

# Impact of Detour-Aware Policies on Maximizing Profit in Ridesharing

**Arpita Biswas**  
arpitab@iisc.ac.in  
Indian Institute of Science

**Ragavendran Gopalakrishnan**  
rg584@cornell.edu  
Cornell University

**Theja Tulabandhula**  
tt@theja.org  
University of Illinois Chicago

**Asmita Metrewar**  
metrewar.asmita@gmail.com  
Xerox Research Center India

**Koyel Mukherjee**  
kmukherj@in.ibm.com  
IBM Research India

**Raja Subramaniam Thangaraj**  
rasubram@amazon.com  
Amazon

## Abstract

This paper provides efficient solutions to maximize profit for commercial ridesharing services, under a pricing model with detour-based discounts for passengers. We propose a greedy heuristic for real-time ride matching and compare with other heuristics. We study the trade-offs between optimality and execution time that the solutions offer. Simulations on New York City (NYC) taxi trip data show that our heuristics are up to 93% optimal and  $10^5$  times faster than the exponential-time optimal algorithm.

Commercial ridesharing service providers generate significant savings by matching multiple ride requests, which are typically shared between the service provider (increased profit) and the ridesharing passengers (discounts). It is not clear a priori how to split, since higher discounts would encourage more ridesharing, thereby increasing total savings, but the fraction of savings taken as profit is reduced. We simulate a scenario where the decisions of the passengers to opt for ridesharing depend on the discount offered by the provider. We provide an adaptive learning algorithm IDFLA that learns the optimal profit-maximizing discount factor for the provider. An evaluation over NYC data shows that IDFLA, on average, learns the optimal discount factor under 16 iterations.

Finally, we investigate the impact of imposing a detour-aware routing assuring sequential individual rationality which offers a better ride experience and increasing the provider's market share, but at the cost of decreased average per-ride profit due to the reduced number of matched rides. We construct a model that captures these opposing effects, wherein simulations on NYC data show that 7% increase in market share would suffice to offset the decreased average per-ride profit.

*Keywords:* Ridesharing, Profit Maximization, Adaptive Learning, Simulation, Heuristics

This work was done when the authors were working with Xerox Research Center India (XRCI).

## 1 Introduction

Ridesharing is a key initiative that can curb ever-increasing congestion on the urban transportation network and is an important means to move the world towards a sustainable future. While the term *ridesharing* has been used to include peer-to-peer carpooling platforms as well, here we focus on commercial ridesharing service providers such as LyftLine and UberPool, that hold a significant share of the ridesharing population.

While ridesharing is undoubtedly appealing from a sustainability perspective, profit maximization is the primary goal of commercial providers. Given the emerging literature on detour-aware pricing and routing policies, e.g., [Jung *et al.*, 2013; Gopalakrishnan *et al.*, 2016] that enhance the ride experience, algorithms for profitable ride-matching should adapt to keep up. Such quality-enhancing policies encourage increased adoption of ridesharing, but perhaps at the cost of reduced average per-ride profit. We study three connected issues centered around profit optimization in commercial ridesharing.

First, it is unclear how to tune real-time ride-matching algorithms to quickly find a near-optimal set of matches that results in high profit under detour-based discounts (the exact optimization is an NP-hard problem [Cordeau *et al.*, 2006]).

Second, ridesharing generates significant economic savings by reducing the driver-miles necessary to serve a set of passengers, which are split between the service provider (their profit) and the passengers (as discounts). It is not obvious how this split should be effected, since higher discounts encourage more ridesharing, increasing total savings, but the fraction of savings that constitutes the profit is smaller.

Third, in order to survive in an increasingly competitive market, ridesharing providers adopt additional quality-enhancing features which do not align fully with the goal of profit maximization, perhaps resulting in lower profit (due to the smaller feasible set of quality-compliant rides). It is important to understand the overall impact on profitability, and ensure that any resulting increase in market share can counter the reduced average per-ride profit.

The issues mentioned above are crucial for commercial ridesharing platforms; however, they are less explored. We address these issues, give solutions, and provide extensive evaluation on a real-world dataset.

**Our Contributions:** We investigate three important aspects of profit optimization in commercial ridesharing, resulting in the following contributions:

- We provide an Integer Linear Program (ILP), and also an efficient *greedy* heuristic that quickly matches passengers to rides to approximately maximize profit under a detour-based discount policy. We evaluate their performance experimentally (Section 3).
- Under a model in which the likelihood of opting for ridesharing depends on the offered discount, we propose an adaptive algorithm that learns the profit-maximizing discount parameter (Section 4).
- We additionally impose a detour-aware routing policy based on *sequential individual rationality*, introduced in [Gopalakrishnan *et al.*, 2016]. Using simulations, we investigate the minimum increase in market share needed to offset the reduced average per-ride profit (Section 5).

## 2 Related Work

There is a huge body of literature that studies several optimization problems related to ride sharing [Furuhata *et al.*, 2013; Agatz *et al.*, 2012; Pelzer *et al.*, 2015], but the problem of pricing in ridesharing has been relatively less studied, with profit optimization rarely discussed. [Banerjee *et al.*, 2015] characterizes the pros/cons of static versus dynamic pricing. [Jung *et al.*, 2013] provides heuristics for maximizing profit subject to bounded detours. In contrast, we study profit maximization over a pricing model that incorporates a detour-based discount.

[Nguyen, 2013] proposes a dynamic, demand-responsive ridesharing system using auction mechanisms, where customers enjoy additive detour-based discounts. Our model considers *multiplicative* detour-based discounts; moreover, we also provide an algorithm to learn the profit-maximizing discount parameter under a suitable user choice model.

