Research Article

TRR
JOURNAL OF THE TRANSPORTATION RESEARCH BOARD

Transportation Research Record
2022, Vol. 2676(6) 156–169

DOI: 10.1177/03611981211073103
journals.sagepub.com/home/trr
SAGE

Order-Dispatching Strategy Induced by Optimal Transport Plan for an Online Ride-Hailing System

Dechao Lei[1] and Yuanshan Wu[1]

Abstract

Order-dispatching strategy is fundamental for online ride-hailing systems in three major ways. First, it should be capable of effectively matching hundreds of orders to thousands of vehicles in less than 1 min, whereas both orders and vehicles are geographically widely distributed. Second, it should decide how to fairly allocate profit among participators. Third, it should unify interests of present and future. Conflicts always exist within the three objectives. We observe that the system is dynamically balanced during fine-tuned time intervals. Based on this observation, we propose a novel dispatching strategy capable of juggling the three objectives. By applying optimal transport theory, we demonstrate that, under appropriate presumptions, this dispatching strategy is ruled by the optimal transport plan realizing the Wasserstein distance between distributions of orders and vehicles. Furthermore, we develop a kit of methodological and algorithmic tools which has substantial and sensible advantages in characterizing and optimizing the system.

Keywords

decision making, planning and analysis, problem solving, transportation planning analysis and application

A modern online ride-hailing platform such as Uber, Didi Chuxing, or Lyft, on the one hand allows passengers to book routes starting from almost every corner of a city of a certain scale at every time with a smart phone, and on the other hand makes it possible for full-time, part-time, or even casual drivers to render transportation services at their greatest spatiotemporal convenience. A ride-hailing platform is a typical two-sided market, so matching demand and supply, that is, dispatching passenger orders to drivers, is a key feature of the service (*1*).

Orders are immediately responded to under many dispatching strategies. Introduced by Lee (*2*) and Lee et al. (*3*), the nearest driver or the shortest-travel-time driver is matched immediately after an order comes in. This strategy may be myopic: any single dispatch is the "optimal" choice at the moment but a sequence of consecutive dispatches following this rule one by one may not be optimal as a whole. The strategy by Seow et al. (*4*) starts from a collaborative multi-agent architecture called NTuCab. Each unit of the NtuCab is composed of N vehicles awaiting N requests and the optimal matching is aimed to minimize total pickup time among the unit. To make the computation burden arising from arbitrary permutation within the unit affordable, the size of the unit, namely N, should not be too large. Wong and Bell (*5*) aim to minimize passenger request waiting time in a heuristic approach with rolling horizon considering anticipation of future requests and traffic conditions. Wang et al. (*6*) try to optimize multiple objectives including system-wide measures and individual benefits using stable matching with rolling horizon. Bertsimas et al. (*7*) consider maximization of total profit via a backbone algorithm with a restricted set of candidate actions for a sparser problem. Xu et al. (*8*) introduce a strategy aimed at maximizing of expectation of drivers' total present value. In the design, each spatiotemporal grid is assigned a value which was learned from historical observations. If a driver is to move from grid A to grid B to fulfill an order, his reward from this action is the price of the

[1]School of Statistics and Mathematics, Zhongnan University of Economics and Law, Wuhan, Hubei, China

Corresponding Author:
Yuanshan Wu, wu@zuel.edu.cn

order plus the value difference of B to A. For a queue of limited orders to be picked up by another queue of limited drivers, the total reward of each candidate combination of dispatches is calculated and the combination with the maximum reward is to be implemented. Zhou et al. (*9*) impose a KL-divergence penalty on the maximizing process proposed by Xu et al. (*8*) and the KL-divergence is roughly defined between normalized distributions of orders and drivers after dispatch. Xu et al. (*8*) and Zhou et al. (*9*) aim to balance interests of present and future via maximizing of expectation of drivers' total present value while raising another tricky question of seeking appropriate weights for present and future. This question is thus at the core of myopia handling.

Instead of dispatching a driver immediately after an order arrives, platforms often hold unserved orders as well as vacant drivers for a certain period for better matching. Intuitively, as the length of the period increases, more demand/supply information is revealed, but more passengers may cancel orders as the waiting time increases. Akbarpour et al. (*10*) demonstrate that if the platform can identify passengers who are about to cancel and take action accordingly, then waiting to thicken the market can substantially increase successful matches. Otherwise, matching agents with the nearest distance or expected shortest pickup time is close to optimal.

In the domain of dispatch optimization research, there are generally two objectives to consider: one is oriented to passengers' interests, to maximize the total order response rate (ORR), that is, the proportion of served orders to total orders in one day, whereas the other is oriented to drivers' interests, to maximize the accumulated driver income (ADI), that is, the yield of orders served during one day. The two objectives may result in different dispatches unless all the orders have an identical price. Both ORR and ADI are relatively fair but unilateral objectives under the assumption of supply insufficiency. In this article, we study another potential objective, that is, the system's productivity which is defined as the inverse of the expectation of the driver cycle for which the basic connotation is the expected time in completing an order. In definition, the driver cycle consists of vacant time, pickup time, and service time. Vacant time is the slack of supply playing an important role in achieving equilibrium with demand, service time is the time determined by the trip of the order and is taken as exogenous, while pickup time is the time determined by the trip between the matched driver and passenger. Pickup time also constitutes an important part of a passenger's waiting time. So optimization around pickup time benefits both driver and passenger while improving system productivity.

As to dispatching structure design, our idea is to combine virtues of both instant matching and delayed matching. First, we need to fix a time window of certain length and generate the optimal dispatching plan based on information about that window. Then each upcoming order is immediately matched with a candidate driver by some greedy algorithm guided by the plan. We observe that the system is dynamically balanced if the time windows are tuned to the length of the driver cycle. Then, in our opinion, such a window is enough to provide the necessary information for optimal matching while eliminating myopia. However, the number of drivers and orders to be matched even in one window is tremendous, comparing with the unit size of NTuCab in Seow et al. (*4*). Arbitrary permutation apparently is not a feasible choice to do the optimization. Linear programming can effectively handle cases with small to middle-sized amounts of data (*11–14*). But when facing a large-scale problem like online ride-hailing order-dispatching, it seems to be powerless. In this paper, we introduce a new approach. The approach is nourished by research around the Monge–Kantorovich problem, or the optimal transport problem, to use another name. In the formulation of the problem, both supply and demand are generalized to distributions in feasible space and the optimal transport plan is the one to realize the minimum transport cost, or, to give it its formal name, the Wasserstein distance, between the pair of distributions (*15*). The theory of the optimal transport plan and Wasserstein distance has demonstrated itself to be a powerful tool in dealing with complicated matching problems. Fast numerical solutions to a balanced transport provision (the mass of supply equals that of demand) are available because of the work of Cuturi (*16*), Altschuler et al. (*17*), Peyré and Cuturi (*18*), and others. Inspired by the theory and corresponding numerical methods, we translate the original problem into an optimal transport problem. In the translation, distributions of supply and demand are calculated at grid level, and cost incurred in moving mass between grids is primarily defined as travel time. Specially , if the assumption of isotropism holds, the cost is then simplified to Euclidean distance of travel. Then our optimal dispatching plan is immediately obtained via sequentially applying a usual algorithm solving optimal transport problems and our own developed algorithm "local integralization." It is understood that the plan is at grid-to-grid level, where individuality differentials of driver and order within grid are not considered. Guided by the plan, a driver-to-passenger level greedy matching algorithm, such as the "first in first serve guided by plan" developed in this article, can complete the work of dispatching.

Preliminary

In 1781 Monge published one of his first famous works, *Mémoire sur la théorie des déblais et des remblais* ("déblai" is an amount of material that is extracted from the earth or a mine; "remblai" is material that is input into a new construction). The problem considered by Monge is as follows. Assume you have a certain amount of soil to extract from the ground and transport to places where it should be incorporated in a construction. The places where the material should be extracted and the ones where it should be transported to are all known. But the assignment has to be determined: To which destination should one send the material that has been extracted at a certain place? The answer does matter because transport is costly, and you want to minimize the total cost. Monge assumed that the transport cost of one unit of mass along a certain distance was given by the product of the mass and the distance (*19*).

