Book-Ahead & Supply Management for Ridesourcing Platforms

Cesar N. Yahia[a,*], Gustavo de Veciana[b], Stephen D. Boyles[a], Jean Abou Rahal[b], Michael Stecklein[b]

[a]*Department of Civil, Architectural and Environmental Engineering, The University of Texas at Austin*
[b]*Department of Electrical and Computer Engineering, The University of Texas at Austin*

Abstract

Ridesourcing platforms recently introduced the "schedule a ride" service where passengers may reserve (book-ahead) a ride in advance of their trip. Reservations give platforms precise information that describes the start time and location of anticipated future trips; in turn, platforms can use this information to adjust the availability and spatial distribution of the driver supply. In this article, we propose a framework for modeling/analyzing reservations in time-varying stochastic ridesourcing systems. We consider that the driver supply is distributed over a network of geographic regions and that book-ahead rides have reach time priority over non-reserved rides. First, we propose a state-dependent admission control policy that assigns drivers to passengers; this policy ensures that the reach time service requirement would be attained for book-ahead rides. Second, given the admission control policy and reservations information in each region, we predict the "target" number of drivers that is required (in the future) to probabilistically guarantee the reach time service requirement for stochastic non-reserved rides. Third, we propose a reactive dispatching/rebalancing mechanism that determines the adjustments to the driver supply that are needed to maintain the targets across regions. For a specific reach time quality of service, simulation results using data from Lyft rides in Manhattan exhibit how the number of idle drivers decreases with the fraction of book-ahead rides. We also observe that the non-stationary demand (ride request) rate varies significantly across time; this rapid variation further illustrates that time-dependent models are needed for operational analysis of ridesourcing systems.

Keywords: ride-hailing, book-ahead, reservation, admission control, supply management

1. Introduction

Recent growth of ridesourcing services is further exacerbating fleet management challenges associated with dynamic and spatially asymmetric passenger demands. Ridesourcing platforms (e.g., Uber and Lyft) need to locate a sufficient number of drivers near anticipated passenger demand to reduce the reach time (i.e., the customer wait time between ride request and the arrival of a driver). However, an abundance of drivers may lead to increased driver idle time. Thus, with the objective of guaranteeing low customer waiting times and low driver idle time, the following

[*] Corresponding author
E-mail addresses: cesaryahia@utexas.edu (C.N. Yahia), gustavo@ece.utexas.edu (G. de Veciana), sboyles@mail.utexas.edu (S.D. Boyles), jeanabourahal@utexas.edu (J.A. Rahal), michaelrstecklein@gmail.com (M. Stecklein)

questions arise: how many drivers should a ridesourcing platform supply?, and, how should the platform spatially manage idle drivers based on anticipated demand?

In this article, the primary objective is to investigate the role of book-ahead/reserved rides in the management of driver supply. Reservations give precise information characterizing the start time and location of anticipated trips; in turn, the platform can use this information to adjust the availability and spatial distribution of its driver supply. Thus, given a reach time service requirement that the platform seeks to maintain, we analyze the impact of reservations on the number of drivers supplied throughout the network. Moreover, since passengers that schedule a ride in advance expect the driver to arrive within a desired pickup window, our analysis incorporates such priority of book-ahead rides over non-reserved rides.

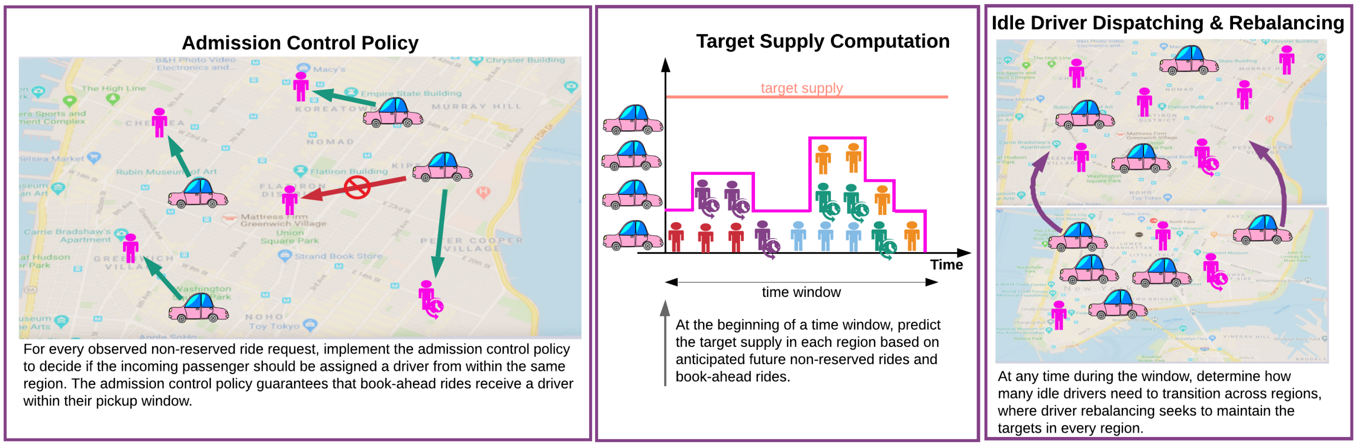
In practice, ridesourcing platforms have several control levers that they can use to manage driver supply. These levers include earning guarantees for new drivers, bonuses, and heat maps that show high demand locations where drivers earn more due to surge pricing (Lyft, 2019a,c). In addition, as implemented by Lyft in New York City, platforms can restrict the number of active drivers or force them to drive towards high demand areas if they wish to remain online (Lyft, 2019b).

The proposed supply management framework parallels existing research on ridesourcing systems (Djavadian and Chow, 2017; Lei et al., 2019; Wang and Yang, 2019). The majority of existing studies assume a fixed number of driver supply and/or steady-state (equilibrium) conditions. However, it is increasingly apparent that demand and supply patterns in ridesourcing systems are time-varying. In addition, these variations in demand and supply occur at a fast pace, and the system may never attain a steady state equilibrium.

Thus, our proposed framework for analyzing reservations in ridesourcing systems focuses on the *transient* nature of time-varying stochastic demand/supply patterns. Precisely, for any future point in time, we seek to probabilistically characterize the total number of active (non-idle) drivers; this time-dependent probabilistic characterization is determined by the fraction of book-ahead rides, the stochasticity of non-reserved rides, the anticipated time-varying profile of book-ahead rides, and control policies that aim to maintain reach time priority for book-ahead rides. In more detail, as shown in Figure 1, the proposed framework consists of the following three components for managing driver supply:

1. We develop a state-dependent admission control policy that assigns drivers to passengers. The objective of this control policy is to guarantee the reach time service requirement for book-ahead rides. Effectively, the admission control policy ensures that there is a sufficient number of drivers near the location of anticipated book-ahead rides such that the driver can reach the passenger within the pickup window.

2. Given this admission control policy and reservations information, we predict the "target" number of drivers that is required (in the future) to *probabilistically* guarantee the reach time service requirement for stochastic non-reserved rides. The target computations are derived


Admission Control Policy
Target Supply Computation
target supply
Time
time window
Idle Driver Dispatching & Rebalancing

For every observed non-reserved ride request, implement the admission control policy to decide if the incoming passenger should be assigned a driver from within the same region. The admission control policy guarantees that book-ahead rides receive a driver within their pickup window.

At the beginning of a time window, predict the target supply in each region based on anticipated future non-reserved rides and book-ahead rides.

At any time during the window, determine how many idle drivers need to transition across regions, where driver rebalancing seeks to maintain the targets in every region.

Figure 1: Proposed framework for assigning drivers to passengers to guarantee the arrival of drivers to book-ahead rides within the pickup window, computing the target supply, and rebalancing drivers across regions to maintain the targets.

from an upper bound on the time-dependent probability that a non-reserved ride will experience waiting times in excess of the reach time service requirement, and this upper bound can be evaluated using transient analysis of $M_t/GI/\infty$ queues.

3. We develop a minimum cost flow driver dispatching/rebalancing mechanism that seeks to maintain the targets across regions. In particular, due to the transition of drivers across geographic regions and the associated passenger demand patterns, the driver supply in a specific region may deviate from the predicted target. Thus, the proposed minimum cost flow mechanism determines the adjustments to the driver supply that are needed to maintain the targets throughout the network.

The remainder of this article proceeds as follows: In Section 2 we review related work addressing the operation of ridesourcing systems. Section 3 describes the proposed model for analyzing time-dependent ridesourcing dynamics. Section 4 presents the admission control policy. Section 5 derives an upper bound on the performance of the admission control policy and computes the target supply. Section 6 presents the driver dispatching/rebalancing mechanism. Section 7 exhibits simulation results using data from Lyft operations in Manhattan. Section 8 concludes the article.

2. Related Work

Ridesourcing platforms are aggressively implementing supply and demand management strategies that drive their expansion into new markets (Nie, 2017). These strategies can be broadly classified into one or more of the following categories: pricing, fleet sizing, empty vehicle routing (rebalancing), or matching passengers to drivers. Apart from increasing their market share, platforms seek to improve their operational efficiency by minimizing the spatio-temporal mismatch between supply and demand (Zuniga-Garcia et al., 2020). In this section, we provide a brief survey of existing methods that are used to analyze the operations of ridesourcing platforms.

2.1. *Equilibrium analysis of ridesourcing systems*

The majority of existing studies on ridesourcing systems focus on analyzing interactions between driver supply and passenger demand under *static* equilibrium conditions. These studies seek to evaluate the market share of ridesourcing platforms, competition among platforms, and the impact of ridesourcing platforms on traffic congestion (Bahat and Bekhor, 2016; Ban et al., 2019; Di and Ban, 2019; Qian and Ukkusuri, 2017; Wang et al., 2018). In addition, following Yang and Yang (2011), researchers examined the relationship between customer wait time, driver search time, and the corresponding matching rate at market equilibrium (Xu et al., 2019; Zha et al., 2016). Recently, Di et al. (2018) incorporated ridesharing user equilibrium in a network design problem; Zha et al. (2018) proposed an equilibrium model to investigate the impact of surge pricing on driver work hours; Zhang and Nie (2019) studied passenger pooling under market equilibrium for different platform objectives and regulations; and Rasulkhani and Chow (2019) generalized a static many-to-one assignment game that finds equilibrium through matching passengers to a set of routes. While static equilibrium analysis provides valuable strategic decision-making insights, it fails to address stochasticity and time-dependence in ridesourcing dynamics.

2.2. *Steady state analysis of stochasticity in ridesourcing systems*

To investigate stochasticity in demand/supply management, researchers have developed queueing theoretic models for ridesourcing systems. In particular, closed queueing networks were used to analyze rebalancing and pricing policies (Banerjee et al., 2017; Braverman et al., 2019; Zhang and Pavone, 2016). In these closed queueing networks, the difficulty in designing supply management strategies arises from equilibrium (steady-state) constraints that result in high dimensional non-convex problems (Banerjee et al., 2017). Other queueing based approaches include a double-ended queue to characterize stochasticity in matching (Xu et al., 2019) and an M/G/N queue where each driver is considered to be a server (Li et al., 2019). Spatial stochasticity associated with matching was also investigated using Poisson processes to describe the distribution of drivers near a passenger (Chen et al., 2019; Zhang et al., 2019; Zhang and Nie, 2019). The previously mentioned studies focus on steady-state (equilibrium) analysis that disregards the time-dependent variability in demand/supply patterns. Furthermore, temporal variations in demand/supply patterns may occur rapidly, and the system may not attain the steady-state equilibrium conditions (Braverman et al., 2019; Ozkan and Ward, 2019). In addition, policies generated from steady-state optimization in closed queueing networks are open-loop (static); this implies that the policies do not react to the time-dependent stochastic state of the system.

2.3. *Time-varying ridesourcing dynamics*

The importance of time dynamics has been emphasized in recent articles that design time-dependent demand/supply management strategies (Ramezani and Nourinejad, 2018). Wang et al. (2019) proposed a dynamic user equilibrium approach for determining the optimal time-varying driver compensation rate. Similarly, Nourinejad and Ramezani (2019) developed a dynamic model to study pricing strategies; their model allows for pricing strategies that incur losses to the platform over short time periods (driver wage greater than trip fare), and they emphasized that time-invariant static equilibrium models are not capable of analyzing such policies. An alternative dynamic model was proposed by Daganzo and Ouyang (2019); however, the authors focus on the steady-state performance of their model. While these models can be used to analyze time-dependent policies, the authors do not explicitly consider the spatio-temporal stochasticity that results in the mismatch between supply and demand.

2.4. *Analysis of stochasticity in time-varying ridesourcing dynamics*

The most common approach for analyzing time-dependent stochasticity in ridesourcing systems is to apply steady-state probabilistic analysis over fixed time intervals. However, in the context of driver rebalancing, experimental analysis by Braverman et al. (2019) suggests that the time needed to converge to steady-state (equilibrium) in ridesourcing systems is on the order of 10 hours. Thus, since parameters (e.g., passenger arrival rate) vary over much shorter time intervals, the system would not reach the steady-state condition. Subsequently, Braverman et al. (2019) proposed a time-dependent look-ahead policy that can be used to make rebalancing decisions at any point in time. Recent studies that addressed operational challenges in ridesourcing systems also advocate for transient analysis instead of steady-state models (Nourinejad and Ramezani, 2019; Ozkan and Ward, 2019).

Another limitation of steady-state policies is that they are independent of the system state. In particular, those policies are based on probabilistic predictions over entire time intervals, and they do not react to the stochastic system state that is realized at a specific time within the time interval. In contrast, state-dependent policies react to the observed fluctuations in the stochastic system state (Banerjee et al., 2018).

Our study falls into this category of analyzing time-dependent stochasticity in ridesourcing systems.

- First, we propose a state-dependent admission control policy that reacts to the observed ride requests and available driver supply. This admission control policy ensures that the reach time service requirement is attained for book-ahead rides by choosing which driver to assign to every realized non-reserved ride request.

- Second, in a predictive approach over an upcoming time-interval, we provide an upper bound on the performance of the state-dependent admission control policy; precisely, the performance of the policy is measured in terms of the probability that the reach time service requirement would be violated for a non-reserved ride. In contrast to steady-state methods, we use *transient* analysis of $\mathrm{M}_t/\mathrm{GI}/\infty$ to determine the aforementioned upper bound at any point in time throughout the window. In other words, we derive a time-dependent upper bound on the probability of reach time violation for non-reserved rides. Subsequently, we use the time-averaged value of the upper bound to compute the "target" number of drivers that is required during the upcoming time window; thus, this target limits the probability of reach time service violation to be within a desired performance level.

- Third, we propose another reactive state-dependent policy for dispatching/rebalancing drivers across multiple regions. Given the predicted “target” supply for an upcoming time window, the minimum cost flow dispatching/rebalancing policy seeks to maintain the targets across multiple regions. For a specific system state at some time within the time window, the dispatching/rebalancing mechanism determines the number of idle drivers that should transition to adjacent regions to maintain the targets.