A closely related problem is that of cost sharing in peer-to-peer ridesharing or carpooling, which, until recently, received little attention. Individual passengers are either asked to post what they are willing to pay in advance [Cao *et al.*, 2015], to share the total cost proportionately among themselves according to the distances travelled [Geisberger *et al.*, 2010; Agatz *et al.*, 2011], or negotiate their shares on their own during/after the ride. Such methods ignore the real-time costs and delays incurred during the ride and are generally insensitive to the disproportionate delays encountered during the ride. Fair cost sharing in ridesharing has been studied [Bistaffa *et al.*, 2015; Kleiner *et al.*, 2011; Zhao *et al.*, 2014; Kamar and Horvitz, 2009; Gopalakrishnan *et al.*, 2016], but its impact on profitability is left unexplored. We fill this gap by studying the trade-off between average per-ride profits and market share.

## 3 Matching Rides to Maximize Profit

Ridesharing generates significant profit to commercial platforms such as Uber and Lyft. The profit from a single cab is the difference between the fare paid by the passengers and the

amount paid to the driver. When compared to assigning each user their own cab, ridesharing requires fewer cabs to serve the same set of users, leading to increased profit.

At the core of these platforms is a real-time matching algorithm that runs at regular time intervals, and generates a set of matched users from a set of waiting users (preprocessed to accommodate other spatio-temporal constraints). Such a system would maintain a pool of pending requests, and then invoke the matching algorithm every few minutes to determine optimal matches. We provide a matching heuristic that directly optimizes profit, under a detour-based discount scheme. Any unmatched requests left at the end of the algorithm either wait for another round, or, after a certain waiting time threshold elapses, are assigned fresh, empty cabs. Thus, it is important to provide a time-efficient matching algorithm for the larger dynamic ridesharing system to be sufficiently responsive to incoming user requests.

Moreover, under a detour-based discount policy, shared rides with smaller detours are more likely to be profitable. However, such a myopic observation could lead to suboptimal solutions, thus rendering the problem nontrivial.

In Section 3.1, we provide an Integer Linear Program (ILP) that matches  $n$  users to shared cabs (each with capacity  $\zeta$ ) to maximize profit, given a multiplicative detour-based discount  $f_p(\delta_i, \tau_i)$ , which is a *linear* function of the distance-wise ( $\delta_i$ ) and time-wise ( $\tau_i$ ) fractional detours experienced by user  $i$ .  $c_b$  is the “base” (fixed) cost for a ride,  $c_d$  and  $c_t$  are the costs per unit distance and time, respectively.  $f_d$  is the fraction of driver earnings taken by the service provider.

### 3.1 ILP Formulation

Each user  $i$  is associated with a source-destination pair  $(S_i, D_i)$ . We consider  $n$  initially empty cabs and construct a complete directed graph  $G = (V, E)$  with  $2n$  vertices that represent all the sources and destinations, that is,  $V = \{S_i, D_i\}_{i=1}^n$ . Each directed edge  $e = (u, v)$  represents the best route from location  $u$  to  $v$ , and weights  $w_d(e)$  and  $w_t(e)$  denote the corresponding distance and time, respectively.

The ILP seeks values to assign to a set of optimization variables in order to maximize an objective function subject to a set of constraints.

**Optimization Variables:** Let  $x_{i,j,e} \in \{0, 1\}$  denote whether or not user  $i$  is served by cab  $j$  along edge  $e$ . Let  $y_{j,e} \in \{0, 1\}$  denote whether cab  $j$  travels along edge  $e$  while serving one or more users. Let  $z_j \in \{0, 1\}$  denote whether cab  $j$  serves at least one user. We set  $x_{i,j,(u,v)} = 0$  if  $(u, v) \notin E$ .

### Objective Function:

For each user  $i$ , we have:

- Distance travelled,  $d_i = \sum_{j=1}^n \sum_{e \in E} x_{i,j,e} w_d(e)$ .
- Time in travel,  $t_i = \sum_{j=1}^n \sum_{e \in E} x_{i,j,e} w_t(e)$ .
- Distance-wise fractional detour,  $\delta_i = \frac{d_i - w_d(S_i, D_i)}{w_d(S_i, D_i)}$ .
- Time-wise fractional detour,  $\tau_i = \frac{t_i - w_t(S_i, D_i)}{w_t(S_i, D_i)}$ .
- Fare,  $C_i = (1 - f_p(\delta_i, \tau_i))(c_b + c_d w_d(S_i, D_i) + c_t w_t(S_i, D_i))$ .

For each cab  $j$ , we have:

- (a) Distance traveled,  $\sigma_j = \sum_{e \in E} y_{j,e} w_d(e)$ .
- (b) Time in travel,  $\eta_j = \sum_{e \in E} y_{j,e} w_t(e)$ .
- (c) Driver earnings,  $E_j = (1 - f_d)(c_b z_j + c_d \sigma_j + c_t \eta_j)$ .

Thus, the objective function to be maximized is the total profit, given by  $p = \sum_{i=1}^n C_i - \sum_{j=1}^n E_j$ .

### Constraints:

- (i)  $y_{j,e} = 1$  if and only if  $x_{i,j,e} = 1$  for some  $i$ :
 
$$y_{j,e} \geq x_{i,j,e} \quad \forall i, j \in \{1, 2, \dots, n\} \quad \forall e \in E$$

$$y_{j,e} \leq \sum_{i=1}^n x_{i,j,e} \quad \forall j \in \{1, 2, \dots, n\} \quad \forall e \in E$$
- (ii)  $z_j = 1$  if and only if  $y_{j,e} = 1$  for some  $e$ :
 
$$z_j \geq y_{j,e} \quad \forall j \in \{1, 2, \dots, n\} \quad \forall e \in E$$

$$z_j \leq \sum_{e \in E} y_{j,e} \quad \forall j \in \{1, 2, \dots, n\}$$
- (iii) Each user is picked up by exactly one cab:
 
$$\sum_{u \in V} \sum_{j=1}^n x_{i,j,(S_i,u)} = 1 \quad \forall i \in \{1, 2, \dots, n\}$$
- (iv) Each user is dropped off by exactly one cab:
 
$$\sum_{u \in V} \sum_{j=1}^n x_{i,j,(u,D_i)} = 1 \quad \forall i \in \{1, \dots, n\}$$
- (v) Each user is not served before pickup:
 
