than

	Factual	Myopic	Greedy	Patient	Assortative
Match Rate	0.5117	0.6027	0.6527	0.6786	0.5731
	0.0105	0.0227	0.0299	0.0317	0.0254
Driver	2.6306	2.0913	2.5516	3.0365	2.7341
Utility	0.2643	0.1667	0.1732	0.2192	0.248
# Passenger	21.8671	17.9406	16.6455	42.0626	21.5435
	1.9083	2.0074	2.2776	3.6521	2.1024
# Driver	53.7988	43.1229	55.4194	79.0313	79.4418
	3.0358	3.3922	6.9483	8.6678	10.6705

Table 5: Counterfactual results: baseline

The average driver utility is calculated as the total driver utility divided by the number of drivers, including those who exit without a match. The third and fourth rows report the average numbers of waiting passengers and drivers per minute. The standard errors are from a 100-reptition bootstrap on the days, preserving the correlations across time within a day.

the driver arrival rate, this outcome could achieve a match rate of $\frac{\rho^0}{\kappa^0} = 84\%$ and driver utility of $u_0 = 10.3219$.

We next turn to several other aspects of the counterfactual results. The percentage of matched drivers in the decentralized market is about 60% and changes in the same way as the driver utility across simulations. The simulated average amount of time a passenger stays in the market is about 4 to 5 minutes for "Factual", "Myopic", "Greedy" and "Assortative", and is much higher (9.6 minutes) under the patient algorithm. Although we do not think agents incur significant waiting cost in our application, we caution that long wait time may be a downside of the patient algorithm in other contexts.

Including the control for the distance of a driver's ideal route in the tolerance of mismatches does not change the result. The result is reported in Table 6.

Our institutional contexts suggest that modeling passenger's waiting as an exogenous process may be a close proximity to the reality, but passenger and driver participation could still be endogenous. Because participation likely responds positively to increased match rates and driver utility (as would be the case in an entry model), our results likely understate the benefit of the centralized algorithm that increases both measures.

	Factual	Myopic	Greedy	Patient	Assortative
Match Rate	0.5117	0.6027	0.6527	0.6786	0.5731
	0.0105	0.0227	0.0299	0.0317	0.0254
Driver	2.6306	2.0913	2.5516	3.0365	2.7341
Utility	0.2643	0.1667	0.1732	0.2192	0.248
# Passenger	21.8671	17.9406	16.6455	42.0626	21.5435
	1.9083	2.0074	2.2776	3.6521	2.1024
# Driver	53.7988	43.1229	55.4194	79.0313	79.4418
	3.0358	3.3922	6.9483	8.6678	10.6705

Table 6: Counterfactual results: including the control for distance

The average driver utility is calculated as the total driver utility divided by the number of drivers, including those who exit without a match. The third and fourth rows report the average numbers of waiting passengers and drivers per minute. The standard errors are from a 100-reptition bootstrap on the days, preserving the correlations across time within a day.

7 Robustness

In addition to the alternative specification that includes the distance of driver routes in the utility function, we consider three additional sets of robustness checks in this section. First, we check whether the estimates and counterfactual predictions are robust if drivers use more complex forms of strategies. Secondly, we check whether allowing drivers to be myopic immediately before they leave the market without a match changes the result. Lastly, we examine the results when we add more heterogeneity in the driver tolerance of mismatches.

7.1 Driver Rationality

A key assumption in the analysis above is that drivers employ a Markovian strategy based on a single state variable. One may conjecture that a more sophisticated driver could improve her strategy by conditioning on other non-payoff relevant variables. In the extreme, a driver who makes decisions at the instant t can derive a conditional distribution of potential drivers based on the sequence of the sets of requests observed since the driver arrives to the market. We do not find this level of rationality realistic: many of these drivers use the service once or twice per month, and online forums for these drivers typically recommend a cutoff rule for when to answer a ride. As indirect evidence for our claim, among the trips that were

	Est*		Factual	Myopic	Greedy	Patient
ρ^0	4.1926	Match Rate	0.5301	0.6157	0.6403	0.6873
κ	0.0727	Drv. U	2.2902	1.8232	2.1329	2.687
ρ	0.0471	# Passenger	26.0966	21.1535	21.4741	42.6022
u_0	10.4342	# Driver	36.7146	30.3734	29.2146	51.3267