3. System Model

In this section, we describe a general model for time-varying dynamics in ridesourcing systems. The proposed model represents the number of future *active* rides that initiate in a region. A ride/driver is active from the moment the driver is dispatched to pick up the passenger until the trip is completed. For non-reserved rides, the ride becomes active at the same time as the request is initiated. On the other hand, for book-ahead rides, there is a lag between the time that the request is initiated and the time that the drivers is dispatched to pick up the passenger. While active, drivers are associated with the passenger and can not take on other requests. The ride duration (service time) is the time spent while the driver is active which includes the pick up time. A ride starts when the driver becomes active and ends when the driver is idle again.

The active rides are represented over a set of geographic regions $R = \{1, .., m\}$. These regions are sufficiently small that if a ride request initiates in a region and the assigned driver is operating in the same region, then the reach time is within a desired service level. In other words, if we want the reach time to be under 10 minutes, then the time it takes to drive from any point to any other point within the defined region should be under 10 minutes.

Consequently, we incorporate reservations by providing reach-time priority for book-ahead rides. In particular, for a driver to arrive within the book-ahead ride pickup window, the driver must be geographically close to the passenger at the anticipated trip start time. Thus, we consider that book-ahead ride requests must be assigned a driver from within the same region in which the request initiates, and that satisfying the reach time service requirement for book-ahead rides is equivalent to a driver arriving to the passenger within the pickup window. In Section 4, we design an admission control policy that guarantees that book-ahead rides will be assigned a driver from within the same region.

In the proposed ridesourcing model, we do not explicitly analyze ridesharing (i.e., passenger pooling); however, the predicted number of active rides would be a conservative estimate on the corresponding value in ridesharing systems. Furthermore, for tractable target computations, we examine each region separately. In other words, the admission control and corresponding targets assume passengers remain within the zone, disregarding the variation in destinations. Then, to account for the spatial distribution of passenger destinations and the associated movement of drivers across regions, we implement a min-cost flow rebalancing methods that maintains the targets across regions. Note that the targets themselves represent a desired number of drivers that is

Table 1: Table of Notation & Definitions

active driver	$\triangleq$	drivers are active from the moment they are dispatched to pick up a passenger and until the passenger leaves the vehicle
idle driver	$\triangleq$	driver waiting to be dispatched (not active)
ride initiation/start	$\triangleq$	time driver is dispatched to pick up passenger
ride completion	$\triangleq$	time passenger leaves vehicle
ride duration	$\triangleq$	total time while driver is active (includes pick up time)
R	$\triangleq$	set of regions $\{1, .., r, .., m\}$
window k	$\triangleq$	time window $(kw, (k+1)w]$
w	$\triangleq$	duration of time window
c_r^k	$\triangleq$	target number of drivers in region r during window k that would probabilistically guarantee a desired reach time service level
$f_r^{P,k}(t)$	$\triangleq$	deterministic process representing active drivers at time $t \in (kw, (k+1)w]$ that are serving requests which initiated in r during *previous* time windows
$f_r^{BA,k}(t)$	$\triangleq$	deterministic process representing active drivers at time $t \in (kw, (k+1)w]$ that are associated with *book-ahead* trips that initiate within window $(kw, (k+1)w]$ in region r
$N_r^k(t)$	$\triangleq$	stochastic process representing active drivers at time $t \in (kw, (k+1)w]$ that are associated with *admitted* stochastic non-reserved rides that initiate within window $(kw, (k+1)w]$ in region r
$\lambda_r^k(t)$	$\triangleq$	demand rate at which stochastic non-reserved ride requests initiate during window k in region r
$g_r^k(\cdot)$	$\triangleq$	probability density function characterizing the ride duration (completion time - trip request time) of stochastic non-reserved rides that appear during window k in region r
$G_r^k(\cdot)$	$\triangleq$	cumulative density function of $g_r^k(\cdot)$
$f_r^{A(\tau_i),k}(t)$	$\triangleq$	active drivers at time $t \in (\tau_i, \min\{\tau_i + D_i, (k+1)w\}]$ corresponding to non-reserved rides that were *previously admitted* between $(kw, \tau_i]$ in region r
τ_i	$\triangleq$	arrival time of the i^{th} non-reserved ride request
D_i	$\triangleq$	ride duration of the i^{th} non-reserved ride
γ_i	$\triangleq$	indicator function/random variable characterizing the event that the i^{th} non-reserved ride request is admitted
B_r^k	$\triangleq$	average blocking probability during window k in region r
δ	$\triangleq$	desired reach time quality of service for non-reserved rides (upper bound on the average blocking probability)
$N_r^{k,\infty}(t')$	$\triangleq$	number of busy servers at time $t' \in (0, w]$ in a transient $\mathrm{M}_t/\mathrm{GI}/\infty$ queue that starts empty at $t' = 0$; equivalently, the number of active non-reserved rides assuming that all stochastic non-reserved requests are admitted
$\rho_r^k(t')$	$\triangleq$	time-dependent mean/variance of the Poisson distribution characterizing $N_r^{k,\infty}(t')$ at time $t' \in (0, w]$
a_r	$\triangleq$	number of active drivers in region r
e_r	$\triangleq$	number of idle drivers in region r
s_r^v	$\triangleq$	virtual supply in region r representing drivers in excess of the target c_r^k that can be removed from region r
d_r^v	$\triangleq$	virtual demand in region r representing drivers that should be added to region r to meet the target c_r^k
Δ_r	$\triangleq$	if region r has virtual demand, then $\Delta_r = -d_r^v$; otherwise, if the region has virtual supply, then $\Delta_r = s_r^v$
h_{ij}	$\triangleq$	recommended driver transitions between region i and j
$\mathbf{1}\{\cdot\}$	$\triangleq$	indicator function or random variable

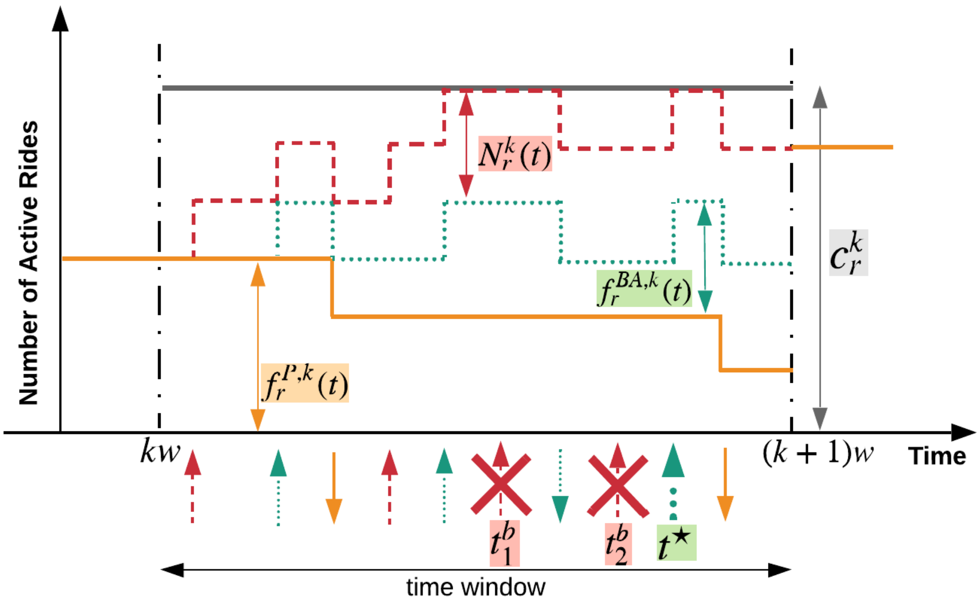


Figure 2: System model characterizing the *cumulative* number of rides that will be active in the future at time $t \in (kw, (k+1)w]$. Arrows pointing upwards indicate ride start time. Arrows pointing downwards indicate ride completion. Solid lines correspond to $f_r^{P,k}(t)$, dotted lines correspond to $f_r^{BA,k}(t)$, and dashed lines correspond to $N_r^k(t)$. Non-reserved requests marked with an "X" are blocked requests.

determined by passenger demand; this implies that the targets do not depend on the stochasticity of drivers entering and exiting the system.

We proceed by describing the model for active rides in each region. For each region, this model consists of processes representing book-ahead rides and non-reserved stochastic rides. The processes form the basis of subsequent sections that discuss the admission control policy and the computation of targets.

3.1. *Time-varying profiles representing rides that will be active in the future*

In each region $r \in R$, we represent ridesourcing dynamics over future time windows of length w. At the beginning of each window k, corresponding to time interval $(kw, (k+1)w]$, the ridesourcing platform can characterize three processes (two deterministic and one stochastic) that will be realized during the upcoming window $(kw, (k+1)w]$. The processes represent *active* drivers at time $t \in (kw, (k+1)w]$ that are serving requests initiated within the region.

First, we assume that the platform knows the anticipated start time for *book-ahead* rides that will initiate during window k. We also assume that the platform can accurately estimate the corresponding ride duration (i.e. the platform has full trip information for future book-ahead rides). Thus, at the start of window k, the platform can characterize the *deterministic* process $\{f_r^{BA,k}(t) : t \in (kw, (k+1)w]\}$ that represents the number of active drivers at time t associated with book-ahead trips that will initiate in region r within window k.

Second, at the beginning of time window $(kw, (k+1)w]$, currently active drivers serving rides that started in region r prior to time $t = kw$ are known to the platform. For those *previously*

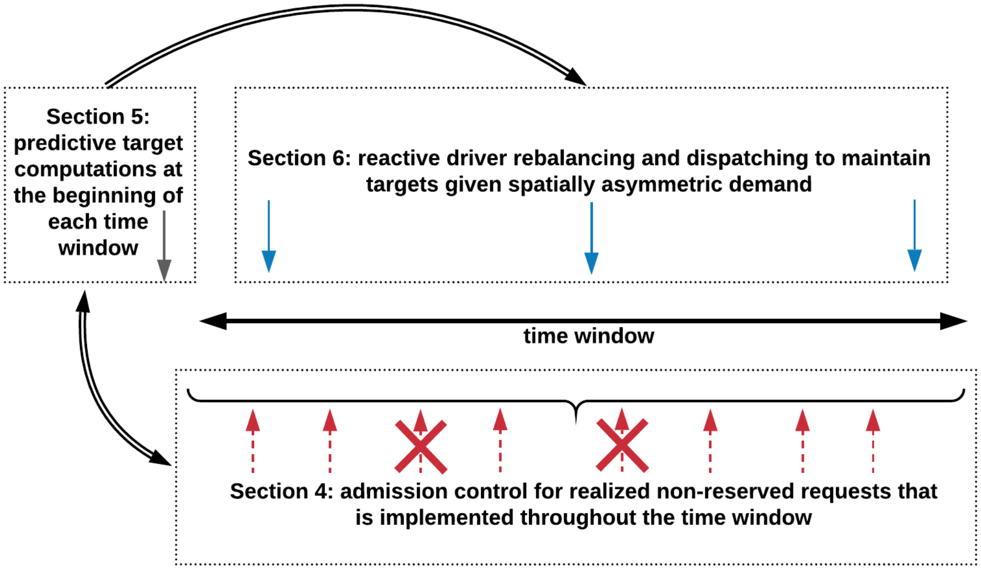

Section 5: predictive target computations at the beginning of each time window
Section 6: reactive driver rebalancing and dispatching to maintain targets given spatially asymmetric demand
time window
Section 4: admission control for realized non-reserved requests that is implemented throughout the time window

Figure 3: Implementation of the proposed framework across time windows.

observed trips, we assume that the platform can accurately estimate the trip completion time. Thus, at the start of window k, the platform can characterize the deterministic process $\{f_r^{P,k}(t) : t \in (kw,(k+1)w]\}$. This process represents the number of active drivers at time t that are serving rides started in region r during previous time windows. In other words, those are previously observed rides that haven't ended yet and may correspond to either passenger type (book-ahead or non-reserved).

Third, at the beginning of window k, the platform also anticipates *non-reserved* stochastic rides that will arise throughout the upcoming window in region r. For those rides, we assume that the platform can estimate the demand (ride request) rate $\{\lambda_r^k(t) : t \in (kw,(k+1)w]\}$. We also assume that the platform can estimate a general distribution $g_r^k(\cdot)$ that corresponds to the ride duration (the CDF of $g_r^k(\cdot)$ is $G_r^k(\cdot)$), and we consider that the duration of any specific non-reserved trip is independent of other trips. Then, we define a stochastic process $\{N_r^k(t) : t \in (kw,(k+1)w]\}$ that represents the number of active drivers at time t associated with *admitted* stochastic rides which initiate in region r during window k. In this case, a non-reserved ride request would be admitted if it is assigned a driver from within the same region.

The deterministic processes $\{f_r^{P,k}(t), f_r^{BA,k}(t) : t \in (kw,(k+1)w]\}$ and the stochastic process $\{N_r^k(t) : t \in (kw,(k+1)w]\}$ are illustrated in Figure 2. The figure shows the *cumulative* number of active drivers at time $t \in (kw,(k+1)w]\}$.

The next section describes the admission control policy that decides whether to admit non-reserved rides based on the difference between the predicted targets and the number of active drivers. The admission control policy is state-dependent such that the admission decision is determined for each ride request once the request is observed. In more detail, the admission decision depends on the current *known* state of the system for the entire duration that the observed ride will be active. Given this policy, we discuss in Section 5 how the targets are evaluated at the beginning of the window. However, to compute the targets, we refer to the *predicted* future system state under the control policy, and we resort to a probabilistic characterization of the anticipated non-reserved rides (i.e., we further analyze the stochastic process $\{N_r^k(t) : t \in (kw, (k+1)w]\}$). In other words, the admission control policy uses the targets in determining the *deterministic* admission decisions while the targets are evaluated using the predicted *stochastic* system state that will arise under the control policy. Then, in Section 6, we present the driver dispatching and rebalancing mechanism that maintains the targets given the *observed* demand patterns. Figure 3 illustrates the relationship between different components of this article and the time at which those components would be implemented.

4. Admission Control Policy

In this section, we present an admission control policy that is used to assign drivers to realized non-reserved ride requests. In each region, when a non-reserved ride request is observed, the proposed state-dependent control policy determines whether the request should be *admitted* or *blocked*. If the request is admitted, then a driver from within the same region is assigned to serve the request.

The admission decision is based on the supply in the region, the anticipated book-ahead rides, and the previously admitted non-reserved rides. In particular, the policy seeks to guarantee that a driver from within the same region would be available to serve anticipated future book-ahead rides. Thus, admission control aims to guarantee that drivers arrive within the pickup window for future book-ahead rides. Since the same policy is implemented for each region, we restrict our discussion in this section to a single region $r \in R$.

