Optimizing Efficiency of Taxi Systems: Scaling-up and Handling Arbitrary Constraints

Jiarui Gan
University of Chinese Academy of Sciences
The Key Laboratory of Intelligent Information Processing, ICT, CAS, Beijing 100190, China
ganjr@ics.ict.cn

Bo An, Chunyan Miao
School of Computer Engineering
Nanyang Technological University
Singapore 639798
{boan,ascymiao}@ntu.edu.sg

ABSTRACT

Taxi service is an indispensable part of public transport in modern cities. However, due to its decentralized operation mode, taxi services in many cities are inefficient. Besides, the decentralized nature also poses significant challenges to analyzing and regulating taxi services. State of the art computational methods for optimizing taxi market efficiency suffer from two important limitations: 1) they cannot be scaled up efficiently; and 2) they cannot address complex real-world market situations where additional scheduling constraints need to be handled. In this paper, we propose two novel algorithms—FLORA and FLORA-A—to address the inadequacies. Using convex polytope representation techniques, FLORA provides a fully compact representation for taxi drivers' strategy space and scales up more efficiently than existing algorithms. FLORA-A avoids enumerating the entire exponentially large pure strategy space by gradually expanding the strategy space. It is the first known method capable of handling arbitrary scheduling constraints for optimizing taxi system efficiency. Experimental results show orders of magnitude improvement in speed FLORA provides, and the necessity of using FLORA-A as suggested by changes in the taxi drivers' operation strategy under different market conditions.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Intelligent agents, Multiagent systems

Keywords

Taxi System; Game Theory; Optimization

1. INTRODUCTION

Taxi service has long been an indispensable part of public transport in modern cities. However, taxi systems are quite inefficient and are very difficult to analyze and regulate since they are highly decentralized and are operated by profit-driven drivers (unlike other types of public transport, such as bus and metro systems, operation schedules and movements of taxis can hardly be controlled by regulators). Besides, taxi service can be affected by many factors such as road condition, customer demand, fare price, etc. These factors not only depend on each other in complex ways, but also vary with time. Analyzing, regulating, and optimizing taxi systems are thus important but challenging problems, which have attracted many research interests over the past decades [9, 3, 4, 7]. In this paper, we study the taxi system efficiency optimization problem (through adjusting taxi fare price) with a focus on the computational challenges.

Figure 1: People waiting for taxis during rush hours in Beijing (left) and Chongqing (right) in China.

There has been some existing work concerning taxi system efficiency optimization and taxi pricing. For instance, Schaller studied the effects of fare price increase on taxi market in New York [17]. Kim and Hwang studied an incremental discount policy on the taxi fare with the objective of maximizing average profit of taxis [15]. Yang et al. examined a nonlinear fare structure and showed its advantages over the existing linear fare structure in Hong Kong [21]. These work focused on modeling the interdependencies among market factors, but ignored taxi drivers' profit-driven behaviors and their competitive relations, which are the key to the decentralized nature and the low efficiency of taxi markets. Recent successful applications of AI techniques in the transportation domain (i.e., AI-based traffic control and intersection management [10, 2, 16, 20], and decision support for taxi drivers [19]) have indicated a promising way to analyze the intelligent human behaviors and to address the limitation of above work. Indeed, Gan et al. have proposed a game theoretic approach to address a taxi fare price optimization problem [13, 12]. A time-varying taxi market model was proposed, and taxi drivers' profit-driven and competing behaviors were considered in the model. This work has opened a new door to AI-based taxi market studies.