Table 7: Result: strategy based on two variables

*: All estimates are significant at the 99% level
**: The assortative matching algorithm matches agents at the 2-minute frequency.

answered in our data, 0.36% of the drivers answered more than one request during the same day, and 70% of the drivers answered only one request over 20 days. Learning within a day and learning across days both seem unlikely. It is possible that drivers register under multiple ids to earn a new driver award, or drivers and passengers may re-connect offline and circumvent the platform fee, but either case is considered fairly unlikely according to conversations with DiDi. In addition, drivers who pick up passengers without DiDi is not covered by DiDi's insurance and also considered illegally operating a taxi.

As a robustness check, we investigate the effect of a more sophisticated Markov strategy based on s_i and ℓ_{ij_2}, where j_2 is the trip request second closest to a driver's ideal route (second lowest mismatch). We report the estimates and counterfactual result in Table 7.

The result is qualitatively similar to the baseline. The patient algorithm increases the driver utility and match rate by about the same percentage. The estimate of a driver's no-match exit rate is higher than the baseline, so we simulate the assortative matching algorithm at the 2-minute frequency. This algorithm still improves upon the baseline match rate by 13%, but the driver utility is slightly lower (-3%). The match rate improvement is significant at the 95% level and the driver utility decrease is not.

7.2 Alternative Assumption on Driver Behavior Before Exit

One might be concerned that drivers are unfairly disadvantaged in the decentralized market by a modeling artifact, that before they exit, they are not allowed to choose passengers based on their reservation value instead of the value function. We show that this modeling feature

	Est*		Factual	Myopic	Greedy	Patient
ρ^0	4.0341	Match Rate	0.4938	0.5924	0.5898	0.6267
κ	0.0870	Drv. U	2.2233	1.7402	1.8913	2.2602
ρ	0.0400	# Passenger	25.9128	20.7823	20.5553	39.4619
u_0	9.5471	# Driver	57.3520	36.9818	36.4342	55.5629

Table 8: Alternative Driver Behavior Before Exit: Estimates and Counterfactual

*: All estimates are significant at the 99% level

implies little change in counterfactual predictions for the number of matches, but limits the gains of driver utility under the patient algorithm. In this robustness check, we allow the drivers one additional move opportunity immediately before they randomly exit when they are unmatched. The driver problem in Eq. (3) is modified as

$$
\begin{aligned}
V_i(s_i) = & \underbrace{\Delta\kappa E_{D,R}^{F^i_{DR|s}}\left(V\left(\ell_i^{(2)}(R)\right)\middle|\, s_i\right)}_{\text{best trip cancelled}} + \underbrace{\Delta E_j^{F_J}\kappa_j^0\left(\ell_{ij} < s_i\right) V_i\left(\ell_{ij}\right)}_{\text{new passengers}} \\
& + \underbrace{E_{D,R}^{F^i_{DR|s}}\left(\sum_{i'\in D\setminus i} \Delta\gamma \cdot p_{i'j_i^\star(R)}(R)\cdot V\left(\ell_i^{(2)}(R)\right)\middle|\, s_i\right)}_{\text{some other driver answers the request } j^\star} \\
& + \underbrace{\Delta\gamma E_{\varepsilon_{it}}^{F_{\varepsilon_{it}}} \max\left\{u_{i0} - s_i + \varepsilon_{it}, V(s_i)\right\}}_{i\text{'s decision at her turn}} + \underbrace{\Delta\rho \max\left\{u_{i0} - s_i + \varepsilon_{it}, \underline{u}_i\right\}}_{\text{driver exit}} \\
& + \underbrace{\left(1 - \Delta\kappa E_{D,R}^{F^i_{DR|s}} - \Delta E_j^{F_J}\kappa_j^0\left(\ell_{ij} < s_i\right) - \Delta E_{D,R}^{F^i_{DR|s}}\left(\sum_{i'\in D\setminus i} \Delta\gamma \cdot p_{i'j_i^\star(R)}(R)\middle|\, s_i\right) - \Delta\rho\right) V_i(s_i)}_{\text{nothing happens}}.
\end{aligned}
$$

Table 8 reports the estimates and counterfactual results. The estimates of driver entry and exit rates are higher than the baseline result. In counterfactual simulations, the number of matches (match rate) increases by over 20% under the patient algorithm, which is comparable to the baseline result. In other words, the additional matches gained by the patient algorithm indeed comes from preventing drivers from passing over passengers that should be matched in a near-social optimum. However, the driver utility only increases by 1% under the patient algorithm. The patient algorithm also mainly increase the thickness on the passenger side and has little effect on the driver side thickness.