Mathematically, the optimal transport problem consists of finding, from among all transformations $y = T(x)$ that push forward a source distribution $\mu(x)$ to a target $\nu(y)$, the map that minimizes the expected transportation:

$$\inf_{T} \int_{\mathcal{X}} c(x, T(x))\mu(x)dx, \quad \mu \circ T^{-1} = \nu, \tag{1}$$

where $c(x, y)$ is the externally provided cost of moving a unit of mass from x to y (*19*); μ and ν are normalized distributions supported on a measurable space $\mathcal{X}$; T is any measurable map from $\mathcal{X}$ to $\mathcal{X}$, and $\mu \circ T^{-1} = \nu$ functions in the way that, for any measurable set $A \subset \mathcal{X}$, $\mu \circ T^{-1}(A) = \mu(T^{-1}(A)) = \nu(A)$.
If μ and ν are defined in Euclidean space $\mathbb{R}^d$ and

$$c(x, y) := \|x - y\|_2$$

with $\|\cdot\|_2$ denoting the Euclidean distance, the optimal transport cost is the L_1 Wasserstein distance of μ and ν (*20*):

$$W_1(\mu, \nu) := \inf_{T} \int_{\mathbb{R}^d} \|x - T(x)\|_2 \mu(x)dx, \quad \mu \circ T^{-1} = \nu. \tag{2}$$

For generalization, Kantorovich (*15*) considers the following set of "couplings"

$$\begin{aligned}\mathcal{M}(\mu, \nu) :=&\{\pi \in \mathcal{P}(\mathbb{R}^d \times \mathbb{R}^d)\ s.t.\ \forall \text{Borel set } A, B \subset \mathbb{R}^d,\\ &\pi(A \times \mathbb{R}^d) = \mu(A), \pi(\mathbb{R}^d \times B) = \nu(B)\},\end{aligned}$$

where $\mathbb{R}^d \times \mathbb{R}^d$ is the product space of $\mathbb{R}^d$ and $\mathcal{P}(\mathbb{R}^d \times \mathbb{R}^d)$ stands for the probability measure space on $\mathbb{R}^d \times \mathbb{R}^d$. Intuitively, a coupling $\pi(\mu, \nu)$ is a joint distribution of μ and ν, such that two particular marginal distributions of π are equal to μ and ν, respectively. Instead of finding the optimal transport map, Kantorovich formulates the optimal transport problem as finding the optimal coupling,

$$\pi^* := \underset{\pi \in \mathcal{M}(\mu, \nu)}{\arg\inf} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|_2 d\pi(x, y). \tag{3}$$

Meanwhile, the L_1 Wasserstein distance between μ and ν is

$$W_1(\mu, \nu) := \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|_2 d\pi^*(x, y). \tag{4}$$

W_1 possesses all the properties of distance, especially the property of triangle inequality. Optimal transport plan (OTP) is used to address the solution when it is of little interest to verify whether the coupling is precisely a map.

When the two measures μ and ν are supported on a discrete set $\{x_i\}_{i=1}^n$, where $x_i \in A$ for $i = 1, \ldots n$ and $A \subset \mathbb{R}^d$ is bounded, the relaxation of optimal transport from Kantorovich (*15*) becomes a linear programming problem, which can be solved effectively for small to medium size. We identify μ and ν as the vectors located on the simplex

$$\Delta_n := \{w \in \mathbb{R}^d : \sum_{i=1}^{n} w_i = 1, \text{ and } w_i \geqslant 0, i = 1, \ldots, n\},$$

whose entries denote the weight of each distribution assigned to the points of $\{x_i\}_{i=1}^n$. Let $C \in \mathbb{R}^{n \times n}$ be the pair-wise distance matrix, where $C_{i,j} = \|x_i - x_j\|_2$, and 1_n be the all-ones vector with n elements. We denote by $\mathcal{M}(\mu, \nu)$ the set of coupling matrices between μ and ν, that is,

$$\mathcal{M}(\mu, \nu) := \{P \in \mathbb{R}^{n \times n} : P1_n = \mu, P^{\mathrm{T}}1_n = \nu\}. \tag{5}$$

Let $\langle\langle \cdot, \cdot \rangle\rangle$ denote the summation of the element-wise multiplication, such that, for any two matrix $A = (a_{ij}), B = (b_{ij}) \in \mathbb{R}^{n \times n}$, $\langle\langle A, B \rangle\rangle = \Sigma_{i=1}^n \Sigma_{j=1}^n a_{ij} b_{ij}$. According to the Kantorovich formulation in Equations 3 and 4, calculating the Wasserstein distance between μ and ν thus is equivalent to solving the optimization problem

$$W_1(\mu, \nu) := \min_{P \in \mathcal{M}(\mu, \nu)} \langle\langle P, C \rangle\rangle, \tag{6}$$

which is a linear program with $O(n)$ linear constraints. Practical algorithms to solve the problem in Equation 6 through linear programming require a computational time of order $O(n^3 \log(n))$ for fixed d (*21*). When the size of the problem grows, its solution can be accelerated significantly through the addition of an entropic regularization and a Sinkhorn-type iterative algorithm (*21*, *22*).

It assumes that the total masses of the given distributions are equal, which often does not hold in practice. Therefore, it is more realistic to generalize the problem and the Wasserstein metric defined by Equations 1 to 6 to adapt to distributions with unbalanced masses. Much work has been done in this direction (*23–26*). In general, in the case of unbalanced transport, it allows

$$\int_{\mathbb{R}^d} \mu(x)dx \neq \int_{\mathbb{R}^d} \nu(y)dy;$$

without loss of generality, assuming

$$\int_{\mathbb{R}^d} \mu(x)dx \geqslant \int_{\mathbb{R}^d} \nu(y)dy,$$

the unbalanced L_1 optimal transport problem solves

$$U(\mu,\nu) = \min_{\substack{0 \leqslant \tilde{\mu} \leqslant \mu, \\ \int \tilde{\mu}(x)dx = \int \nu(y)dy}} W_1(\tilde{\mu},\nu). \tag{7}$$

In the above, $\tilde{\mu}$ and ν are both standardized measures and μ is a greater measure than $\tilde{\mu}$ everywhere on $\mathbb{R}^d$. Compared with balanced problems, there is an additional outer layer of minimization over the set of distributions which are "no greater than"μ everywhere and have total mass equal to ν. This additional minimization work surely gives rise to a much greater computation burden.

Methodology

In this section, we first refine major characteristics of the system from the perspective of the supply chain and then reformulate our objective, optimization of system productivity, into an optimal transport problem.

Definitions

Definition 1. *Let $\mathcal{X}$ denote the 2-dimensional geographic space where positions of passenger and driver dwell. The function of order generation rate at time t, denoted by $\lambda(x)$, is the limit of count of orders generated at x in $[t, t + \Delta t)$ divided by Δt when $\Delta t \to 0$. The function of active drivers at time t, denoted by $m(x)$, is the count of vacant drivers at x plus the count of drivers in serving but expected to drop off at x. Let $\Lambda := \int_{\mathcal{X}} \lambda(x)dx$ and $M := \int_{\mathcal{X}} m(x)dx$.*

Definition 2. *(standardized) Distribution of active drivers at t, denoted by μ_{driver}, is the measure of driver count on $\mathcal{X}$, namely the measure at some $x \in \mathcal{X}$ is defined as $m(x)/M$. (standardized) Distribution of orders at t, denoted by ν_{order}, is the measure of order generation rate on $\mathcal{X}$, namely the measure at some $x \in \mathcal{X}$ is defined as $\lambda(x)/\Lambda$.*

Since the precise position of each driver or order is individual, we actually calculate the measures by grid after gridding the operational geographical space (see "Geographical mesh").

Definition 3. *Response time to an order, denoted by d_{res}, is the duration from the order request being sent to a dispatch being arranged, where if the driver is in serving when dispatched, the arrangement actually happens once the serve ends; the vacant time of a driver, denoted by d_{vac}, is the duration from the ending of the last drop-off to the next dispatch being arranged; pickup time, denoted by d_{pic}, is the duration from a dispatch being arranged to the passenger being picked up; service time, denoted by d_{ser}, is the duration from a passenger being picked up to the passenger being dropped off. Fulfillment time, denoted by d_{ful}, is the summation of d_{pic} and d_{ser}. Driver cycle, denoted by T_{dri}, is the summation of d_{vac} and d_{ful}. Order cycle, denoted by T_{ord}, is the summation of d_{res} and d_{ful}. The corresponding expectations are $\text{E}d_{\text{ful}}$, $\text{E}T_{\text{dri}}$ and $\text{E}T_{\text{ord}}$ respectively, where the expectations are taken over $\mathcal{X}$ at time t.*

Driver cycle and order cycle are the time costs in fulfilling an order for driver and passenger respectively. They are critical factors affecting service quality and driver income level (*1*).