Unfortunately, the current state of the art algorithms proposed by Gan et al. (e.g., the *atom schedule method* [13]) for computing optimal taxi fare structure are inadequate for dealing with real-world taxi system efficiency optimization problems. Firstly, the algorithm scales up poorly with the

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*

number of periods (in order to model a time-varying taxi market, the optimization horizon is discretized into a set of periods, and fine-grained discretization indicates more periods). Secondly, their work only considered taxi drivers' scheduling constraints on maximum total/continuous working time, which is insufficient for some specific cases. For example, in some cities, taxi drivers are not allowed to switch from working to resting during peak hours, and the regulation is effectively implemented with heavy fines and intensive supervision. The effects of the market regulations on taxi drivers' decision-making cannot be ignored, and new algorithms are needed to compute solution under these effects.

In this paper, we propose two novel algorithms to address the limitations of existing work. The first algorithm, FLORA, provides a *fully* compact representation for the complex optimization problem associated with taxi drivers' decision-making process. It utilizes convex polytope representation conversion techniques and reduces the number of variables and constraints to $O(n)$ in the number n of periods. Our second algorithm, FLORA-A, is the first known method for solving taxi system optimization problems with *arbitrary* scheduling constraints. FLORA-A takes advantage of the small support size of the taxi drivers' mixed strategies and the concavity and differentiability of the taxi drivers' utility function. It avoids enumerating the entire exponentially large pure strategy space by gradually expending the strategy space with useful pure strategies. Using FLORA-A, we are able to consider additional constraints existing algorithms fail to handle, such as disabling state switching during rush hours, and setting the minimum continuous working/resting time to eliminate unrealistic schedules where the drivers switch states in a very frequent manner. Experimental results show that 1) FLORA provides orders of magnitude improvement in speed over existing algorithms; and 2) significant changes in the market caused by additional scheduling constraints, which suggests the necessity of using FLORA-A to handle additional constraints when they indeed exist.

2. TAXI SYSTEM EFFICIENCY OPTIMIZATION PROBLEM

The goal of the Taxi system Efficiency optiMization Problem (**TEMP**) is to maximize the *efficiency* of a taxi system through adjusting the *taxi fare*. To this end, the first step is to know how system efficiency is affected by fare price in a taxi market. Results from transportation researches [22, 23] suggest that a taxi market is determined by two key factors—the *fare price* and the *number of working taxis* (i.e., taxis in service, either vacant or occupied). Furthermore, when the taxi drivers' profit-driven behavior is taken into consideration, it turns out that the number of taxis in service also relies on the fare price, because the taxi drivers always adjust their strategies to fit the given fare price, and keep their profits maximized. In this section, we first present a taxi market model based on transportation science researches. Then we model the taxi drivers' profit-driven behavior, and formulate TEMP.

2.1 A Multi-period Taxi Market Model

A taxi market is a dynamic time-varying system. Consider the peak time and the non-peak time of a day, there are huge variances in factors such as the number of potential

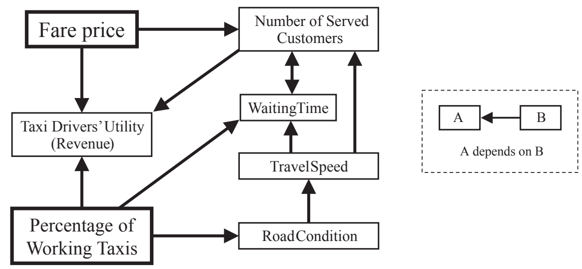


Figure 2: Complex interdependencies among factors in the taxi market.

customers and the travel speed on road. To deal with such variances, we discretize the optimization horizon T (e.g., a whole day) into a set of n equal-length periods, such that when the duration of each period is sufficiently short, the market can be treated as static in each period. We adopt a static taxi market model to each period i. The model involves many factors, and the factors depend on each other in very complex ways (see Figure 2). We start with the number of served customers. The number of customers served by the whole taxi system is determined by average monetary and time cost of a trip [22], i.e.,

$$D^i(F^i, L^i, W^i) = \bar{D}^i \exp\{-\beta(\frac{F^i}{\gamma} + \phi_1 L^i + \phi_2 W^i)\}, \quad (1)$$