$$\sum_{u \in V} \sum_{j=1}^n x_{i,j,(u,S_i)} = 0 \quad \forall i \in \{1, \dots, n\}$$
- (vi) Each user is not served after dropoff:
 
$$\sum_{u \in V} \sum_{j=1}^n x_{i,j,(D_i,u)} = 0 \quad \forall i \in \{1, \dots, n\}$$
- (vii) Each user is served by a single cab, only between pickup and dropoff:
 
$$\sum_{u \in V} x_{i,j,(u,v)} = \sum_{u \in V} x_{i,j,(v,u)} \quad \forall v \in V \setminus \{S_i, D_i\} \quad \forall i \quad \forall j$$
- (viii) Each cab serves at most  $\zeta$  users along any edge:
 
$$\sum_{i=1}^n x_{i,j,e} \leq \zeta \quad \forall j \in \{1, \dots, n\} \quad \forall e \in E$$
- (ix) Each cab's overall route is not disjoint, i.e., there are no gaps between serving users during which the cab is empty. Equivalently, if a cab serves a total of  $k \geq 1$  users, then it must do so using exactly  $2k - 1$  edges:
 
$$\sum_{e \in E} y_{j,e} \geq 2 \left( \sum_{i=1}^n \sum_{u \in V} x_{i,j,(S_i,u)} \right) - 1$$

The inequality ( $\geq$ ) is to allow empty cabs ( $k = 0$ ).

Computing the optimal solution to the ILP takes a lot of time and memory, and is unsuitable for real-time ride matching, which requires fast, near-optimal techniques. However, it is a handy benchmark against which to experimentally evaluate the performance of our proposed methods (Section 3.3).

## 3.2 Heuristic for Real-Time Ride Matching

Given a set  $\mathcal{S}$  of users matched to a cab, the driving distance and time for the best route for serving all the users  $i \in \mathcal{S}$  are denoted by  $d(\mathcal{S})$  and  $t(\mathcal{S})$ , respectively. For each user  $i \in \mathcal{S}$ ,  $d_i(\mathcal{S})$  and  $t_i(\mathcal{S})$  denote the driving distance and time from  $S_i$  to  $D_i$  along that route, respectively. Thus, the profit from a cab serving a set of users  $\mathcal{S}$  can be computed as follows:

- **User Cost:** The cost of the ride to user  $i \in \mathcal{S}$ ,
 
$$C_i(\mathcal{S}) = (1 - f_p(\delta_i(\mathcal{S}), \tau_i(\mathcal{S}))) (c_b + c_d d_i(\{i\}) + c_t t_i(\{i\})), \quad (1)$$

where  $\delta_i(\mathcal{S}) = \frac{d_i(\mathcal{S}) - d_i(\{i\})}{d_i(\{i\})}$ , and  $\tau_i(\mathcal{S}) = \frac{t_i(\mathcal{S}) - t_i(\{i\})}{t_i(\{i\})}$ .

- **Driver Earnings:** The earnings to the driver from the ride,
 
$$E(\mathcal{S}) = (1 - f_d) \cdot (c_b + c_d d(\mathcal{S}) + c_t t(\mathcal{S})). \quad (2)$$

- **Profit:** The profit to the service provider,

$$p(\mathcal{S}) = \sum_{i \in \mathcal{S}} C_i(\mathcal{S}) - E(\mathcal{S}). \quad (3)$$

The incremental profit of *merging* two cabs (combining the users matched to two cabs into one and removing the other cab from the system), serving sets of users  $\mathcal{S}_j$  and  $\mathcal{S}_k$  is:

$$\Delta p(\mathcal{S}_j, \mathcal{S}_k) = p(\mathcal{S}_j \cup \mathcal{S}_k) - p(\mathcal{S}_j) - p(\mathcal{S}_k). \quad (4)$$

The above framework for real-time ride matching is more general than that of the ILP. A limitation of the ILP is that  $f_p(\cdot)$  is restricted to be a linear function. Moreover, it may not be possible for an ILP to explicitly incorporate additional detour-aware routing constraints and user preferences. However, during performance evaluation of our proposed methods, we limit the scope to the constraints listed in Section 3.1, in order to compare against the optimal ILP solution.

### Greedy-Max-Profit-Based Method

We now describe a greedy method for profit-maximizing real-time matching. We begin by assigning each of  $n$  input users to their "own" cab, i.e., no two users are assigned to the same cab.<sup>1</sup> We maintain two collections of cabs, namely, a set of unavailable cabs  $\mathcal{U}$  and a current pool of cabs  $\mathcal{A}$ . We initialize  $\mathcal{A}$  with the  $n$  single-user cabs and  $\mathcal{U} = \emptyset$ . We repeatedly perform the following steps until  $\mathcal{A}$  is empty, at which point, we return  $\mathcal{U}$  as the output:

- (a) For every two cabs  $j$  and  $k$  in  $\mathcal{A}$ , if it is feasible to *merge* them subject to the capacity constraint (and any other optional constraints), compute the corresponding incremental profit  $p_{jk} = \Delta p(\mathcal{S}_j, \mathcal{S}_k)$ .
- (b) Find two cabs  $j^*$  and  $k^*$  such that  $p_{j^*k^*}$  is maximum. If  $p_{j^*k^*} < 0$ , terminate and return  $\mathcal{U} \cup \mathcal{A}$  as output. Otherwise, *merge* them and replace the individual cabs  $j^*$  and  $k^*$  in  $\mathcal{A}$  with the merged cab. If the merged cab has  $\zeta$  passengers, move it from  $\mathcal{A}$  to  $\mathcal{U}$ .

Note that each iteration either terminates the method, or decreases the size of  $\mathcal{A}$  by at least one; therefore, the method terminates after at most  $n - 1$  iterations. The time complexity of this method is  $O(n^2 \log n)$ .