At any time $t \in (kw, (k+1)w]$, the admission control policy determines if idle drivers will be available in the region by comparing the number of active rides to the *target supply* c_r^k. The target supply c_r^k, illustrated in Figure 2, is the total number of drivers associated with region r during window k; this total includes drivers that are serving ride requests initiated in region r *and* drivers idling in region r. The target c_r^k represents a desired level of driver supply that would probabilistically guarantee the reach time service requirement for non-reserved rides (Section 5). The admission control policy assumes that the targets c_r^k will be maintained in each region r throughout the time window k. For tractable computation, the admission control policy also assumes that the passengers destinations remain within the region (in Section 6, we devise a driver dispatching/rebalancing mechanism that considers the spatial distribution of demand and seeks to maintain the target across regions).

4.1. *Policy Implementation*

A non-reserved ride request is admitted if, upon admission, the total number of active rides does not exceed the target supply for the entire ride duration. Once a non-reserved ride request is observed, the associated ride duration would be also revealed to the platform. Then, there are two cases where the admission control policy would *block* the non-reserved ride request: (1)

There are not enough available drivers within the region at the time of request initiation; this is illustrated in Figure 2 at time t_1^b, where the sum $N_r^k(t_1^b) + f_r^{BA,k}(t_1^b) + f_r^{P,k}(t_1^b)$ is equal to the target c_r^k. In other words, admission of the non-reserved ride would result in the total number of active rides *exceeding* the target supply at the time of request initiation. (2) Admission of the non-reserved ride would result in reach time service violation for an anticipated book-ahead ride; in Figure 2, admission of the non-reserved ride request that initiates at time t_2^b would lead to reach time violation for the book-ahead trip that initiates at $t^\star$ (considering that the observed ride duration of the request that initiates at t_2^b extends beyond $t^\star$). In other words, if the non-reserved ride was admitted at t_2^b, then at $t^\star$ (just before the book-ahead request is anticipated) the sum $N_r^k(t^\star) + f_r^{BA,k}(t^\star) + f_r^{P,k}(t^\star)$ would be equal to the target supply c_r^k; this implies that the total number of active rides would *exceed* the target supply when the book-ahead ride at $t^\star$ starts (equivalently, the book-ahead ride would not be assigned a driver from within the same region).

In more detail, let τ_i be the arrival time of the i^{th} non-reserved ride request, and let D_i be the corresponding ride duration. In addition, let γ_i be an indicator function that takes the value one if the i^{th} non-reserved ride request is admitted. Equation 1 gives the expression for γ_i (i.e., Equation 1 represents the condition for admission). In Equation 1, $f_r^{A(\tau_i),k}(t)$ represents *previously admitted* non-reserved rides that would be active at time $t \in (\tau_i, \min\{\tau_i + D_i, (k+1)w\}]$. In other words, $f_r^{A(\tau_i),k}(t)$ represents previously admitted non-reserved rides that would be active during the time that the i^{th} non-reserved ride request is being served. Note that the projected ride duration of the i^{th} non-reserved user is restricted to $t \in (\tau_i, \min\{\tau_i + D_i, (k+1)w\}]$ instead of $t \in (\tau_i, \tau_i + D_i]$ since admission control decisions are made per window k (i.e., the rides whose duration extends beyond $t = (k+1)w$ would become part of $f_r^{P,k+1}(t)$).

$$\gamma_i = \mathbf{1}\left\{1 + f_r^{P,k}(t) + f_r^{BA,k}(t) + f_r^{A(\tau_i),k}(t) \leq c_r^k, \quad \forall t \in (\tau_i, \min\{\tau_i + D_i, (k+1)w\}]\right\} \tag{1}$$

If we let τ_n and D_n be the arrival time and ride duration of the n^{th} previously observed non-reserved ride (where $n \in \{1, ..., i-1\}$), we can express $f_r^{A(\tau_i),k}(t)$ as shown in Equation 2. In this equation, $\mathbf{1}\{\tau_n + D_n > t\}$ takes the value one if the n^{th} previously observed non-reserved ride would be active at time t, and γ_n takes the value one if the n^{th} non-reserved request was admitted.

$$f_r^{A(\tau_i),k}(t) = \sum_{n=1}^{i-1} \mathbf{1}\{\tau_n + D_n > t\}\gamma_n, \quad t \in (\tau_i, \min\{\tau_i + D_i, (k+1)w\}] \tag{2}$$

We emphasize that the control policy is state-dependent and applied upon the receipt of each ride request; this implies that the state of the system is deterministic and all the variables (including $\tau_n, D_n, \gamma_n, f_r^{A(\tau_i),k}(t), \tau_i, D_i, \gamma_i$) are known at time τ_i. Then, the admission decision for the i^{th} non-reserved user follows directly from evaluating expressions 1 and 2.

A non-reserved ride request that is blocked may be assigned a driver from an external region (i.e., the passenger will experience a long wait time). Alternatively, blocked non-reserved requests may be dropped from the system, where this indicates a passenger canceling the ride due to the extended wait time. In the simulation experiments (Section 7), we follow the latter approach.

5. Target Supply for Probabilistically Guaranteeing the Reach Time Quality of Service

While the admission control policy is a state-dependent policy that is applied during the time window $(kw, (k+1)w]$, it is based on the target supply c_r^k that is determined at the beginning of the time window $t = kw$. For a specific region r, the target c_r^k represents the total number of drivers that is required during window k to probabilistically guarantee the reach time service requirement for non-reserved rides. Drivers are considered to be associated with a region if they are either serving requests that initiated in the region or they are idle within the region. In this section, we discuss how the targets can be computed at the beginning of the time window. First, we derive a time-dependent upper bound on the blocking probability corresponding to the admission control policy. Then, we determine the target number of drivers that limits the time-averaged blocking probability to be below a certain quality of service threshold. In turn, limiting the time-averaged blocking probability is equivalent to limiting the probability of reach time violation for non-reserved ride requests.

In Equations 1 and 2, representing the admission control policy when the i^{th} non-reserved ride request is received, the values of all the variables are known (for every non-reserved ride request that was previously received, the trip information would have been revealed to the platform). However, at the beginning of the time window, the platform would not know the arrival time, ride duration, and admission decision of a future non-reserved request. Therefore, at the beginning of the time window, $\tau_n, D_n, \gamma_n, f_r^{A(\tau_i),k}(t), \tau_i, D_i, \gamma_i$ are all random variables. To express the probability of admission, we can re-write Equation 1 as shown in Equation 3. Hence, Equation 4 represents the probability that the i^{th} non-reserved ride request would be blocked.

$$P(\gamma_i = 1) = P\left(1 + f_r^{P,k}(t) + f_r^{BA,k}(t) + f_r^{A(\tau_i),k}(t) \le c_r^k, \quad \forall t \in (\tau_i, \min\{\tau_i + D_i, (k+1)w\}]\right) \tag{3}$$

$$\begin{aligned} &P(\gamma_i = 0) = 1 - P(\gamma_i = 1) = \\ &P\left(\exists t \in (\tau_i, \min\{\tau_i + D_i, (k+1)w\}] : 1 + f_r^{P,k}(t) + f_r^{BA,k}(t) + f_r^{A(\tau_i),k}(t) > c_r^k\right) = \\ &P\left(\exists t \in (\tau_i, \min\{\tau_i + D_i, (k+1)w\}] : 1 + f_r^{P,k}(t) + f_r^{BA,k}(t) + \sum_{n=1}^{i-1} \mathbf{1}\{\tau_n + D_n > t\}\gamma_n > c_r^k\right) \end{aligned} \tag{4}$$

Observe that for *predictive* target computations, $f_r^{A(\tau_i),k}(t) = \sum_{n=1}^{i-1} \mathbf{1}\{\tau_n + D_n > t\}\gamma_n$ represents stochastic non-reserved ride requests that will be admitted between $(kw, \tau_i]$ *and* will be active at time $t \in (\tau_i, \min\{\tau_i + D_i, (k+1)w\}]$. Recall that future stochastic non-reserved ride requests appear at a demand rate $\{\lambda_r^k(t) : t \in (kw, (k+1)w]\}$ and the corresponding ride durations are generally distributed according to a distribution $g_r^k(\cdot)$. Previously, we defined the stochastic process $\{N_r^k(t) : t \in (kw, (k+1)w]\}$ that represents the number of future active drivers associated with *admitted* non-reserved rides. Notice that $N_r^k(\tau_i) = f_r^{A(\tau_i),k}(\tau_i)$ is the number of admitted non-reserved ride requests that will be active at time τ_i. However, for $t \in (\tau_i, \min\{\tau_i + D_i, (k+1)w\}]$,

$N_r^k(t) \neq f_r^{A(\tau_i),k}(t)$ since $N_r^k(t)$ includes non-reserved ride requests that will be admitted between $(kw, t]$ while $f_r^{A(\tau_i),k}(t)$ is restricted to non-reserved ride requests admitted between $(kw, \tau_i]$.

To determine the target supply c_r^k, we need to evaluate the blocking probability expression in Equation 4 for different values of c_r^k. However, this probability expression is difficult to analyze due to the dependence of γ_i (admission of i^{th} non-reserved request) on the random variables τ_n, D_n (arrival time, ride duration) and γ_n (admission) associated with previously arriving non-reserved ride requests $n \in \{1, ..., i-1\}$. In addition, the arrival time τ_i of the i^{th} non-reserved ride request also depends on the arrival time τ_n of all previous requests. Moreover, the correlations between the random variables have to be considered over the entire time interval $(\tau_i, \min\{\tau_i + D_i, (k+1)w\}]$ and this interval also has time-varying functions $f_r^{P,k}(t)$ and $f_r^{BA,k}(t)$ that impact the admission probability.

Thus, instead of attempting to evaluate Equation 4, we provide an upper bound on the blocking probability. In particular, let $\{N_r^{k,\infty}(t) : t \in (kw, (k+1)w]\}$ be the number of busy servers in a *transient* $\mathrm{M}_t/\mathrm{GI}/\infty$ queue that starts empty at the beginning of the window $t = kw$, where the arrivals to the $\mathrm{M}_t/\mathrm{GI}/\infty$ queue appear according to a Poisson process with rate $\{\lambda_r^k(t) : t \in (kw, (k+1)w]\}$ and the service distribution is $g_r^k(\cdot)$.

Theorem 1. *The blocking probability, $P(\gamma_i = 0)$, for the i^{th} stochastic non-reserved ride request that appears at time τ_i is bounded above by $P\left(N_r^{k,\infty}(\tau_i) \geq c_r^k - \max\limits_{t \in (\tau_i, (k+1)w]} \left[f_r^{P,k}(t) + f_r^{BA,k}(t)\right]\right)$*

Proof. See Appendix A. □

Given this upper bound in Theorem 1, we can limit the blocking probability at time τ_i to be below a certain quality of service threshold δ by ensuring that the upper bound is below δ (as shown in Inequality 5). Importantly, while $P(\gamma_i = 0)$ is difficult to evaluate as mentioned earlier, the upper bound can be evaluated for any value c_r^k and at any time τ_i using transient analysis of $\mathrm{M}_t/\mathrm{GI}/\infty$ queues (Section 5.1). Subsequently, after illustrating how the upper bound can be evaluated at any time for a specific value of c_r^k, we discuss (Section 5.2) how to use this upper bound to determine the target supply, where the target supply is the minimal c_r^k that limits the time-averaged blocking probability to be below the threshold δ.

$$P(\gamma_i = 0) \leq P\left(N_r^{k,\infty}(\tau_i) \geq c_r^k - \max_{t \in (\tau_i, (k+1)w]} \left[f_r^{P,k}(t) + f_r^{BA,k}(t)\right]\right) \leq \delta \tag{5}$$

5.1. Time-Dependent Distribution of the Number of Busy Servers in an $M_t/GI/\infty$ Queue

To evaluate the upper bound $P\left(N_r^{k,\infty}(\tau_i) \geq c_r^k - \max\limits_{t \in (\tau_i, (k+1)w]} \left[f_r^{P,k}(t) + f_r^{BA,k}(t)\right]\right)$ at time τ_i and for a specific c_r^k, we use a graphical approach that was first recognized by Prékopa (1958) and was subsequently further discussed in articles that analyze $\mathrm{M}_t/\mathrm{GI}/\infty$ queues (Eick et al., 1993; Foley, 1982). We show that the number of busy servers in an $\mathrm{M}_t/\mathrm{GI}/\infty$ queue that starts empty,

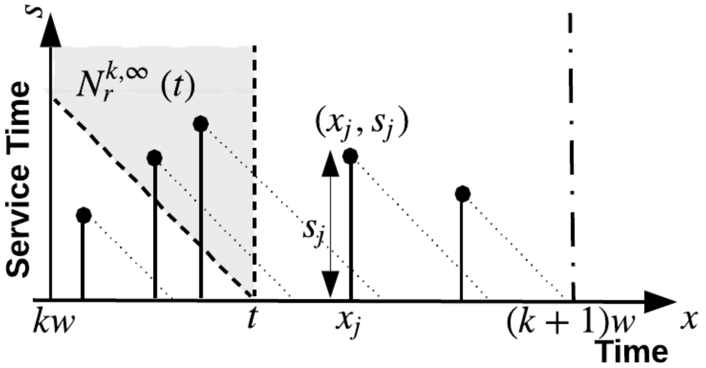


Figure 4: Service time vs. arrival time associated with a transient M$_t$/GI/∞ queue that starts empty at time kw. Since there are an infinite number of servers, all arrivals start being serviced immediately. The dotted diagonal lines represent the decrease in remaining service time as the user is being served. For any time t, the number of users still being served is equal to the number of diagonal lines that intersect a vertical line from t; equivalently, the number of users still being served at t is the number of points in the shaded area.

$N_r^{k,\infty}(\tau_i)$, has a *time-dependent* Poisson distribution, and we derive the time-dependent mean associated with this distribution. Thus, since $\max\limits_{t\in(\tau_i,(k+1)w]}\left[f_r^{P,k}(t)+f_r^{BA,k}(t)\right]$ and c_r^k are known values at time τ_i, evaluating the upper bound is equivalent to computing the probability that a Poisson random variable is greater than or equal to a constant.

Referring to Figure 4, consider stochastic arrivals to an M$_t$/GI/∞ queue such that x_j denotes the j^{th} arrival time according to the Poisson process and s_j denotes the corresponding generally distributed service time. In time window $(kw,(k+1)w]$, the M$_t$/GI/∞ queue is *initially empty* at time kw.

We can think of (x_j, s_j) as a random point in the two-dimensional plane $(kw,(k+1)w] \times [0,\infty)$ that represents the arrival time and service duration. For any two-dimensional set S in $(kw,(k+1)w] \times [0,\infty)$, the number of points in the set represents random sampling of the arrivals Poisson process; thus, the number of points in the set S is *Poisson distributed*. We also know that disjoint two-dimensional sets correspond to independent sampling of a Poisson process; this implies that the number of points in each set is independent of other disjoint sets.