where F^i is the average fare price, L^i is the average travel time, and W^i is the average customer waiting time; $\beta > 0$ is a sensitivity parameter; ϕ_1 and ϕ_2 are parameters used for converting time costs into monetary costs; γ is the average number of passengers per ride; $\bar{D}^i$ is the number of potential customers, which is obtained when the total cost is zero. The waiting time W^i is negatively related to the number of vacant taxis available, and in turn depends on D^i as

$$W^i(D^i, L^i, p^i) = \frac{\omega}{p^i \cdot N_T - \frac{D^i L^i}{\gamma \cdot \tau}}, \quad (2)$$

where $\omega > 0$ is a parameter depending on the density of taxi stands; p^i is the Percentage of Working taxis (**PoW**), and given a fixed number N_T of total licensed taxis, $p^i \cdot N_T - D^i L^i/(\gamma \cdot \tau)$ represents vacant taxis available in period i. It can be proven that when F^i, L^i and p^i are fixed, D^i and W^i are uniquely determined by Eqs. (1) and (2) [23]. Therefore, D^i and W^i are in fact implicit functions of F^i, L^i and p^i. We can thus denote them as $D^i = D^i(F^i, L^i, p^i)$ and $W^i = W^i(F^i, L^i, p^i)$.

Given the average trip distance d^i, the travel time is then determined by travel speed V^i as $L^i = d^i/V^i$. Travel speed in a road network can be approximated by a linear function of number of on-road vehicles [18], which is furthermore linear to PoW p^i as we assume that the number N_v^i of non-taxi vehicles on the road is a period-specified constant. Thus, V^i is a linear function of p^i, i.e.,

$$V^i(p^i) = \mu(p^i \cdot N_T + N_v^i) + \lambda,$$

where μ and λ are parameters depending on the road condition. Similarly, we denote L^i, D^i, and W^i as $L^i{=}L^i(p^i)$, $D^i{=}D^i(F^i, p^i)$, and $W^i{=}W^i(F^i, p^i)$, respectively.

We adopt a *distance-based* fare structure (the model can also be extended to other fare structures):

$$F^i = f_0 + f^i \cdot (d^i - d_0),$$

where f_0 is the initial charge and d_0 is the distance covered by f_0; f^i is the charge rate for period i, i.e., the per unit distance charge. We optimize the fare structure through adjusting the charge rate f^i, and thus treat F^i as a function $F^i(f^i)$. Accordingly, all the market factors, particularly the number D^i of served customers, now depend on f^i and p^i, i.e., $D^i = D^i(f^i, p^i)$. For ease of description, we denote a market factor over T as a column vector with each component corresponding to a period. For example, we denote charge rate over all periods as $\mathbf{f} = \langle f^i \rangle$.

2.2 Modeling the Taxi drivers' Strategy

From a game theoretic perspective, the taxi drivers have a set $\mathcal{S}$ of *pure strategies*, and they adopt a *mixed strategy* (*strategy* for simplicity), which is a probability distribution $\mathbf{x}$ over their pure strategies in $\mathcal{S}$. Each pure strategy is a schedule that specifies working and resting periods. Formally, a pure strategy can be denoted as a 0/1 vector $\mathbf{s} \in \{0,1\}^n$, where s^i=1 (resp. s^i=0) represents working (resp. not working) in period i. Each pure strategy has to satisfy some constraints to ensure its feasibility in practice. For example:

- **Constraint 1 (C1):** A taxi driver cannot work for more than $\hat{n}_w$ periods in any schedule in $\mathcal{S}$.
- **Constraint 2 (C2):** A taxi driver cannot work *continuously* for more than $\hat{n}_c$ periods in any schedule in $\mathcal{S}$.