<sup>1</sup>We assume that all cabs are initially empty; however, our methods extend to a more general case where some cabs already have passengers in them with their own constraints to begin with.

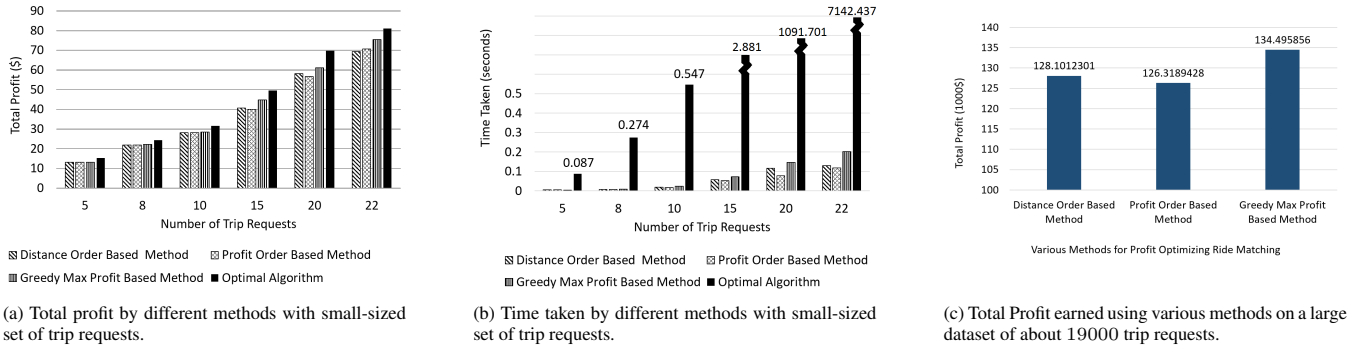


Figure 1: Comparison of various heuristic matching methods according to the total profit and running time.

### 3.3 Experimental Performance Evaluation

In order to compare the performance of our proposed greedy method, we evaluate them against the optimal solution computed by solving the ILP of Section 3.1. Since the ILP takes more than 2 hours on an average (even with only 22 requests) to output the optimal solution, we first evaluate the near-optimality of our methods using a small dataset, and then use a larger dataset for demonstrating the scalability and comparing the speed-optimality trade-off of our proposed heuristic and order-based greedy heuristics mentioned in [Biswas *et al.*, 2017]<sup>2</sup>. To compare these order based methods with our proposed method, the orderings that we consider are:

- (a) Distance Order: Decreasing order of  $d(S)$ .
- (b) Profit Order: Increasing order of  $p(S)$ .

#### Experimental Setup

We evaluate our methods using publicly available New York City (NYC) taxi trip data [Whong and Monroy-Hernández, 2013]. The 19GB dataset has logs for all NYC trips taken in 2013. Each trip in the dataset specifies pickup and dropoff coordinates, pickup time, dropoff time, trip distance and travel time. Each trip is considered as a “user” for our simulation. We assume that each cab serves at most  $\zeta = 3$  users.

We discretize the underlying NYC map into grids of  $100m^2$  area and select 2543 representative landmarks using the methodology of [Rajasubramaniam *et al.*, 2017]. We pre-compute the inter-landmark distance and time using Open Trip Planner [OTP, 2016]. For each user, we map their source and destination coordinates to their nearest representative landmarks and use the precomputed values for comput-

<sup>2</sup>For completeness, we now briefly describe the greedy heuristics which are parameterized by an order protocol (originally proposed in [Biswas *et al.*, 2017]). An ordered list  $\mathcal{L}$  of the initial  $n$  single-rider cabs is created according to a certain order, and the set of unavailable cabs  $\mathcal{U} = \emptyset$  is initialized. At each iteration, the cab at the top of the list (say  $j$ ) is picked and merged with the first cab  $k$  down the list such that  $p_{jk} > 0$ , subject to the capacity constraint (and any other optional constraints). Then, the individual cabs  $j$  and  $k$  are removed from  $\mathcal{L}$  and the merged cab is inserted at the appropriate position in  $\mathcal{L}$ . If no such cab  $k$  exists, then  $j$  is moved from  $\mathcal{L}$  to  $\mathcal{U}$ . If the merged cab is full to its capacity (that is, it has  $\zeta$  passengers), it is moved from  $\mathcal{L}$  to  $\mathcal{U}$ . These methods have a time complexity of  $O(n \log n)$ .

ing necessary metrics such as profit, detour, driving distance and time. The values  $c_b$ ,  $c_d$ ,  $c_t$ , and  $f_d$  are taken from publicly available Lyft user pricing and driver payment for NYC [Lyft-Line, 2016a; LyftLine, 2016b].

#### Experimental Results

We select NYC taxi trips on randomly chosen days of 2013, between 7:45 - 8:00 pm. We intentionally choose the rush times, since many shared rides would be possible, and it would be interesting to see how our proposed heuristic performs. We run our method on several subsamples by considering a different number of users each time (5, 8, 10, 15, 20, and 22). We compare the total profit generated (Figure 1a) and the total time taken (Figure 1b) across each method, averaged over the small-sized subsamples of users. We observe that the total profit obtained by the Greedy-Max-Profit-Based method is up to 93% optimal and runs  $10^5$  times faster than the optimal algorithm on a machine with a quad Intel Core i7 processor with 32GB RAM. The optimal algorithm becomes intractable for instances where the number of users exceeds 25 (terminated after over 14.22 hours due to lack of memory).

Our proposed algorithm scales to a larger input set with  $\sim 19000$  users (obtained from a one-hour slot). The Greedy-Max-Profit-Based method takes  $\sim 35$  minutes while the Distance-Order-Based and Profit-Order-Based methods take 7 and 4 minutes, respectively.<sup>3</sup> However, figure 1c shows that the performance of the former is significantly better than that of the latter methods in terms of the total profit. Thus, the speed-optimality trade-off becomes important in choosing the method that best suits the needs of a service provider. Finally, we also observe that the Greedy-Max-Profit-Based method matches  $\sim 19000$  users into 6701 cabs (reducing the operational cost significantly), obtaining a profit of  $\sim \$134,500$ .