Definition 4. Λ *is also the demand (D) of the system. Supply potential of the system at time t, denoted by S_{pot}, is defined as $S_{\text{pot}} := \frac{M}{\text{E}d_{\text{ful}}}$; actual supply, denoted by S_{act}, is defined as $S_{\text{act}} := \frac{M}{\text{E}T_{\text{dri}}}$. Ratio of supply to demand for the system at time t, denoted by $r_{s/d}$, is defined as $r_{s/d} := \frac{S_{\text{pot}}}{D} = \frac{M}{\Lambda \text{E}d_{\text{ful}}}$. System productivity is defined as $\gamma := 1/\text{E}T_{\text{dri}} = 1/(\text{E}d_{\text{vac}} + \text{E}d_{\text{pic}} + \text{E}d_{\text{ser}})$.*

As introduced previously, system productivity is our primary objective to optimize.

Assumptions

Assumption 1. *The system is dynamically balanced all through in that*

(i) given M and Λ, $\text{E}d_{\text{vac}}$ is self-adjusted to guarantee system balance in the sense that $S_{\text{act}} = D$, that is, $M = \Lambda(\text{E}d_{\text{vac}} + \text{E}d_{\text{ful}})$ and

(ii) sustainable $\text{E}d_{\text{vac}}$ lies in a positive interval $(0, U_{\text{E}d_{\text{vac}}})$; if $\text{E}d_{\text{vac}}$ surpasses $U_{\text{E}d_{\text{vac}}}$, M will gradually decline till $\text{E}d_{\text{vac}}$ falls back below $U_{\text{E}d_{\text{vac}}}$.

The vacant rate, $\frac{d_{\text{vac}}}{T_{\text{dri}}} \times 100\%$ is the major factor that affects the possibility of a driver logging off/on. The upper bound of $\text{E}d_{\text{vac}}$ could be the threshold beyond which an ordinary driver will get negative profit and be very likely to log off.

By Definition 4, $r_{s/d}$ should be above 1 all through under the assumption.

Assumption 2. $\mathrm{Ed}_{\mathrm{ser}}$ *is exogenous, that is, independent of the dispatching strategy.*

For each order, d_{ser} is determined by the its travel course and travel speed, where both are almost given unless the individuality differential of driver is considered. A global order-dispatching strategy usually takes any driver as ordinary.

Assumption 3. *Isotropism. Let $c_t(x,y)$ be the travel time from a point $x \in \mathcal{X}$ to another point $y \in \mathcal{X}$ at moment t, and $||x-y||_2$ the Euclidian distance between x and y, then $c_t(x,y) = k(t)||x-y||_2$, where $k(t)$ is a global speed factor dependent on t only.*

This assumption is for the simplification of $c_t(x,y)$. When $c_t(x,y)$ is unobtainable, $||x-y||_2$ may not be a bad substitute.

Isotropism is also for the formulation of the Wasserstein distance, whose property of triangle inequality is critical in myopia control (see Theorem 3). Actually, the cost function has only to possess the property of triangle inequality to make the total cost of the optimal transport possess the same property (*21*). This requirement on the cost function is not excessive in most cases. Furthermore, the computation efficiency of the numerical method is less affected by the specific form of cost function (*16–18*).

Theoretical Results

Theorem 1. *Under Assumption 1, each order can be matched to a driver and*

(i) ORR = 100% *and* ADI = *total order prices*,

(ii) $\mathrm{Ed}_{\mathrm{res}} \approx 0$.

Under Assumptions 1 and 2,

(iii) γ *is optimized while* $\mathrm{Ed}_{\mathrm{pic}}$ *is minimized.*

Proof. *(i) and (iii) are immediate deductions of Assumptions 1 and 2. For (ii), it is trivial when both order generates and order completes with no randomness; otherwise assume that the queuing is "M/M/n" type which adequately incorporates randomness, that is, both the time between successive order generations and the time to complete an order are exponentially distributed while the system has n servers (active drivers), then by queueing theory (27), the expected response time*

$$
\begin{aligned}
\mathrm{Ed}_{\mathrm{res}} &= \frac{n^n r_{s/d}^{-(n+1)}/[n!\Lambda(1-r_{s/d}^{-1})^2]}{\sum_{k=0}^{n-1} \frac{1}{k!}(nr_{s/d}^{-1})^k + \frac{1}{n!}\frac{(nr_{s/d}^{-1})^n}{1-r_{s/d}^{-1}}} \\
&\overset{(a)}{\leqslant} \frac{n^n r_{s/d}^{-(n+1)}/[n!\Lambda(1-r_{s/d}^{-1})^2]}{\sum_{k=\lceil nr_{s/d}^{-1}\rceil-1}^{n-1} \frac{1}{k!}(nr_{s/d}^{-1})^k + \frac{1}{n!}\frac{(nr_{s/d}^{-1})^n}{1-r_{s/d}^{-1}}} \\
&\overset{(b)}{\leqslant} \frac{n^n r_{s/d}^{-(n+1)}/[n!\Lambda(1-r_{s/d}^{-1})^2]}{\sum_{k=\lceil nr_{s/d}^{-1}\rceil-1}^{n-1} \frac{1}{n!}(nr_{s/d}^{-1})^n + \frac{1}{n!}\frac{(nr_{s/d}^{-1})^n}{1-r_{s/d}^{-1}}} \\
&= \frac{r_{s/d}^{-1}/[\Lambda(1-r_{s/d}^{-1})^2]}{(n+1-\lceil nr_{s/d}^{-1}\rceil) + \frac{1}{1-r_{s/d}^{-1}}},
\end{aligned}
\tag{8}
$$

where n is the quantity of active drivers (servers), "$\lceil x \rceil$" stands for the minimum integer no less than x, inequality (a) holds because the items with $k = 0, \ldots, \lceil nr_{s/d}^{-1}\rceil - 2$ are just omitted from the summation, inequality (b) holds as

$$
1 \geqslant \frac{nr_{s/d}^{-1}}{\lceil nr_{s/d}^{-1}\rceil} > \frac{nr_{s/d}^{-1}}{\lceil nr_{s/d}^{-1}\rceil + 1} > \ldots > \frac{nr_{s/d}^{-1}}{n} \text{ and}
$$

$$
\begin{aligned}
\frac{(nr_{s/d}^{-1})^n}{n!} &= \frac{(nr_{s/d}^{-1})^{\lceil nr_{s/d}^{-1}\rceil-1}}{(\lceil nr_{s/d}^{-1}\rceil-1)!} \times \frac{nr_{s/d}^{-1}}{\lceil nr_{s/d}^{-1}\rceil} \times \ldots \times \frac{nr_{s/d}^{-1}}{n} \\
&< \frac{(nr_{s/d}^{-1})^{n-1}}{(n-1)!} < \ldots < \frac{(nr_{s/d}^{-1})^{\lceil nr_{s/d}^{-1}\rceil}}{(\lceil nr_{s/d}^{-1}\rceil)!} \\
&\leqslant \frac{(nr_{s/d}^{-1})^{\lceil nr_{s/d}^{-1}\rceil-1}}{(\lceil nr_{s/d}^{-1}\rceil-1)!}.
\end{aligned}
$$

Since $r_{s/d}>1$ by Assumption 1, $\mathrm{Ed}_{\mathrm{res}} \to 0$ *when $n \to \infty$ by Inequality 8.*

(*i*) tells us that the optimization of ORR or ADI is ineffective under Assumptions 1 and 2. (*ii*) demonstrates that $\mathrm{Ed}_{\mathrm{res}}$ is close to 0 and cannot be optimized any more. It holds even when $r_{s/d}$ is only slightly above 1. (*iii*) shows that since $\mathrm{Ed}_{\mathrm{vac}}$ is self-adjusting and bounded by Assumption 1, $\mathrm{Ed}_{\mathrm{ser}}$ is exogenous by Assumption 2, the only way to improve system productivity is to minimize $\mathrm{Ed}_{\mathrm{pic}}$. Putting together (*i*), (*ii*), and (*iii*), both $\mathrm{ET}_{\mathrm{dri}}$ and $\mathrm{ET}_{\mathrm{ord}}$ are minimized while ADI is fixed under the strategy and assumptions. Since $\mathrm{ET}_{\mathrm{dri}}$ and $\mathrm{ET}_{\mathrm{ord}}$ are the expected costs in time per order, then it can be said that profit per order is almost maximized simultaneously under the strategy and assumptions.

The dynamic balance makes it reasonable to construct a static matching between current active drivers and upcoming orders. (*iii*) of Theorem 1 indicates the objective to optimize the static matching. We demonstrate this idea as follows.