Furthermore, considering an infinitesimal two-dimensional square set with an area $ds(dx)$, we can see that the mean number of points in that set is $\lambda_r^k(x)dx\,\left(g_r^k(s)(ds)\right)$; this implies that the intensity of the two-dimensional Poisson distribution is $\lambda_r^k(x)g_r^k(s)$. Thus, the distribution of points defined as (arrival time, service duration) is Poisson over the two-dimensional space, and the *mean* number of points for any set S is given by $\int_S \lambda_r^k(x)g_r^k(s)dsdx$.

To determine the *mean* number of busy servers $\rho_r^k(t)$, we evaluate the integral $\int_S \lambda_r^k(x)g_r^k(s)dsdx$ over the shaded area illustrated in Figure 4. This shaded area represents arrivals to the M$_t$/GI/∞ queue since time kw that have not yet completed at time t. The resulting expression for $\rho_r^k(t)$ is given in Equation 6. If we further consider that the arrival rate $\lambda_r^k(x)$ is constant over the time window such that $\lambda_r^k(x) = \lambda_r^k$, the expression for $\rho_r^k(t)$ simplifies as shown in Equation 7.

Thus, within each window, $N_r^{k,\infty}(\tau_i)$ is Poisson distributed with a time-dependent mean $\rho_r^k(\tau_i)$.

Given a specific value c_r^k, we can use this characterization of $N_r^{k,\infty}(\tau_i)$ to evaluate the upper bound at any time τ_i.

$$\rho_r^k(t) = \int_{kw}^{t} \int_{t-x}^{\infty} \lambda_r^k(x) g_r^k(s) ds dx \tag{6}$$

$$\begin{aligned} \rho_r^k(t) &= \int_{kw}^{t} \int_{t-x}^{\infty} \lambda_r^k g_r^k(s) ds dx \\ &= \lambda_r^k \left[t - kw - \int_0^{t-kw} G_r^k(x) dx \right] \end{aligned} \tag{7}$$

5.2. Target Predictions for Bounding the Time-Averaged Blocking Probability

Knowing that we can evaluate the upper bound on the blocking probability at any time and for any c_r^k, we now investigate the minimal value of c_r^k that limits the *time-averaged* blocking probability to be below a threshold δ. This minimal c_r^k will be referred to as the *target*, and it represents the number of drivers that the platform seeks to supply during the upcoming time window to limit reach time service violations (i.e., to limit the fraction of non-reserved requests whose reach time will exceed the reach time service requirement).

Precisely, the time-averaged blocking probability in region $r \in R$ during window $(kw, (k+1)w]$ is given in Equation 8, where γ_t is an indicator random variable that takes the value one if a passenger that arrives at time t would be admitted. Since Poisson arrivals see time averages (PASTA property), the time-averaged blocking probability is equivalent to the blocking probability of a typical non-reserved ride request that appears between $(kw, (k+1)w]$. Then, the target c_r^k is the desired number of drivers that restricts this time-averaged blocking probability. In other words, the target c_r^k is the desired number of drivers that limits the blocking probability of a typical non-reserved ride request that will appear during the upcoming window. As previously mentioned, evaluating the blocking probability in Equation 8 is challenging. Thus, to compute the target, we use the time-averaged value of the upper bound in Theorem 1. As shown in Inequality 9, if we find the value of c_r^k that limits the time-averaged upper bound to be less than the threshold δ, then this c_r^k will also limit the time-averaged blocking probability to be less than δ.

$$B_r^k = \frac{1}{w} \int_{kw}^{(k+1)w} P(\gamma_t = 0) dt \tag{8}$$

$$B_r^k \leq \frac{1}{w} \int_{kw}^{(k+1)w} P\left(N_r^{k,\infty}(t) \geq c_r^k - \max_{\hat{t} \in (t,(k+1)w]} \left[f_r^{P,k}(\hat{t}) + f_r^{BA,k}(\hat{t}) \right] \right) dt \leq \delta \tag{9}$$

Therefore, as shown in Equation 10, we seek the minimal value c_r^k that restricts B_r^k to be less than or equal to the threshold δ. In Equation 10, observe that the time-averaged upper bound on the blocking probability decreases monotonically with increasing values of c; consequently, since c must be a non-negative integer, we can iterate through increasing integer values of c until we find the minimal target c_r^k that ensures that the time-averaged blocking probability is less than δ (alternatively, we may use faster line search techniques). Note that just as we can evaluate the upper bound in Theorem 1 for a specific value of c and at a specific time (Section 5.1), we can evaluate the time-averaged upper bound for a specific value of c using numerical integration.

$$\begin{aligned} c_r^k = \min_{c\geq 0,\, c\in\mathbb{Z}} \Bigg[c : & \\ & \frac{1}{w}\int_{kw}^{(k+1)w} P\left(N_r^{k,\infty}(t) \geq c - \max_{\hat{t}\in(t,(k+1)w]} \left[f_r^{P,k}(\hat{t}) + f_r^{BA,k}(\hat{t})\right]\right) dt \leq \delta \Bigg] \end{aligned} \tag{10}$$

The targets c_r^k are computed for every region $r \in R$ at the beginning of window k (i.e., at time $t = kw$). If the number of drivers supplied by the platform in each region (either idling in the region or serving requests that initiate in the region) is equal to the corresponding target, then the blocking probability for future non-reserved requests would be less than the threshold δ. Thus, if the targets are provided in each region, the reach time service requirement is probabilistically guaranteed for stochastic non-reserved rides (for book-ahead rides, the reach time service requirement is guaranteed based on the admission control policy in Section 4). Apart from target computations, the upper bound on the blocking probability can be used as a performance measure for the admission control policy, where performance of the policy refers to the probability of reach time service violation (for a given level of driver supply).

6. Driver Dispatching & Rebalancing Mechanism

In this section, we develop a driver dispatching and rebalancing mechanism that aims to maintain the targets across multiple regions. The targets computed in Section 5 represent a desired level of driver supply such that providing the targets in a region probabilistically guarantees the reach time service requirement for non-reserved ride requests. In practice, within the time window $(kw, (k+1)w]$, drivers serving requests that initiated in a region $r \in R$ may finish their trips in other regions. Similarly, drivers serving requests that initiated in an external region $r' \in R\backslash\{r\}$ may finish their trip in region r. Thus, the number of drivers associated with each region may deviate from the corresponding target c_r^k due to observed origin-destination trip patterns. This section presents a dispatching/rebalancing mechanism that computes the minimum number of driver transitions that achieve the targets, where only idle drivers are allowed to transition between adjacent regions. We show that the proposed optimization formulation reduces to a *minimum cost flow* formulation on a transformed network of regions.

In more detail, consider that at some time t the platform aims to determine the necessary driver transitions that maintain the targets. In this section, all the defined variables represent the network conditions at time t; this time t could be either at the beginning of time window $(kw, (k+1)w]$ or within the window. For every region i, let a_i be the number of active drivers serving requests initiated in the region, and let e_i be the number of idle drivers in the region. In addition, for every region, define a virtual supply s_i^v as shown in Equation 11, where the virtual supply represents the number of excess drivers (beyond the target) that can transition to adjacent regions. The virtual supply s_i^v is limited by the number of idle drivers in the region; thus, it is the minimum of the idle drivers e_i and the number of drivers in excess of the target $(a_i + e_i) - c_i^k$. Similarly, define a virtual demand d_i^v as shown in Equation 12, where the virtual demand represents the number of additional drivers needed in region i to meet the target c_i^k at time t. Furthermore, for every region i, define Δ_i as shown in Equation 13, where Δ_i represents either the demand (expressed as a negative value) or the supply.

$$s_i^v = \begin{cases} \min\left\{e_i, (a_i + e_i) - c_i^k\right\} & \text{if} \quad c_i^k - (a_i + e_i) \leq 0 \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

$$d_i^v = \begin{cases} c_i^k - (a_i + e_i) & \text{if} \quad c_i^k - (a_i + e_i) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

$$\Delta_i = \begin{cases} -\left[c_i^k - (a_i + e_i)\right] & \text{if} \quad c_i^k - (a_i + e_i) > 0 \\ \min\left\{e_i, (a_i + e_i) - c_i^k\right\} & \text{otherwise} \end{cases} \tag{13}$$

For the regions defined in Section 3, we construct a directed network $G = (R, E)$. The set of regions R corresponds to the nodes of the network. The set of edges E includes links (i, j) and (j, i) for every pair of *adjacent* regions i and j (see original network in Figure 5). Define h_{ij} as the number of drivers that need to transition from region i to the adjacent region j on link (i, j). The platform rebalancing optimization formulation is shown in Equations 14–18. In this formulation, the platform seeks to minimize the number of driver transitions (objective 14) while ensuring that the targets are maintained (constraint 15). In particular, constraint 15 specifies that the difference between drivers leaving a region and drivers arriving to a region should match the supply/demand in the region. Constraint 16 restricts the number of drivers leaving a region to the number of idle drivers in the region; in other words, this constraint ensures that the optimal solution to formulation 14–18 (if it exists) describes the number of *idle* drivers transitions to *adjacent* regions (i.e., idle drivers do not transition across multiple regions). The remaining constraints 17 and 18 ensure that the decision variables h_{ij} are non-negative integers.

$$\min_{h_{ij}:(i,j)\in E} \quad \sum_{(i,j)\in E} h_{ij} \tag{14}$$

$$\text{s.t.} \quad \sum_{j:(i,j)\in E} h_{ij} - \sum_{j:(j,i)\in E} h_{ji} = \Delta_i \qquad \forall i \in R \tag{15}$$

$$\sum_{j:(i,j)\in E} h_{ij} \leq e_i \qquad \forall i \in R \tag{16}$$

$$h_{ij} \geq 0 \qquad \forall (i,j) \in E \tag{17}$$

$$h_{ij} \in \mathbb{Z} \qquad \forall (i,j) \in E \tag{18}$$

In formulation 14–18, unless the total supply matches the total demand ($\sum_{i\in R} s_i^v = \sum_{i\in R} d_i^v$)

and the network is strongly connected, the optimization problem may not have a feasible solution. Thus, we consider instead the revised formulation 19–24, where h_i corresponds to drivers added/removed from region i by adjusting the total number of drivers in the network. Since adding or removing drivers would be costly to the platform (e.g., requires incentivizing new drivers or taking drivers offline), we associate a high cost M with such transitions. As a result, in the optimal solution to formulation 19–24, the total number of drivers is adjusted only if the targets could not be maintained internally via transitions of idle drivers across adjacent regions.

$$\begin{aligned}
&\min_{h_{ij}:(i,j)\in E,\, h_i:i\in R} \quad \sum_{(i,j)\in E} h_{ij} + M \sum_{i\in R} |h_i| && (19)\\
&\text{s.t.} \quad \sum_{j:(i,j)\in E} h_{ij} - \sum_{j:(j,i)\in E} h_{ji} + h_i = \Delta_i \qquad \forall i \in R && (20)\\
&\quad \sum_{j:(i,j)\in E} h_{ij} \le e_i \qquad \forall i \in R && (21)\\
&\quad h_{ij} \ge 0 \qquad \forall (i,j) \in E && (22)\\
&\quad h_{ij} \in \mathbb{Z} \qquad \forall (i,j) \in E && (23)\\
&\quad h_i \in \mathbb{Z} \qquad \forall i \in R && (24)
\end{aligned}$$

Let $h_{i\bullet}$ and $h_{\bullet i}$ be defined as in Equations 25 and 26. In this case, $h_{\bullet i}$ corresponds to drivers added to region $i \in R$ by adjusting the total number of drivers, and $h_{i\bullet}$ corresponds to drivers removed from region $i \in R$ by adjusting the total number of drivers (i.e., $h_{i\bullet}$ represents drivers that can be removed from the system to avoid having excess idle drivers).

$$h_{i\bullet} = \begin{cases} h_i & \text{if} \quad h_i > 0 \\ 0 & \text{otherwise} \end{cases} \tag{25}$$

$$h_{\bullet i} = \begin{cases} |h_i| & \text{if} \quad h_i < 0 \\ 0 & \text{otherwise} \end{cases} \tag{26}$$

Moreover, for notational convenience in mapping the problem to a min-cost flow reformulation, define for each region $i \in R$ variables $h_{ii^\star}$ that represent the total number of drivers leaving region i to adjacent regions (Equation 27). In addition, for each link $(i,j) \in E$, define variables $h_{i^\star j} = h_{ij}$. Thus, we can define $h_{ii^\star}$ in terms of $h_{i^\star j}$ as in Equation 28. Since h_{ij} is a non-negative integer for all $(i,j) \in E$, we have that $h_{ii^\star}$ and $h_{i^\star j}$ are non-negative integers as well.

$$\begin{aligned}
h_{ii^\star} &= \sum_{j:(i,j)\in E} h_{ij} \qquad \forall i \in R && (27)\\
&= \sum_{j:(i,j)\in E} h_{i^\star j} \qquad \forall i \in R && (28)
\end{aligned}$$

In Appendix B, through a sequence of reformulations, we show that optimization problem 19–24 reduces to the formulation 29–41.

$$\min_{h_{i^\star j}:(i,j)\in E,\; h_{i\bullet},h_{\bullet i},h_{ii^\star}:i\in R,\; \bar{h}} \quad \sum_{i\in R} h_{ii^\star} + M\sum_{i\in R}\left[h_{i\bullet} + h_{\bullet i}\right] \tag{29}$$

$$\text{s.t.} \quad h_{ii^\star} - \sum_{j:(j,i)\in E} h_{j^\star i} + h_{i\bullet} - h_{\bullet i} = \Delta_i \qquad \forall i \in R \tag{30}$$

$$\sum_{i\in R} h_{\bullet i} + \bar{h} = \sum_{i\in R} d_i^v \tag{31}$$

$$-\left[\sum_{i\in R} h_{i\bullet} + \bar{h}\right] = -\sum_{i\in R} s_i^v \tag{32}$$

$$\sum_{j:(i,j)\in E} h_{i^\star j} - h_{ii^\star} = 0 \qquad \forall i \in R \tag{33}$$

$$0 \le h_{ii^\star} \le e_i \qquad \forall i \in R \tag{34}$$

$$h_{i^\star j} \ge 0 \qquad \forall (i,j) \in E \tag{35}$$

$$h_{i\bullet}, h_{\bullet i} \ge 0 \qquad \forall i \in R \tag{36}$$

$$\bar{h} \ge 0 \tag{37}$$

$$h_{ii^\star} \in \mathbb{Z} \qquad \forall i \in R \tag{38}$$

$$h_{i^\star j} \in \mathbb{Z} \qquad \forall (i,j) \in E \tag{39}$$

$$h_{i\bullet}, h_{\bullet i} \in \mathbb{Z} \qquad \forall i \in R \tag{40}$$

$$\bar{h} \in \mathbb{Z} \tag{41}$$

Consider the standard minimum cost flow problem given in formulation 42–44 for a network $G' = (V, A)$ (Ahuja et al., 1993; Wolsey, 1998), where c_{pq} is the cost of a unit flow on link $(p,q) \in A$, x_{pq} are decision variables corresponding to flows on each link $(p,q) \in A$, b_p is the equivalent of supply/demand at node p, and u_{pq} is an upper bound on the flows x_{pq} (i.e., capacity of link $(p,q) \in A$). A necessary condition for feasibility of the optimization problem is $\sum_{p\in V} b_p = 0$.

