Management Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Detours in Shared Rides

Ilan Lobel; , Sébastien Martin

informs
https://pubsonline.informs.org/journal/mnsc

MANAGEMENT SCIENCE
Vol. 71, No. 2, February 2025, pp. 1716–1736
ISSN 0025-1909 (print), ISSN 1526-5501 (online)

Detours in Shared Rides

Ilan Lobel,[a] **Sébastien Martin**[b,*]

[a] Stern School of Business, New York University, New York, New York 10012; [b] Kellogg School of Management, Northwestern University, Evanston, Illinois 60208

*Corresponding author

Contact: ilobel@stern.nyu.edu, https://orcid.org/0000-0002-5396-8117 (IL); sebastien.martin@kellogg.northwestern.edu, https://orcid.org/0000-0003-1021-6086 (SM)

Received: October 13, 2020
Revised: December 17, 2021; August 17, 2023
Accepted: December 2, 2023
Published Online in Articles in Advance: May 17, 2024

https://doi.org/10.1287/mnsc.2020.03125

Abstract. Detours are considered key for the efficient operation of a shared rides service, but they are also a major pain point for consumers of such services. This paper studies the relationship between the value generated by shared rides and the detours they create for riders. We establish a limit on the sum of value and detour, and we prove that this leads to a tight bound on the Pareto frontier of values and detours in a general setting with an arbitrary number of requests. We explicitly compute the Pareto frontier for one family of city topologies and construct it via simulation for several more networks, including one based on ride-sharing data from commute hours in Manhattan. We find that average detours are usually small, even in low-demand-density settings. We also find that by carefully choosing the match objective, detours can be reduced with a relatively small impact on values and that the density of ride requests is far more important than detours for the effective operations of a shared rides service. In response, we propose that platforms implement a two-product version of shared rides and limit the worst-case detours of its users.

History: Accepted by Hamid Nazerzadeh, data science.

Supplemental Material: The online appendix and data files are available at https://doi.org/10.1287/mnsc.2020.03125.

Keywords: ride-sharing • matching • networks • simulation • transportation

1. Introduction

Shared rides are trips provided by ride-sharing platforms, such as Lyft and Uber, where multiple ride requests are served together by a single vehicle. Shared rides are a socially desirable means of serving passengers as they reduce both congestion and carbon emissions as compared with solo rides. Shared rides are also cheaper to serve as they enable drivers to fulfill multiple ride requests simultaneously. With these advantages in mind, both Lyft and Uber have said in the past that they aim for more of their rides to be shared. For instance, before the coronavirus pandemic forced the ride-sharing platforms to pause shared rides services, Lyft had said that it aimed to have half of its rides be shared by the end of 2020 (Griswold 2018, Lyft 2018).

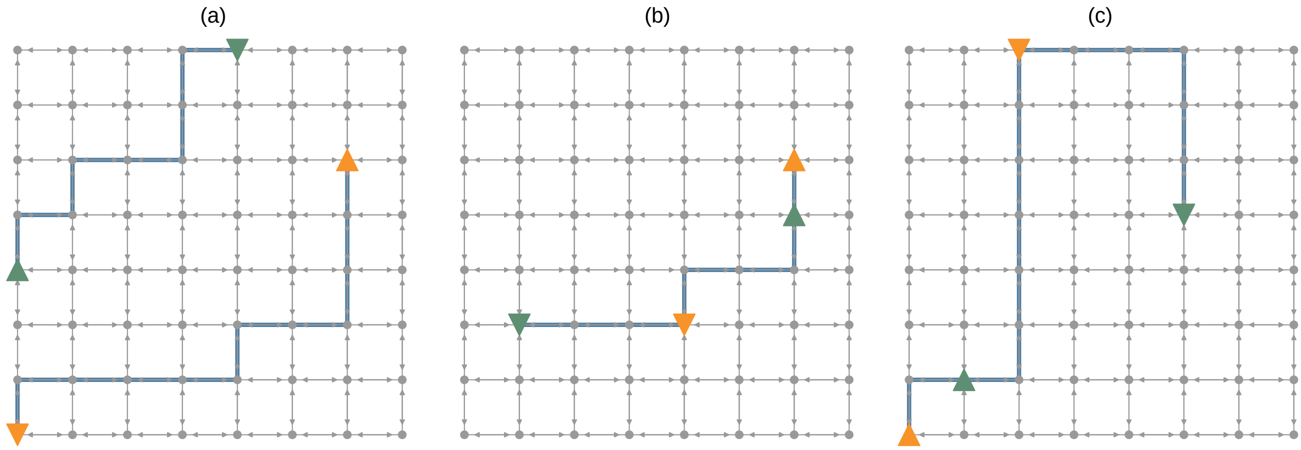
A key challenge in attaining these goals is that the experience of taking a shared ride can be a quite a bit worse than the experience of taking a solo ride. In particular, a major pain point for consumers of shared rides is detours, which are defined as drivers taking a longer way to one's destination than they would if it was a solo ride. Lo and Morseman (2018) show via an ethnographic study of Uber users that short on-trip duration is the number 1 feature that passengers would like from a shared rides service (see Lo and Morseman 2018, figure 7), and being able to offer short on-trip duration requires ensuring that detours are small.

Ride-sharing companies, like Lyft and Uber, allow for detours in their shared rides products in order to make it easier to match passengers to each other. The key question we aim to address in this paper is how important these detours are to the operations of a shared rides service. That is, we will shed light on the relationship between the detour and the value of a shared ride, where value is defined as the driving distance saved by serving two requests using a single car. Understanding this relationship will allow us to gain insights on how a shared rides marketplace should be run.

1.1. A Motivating Example: Detours on a Grid

As an example, consider a city where the streets form a grid, and all the streets are two-way streets. Requests arrive with origin-destination pairs, with both origins and destinations being located at intersections (or equivalently, each customer is requested to walk to/from the closest intersection). Consider two ride requests generated at random, with both origins and destinations being uniformly distributed among all possible locations. Figure 1(a) shows an example of two ride requests, with the upward triangles representing pickup points and

Figure 1. (Color online) Examples of Pairs of Requests on a 8×8 Grid Network



Notes. Upward triangles are pickups, and downward triangles are drop-offs. (a) No match and negative value. (b) Match, detour = 0, and value = 4. (c) Match, detour = 6, and value = 1.

the downward triangles representing drop-off locations. For some pairs of ride requests, such as the one depicted in Figure 1(a), it does not make sense for the platform to match them as serving those pairs of ride requests in a single car would create workload (increase the required driving distance) rather than reduce it. For such pairs, we assume the platform dispatches each request as a solo ride rather than match them. For other pairs of ride requests, such as the ones represented in Figure 1, (b) and (c), the platform could reduce the amount of driving necessary to fulfill the requests by matching the requests. We use the term value to represent the amount of driving saved by the match, and we measure it in city blocks (more formally, using the ℓ_1 norm). In Figure 1(b), the value of the match is four, and the match does not create a detour for either passenger. In Figure 1(c), the value of the match is one, but the match does cause one of the riders to have a detour of length 6.

As one begins to analyze this problem, a property that emerges is that creating matches with both high value and high detour is harder than it might appear at first sight. We encourage the reader to try it in a grid like the one used in Figure 1. Rides that go in opposite or divergent directions cannot create detours because the platform will prefer not to match those rides. Creating longer detour often leads to lower values, as is the case in Figure 1(c). This exercise highlights that there exists a connection between the value of a match and its associated detour, which is the topic explore in this paper.

1.2. Our Contributions

Our first main result, Theorem 1, shows that for a pair of rides, there exists an optimal route that both maximizes value and minimizes detour. That means that for a single pair of requests, there is no trade-off between these two objectives (this trade-off will emerge when we consider which pairs of requests to match together). Theorem 1 also shows that detours and values always add up to an object that we call the shared component. The shared component is the part of the ride where both riders are in the car simultaneously. This allows us to obtain Theorem 2, which says that values and detour (normalized by the length of the unmatched rides) add up to no more than 1/2. This theorem proves that there is a limit to the sum of detours and values, and therefore, creating a matched pair with a large normalized value and a large normalized detour at the same time is infeasible.

The results apply to any city that can be described by a metric space. We next focus on specific city networks in order to understand joint value-detour distributions. In Propositions 3 and 4 in the Online Appendix, we explicitly compute joint distributions for two specific network structures: a circle city and a tree city. It turns out that even in the single-dimensional circle city, computing the joint distribution of value and detour is a difficult exercise because of the combinatorial nature of the problem. Because of this fact, we turn our attention to generating joint distributions for more complex networks via simulation. We do so both for some synthetic networks, like the grid from Figure 1, as well as for a network representing Manhattan using data from all ride requests during morning commutes that occurred in February 2020 (see Figure 2). Despite Manhattan being famous for being roughly a grid, the shared rides statistics of Manhattan turned out to be closer to the tree network's statistics and those of a topology we call the highway network.

We then turn our attention to settings with multiple requests rather than pairs. In Theorem 3, we show that the 1/2 bound on the sum of normalized values and detours also applies to the Pareto frontier of normalized values and detours with multiple requests, and that

Figure 2. (Color online) Thirty Ride Requests in Manhattan, with Some Requests Dispatched as Solo Rides and Others Dispatched as Matched Rides

such a bound is tight. We also prove that this Pareto frontier is a nondecreasing concave function, that the Pareto frontier moves up whenever density doubles, and that the limit normalized value is 1/2 when the number of requests goes to infinity. We also compute Pareto frontiers for many of the same networks discussed earlier, including the Manhattan network with real-world data. The Pareto frontiers for the Manhattan network look similar to the ones from the highway network. In all of the networks, expected detours in the Pareto frontier are relatively small regardless of the demand density, and the gains from penalizing detours are large (that is, there is little cost in terms of value from adding at least some detour penalty). Furthermore, the returns to density are far greater than the gains from allowing detours.

We then study two strategies that leverage these findings and that the platform can use to improve user experience and profit. First, we suggest in Section 5 that the platform should consider limiting the worst-case detours of its passengers. We first provide a characterization of request-level detours (instead of the average or sum of detours) in Theorem 4, and then, we prove in Theorem 5 and via extensive simulation with real-world data that introducing request-level detour constraints barely changes our results and can be done at low cost for the platform if users are willing to tolerate detours of at most half of their trip length. Tighter detour constraints can also be used but at a cost in terms of system performance. A second strategy is to leverage riders' heterogeneous sensitivity to detours to increase shared rides demand and profit. We propose a two-product ride-sharing system, with one of the products being a low-detour option. If such differentiation could lead to greater density (because a fraction of potential users do not take shared rides to avoid long detours), this two-product system could significantly outperform a single-product system in terms of both value and detour. In Section 7, we make the case that this two-product strategy is economically viable by calibrating a simple economic model using data from the U.S. Department of Transportation.

1.3. Limitations of Our Study

Our study does not analyze shared rides with more than two ride requests sharing each vehicle, even though shared riders with three ride requests are often used in real life, and some shared systems use even larger vehicles, such as vans and minibuses. The insights of our paper would not apply to larger vehicles, such as minibuses, because the geometry of the detour with many riders would be quite different. Whether they apply to systems with a nontrivial number of cars with three independent riders is left as an open question. Focusing on matches of two ride requests is a deliberate decision that allows us to focus on the simpler (yet still quite complicated) geometry of matching pairs of ride requests together as well as use standard graph matching (rather than hypergraph matching) to analyze shared rides systems.

Many of the insights that we obtain are derived via simulation, and simulation insights come with the standard caveat that one needs to be careful in generalizing from them. In particular, we have conducted simulations on a collection of synthetic networks and in Manhattan. We believe that Manhattan is a sensible place to analyze because its high density and limited highways make it a promising location for a shared rides system, and it is, therefore, often used as a test bed for shared rides deployments in practice. To be clear, Manhattan is obviously not a representative stand-in for an average American city. To gain some understanding of the performance of shared rides in other locations with less density and more highways, we do analyze networks with varying levels of density and a synthetic highway network. We acknowledge that a limitation of our study is that we did not replicate our simulation with data from other real-world cities.

Most of our analysis disregards pickup and drop-off times, focusing on the geometry of detours instead. In Section 6, we add fixed pickup and drop-off times, and we show that they reduce the performance of shared rides systems even in the asymptotic limit as the number of requests goes to infinity. Pickup and drop-off times affect shared rides negatively, and they do so in two ways; they both reduce the value of a match and increase detour times. This is particularly true if pickup and drop-off times are large. Perhaps one way to interpret our results is that in many situations, pickups and drop-off times can be more problematic for shared rides than detours. Consider, for instance, a request that originates at the start of an avenue and ends at the other end of that same avenue, and suppose that this ride stops at every intersection for a passenger pickup or drop-off. This would be a slow ride despite the lack of detours. That is, there are other frictions besides detours that can make the experience of a shared ride significantly worse than that of a solo ride.

Finally, our analysis is based on an assumption that we can ignore drivers. In Section 8, we argue that this assumption is a reasonable approximation. In particular, we prove in Theorem 6 that the cost of this approximation can be bounded by the product of three factors. Furthermore, under reasonable conditions (that assume relatively plentiful supply), the product of these factors can be approximately 3%.