Theorem 2. *Let the strategy to minimize* $\mathrm{Ed_{pic}}$ *be implemented for each of the successive windows* $[t_i, t_{i+1}), i = 0, \ldots, L-1,$ *where* t_i *satisfies* $\int_{t_i}^{t_{i+1}} \Lambda(s)ds = M_i, i = 0, \ldots, L-1$ *and L is the count of such windows. Assume a driver logs off/on only at* $t_i, i = 1, \ldots, L$. *Then*
(i) minimization of $\mathrm{Ed_{pic}}$ *for each window is an independent optimal transport problem and*
(ii) the optimal dispatching map is ruled by π_i^*, the OTP from μ_{driver}^i to ν_{order}^i in the sense: $M\pi_i^*(x,y)$ drivers at x are to pick up equal amount of passengers at y.

Specially , under Assumption 3,

$$\min \mathrm{Ed}_{\text{pic}}^i = k(t_i) W_1(\mu_{\text{driver}}^i, \nu_{\text{order}}^i).$$

Proof. *Without loss of generality, we focus on optimization in one window and omit its superscript.*

Let the two measures μ_{driver} *and* ν_{order} *be supported on a discrete set of n grids* $\{x_i\}_{i=1}^n$, *where* $x_i \in A$ *for* $i = 1, \ldots n$ *and* $A \subset \mathcal{X}$ *is bounded. Since* μ_{driver} *and* ν_{order} *are standardized measures, they are vectors located on simplex* $\Delta_n := \{w \in \mathbb{R}^d : \Sigma_{i=1}^n w_i = 1,$ *and* $w_i \geqslant 0, i = 1, \ldots, n\}$. *Let* $C \in \mathbb{R}^{n \times n}$ *be the pair-wise travel time matrix, where* $C_{i,j} = c_t(x_i, x_j)$, *and* 1_n *be the all-ones vector with n elements. We denote by* $\mathcal{M}(\mu_{\text{driver}}, \nu_{\text{order}})$ *the set of coupling matrices between* μ_{driver} *and* ν_{order}, *following Equation 5. Minimizing* $\mathrm{Ed_{pic}}$ *is equivalent to solving*

$$\min_{P \in \mathcal{M}(\mu_{\text{driver}}, \nu_{\text{order}})} \langle\langle P, C \rangle\rangle,$$

which is thus an optimal transport problem. Specially when Assumption 3 holds, we have

$$\begin{aligned} \min \mathrm{Ed_{pic}} &= \min_{P \in \mathcal{M}(\mu_{\text{driver}}, \nu_{\text{order}})} \langle\langle P, C \rangle\rangle \\ &= k(t) \min_{P \in \mathcal{M}(\mu_{\text{driver}}, \nu_{\text{order}})} \langle\langle P, D \rangle\rangle \\ &= k(t) W_1(\mu_{\text{driver}}, \nu_{\text{order}})), \end{aligned}$$

where D is an $n \times n$ *matrix with entity* $D_{i,j} = \|x_i - x_j\|_2$.

The independence of optimization in each window relies on:

(i) there being no surplus of driver or order after optimization in each window. It is just the virtue of balance.
(ii) a driver logging off independently with dispatching strategy. All drivers will be redistributed by the destinations of orders in the window. The redistribution and consequently the distribution of logoff is independent of the dispatching strategy implemented unless a driver's individuality differential is to be considered.
(iii) a driver logs on independently with dispatching strategy. The new logon has no a priori information about matches under the strategy in the near past.

Theorem 2 sets up a frame for the optimization work under ideal conditions. Under the frame, the optimal transport plan is 100% executed for each window. Then the distributions of active drivers as well as orders to fulfill in the next window are independent of the strategy implemented in the current window. In this sense, there will be no myopia under the frame. Actually, since the windows are set under $\int_{t_i}^{t_{i+1}} \Lambda(s)ds = M_i$ where Λ can only be predicted with error, some drivers may have to stay idle through the current window and some orders may have to be deferred. The surplus of supply/demand is surely affected by the strategy implemented in current window in relation to position and quantity and may be a part of the distributions in the next window. So myopia may happen in the optimization sequences. We develop a theorem to control myopia arising from a surplus of drivers and similar results hold for that arising from deferred orders.

Theorem 3. *Let* $D_{\mathcal{X}} : \sup_{x_1, x_2 \in \mathcal{X}} \|x_1 - x_2\|_2$ *be the diameter of* $\mathcal{X}$, $\Delta\mu_{\text{driver}}^i$ *the surplus of driver after actual dispatch in window* $[t_i, t_{i+1})$ *under the strategy,* $\widetilde{\mu}_{\text{driver}}^{i+1}$ *the affected distribution of active drivers in* $[t_{i+1}, t_{i+2})$. *Then under Assumption 3 we have*

$$\begin{aligned} |W_1(\widetilde{\mu}_{\text{driver}}^{i+1}, \nu_{\text{order}}^{i+1}) - W_1(\mu_{\text{driver}}^{i+1}, \nu_{\text{order}}^{i+1})| & \\ \overset{(a)}{\leqslant} W_1(\widetilde{\mu}_{\text{driver}}^{i+1}, \mu_{\text{driver}}^{i+1}) & \\ \overset{(b)}{\leqslant} \frac{M_i}{M_{i+1}} D_{\mathcal{X}} \int_{\mathcal{X}} \Delta\mu_{\text{driver}}^i dx. & \end{aligned}$$

Proof. *Inequality* (a) *is by property of Wasserstein distance. Following Equation 2,*

$$\begin{aligned} W_1(\widetilde{\mu}_{\text{driver}}^{i+1}, \mu_{\text{driver}}^{i+1}) &= \inf_T \int_{\mathcal{X}} \|x - T(x)\|_2 \mu(x) dx \\ &\leqslant \int_{\mathcal{X}} \|x - T'(x)\|_2 \widetilde{\mu}_{\text{driver}}^{i+1}(x) dx, \end{aligned}$$

where T' *is any map satisfying* $\widetilde{\mu}_{\text{driver}}^{i+1} \circ T'^{-1} = \mu_{\text{driver}}^{i+1}$.

Note that the difference between $\widetilde{\mu}_{\text{driver}}^{i+1}$ *and* $\mu_{\text{driver}}^{i+1}$ *lies only on* $\frac{M_i}{M_{i+1}} \Delta\mu_{\text{driver}}^i$ *which is nonnegative everywhere, then the most costly transport* T' *between* $\widetilde{\mu}_{\text{driver}}^{i+1}$ *and* $\mu_{\text{driver}}^{i+1}$ *is to move a mass equal to* $\frac{M_i}{M_{i+1}} \int_{\mathcal{X}} \Delta\mu_{\text{driver}}^i dx$. *Since the distance* *to move any piece of mass is upper bounded by $D_{\mathcal{X}}$, we thus obtain inequality* (*b*).

In operational circumstances, $\frac{M_i}{M_{i+1}}$ is close to 1 and $D_{\mathcal{X}}$ is less than 60 km. Then by Theorem 3, if only the surplus is trivial compared with the count of active drivers, myopia in respect of possible change on $\mathrm{Ed}_{\mathrm{pic}}$ will not be serious. It is also based on this virtue that we argue that the length window of the driver cycle provides enough market information for optimization.

Dispatching Structure

The dispatching structure consists of two layers: the fundamental layer generating OTP within pairs of the source grid, where the vacant driver locates, and target grid, where the order request is sent; and the dispatching layer, executing an instant match guided by the optimal transport plan. The fundamental layer is critical and operates consistently with Theorems 2 and 3 while the dispatching layer applies in some greedy algorithms. Applicable algorithms for each layer are developed in the next subsection.

Algorithms

For generalization, we formulate the following equations for an unbalanced optimization problem. The balanced one is a special case therein.

$$U(\mu_{\mathrm{driver}}, \nu_{\mathrm{order}}) = \left\{ \begin{array}{ll} \min & W_1(\tilde{\mu}_{\mathrm{driver}}, \nu_{\mathrm{order}}) \\ \text{s. t.} & 0 \leq \tilde{\mu}_{\mathrm{driver}} \leq \mu_{\mathrm{driver}}, \\ & \int \tilde{\mu}_{\mathrm{driver}}(x)dx = \int \nu_{\mathrm{order}}(y)dy, \\ & N \times \pi(\tilde{\mu}, \nu)(x, y) \in \mathbb{N} \quad \text{for} \quad \forall (x, y) \in \mathbb{R}^2 \end{array} \right\}$$

where $\mathbb{N}$ is the set of positive integers. Equation 9 is an unbalanced optimal transport problem with additional integer constraints on mass transported. While unbalance scales up the space in which a feasible solution exists, the integer constraint erodes optimum continuity with parameters. One can take the comprehensive problem as an integer linear programming problem while enduring the computation cost as follows,

$$U(\mu_{\mathrm{driver}}, \nu_{\mathrm{order}}) = \left\{ \begin{array}{ll} \min & \langle\langle P, C \rangle\rangle \\ s.\ t. & P \in \{A \in \mathbb{N}^{n \times n} : A\mathbf{1}_n \leq \mu, A^T\mathbf{1}_n = \nu\} \\ & \sum_{i=1}^{n} (A\mathbf{1}_n)_i = \sum_{i=1}^{n} (A^T\mathbf{1}_n)_i \end{array} \right\}$$

where n stands for the scale of discretization and C the pair-wise distance matrix.