7.3 Heterogeneity of Mismatch Tolerance

As discussed in Section 5.1, ideally we need a panel on driver search and match behavior to identify driver heteorgeneity in mismatch preferences. With passenger side data alone, we can nonetheless estimate a dispersion parameter on driver mismatch preferences using intuition from demand estimation and relying on a few restrictions to the model. Assume that the utility of driver i matching request j is

$$u_{0i} - \ell_{ij},$$

where $\ln u_{0i} \sim N(\mu, \sigma^2)$. We further assume that u_{0i} is distributed independently of a driver's preferred route. Then given a fixed set of passenger requests, a high variation of answer rates suggest a higher dispersion σ. We thus add a moment for the variance of driver answer rates conditional on the type distribution of requests in a minute (see Appendix C). Our estimates show that the dispersion is limited. Table 9 reports the estimates and counterfactual results. The implied heterogeneity of mismatch tolerance is small: we simulate 20 draws in estimation, and the maximum u_{0i} is 13.6KM and the minimal 10.3KM. The counterfactual predictions show large gains under the patient algorithm for both the number of matches and driver utility.

This formulation also provides an opportunity to consider a comparative statics exercise. We increase the dispersion measure from 0.0832 to 0.5 and then 1. The counterfactual results are reported in Table 10. We find that when there is substantial variation of mismatch tolerance ($\sigma = 1$), the patient algorithm perform strictly worse than the greedy algorithm. In this case, 25% of the drivers are willing to tolerate more than 23.7 KM of mismatch. Even so, the match rate is still higher under the patient algorithm than the decentralized market.

Why might the patient algorithm match fewer agents than the greedy algorithm? We first observe that the large dispersion of mismatch tolerances substantially increases the set of requests compatible with certain drivers, thus increasing the probability that a request

is the most preferred route for two or more drivers with one of the drivers having a much smaller set of alternatives. This is especially true for the more popular routes, and the scenarios similar to the second case discussed in Section 3 become much more common. We thus discuss intuition by considering the greedy and patient algorithms using this example. Recall that in this simplified example, two drivers A and B and two passengers a and b stochastically enter the market. A prefers a to b to being unmatched, and B prefers a to being unmatched to b. Also assume that b will be matched with A in the greedy algorithm if both A and B are present and also in the patient algorithm if b exits first. The exit rate is sufficiently small that the probability of no-match exit is close to 0. In the greedy algorithm, both agents will be matched only if

1. (B, a) or (A, b) show up before other agents; or

2. (a, b) show up and then B show up.

Therefore all agents are matched with probability $\frac{2}{\binom{4}{2}} + \frac{1}{\binom{4}{2}}\frac{1}{2} = \frac{5}{12}$. With probability close to $1 - \frac{5}{12}$ only one match will be formed. In the patient algorithm, with probability close to 1 matches only occur when all four agents are in the market. Assume that the driver exit rates are equal to the passenger exit rates, then all agents are matched only if B moves first, which occurs with probability $\frac{1}{4} < \frac{5}{12}$. Therefore the expected number of matches is lower under the patient algorithm. The key insight from this calculation is that when the choice of some agent, in this case passenger a or driver A, has a large negative externality on other agents, the greedy algorithm can match more agents than the patient algorithm. The exercise also suggests that if driver exit rates are higher than passenger exit rates, the patient algorithm match rate is increased relative to the greedy algorithm. We verify this idea by simulation: we set $\kappa = 0.0152$ and $\rho = 0.079$ (while maintaining $\sigma = 1$). The resulting match rate under the greedy algorithm is 0.6652, which is indeed lower than the match rate of 0.7402 under the patient algorithm.