1.4. Related Literature

Ride-sharing (or ride hailing) has become a popular topic in the operations literature over the last few years. Most of the work in this literature studies the setting with solo rides and focuses on topics such as spatiotemporal pricing with strategic drivers (Lu et al. 2018, Bimpikis et al. 2019, Garg and Nazerzadeh 2019, Besbes et al. 2021, Ma et al. 2021, Afèche et al. 2023), dispatch rules for drivers and the need for idle drivers (Alonso-Mora et al. 2017, Castillo et al. 2017, Ozkan and Ward 2020, Besbes et al. 2022), and the consequences of drivers choosing their own schedules (Cachon et al. 2017, Taylor 2018, Gurvich et al. 2019). Shared rides are a less explored topic in the literature, even though managing a shared rides service is perhaps the most operationally complex problem that ride-sharing platforms face. Two papers that do consider shared rides are Gopalakrishnan et al. (2016) and Biswas et al. (2018). The first of these papers introduces a notion of sequential individual rationality that captures the inconvenience caused to riders by adding new passengers to their rides. The second paper uses this notion to construct an integer programming formulation for the problem of detour-aware dispatch. The survey on ride-hailing systems by Wang and Yang (2019) includes a section on ride-splitting operations, which is their terminology for papers on the operations of shared rides.

One of the most relevant papers to our work is Santi et al. (2014), a paper that defined the notions of shared component and shareability networks, and it argued via simulation that New York City has enough taxi demand density to allow for a shared taxi system with relatively small detours. Remarkably, Santi et al. (2014) was written before the launch of shared rides systems by Lyft and Uber. Our work goes a step farther than Santi et al. (2014) by building a model of shared rides and proving limits on the size of detours and the economic value of long detours. Our framework also allows us to show several interesting managerial insights that were not shown in prior work, such as (1) that expected detours are typically small regardless of density, (2) that penalizing detours can generate a great reduction in detours at a mild cost in terms of value, (3) that density is more important for the operation of a shared rides system than significant detours, and (4) that a two-product shared rides marketplace could be superior to a single-product system.

A recent relevant paper is Daganzo et al. (2020); this paper studies a shared rides service with detour constraints via a queuing model. Like us, they argue that imposing detour constraints is a good idea for shared rides services. At the same time, their analysis and results are quite different from ours, with their paper focusing on the effect of fleet size. Even though they also use the terminology of premium service, their meaning of the term is fairly different from ours. They use it to describe their single-product detour-constrained proposal, whereas we use it to represent the superior product within the context of a two-product shared system.

The model in our paper is a static one, where the platform sees a collection of requests at once and chooses which pairs to match. This is consistent with the primary way that dispatching is done in industry, which is via batching and reoptimization (Yan et al. 2019). That is, requests are collected over a period of a few seconds and then matched together, with this process repeating itself every few seconds. The academic literature also defends this approach, with a long line of papers supporting it (Wong and Bell 2006, Berbeglia et al. 2010, Bertsimas et al. 2019, Ashlagi et al. 2022). This last paper, in particular, proves that batching leads to a constant approximation of the optimal solution under mild conditions. A noteworthy exception is Aouad and Saritaç (2022), which advocates against batching and proposes an approximate dynamic programming as an alternative approach to matching.

2. The Model

We consider a ride-sharing platform operating in a city described by a compact metric space $(\mathcal{C}, d)$, where $\mathcal{C}$ is the set of locations and d is its associated distance metric. We will use the shortcut $d(A, B, \ldots, Y, Z)$ to represent the sum of distances $d(A, B) + \cdots + d(Y, Z)$.

Let $n \in \mathbb{N}$ be a number of ride requests, which we will assume are simultaneous. For each $i = 1, \ldots, n$, ride request i corresponds to a pair (O_i, D_i), where $O_i \in \mathcal{C}$ represents the request origin and $D_i \in \mathcal{C}$ represents the request destination. Dispatching a car to serve a ride request i by itself (as a solo ride) causes the platform to incur a cost equal to the distance between the origin and the destination. That is, the unmatched cost of ride i is $c_u(i) \triangleq d(O_i, D_i)$. All of these requests are shared requests, which in our model, means we can serve up to two requests with a single car. Now, consider an ordered pair of requests (i, j) that we choose to serve with a single car. There are four possible routings that the platform could choose depending on which rider is picked up first and which rider is dropped off last. We use the notation $\Pi(i, j)$ to denote the routes according to

$$\Pi(i,j) \triangleq \begin{cases} 1 & \text{to represent route } O_i \to O_j \to D_i \to D_j; \\ 2 & \text{to represent route } O_i \to O_j \to D_j \to D_i; \\ 3 & \text{to represent route } O_j \to O_i \to D_i \to D_j; \\ 4 & \text{to represent route } O_j \to O_i \to D_j \to D_i. \end{cases}$$

For a given matched pair of ride requests (i, j) and a given route $\pi \in \{1, \ldots, 4\}$, we denote its matched cost by $c_m(i, j, \pi)$ to correspond to the sum of distances required to fulfill these ride requests along this route. For example, $c_m(i, j, 1) = d(O_i, O_j, D_i, D_j)$. We ignore routes, such as $O_i \to D_i \to O_j \to D_j$, that can be decomposed into separate unmatched rides $O_i \to D_i$ and $O_j \to D_j$ with a reduction in driving costs.

The detour $r(i, j, \pi)$ of a matched pair of requests (i, j) corresponds to the additional distance incurred by the rider because of sharing a ride under route $\pi \in \{1, \ldots, 4\}$. For example, for a matched pair (i,j) that uses route $\Pi(i, j) = 1$ (that is, $O_i \to O_j \to D_i \to D_j$), the detour is equal to $(d(O_i, O_j, D_i) - d(O_i, D_i)) + (d(O_j, D_i, D_j) - d(O_j, D_j))$, with the first set of parentheses corresponding to rider i's individual detour and the second set of parentheses corresponding to rider j's individual detour.

We use $G = (N, E)$ to represent the matching graph, with the node set being identical to the set of ride requests, $N = \{1, 2, \ldots, n\}$, and the edge set E corresponding to all possible ordered pairs of distinct ride requests. A match M is defined as a subset of E, where every node belongs to at most one edge. A dispatch plan is defined as a pair (M, Π), where M is a match and Π is a routing plan that associates each edge in M with a route in $\{1, \ldots, 4\}$. The cost of a dispatch plan (M, Π) is defined as the sum of its associated driving costs:

$$C(M, \Pi) \triangleq \sum_{(i,j) \in M} c_m(i, j, \Pi(i, j)) + \sum_{k \notin M} c_u(k), \tag{1}$$

where $\{k \notin M\}$ is used to represent the set of nodes not included in any edge of M (requests that are dispatched as solo rides). We will use the notation $C(\emptyset)$ to represent the cost of dispatching all ride requests as solo rides (i.e., $C(\emptyset) \triangleq \sum_{k \in N} c_u(k)$).

We define the value of a matched pair of requests (i, j) and its associated route $\pi \in \{1, \ldots, 4\}$ by $v(i, j, \pi) \triangleq c_u(i) + c_u(j) - c_m(i, j, \pi)$. Intuitively, if $v(i, j, \pi)$ is positive, including (i, j) with route π into our dispatch plan reduces costs, whereas the opposite is true if $v(i, j, \pi)$ is negative. We say a match M is nonnegative if there exists some Π such that $v(i, j, \Pi(i, j)) \geq 0$ for all $(i, j) \in M$ (and say M is positive if the value is strictly positive for all edges in M). Throughout the paper, we assume all matches are nonnegative. We also define the value of an entire dispatch plan:

$$V(M, \Pi) \triangleq \sum_{(i,j) \in M} v(i, j, \Pi(i, j)). \tag{2}$$

Combining the definition with the definition of the cost of a dispatch plan (Equation (1)), we get that $C(M, \Pi) = C(\emptyset) - V(M, \Pi)$. Because $C(\emptyset)$ is a constant, minimizing $C(M, \Pi)$ is equivalent to maximizing $V(M, \Pi)$. In a similar spirit, we will define the detour of a dispatch plan by

$$R(M, \Pi) \triangleq \sum_{(i,j) \in M} r(i, j, \Pi(i, j)). \tag{3}$$

At a high level, designing a shared rides service can be thought of as a multiobjective optimization problem, where the platform creates algorithms that choose dispatch plans (M, Π) with two simultaneous goals in mind: maximizing value (Equation (2)) and minimizing detours (Equation (3)). Our goal in this paper is to explore the trade-offs between these two objectives. That is, we want to understand the extent to which detours are necessary to achieve high-value matches as well as shed as much light as possible on the relationship between value and detour.

2.1. Discussion of Model Assumptions

In order to gain insight into the relationship between value and detour, we have abstracted away a few real-life features of ride-sharing systems. We now discuss some of these simplifying assumptions.

In our theoretical analysis, we require distances to be symmetric, excluding network features such as one-way streets. Although allowing asymmetric distances breaks some of our theoretical results (for example, it becomes possible to construct a pair of requests that violates Theorem 2), we show via simulation in Section 3.3 that one-way streets barely change results in an 8×8 grid. We also allow for asymmetric travel times when simulating rides in Manhattan.

Another way we simplify the model is by making it static. The real-world problem of matching requests is obviously a dynamic one, with requests arriving over time. We deal with this assumption by building a complex dynamic ride-sharing simulator and showing that key insights remain valid in a fully dynamic model (see

Online Appendix B). Our dynamic model also incorporates ride chaining, a feature that is hard to capture in a static model.

We also assume that matches include at most two riders. Optimally matching three or more riders corresponds to maximum-weight matching in a hypergraph, which is an *APX*-hard problem (Kann 1991). Furthermore, we deliberately want to model small-vehicle ride-sharing systems and not large-vehicle systems, such as buses or shuttles. Door-to-door service of a large-vehicle ride-sharing system is not likely to be feasible without creating large detours.

A further simplification is that we treat drivers as being plentiful and distributed over the city. Whenever we need a new driver, we assume that one is available at the pickup location where he or she is needed. We discuss in depth why this assumption is reasonable in Section 8.

An important question is why study detours. A different metric of interest is estimated time to destination (ETD), which includes both travel time with detour and pickup time. A curious reader might wonder why we chose to study detours rather than ETDs. Our answer is twofold. First, Lo and Morseman (2018) shows that riders care more about on-trip duration (which is closely tied to detours) than ETDs (in Lo and Morseman 2018, figure 7, short on-trip duration is rider concern number 1, whereas ETD is denoted as "ETA" and is rider concern number 3). This is because waiting at your pickup spot (say home or a restaurant) is usually far more pleasant than time spent with a stranger sharing a car. Another reason is that detours are a more geometric and thus, structural property of the system. ETDs are highly dependent on pickup times and thus, very sensitive to the static modeling and to our unlimited supply of drivers assumption. In contrast, we have shown our metric of detour to be robust to dynamics and chaining in Online Appendix B.

3. The Value and Detour for a Pair of Requests

In this section, we study properties of pairs of ride requests. We first derive general results on the relationship between the detour and the value of a matched pair of requests, and then, we study the implications for individual ride requests. Our next step is to specialize our results to specific city topologies, where we can compute value-detour distributions in closed form. We finish the section by studying the value-detour distributions in several other settings via simulation.

3.1. The Pairwise View of Values and Detours

Given a pair of matched requests, we first try to understand the relationship between its value and its detour. Before we present our first theorem, we need to define the concept of the *shared component* of a match. For a given matched pair (i,j) and an associated route $\pi \in \{1,\ldots,4\}$, the shared component $s(i,j,\pi)$ corresponds to the distance driven with two riders in the car. For example, with route $O_i \to O_j \to D_i \to D_j$, the shared component is equal to $d(O_j, D_i)$ because the segment $O_j \to D_i$ is the only part of the route where both riders i and j are sharing the car.

Theorem 1. *Consider a matched pair of requests $(i,j) \in E$. Then, the following are true.*

1. *The route(s) that maximize value are also the route(s) that minimize detour and the shared component*:

$$\arg\max_{\pi\in\{1,\ldots,4\}} v(i,j,\pi) = \arg\min_{\pi\in\{1,\ldots,4\}} r(i,j,\pi) = \arg\min_{\pi\in\{1,\ldots,4\}} s(i,j,\pi).$$

2. *For any route $\pi \in \{1,..,,4\}$, the value, detour, and shared component satisfy*

$$v(i,j,\pi) + r(i,j,\pi) = s(i,j,\pi).$$

The proofs of all results are presented in the Online Appendix. The first part of the theorem states that when we are considering only a matched pair of rides, there are no trade-offs between value and detour. A route that optimizes value is also a route that minimizes detour. This implies that we do not need to consider the full set of potential dispatch plans (M,Π) but that we can instead consider only matches M and assume the platform will use an optimal routing plan Π as determined by Theorem 1, part (1). As a consequence, we will use from now on the shorthand notation $v(i,j)$, $r(i,j)$, $c_m(i,j)$, and $s(i,j)$ to refer, respectively, to the value, detour, matched distance, and shared component of a matched pair assuming that an optimal route is used. Similarly, we will use $V(M)$ and $R(M)$ to refer to the value and detour, respectively, of a match M assuming an optimal routing plan is used.

Furthermore, the first part of Theorem 1 also tells us an easy way to determine the optimal route of a matched pair; the optimal route is the route that minimizes the shared component. The shared component is an easier object to understand than the value of the detour because it is equal to a single distance between two points (see Table 2 in Online Appendix C, which is inside the proof of Theorem 1). To be clear, value and detour are not perfectly aligned objectives. Theorem 1 only says that there are no trade-offs between them when we consider potential routes for a matched pair of rides. Trade-offs between these two objectives will, in fact, emerge when we consider the problem of how to match ride requests to each other.