There are two factors which may neutralize the effort to solve Equation 10 directly. One is the relatively high computation cost mentioned above and the other is that errors in predicting μ_{driver} and ν_{order} will invalidate the optimality. Instead, we take it as balanced when deviation from balance is small, apply a usual algorithm for balanced problems on it, and generate the decimal OTP. Integer OTP is obtained by optimally replacing each piece of the decimal OTP either with its floor or with its ceiling under supply and demand constraints. To find the replacements with minimum total transport costs, we proceed as follows. First, we initialize each piece of integer OTP to its floor. Then, one by one in ascending order of unit transport cost, we flip each piece to its ceiling unless rejected by related supply or demand constraints. The process stops either when all mass is completely transported or when "flipping" has been tried on each piece. We streamline these ideas by Algorithm 1. Algorithm 2 handles remainders of supply and demand after the "flipping" process if there are any. Algorithm 3 is for instant dispatching guided by the integer OTP.

Application

In this section, we apply the methodology developed above to a real order-dispatching problem.

The Data

The data involved is disclosed by the Didi Chuxing incorporation. We use a 24 h record of the city Chengdu. The record consists of orders being picked up and dropped off by name-masked drivers. Tracks of vehicles (drivers) are marked with timestamps in longitude and latitude coordinates. The total count of orders is more than 180,000 and about 35,000 drivers stay in the system at all times. The metropolis of Western China, Chengdu, lies on the broad Chengdu Plain with no big river running through; terrain advantage and a developed network of highroads make traffic within the city almost isotropic.

Figures 1 and 2 show the 24-h diagrams of strength of supply and demand respectively; the strength of supply is measured by the count of active drivers at the beginning of a minute and demand by count of orders generated in the minute.

Figure 3 shows the dynamics of average time (in minutes) to pick up as well as to serve an order in 24 h; the sum of pickup time d_{pic} and service time d_{ser} is fulfillment time d_{ful} by previous definition. Dividing the count of active drivers by average d_{ful}, we get the ratio of supply to demand $r_{s/d}$ and its evolution trend is shown in Figure 4. Obviously, from 8:00 to 23:00, during which the system is most active (87 percent of total orders are generated during the period), the ratio stays slightly above 1. It thus

Algorithm 1: Local integralization

```
Input: decimal Plan(source, target, plan),
       Supply(source, remain), Demand(target,
       remain)
Output: integer Plan
Step 1 Initialization;
foreach row in Plan do
  Plan.plan=floor(Plan.plan × sum(Demand.remain));

  Supply[source=Plan.source].remain − Plan.plan;
  Demand[target=Plan.target].remain − Plan.plan;
end
Step 2 Check and handle over dispatch;
foreach source in Plan do
  over_dispatch= − Supply[source=Plan.source].remain;

  if over_dispatch> 0 then
    sort Plan[source=Plan.source] by cost
    descending;
    reduce Plan.plan in the descending order
    gradually till eliminate over dispatch;
  end
end
Step 3 Local optimization;
sort Plan by cost ascending;
to_dispatch=sum(Demand.remain);
foreach row in Plan do
  if Supply[source=Plan.source].remain > 0 and
    Demand[target=Plan.target].remain > 0 then
      Plan.plan + 1;
      Supply[source=Plan.source].remain − 1;
      to_dispatch − 1;
      Demand[target=Plan.target].remain − 1;
  end
  if to_dispatch=0 then
    break;
  end
end
Step 4 Handle remainder;
Supply_remainder=Supply [remain> 0];
  Demand_remainder=Demand[remain> 0];
Plan_remainder=Algorithm 2(Supply_remainder,
Demand_remainder);
Step 5 Plan=Plan.Append(Plan_remainder).
```

validates Assumption 1 and also suggests that $U_{\mathrm{Ed}_{\mathrm{vac}}}$ is small compared with $\mathrm{Ed}_{\mathrm{ful}}$.

Figure 5 records the event counts of driver logoff/logon. The logon curve experiences three jumps: at around 0:00, at around 7:00 and at around 9:00; almost simultaneous but less obvious jumps of order generation can be observed. The logoff curve is much more continuous than the logon curve; it slowly arrives at its first peak at around 1:00 and second at around 10:30 and then stays at a nearly constant level; the two intervals are exactly the shift times for most drivers. Drivers response to market demand via dynamical logon and logoff after estimation of personal return on investment all the time, and unconsciously leave behind a market equilibrium.

Algorithm 2: Dispatch by ascending costs

```
Input: Supply, Demand
Output: Plan
Cost(source, target, cost)=Stacked costs for each
permuted pair of (source, target) in (Supply,
Demand);
sort Cost by cost ascending;
to_dispatch=sum(Demand.remain);
foreach row in Cost do
  feasible=min(Supply[source=Cost.source].remain),
  Demand[target=Cost.target].remain);
  Plan.Append(source, target, feasible);
  to_dispatch − feasible;
  Supply[source=Cost.source].remain − feasible;
  Demand[target=Cost.target].remain − feasible;
  if to_dispatch=0 then
    break;
  end
end
```

Algorithm 3: First in first serve guided by plan

```
Input: Driver(id_driver, source, time, matched=False),
Order(id_order, target, time, matched=False),
Plan(source, target, remain=plan)
Output: Match(id_driver, id_order)
sort Order by time ascending;
foreach id_order in Order do
  target=Order[id_order].target;
  Sources=Plan[Plan[target=Order.target].remain> 0].source;
  if count(Sources)> 0 then
    FeasibleDrivers=Driver[source in Sources and
    matched=False]
    sort FeasibleDrivers by time ascending;
    id_driver=FeasibleDrivers[1].id_driver;
    source=FeasibleDrivers[1].source;
    Match.Append(id_driver, id_order);
    Driver[id_driver].matched=True;
    Order[id_order].matched=True;
    Plan[source=source and
    target=target].remain − 1.
  end
end
```

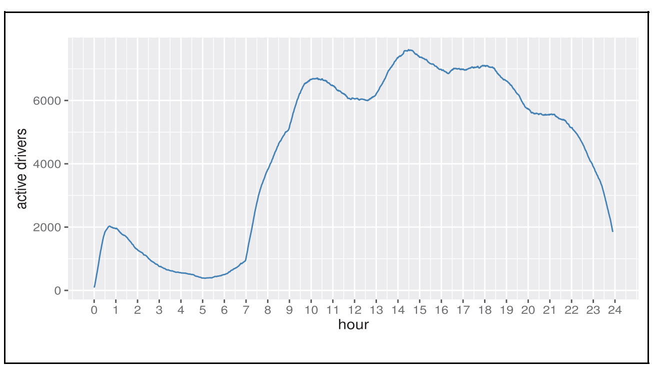

Figure 1. Active drivers of the city in 24 h.

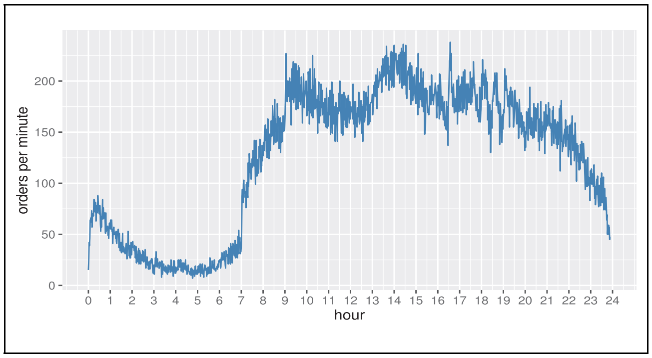

Figure 2. Order generation rate of the city in 24 h.