As we will show later, more realistic constraints might be needed in specific scenarios, and different algorithms are developed to cope with different constraint settings. We assume that the taxi drivers adopt the same mixed strategy. Thus when they choose strategy $\mathbf{x}$, PoW is determined as:

$$\mathbf{p}(\mathbf{x}) = \sum\nolimits_{\mathbf{s} \in \mathcal{S}} x_{\mathbf{s}} \cdot \mathbf{s}. \tag{3}$$

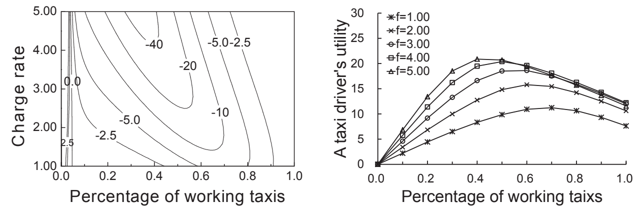
Furthermore, since taxi drivers are profit-driven, they always choose the best strategy which maximizes their utility, i.e.,

$$\mathbf{x}^* \in \arg \max_{\mathbf{x}:\, \mathbf{x} \geq \mathbf{0},\, \mathbf{1}^{\mathsf{T}}\mathbf{x}=1} U\big(\mathbf{f}, \mathbf{p}(\mathbf{x})\big). \tag{4}$$

The utility function $U(\mathbf{f}, \mathbf{p})$ is defined as the sum of utilities in all periods, i.e., $U = \sum_i U^i$, and U^i is defined as

$$U^i(f^i, p^i) = \frac{D^i}{\gamma \cdot N_T} \cdot F^i - p^i \cdot c_f \cdot \tau,$$

where $D^i \cdot F^i/(\gamma \cdot N_T)$ represents the income as $D^i/(\gamma \cdot N_T)$ is the average number of trips each taxi serves, and c_f is the cost of gasoline per unit time. In such a way, the fare price determines the taxi drivers' strategy via the optimization in Eq. (4), and taxi drivers' strategy in turn determines PoW via Eq. (3). Particularly, as shown in Figure 3(a), for a realistic range of charge rates, the second-order partial derivative $\frac{\partial^2 U^i}{\partial {p^i}^2} < 0$ only except for a very small part where PoW is very close to 0. We exclude this part of the function since it occurs when U^i is very close to 0 and has negligible affect on the overall shape of the curve of U^i (Figure 3(b)). Therefore, for a given charge rate f^i, U^i can be treated as a *concave* function with respect to p^i; and U is concave with respect to $\boldsymbol{p}$ since the Hessian matrix $\nabla^2 U = \mathrm{diag}(\frac{\partial^2 U^1}{\partial (p^1)^2}, \ldots, \frac{\partial^2 U^n}{\partial (p^n)^2})$ is positive definite. Given a convex feasible set of $\mathbf{p}$ there is only one $\mathbf{p}$ that maximizes U [6]. This means even if there are different solutions to Eq. (4), the solutions must all yield the same PoW, such that a one-to-one correspondence from $\mathbf{f}$ to $\mathbf{p}$ is guaranteed.



(a) Contours of $\frac{\partial^2 U^i}{\partial {p^i}^2}$ (b) Strict concaveness of U^i

Figure 3: Properties of taxi drivers' utility function

2.3 A Bilevel Optimization Problem

We measure the system efficiency with the total number of served customers, i.e., $D(\mathbf{f}, \mathbf{p}) = \sum_i D^i(f^i, p^i)$, and formulate a TEMP as the following bilevel optimization program.

$$\max_{\mathbf{f}, \mathbf{x}} \quad D\big(\mathbf{f}, \mathbf{p}(\mathbf{x}^*)\big) \tag{5}$$

$$\text{s.t.} \quad \mathbf{x}^* = \arg \max_{\mathbf{x}:\, \mathbf{x} \geq \mathbf{0},\, \mathbf{1}^{\mathsf{T}}\mathbf{x}=1} U\big(\mathbf{f}, \mathbf{p}(\mathbf{x})\big) \tag{6}$$