We now turn our attention to the implications of the second part of Theorem 1. Focusing our attention on the optimal route, that theorem states that the value and the detour of a matched pair are equal to its shared component:

$$v(i,j) + r(i,j) = s(i,j). \tag{4}$$

Equation (4) provides valuable structural insights into the relationship between value and detour. First, conditional on the shared component, value and detour are perfect substitutes. That is, if you consider a family of matched pairs that have the same shared component, value will go up within this family as detour goes down. Second, because values and detours are both nonnegative quantities, both must also be bounded by the shared component. For instance, if a matched pair has a small shared component, then both its value and its detour must be small as well. Third, Equation (4) hints that value and detour are not completely free variables and that there might exist some feasibility constraints tying them together. We will formalize this intuition in Theorem 2. Before we are able to do that, we need to introduce two new objects: value ratio and detour ratio. For a matched pair (i,j), we refer to its value ratio $\hat{v}(i,j)$ as the value of the match normalized by the sum of its unmatched distances:

$$\hat{v}(i,j) \triangleq \frac{v(i,j)}{c_u(i) + c_u(j)}. \tag{5}$$

The detour ratio $\hat{r}(i,j)$ is defined by a similar normalization:

$$\hat{r}(i,j) \triangleq \frac{r(i,j)}{c_u(i) + c_u(j)}. \tag{6}$$

We will say that a pairwise value-detour ratio pair $(\hat{v},\hat{r})$ is feasible if there exists a city $(\mathcal{C},d)$ and a nonnegative-valued pair of rides requests i and j such that $\hat{v}(i,j) = \hat{v}$ and $\hat{r}(i,j) = \hat{r}$. The next result characterizes the set of feasible pairwise value-detour ratio pairs.

Theorem 2. *Let the set $\mathcal{S}_0$ be defined as*

$$\mathcal{S}_0 = \left\{ (\hat{v},\hat{r}) \in \mathbb{R}^2 \,\middle|\, \hat{v} + \hat{r} \le \frac{1}{2},\ \hat{r} \ge 0,\ \hat{v} \ge 0 \right\}.$$

Then, a pairwise value-detour ratio pair $(\hat{v},\hat{r})$ is feasible if and only if it belongs to $\mathcal{S}_0$.

The most important part of Theorem 2 is that $\hat{v}(i,j) + \hat{r}(i,j) \le 1/2$ for any pair of rides i and j in any city $(\mathcal{C},d)$. This result immediately implies that $\hat{r}(i,j) \le 1/2$, which means that the average detour in any city and any matching algorithm cannot be more than half of the average trip length. This result also implies that high-value pairs (where $\hat{v}(i,j)$ is close to $1/2$) must also be low-detour pairs. Similarly, high-detour pairs (where $\hat{r}(i,j)$ is close to $1/2$) must be low-value pairs. That is, there a limit to the economic value of detours in shared rides. If a platform is able to generate high-value pairs in its system, those pairs will automatically be low-detour ones. In Section 1.1, we discussed how constructing a matched pair of requests on a grid with both a high value and a high detour seemed difficult. Theorem 2 elucidates why; there is a limit to what the sum of value and detour can be.

3.2. Matching Pairs of Requests on Specific City Topologies

Theorem 2 established general bounds for value-detour distributions. This subsection explores how the choice of specific city topologies and request distributions influences the relationship between value and detour. Because a pair of rides can be matched using four different routes, obtaining the closed-form joint value-detour distribution for a given city turns out to be a combinatorial and laborious process.

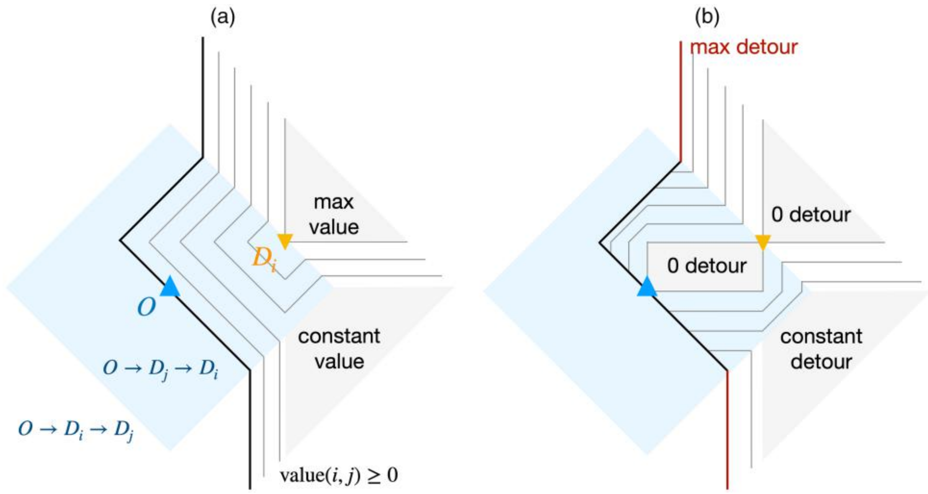
For instance, consider a city that is a compact subset of $\mathbb{R}^2$ with the distance being the ℓ_1 norm. This city represents the asymptotic limit of a grid network. For simplicity, assume that the two requests i and j have the same origin. Figure 3 assumes a given O and D_i and shows how value and detour change as we vary the location of D_j. The lines in Figure 3, (a) and (b) represent points with the same value and detour, respectively. Even for this simple example, the distribution of value and detour is quite cumbersome and combinatorial in nature, and the distribution becomes significantly more complex if we allow the two origins to be distinct points.

The joint value-detour distributions depend strongly on the city topology. To illustrate the impact of topology, we provide closed forms for two simple (yet nontrivial) city topologies: `circle` and `tree`. Getting these closed forms requires extensive analytical work, and all the results are listed in Online Appendix A. `Circle` illustrates a network that is favorable for shared rides; all trips are uniformly distributed along a main artery (a unit circle). In Proposition 3 in Online Appendix A, we show that more than 70% of the matches with positive value in `circle` do not have any detour. Even in the worst case, this setting has a maximum detour ratio $\hat{r} = 1/3$ instead of the $1/2$ general bound. On the other hand, `tree` is designed to present limited sharing opportunities. All requests share the same origin (the root of a binary tree), but the destinations are distributed along the leaves of the tree, which means that it is impossible to match two nonidentical requests without detour. Proposition 4 in Online Appendix A shows that matches with high detours and low value are likely in `tree`.

3.3. Comparing Value-Detour Distributions

To obtain a more general sense of what value-detour joint distributions look like in other settings, we use extensive simulations on a collection of networks, including both synthetic networks and a network based on Manhattan ride-share data.

Figure 3. (Color online) An ℓ_1 City, with Given Origin O and Destination D_i



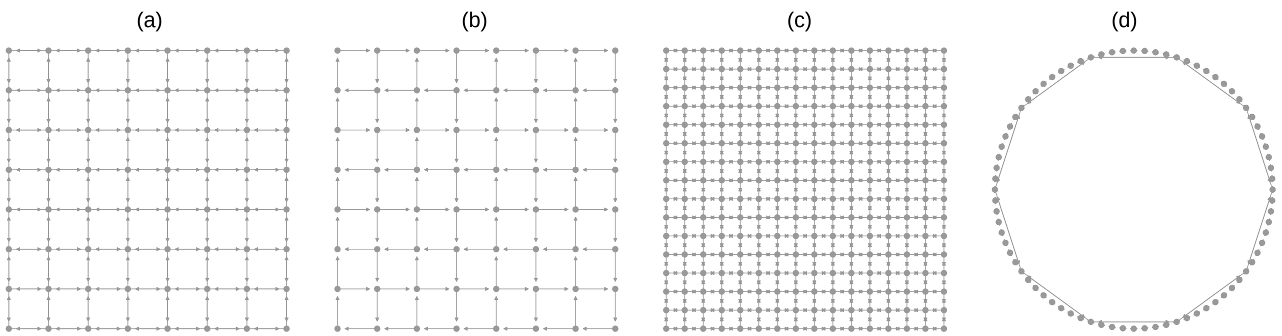
Notes. As we vary D_j, we get a different value and detour for the match. The diamonds represent the area where $O \to D_j \to D_i$ is the optimal route. The thick lines represent the zero-value line, and the other lines represent points with identical value (panel (a)) and identical detour (panel (b)). (a) Value as a function of D_j. (b) Detour as a function of D_j.

3.3.1. Synthetic Networks. We first consider ride-sharing in the four synthetic networks presented in Figure 4. Specifically, we consider discrete networks where the distance between two adjacent nodes is set to one and where the distribution of the origins and destinations of trips is independent and uniform across the nodes of the network. Therefore, we have $v, r \in \mathbb{N}$, which simplifies the representation of the joint distributions of values and detours.

The 8×8 grid network from Figure 4(a) is the same as the one presented in Section 1. It is meant to represent trips in a city core with two-way streets. Figure 4(b) considers a grid with alternating one-way streets. Our theorems do not apply to one-way streets (they violate the metric space assumption because they imply that distances are not symmetric), and our simulation is meant to test whether shared rides behave very differently in such networks. Figure 4(c) explores a larger grid. Figure 4(d) is meant to understand the impact of highways on detours. A potential hypothesis is that highways create longer detours as a vehicle with a rider might need to leave the highway to pick up a potential second rider. On each network, we simulate 10,000,000 trips, and we use them to evaluate the empirical joint distribution of detour and value.

3.3.2. Ride-Sharing in Manhattan. On top of the synthetic networks, we also study detours using real data from Manhattan ride requests. We constructed a road network of Manhattan using the publicly available OpenStreetMap geographical data (Boeing 2017, OpenStreetMap Contributors 2020). The distance d corresponds to the average travel time between the 4,013 nodes of this routing network. To estimate these travel times, we use average road speeds from the OpenStreetMap data as well as the Dijkstra shortest-path algorithm

Figure 4. Synthetic Networks We Simulate



Notes. The distance d between two adjacent nodes is always one, and the trips' origins and destinations are uniformly and independently distributed. (a) `grid_08` (8×8 grid). (b) `grid_08_oneway` (8×8 grid with one-way streets). (c) `grid_16` (16×16 grid). (d) `Highway` (circular network of 80 nodes with shortcuts).

to get the fastest time between any two nodes (we compute approximately 16 million such distances), accounting for one-way streets. See the lower right corner of Figure 5 for a visualization of the Manhattan network.

We generate two different settings on this network by making different assumptions regarding the probability distribution of origin-destination pairs. We call the first one `manhattan_uniform`, where both origins and destinations are sampled uniformly from the nodes in the Manhattan network. The nodes are assumed to intersections, assuming that the shared rides system asks users to walk to the nearest intersection. The second one, which we call the `Manhattan` network, is the more interesting one. For the `Manhattan` network, we use historical request data to construct a probability distribution of origin-destination requests. In order to construct the distribution of ride requests, we use the New York City Open Data platform to access the historical record of for-hire vehicle data. Specifically, we have access to the origin and destination time and location information for all the trips of the major ride-sharing platforms in New York City: Lyft, Uber, and Via. In order to match trips that happened in similar conditions, we consider one month of data (February 2020, the last month before the coronavirus stopped shared rides services in New York) during morning commute trips (6 a.m. to 10 a.m., Monday to Friday except holidays) with both origin and destination in Manhattan. After filtering out data errors, this returns a total of 847,699 historical trips. Our requests are uniformly drawn from this set of trips. For privacy reasons, the exact origin and destination of each trip are not available, but we have access to the origin/destination "neighborhoods" (Manhattan is divided into 69 of them). We generate precise origins and destinations by sampling uniformly from intersections within each neighborhood.

Figure 5. (Color online) The Manhattan Road Network

Notes. An example of a high-value match ($v = 10$ minutes 38 s, $r = 2$ minutes 13 s) is represented. Both trips are heading south, and their respective fastest paths are drawn in red and blue. The path of the match is in white. The blue trip incurs a detour because it would be faster for them to use the highway by the East River.

3.3.3. Value-Detour Distributions. Table 1 shows several properties of the joint distributions of value and detour for the four networks described in Figure 4, the two networks that were analyzed in Section 3.2 (`circle` and `tree`), and the two variants where we use the network and request data from Manhattan. There is a clear cluster, with the first four networks of the table sharing some properties. First, the probability of zero detour given match is fairly high (range: 29.2%–71.1%) within this group. This statement also partially applies to the `highway` network, a network that we designed to try to facilitate detours but where the probability of zero detour given match is still 19.5%. Second, in the first four networks, we have $\mathbb{E}[r|v>0] < \mathbb{E}[v|v>0]$; this is counterintuitive because these are matches between two uniformly drawn requests, which a priori, we might expect to be poorly matched. Meanwhile, the final four networks share some similarities too. In these networks, we also have that $\mathbb{E}[r|v>0] \geq \mathbb{E}[v|v>0]$. That is, when matching two random trips, in expectation, the total detour incurred is greater than the value created. If we exclude the `highway` network from this cluster, the probability of matching without detour is very low (range: 0.05%–0.16%). We plot in Figure 6 the empirical joint distributions of value and detour in six of the test networks, and we show the expected detour given value $\mathbb{E}[r|v]$ in each of the networks in red.