	Est*		Factual	Myopic	Greedy	Patient
ρ^0	4.338	Match Rate	0.4559	0.7368	0.7726	0.8185
κ	0.0797	Drv. U	3.0798	2.3727	3.0648	4.0188
ρ	0.0152	# Passenger	29.4875	14.3202	12.3847	43.94
μ	2.4493	# Driver	74.3697	45.944	61.8154	104.9694
σ	0.0832					

Table 9: Additional Heterogeneity of Mismatch Tolerance: Estimates and Counterfactual

*: All estimates are significant at the 99% level

	$\sigma = 0.5$			
	Factual	Myopic	Greedy	Patient
Match Rate	0.4492	0.6893	0.7285	0.7364
Drv. U	4.3179	3.617	3.9005	3.8154
# Passenger	28.8168	17.3264	14.7986	47.5205
# Driver	77.3887	52.2088	76.8129	113.7458
	$\sigma = 1$			
Match Rate	0.4461	0.6248	0.6652	0.6522
Drv. U	11.936	11.0243	11.3357	8.7472
# Passenger	31.3024	20.5345	18.2736	49.8147
# Driver	71.9775	58.1411	95.4436	125.1411

Table 10: Additional Heterogeneity of Mismatch Tolerance: Comparative Statics

8 Conclusion

We present an empirical framework to analyze the efficiency of a decentralized dynamic matching market. We find that (1) drivers' strategic waiting increases the market thickness but only benefits the drivers and (2) there exist centralized algorithms that can substantially benefit both sides of the market. In particular, we show that the effect of the strategic incentives and the magnitude of the efficiency improvement from the centralized algorithms depends on the level of competition and agent heterogeneity.

References

Adachi, Anna, "Competition in a Dynamic Auction Market: Identification, Structural Estimation, and Market Efficiency," *The Journal of Industrial Economics*, 2016, *64* (4), 621–655.

Agarwal, Nikhil, Itai Ashlagi, Eduardo Azevedo, Clayton Featherstone, and Ömer Karaduman, "Market Failure in Kidney Exchange," 2017.

Akbarpour, Mohammad, Shengwu Li, and Shayan Oveis Gharan, "Thickness and information in dynamic matching markets," 2017.

Allen, Jason, Robert Clark, and Jean-François Houde, "The effect of mergers in search markets: Evidence from the Canadian mortgage industry," *The American Economic Review*, 2014, *104* (10), 3365–3396.

Andreevska, Daniela, "What Kind of Airbnb Occupancy Rate Can You Expect?," *MASHVISOR*, 2016.

Arcidiacono, Peter, Patrick Bayer, Jason R Blevins, and Paul B Ellickson, "Estimation of dynamic discrete choice models in continuous time with an application to retail competition," *The Review of Economic Studies*, 2016, *83* (3), 889–931.

Arnosti, Nick, Ramesh Johari, and Yash Kanoria, "Managing congestion in matching markets," 2015.

Ashlagi, Itai, Maximillien Burq, Patrick Jaillet, and Vahideh Manshadi, "On matching and thickness in heterogeneous dynamic markets," *arXiv preprint arXiv:1606.03626*, 2016.

Baccara, Mariagiovanna, SangMok Lee, and Leeat Yariv, "Optimal dynamic matching," 2016.

Backus, Matthew and Gregory Lewis, "Dynamic demand estimation in auction markets," Technical Report, National Bureau of Economic Research 2016.

Bajari, Patrick, C Lanier Benkard, and Jonathan Levin, "Estimating dynamic models of imperfect competition," *Econometrica*, 2007, *75* (5), 1331–1370.

Banerjee, Siddhartha, Carlos Riquelme, and Ramesh Johari, "Pricing in ride-share platforms: A queueing-theoretic approach," 2015.

Bimpikis, Kostas, Wedad J Elmaghraby, Ken Moon, and Wenchang Zhang, "Managing Market Thickness in Online B2B Markets," Technical Report 2017.

Bodoh-Creed, Aaron, Joern Boehnke, and Brent Richard Hickman, "How Efficient are Decentralized Auction Platforms?," 2017.

Buchholz, Nicholas, "Spatial equilibrium, search frictions and efficient regulation in the taxi industry," Technical Report 2017.