The model can also handle other measures of the system efficiency with the same form of optimization program as long as the optimization objective is a function of $\mathbf{f}$ and $\mathbf{p}$. To solve this bilevel optimization problem, a practical way is to discretize the continuous fare price space into a small set of candidate prices, e.g. $\{¥1.00, ¥1.20, \ldots, ¥5.00\}^n$, and solve the lower level program (Eq. (6)) under each of the candidate prices to find the optimal fare price. Thus the key is to compute the lower level program. Unfortunately, the lower level program suffers from a scalability issue caused by the exponential explosion of the taxi drivers' pure strategy space. Next, we present our algorithms to deal with the scalability issue under different circumstances.

3. ALGORITHM WITH FULLY COMPACT REPRESENTATION

In this section, we present our first algorithm FLORA (FuLly cOmpact RepresentAtion). FLORA deals with the scalability issue under constraints **C1** and **C2**. It reduces the scale of the lower level program to $O(n)$ and avoids enumerating any schedule by utilizing a *fully* compact representation of the taxi drivers' strategy space, which is in contrast with the partially compact representation offered by the state-of-the-art algorithm [13].

3.1 FLORA

According to the bilevel program in Eqs. (5)–(6), PoW acts a bridge linking the taxi drivers' strategy and the rest of the system. FLORA uses PoW to compactly represent the taxi drivers' strategy and reformulate the bilevel program as:

$$\max_{\mathbf{f}, \mathbf{p}} \quad D(\mathbf{f}, \mathbf{p}^*) \tag{7}$$

$$\text{s.t.} \quad \mathbf{p}^* = \arg \max_{\mathbf{p} \in \mathcal{P}} U(\mathbf{f}, \mathbf{p}) \tag{8}$$

Thus the number of variables of the lower level program is reduced to n. The major challenge of this approach is to guarantee that the PoW obtained can always be implemented by a feasible mixed strategy of the taxi drivers, namely, to define the feasible set $\mathcal{P}$ of PoW. In an implicit way, $\mathcal{P}$ can be defined as:

$$\mathcal{P} = \left\{ \mathbf{p} \in \mathbb{R}^n \;\middle|\; \begin{array}{l} \mathbf{p} = \sum_{\mathbf{s}\in\mathcal{S}} x_{\mathbf{s}}, \\ \mathbf{0} \le \mathbf{x} \le \mathbf{1} \text{ and } \mathbf{1}^{\mathsf{T}}\mathbf{x} = 1 \end{array} \right\}, \tag{9}$$

which is a convex polytope of $\mathcal{S}$ as a set of n-dimensional points. A convex polytope can be defined in two ways: 1) **V**ertex Representation (V-rep), as the convex hull of a set of points (the set of points contains at least all the vertices of the convex hull); 2) **H**alf-space Representation (H-rep), as the intersection of the set of its facet-inducing half-spaces [14]. Eq. (9) is the V-rep of $\mathcal{P}$. When $\mathcal{S}$ satisfies **C1** and **C2**, the H-rep of $\mathcal{P}$ is the following inequality set, where each inequality defines a half-space.

$$\left(\begin{array}{ccccc} & & I_n & & \\ \hdashline & & -I_n & & \\ \hdashline 1 & & \cdots & & 1 \\ \hdashline 1 & \cdots & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & \multicolumn{3}{c}{\underbrace{1 \quad \cdots \quad 1}_{\hat{n}_c+1}} \end{array}\right) \mathbf{p} \le \left(\begin{array}{c} 1 \\ \vdots \\ 1 \\ \hdashline 0 \\ \vdots \\ 0 \\ \hdashline \hat{n}_w \\ \hdashline \hat{n}_c \\ \vdots \\ \hat{n}_c \end{array}\right). \tag{10}$$

Let the convex polytope defined by the above H-rep be $\mathcal{P}_{\mathrm{H}}$. $\mathcal{P}_{\mathrm{H}}$ is equivalent to $\mathcal{P}$ because:

1. $\mathcal{P} \subseteq \mathcal{P}_{\mathrm{H}}$, since all the vertices of $\mathcal{P}$, i.e., points in $\mathcal{S}$, satisfy Eq. (10). Specifically, the inequality set is divided into four parts by the dashed lines. The first two parts, i.e., $\mathbf{0} \le \mathbf{p} \le \mathbf{1}$, are obviously satisfied by all 0/1 vectors. The third and the fourth parts are equivalent to **C1** and **C2** respectively, and are thus satisfied for all points in $\mathcal{S}$.