It is perhaps surprising that `Manhattan`, with its world-famous grid layout, does not behave like a grid network in terms of the metrics from Table 1 and that the Manhattan network has shared rides statistics closer to those of the `highway` and the `tree`. One hypothesis is that the density of requests being nonuniform is the driver behind this difference. We can rule out this hypothesis as the `manhattan_uniform` network has similar properties to the `Manhattan` network. When it comes to the low probability of zero detour, the actual answer has to do with the (near) uniqueness of shortest paths. Zero detours are frequent in a grid network because there are typically many routes with the same distance to get from an origin to a destination. Meanwhile, in the `Manhattan` network, the shortest path from an origin to a destination is often unique or near unique because different streets and avenues can have different speeds, and also, there are roads that break the symmetric pattern, like the diagonal Broadway. In fact, the existence of streets like Broadway and the highways along the East River and the Hudson River on both sides of the island of Manhattan creates the possibility that the `Manhattan` network behaves more like the `highway` network than any other network, a possibility we will confirm in Section 4, where we study shared rides with many simultaneous requests. Also notice that `manhattan_uniform` and `Manhattan` have significantly different detour-value distributions. For example, $\mathbb{E}[r|v]$ is increasing in v for small values of v in `manhattan_uniform`. This illustrates that the distribution of the

Table 1. Metrics Obtained from Simulations

City	$\mathbb{P}(v > 0)$, %	$\mathbb{P}(r = 0 \mid v > 0)$, %	$\mathbb{E}[r \mid v > 0]/\mathbb{E}[c_u(i)]$	$\mathbb{E}[v \mid v > 0]/\mathbb{E}[c_u(i)]$
`grid_08`	21.0	49.6	0.255	0.471
`grid_08_oneway`	21.3	51.7	0.345	0.401
`grid_16`	22.2	29.2	0.304	0.415
`circle`	35.2	71.1	0.114	0.500
`highway`	8.2	19.5	0.775	0.631
`tree`	50.0	0.10	0.665	0.335
`manhattan`	13.0	0.17	0.409	0.322
`manhattan_uniform`	24.0	0.05	0.396	0.391

Note. From left to right in the table, we show the probability of match, the probability of having zero detour given a match, the expected detour, and the value given a match (normalized by the average trip length).

requests can significantly influence the relationship between value and detour.

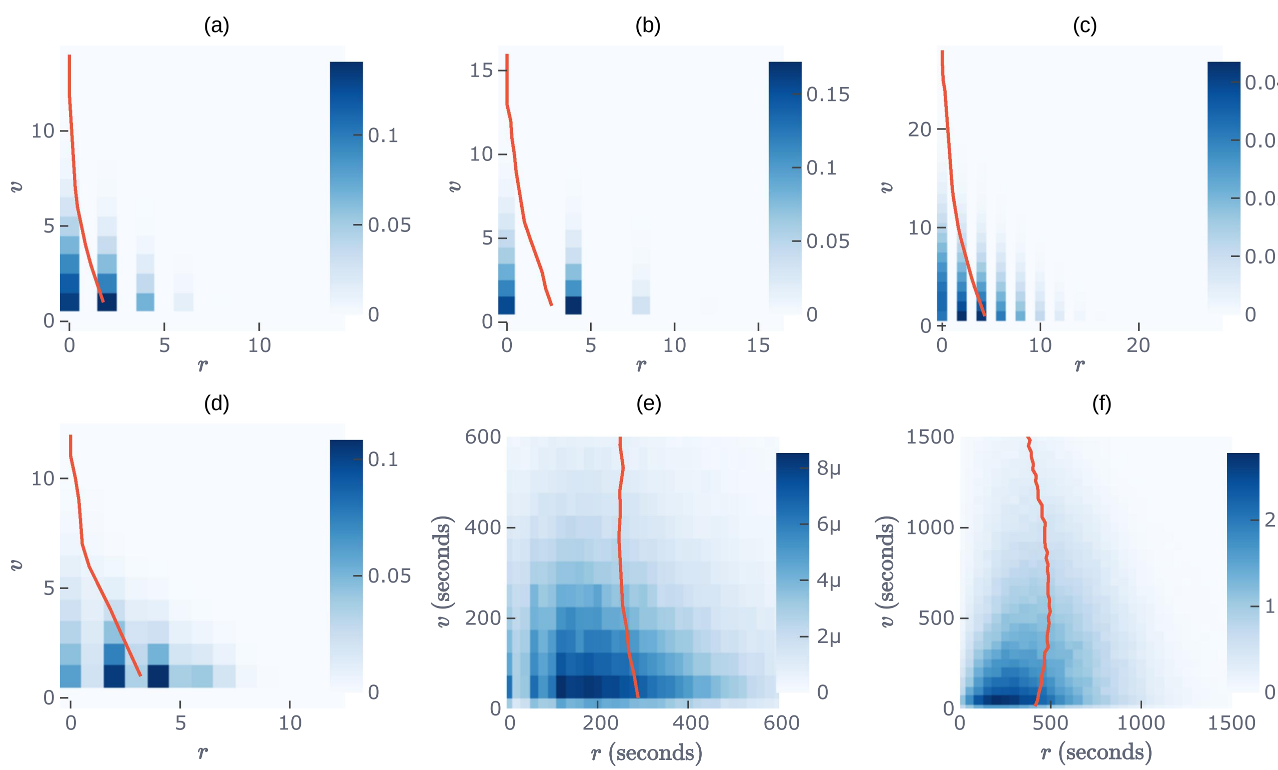
4. The Pareto Frontier of Value and Detour

Up until now, we have limited our study to the match of two random trips; we will now assume that more than two requests are generated probabilistically. Matching more trips will allow us to choose matches that increase the average value of a match and/or decrease its average detour. We will actually have the possibility to choose a trade-off between value and detour, and we will study the Pareto frontier of these two metrics. We will then come back to the study of our various city settings to derive some key results. (a) The average detour of a request is fairly low, regardless of the request density, and (b) we can reduce detours significantly with little impact on value. We will then recommend a new type of shared rides marketplace based on these findings.

4.1. General Properties of the Pareto Frontier

Let us assume that the set of n ride requests is drawn according to some probability distribution, with the origin-destination pairs (O_i, D_i) being independent and identically distributed. Let $\mathcal{M}$ be a matching algorithm that generates a nonnegative match given a set of ride

Figure 6. (Color online) Empirical Joint Distributions of Value and Detour in Our Test Networks Given That $v > 0$ (Positive-Value Matches)



Notes. These are generated from 10 million random pairs of trips. The vertical curves show our estimates of $\mathbb{E}[r \mid v]$, the expected detour given each match value. (a) `grid_08`. (b) `grid_08_oneway`. (c) `grid_16`. (d) `highway`. (e) `Manhattan`. (f) `manhattan_uniform`.

requests. We will use the notation $V_n(\mathcal{M})$ and $R_n(\mathcal{M})$ to denote the value and detour of a match M generated by $\mathcal{M}$, where the value and detour functions are defined in Equations (2) and (3) and the routing plan Π is assumed to be an optimal one (see Theorem 1). For clarity, we will also add the subscript n to the sum of n unmatched ride costs, $C_n(\emptyset)$. The n trips are randomly drawn, so $V_n(\mathcal{M})$, $R_n(\mathcal{M})$, and $C_n(\emptyset)$ are also random variables. An important metric throughout the paper will be the expectations of value and detour normalized by the expected sum of the unmatched costs. Formally, the value and detour ratios for n requests of a matching algorithm $\mathcal{M}$ are defined to be

$$\hat{V}_n(\mathcal{M}) = \frac{\mathbb{E}[V_n(\mathcal{M})]}{\mathbb{E}[C_n(\emptyset)]} \quad \text{and} \quad \hat{R}_n(\mathcal{M}) = \frac{\mathbb{E}[R_n(\mathcal{M})]}{\mathbb{E}[C_n(\emptyset)]}.$$

The matching problem is a multiobjective optimization problem, with each edge (i,j) in the matching graph having a corresponding detour $r(i,j)$ and value $v(i,j)$. Therefore, we can choose a match that maximizes value, or we could choose to also limit the amount of detour at the expense of a potentially suboptimal value. Generally, our goal is to understand the Pareto frontier of value and detour ratios for different density levels (different values of n). A point $(\hat{v},\hat{r})$ is in the Pareto frontier for n requests if there exists an algorithm $\mathcal{M}$ such that $\hat{V}_n(\mathcal{M}) = \hat{v}$ and $\hat{R}_n(\mathcal{M}) = \hat{r}$, and at the same time, there does not exist a matching algorithm $\mathcal{M}'$ such that $\hat{V}_n(\mathcal{M}') > \hat{v}$ and $\hat{R}_n(\mathcal{M}') \leq \hat{r}$ nor that $\hat{V}_n(\mathcal{M}') \geq \hat{v}$ and $\hat{R}_n(\mathcal{M}') < \hat{r}$. That is, we could only obtain more value from n rides if we sacrifice detour, and we could only obtain less detour if we sacrifice value. We will denote the Pareto frontier for n requests by the function $PF_n(\cdot)$, where $PF_n(\hat{r})$ represents the maximum value ratio obtainable with a detour ratio bounded above by $\hat{r}$. We denote by $\overline{R}_n$ the point in the n-requests Pareto frontier with maximum detour (i.e., the lowest value $\hat{r}$ such that $PF_n(\hat{r})$ is maximized). The next theorem establishes several properties of these Pareto frontiers.

Theorem 3. *The following statements are true about the Pareto frontiers of value and detour.*

1. *For any n, the Pareto frontier for n requests is bounded by $0 \leq PF_n(\hat{r}) \leq \frac{1}{2} - \hat{r}$ for all $\hat{r} \in [0,\overline{R}_n]$.*
2. *For any even n and any pair $(\hat{v},\hat{r})$ in $\{(\hat{v},\hat{r}) \in \mathbb{R}^2 \mid \hat{v}+\hat{r} \leq 1/2,\ v \geq 0,\ r \geq 0\}$, there exists a city topology $(\mathcal{C},d)$ such that $PF_n(\hat{r}) = \hat{v}$.*
3. *For any n, $PF_n(\cdot)$ is a nondecreasing concave function.*
4. *If we double the density of requests, the Pareto frontier moves up (i.e., $PF_{2n}(\hat{r}) \geq PF_n(\hat{r})$ for all $n \in \mathbb{N}$ and $\hat{r} \in [0,\overline{R}_n]$).*
5. *The following asymptotic limit holds:*

$$\lim_{n\to\infty} \sup_{\hat{r}\in[0,\overline{R}_n]} PF_n(\hat{r}) = \frac{1}{2}.$$

We now discuss the interpretation of the different parts of Theorem 3. The first part of Theorem 3 takes the key result from Theorem 2 that there is a constraint on the sum of value and detour of a matched pair of requests, and turns it into a statement about the sum of the value and detour of an entire match. That is, the Pareto frontier of value and detour ratios is bounded by the same triangle that matched pairs are, with the key feasibility constraint being $\hat{V}_n(\mathcal{M}) + \hat{R}_n(\mathcal{M}) \leq 1/2$ for any n and any matching algorithm $\mathcal{M}$.

The next statement of the theorem says that the Pareto frontier bound from part (1) of Theorem 3 is tight in the sense that for any even n and any feasible value-detour ratio pair $(\hat{v},\hat{r})$, we can construct a network that induces a value ratio of $\hat{v}$ and a detour ratio of $\hat{r}$. Thus, no better bounds are possible without specializing to specific city topologies.

The third point of the theorem is about the shape of the Pareto frontier. The nondecreasing concavity of the frontier is useful because it implies that the Pareto frontier can be computed by solving a sequence of parameterized objective functions. That is, by solving

$$\max_{M} \ \hat{V}_n(M) - \alpha\hat{R}_n(M) \tag{7}$$

for different nonnegative values of α, we can compute all of the points on the Pareto frontier.

The fourth part of the theorem says that doubling the density improves the Pareto frontier. Because of this result, when we use simulations to compute Pareto frontiers, we will consider how the frontier changes as the density doubles. One might suspect that the Pareto frontier is always nondecreasing in n without the need for doubling. However, this is not true. Consider the following simple counterexample; there exist only two locations O and D, and all requests are from O to D. With $n = 2$, we can match the requests perfectly, obtaining a value ratio equal to $1/2$. With $n = 3$, one of the requests is left unmatched, leading to the lower value ratio of $1/3$.

Finally, the fifth statement shows that in the limit as n grows large, the value ratio tends toward its theoretical upper bound of $1/2$. Combined with part (1) of the theorem, this also implies that detours become small as n grows. The proof of this statement is based on a simple idea, which is that as the number of ride requests increases, a very simple matching algorithm becomes very appealing: only match two requests if they are nearly identical. This algorithm becomes approximately optimal in the limit as the number of requests grows. The key assumption driving this part of the theorem is the compactness of the city, which allows us to match most requests to nearly identical requests in the limit as n goes to infinity.

This result settles what happens in the asymptotic limit as n grows, but it leaves the more interesting transient (lower-density) results fairly open. It establishes a bound on the Pareto frontier and determines that doubling density leads to an improvement in the Pareto frontier. However, the shape of the frontier and how precisely it is affected by the density are questions we study in the next subsections, where we focus on specific city topologies in order to be able to dive deeper.