$$\min_{x_{pq}:(p,q)\in A} \quad \sum_{(p,q)\in A} c_{pq} x_{pq} \tag{42}$$

$$\text{s.t.} \quad \sum_{\{q:(p,q)\in A\}} x_{pq} - \sum_{\{q:(q,p)\in A\}} x_{qp} = b_p \qquad \forall p \in V \tag{43}$$

$$0 \le x_{pq} \le u_{pq} \qquad \forall (p,q) \in A \tag{44}$$

Apart from the integrality constraints, the formulation 29–41 has the same structure as the minimum cost flow optimization problem 42–44; this implies that the constraint matrix associated with formulation 29–41 is totally unimodular. Thus, since Δ_i, d_i^v, s_i^v, and e_i are all integer values, each extreme point in the constraint set will be integral. Then, solving the linear programming relaxation in 45–53 will give us the *integer optimal solution* of optimization problem 29–41.

$$\min_{h_{i^\star j}:(i,j)\in E,\ h_{i\bullet},h_{\bullet i},h_{ii^\star}:i\in R,\ \bar{h}} \quad \sum_{i\in R} h_{ii^\star} + M\sum_{i\in R}\left[h_{i\bullet}+h_{\bullet i}\right] \tag{45}$$

$$\text{s.t.}\quad h_{ii^\star} - \sum_{j:(j,i)\in E} h_{j^\star i} + h_{i\bullet} - h_{\bullet i} = \Delta_i \qquad \forall i \in R \tag{46}$$

$$\sum_{i\in R} h_{\bullet i} + \bar{h} = \sum_{i\in R} d_i^v \tag{47}$$

$$-\left[\sum_{i\in R} h_{i\bullet} + \bar{h}\right] = -\sum_{i\in R} s_i^v \tag{48}$$

$$\sum_{j:(i,j)\in E} h_{i^\star j} - h_{ii^\star} = 0 \qquad \forall i \in R \tag{49}$$

$$0 \le h_{ii^\star} \le e_i \qquad \forall i \in R \tag{50}$$

$$h_{i^\star j} \ge 0 \qquad \forall (i,j) \in E \tag{51}$$

$$h_{i\bullet}, h_{\bullet i} \ge 0 \qquad \forall i \in R \tag{52}$$

$$\bar{h} \ge 0 \tag{53}$$

The linear program 45–53 can be mapped to a minimum cost flow program 42–44 applied on a transformed network illustrated in Figure 5. In particular, consider a source node SO where links (SO, i) that connect SO to region $i \in R$ dispatch flows $h_{\bullet i}$. In addition, consider a sink node SI where links (i, SI) that connect region $i \in R$ to SI dispatch flows $h_{i\bullet}$. Let $\bar{h}$ represent the flow between SO and SI. Then, observe that constraint 46 is equivalent to constraint 43 at all un-starred nodes in the network transformation of Figure 5. Similarly, constraint 49 is equivalent to constraint 43 at all starred nodes. Constraint 47 corresponds to constraint 43 applied at the source node SO, and constraint 48 corresponds to constraint 43 applied at the sink node SI. In the network transformation, each link is associated with a $(\text{cost}, \text{capacity})$ label. Observe that the objective function 45 can be obtained by plugging the link costs and flow variables in the minimum cost flow objective function 42. Also, observe that constraints 50–53 are the link capacity constraints 44 in the transformed network. Furthermore, by definition, $\sum_{i\in R}\Delta_i + \sum_{i\in R} d_i^v - \sum_{i\in R} s_i^v = 0$; this implies that the necessary condition for feasibility in the minimum cost flow program ($\sum_{p\in V} b_p = 0$) is satisfied. Thus, solving the linear program 45–53 is equivalent to solving the minimum cost flow program 42–44 using the transformed network.

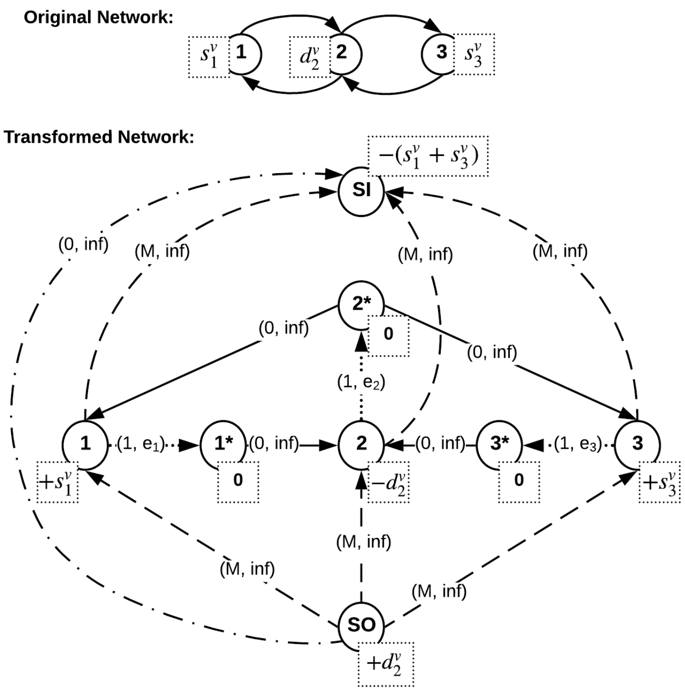


Figure 5: Network transformation corresponding to the minimum cost flow program, where solving the integer program 19–24 using the original network is equivalent to solving the minimum cost flow program 42–44 using the transformed network. Each link in the transformed network is associated with a (cost, capacity) label. Each node in the transformed network is either a supply, demand, or transmission node such that values of b_p in constraint 43 are within the squares.

Consequently, since the integer program 19–24 reduces to formulation 45–53, then solving the integer program 19–24 on the original network (Figure 5) is equivalent to solving the minimum cost flow program 42–44 on the illustrated transformed network. As a minimum cost flow program, the driver dispatching and rebalancing optimization problem can be solved in polynomial time. The optimal solution of the optimization program represents recommended idle driver transitions that are needed to maintain the targets across regions. Specifically, the optimal solution includes idle drivers that should transition to adjacent regions *and* idle drivers that should be added to the network by adjusting the total number of drivers in the system. In addition, the optimal solution also includes excess idle drivers that can be removed from the system.

7. Simulation Results

In this section, we present experimental results using data from Lyft operations in Manhattan, NYC on Friday December 14th, 2018 (NYCTLC, 2019). We consider trips that started between 16:00–19:00 (local time) in four regions. The regions chosen roughly correspond to four sections of the city as illustrated in Figure 6 (1-lower Manhattan, 2-midtown Manhattan, 3-upper west side, and 4-upper east side). For time windows of duration $w = 20$ minutes, we use trip initiation and completion time data available on the New York City Taxi and Limousine Commission website to characterize the processes $\{f_r^{P,k}(t), f_r^{BA,k}(t), N_r^k(t) : t \in (kw, (k+1)w]\}$. Our primary findings

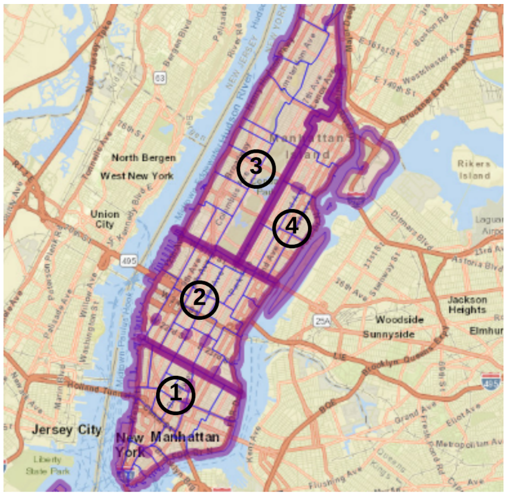


Figure 6: Manhattan divided into four regions

suggest that an increase in the fraction of book-ahead rides leads to a reduction in the total number of drivers that are needed to probabilistically guarantee the reach time service requirement. This reduction in the total number of drivers is also associated with a lower number of idling drivers (i.e., an increase in the driver utilization rate).

7.1. System model specification and comparison to observed data

The process $\{f_r^{P,k}(t) : t \in (kw, (k+1)w]\}$ is generated at the beginning of every window k. Specifically, using the available data, $f_r^{P,k}(t)$ represents *previously observed* rides that initiated in region r prior to $t = kw$ and will be active at time $t \in (kw, (k+1)w]$.

To generate the process $\{f_r^{BA,k}(t) : t \in (kw, (k+1)w]\}$ from the New York City data, we randomly sample a fraction p_{BA} of the trips that start during window k in region r. We choose to generate $f_r^{BA,k}(t)$ as the fraction of anticipated rides since we are interested in analyzing the change in the target number of drivers as the fraction of book-ahead rides increases.

As for the stochastic process $\{N_r^k(t) : t \in (kw, (k+1)w]\}$, at the beginning of each window k, we calibrate the demand rate λ_r^k corresponding to ride requests that will appear during the upcoming window in region r. In the following simulation, for simplicity, the demand rate varies across time-windows but is assumed constant within each time window; however, the proposed framework can be implemented using time-dependent demand rate functions by evaluating Equation 6. Moreover, even with window-constant demand rates, the Poisson distribution describing active drivers is time-varying within each window such that the mean is given by Equation 7. We emphasize that this transient analysis does not assume an equilibrium or steady-state conditions in any time window. The arrival rate for region 2 is shown in Figure 7; as observed, the demand rate increases rapidly showing the need for non-equilibrium methods. For the distribution $g_r^k(\cdot)$ representing ride duration, we use the empirical distribution that is derived from the observed rides in each region. Note that to analyze the change in the target number of drivers with increasing book-ahead rides, we effectively assume that the arrival rate of non-reserved ride requests is

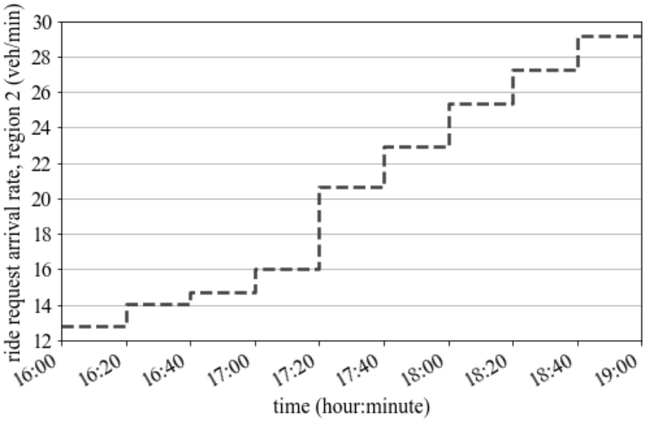


Figure 7: Arrival rate for ride requests that initiate in region 2.

$(1 - p_{BA})\lambda_r^k$ (where a fraction p_{BA} of the anticipated trips that will initiate during window k are book-ahead rides).

As illustrated in Figure 8, the proposed model for predicting the number of active rides (Section 3) accurately represents the observed data. In this Figure, for comparison with observed trip data, we consider that all rides are admitted and that there are no book-ahead rides (effectively assuming $N_r^k(t) = N_r^{k,\infty}(t)$). Recall that $N_r^k(t)$ represents the *predicted* non-reserved ride requests that will appear during window k; in contrast, during window $k+1$, the process $f_r^{P,k+1}(t)$ consists of observed trips (as given in the data) that differ from the previously predicted trips.

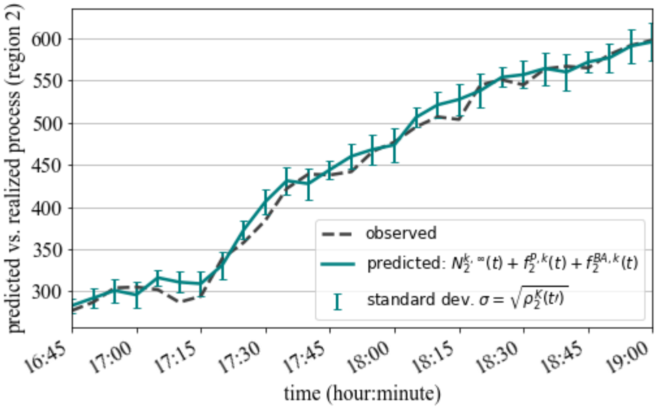


Figure 8: Predicted total number of active rides vs. observed number of active rides, where predictions were made over time windows with a duration of 20 minutes. The error bars correspond to one standard deviation of the time-dependent Poisson distribution characterizing $N_r^{k,\infty}$. In this figure, to compare with the observed trip data, we assume that all rides are admitted (i.e., we consider that $N_r^k(t) = N_r^{k,\infty}(t)$).

7.2. Upper bound on the blocking probability

To evaluate how tight is the upper bound in Inequality 22, we implement the admission control policy in region 2 and average the observed proportion of blocked rides B_r^k across time windows.

For this upper bound numerical analysis, the assumptions involved in target evaluation and admission control apply; specifically, the total supply (active and idle) is maintained at the target level, drivers switch between active and idle within the region, and non-reserved rides are blocked if upon admission the total number of active rides would exceed the target at some point in time throughout the ride duration. Figure 9 shows the variation in the blocking proportion B_r^k relative to the upper bound δ. As observed, the blocking proportion B_r^k increases with larger tolerance values. We also observe that the blocking proportion increases with the fraction of book-ahead rides p_{BA} as a result of fewer idle drivers being available for non-reserved rides.