2. $\mathcal{P}_{\mathrm{H}} \subseteq \mathcal{P}$, since it can be proven that all the vertices of $\mathcal{P}_{\mathrm{H}}$ are 0/1 vectors (Proposition 1), and they are all in $\mathcal{S}$ since $\mathcal{S}$ contains all 0/1 vectors satisfying **C1** and **C2**.

Thus using the H-rep (written as $\mathbf{A}\mathbf{p}\le\mathbf{b}$ for simplicity), the lower level program is compactly represented as

$$\max_{\mathbf{p}} \quad U(\mathbf{f}, \mathbf{p}) \tag{11}$$

$$\text{s.t.} \quad \mathbf{A}\mathbf{p} \le \mathbf{b} \tag{12}$$

which only has n variables and less than $3n$ constraints.

Proposition 1. *Let $\mathcal{P}_{\mathrm{H}}$ be the convex polytope defined by Eq.* (10). *All the vertices of $\mathcal{P}_{\mathrm{H}}$ are 0/1 vectors.*

Proof. Let $\mathbf{v} = (v_1, \dots, v_n)^{\mathsf{T}}$ be a vertex of $\mathbf{P}_{\mathrm{H}}$. We show that $\mathbf{v} \in \{0,1\}^n$. A vertex of a polytope in an n-dimensional space is in n of the polytope's facets, and is the only solution to an equality system with n equalities each corresponding to a facet. For $\mathbf{v}$, each of the equalities is derived from a row of Eq. (10) by setting inequality to equality. We write the equality system as

$$H'\mathbf{v} = \mathbf{b}, \tag{13}$$

and the augmented form as

$$\widetilde{H}' = \left(\begin{array}{c|c} H' & \mathbf{b} \end{array} \right). \tag{14}$$

Since H' is part of the matrix in the left of Eq. (10), it has the following properties:

a) it contains only 0 and 1 as its elements;

b) all the 1s in each row are consecutive.

According to these properties, for the i^{th} row $\widetilde{\mathbf{h}}'_i = [\mathbf{h}'_i \,|\, b_i]$ of $\widetilde{H}'$, we define $L(\mathbf{h}'_i) = \min\{j \,|\, h'_{ij} = 1,\, j \in \{1,\dots,n\}\}$, where h'_{ij} is the j^{th} element of $\mathbf{h}'_i$; and $R(\mathbf{h}'_i) = \max\{j \,|\, h'_{ij} = 1,\, j \in \{1,\dots,n\}\}$. Namely, $L(\mathbf{h}'_i)$ and $R(\mathbf{h}'_i)$ are the indices of the leftmost and rightmost 1 in $\mathbf{h}'_i$, respectively.