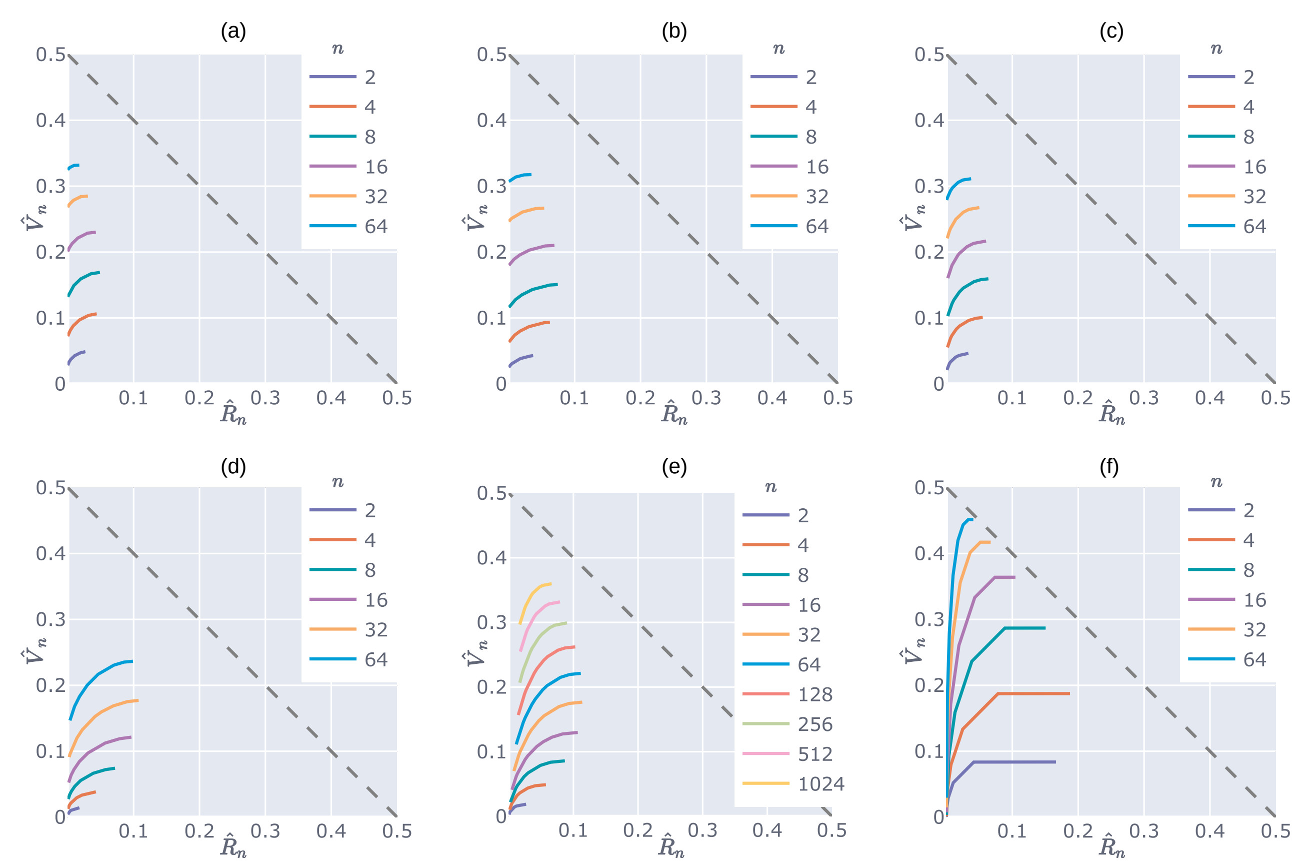
4.2. Obtaining Pareto Frontiers

For a general network topology, obtaining closed form of the Pareto frontiers is an impossible task because most optimal matching solutions do not have a closed-form characterization. We can compute optimal matches for general networks in a computationally efficient way, but this is not sufficient for obtaining closed-form Pareto frontiers. Nonetheless, in the specific case of the tree city, we are able to overcome this challenge and derive a closed form of the Pareto frontier in Online Appendix A.2.2. In this particular case, illustrated in Figure 7(f), the maximum value ratio increases quickly with the density of requests, and the Pareto frontier is very flat, which means that a lot of detour can be saved with minimal consequences on the platform's value. We now turn to simulation in order to obtain Pareto frontiers for a wider class of networks.

In Section 3.3, we analyzed value-detour joint distributions for pairs of ride requests for a variety of networks, including the synthetic networks displayed in Figure 4 and the `Manhattan` network from Figure 5. We now compute Pareto frontiers with n ride requests for all of these networks using simulation. To generate a sample of the Pareto frontier with n requests, we first generate n random requests and then solve the optimal matching problem described in Equation (7) for different values of $\alpha \geq 0$. We repeat this process each time sampling new requests to estimate the expected value and detour ratios $\hat{V}_n$ and $\hat{R}_n$. Specifically, the curves plotted in Figure 7 use 2^{18} ride requests for each point in the Pareto frontier (2^{14} for the `Manhattan` network as matching with larger networks is more computationally challenging). We also compute the Pareto frontier for the `tree` network, but this one is done via explicit computation using the results from Proposition 6 in Online Appendix A.

It is worth revisiting Theorem 3 when analyzing Figure 7. Part (1) of the theorem says that all of the Pareto frontiers must live under the dashed lines. Part (3) says the Pareto frontiers are nondecreasing concave functions. Part (4) says that when we double density, the Pareto frontiers move up. Part (5) says that in the limit as n grows large, the top of the Pareto frontier moves toward the value ratio of 1/2. Combining parts (1) and

Figure 7. (Color online) Pareto Frontiers of Detour and Value for Various Networks Studied in This Paper



Notes. The tree setting is obtained from Proposition 6 in Online Appendix A, whereas the others are evaluated using simulations. (a) `grid_08`. (b) `grid_08_oneway`. (c) `grid_16`. (d) `highway`. (e) `Manhattan`. (f) `tree`.

(5), we have that the maximum detour ratio $\overline{R}_n$ approaches zero as the value ratio grows to $1/2$. These effects can be easily seen in Figure 7.

At the same time, Theorem 3 says little about what happens for small n's. As can be seen in Figure 7, for small values of n, doubling the density has a significant impact on values and detours. Consider, for example, the tree network. The maximum value ratio more than doubles when we move from $n=2$ to $n=4$. At $n=32$, we already obtain value ratios of over 0.4, which is close to the theoretical optimum of $1/2$. Note that this is a network with $2^{10}=1024$ nodes, so the probability of request to a particular destination is only $1-(1{,}023/1{,}024)^{32}\approx 3.1\%$. That is, the asymptotic limit of $1/2$ for the value ratio (Theorem 3, part (5)) is a reasonable approximation of the performance of the shared service, even for a relatively low density of 3.1% of destination requests.

The Pareto frontiers for the `Manhattan` and `highway` networks are fairly similar, despite the fact that the `Manhattan` network is known to be a grid. That is because the `Manhattan` network actually deviates from a grid in several ways, including the presence of diagonal Broadway and highways along the rivers on both sides of Manhattan. Despite the `Manhattan` network having potentially high detours for a random pair of rides (see Table 1), the detour ratio in the `Manhattan` network typically stays under 10%. This implies that it is possible run a shared rides service in the `Manhattan` network with low expected detours, despite what the earlier pairwise result might suggest. The reason for these low expected detours is different under high- and low-density cases. In a high-density case, the detours are low because the matches are high-value ones (and such matches by Theorem 2 are low-detour ones). In a low-density case, the expected detour is low because most rides are unmatched, and solo rides have zero detour. This effect can be seen in all the studied networks, and it explains why shared rides can be attractive, regardless of the factors influencing the level of demand, such as the choice of platform or the time of day. Indeed, a passenger has either a high probability of not being matched and having a solo ride if the demand is low (and sometimes have a trip with nonnegligible detour, although still capped at 50%) or a high probability of match with low detour if the demand is high enough. What this study shows is that the two effects balance each other in realistic networks, and the average detour stays consistently low, regardless of the demand density.

Meanwhile, the three Pareto frontiers for the grid networks are qualitatively quite similar. A few of the network features that have the potential to have detour implications, including one-way streets and larger grids, do not have a major impact on the shape of the Pareto frontiers. In fact, the Pareto frontiers plotted in Figure 7, (a)–(c) satisfy the following relation: $PF_{2n}(0)\geq PF_n(\overline{R}_n)$. That is, doubling the density is more important for the efficiency of the shared rides platform than allowing detours at all! This raises a tantalizing option for ride-sharing platforms; instead of trying to generate value by allowing more detour, they could potentially create value by reducing detours instead. Section 7 will explore how this idea could potentially lead to a new design for shared rides systems. First, it is important to make sure that we limit detours in a fair way; we explore request-level detours and how to limit them in the next section.

5. Request-Level Detours

The *average* detour of shared rides has been the focus of the paper so far. However, even with a low average detour, it is possible that these detours are distributed in an unfair way. Consider the example of an $O_iO_jD_jD_i$ match; request i may experience detours, but request j is guaranteed to have zero detour. In this section, we first show that there are limits to request-level detours, and then, we argue that the platform can limit the worst cases without large effects on value.

5.1. The Request-Level View of Values and Detours

Given two requests i and j, we have introduced the aggregated detour $r(i,j)$ of the match, but we did not study how this aggregated detour is shared between the two requests. We now switch views and focus on a specific ride request i. We abstract away from the specific ride request it is matched with, focusing only on the elements that matter for ride request i. In particular, we want to understand the relationship between the detour of this particular passenger and the value of the platform. Let us consider a match M that contains a matched ride request i. We will introduce the notation $d_M(i)$ to represent the distance traveled by rider i under match M. Similarly, we will use notation $s_M(i)$ to represent the shared component of rider i's trip. That is, $s_M(i)$ represents the distance traveled with rider i with someone else in the car. With these objects in hand, we can introduce the notion of the value of a ride request i under match M:

$$v_M(i)\triangleq c_u(i)-\left(d_M(i)-\frac{s_M(i)}{2}\right). \tag{8}$$

In words, the value of a request i in a match M corresponds to the savings generated by matching rider i. By matching i, we save the unmatched cost $c_u(i)$, but we incur a cost of $d_M(i)-s_M(i)/2$, where the division by two corresponds to splitting the driving cost of the shared component between the two riders in the car. The detour of request i in match M is given by the difference between the distance traveled under the match and the unmatched distance:

$$r_M(i)\triangleq d_M(i)-c_u(i). \tag{9}$$

If a request i is unmatched in M, then we have $v_M(i)=r_M(i)=s_M(i)=0$ and $d_M(i)=c_u(i)$. Similarly to the previous section, we also define the ratios $\hat{v}_M(i)=v_M(i)/c_u(i)$,

$\hat{r}_M(i) = r_M(i)/c_u(i)$, and $\hat{s}_M(i) = s_M(i)/c_u(i)$. We now show that these objects are consistent with the objects defined for pairs of requests as well as prove two properties of these objects, including a request-level analogue of Equation (4).

Proposition 1. *Consider a match M and a pair of requests $(i,j) \in M$. The value, detour, and shared component defined at the request level are consistent with the terms as defined for pairs: $s(i,j) = s_M(i) = s_M(j)$, $v(i,j) = v_M(i) + v_M(j)$, and $r(i,j) = r_M(i) + r_M(j)$. Furthermore, we also have $v_M(i) + r_M(i) = s_M(i)/2$ and $s_M(i) \le c_u(i)$.*

We are now ready to state Theorem 4, which characterizes the set of feasible request-level value-detour ratio pairs. We say a request-level value-detour ratio $(\hat{v},\hat{r})$ is feasible if there exists a city $(\mathcal{C},d)$, a request i, and a nonnegative match M such that $\hat{v}_M(i) = \hat{v}$ and $\hat{r}_M(i) = \hat{r}$. It is analogous to Theorem 2, but it applies to request-level objects.

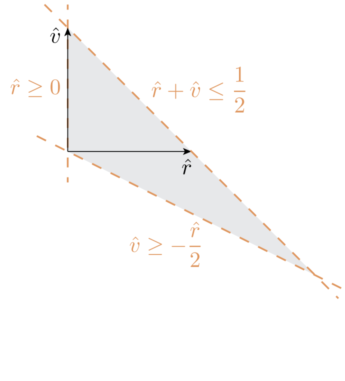
Theorem 4. *Let the set $\mathcal{S}_1$ be defined as*

$$\mathcal{S}_1 = \left\{ (\hat{v},\hat{r}) \in \mathbb{R}^2 \,\middle|\, \hat{v} + \hat{r} \le \frac{1}{2},\ \hat{r} \ge 0,\ \hat{v} \ge -\frac{\hat{r}}{2} \right\}.$$

Then, a request-level value-detour ratio pair $(\hat{v},\hat{r})$ is feasible if and only if it belongs to $\mathcal{S}_1$.

Theorem 4 establishes that the set of feasible value-detour ratios forms a downward-sloping triangle (Figure 8). It is perhaps counterintuitive that value ratios can be negative. A negative request value is possible in a nonnegative match because a request could be matched to a positive request value, yielding a pair with a nonnegative sum of value. Nonetheless, this would typically be a low-value match, and it would only be likely to happen if the demand density is low and no good matches are available. The individual detours can never be above 100% ($\hat{r}_M(i) \le 1$), which means that passengers will spend twice as long in the car in the worst case (whereas the average detour can never be more than 50%). A 100% detour may seem unfair when the average detours are typically less than 10%, regardless of the demand density (as shown in Figure 7). This is why the platform may want to limit these extreme cases as explored next.

Figure 8. (Color online) The Set $\mathcal{S}_1$ of Feasible Request-Level Value-Detour Ratios as Characterized by Theorem 4



5.2. Limiting Request-Level Detours

Suppose that the platform decides to limit the individual detours and imposes the constraint $\hat{r}_M(i) \le \bar{r}$ for all requests i for a given constant $0 \le \bar{r} \le 1$. That is, given the requests i, j, we do not allow any route that would lead to $\hat{r}_M(i) > \bar{r}$ or $\hat{r}_M(j) > \bar{r}$ should i and j be matched in M.