### 4 Learning the Optimal Discount Policy

In the previous section, we laid out efficient ride-matching heuristics to maximize profit under a *fixed* detour-based discount for users. In this section, our goal is to learn the optimal detour-based discount policy, under a *fixed* ride-matching

<sup>3</sup>If the matching algorithm is invoked once per minute, in NYC, there are only  $\sim 300$  requests per minute on an average, for which all the greedy methods finish in less than a second.

algorithm. For simplicity, we assume that the detour-based discount is a linear function of the fractional distance-wise detour. Thus, the discount to a user  $i \in \mathcal{S}$  is given by

$$f_p^\theta(\delta_i(\mathcal{S})) = \tan(\theta)\delta_i(\mathcal{S}) + b, \quad (5)$$

where  $b$  is a constant denoting the minimum discount given to a user (to incentivize them to opt for ridesharing in the first place), and the discount parameter  $\theta \in [0^\circ, 90^\circ)$  governs how steeply the discount increases with the detour. Figure 2 plots (5) for various values of  $\theta$ , assuming  $b = 10\%$ .

Let  $\mathbb{P}(\theta)$  denote the probability of a user opting to rideshare, a nondecreasing function of  $\theta$ . A larger  $\theta$  would lead to a larger population of users who are willing to rideshare, and potentially more shared rides; however, the profit from each shared ride would be smaller. Since the expected total profit depends on the number of users opting to rideshare, as well as the per-ride profit, it is important to find a value of  $\theta$  that provides the optimal balance to maximize the expected total profit. We propose a method, which we call *Iterative Discount Function Learning Algorithm (IDFLA)*, that learns such a  $\theta$  over a period of time.

Each day, a value of  $\theta$  is declared by the service provider. In response, the users who opt for ridesharing are then matched throughout the day using one of the heuristics from Section 3.2. The total profit earned is computed at the end of the day. **IDFLA** (Algorithm 1) learns the expected daily profit earned using different values of  $\theta$ , and eventually converges to the best  $\theta^*$  that maximizes this quantity. This kind of sequential decision making in stochastic environment can be categorized as a stochastic multi-armed bandit problem [Bubeck and Cesa-Bianchi, 2012], where the  $\theta$  values are the arms, and the daily profit earned by using a certain  $\theta$  is the reward earned by pulling the corresponding arm. **IDFLA** uses a popular technique for solving stochastic multi-armed bandit problems, called **UCB1** [Auer et al., 2002].

---

#### Algorithm 1 IDFLA

---

- 1: **Input:** A finite number of  $\theta$  values in an array  $\Theta$
  - 2:  $P_h$  stores the total profit with  $\Theta[h]$
  - 3:  $k_h$  stores the number of days  $\Theta[h]$  is declared
  - 4: **for**  $t \leftarrow 1$  **to**  $|\Theta|$  **do**
  - 5:     Choose  $\Theta[t]$  on  $t^{\text{th}}$  day and observe profit earned  $p$ ;
  - 6:     Set  $P_t \leftarrow p$ ;   Set  $k_t \leftarrow 1$ ;
  - 7:     Find  $h^* = \arg \max_h \left( P_h + \sqrt{2 \ln |\Theta|} \right)$ ;
  - 8:     **for**  $t \leftarrow |\Theta| + 1, |\Theta| + 2, \dots$  **do**
  - 9:         Choose  $\Theta[h^*]$  on  $t^{\text{th}}$  day and observe profit earned  $p$ ;
  - 10:         Update  $P_{h^*} \leftarrow P_{h^*} + p$ ;   Update  $k_{h^*} \leftarrow k_{h^*} + 1$ ;
  - 11:         Find  $h^* = \arg \max_h \left( \frac{P_{h^*}}{k_{h^*}} + \sqrt{\frac{2 \ln t}{k_h}} \right)$  for  $(t + 1)^{\text{th}}$  day;
- 

### 4.1 Performance Evaluation of IDFLA

We evaluate our learning method **IDFLA** under the Greedy-Max-Profit-Based ride-matching heuristic. We take  $\Theta = \{10^\circ, 20^\circ, 30^\circ, 40^\circ, 50^\circ, 60^\circ, 70^\circ, 80^\circ\}$ , and run **IDFLA** to

find near-optimal discount parameter,  $\Theta[h^*]$ . The value of optimal discount parameter depends on  $\mathbb{P}(\cdot)$  which is the probability of opting rideshare. For simulating the user-behavior, we need to assume  $\mathbb{P}(\cdot)$  to be any non-decreasing function of  $\theta$ . In practice, this probability  $\mathbb{P}(\theta)$  is estimated while the optimal  $\theta$  is learnt (step 14 of **IDFLA**). This corresponds to the fraction of times ( $k_\theta$ ) the users have opted for ridesharing whenever the discount parameter was  $\theta$ . The  $\mathbb{P}(\theta)$  function that we have assumed for our simulation, is shown in Figure 3.

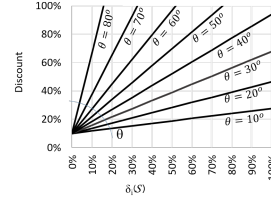


Figure 2: Various discount functions.

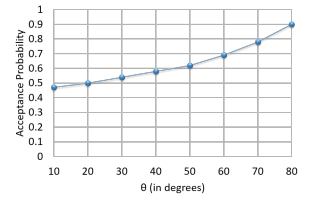


Figure 3: Acceptance probability.

Taxi trips in NYC between 7:45 - 8:00 pm of the 52 Wednesdays of 2013 were used to compare the performance of **IDFLA** with an “oracle” method (**BDF**) that knows the Best Discount Function a priori. Figure 4 shows the average daily profit obtained by declaring the same discount function everyday. We observe that the best discount function is obtained when  $\theta = 40^\circ$ ; therefore, **BDF** chooses  $\theta = 40^\circ$  everyday. Figure 5a shows that the discount parameter selected by **IDFLA** converges to that of **BDF** in 16 days.