Canals, José J and Steven Stern, "Empirical search models," in "Search Theory and Unemployment," Springer, 2002, pp. 93–129.

Castillo, Juan Camilo, Dan Knoepfle, and Glen Weyl, "Surge pricing solves the wild goose chase," in "Proceedings of the 2017 ACM Conference on Economics and Computation" ACM 2017, pp. 241–242.

Chiappori, Pierre-André and Bernard Salanié, "The econometrics of matching models," *Journal of Economic Literature*, 2016, *54* (3), 832–861.

Choo, Eugene, "Dynamic marriage matching: An empirical framework," *Econometrica*, 2015, *83* (4), 1373–1423.

_ **and Shannon Seitz**, "The Collective Marriage Matching Model: Identification, Estimation, and Testing," in "Structural Econometric Models," Emerald Group Publishing Limited, 2013, pp. 291–336.

Collard-Wexler, Allan, "Demand Fluctuations in the Ready-Mix Concrete Industry," *Econometrica*, 2013, *81* (3), 1003–1037.

Doraszelski, Ulrich and Kenneth L Judd, "Avoiding the curse of dimensionality in dynamic stochastic games," *Quantitative Economics*, 2012, *3* (1), 53–93.

Doval, Laura, "A theory of stability in dynamic matching markets," Technical Report 2018.

Eckstein, Zvi and Gerard J Van den Berg, "Empirical labor search: A survey," *Journal of Econometrics*, 2007, *136* (2), 531–564.

Eisenhauer, Philipp, James J Heckman, and Stefano Mosso, "Estimation of dynamic discrete choice models by maximum likelihood and the simulated method of moments," *International economic review*, 2015, *56* (2), 331–357.

Feng, Guiyun, Guangwen Kong, and Zizhuo Wang, "We are on the Way: Analysis of On-Demand Ride-Hailing Systems," 2017.

Fox, Jeremy T, "An Empirical, Repeated Matching Game Applied to Market Thickness and Switching," 2008.

_ , "Specifying a Structural Matching Game of Trading Networks with Transferable Utility," *American Economic Review*, 2017, *107* (5), 256–260.

Frechette, Guillaume R, Alessandro Lizzeri, and Tobias Salz, "Frictions in a Competitive, Regulated Market Evidence from Taxis," 2016.

Gallant, A Ronald, Han Hong, and Ahmed Khwaja, "The dynamic spillovers of entry: an application to the generic drug industry," *Management Science*, 2017.

Gavazza, Alessandro, "The role of trading frictions in real asset markets," *The American Economic Review*, 2011, *101* (4), 1106–1143.

_ , "An empirical equilibrium model of a decentralized asset market," *Econometrica*, 2016, *84* (5), 1755–1798.

Hall, Jonathan, Cory Kendrick, and Chris Nosko, "The effects of Uber′s surge pricing: A case study," 2015.

Hall, Jonathan V, John J Horton, and Daniel T Knoepfle, "Labor Market Equilibration: Evidence from Uber," 2017.

Hatfield, John William, Scott Duke Kominers, Alexandru Nichifor, Michael Ostrovsky, and Alexander Westkamp, "Stability and competitive equilibrium in trading networks," *Journal of Political Economy*, 2013, *121* (5), 966–1005.

Hendricks, Kenneth and Alan Sorensen, "The Value of Intermediaries in Dynamic Auction Markets," 2016.

Hu, Ming and Yun Zhou, "Dynamic type matching," 2016.

Iyer, Krishnamurthy, Ramesh Johari, and Mukund Sundararajan, "Mean field equilibria of dynamic auctions with learning," *Management Science*, 2014, *60* (12), 2949–2970.

Krusell, Per and Anthony A Smith Jr, "Income and wealth heterogeneity in the macroeconomy," *Journal of political Economy*, 1998, *106* (5), 867–896.

Lagos, Ricardo, "An analysis of the market for taxicab rides in New York City," *International Economic Review*, 2003, *44* (2), 423–434.

Loertscher, Simon, Ellen V Muir, and Peter G Taylor, "Optimal Market Thickness and Clearing," 2016.

Mortensen, Dale T, "Job search and labor market analysis," *Handbook of labor economics*, 1986, *2*, 849–919.