Theorem 5. *Under request-level detour constraints with $\bar{r} \ge 0.5$, Theorems 2 and 3 still apply. If $\bar{r} < 0.5$, Theorem 2 is still valid, except that the feasible detour ratios have to be less than $\bar{r}$:*

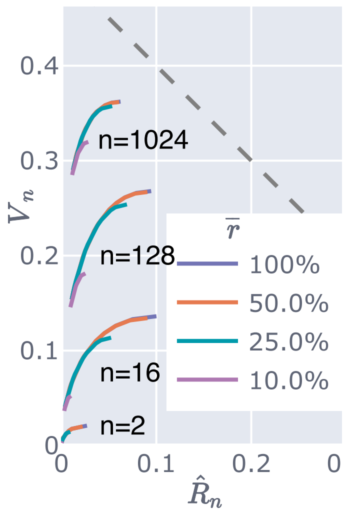
$$\mathcal{S}_2 = \left\{ (\hat{v},\hat{r}) \in \mathbb{R}^2 \,\middle|\, \hat{v} + \hat{r} \le \frac{1}{2},\ 0 \le \hat{r} \le \bar{r},\ 0 \le \hat{v} \right\}.$$

Similarly, Theorem 3 is also still valid, with the simple change that the Pareto frontiers now live in the same triangle $\{(\hat{v},\hat{r}) \in \mathbb{R}^2 | \hat{v} + \hat{r} \le \frac{1}{2},\ 0 \le \hat{r} \le \bar{r},\ 0 \le \hat{v}\}$.

Theorem 5 shows that adding individual detour constraints does not drastically change the relationship between value and detour. The shape of the feasible region is just changed with the addition of the $\hat{r} \le \bar{r}$ constraint. It is easy to see why we need to add this constraint as an $\bar{r}$ request-level detour constraint implies an $\bar{r}$ average detour constraint. However, what is less obvious is that this condition is the only change to our previous results.

Nonetheless, even if our general results still apply, adding individual detour constraints will still lower the value that we can achieve for a given amount of detour (i.e., decreasing $PF_n(\hat{r})$). However, Section 5.1 and Theorem 5 suggest that adding reasonable individual detours constraints will only remove low-value matches and therefore, will not reduce $PF_n(\hat{r})$ too much. As this effect depends heavily on the network topology, we verify this hypothesis in our `Manhattan` real-data simulation. We simulated this same matching setting, adding the constraint $\bar{r} \in \{0.1, 0.25, 0.5, 1\}$, as shown in Figure 9. $\bar{r} = 1$ corresponds to the case without individual detour requests. As expected, adding request-level detour constraints reduces the Pareto frontier. In the `Manhattan` network, this reduction leaves the new Pareto frontier close to the original one (e.g., the new curve appears to be almost "on top" of the old one, even if it is shorter). This means that adding individual detour constraints does not reduce the value significantly more than increasing the detour penalty α. This is especially true for higher values of α (the lower left side of the Pareto frontiers), where the high detour penalty already removes most matches with high individual detours, which means that the constraint $\bar{r}$ does not change much. An interesting fact is the case $\bar{r} = 0.5$. As discussed previously, this constraint only removes the most unfair individual detours (which can go up to $\hat{r} =$ 100% otherwise). Additionally, Figure 9 shows that in Manhattan, adding this constraint barely changes the

Figure 9. (Color online) Relationship Between Value and Detour Ratio in Manhattan with Request-Level Detour Constraints



Notes. The color of the curve identifies the maximum request-level detour ratio $\bar{r}$. $\bar{r}$ = 100% is unconstrained.

maximum achievable value, especially in the high-density case; for $n = 1{,}024$, we never lose more than 0.1% of the value when imposing $\bar{r} = 0.5$. This suggests that a platform should consider limiting the worst-case individual detours instead of just focusing on the total detours in the market. In contrast, adding a very tight detour constraint, such as $\bar{r} = 0.1$, does significantly impact the Pareto frontier, but it is not very different from increasing the detour penalty α (restricting yourself to the lower/left side of the Pareto frontier).

6. Incorporating Pickup and Drop-off Times

So far, we assumed that the time it takes for a driver to pick up and drop off a rider was negligible. However, this may not be true in practice as extra time is needed for the car to park and find a waiting rider. A rider waiting in a car while another rider is being picked up or dropped off also incurs a "detour" in the sense that the additional pickup/drop-off time will delay them. This section adds pickup and drop-off times to our model and simulations. We show that our main insights still generally hold but that the larger the added fixed times, the worse the Pareto frontier between value and detour becomes.

We add fixed pickup and drop-off costs $c_p, c_d \geq 0$ to our model such that the new unmatched cost of a ride $c_u^{pd}(i) \triangleq c_u(i) + c_p + c_d = c_p + d(O_i, D_i) + c_d$ (we use the *pd* notation to distinguish this section's notation from the main model's notation). For example, if $d(O_i, D_i) =$ 30 minutes, $c_p = 3$ minutes, and $c_d = 1$ minutes, then the total time it takes to serve rider i is 34 minutes. Similarly, for any route $\pi \in \{1, \ldots, 4\}$, the cost of serving two matched requests is $c_m^{pd}(i,j,\pi) \triangleq 2c_p + 2c_d + c_m(i,j,\pi)$ as we need to pick up and drop off the two riders. The value of a match is unchanged as sharing does not save on pickup and drop-off times: $v^{pd}(i,j,\pi) \triangleq c_u^{pd}(i) + c_u^{pd}(j) - c_m^{pd}(i,j,\pi) = v(i,j,\pi)$. For the detour, we have $r^{pd}(i,j,\pi) \triangleq r(i,j,\pi) + c_p + c_d$. For example, for route $\pi = 1$ $(O_i \to O_j \to D_i \to D_j)$, rider i has an added c_p detour to pick up j, and j has an added c_d detour to drop off i. The total added detour is $c_p + c_d$. For $\pi = 2$ $(O_i \to O_j \to D_j \to D_i)$, j has no added detour, but i has an added $p_u + p_d$ detour to pick up and drop off j—still $c_p + c_d$ in total. Finally, we define the "shared" component of a match as $s^{pd}(i,j,\pi) \triangleq s(i,j,\pi) + c_p + c_d$ because exactly one pickup and one drop-off must be made with a rider already in the car for any choice of route π. All other notations of the original model are functions of the quantities that we already redefined, and their new definition is immediate. For the sake of brevity, we assume that they have been defined accordingly (e.g., $\overline{R}_n^{dp}(\mathcal{M})$ is well defined and so on).

This new formulation is equivalent to the original model, with an added fixed detour for each match; the value of a match is unchanged, and $c_p + c_d$ is added to all match detours. As a consequence, the equivalent of Theorem 1 is immediately true (the proof is unchanged), and the proof of Theorem 3, parts (1), (3), and (4) still directly applies without changes. In particular, this means that the Pareto frontier PF_n^{pd} is still contained in the triangle $\{(\hat{v}, \hat{r}) \in \mathbb{R}^2 | \hat{v} + \hat{r} \leq 1/2,\ v \geq 0,\ r \geq 0\}$. However, part (5) of the theorem is no longer true, and the expected value ratio cannot be arbitrarily close to 1/2 as we increase density as shown.

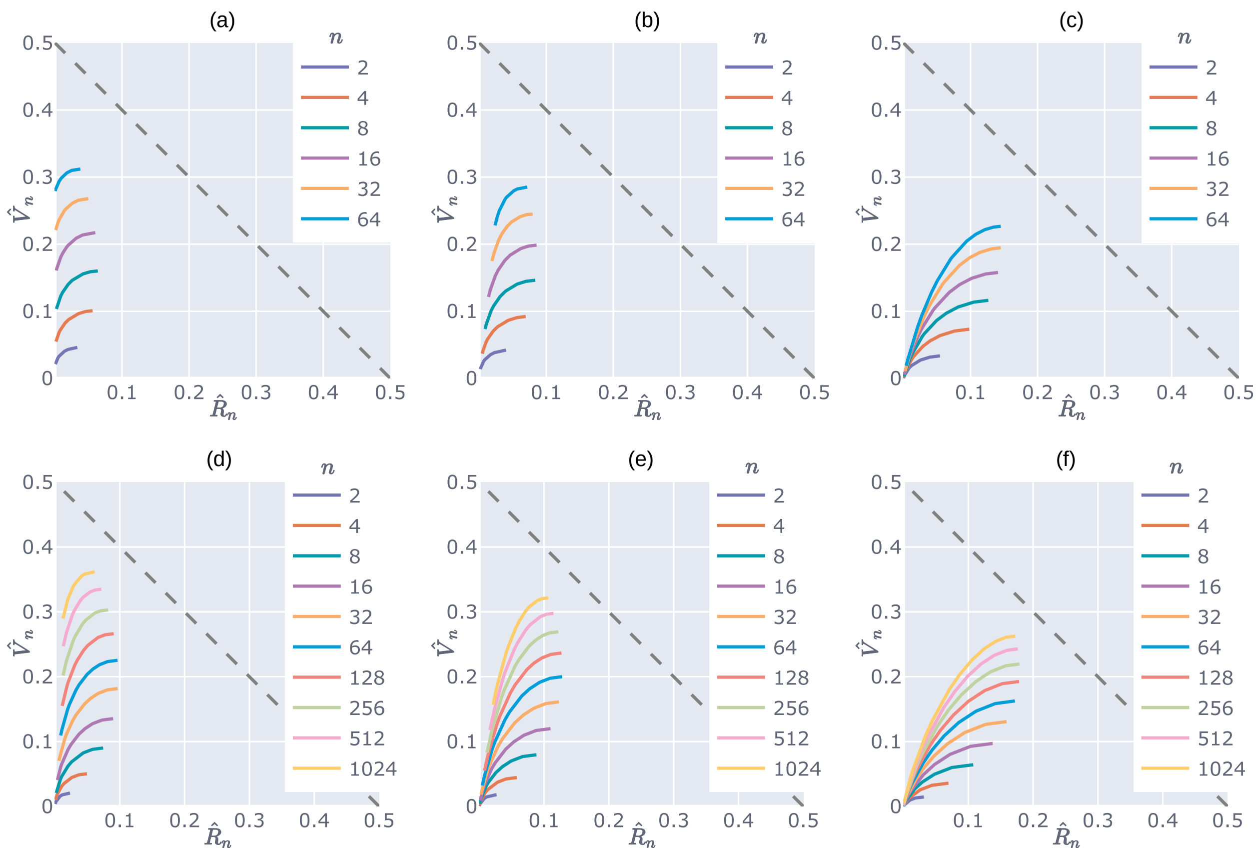
Proposition 2. *The following asymptotic limit holds*:

$$\lim_{n\to\infty} \sup_{\hat{r}\in[0,\overline{R}_n^{pd}]} PF_n^{pd}(\hat{r}) = \frac{1}{2\left(1 + \frac{c_p + c_d}{\mathbb{E}[d(O,D)]}\right)}.$$

This proposition illustrates that pickup and drop-off costs limit the value of sharing rides as c_p, c_d are always incurred, whether we match rides or not. Therefore, if $c_p + c_d$ is not negligible compared with the average length of a ride $\mathbb{E}[d(O,D)]$, the maximum expected value ratio will be less than 1/2.

Figure 10 clearly shows how long pickup or drop-off times can hurt shared rides. First, we can see that the value ratios are reduced (even without a detour penalty). This is because the value ratios are defined as $\frac{\mathbb{E}[V_n^{pd}(\mathcal{M})]}{\mathbb{E}[C_n^{pd}(\emptyset)]}$. With added pickup and drop-off times, we do not add value (e.g., the numerator) but make the rides longer (e.g., the denominator); this effect is similar to the result of Proposition 2. Second, the Pareto frontiers are "flattened" and have higher detours. This is because with added pickup and drop-off times, we add a

Figure 10. (Color online) Pareto Frontiers of Detour and Value with Pickup and Drop-off Costs



Notes. In `Manhattan`, we consider the cases of 0, 2, and 6 minutes of pickup + drop-off time (for an average trip length of 16 minutes). In `grid_16`, we consider 0, 1, and 4 time steps of pickup + drop-off time (for an average trip length of 10.6 time steps). All simulations have a maximum allowed individual detour ratio of 50%. (a) `grid_16` $\frac{c_p+c_d}{\mathbb{E}[d(O,D)]} = 0\%$. (b) `grid_16` $\frac{c_p+c_d}{\mathbb{E}[d(O,D)]} = 9.5\%$. (c) `grid_16` $\frac{c_p+c_d}{\mathbb{E}[d(O,D)]} = 37.5\%$. (d) `Manhattan` $\frac{c_p+c_d}{\mathbb{E}[d(O,D)]} = 0\%$. (e) `Manhattan` $\frac{c_p+c_d}{\mathbb{E}[d(O,D)]} = 12.5\%$. (f) `Manhattan` $\frac{c_p+c_d}{\mathbb{E}[d(O,D)]} = 37.5\%$.

constant to the matched detours, which in turn, increases detours in the Pareto frontiers.

7. A Two-Product Approach to Detours

Up until now, we have explored Pareto frontiers of value and detour for different density levels. This analysis could be interpreted as hinting that the shared rides platform needs to choose a point in the Pareto frontier where it would like to operate. In reality, there are other ways a shared rides platform could operate. One potentially interesting approach would be to separate customers who are very sensitive to price from customers who are very sensitive to detours. A platform could do this by having two products: a shared basic product with a loose detour constraint alongside a shared premium product with a tighter detour constraint.