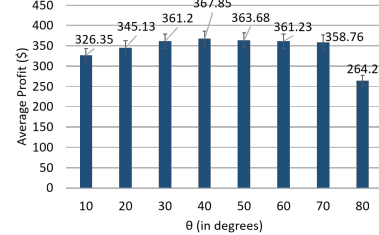
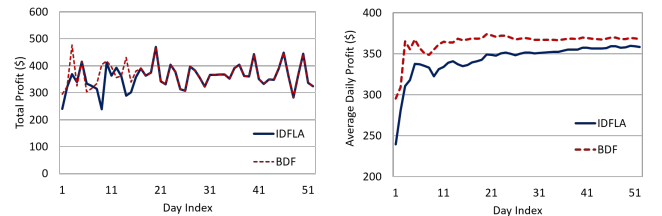


Figure 4: Average daily profit obtained by various discount functions over 52 days



(a) Comparing daily profit obtained by **IDFLA** and **BDF**. (b) Average daily profit of **IDFLA** converging to that of **BDF**.

Figure 5: Comparing **IDFLA** and **BDF**

We also observe (Figure 5b) that the average daily profit, that is, the cumulative total profit over  $t$  days divided by  $t$ , converges to that of the best discount function **BDF**. We show

that after 52 days, the difference<sup>4</sup> in average daily profit obtained by **IDFLA** and **BDF** is just \$9.58, or 2.67%. Thus, **IDFLA** quickly learns the optimal discount parameter that can help the service provider maximize profit while keeping users sufficiently happy by providing suitable discounts.

## 5 Impact of Detour-Aware Routing

In conjunction with desirable pricing schemes for ridesharing users, recent work has proposed detour-aware routing policies that impose upper bounds on the total detour [Jung *et al.*, 2013] (static) or incremental detours [Gopalakrishnan *et al.*, 2016] (dynamic). Such detour-aware routing policies work together with detour-based discount policies in order to enhance the quality of the ride experience for ridesharing users. This induces a similar trade-off as the one in Section 4: imposing additional quality-enhancing constraints into the ride-matching process incentivizes greater adoption of ridesharing and increases the market share of the ridesharing population, but could result in reduced average per-ride profit. In this section, we consider the impact of one such detour-aware routing policy, based on the concept of *Sequential Individual Rationality* (SIR), introduced by [Gopalakrishnan *et al.*, 2016].

SIR guarantees that the *disutility* to existing users (sum of the monetary cost and an “inconvenience cost” due to detours) in a shared ride is non-increasing as additional users are picked up. A parameterized version of SIR, called SIR- $\gamma$ , ensures that the incremental benefit (decrease in disutility) to an existing user upon picking up a new user is at least  $\gamma$ , where  $\gamma \geq 0$ . Imposing SIR- $\gamma$  results in a reduced set of feasible matched rides, leading not only to reduced average per-ride profit, but also a potential loss due to the “up-front discounts” (the parameter  $b$  in (5)) given to the passengers who opt for ridesharing, but end up being unmatched. However, the resulting detour-aware routing policy improves the quality of shared rides, thereby increasing the market share of ridesharing users, which could offset these negative effects. Unlike [Gopalakrishnan *et al.*, 2016] that looks at the detour-aware routing policy from the user perspective (especially fairness), in this work we answer such a scheme’s impact on the service provider’s profit.

We investigate this phenomenon by performing experiments to address the following questions:

- (a) What is the trade-off between fewer matched rides and increased market share, compared to a scenario where no SIR- $\gamma$  is imposed? How does this affect the profit?
- (b) How sensitive is the profit to the fraction of additional users who opt for ridesharing in response to the adoption of detour-aware routing by imposing SIR- $\gamma$ ?

### 5.1 Experimental Setup

As before, we use the NYC taxi trip data for our experiments. We randomly select a weekday of 2013 and a one hour time

<sup>4</sup>The difference between the average reward of a learning algorithm to that of the optimal is called *regret*. An upper bound on the regret of **UCB1** after  $T$  iterations is  $O(\log T)$  [Auer *et al.*, 2002], when each arm’s reward is a bounded random variable.

slot on that day. We then split the hour into 60 one-minute instances, and collect the sources and destinations for the users who initiated trips in each instance. The average number of users per instance is 210.85. While reporting our results, we take an average across these 60 instances.

The parameters  $c_b$ ,  $c_d$ ,  $c_t$ , and  $f_d$  are taken from publicly available Lyft user pricing and driver payment for NYC [LyftLine, 2016a; LyftLine, 2016b]. We select “detour-sensitivities”  $\alpha_i$  (higher values imply more aversion to detours) randomly from [\$0, \$5] per mile. We assume that each cab serves at most  $\zeta = 2$  users, so that the profit maximization problem can be solved optimally in polynomial time using Edmond’s algorithm [Edmonds, 1965].<sup>5</sup> The driving distance and time between locations are obtained by querying Open Trip Planner [OTP, 2016]. The parameters of the linear detour-based discount policy (5) are  $\theta = 40^\circ$  and  $b = 10\%$ . To begin with, before imposing SIR- $\gamma$ , we assume that the service provider’s initial market share is 60%, within which each user opts for ridesharing with a probability directly proportional to  $\theta$  and  $b$ , and inversely proportional to  $\alpha_i$ .