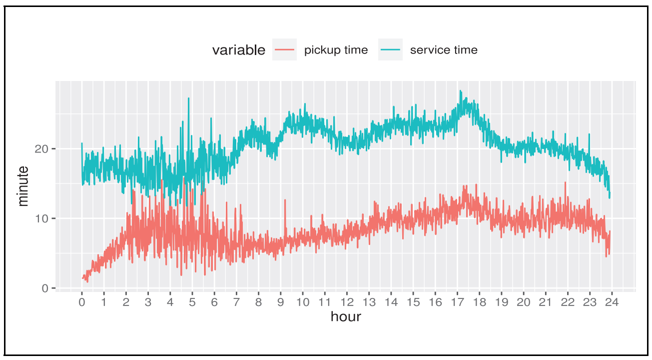

Figure 3. Diagram of order pickup time and service time.

Preparation of μ_{driver} *and* ν_{order}

Time Window. We recount Ed_{ful} in Figure 6 and find it is relatively stable. Additionally, with information provided by Figure 4, Ed_{vac} is small compared with Ed_{ful} almost all through. Encouraged by these, and also for convenience of computation, we uniformly split the 24 h of daytime into 48 equal-length time intervals such that each interval is 30 min long and succeeds the previous one. For each geographical grid (defined in "Geographical mesh"), orders generated are counted by interval. Active driver for the grid is summed up by vacant drivers in the grid at the beginning and drivers anticipated to be released from serving in the grid during the interval. However, driver logon/logoff during the interval is regarded as unpredictable at grid level and omitted in the distribution calculation. These settings and the way to calculate distributions will possibly cause unbalance. For illustration, deviation from balance, measured in supply/demand – 1, is computed for each interval and shown in Figure 7. It demonstrates that the unbalance is negligible, especially for those intervals with high order generation rate.

Geographical Mesh. The geographical gridding is conducted with parameters in Table 1.

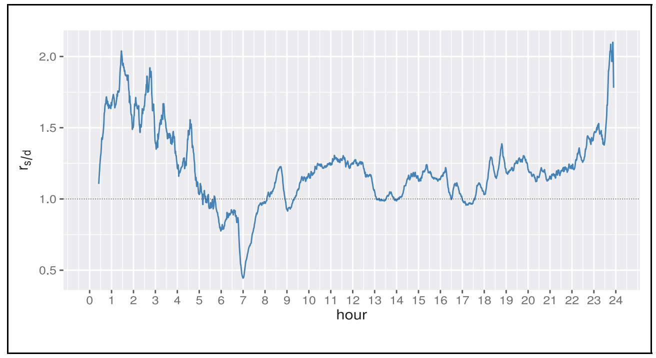

Figure 4. Ratio of supply to demand in 24 h.

Figure 5. Drivers logon and logoff rates in 24 h.

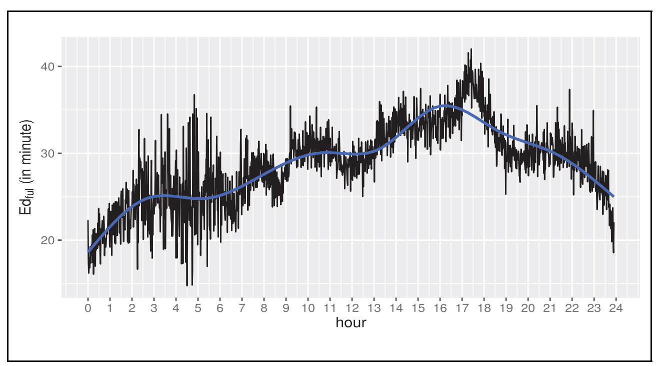

Figure 6. Diagram of order fulfillment time Ed_{ful}.

The mesh is sized at 450×500 but the actual count of grids with positive density of either supply or demand is precisely 9183. Transport cost between each pair of grids is computed as the Euclidean distance between their centers.

Results

Since we have no access to the point-to-point cost function of travel time, we do optimization under Assumption 3 all through. It is understood that d_{pic}

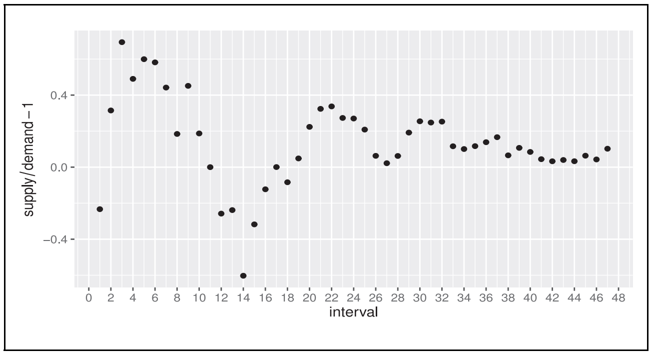

Figure 7. Deviation from balance of the isometric intervals.

Table 1. Parameters for Geographical Gridding

	Longitude (East)	Latitude (North)
Start	103.48	30.18
End	103.70	31.34
Span	1.22	1.16
Steps (number)	450	500
Step span	0.0027	0.0023
Step span (km)	0.26	0.26

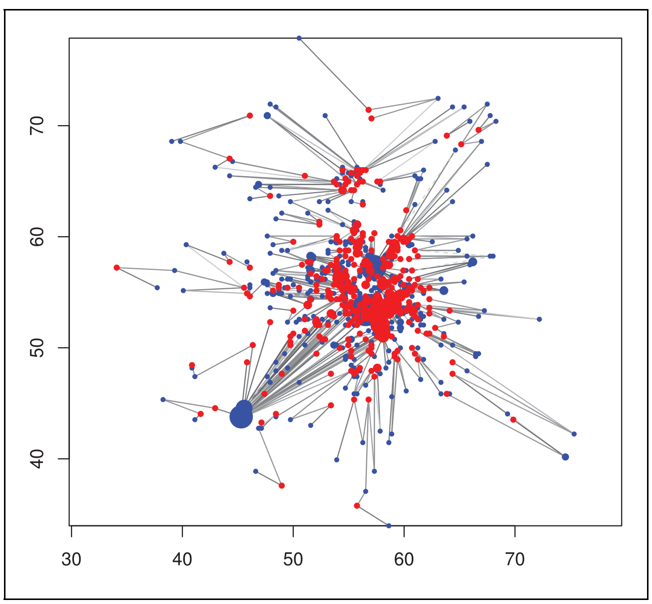

Figure 8. Optimal transport plan (OTP) for interval 9. Each blue/red disk stands for a mass of supply/demand measured in disk area. Each gray line stands for a piece of OTP and the mass transported by that piece is measured in the gray level of the line. Coordinates originate from the southwesternmost grid in operation and axes are ticked every 10 km.

differentiates with the corresponding Euclidean distance to travel only by a speed factor under the assumption.

Table 2. Ed_{pic} Comparison

Interval	S/D	dis0	dis1	dis2	dis1−dis2	dis0−dis2
2	0.77	1.32	1.55	1.37	0.18	−0.05
3	1.31	1.95	2.37	1.64	0.72	0.31
4	1.69	2.59	2.44	1.45	0.98	1.14
5	1.49	3.58	2.62	1.65	0.96	1.93
6	1.60	3.28	2.82	1.68	1.14	1.60
7	1.58	3.59	2.80	1.73	1.07	1.86
8	1.44	3.66	3.27	2.07	1.20	1.59
9	1.18	3.69	3.00	2.29	0.71	1.40
10	1.45	3.69	2.39	1.73	0.65	1.96
11	1.19	3.91	2.88	2.48	0.41	1.44
12	1.00	3.36	2.96	2.99	−0.04	0.37
13	0.74	3.32	3.40	2.93	0.46	0.38
14	0.76	2.59	3.08	2.83	0.25	−0.24
15	0.40	2.20	2.51	1.66	0.85	0.54
16	0.68	1.79	1.54	1.29	0.26	0.50
17	0.88	1.62	1.28	1.16	0.13	0.47
18	1.00	1.68	1.22	1.24	−0.02	0.44
19	0.92	1.90	1.37	1.25	0.12	0.65
20	1.05	1.94	1.11	1.05	0.06	0.89
21	1.22	2.08	0.90	0.76	0.14	1.33
22	1.32	2.20	0.77	0.59	0.18	1.61
23	1.34	2.25	0.74	0.56	0.18	1.69
24	1.27	2.30	0.89	0.66	0.23	1.64
25	1.27	2.55	0.89	0.68	0.21	1.87
26	1.21	2.68	0.94	0.72	0.22	1.96
27	1.06	2.87	0.88	0.74	0.14	2.13
28	1.02	2.76	0.94	0.83	0.11	1.92
29	1.06	2.87	0.83	0.70	0.13	2.17
30	1.19	2.69	0.91	0.70	0.21	1.99
31	1.25	2.83	0.87	0.65	0.21	2.18
32	1.25	2.85	1.09	0.81	0.28	2.04
33	1.25	3.14	1.20	0.90	0.29	2.23
34	1.12	3.10	1.32	1.05	0.27	2.05
35	1.10	3.12	1.45	1.18	0.26	1.94
36	1.12	2.94	1.44	1.16	0.29	1.79
37	1.14	2.78	1.27	1.01	0.25	1.77
38	1.17	2.83	1.05	0.83	0.22	2.00
39	1.07	2.91	1.20	1.05	0.16	1.87
40	1.11	2.95	1.37	1.14	0.23	1.80
41	1.08	3.12	1.36	1.14	0.22	1.98
42	1.04	3.24	1.49	1.27	0.22	1.97
43	1.03	3.29	1.54	1.36	0.18	1.94
44	1.04	3.48	1.52	1.31	0.21	2.17
45	1.03	3.57	1.41	1.25	0.16	2.32
46	1.06	3.67	1.38	1.18	0.21	2.49
47	1.04	3.61	1.89	1.61	0.28	1.99
48	1.10	3.40	2.43	1.93	0.51	1.47

Note: S/D = simplification for the ratio of count of active drivers to count of orders generated in each interval; dis = distance. Data for interval 1 was unavailable.