Having two shared products could allow the platform to put to use the heterogeneity of the customer base. The two products would be highly complementary as the shared premium users would see little detour but could subsidize some of the cost of the shared basic users. Depending on the underlying city network, one could even envision a very-small-detour shared premium product where premium users essentially "lay roads" that the basic users could use. This product would be quite valuable if there is a sizable population of potential users who do not mind sharing a vehicle but do mind detours. In this case, we could potentially generate substantial density gains by offering this low-detour product that could more than offset the value loss because of tighter detour constraints.

To understand the value of the proposed product differentiation, let us build a simple economic model to capture the basic trade-offs faced by users. We model a customer's disutility for a shared ride as

$$\text{Disutility} = \text{Payment} + \frac{\theta}{1-\theta} \cdot (\alpha \cdot r + \beta). \tag{10}$$

This is a parametric utility model, where the first term is the cost of the ride and where the second term is the disutility of sharing the car. The parameter α is the disutility of detour, and it is multiplied by a term r, which captures the amount of detour that the user faces. The term r is normalized to be equal to one in the status quo: a shared user in a platform without differentiated shared products. The parameter β is the disutility of all other features of shared rides, such as having another person in the car and having potential pickups/drop-offs for them. Finally, each customer is associated with a parameter $\theta \in [0, 1]$ such that $\theta = 0$ means they care only about monetary costs and that $\theta = 1$ means they care only about the inconvenience of shared products. θ is assumed to be distributed uniformly on $[0, 1]$.

We will use data from a U.S. Department of Transportation study to calibrate α and β. The study (Middleton et al. 2021) considers ride-sharing data from 15 cities and also analyzes the results of a survey where users are asked about hypothetical trade-offs between money and time in ride-sharing. In their data, which are from late 2018, they find that 29.9% of ride requests are shared requests, and it can be inferred from Middleton et al. (2021, figure 4) and the discussion surrounding it that average prices are \$14.97 for solo rides and \$9.73 for shared rides. We can use this to begin to calibrate α and β from Equation (10). The marginal user is the one who is indifferent between shared and solo, and this user has $\theta = 0.299$. Therefore,

$$\$14.97 = \$9.73 + \frac{0.299}{0.701}(\alpha + \beta),$$

where we ignore r because it is assumed to be one under current detour conditions. Thus, $\alpha + \beta = \$12.29$.

At the same time, Middleton et al. (2021, figure 4) also say that 33% of users would not accept to share a ride even with a 75% discount and no delays or detours. That is, at a price of $\$14.97/4 = \$3.74$ and without detours ($r = 0$), the user with $\theta = 0.67$ is now the marginal user, being indifferent between shared and solo rides:

$$\$14.97 = \frac{\$14.97}{4} + \frac{0.67}{0.33} \cdot \beta,$$

implying that $\beta = \$5.53$ and thus, that $\alpha = \$6.76$.

Suppose the company introduces a new shared premium product without detours while keeping the prices of the previous two products constant. Let us make the very conservative assumption that no new customers join the platform because of this product (the results could only be stronger if we assumed that new customers are attracted by the new product). Let us say the product is priced to increase the shared market by 25% via cannibalization of the solo market. To achieve this, we need the marginal user deciding between shared premium and solo to be at $\theta = 1.25 \cdot 0.299 = 0.374$. Because this product has no detour, this is achieved at price \$11.67 because

$$\$14.97 = \$11.67 + \frac{0.374}{1 - 0.374} \cdot \$5.53.$$

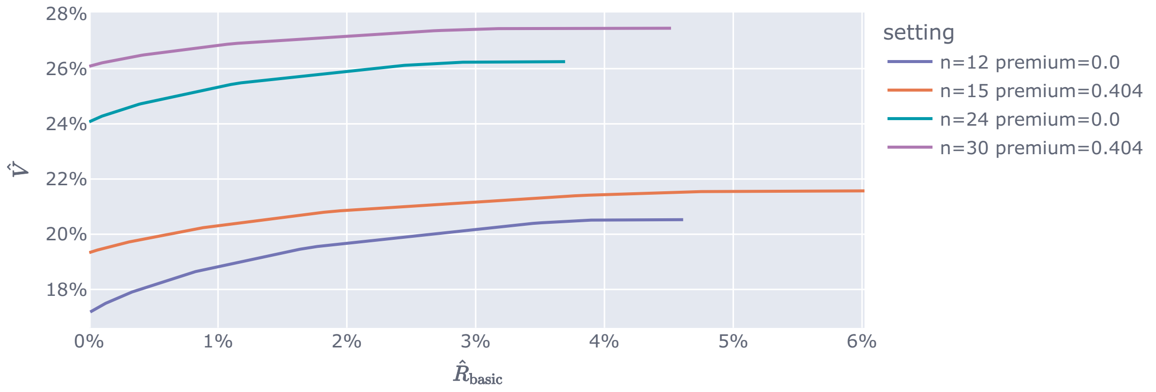
We also have to compute what fraction of the users would use the shared basic product in the presence of shared premium. The marginal user's θ is

$$\$11.67 + \frac{\theta}{1 - \theta} \cdot \$5.53 = \$9.73 + \frac{\theta}{1 - \theta}(\$6.76 \cdot r + \$5.53).$$

If we are able to maintain detour levels of shared basic users at the same levels that they faced with a single shared product, then we could assume $r = 1$, which leads to the marginal θ being 0.223. Therefore, under this proposal, the shared rides market grows by 25% in size, and these requests are 59.6% (0.223/0.374) basic and 40.4% premium.

Is it possible for a platform to improve its profits by adding shared premium at this price? The answer will depend on the specific network being considered, and this is where we can use our detour simulations to help our analysis. In Figure 11, we show two different density scenarios ($n = 12$ and $n = 24$) in an 8×8 grid, and we show that adding 25% more demand while enforcing zero detour for 40.4% of the requests at the same time strictly improves the Pareto frontiers. Whether a

Figure 11. (Color online) Pareto Frontiers in the 8×8 Grid Under Two Pairs of Scenarios: $n = 12$ and $n = 24$ with All Customers Being Shared Basic Customers and $n = 15$ and $n = 30$ with 40.4% of the Customers Being Shared Premium Customers



Note. The Pareto frontiers for $n = 15$ and $n = 30$, respectively, dominate the ones for $n = 12$ and $n = 24$, despite the fact that 40.4% of the customers in the former curves are constrained not to have detours.

customer requests shared basic or shared premium is generated at random. The detours plotted exclude the zero detour of the shared premium riders (they are detours of basic customers).

To calculate the profit impact of the proposal, we need to compute both the revenue and cost impacts. Because we are considering only cannibalization of current solo riders, the revenue per request naturally goes down: from $\$14.97 \cdot 0.701 + \$9.73 \cdot 0.299 = \$13.40$ to $\$14.97 \cdot 0.626 + \$11.67 \cdot 0.151 + \$9.73 \cdot 0.223 = \$13.30$. If new customers were to enter the system because they like the new shared premium option, we naturally could recover the old profit per ride (for instance, if the mass of shared premium customers was 0.160 instead of 0.151, we would be at revenue parity). However, let us be conservative, and let us assume that no new customers have entered the system and that we are considering cannibalization effects only; therefore, we have a revenue loss.

Now, let us look at costs. To model the driver pay rules of realistic platforms, we assume that driver pay costs 75% of the price for solo rides (a commission rate of 25%), whereas the shared rides drivers are paid for their time at the same rate as if they had served solo rides. Let us say we are initially operating according to the $n = 24$ curve in Figure 11 at the $\hat{R} = 2\%$ point, and therefore, we are obtaining value equal to 26.01%. Then, we pay the solo ride drivers an average of $\$14.97 \cdot 0.75 = \$11.228$ per request, whereas the shared requests drivers are paid an average of $\$14.97 \cdot (1 - 0.2601) \cdot 0.75 = \$8.307$. To obtain the last expression, we multiplied the solo driver pay by one minus the value ratio that represents the fraction of driver time we save via matches. The average driver cost per request is, therefore, $\$11.228 \cdot 0.701 + \$8.307 \cdot 0.299 = \$10.35$. Figure 11 says that the new system with two shared products is more efficient, operating with value equal to 27.54% with $\hat{R}_{basic} = 2\%$ (basic detours are the same length as detours in the original system). We use the same approach to compute the driver costs of the new system: $\$14.97 \cdot 0.75 \cdot 0.626 + \$14.97 \cdot (1 - 0.27) \cdot 0.75 \cdot 0.374 = \$10.07$. Therefore, despite the revenue decrease, profit per ride in the new system goes from $\$13.40 - \$10.35 = \$3.05$ to $\$13.30 - \$10.07 = \$3.23$, a 6% gain.

Like us, Middleton et al. (2021) also propose that platforms should consider adding a premium shared rides service with a more direct route (see Middleton et al. 2021, p. 39). Even though we currently use their survey data in our economic analysis of a two-product system, our shared premium proposal precedes the release of Middleton et al. (2021). We find it encouraging that Middleton et al. (2021) independently reached the same conclusion regarding the desirability of a multiproduct shared system.

To be clear, the analysis in this section does make some simplifying assumptions, such as the uniform distribution of θ, but our main point is not the precise counterfactual estimates. Instead, it is to highlight that the findings of our paper (density is more important than detours) strongly suggest that ride-sharing companies should consider a more complex product portfolio.

8. Ignoring Drivers

Up until this point, we have ignored the issue of limited availability of drivers, and we have assumed that pickup times were zero because drivers were plentiful. This situation is sometimes approximately true but not always. In this section, we drop this assumption in order to understand how having limited drivers affects our results.

We will prove in this section that our results are robust to having limited drivers under certain assumptions. To prove this, we make a list of assumptions inspired by industry practice. (1) Drivers are paid only for the time they are transporting customers, not while driving to a pickup location. (2) Platform commissions are at most 25% of total ride costs.[1] (3) Driver utilization (the fraction of time a driver has a rider in the car) is rarely above 75% because of frictions, such as having to wait for requests and pickup times. (4) On average, pickup times are generally less than 30% of trip (in-ride) times (especially in shared rides, where drivers can serve several requests in a row without in-between pickup time). (5) The pickup time inefficiency caused by matching requests without regard to vehicle locations cannot increase pickup times by more than 50%. These assumptions may not always be verified in practice. However, in what follows, we will both justify why we believe that these assumptions are often realistic and use assumptions (2)–(5) in a way that is easy for readers to test how our bounds change if the numbers we propose are different than what they believe to be true.

Consider the following model of an online matching problem where the platform tries to minimize its driver costs associated with serving some exogenous shared rides demand. Let T_{trip} be the expected number of driver hours that will be used to serve demand (with passengers in the car) during a given week, and let T_{pickup} be the corresponding expected driver hours during the same week spent doing pickups (without passengers in the car and driving toward a pickup location). The platform can choose any online matching policy that serves the overall demand, and any policy can be mapped to a choice $(T_{\text{trip}}, T_{\text{pickup}}) \in \mathcal{P}$, with $\mathcal{P}$ being a compact set that defines the feasible trade-offs between T_{trip} and T_{pickup}.

The platform pays the drivers a cost c for each hour of driving time with passengers, for a total cost of cT_{trip}. Assumption (1) says that there are no direct costs for pickup time, so there is no cT_{pickup} cost term. However, there is also an opportunity cost of driver time; saving driver time means that we could potentially serve extra requests (accompanied by pricing changes). Reducing pickup and trip time would also improve the service levels because of lower wait and/or detours, also increasing demand. We will capture this opportunity cost by a parameter $\gamma \geq 0$; we assume that it is the same for drivers doing pickup as well as transporting passengers, so it is equal to $\gamma(T_{\text{trip}} + T_{\text{pickup}})$. Intuitively, γ is the marginal profit we would get if we had one more hour of driver time available. This parameter allows us to capture in a minimalist way the importance of minimizing the total driver time (including pickup time). Therefore, the platform's choice of matching policy should minimize the sum of direct driver costs and opportunity costs:

$$\min_{(T_{\text{trip}}, T_{\text{pickup}}) \in \mathcal{P}} (c+\gamma) T_{\text{trip}} + \gamma T_{\text{pickup}}. \tag{11}$$

We now consider two scenarios; in the first situation, the firm chooses the optimal dispatch solution (i.e., driver times $(T^*_{\text{trip}}, T^*_{\text{pickup}})$ that constitute an optimal solution of Equation (11)). In the second scenario, similarly to the earlier parts of this paper, we disregard T_{pickup} and choose a policy that minimizes T_{trip}. That is, we match the requests to each other first by maximizing value (i.e., without taking into account where the drivers are), and in a second stage, we match drivers to matched request pairs and solo rides. We use the superscript v to denote this solution because it maximizes values: $T^v_{\text{trip}}, T^v_{\text{pickup}}$. Formally,

$$(T^*_{\text{trip}}, T^*_{\text{pickup}}) \in \underset{(T_{\text{trip}}, T_{\text{pickup}}) \in \mathcal{P}}{\arg\min} \; (c+\gamma) T_{\text{trip}} + \gamma T_{\text{pickup}},$$

$$(T^v_{\text{trip}}, T^v_{\text{pickup}}) \in \underset{(T_{\text{trip}}, T_{\text{pickup}}) \in \mathcal{P}}{\arg\min} \; T_{\text{trip}} \quad \text{and}$$

$$T^v_{\text{pickup}} \in \underset{\substack{T_{\text{pickup}} \\ (T^v_{\text{trip}}, T_{\text{pickup}}) \in \mathcal{P}}}{\arg\min} \; T_{\text{pickup}}.$$

We define the corresponding total costs to be $C^* = (c+\gamma) T^*_{\text{trip}} + \gamma T^*_{\text{pickup}}$ and $C^v = (c+\gamma) T^v_{\text{trip}} + \gamma T^v_{\text{pickup}}$, which yield that $C^* \leq C^v$ by definition. We are interested in understanding in which settings matching requests without taking into account the drivers are a good heuristic, and therefore, the approach followed in the earlier parts of the paper will offer realistic results. In particular, we are interested in whether C^v is close to C^* in the sense that the relative error $(C^v - C^*)/C^*$ is close to zero. To this end, we obtain the following upper bound.