For each instance, we keep the following fixed: user origins, destinations, whether they are part of the initial market share, and if so, whether they opted for ridesharing. Then, we generate 30 realizations, where each user reacts to the adoption of a detour-aware routing policy (SIR- $\gamma$ ), by flipping biased coins parameterized by  $c_{in}$ , which is a measure of how strongly users value a quality-enhancing detour-aware routing policy. This participatory behavior is one of many that can be used. For any behavior model, our primary interest is the fraction of users who end up opting for ridesharing, and not the generative process itself. In the coin-flip-based process, the first group of users, who are in the initial market share but did not opt for ridesharing, flip with bias  $p_{in}$  that is directly proportional to  $\theta$ ,  $b$ , and  $\gamma$ , and inversely proportional to  $\alpha_i$ , with  $c_{in}$  denoting the constant of proportionality. The second group of users, who are outside the initial market share, flip with bias  $0.5p_{in}$  (the factor 0.5 is hand-picked). The users who had already opted for ridesharing remain so. For consistency, we ensure that as we increase the value of  $\gamma$ , only users who have previously not opted for ridesharing flip again.

We calculate the profit obtained by matching users in the new market share. The parameter  $c_{in}$  is varied from 100 and 1000, while  $\gamma$  is varied from 0 to 0.9.

### 5.2 Experimental Results

Figure 6 shows profitability as  $\gamma$  increases for various values of  $c_{in}$ . The baseline profit, without SIR- $\gamma$ , is shown as a horizontal line. For any value of  $c_{in} \geq 200$ , there is a range of  $\gamma$  for which a higher profit can be obtained by imposing the appropriate SIR- $\gamma$  policy. Moreover, when  $\gamma$  is high, the profit declines, because too many users remain unmatched.

Figure 7 shows a different visualization by plotting profitability as a function of  $c_{in}$ , for different values of  $\gamma$ .

Figure 8 plots the minimum increase in market share that is necessary to meet a given lower bound on the profit, as

<sup>5</sup>The experiments can be carried out for  $\zeta > 2$  as well, by using one of our proposed heuristics from Section 3.2 instead.

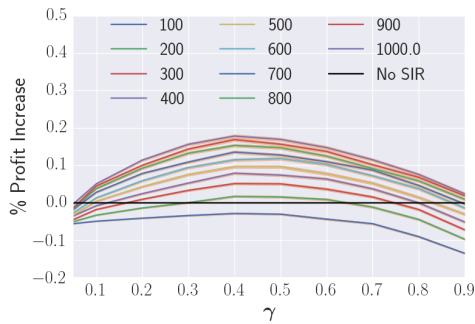


Figure 6: Profit versus  $\gamma$ .

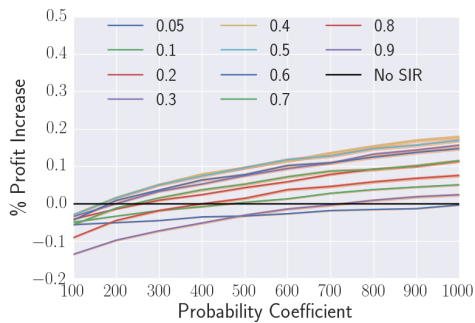


Figure 7: Sensitivity of profit to  $c_{in}$ .

a function of  $\gamma$ . For instance, to be 5% more profitable, a 13% increase is needed, as shown by the green curve. Alternatively, Figure 9 shows the maximum possible increase in profit (over all  $\gamma$ ), as a function of the increase in market share. The key takeaway here is that a 7% increase<sup>6</sup> in market share is sufficient to recover from the negative effects (on the profit) of adopting a detour-aware routing policy.

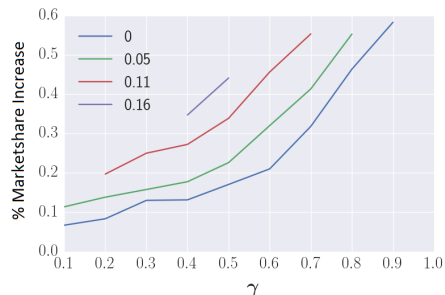


Figure 8: Marketshare versus  $\gamma$  for a given profit lower bound.

## 6 Concluding Remarks

In this paper, we adopt a profit-centric view of real-time ridesharing system design, and undertake an empirical investigation of three key elements that impact profit, namely

<sup>6</sup>A 7% increase in market share in response to a quality-enhancing policy is not unrealistic. Lyft’s market share increased by over 70% in 2016, owing in large part to better quality of service, resulting in Lyft’s customers being happier than those of their primary competitor, Uber [Swan, 2016].

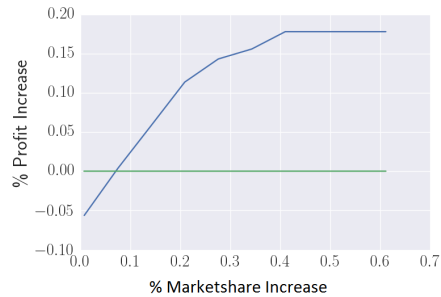


Figure 9: Profit versus marketshare increase.

(a) real-time ride matching, (b) user discounts, and (c) detour-aware routing, by using NYC taxi trip data.

An optimal ride matching algorithm, even on a small set of 22 requests, can take up to 2 hours, however, our greedy heuristic generates near-optimal profit in less than  $1/10^5$  of the time, and scales well — it is able to match 18794 trip requests into 6701 cabs — while offering different trade-offs between optimality and running time. Our adaptive learning algorithm called **IDFLA** learns the optimal detour-based discount parameter that achieves the right balance between the number of passengers who opt for ridesharing and the portion of savings taken as profit from each shared ride, within 16 iterations. Finally, we demonstrate that even a small market share increase of 7%, in response to other restricted detour-aware policies such as sequential individual rationality, is sufficient to counter the reduced per-ride profit due to the restricted constraints. Thus, one of our contributions is also a holistic treatment of the framework for detour-aware policies.

Given the encouraging empirical results of our proposed solutions for profit optimization, future work should provide approximation guarantees for the proposed greedy ride-matching heuristics, and a formal mathematical analysis to characterize the dependence between profit under a detour-aware routing policy. Another interesting direction to explore is the interplay between these different elements, e.g., how does a detour-aware routing policy affect the performance of the ride-matching heuristics and the convergence of the learning algorithm?