Algorithm 1: Row reduction

1 **for** $k = 1$ *to* n **do**
2 $\quad \mathcal{L}_k \leftarrow \{i \,|\, L(\mathbf{h}'_i) = k,\, i \in \{1,\dots,n\}\}$;
3 **while** *some* $\mathcal{L}_k$ *contains more than one element* **do**
4 $\quad i_0 \leftarrow \arg\min_i\{R_i \,|\, i \in \{1,\dots,n\}\}$;
5 $\quad$ **for** *each* $i \in \mathcal{L}_k \setminus \{i_0\}$ **do**
6 $\quad\quad \widetilde{\mathbf{h}}'_i \leftarrow \widetilde{\mathbf{h}}'_i - \widetilde{\mathbf{h}}'_{i_0}$;
7 $\quad\quad \mathcal{L}_k \leftarrow \mathcal{L}_k \setminus \{i\}$;
8 $\quad\quad l \leftarrow L(\mathbf{h}'_i)$;
9 $\quad\quad \mathcal{L}_l \leftarrow \mathcal{L}_l \cup \{i\}$;
10 Rearrange the rows of $\widetilde{H}'$ by the index of the leftmost 1 in ascending order;
11 **for** $i = n-1$ *to* 1 **do**
12 $\quad$ **for** $j = n$ *to* 1 **do**
13 $\quad\quad \widetilde{\mathbf{h}}'_i \leftarrow \widetilde{\mathbf{h}}'_i - \widetilde{h}'_{ij}\widetilde{\mathbf{h}}'_j$;

We perform row reduction on $\widetilde{H}'$ in the way described by Algorithm 1. Throughout the process, $\widetilde{H}'$ is transformed into the patterns shown in Eq. (15), where entries marked as $*$ are integers. The first step (Lines 1 to 10) is explained as follows:

1. Throughout Lines 1 to 10, $\widetilde{H}'$ always maintains its two properties. This is because the only operation that might break the properties is the subtraction in Line 6, but they always hold since $L(\mathbf{h}'_i) = L(\mathbf{h}'_{i_0})$ and $R(\mathbf{h}'_i) \ge R(\mathbf{h}'_{i_0})$.

2. Line 3 indicates that each $\mathcal{L}_k$ contains less than one element. Moreover, if some $\mathcal{L}_k$ contains no element, then the rank of $\widetilde{H}'$ will be smaller than n, which contradicts the fact that Eq. (13) has only one solution. Therefore, each $\mathcal{L}_k$ contains exactly one row at the end of the while-loop, i.e., there is one and exactly one row in H' where the leftmost 1 has index k, for all $k = 1, \dots, n$.

$$
\begin{aligned}
\widetilde{H}' &\xrightarrow{\text{Lines 1 to 10,}} \left(\begin{array}{cccc:c} 1 & * & \cdots & * & * \\ & \ddots & \ddots & \vdots & \vdots \\ & & 1 & * & * \\ & & & 1 & * \end{array}\right) \\
&\xrightarrow{\text{Lines 11 to 13,}} \left(\begin{array}{cccc:c} 1 & & & & * \\ & 1 & & & * \\ & & \ddots & & \vdots \\ & & & 1 & * \end{array}\right)
\end{aligned}
\tag{15}
$$

Lines 11 to 13 further transform $\widetilde{H}'$ into the final pattern. All the entries in the rightmost column remain to be integers, because all the operands in Line 13 are integers. Therefore, the solution to Eq. (13) are integers. We have $\mathbf{v} \in \mathbb{Z}^n$. Since the first two parts of Eq. (10) require that $\mathbf{0} \le \mathbf{v} \le \mathbf{1}$, we conclude that $\mathbf{v} \in \{0,1\}^n$. $\square$

4. ARBITRARY CONSTRAINTS

To enhance the practical effectiveness of the optimized fare, we need to build a realistic model of the taxi system and one important modeling component is the scheduling constraints of taxi drivers. FLORA and existing algorithms are all designed for TEMPs with scheduling constraints **C1** and **C2**. In some other cases, it is also necessary to consider other types of scheduling constraints. For example, when the model is more fine-grained, a period has a shorter duration (e.g., half an hour), and it is thus unrealistic that a taxi driver rests or works for only one period each time as it takes time and efforts to find parking when they switch. Therefore, the following constraints need to be considered.

- **C3:** In all schedules in $\mathcal{S}$, if a taxi driver switch to work, she should work continuously for at least $\check{n}_c$ periods.
- **C4:** A taxi driver should rest for at least $\check{n}_r$ periods before she works again in all schedules in $\mathcal{S}$.