Theorem 6. *The relative error of the two-stage optimization satisfies the following upper bound*:

$$\frac{C^v - C^*}{C^*} < \frac{\gamma}{\gamma + c} \cdot \frac{T^*_{\text{pickup}}}{T^*_{\text{trip}}} \cdot \left(\frac{T^v_{\text{pickup}}}{T^*_{\text{pickup}}} - 1 \right). \tag{12}$$

The upper bound in Theorem 6 is never binding and generally fairly conservative. We will study separately the three factors on the right-hand side of Equation (12) and argue that all three factors are on their own typically small.

8.1. The Opportunity Cost of Drivers Is Much Lower Than Their Costs

Consider the first term of the upper bound: $\gamma/(c+\gamma)$. In oversupplied conditions (when the number of drivers available is high compared with the demand), then $\gamma \approx 0$ by definition as the platform cannot benefit from additional driver time as there is already too much of it. This confirms our intuition that we can disregard drivers if we have many drivers.

The opportunity cost γ will be higher if not enough drivers are available, but the ratio $\gamma/(c+\gamma)$ will still be typically small. Consider assumption (2); platform commission rates are usually at most 25% of the price paid by the rider. This means that as drivers are paid c per hour of work, the platform earns a maximum profit of $c \cdot 0.25/0.75 = c/3$ for each hour of busy driver time. Additionally, the opportunity cost of driver time cannot be higher that the profit we get from a busy driver, which implies than $\gamma \leq c/3$. This does not take into account platform variable costs, such as computation costs, insurance costs, and credit card processing fees, which would lower γ further relative to c.

If the platform's commission rate is 25%, for γ to be equal to $c/3$ would require driver utilization to be 100% as extra drivers would need to be fully utilized to obtain $c/3$ in profits from them. Let U be the utilization of the drivers in the market (i.e., the fraction of time that they spend driving customers and thus, generating revenue). This argument can easily be tightened to $\gamma \leq cU/3$. Assumption (3) says that $U \leq 3/4$, implying that $\gamma \leq c/4$. Therefore, we obtain $\gamma/(c+\gamma) \leq 1/5$, with the true ratio often being far less than 1/5 except in severely undersupplied markets.

8.2. Pickup Times Are Small Compared with Trip Times

Consider now the second term of the upper bound: $T^*_{\text{pickup}}/T^*_{\text{trip}}$. This is the ratio of the expected total driver pickup time and the expected driver trip time in an optimal solution. This ratio is generally a function of the density of the market; as demand and supply grow, the amount of pickup time needed is proportionally smaller. Therefore, this ratio should approach zero as the market becomes more dense (even if we are in an undersupplied situation). Shared rides are not typically run in very sparse markets because it is hard to match riders in such markets. Furthermore, there is less total empty-car pickup time in shared rides than in solo rides as the driver can go from customer to customer without in-between pickup time. For all of these reasons, even when demand is not that dense, it is rare for the pickup time to be long compare with the ride time, and here is where we bring in assumption (4); in most practical situations, $T^*_{\text{pickup}}/T^*_{\text{trip}} \leq 0.3$. This ratio could be even lower in relatively dense markets.

8.3. There Is a Limit to How Much Pickup Times Can Be Reduced

Applying Theorem 6 and the assumptions (2)–(4), we have so far that

$$\frac{C^v - C^*}{C^*} < 0.06\left(\frac{T^v_{\text{pickup}}}{T^*_{\text{pickup}}} - 1\right).$$

The ratio $T^v_{\text{pickup}}/T^*_{\text{pickup}}$ represents how much more pickup time we add by disregarding the drivers' locations when matching the shared rides requests together. This ratio is the most complex one to bound, but we want to provide some intuition of why it cannot be much higher than one in practice. First, note that if $T^*_{\text{pickup}} < T^v_{\text{pickup}}$, then we must have $T^*_{\text{trip}} > T^v_{\text{trip}}$ as by definition, $(T^*_{\text{pickup}}, T^*_{\text{trip}})$ and $(T^v_{\text{pickup}}, T^v_{\text{trip}})$ are both on the Pareto frontier of minimizing $(T_{\text{pickup}}, T_{\text{trip}})$. That is, in order to improve pickup times, the optimal matching must sacrifice overall value by either matching pairs that are not as high valued or modifying routes. However, the optimal matching must do this trade-off in a way that is advantageous. Equation (11) plus assumptions (2) and (3) imply that if we are able to reduce T_{pickup} by δ, then we must not reduce T_{trip} by more than $\delta\gamma/(c+\gamma) \leq 0.2\delta$ for it to be optimal and even less if we are not severely undersupplied. Furthermore, there is another constraint on the potential pickup time savings; the optimal approach and the two-stage approach both use the same drivers and requests. If we match the requests i, j in the optimal approach or the two-stage approach, the driver will have to pick up either i or j first, and therefore, the pickup time cannot be better than the minimum pickup time of the two. These combined effects offer relatively little margin to save on pickup times, but getting an accurate estimation of the saving potential involves a complex dynamic spatial hypergraph matching. Instead, we introduce assumption (5), which conservatively allows the two-stage approach to have 50% more pickup times than the optimal one. This translates to $T^v_{\text{pickup}}/T^*_{\text{pickup}} \leq 1.5$, which gives us

$$\frac{C^v - C^*}{C^*} < 0.03.$$

If we adapt assumption (5) to allow for 100% more pickup time in the two-stage approach instead of 50%, our bound would be 6%. If the market is oversupplied, the ratio $T^v_{\text{pickup}}/T^*_{\text{pickup}}$ would be close to one, and the bound would be very close to zero.

8.4. Disregarding Pickup Times Is Realistic

Putting everything together, we have that in many realistic scenarios, $\frac{C^v - C^*}{C^*} \leq 0.5 \times 0.2 \times 0.3 = 3\%$. The error is much lower (close to zero) if supply is in excess or balanced with demand and/or if the market is dense. These would be the ideal conditions for our model as well as the most typical conditions seen in shared rides marketplaces.

9. Conclusions

In this paper, we studied detours in shared rides via a combination of theory and simulation. Our first contribution emerges from Theorem 1, where we showed that the sum of detours and values is equal to the shared component. This leads to a bound; the sum of value and detour ratios cannot exceed 1/2 (Theorem 2, Theorem 3, part (1), and Theorem 4 show this for pairs of rides, for the Pareto frontier, and for individual rides, respectively). That is, regardless of the city topology and the request density, large detours and large values cannot occur simultaneously.

Our extensive set of simulations and analytical results leads to a second set of contributions. We showed that the expected detours are typically small and that request-level detour constraints (on the order of half of trip lengths) can be added at little cost on top of average detour constraints. In grid networks, this occurs because detours are generally small and many times, even equal to zero. Meanwhile, in the tree and Manhattan networks, this occurs because of the low match probability when the demand density is low and the fact that pairs are usually low detour when the density is high. We also find that, from a shared rides perspective, the Manhattan network looks more like a highway network than a grid. In our study of the Pareto frontier, we further showed that mild detour penalties have little cost in terms of value and great upside in reducing detours. In some networks, even significant detour penalties cause only minor reductions in values. This suggests that in some settings, detours might not be the most important friction to the adoption of shared rides, with other frictions, such as additional pickup and drop-off times, playing a bigger role. Overall, density plays a much more significant role than detours in the value generated by a shared rides platform. This suggests that a powerful new approach for dealing with detours might be a two-product shared rides marketplace that allows customers to express their individual preferences with respect to a detour-price trade-off.

Acknowledgments

This work was partially conducted while Sébastien Martin was a postdoctoral fellow at Lyft and while Ilan Lobel was a consulting data scientist for Lyft.

Endnote

[1] Although 25% is the stated commission rate of Uber in many markets at the time of writing this paper, other fees can be added to it,

resulting in a total cut above 25%. See https://therideshareguy.com/how-much-do-uber-drivers-make/.

References

Afèche P, Liu Z, Maglaras C (2023) Ride-hailing networks with strategic drivers: The impact of platform control capabilities on performance. *Manufacturing Service Oper. Management* 25(5):1890–1908.

Alonso-Mora J, Samaranayake S, Wallar A, Frazzoli E, Rus D (2017) On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proc. Natl. Acad. Sci. USA* 114(3):462–467.

Aouad A, Saritaç Ö (2022) Dynamic stochastic matching under limited time. *Oper. Res.* 70(4):2349–2383.

Ashlagi I, Burq M, Dutta C, Jaillet P, Saberi A, Sholley C (2022) Edge weighted online windowed matching. *Math. Oper. Res.* 48(2):999–1016.

Berbeglia G, Cordeau J-F, Laporte G (2010) Dynamic pickup and delivery problems. *Eur. J. Oper. Res.* 202(1):8–15.

Bertsimas D, Jaillet P, Martin S (2019) Online vehicle routing: The edge of optimization in large-scale applications. *Oper. Res.* 67(1):143–162.

Besbes O, Castro F, Lobel I (2021) Surge pricing and its spatial supply response. *Management Sci.* 67(3):1350–1367.

Besbes O, Castro F, Lobel I (2022) Spatial capacity planning. *Oper. Res.* 70(2):1271–1291.

Bimpikis K, Candogan O, Saban D (2019) Spatial pricing in ride-sharing networks. *Oper. Res.* 67(3):744–769.

Biswas A, Gopalakrishnan R, Tulabandhula T, Metrewar A, Mukherjee K, Thangaraj RS (2018) Impact of detour-aware policies on maximizing profit in ridesharing. *Proc. Internat. Workshop Agents Traffic Transportation*.

Boeing G (2017) Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Comput. Environ. Urban Systems* 65:126–139.

Cachon GP, Daniels KM, Lobel R (2017) The role of surge pricing on a service platform with self-scheduling capacity. *Manufacturing Service Oper. Management* 19(3):368–384.

Castillo JC, Knoepfle D, Weyl G (2017) Surge pricing solves the wild goose chase. *Proc. ACM Conf. Econom. Comput.* (ACM, New York), 241–242.

Daganzo SF, Ouyang Y, Yang H (2020) Analysis of ride-sharing with service time and detour guarantees. *Transportation Res. Part B Methodological* 140:130–150.

Garg N, Nazerzadeh H (2022) Driver surge pricing. *Management Sci.* 68(5):3219–3235.

Gopalakrishnan R, Mukherjee K, Tulabandhula T (2016) The costs and benefits of sharing: Sequential individual rationality and sequential fairness. Working paper, Queen's University, Kingston, ON.

Griswold A (2018) The future of Lyft is more carpooling. *Quartz* (June 7), https://qz.com/1299055/lyft-wants-half-its-rides-to-be-shared-by-2020.

Gurvich I, Lariviere M, Moreno A (2019) Operations in the on-demand economy: Staffing services with self-scheduling capacity. Hu M, ed. *Sharing Economy*, Springer Series in Supply Chain Management, vol. 6 (Springer, Cham, Switzerland), 249–278.

Kann V (1991) Maximum bounded 3-dimensional matching is MAX SNP-complete. *Inform. Processing Lett.* 37(1):27–35.

Lo J, Morseman S (2018) The perfect UberPOOL: A case study on trade-offs. *Ethnographic Praxis Indust. Conf. Proc.* (Wiley, Hoboken, NJ), 195–223.

Lu A, Frazier PI, Kislev O (2018) Surge pricing moves Uber's driver-partners. *Proc. ACM Conf. Econom. Comput* (ACM, New York), 3.

Lyft (2018) Lyft's new app creates positive change for passengers and cities. *Lyft Blog* (June 7), https://blog.lyft.com/posts/new-app.

Ma H, Fang F, Parkes DC (2021) Spatio-temporal pricing for ridesharing platforms. *Oper. Res.* 70(2):1025–1041.

Middleton S, Schroeckenthaler K, Papayannoulis V, Gopalakrishna D (2021) Analysis of travel choices and scenarios for sharing rides. Technical report, Federal Highway Administration, U.S. Department of Transportation, Washington, DC.

OpenStreetMap Contributors (2020) Planet dump retrieved from Planet OSM. Accessed September 24, 2020, https://www.openstreetmap.org.

Ozkan E, Ward AR (2020) Dynamic matching for real-time ridesharing. *Stochastic Systems* 10(1):29–70.

Santi P, Resta G, Szell M, Sobolevsky S, Strogatz SH, Ratti C (2014) Quantifying the benefits of vehicle pooling with shareability networks. *Proc. Natl. Acad. Sci. USA* 111(37):13290–13294.

Taylor TA (2018) On-demand service platforms. *Manufacturing Service Oper. Management* 20(4):704–720.

Wang H, Yang H (2019) Ridesourcing systems: A framework and review. *Transportation Res. Part B Methodological* 129:122–155.

Wong K-I, Bell MGH (2006) The optimal dispatching of taxis under congestion: A rolling horizon approach. *J. Advanced Transportation* 40(2):203–220.

Yan C, Zhu H, Korolko N, Woodard D (2019) Dynamic pricing and matching in ride-hailing platforms. *Naval Res. Logist.* 67(8):705–724.