## References

- [Agatz *et al.*, 2011] Niels AH Agatz, Alan L Erera, Martin WP Savelsbergh, and Xing Wang. Dynamic ride-sharing: A simulation study in metro atlanta. *Transportation Research Part B: Methodological*, 45(9):1450–1464, 2011.
- [Agatz *et al.*, 2012] Niels Agatz, Alan Erera, Martin Savelsbergh, and Xing Wang. Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research*, 223(2):295–303, 2012.
- [Auer *et al.*, 2002] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.
- [Banerjee *et al.*, 2015] Siddhartha Banerjee, Ramesh Johari, and Carlos Riquelme. Pricing in ride-sharing platforms:

- A queueing-theoretic approach. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, pages 639–639, 2015.
- [Bistaffa *et al.*, 2015] Filippo Bistaffa, Alessandro Farinelli, Georgios Chalkiadakis, and Sarvapali Ramchurn. Recommending fair payments for large-scale social ridesharing. In *Proceedings of the 9th ACM Recommender Systems Conference (RecSys)*, pages 139–146, 2015.
- [Biswas *et al.*, 2017] Arpita Biswas, Ragavendran Gopalakrishnan, Theja Tulabandhula, Koyel Mukherjee, Asmita Metrewar, and Raja Subramaniam Thangaraj. Profit optimization in commercial ridesharing. In *Proceedings of the 16th Conference on Autonomous Agents and Multi-Agent Systems*, pages 1481–1483. International Foundation for Autonomous Agents and Multiagent Systems, 2017.
- [Bubeck and Cesa-Bianchi, 2012] Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems, 2012. arXiv preprint arXiv:1204.5721.
- [Cao *et al.*, 2015] Bin Cao, Louai Alarabi, Mohamed F Mokbel, and Anas Basalamah. SHAREK: A scalable dynamic ride sharing system. In *Proceedings of the 16th IEEE International Conference on Mobile Data Management (MDM)*, pages 4–13, 2015.
- [Cordeau *et al.*, 2006] Jean-François Cordeau, Gilbert Laporte, Martin WP Savelsbergh, and Daniele Vigo. Vehicle routing. In *Handbooks in operations research and management science: Transportation*, volume 14, chapter 6, pages 367–428. Elsevier, 2006.
- [Edmonds, 1965] Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of mathematics*, 17(3):449–467, 1965.
- [Furuhata *et al.*, 2013] Masabumi Furuhata, Maged Dessouky, Fernando Ordóñez, Marc-Etienne Brunet, Xiaoqing Wang, and Sven Koenig. Ridesharing: The state-of-the-art and future directions. *Transportation Research Part B: Methodological*, 57:28–46, 2013.
- [Geisberger *et al.*, 2010] Robert Geisberger, Dennis Luxen, Sabine Neubauer, Peter Sanders, and Lars Volker. Fast detour computation for ride sharing. In *Proceedings of the 10th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems, OASiCS-OpenAccess Series in Informatics*, volume 14, pages 88–99. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2010.
- [Gopalakrishnan *et al.*, 2016] Ragavendran Gopalakrishnan, Koyel Mukherjee, and Theja Tulabandhula. The costs and benefits of ridesharing: Sequential individual rationality and sequential fairness. *CoRR*, abs/1607.07306, 2016.
- [Jung *et al.*, 2013] Jaeyoung Jung, R Jayakrishnan, and Ji Young Park. Design and modeling of real-time shared-taxi dispatch algorithms. In *Proc. Transportation Research Board 92nd Annual Meeting*, 2013.
- [Kamar and Horvitz, 2009] Ece Kamar and Eric Horvitz. Collaboration and shared plans in the open world: Studies of ridesharing. In *IJCAI*, pages 187–194, 2009.
- [Kleiner *et al.*, 2011] Alexander Kleiner, Bernhard Nebel, and Vittorio Amos Ziparo. A mechanism for dynamic ride sharing based on parallel auctions. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, pages 266–272, 2011.
- [LyftLine, 2016a] LyftLine. Lyft Line Driver Pay. <https://help.lyft.com/hc/en-us/articles/213582268-Lyft-Line-Driver-Pay>, 2016.
- [LyftLine, 2016b] LyftLine. Lyft Line Pricing. <https://help.lyft.com/hc/en-us/articles/213815178-Lyft-Line-Pricing>, 2016.
- [Nguyen, 2013] Duc Thien Nguyen. *Fair cost sharing auction mechanisms in last mile ridesharing*. PhD thesis, Singapore Management University, 2013.
- [OTP, 2016] Open Trip Planner. <https://www.opentripplanner.org>, 2016.
- [Pelzer *et al.*, 2015] Dominik Pelzer, Jiajian Xiao, Daniel Zehe, Michael H Lees, Alois C Knoll, and Heiko Aydt. A partition-based match making algorithm for dynamic ridesharing. *IEEE Transactions on Intelligent Transportation Systems*, 16(5):2587–2598, 2015.
- [Rajasubramaniam *et al.*, 2017] T. Rajasubramaniam, K. Mukherjee, G. Raravi, A. Metrewar, N. Annamaneni, and K. Chattopadhyay. Xhare-a-ride: A search optimized dynamic ride sharing system with approximation guarantee. In *IEEE International Conference on Data Engineering*, 2017.
- [Swan, 2016] Andy Swan. Sorry, Uber. Social Data Validates The Lyft Growth Story And Valuation Demands. <http://www.forbes.com/sites/andyswan/2016/08/31/sorry-uber-social-data-validates-the-lyft-growth-story-and-valuation-demands/>, 2016.
- [Whong and Monroy-Hernández, 2013] Chris Whong and Andrés Monroy-Hernández. New York City taxi trips. <http://www.andresmh.com/nyctaxitrips>, 2013. Last accessed: 2016-06-18.
- [Zhao *et al.*, 2014] Dengji Zhao, Dongmo Zhang, Enrico H Gerding, Yuko Sakurai, and Makoto Yokoo. Incentives in ridesharing with deficit control. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems*, pages 1021–1028, 2014.