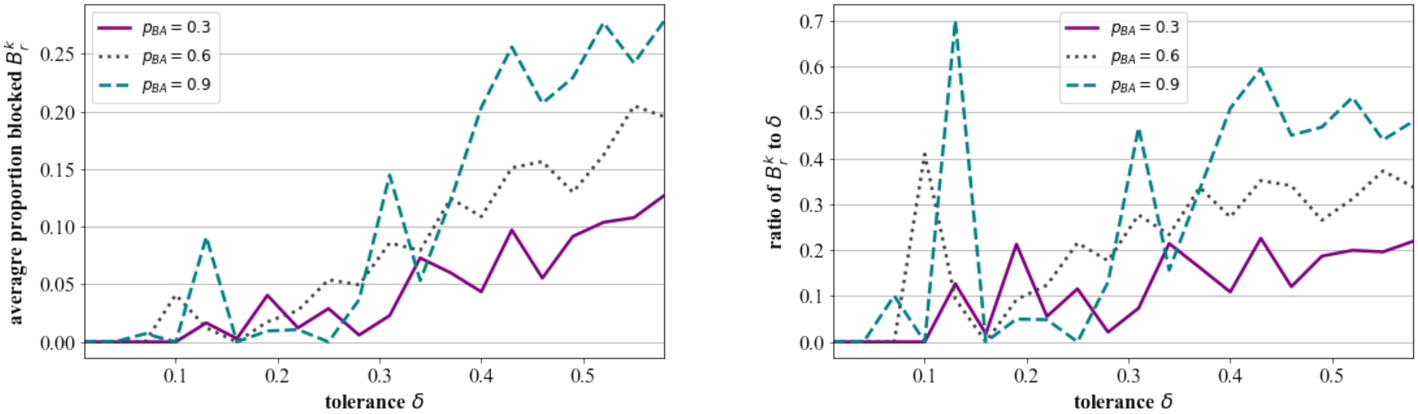


Figure 9: The change in observed blocking proportion B_r^k and the ratio B_r^k/δ relative to the upper bound δ.

7.3. *Target computations, admission control, and minimum cost flow dispatching/rebalancing*

Then, to account for the spatial distribution of demand and the variation in supply across regions, we implement the proposed framework in Sections 3–6 (see Figure 3). In particular, we demand moves between regions and that the supply deviates from the target, and we implement the min. cost flow to maintain the target.

First, as mentioned in Section 7.1, we characterize the processes $\{f_r^{P,k}(t), f_r^{BA,k}(t), N_r^k(t) : t \in (kw, (k+1)w]\}$ representing the predicted number of active rides in each region r. Then, using the upper bound on the time-dependent blocking probability of the admission control policy, we determine the target number of drivers associated with every region r during the upcoming window. After that, at the beginning of the time window, we apply the driver dispatching/rebalancing mechanism to attain the targets across regions. Then, throughout the time window, for every non-reserved ride request that is received, we implement the admission control policy to determine whether the request should be admitted or blocked; the received non-reserved ride requests are directly retrieved from the New York City data (as opposed to the predictions $N_r^k(t)$). We also implement the driver dispatching/rebalancing mechanism halfway through the time window. However, at the beginning of the time window we allow for total adjustments of the driver supply while halfway through the window we consider that only existing idle drivers can transition across adjacent regions. This process is then repeated for every time window.

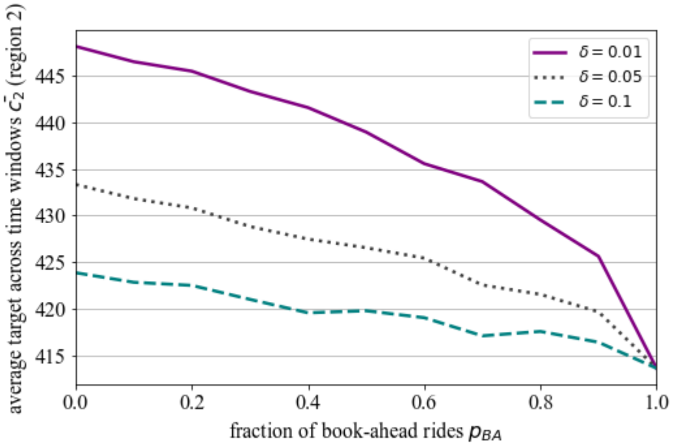


Figure 10: Change in the time-averaged target number of drivers with an increase in the fraction of book-ahead rides (for different quality of service thresholds δ). For each data point (i.e., every (p_{BA}, δ) pair), the plotted time-averaged target is the average of the corresponding value obtained from 30 different iterations of the proposed framework, where this averaging is needed due to the randomness in generation of the book-ahead profile $f_r^{BA,k}(t)$.

For simulation purposes, we disregard the stochasticity of drivers entering and exiting the system across time windows. However, the admission control policy, target computations, and subsequent driver dispatching policy allow for a time-varying and stochastic variation in the supply that is joining or leaving the platform. In fact, target evaluation is based on the demand process and the admission control assumes that the target is maintained throughout the time window. Even if the actual supply deviates from the target, the admission control policy is still implemented by finding if there are any idle drivers and measuring the change in idle drivers relative to the target. On the other hand, the driver dispatching is only concerned with the instantaneous state of the supply relative to the target.

Note that the presented driver rebalancing strategy only uses information from the current time window. In other words, while the proposed state-dependent strategy does not assume steady-state conditions in a time-varying environment, it does not look into future windows to determine the current rebalancing recommendations. Alternative policies that predict future dynamics multiple windows in advance may also be effective since they would have more information on the anticipated variation in driver supply.

We apply the same framework for different fractions of book-ahead rides and record the target c_r^k across windows. In Figure 10, we illustrate the change in targets for different fractions of book-ahead rides. In particular, we measure the time-averaged target $\bar{c}_r$ for increasing values of p_{BA} and different quality of service thresholds δ (as defined in Section 5.2, δ bounds the time-averaged blocking probability such that a lower value of δ indicates a higher quality of service). As expected, we observe that the target number of drivers increases with decreasing δ; this result implies that a larger number of drivers is needed to guarantee the reach time service requirement for a greater fraction of non-reserved ride requests. We also observe that the target number of drivers decreases

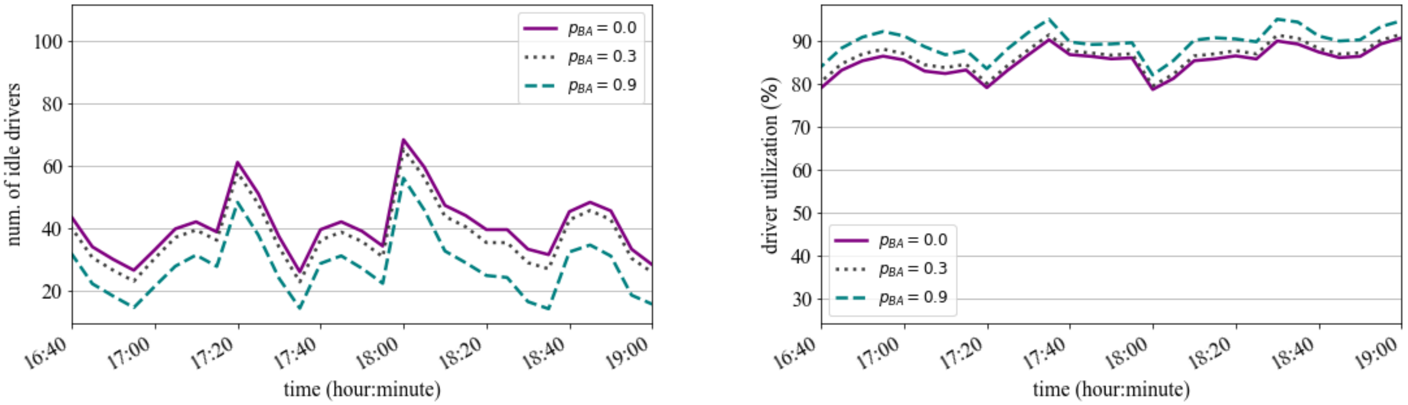


Figure 11: The number of idle drivers and the driver utilization rate 100*(active/(active+idle)) averaged across regions. The quality of service threshold δ is set at 0.01.

as the fraction of book-ahead rides increases. The decrease in targets indicates that the number of drivers needed decreases with more information on anticipated trips.

For the simulation setting, the ratio of internal driver transitions $\sum_{(i,j)\in E} h_{ij}$ to the total flows $(\sum_{(i,j)\in E} h_{ij} + \sum_{i\in R} |h_i|)$ was approximately 0.5 when averaged across min-cost flow evaluations. The recommended external flows reflect the additional drivers needed to satisfy increasing demand (Figure 7). This ratio depends on the demand rates, frequency of driver rebalancing, and the spatial distribution of regions. All these parameters would vary between different areas and time periods.

As the target decreases with increasing fractions of book-ahead rides, the number of idling drivers in the system also decreases. Figure 11 illustrates the average number of idling drivers for different reservation levels. We observe that when $p_{BA} = 0.9$ the average number of idle drivers can be up to 17.3 less than the corresponding value when $p_{BA} = 0.0$. This reduction in the number of idle drivers with increasing p_{BA} translates to a higher driver utilization rate.

Figure 12 illustrates the average number of rides that are blocked by the admission control policy (i.e., the reach time service requirement was not met for these rides). As shown, the average number of blocked rides increases with reservation levels. This increase in blocking results from the reduction in the overall number of drivers in the system. However, the fraction of blocked requests is (mostly) within the specified threshold $\delta = 0.01$. For $p_{BA} = 0.9$, the fraction of blocked requests slightly exceeds the level of service threshold δ; this discrepancy may be attributed to the randomness in the system and the fact that the targets are not perfectly maintained throughout the entire time window.

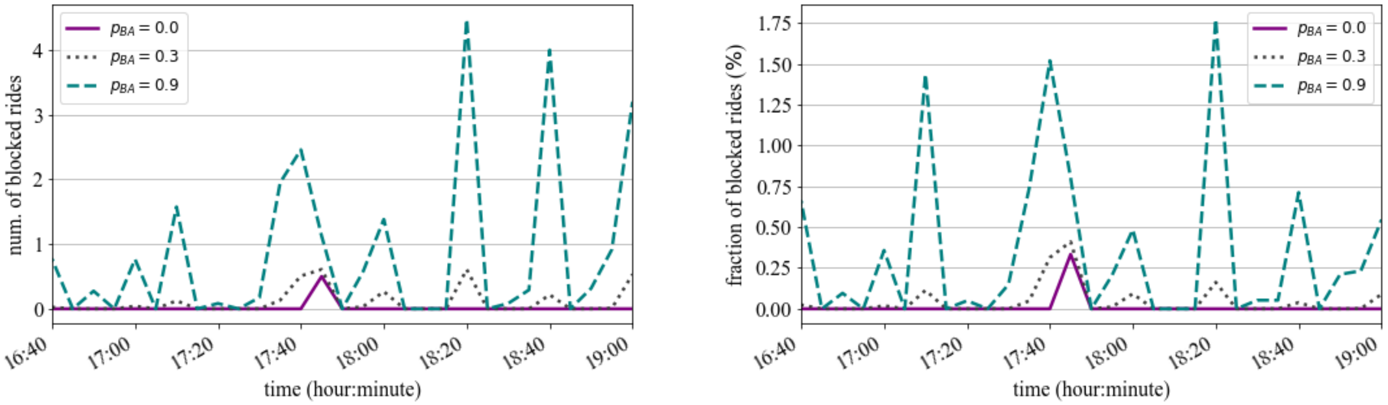


Figure 12: The number of blocked ride requests and the fraction of blocked requests 100*(blocked/(admitted+blocked)) averaged across regions. The quality of service threshold δ is set at 0.01.

The previous analysis assumed perfect compliance with inter-regional driver transitions at the simulation-specific driver rebalancing stages (beginning and mid-window). However, the drivers may not follow platform recommendations and that would result in greater difficulty maintaining the targets. Figure 13 shows the number of blocked rides and fraction of blocked rides in the worst-case scenario where drivers do not follow inter-regional transition recommendations. As observed, the number of blocked rides almost doubles in some cases and the fraction of blocked rides also increases up to 3.5%.

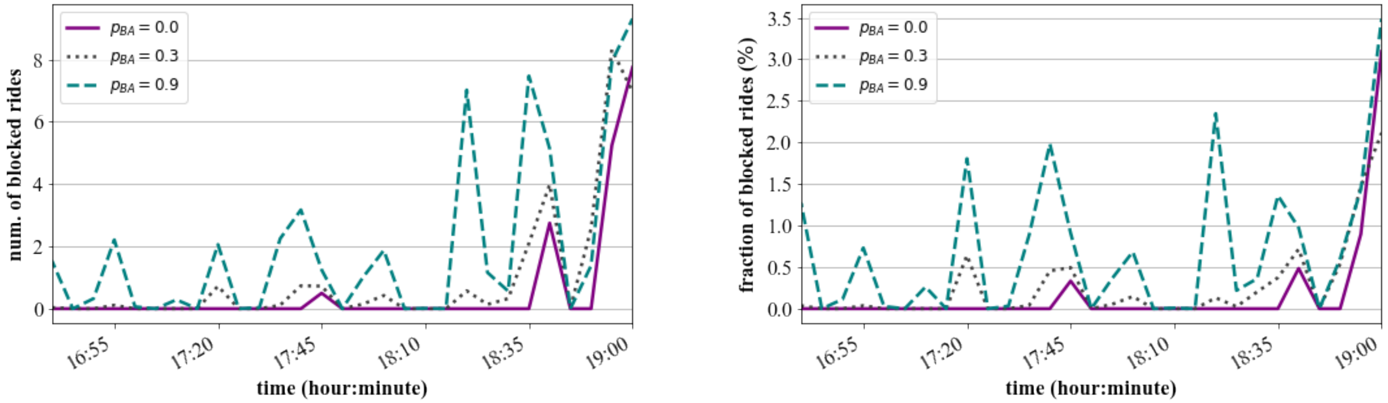


Figure 13: For the case when idle drivers do not follow platform-recommended transitions between regions, we observe an increase in the number blocked rides and the fraction of blocked rides. The quality of service threshold δ is set at 0.01.

8. Conclusion

In this article, we propose a model for transient analysis of stochasticity in ridesourcing systems. As opposed to steady-state equilibrium methods, we characterize the time-dependent state of the system and design control policies for managing driver supply. Furthermore, we incorporate book-ahead rides (reservations) in our framework and analyze the impact of book-ahead rides on driver supply management.

In more detail, we propose a state-dependent control policy that assigns drivers to observed ride requests with the objective of guaranteeing the reach time service requirement for book-ahead rides. Then, we derive a time-dependent upper bound on the performance of the control policy, where the performance of the policy is measured in terms of the probability of reach time service violations for non-reserved rides. Subsequently, this upper bound is used to determine the target number of drivers that probabilistically guarantees the reach time service requirement for non-reserved rides. The targets represent the total number of drivers that are associated with a region such that the drivers are either idling in the region or serving requests that initiate in the region. Then, considering a set of regions with different targets, we propose a driver dispatching/rebalancing optimization program that seeks to maintain the targets across regions. We show that the dispatching/rebalancing problem reduces to a minimum cost flow program that is solved on a transformed network.