For illustration, we calculate the OTP for interval 9 following the above methodology and settings. The result is depicted in Figure 8.

Table 3. Effectiveness in Myopia Elimination

Interval	Extrema	Distance achieved	Effectiveness	Interval	Extrema	Distance achieved	Effectiveness
(30,120]	2.06	2.09	0.99	(720,810]	0.86	0.90	0.95
(60,150]	2.42	2.45	0.99	(750,840]	0.89	0.92	0.96
(90,180]	2.55	2.60	0.98	(780,870]	0.85	0.88	0.97
(120,210]	2.69	2.73	0.99	(810,900]	0.86	0.89	0.96
(150,240]	2.86	2.94	0.97	(840,930]	0.84	0.87	0.97
(180,270]	2.92	3.01	0.97	(870,960]	0.93	0.95	0.97
(210,300]	2.69	2.92	0.92	(900,990]	1.02	1.05	0.97
(240,330]	2.55	2.78	0.92	(930,1020]	1.17	1.20	0.97
(270,360]	2.52	2.76	0.91	(960,1050]	1.30	1.32	0.98
(300,390]	2.84	3.10	0.92	(990,1080]	1.39	1.40	0.99
(330,420]	3.00	3.14	0.95	(1020,1110]	1.34	1.39	0.97
(360,450]	2.83	2.91	0.97	(1050,1140]	1.22	1.26	0.97
(390,480]	1.96	2.06	0.95	(1080,1170]	1.12	1.17	0.96
(420,510]	1.50	1.55	0.97	(1110,1200]	1.17	1.20	0.97
(450,540]	1.28	1.32	0.97	(1140,1230]	1.29	1.31	0.99
(480,570]	1.24	1.30	0.96	(1170,1260]	1.39	1.41	0.99
(510,600]	1.17	1.23	0.95	(1200,1290]	1.44	1.46	0.98
(540,630]	1.08	1.12	0.96	(1230,1320]	1.50	1.52	0.99
(570,660]	0.87	0.93	0.93	(1260,1350]	1.47	1.49	0.99
(600,690]	0.71	0.81	0.87	(1290,1380]	1.42	1.44	0.98
(630,720]	0.69	0.80	0.86	(1320,1410]	1.50	1.54	0.98
(660,750]	0.77	0.84	0.91	(1350,1440]	1.79	1.82	0.98
(690,780]	0.87	0.91	0.95				

Table 4. Fragment of Dispatching

OrderId	Target grid	Order time	Driver matched	Driver time	Source grid	d_{res}
7c297b1cd7	175 × 214	1420.53	idle1at174	1410.00	174 × 217	−10.53
17b8f89ea7	175 × 218	1416.30	dcd063bd11	1430.61	175 × 219	14.31
8c178f7f03	176 × 212	1413.63	2ff7ebaf10	1426.46	175 × 212	12.83
7676abfc5b	176 × 219	1413.46	f020e07830	1433.41	175 × 219	19.95
42fbb6984f	178 × 214	1421.93	d051cdc0e7	1426.15	177 × 214	4.21
33363ebfc3	179 × 250	1412.33	d4e5ff38d5	1421.86	179 × 250	9.53
24b3d42c60	181 × 213	1418.96	idle1at175	1410.00	173 × 213	−8.96
c626f1af09	181 × 240	1412.33	a87e7dba69	1415.18	147 × 269	2.85
4f9b03af21	182 × 203	1411.81	aa5a7e4958	1420.03	174 × 202	8.21
97d876ad42	182 × 208	1424.41	c8591a3863	1428.15	176 × 209	3.73
56549bdd0a	182 × 231	1419.75	8bd05d52c3	1418.21	181 × 232	−1.53
99d7970cd8	182 × 232	1417.30	d1d125316d	1416.50	182 × 232	−0.80
4192d9d56f	182 × 233	1411.26	idle1at176	1410.00	176 × 237	−1.26
2e70430a29	182 × 235	1410.31	7c82421fee	1417.80	161 × 247	7.48
cb5907bfe9	183 × 240	1419.35	idle1at173	1410.00	173 × 250	−9.35
......						

$\mathrm{Ed}_{\mathrm{pic}}$ *Comparison.* For each interval, we compute the standard W_1 distances (dis1) (through a standard algorithm applicable to balanced optimal transport problems) and the distances by proposed Algorithm 1 (dis2). We compare the results to the distances extracted from actual dispatches by the operating platform (dis0). The strategy underlying dis0 has not been explicitly stated but will probably be value-based (*8*, *9*). Distance is scaled in kilometers if not specified throughout this paper.

We actually start computation from interval 2 because of data missing in interval 1. For the 47 experiments, as reported in Table 2, our proposed methodology only loses twice to the one implemented by operating platform by a slight margin while wins significantly in the other 45. Furthermore, the advantage of Algorithm 1 over standard Wasserstein method is apparent. It shows that the larger the system deviates from balance, the greater the advantage is. In the tables, S/D is a simplification for

Table 5. Ed_{res} Statistics

Interval	Ed_{res} within (in minutes)					Interval	Ed_{res} within (in minutes)				
	$\leqslant 0$ (%)	(0,3] (%)	(3,8] (%)	(8,15] (%)	>15 (%)		$\leqslant 0$ (%)	(0,3] (%)	(3,8] (%)	(8,15] (%)	>15 (%)
2	73	8	9	8	2	26	94	2	2	2	1
3	92	3	2	2	1	27	91	3	3	2	1
4	92	3	3	2	1	28	87	5	5	2	1
5	94	2	2	2	1	29	89	4	4	2	1
6	92	2	3	3	1	30	90	3	3	2	1
7	92	3	2	3	1	31	92	3	3	2	1
8	92	3	3	2	0	32	92	3	2	2	1
9	90	3	3	3	0	33	92	3	3	2	1
10	90	2	4	3	2	34	91	3	4	2	1
11	91	2	3	2	1	35	90	3	3	2	1
12	86	2	6	4	2	36	88	5	4	3	1
13	85	3	7	3	2	37	86	5	5	3	1
14	82	4	6	6	2	38	89	4	4	2	1
15	80	7	6	5	2	39	89	4	4	2	1
16	74	7	10	7	3	40	88	4	4	2	1
17	80	6	7	5	2	41	90	4	4	2	1
18	84	5	6	4	1	42	88	4	4	3	1
19	86	4	5	3	1	43	88	4	4	2	1
20	85	5	5	4	2	44	88	4	3	3	1
21	91	3	3	2	1	45	88	4	4	3	1
22	93	2	2	2	1	46	87	5	5	3	1
23	93	2	3	2	1	47	86	4	5	3	1
24	93	2	2	2	1	48	82	6	6	4	2
25	93	2	2	2	1						

the ratio of count of active drivers to count of orders generated in each interval.

Myopia Control. To verify that myopia is nearly eliminated during the successive 30-min intervals, we construct an extreme benchmark. Our idea is to widen the window to its practical extreme to contain as many candidate dispatches as possible. The extreme window is set 90 min long. For within a 90-min interval, if a driver is not dispatched throughout he will almost certainly log off; meanwhile, there will be a high probability of a passenger canceling the order if it is not responded to within less than 10 min. So the OTP for a 90-min interval is extremely considerate and the corresponding W_1 distance is an extremum to achieve for all dispatching strategies.