Ostrovsky, Michael and Michael Schwarz, "Carpooling and the Economics of Self-Driving Cars," 2018.

Ozkan, Erhun and Amy R Ward, "Dynamic matching for real-time ridesharing," 2016.

Pissarides, Christopher A, *Equilibrium unemployment theory*, MIT press, 2000.

Rogerson, Richard, Robert Shimer, and Randall Wright, "Search-theoretic models of the labor market: A survey," *Journal of economic literature*, 2005, *43* (4), 959–988.

Roth, Alvin E and Marilda Sotomayor, "Two-sided matching," *Handbook of game theory with economic applications*, 1992, *1*, 485–541.

_ , Tayfun Sönmez, and M Utku Ünver, "Pairwise kidney exchange," *Journal of Economic theory*, 2005, *125* (2), 151–188.

_ , _ , and _ , "Efficient kidney exchange: Coincidence of wants in markets with compatibility-based preferences," *American Economic Review*, 2007, *97* (3), 828–851.

Shimer, Robert and Lones Smith, "Matching, search, and heterogeneity," *Advances in Macroeconomics*, 2001, *1* (1).

Ünver, M Utku, "Dynamic kidney exchange," *The Review of Economic Studies*, 2010, *77* (1), 372–414.

Weintraub, Gabriel Y, C Lanier Benkard, and Benjamin Van Roy, "Markov perfect industry dynamics with many firms," *Econometrica*, 2008, *76* (6), 1375–1411.

A Numerical Solution for Section 3

We begin by writing down the driver problem. Consider driver A. Use $\{o, m, e\}$ to denote whether an agent has not entered the market, is in the market or has exited the market. For example,

$$S_{At} = \left\{ \begin{array}{cc} a & o \\ b & m \\ B & e \end{array} \right\}$$

represents the state a has not entered the market, b is in the market, and B has left the market. The value function for A for an infinitely small period Δ thus is

$$\begin{aligned} V_A\left(S_{At}\right) = & \underbrace{\Delta\kappa V\left(S_{At}^1\right)}_{b \text{ leaves}} + \underbrace{\Delta\kappa^0 V\left(S_{At}^2\right)}_{a \text{ enters}} + \underbrace{\Delta\gamma \max\left\{u_{bA}, V\left(S_{At}\right)\right\}}_{A \text{ moves}} \\ & + \underbrace{\Delta\rho \cdot 0}_{A \text{ exits}} + \underbrace{\left(1 - \Delta\kappa - \Delta\kappa^0 - \Delta\gamma - \Delta\rho\right) V_i\left(S_{At}\right)}_{\text{nothing happens}}, \end{aligned}$$

where

$$S_{At}^1 = \left\{ \begin{array}{cc} a & o \\ b & e \\ B & e \end{array} \right\}, S_{At}^2 = \left\{ \begin{array}{cc} a & m \\ b & m \\ B & e \end{array} \right\}$$

which simplifies to

$$V_A\left(S_{At}\right) = \frac{\kappa V\left(S_{At}^1\right) + \kappa V\left(S_{At}^2\right) + \gamma \max\left\{u_{bA}, V\left(S_{At}\right)\right\}}{\kappa + \kappa^0 + \gamma + \rho}.$$

The value function is more complicated when B is in the market. Consider

$$S_{At} = \left\{ \begin{array}{cc} a & o \\ b & m \\ B & m \end{array} \right\}.$$

We focus on pure strategy equilibrium, and assume that B's strategy is deterministic. For the purpose of the presentation, we assume that B will pick up passenger b if presented the opportunity. Thus the value function can be written as

$$V_A\left(S_{At}\right) = \frac{\left(\gamma+\kappa\right) V\left(S_{At}^1\right) + \kappa V\left(S_{At}^2\right) + \gamma \max\left\{u_{bA}, V\left(S_{At}\right)\right\}}{\kappa + \kappa^0 + 2\gamma + \rho}.$$

We can similarly write down the value functions for other states.