The key findings are as follows: (1) For the desired reach time quality of service, an increase in the fraction of book-ahead rides leads to a reduction in the total number of drivers required. (2) This reduction in the total number of drivers is associated with a decrease in the number of idling drivers. (3) Once the driver supply is decreased, there is a greater risk that the reach time service requirement will be violated for anticipated non-reserved rides. However, the fraction of rides that experience increased reach time beyond the reach time service requirement is within a specified threshold, where this threshold dictates the target number of required drivers. (4) For Lyft rides in Manhattan, we observe rapid variations in demand rates that emphasize the need for transient analysis of ridesourcing dynamics.

The proposed model can be used for operation of ridesourcing systems. Specifically, the proposed control policy can be used for ensuring reach time priority for book-ahead rides, the target supply determines the number of drivers that would probabilistically guarantee the reach time service requirement for non-reserved rides, and the minimum cost flow program determines the necessary driver dispatching/rebalancing that is needed to maintain the targets.

More importantly, the proposed model can inform policy decisions that seek to maximize driver welfare and to reduce congestion externalities associated with ridesourcing platforms. In particular, for a given quality of service and reach time service requirement, policy makers can determine if the ridesourcing platform is employing an excessive number of drivers by comparing the total number of drivers in the system to the target supply. In addition, our results suggest that policy makers should advocate for an increased fraction of book-ahead rides and supply management strategies that use this book-ahead information to reduce the number of idling drivers.

Supplementary Material

Data and code used to generate results in this article are available on `https://github.com/spartalab/book-ahead/`

Acknowledgments

This research was supported by the National Science Foundation under Grants No. 1562291 and 1826320. Partial support was provided by the Data-Supported Transportation Operations and Planning center (D-STOP). The authors would like to thank Ahmad AlAmmouri for his insightful comments and suggestions.

Appendix

A. Theorem 1 Proof:

Theorem. *The blocking probability, $P(\gamma_i = 0)$, for the i^{th} stochastic non-reserved ride request that appears at time τ_i is bounded above by $P\left(N_r^{k,\infty}(\tau_i) \geq c_r^k - \max_{t \in (\tau_i,(k+1)w]} \left[f_r^{P,k}(t) + f_r^{BA,k}(t)\right]\right)$*

Proof. We first start by deriving upper bounds on the blocking probability $P(\gamma_i = 0)$ (Inequalities 56–58). Then, through Equations 60–64, we show that the upper bound in Inequality 58 can be expressed in terms $N_r^{k,\infty}(\tau_i)$, where $N_r^{k,\infty}(\tau_i)$ is the number of busy servers at time τ_i in a transient $\mathrm{M}_t/\mathrm{GI}/\infty$ queue that starts empty at the beginning of the time window.

$$P(\gamma_i = 0) \tag{54}$$

$$= P\left(\exists t \in (\tau_i, \min\{\tau_i + D_i, (k+1)w\}] : 1 + f_r^{P,k}(t) + f_r^{BA,k}(t) + \sum_{n=1}^{i-1} \mathbf{1}\{\tau_n + D_n > t\}\gamma_n > c_r^k\right) \tag{55}$$

$$\leq P\left(\exists t \in (\tau_i, \min\{\tau_i + D_i, (k+1)w\}] : 1 + f_r^{P,k}(t) + f_r^{BA,k}(t) + \sum_{n=1}^{i-1} \mathbf{1}\{\tau_n + D_n > t\} > c_r^k\right) \tag{56}$$

$$\leq P\left(\exists t \in (\tau_i, (k+1)w] : 1 + f_r^{P,k}(t) + f_r^{BA,k}(t) + \sum_{n=1}^{i-1} \mathbf{1}\{\tau_n + D_n > t\} > c_r^k\right) \tag{57}$$

$$\leq P\left(\exists t \in (\tau_i, (k+1)w] : 1 + f_r^{P,k}(t) + f_r^{BA,k}(t) + \sum_{n=1}^{i-1} \mathbf{1}\{\tau_n + D_n > \tau_i\} > c_r^k\right) \tag{58}$$

Inequality 56 holds since we are considering that all requests that are received before the i^{th} request are admitted (i.e, $\gamma_n = 1$ for all $n \in \{1, ..., i-1\}$).
Inequality 57 holds since we are expanding the time horizon until the end of the window.
Inequality 58 follows since $\sum_{n=1}^{i-1} \mathbf{1}\{\tau_n + D_n > \tau_i\} \geq \sum_{n=1}^{i-1} \mathbf{1}\{\tau_n + D_n > t\}$. Specifically, the number of non-reserved ride requests that are received between $(kw, \tau_i]$ and are still active (being served)

at time τ_i is *at least as large as* the corresponding number of non-reserved ride requests that are received between $(kw, \tau_i]$ and are still active at time $t \in (\tau_i, (k+1)w]$ (i.e. $t \geq \tau_i$).
Then, we can rearrange the last expression in Inequality 58 as follows:

$$P\left(\exists t \in (\tau_i, (k+1)w] : 1 + f_r^{P,k}(t) + f_r^{BA,k}(t) + \sum_{n=1}^{i-1} \mathbf{1}\{\tau_n + D_n > \tau_i\} > c_r^k\right) \tag{59}$$

$$= 1 - P\left(1 + f_r^{P,k}(t) + f_r^{BA,k}(t) + \sum_{n=1}^{i-1} \mathbf{1}\{\tau_n + D_n > \tau_i\} \leq c_r^k, \quad \forall t \in (\tau_i, (k+1)w]\right) \tag{60}$$

$$= 1 - P\left(1 + \max_{t \in (\tau_i, (k+1)w]} \left[f_r^{P,k}(t) + f_r^{BA,k}(t)\right] + \sum_{n=1}^{i-1} \mathbf{1}\{\tau_n + D_n > \tau_i\} \leq c_r^k\right) \tag{61}$$

$$= P\left(1 + \max_{t \in (\tau_i, (k+1)w]} \left[f_r^{P,k}(t) + f_r^{BA,k}(t)\right] + \sum_{n=1}^{i-1} \mathbf{1}\{\tau_n + D_n > \tau_i\} > c_r^k\right) \tag{62}$$

$$= P\left(\sum_{n=1}^{i-1} \mathbf{1}\{\tau_n + D_n > \tau_i\} > c_r^k - \max_{t \in (\tau_i, (k+1)w]} \left[f_r^{P,k}(t) + f_r^{BA,k}(t)\right] - 1\right) \tag{63}$$

$$= P\left(\sum_{n=1}^{i-1} \mathbf{1}\{\tau_n + D_n > \tau_i\} \geq c_r^k - \max_{t \in (\tau_i, (k+1)w]} \left[f_r^{P,k}(t) + f_r^{BA,k}(t)\right]\right) \tag{64}$$

Equality 61 follows since $f_r^{P,k}(t) + f_r^{BA,k}(t)$ are the only components that depend on t in expression 60, and if the sum $1 + f_r^{P,k}(t) + f_r^{BA,k}(t) + \sum_{n=1}^{i-1} \mathbf{1}\{\tau_n + D_n > \tau_i\}$ is less than or equal to c_r^k at $\tilde{t} = \underset{t \in (\tau_i, (k+1)w]}{\arg\max} \left[f_r^{P,k}(t) + f_r^{BA,k}(t)\right]$, then the aforementioned sum is less than or equal to c_r^k for all $t \in (\tau_i, (k+1)w]$.
Equality 64 follows since $\sum_{n=1}^{i-1} \mathbf{1}\{\tau_n + D_n > \tau_i\}$, $\max_{t \in (\tau_i, (k+1)w]} \left[f_r^{P,k}(t) + f_r^{BA,k}(t)\right]$, and c_r^k are all integer values representing the number of active drivers or driver supply.

Thus,

$$P(\gamma_i = 0) \leq P\left(\sum_{n=1}^{i-1} \mathbf{1}\{\tau_n + D_n > \tau_i\} \geq c_r^k - \max_{t \in (\tau_i, (k+1)w]} \left[f_r^{P,k}(t) + f_r^{BA,k}(t)\right]\right) \tag{65}$$

let $N_r^{k,\infty}(\tau_i) = \sum_{n=1}^{i-1} \mathbf{1}\{\tau_n + D_n > \tau_i\}$,
Then,

$$P(\gamma_i = 0) \leq P\left(N_r^{k,\infty}(\tau_i) \geq c_r^k - \max_{t \in (\tau_i, (k+1)w]} \left[f_r^{P,k}(t) + f_r^{BA,k}(t)\right]\right) \tag{66}$$

$N_r^{k,\infty}(\tau_i)$ represents the number of stochastic non-reserved ride requests that are received between $(kw, \tau_i]$ and are active at time τ_i. Thus, $N_r^{k,\infty}(\tau_i)$ is similar to $N_r^k(\tau_i)$ with the main difference being that $N_r^k(\tau_i)$ is restricted to admitted non-reserved ride requests while $N_r^{k,\infty}(\tau_i)$ accounts for *all received requests* (i.e., $N_r^{k,\infty}(\tau_i)$ assumes that all requests are admitted regardless of the admission control policy). As previously described, stochastic non-reserved ride requests start arriving *after* *the beginning of the time window* ($t = kw$) according to a Poisson process with demand rate $\{\lambda_r^k(t) : t \in (kw, (k+1)w]\}$ and their ride duration follows the general distribution $g_r^k(\cdot)$. Then, the system corresponding to $N_r^{k,\infty}(\tau_i)$ can be described as a transient $\mathrm{M}_t/\mathrm{GI}/\infty$ queue that *starts empty* at $t = kw$, receives requests at the rate $\{\lambda_r^k(t) : t \in (kw, (k+1)w]\}$, has a generally distributed service rate $g_r^k(\cdot)$, and has an infinite number of servers (all requests are admitted). In this context, $N_r^{k,\infty}(\tau_i)$ (the number of active rides at time τ_i) represents the number of busy servers at time τ_i in the transient $\mathrm{M}_t/\mathrm{GI}/\infty$ queue.

□

B. Minimum Cost Flow Reformulations:

Original Formulation:

$$\min_{h_{ij}:(i,j)\in E,\, h_i:i\in R} \quad \sum_{(i,j)\in E} h_{ij} + M\sum_{i\in R}|h_i| \tag{67}$$

$$\text{s.t.} \quad \sum_{j:(i,j)\in E} h_{ij} - \sum_{j:(j,i)\in E} h_{ji} + h_i = \Delta_i \qquad \forall i \in R \tag{68}$$

$$\sum_{j:(i,j)\in E} h_{ij} \le e_i \qquad \forall i \in R \tag{69}$$

$$h_{ij} \ge 0 \qquad \forall (i,j) \in E \tag{70}$$

$$h_{ij} \in \mathbb{Z} \qquad \forall (i,j) \in E \tag{71}$$

$$h_i \in \mathbb{Z} \qquad \forall i \in R \tag{72}$$

First, observe that formulation 67–72 can be rewritten in terms of $h_{i\bullet}$ and $h_{\bullet i}$ that are defined in Equations 73 and 74. The revised formulation is given in 75–81. In this case, $h_{\bullet i}$ corresponds to drivers added to region $i \in R$ by adjusting the total number of drivers, and $h_{i\bullet}$ corresponds to drivers removed from region $i \in R$ by adjusting the total number of drivers (i.e., $h_{i\bullet}$ represents drivers that can be removed from the system to avoid having excess idle drivers).

$$h_{i\bullet} = \begin{cases} h_i & \text{if} \quad h_i > 0 \\ 0 & \text{otherwise} \end{cases} \tag{73}$$

$$h_{\bullet i} = \begin{cases} |h_i| & \text{if} \quad h_i < 0 \\ 0 & \text{otherwise} \end{cases} \tag{74}$$

$$\min_{h_{ij}:(i,j)\in E,\, h_{i\bullet}, h_{\bullet i}:i\in R} \quad \sum_{(i,j)\in E} h_{ij} + M\sum_{i\in R}[h_{i\bullet} + h_{\bullet i}] \tag{75}$$

$$\text{s.t.} \quad \sum_{j:(i,j)\in E} h_{ij} - \sum_{j:(j,i)\in E} h_{ji} + h_{i\bullet} - h_{\bullet i} = \Delta_i \qquad \forall i \in R \tag{76}$$

$$\sum_{j:(i,j)\in E} h_{ij} \le e_i \qquad \forall i \in R \tag{77}$$

$$h_{ij} \geq 0 \qquad \forall (i,j) \in E \tag{78}$$

$$h_{i\bullet}, h_{\bullet i} \geq 0 \qquad \forall i \in R \tag{79}$$

$$h_{ij} \in \mathbb{Z} \qquad \forall (i,j) \in E \tag{80}$$

$$h_{i\bullet}, h_{\bullet i} \in \mathbb{Z} \qquad \forall i \in R \tag{81}$$

Observe that due to the high costs associated with adjusting the total number of drivers, $h_{\bullet i} \leq d_i^v$ for every region i; this inequality implies that the amount of drivers added to region i is less than demand in the region. Similarly, for every region i, $h_{i\bullet} \leq s_i^v$; this inequality implies that the number of drivers disposed from region i (by adjusting the total number of drivers) is less than the virtual supply in the region. If we sum the latter two inequalities over all regions, we get inequalities 82 and 83. Then, we can rewrite those inequalities using slack variables as shown in Equations 84–86.

$$\sum_{i \in R} h_{\bullet i} \leq \sum_{i \in R} d_i^v \tag{82}$$

$$\sum_{i \in R} h_{i\bullet} \leq \sum_{i \in R} s_i^v \tag{83}$$

$$\sum_{i \in R} h_{\bullet i} + \bar{h}_d = \sum_{i \in R} d_i^v \tag{84}$$

$$\sum_{i \in R} h_{i\bullet} + \bar{h}_s = \sum_{i \in R} s_i^v \tag{85}$$

$$\bar{h}_d, \bar{h}_s \geq 0 \tag{86}$$

Intuitively, $\bar{h}_d$ is a slack variable that represents the *demand* that is satisfied through internal driver transitions (as opposed to adding external drivers $h_{\bullet i}$ by adjusting the total number of drivers). Meanwhile, $\bar{h}_s$ is a slack variable that represents the *supply* that is used to satisfy demand through internal driver transitions (as opposed to disposing off the supply $h_{i\bullet}$ by adjusting the total number of drivers). Therefore, $\bar{h}_d = \bar{h}_s$. A more rigorous approach to show that the equality holds is as follows:

Lemma. $\bar{h}_d = \bar{h}_s = \bar{h}$

Proof. First, we rearrange Equation 84 to arrive at Equation 87. Then, we can restrict the sum to regions where $\Delta_i < 0$ since by definition $d_i^v = 0$ if $\Delta_i \geq 0$, and since $h_{\bullet i} \leq d_i^v$, then $h_{\bullet i} = 0$ if $d_i^v = 0$ (where $h_{\bullet i} \geq 0$ by definition). Thus, $\Delta_i \geq 0 \Rightarrow d_i^v = 0 \Rightarrow h_{\bullet i} = 0$, and we can restrict the sum to $\Delta_i < 0$ as shown in Equation 88.
Equation 89 follows by definition of d_i^v and Δ_i when $\Delta_i < 0$.
Equation 90 follows by rearranging constraint 76. Note that since $\Delta_i < 0$ then $s_i^v = 0$ by definition,

and since $h_{i\bullet} \leq s_i^v$ then $h_{i\bullet} = 0$.