Precisely, we construct overlapped 90-min intervals, that is, $(30, 120]$, $(60, 150]$, ...$(1350, 1410]$. For each such interval, we compute the W_1 distance between supply and demand as the *extrema distance* with which dis2 computed previously is to compare. To have a common basis for comparison, we average dis2 over three successive 30-min intervals, weighting with count of orders of each interval, as *distance achieved*. The ratio of extrema distance, which is from the 90-min interval covering the three, to distance achieved is then a representative of effectiveness in myopia elimination. Table 3 shows the results validating Theorem 3.

Order Response Time Ed_{res}. The driver to order level dispatches can be generated by applying Algorithm 3 on the results of Algorithm 2. Table 4 includes a fragment of dispatching for illustration, wherein a negative value of d_{res} indicates that the appointed driver has already being vacant for that long. Recall that when we constructed the distributions for a interval, we just omitted time differences of events, but in a real world delayed response to an order will lead to serious serving complaints. Ed_{res} for the 47 experiments under our strategy are listed in Table 5. The results provide a validating example for (*ii*) of Theorem 1.

Conclusion and Discussion

We observe that the ride-hailing market is a dynamically balanced system. Encouraged by this observation, we aim to optimize the system's productivity, instead of ADI and ORR which are ineffective under the balanced condition. The system's productivity is the inverse of the driver cycle by definition. Consequently, optimization is done by minimizing expected order pickup time under appropriate assumptions. We then formulate the minimization work into sequential optimal transport problems. Finally, the solution can be effectively obtained via algorithms developed within this article. We also demonstrate that myopia is well controlled under our methodology.

For the convenience of statement of our methodology, we make a simplification that each driver is assumed to serve only once during one interval, while actually there are quite a few short orders which permit second serve or even more. This suggests that distribution of supply changes even in one single interval, which just erodes the foundation of our methodology. More sophisticated techniques, such as Wasserstein distance between spatio-temporal distributions, are probably needed to resolve this question.

Acknowledgments

The authors thank the editor and referees for their insightful suggestions, which strengthened the work immensely. The authors also thank Didi Chuxing incorporation for providing informative operational data and D. Schuhmacher et al. for their powerful R package to compute the classical OTPs and standard Wasserstein distances.

Author Contributions

The authors confirm contribution to the paper as follows: study conception and design: D. Lei, Y. Wu; data collection: D. Lei; analysis and interpretation of results: D. Lei, Y. Wu; draft manuscript preparation: D. Lei, Y. Wu. All authors reviewed the results and approved the final version of the manuscript.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Natural Science Foundation of China (12071483).

ORCID iD

Yuanshan Wu https://orcid.org/0000-0002-2121-8952

Data Accessibility Statement

All data used in this article is accessible from Didi Chuxing incorporation at https://gaia.didichuxing.com.

References

1. Wang, H., and H. Yang. Ridesourcing Systems: A Framework and Review. *Transportation Research Part B: Methodological*, Vol. 129, 2019, pp. 122–155.
2. Lee, C. C. *GPS Taxi Dispatch System Based on A* Shortest Path Algorithm*. PhD dissertation. Malausia University of Science and Technology, Kuala Lumpur, 2005.
3. Lee, D. H., H. Wang, R. Cheu, and S. Teo. Taxi Dispatch System Based on Current Demands and Real-Time Traffic Conditions. *Transportation Research Record: Journal of the Transportation Research Board*, 2004. 1882: 193–200.
4. Seow, K. T., N. H. Dang, and D. H. Lee. A Collaborative Multiagent Taxi-Dispatch System. *IEEE Transactions on Automation Science and Engineering*, Vol. 7, 2010, pp. 607–616.
5. Wong, K. I., and M. G. Bell. The Optimal Dispatching of Taxis under Congestion: A Rolling Horizon Approach. *Joual of Advanced Transportation*, Vol. 40, No. 2, 2006, pp. 203–220.
6. Wang, X., N. Agatz, and A. Erera. Stable Matching for Dynamic Ride-Sharing Systems. *Transportation Science*, Vol. 52, No. 4, 2017, pp. 850–867.
7. Bertsimas, D., P. Jaillet, and S. Martin. Online Vehicle Routing: The Edge of Optimization in Large-Scale Applications. *Operation Research*, Vol. 67, No. 1, 2019, pp. 143–162.
8. Xu, Z., Z. X. Li, Q. W. Guan, D. S. Zhang, Q. Li, J. X. Nan, C. Y. Liu, W. Bian, and J. P. Ye. Large-Scale Order Dispatch in On-Demand Ride-Hailing Platforms: A Learning and Planning Approach. *Proc., 24th ACM International Conference on Knowledge Discovery & Data Mining*, London, Association for Computing Machinery, New York, 2018, pp. 905–913.
9. Zhou, M., J. R. Jin, W. N. Zhang, Z. W. Qin, Y. Jiao, C. X. Wang, G. B. Wu, Y. Yu, and J. P. Ye. Multi-Agent Reinforcement Learning for Order-Dispatching via Order-Vehicle Distribution Matching. *Proc., 28th ACM International Conference on Information and Knowledge Management*, Beijing, China, Association for Computing Machinery, New York, 2019, pp. 2645–2653.
10. Akbarpour, M., S. Li, and S. Oveis Gharan. Thickness and Information in Dynamic Matching Markets. *Journal of Political Economy*, Vol. 128, No. 3, 2020, pp. 783–815.
11. Dickerson, J. P., K. A. Sankararaman, A. Srinivasan, and P. Xu. Allocation Problems in Ride-Sharing Platforms: Online Matching with Offline Reusable Resources. *Proc., 32nd AAAI Conference on Artificial Intelligence*, New Orleans, LA, 2018.
12. Achuthan, N. R., and L. Caccetta. Integer Linear Programming Formulation for a Vehicle Routing Problem. *European Journal of Operational Research*, Vol. 52, No. 1, 1991, pp. 86–89.
13. Zhu, L., Z. Zhao, and G. Wu. Shared Automated Mobility with Demand-Side Cooperation: A Proof-of-Concept Microsimulation Study. *Sustainability*, Vol. 13, No. 5, 2021, p. 2483. https://doi.org/10.3390/su13052483.
14. Kulkarni, R. V., and P. R. Bhave. Integer Programming Formulations of Vehicle Routing Problems. *European Journal of Operational Research*, Vol. 20, 1985, pp. 58–67.
15. Kantorovich, L. V. On the Translocation of Masses. *Doklady Akademii Nauk SSSR*, Vol. 37, 1942, pp. 227–229.
16. Cuturi, M. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. *Advances in Neural Information Processing Systems*, Vol. 26, 2013, pp. 2292–2300.
17. Altschuler, J., J. Weed, and P. Rigollet. Near-Linear Time Approximation Algorithms for Optimal Transport via

Sinkhorn Iteration. *Advances in Neural Information Processing Systems*, 2017, pp. 1964–1974. https://nyuscholars.nyu.edu/en/publications/near-linear-time-approximation-algorithms-for-optimal-transport-v.

18. Peyré, G., and M. Cuturi. Computational Optimal Transport: With Applications to Data Science. *Foundations and Trends® in Machine Learning*, Vol. 11, 2019, pp. 355–607.
19. Villani, C. *Topics in Optimal Transportation*, Vol. 58. American Mathematical Society, Providence, RI, 2003.
20. Kantorovitch, L. V. On the Translocation of Masses. *Management Science*, Vol. 5, 1958, pp. 1–4.
21. Peyré, G., and M. Cuturi. Computational Optimal Transport. *arXiv Preprint arXiv:1803.00567*, 2018.
22. Cuturi, M. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. *Advances in Neural Information Processing Systems*, Vol. 26, 2013, pp. 2292–2300.
23. Caffarelli, L., and R. McCann. Free Boundaries in Optimal Transport and Monge-Ampere Obstacle Problems. *Annals of Mathmatics*, Vol. 171, 2010, pp. 673–730.
24. Chizat, L., G. Peyré, B. Schmitzer, and F. X. Vialard. Unbalanced Optimal Transport: Geometry and Kantorovich Formulation. *arXiv Preprint arXiv:1508.05216*, 2015.
25. Chizat, L., G. Peyré, B. Schmitzer, and F. X. Vialard. An Interpolating Distance Between Optimal Transport and Fischer-Rao Metrics. *Foundations of Computational Mathematics*, Vol. 18, No. 1, 2018, pp. 1–44.
26. Figalli, A. The Optimal Partial Transport Problem. *Archive for Rational Mechanics and Analysis*, Vol. 195, 2010, pp. 533–560.
27. Takagi, H. (ed.). Queueing Analysis: A Foundation of Performance Evaluation. In *Part 1: Vacation and Priority Systems*, Vol. 1, Elsevier Science Publishers B.V., Amsterdam, The Netherlands, 1991.