For the two scenarios discussed in Section 3, we use the following parameterization and solve the Bellman equation for the value function and the strategy function. Use $\emptyset$ to denote the case of being unmatched. In the first case, we assume that

$$A : u_{aA} = 2 > u_{bA} = 1 > u_{\emptyset A} = 0$$

$$B : u_{bB} = 2 > u_{\emptyset B} = 0 > u_{aB} = -1$$

We also assume that $\kappa^0 = \rho^0 = \gamma = 1$ and $\kappa = \rho = 0.1$. In a pure strategy Bayesian equilibrium where B always waits for b, the minimum V_A when a has not exited the market and b is in the market is $1.5565 > u_{bA}$ for state (omo), which means that A will always wait for a if a has not entered. In the second case, if we assume that

$$A : u_{aA} = 2 > u_{bA} = 1 > u_{\emptyset A} = 0$$

$$B : u_{aB} = 2 > u_{\emptyset B} = 0 > u_{bB} = -1.$$

In the pure strategy Bayesian equilibrium where B always waits for a, the minimum V_A when a has not exited the market and b is in the market is $0.7325 < u_{bA}$ for state (omm), which means that A will answer b and not wait for a if both b and B are in the market. If A more strongly prefers a, the result is reversed: if we assume that

$$A : u_{aA} = 5 > u_{bA} = 1 > u_{\emptyset A} = 0$$

$$B : u_{aB} = 2 > u_{\emptyset B} = 0 > u_{bB} = -1.$$

the minimum V_A when a has not exited the market and b is in the market is $1.2585 > u_{bA}$ for state (omm), meaning that A will always wait for a despite the presence of b and regardless of the presence of B.

B Additional Model Details for Section 4

B.1 Utility Function

In this appendix we give two examples of assumptions on driver traveling behavior that affect the functional form in Eq. (2). Assume the cost of traveling per distance is c. The base fare is p_0 and the per distance fare is $p_1 \ll c$. The utility of traveling by oneself is $\underline{u}$. The length of the driver and passenger trips are ℓ_1 and ℓ_2. Denote $u_0 = \underline{u} + p_0 - c\ell_1$.

1. The driver first pick ups the passenger, delivers the passenger and heads to the driver destination

$$\underline{u} + p_0 + p_1\ell_2 - c\left(\ell_{12} + \ell_2\right) = u_0 - c\ell_{12} + c\left(\ell_1 - \ell_2\right) + p_1\ell_2$$

2. The driver first picks up the passenger, then heads back to the driver origin, travels to the driver destination, travels to the passenger destination, drops off the passenger

and returns to the driver destination. The total utility of the trip is

$$\underline{u} + p_0 + p_1\ell_2 - c\,(2\ell_{12} + \ell_1) = u_0 - 2c\ell_{12} + p_1\ell_2$$

Depending on the road and traffic conditions, either assumption might hold for some travelers. If driver preference and choice data are available, one can estimate the distribution of the types of drivers as well as c and u_0. We do not have such data. We note that when ℓ_2 is a linear function of ℓ_1 and ℓ_{12}, our utility function corresponds with the first assumption after normalization on c. When p_1 is sufficiently small compared with c that it can be ignored, our utility function corresponds with the second assumption after (a different) normalization on c. These further assumptions are not unreasonable: driver decisions might rely on linear approximations in real time, and fare differences because of trip distances matter much less than the compatibility of driver preferences with passenger requests. We therefore use the formulation in the main text as a first order approximation for driver preferences and explore heterogeneity in the tolerance of mismatch in robustness checks.

B.2 Equilibrium Existence

We provide conditions for the existence of the equilibrium. Although some of the assumptions have been stated elsewhere in the paper, we state all assumptions here for completeness. Use I for the set of driver types and J the set of passenger types. A type is a vector of four elements representing the coordinates of the pickup and dropoff locations.

Assumption 4. *(Equilibrium Existence)*

- *I and J are finite*
- *$F_{\varepsilon_{it}}$ is absolutely continuous (Lebesgue), has finite first moment and has full support and identical across drivers and move instances*
- *$(\kappa^0, \kappa, \rho^0, \rho)$ are finite and positive*

- *ε, the arrival and the no-match departure processes are independent of each other and across time*

- *The total number of agents at any instant is bounded by a positive integer $\bar{N}$*

The proof relies on an application of the Brouwer's fixed point theorem. The set of drivers and passengers (D, R) follows a stationary distribution, the Markov chain of (D, R) has finite state and each state is in the same recurrent class. The assumptions on $F_{\varepsilon_{it}}$ provide a continuous mapping from the value functions to the probabilities of answering a ride, and then the transition probability of the state. The state transitions continuously map into the "reduced form" δ and G and then the value functions via the Bellman equation. The application of the Brouwer's fixed point theorem thus proves equilibrium existence.