$$\begin{aligned}
\bar{h}_d &= \sum_{i \in R} d_i^v - h_{\bullet i} && (87)\\
&= \sum_{i \in R : \Delta_i < 0} d_i^v - h_{\bullet i} && (88)\\
&= \sum_{i \in R : \Delta_i < 0} -\Delta_i - h_{\bullet i} && (89)\\
&= \sum_{i \in R : \Delta_i < 0} \left[\sum_{j:(j,i) \in E} h_{ji} - \sum_{j:(i,j) \in E} h_{ij} \right] && (90)
\end{aligned}$$

Following a similar approach, we can define $\bar{h}_s$ as illustrated in Equation 91.

$$\bar{h}_s = \sum_{i \in R : \Delta_i > 0} \left[\sum_{j:(i,j) \in E} h_{ij} - \sum_{j:(j,i) \in E} h_{ji} \right] \tag{91}$$

Then, we can represent the difference between $\bar{h}_d$ and $\bar{h}_s$ as in Equation 92.
Observe that if $\Delta_i = 0$, then $\sum_{j:(j,i) \in E} h_{ji} = \sum_{j:(i,j) \in E} h_{ij}$, where this follows by constraint 76 ($h_{i\bullet} = h_{\bullet i} = 0$ since $h_{\bullet i} \leq d_i^v$, $h_{i\bullet} \leq s_i^v$ and $d_i^v = s_i^v = \Delta_i = 0$).
Thus, we can rearrange Equation 92 to get Equation 93.
Then, we can rearrange Equation 93 further to get Equations 94. Finally, note that $\sum_{i \in R} \sum_{j:(j,i) \in E} h_{ji}$ is a summation over all links in the network, and similarly $\sum_{i \in R} \sum_{j:(i,j) \in E} h_{ij}$ is a summation over all links in the network. This gives Equation 95, which proves the lemma.

$$\begin{aligned}
\bar{h}_d - \bar{h}_s &= \sum_{i \in R : \Delta_i < 0} \left[\sum_{j:(j,i) \in E} h_{ji} - \sum_{j:(i,j) \in E} h_{ij} \right] - \sum_{i \in R : \Delta_i > 0} \left[\sum_{j:(i,j) \in E} h_{ij} - \sum_{j:(j,i) \in E} h_{ji} \right] && (92)\\
&= \sum_{i \in R} \left[\sum_{j:(j,i) \in E} h_{ji} - \sum_{j:(i,j) \in E} h_{ij} \right] && (93)\\
&= \sum_{i \in R} \sum_{j:(j,i) \in E} h_{ji} - \sum_{i \in R} \sum_{j:(i,j) \in E} h_{ij} && (94)\\
&= \sum_{(i,j) \in E} h_{ij} - \sum_{(i,j) \in E} h_{ij} = 0 && (95)
\end{aligned}$$

□

Subsequently, we can add Equations 84–86 as constraints in formulation 75–81, where we use $\bar{h} = \bar{h}_d = \bar{h}_s$. The resulting formulation is shown in 96–106 (Equation 85 is first multiplied by a negative sign and then added as a constraint). Note that $\bar{h}$ must be integer since, for each region i, $s_i^v, d_i^v, h_{\bullet i}, h_{i\bullet}$ are all integer.

$$\min_{h_{ij}:(i,j)\in E,\ h_{i\bullet},h_{\bullet i}:i\in R,\ \bar{h}} \quad \sum_{(i,j)\in E} h_{ij} + M\sum_{i\in R}\left[h_{i\bullet} + h_{\bullet i}\right] \tag{96}$$

$$\text{s.t.} \quad \sum_{j:(i,j)\in E} h_{ij} - \sum_{j:(j,i)\in E} h_{ji} + h_{i\bullet} - h_{\bullet i} = \Delta_i \qquad \forall i \in R \tag{97}$$

$$\sum_{j:(i,j)\in E} h_{ij} \leq e_i \qquad \forall i \in R \tag{98}$$

$$\sum_{i\in R} h_{\bullet i} + \bar{h} = \sum_{i\in R} d_i^v \tag{99}$$

$$-\left[\sum_{i\in R} h_{i\bullet} + \bar{h}\right] = -\sum_{i\in R} s_i^v \tag{100}$$

$$h_{ij} \geq 0 \qquad \forall (i,j) \in E \tag{101}$$

$$h_{i\bullet}, h_{\bullet i} \geq 0 \qquad \forall i \in R \tag{102}$$

$$\bar{h} \geq 0 \tag{103}$$

$$h_{ij} \in \mathbb{Z} \qquad \forall (i,j) \in E \tag{104}$$

$$h_{i\bullet}, h_{\bullet i} \in \mathbb{Z} \qquad \forall i \in R \tag{105}$$

$$\bar{h} \in \mathbb{Z} \tag{106}$$

To map the problem to an equivalent min-cost flow formulation, for each region $i \in R$, we define variables $h_{ii^\star}$ that represent the total number of drivers leaving region i to adjacent regions (Equation 107). In addition, for each link $(i,j) \in E$, we define variables $h_{i^\star j} = h_{ij}$. Thus, we can define $h_{ii^\star}$ in terms of $h_{i^\star j}$ as in Equation 108. Since h_{ij} is a non-negative integer for all $(i,j) \in E$, we have that $h_{ii^\star}$ and $h_{i^\star j}$ are non-negative integers as well.

$$h_{ii^\star} = \sum_{j:(i,j)\in E} h_{ij} \qquad \forall i \in R \tag{107}$$

$$= \sum_{j:(i,j)\in E} h_{i^\star j} \qquad \forall i \in R \tag{108}$$

Then, we can express constraint 98 in terms of $h_{ii^\star}$ as $h_{ii^\star} \leq e_i$ for all regions $i \in R$. Moreover, we can express the sum of driver transitions across links $(i,j) \in E$ as shown in Equation 109.

$$\sum_{(i,j)\in E} h_{ij} = \sum_{i\in R}\sum_{j:(i,j)\in E} h_{ij} = \sum_{i\in R} h_{ii^\star} \tag{109}$$

Therefore, we can reformulate optimization problem 96–106 in terms of the newly defined variables as follows: Substitute Equation 109 in the objective function 96, replace the sum of drivers leaving a region to adjacent regions with $h_{ii^\star}$ (as in Equation 107), replace h_{ij} by $h_{i^\star j}$ and h_{ji} by $h_{j^\star i}$, replace constraint 98 with $h_{ii^\star} \leq e_i$, add Equation 108 to the constraints, add constraints that restrict $h_{i^\star j}$ to be non-negative integers for all $(i,j) \in E$, and add constraints that restrict $h_{ii^\star}$ to be non-negative integers for all $i \in R$. The revised formulation is shown in 110–122.

$$\min_{h_{i^\star j}:(i,j)\in E,\ h_{i\bullet},h_{\bullet i},h_{ii^\star}:i\in R,\ \bar{h}} \quad \sum_{i\in R} h_{ii^\star} + M\sum_{i\in R}\left[h_{i\bullet} + h_{\bullet i}\right] \tag{110}$$

$$\text{s.t.} \quad h_{ii^\star} - \sum_{j:(j,i)\in E} h_{j^\star i} + h_{i\bullet} - h_{\bullet i} = \Delta_i \qquad \forall i \in R \tag{111}$$

$$\sum_{i\in R} h_{\bullet i} + \bar{h} = \sum_{i\in R} d_i^v \tag{112}$$

$$-\left[\sum_{i\in R} h_{i\bullet} + \bar{h}\right] = -\sum_{i\in R} s_i^v \tag{113}$$

$$\sum_{j:(i,j)\in E} h_{i^\star j} - h_{ii^\star} = 0 \qquad \forall i \in R \tag{114}$$

$$0 \leq h_{ii^\star} \leq e_i \qquad \forall i \in R \tag{115}$$

$$h_{i^\star j} \geq 0 \qquad \forall (i,j) \in E \tag{116}$$

$$h_{i\bullet}, h_{\bullet i} \geq 0 \qquad \forall i \in R \tag{117}$$

$$\bar{h} \geq 0 \tag{118}$$

$$h_{ii^\star} \in \mathbb{Z} \qquad \forall i \in R \tag{119}$$

$$h_{i^\star j} \in \mathbb{Z} \qquad \forall (i,j) \in E \tag{120}$$

$$h_{i\bullet}, h_{\bullet i} \in \mathbb{Z} \qquad \forall i \in R \tag{121}$$

$$\bar{h} \in \mathbb{Z} \tag{122}$$

References

Ahuja, R.K., Magnanti, T.L., Orlin, J.B., 1993. Network Flows: Theory, Algorithms, and Applications. Prentice Hall.

Bahat, O., Bekhor, S., 2016. Incorporating ridesharing in the static traffic assignment model. Networks and Spatial Economics 16, 1125–1149.

Ban, X., Dessouky, M., Pang, J., Fan, R., 2019. A general equilibrium model for transportation systems with e-hailing services and flow congestion. Transportation Research Part B: Methodological 129, 273–304.

Banerjee, S., Freund, D., Lykouris, T., 2017. Pricing and optimization in shared vehicle systems: An approximation framework. arXiv preprint .

Banerjee, S., Kanoria, Y., Qian, P., 2018. State dependent control of closed queueing networks with application to ride-hailing. arXiv preprint .

Braverman, A., Dai, J.G., Liu, X., Ying, L., 2019. Empty-car routing in ridesharing systems. Operations Research 67, 1437–1452.

Chen, H., Zhang, K., Liu, X., Nie, Y.M., 2019. A physical model of street ride-hail. SSRN 3318557 .

Daganzo, C.F., Ouyang, Y., 2019. A general model of demand-responsive transportation services: From taxi to ridesharing to dial-a-ride. Transportation Research Part B: Methodological 126, 213–224.

Di, X., Ban, X.J., 2019. A unified equilibrium framework of new shared mobility systems. Transportation Research Part B: Methodological 129, 50–78.

Di, X., Ma, R., Liu, H., Ban, X.J., 2018. A link-node reformulation of ridesharing user equilibrium with network design. Transportation Research Part B: Methodological 112, 230–255.

Djavadian, S., Chow, J.Y., 2017. An agent-based day-to-day adjustment process for modeling 'Mobility as a Service' with a two-sided flexible transport market. Transportation research part B: methodological 104, 36–57.

Eick, S.G., Massey, W.A., Whitt, W., 1993. The physics of the $M_t/G/\infty$ queue. Operations Research 41, 731–742.

Foley, R.D., 1982. The nonhomogeneous $M/G/\infty$ queue. Opsearch 19, 40–48.

Lei, C., Jiang, Z., Ouyang, Y., 2019. Path-based dynamic pricing for vehicle allocation in ridesharing systems with fully compliant drivers. Transportation Research Part B: Methodological (forthcoming).

Li, S., Tavafoghi, H., Poolla, K., Varaiya, P., 2019. Regulating TNCs: Should Uber and Lyft set their own rules? Transportation Research Part B: Methodological 129, 193–225.

Lyft, 2019a. Bonuses and Incentives. https://help.lyft.com/hc/en-us/sections/115003494568-Bonuses-and-Incentives.

Lyft, 2019b. New York City Driver Information. https://help.lyft.com/hc/en-us/articles/115012929447-New-York-City-Driver-Information.

Lyft, 2019c. Prime Time for drivers. https://help.lyft.com/hc/en-us/articles/115012926467-Prime-Time-for-drivers.

Nie, Y.M., 2017. How can the taxi industry survive the tide of ridesourcing? Evidence from Shenzhen, China. Transportation Research Part C: Emerging Technologies 79, 242–256.

Nourinejad, M., Ramezani, M., 2019. Ride-Sourcing modeling and pricing in non-equilibrium two-sided markets. Transportation Research Part B: Methodological (forthcoming).

NYCTLC, 2019. TLC Trip Record Data. https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page.

Ozkan, E., Ward, A., 2019. Dynamic matching for real-time ridesharing. Stochastic Systems (forthcoming).

Prékopa, A., 1958. On secondary processes generated by a random point distribution of Poisson type. Annales Univ. Sci. Budapest de Eötvös Nom. Sectio Math 1, 153–170.

Qian, X., Ukkusuri, S.V., 2017. Taxi market equilibrium with third-party hailing service. Transportation Research Part B: Methodological 100, 43–63.

Ramezani, M., Nourinejad, M., 2018. Dynamic modeling and control of taxi services in large-scale urban networks: A macroscopic approach. Transportation Research Part C: Emerging Technologies 94, 203–219.

Rasulkhani, S., Chow, J.Y., 2019. Route-cost-assignment with joint user and operator behavior as a many-to-one stable matching assignment game. Transportation Research Part B: Methodological 124, 60–81.

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

Wang, J.P., Ban, X.J., Huang, H.J., 2019. Dynamic ridesharing with variable-ratio charging-compensation scheme for morning commute. Transportation Research Part B: Methodological 122, 390–415.

Wang, X., Yang, H., Zhu, D., 2018. Driver-rider cost-sharing strategies and equilibria in a ridesharing program. Transportation Science 52, 868–881.

Wolsey, L., 1998. Integer Programming. Wiley.

Xu, Z., Yin, Y., Ye, J., 2019. On the supply curve of ride-hailing systems. Transportation Research Part B: Methodological (forthcoming).

Yang, H., Yang, T., 2011. Equilibrium properties of taxi markets with search frictions. Transportation Research Part B: Methodological 45, 696–713.

Zha, L., Yin, Y., Du, Y., 2018. Surge pricing and labor supply in the ride-sourcing market. Transportation Research Part B: Methodological 117, 708–722.

Zha, L., Yin, Y., Yang, H., 2016. Economic analysis of ride-sourcing markets. Transportation Research Part C: Emerging Technologies 71, 249–266.

Zhang, K., Chen, H., Yao, S., Xu, L., Ge, J., Liu, X., Nie, M., 2019. An efficiency paradox of uberization. SSRN 3462912 .

Zhang, K., Nie, M., 2019. To pool or not to pool: Equilibrium, pricing and regulation. SSRN 3497808 .

Zhang, R., Pavone, M., 2016. Control of robotic mobility-on-demand systems: A queueing-theoretical perspective. The International Journal of Robotics Research 35, 186–203.

Zuniga-Garcia, N., Tec, M., Scott, J.G., Ruiz-Juri, N., Machemehl, R.B., 2020. Evaluation of ride-sourcing search frictions and driver productivity: A spatial denoising approach. Transportation Research Part C: Emerging Technologies 110, 346–367.