B.3 Waiting Cost

Assume that driver choices are observed. We show that waiting cost is not separately identified from other primitives. To see this, suppose that the flow waiting cost is c. Equation (6) then becomes

$$\begin{aligned} V_i(s) = & \frac{1}{\rho + \gamma + \delta_i(s)} \\ & \times \{-c + \gamma \max\{u_{i0} - s, V_i(s)\} + \rho \underline{u}_i \\ & + \delta_i(s) E_{s'}(V_i(s') \mid s)\}. \end{aligned}$$

Adding c on both sides and using $\tilde{V}$ to replace $V + c$, we can re-write the equation as

$$\begin{aligned} \tilde{V}_i(s) = & \frac{1}{\rho + \gamma + \delta_i(s)} \\ & \times \left\{-c + \rho c + \gamma \max\left\{u_{i0} + c - s, \tilde{V}_i(s)\right\} + \rho \underline{u}_i \right. \\ & \left. + \delta_i(s) E_{s'}\left(\tilde{V}_i(s') \mid s\right)\right\}. \end{aligned}$$

The re-written equation is also a Bellman equation and $\tilde{V}$ is its solution. Because

$$(u_{i0} - s) - V_i(s) = (u_{i0} + c - s) - \tilde{V}_i(s),$$

by construction, the primitives $(c, \rho, \underline{u}_i, u_{i0})$ generate the same driver decision as $(c - \rho c, \rho, \underline{u}_i, u_{i0} + c)$ and therefore c is not identified.

C Moment Conditions

To constructively use the identification argument, we aggregate the data to the minute level. We use the following moments in the estimation:

1. Between $t-5$ and t,
   (a) Mean number of answered requests;
   (b) mean number of requests;
   (c) the standard deviations of the answered requests;
   (d) the standard deviation of the total number of requests.
2. Regression of the number of answered requests over two indicators
   (a) (# requests in t) > 41
   (b) (# requests in t) < 20
3. Regression of the number of answered requests over two indicators
   (a) (# requests in $t-6$ and $t-10$) > 70 and (# requests in t) > 41
   (b) (# requests in $t-6$ and $t-10$) < 45 and (# requests in t) > 41
4. Regression of answer rates $\frac{\#answered_t}{\#requests_t} \times 100$ over
   (a) constant

(b) HHI of the number of trip requests by class (15 classes generated from a k-means algorithm)

(c) total number of requests

5. Mean HHI

The second set of moments identifies ρ^0. Our identification argument suggests that the number of answered requests when the number of requests is large traces out the distribution of driver arrival. The third set of moments identifies ρ. To match data, the model must have a low enough ρ so that the coefficient of 3.b is higher than 3.a. The rest of the moments identify u_0. A k-means algorithm classifies all types into 15 classes based on the distance between two routes ℓ_{ij}. If u_0 is high, a high concentration of requests in a class should have little impact on the answer rates. To estimate the dispersion of mismatch tolerance, we add the variance of the residuals from the regression in 4 as an additional moment.

The weighting matrix is the inverse of the simulated variance of each moment. The moment fit of the baseline model is presented in Table 11. After the model is estimated, we followed the recommended procedure in Eisenhauer, Heckman and Mosso (2015) and conducted a Monte Carlo study by repeatedly generating the data using the estimated parameter values and estimating the model on the generated data. The bias is less than 5% for 50 replications.

		Data	Model
1	(a)	2.1275	1.88
	(b)	30.0522	30.7792
	(c)	1.552	1.3723
	(d)	16.2568	9.5612
2	(a)	2.6834	2.1364
	(b)	1.27	1.7338
3	(a)	2.5	2.125
	(b)	3.6	3.3333
4	(a)	6.0092	6.4477
	(b)	4.8759	5.5758
	(c)	-2.5226	-3.7805
5		1.3976	1.1608

Table 11: Baseline Moment Fit