Besides, in a well regulated market, it is essential to take into account the impact of market regulations. For example, in some cities in China, such as Chongqing, Fuzhou, Hefei, and Harbin, taxi drivers are not allowed to switch from working to resting during peak hours. The regulation was introduced because it was frequently reported that large numbers of taxi drivers refuse to take passengers in the name of a daily working switch.[1] Such regulations can be effectively implemented (if not in megacities such as Beijing and Shanghai) with the help of intensive supervision and heavy fines, so that their effects are not negligible. Knowing how taxi drivers react to the regulations is thus essential for optimization. Specially, we consider the following constraint.

- **C5:** A taxi driver cannot switch off during peak time in any schedule in $\mathcal{S}$.

However, the existing atom schedule method cannot handle the above constraints. It is not known whether FLORA can be extended to handle them due to the complexity of polytope representation conversion. Generally, the problem of converting a convex polytope from its V-rep to its H-rep is known as the *facet enumeration problem*. The problem is unpredictable in time complexity and the size of output H-rep. No algorithm that runs in time polynomial in the sizes of input V-rep and output H-rep is known [1]; and an n-dimensional 0/1-polytope may contain as many as $O\big((n-2)!\big)$ hyperplanes [11]. These facts make the approach of FLORA unsuitable for dealing with additional scheduling constraints. Next, we present FLORA-A (FLORA with Arbitrary scheduling constraints) to handle *arbitrary* scheduling constraints, given that the utility function is *differentiable* and *concave*. Here by saying arbitrary scheduling constraints, we mean any constraint that can be seen as a function $g : \{0,1\}^n \mapsto \{0,1\}$ that tells whether every schedule (as an n-dimensional 0/1 vector) is feasible or not.

[1] In these cities, a taxi is usually operated by two drivers working on alternative days. When one's shift ends, the driver needs to hand the taxi to the other driver.

Algorithm 2: FLORA-A

```
S' ← n+1 randomly chosen pure strategies in S;
repeat
    x* ← arg max_{x: a distribution over S'} U(f, p(x));
    p* ← p(x*);
    s ← arg max_{s∈S} (s − p*)^T ∇U(p*);
    S' ← S' ∪ {s};
until (s − p*)^T ∇U(p*) ≤ 0;
```

4.1 FLORA-A

FLORA-A is outlined in Algorithm 2. Recall that calculating the taxi drivers' optimal strategy suffers from the scalability issue due to the exponential explosion of taxi drivers' pure strategy space. In contrast, to support an optimal mixed strategy, the taxi drivers only need a small fraction of pure strategies out of the exponentially large pure strategy set. This can be explained by the Carathéodory's theorem (Theorem 2), as any point in the convex polytope of PoW (Eq. (9)) falls in a convex hull of $n+1$ or fewer points in $\mathcal{S}$, and can thus be 'implemented' by these points as their convex combination. Therefore, the general idea of FLORA-A is to restrict the pure strategy set to a small subset $\mathcal{S}' \subset \mathcal{S}$, e.g., a subset of $n+1$ randomly chosen pure strategies, and calculate taxi drivers' optimal mixed strategy under the restriction. When $\mathcal{S}'$ is very small, the optimal strategy can be simply calculated with the program defined by Eq. (6) (with $\mathbf{x}$ being a probabilistic distribution over pure strategies $\mathcal{S}'$). Since the solution is only optimal against a restricted pure strategy set $\mathcal{S}'$, not being necessarily optimal when the entire pure space $\mathcal{S}$ is considered, FLORA-A furthermore searches for a point $\mathbf{s} \in \mathcal{S}$, such that when $\mathbf{s}$ is added to $\mathcal{S}'$, the solution of the restricted problem can be improved; and FLORA-A repeats the processes of solving the restricted problem and searching for a new pure strategy, until no pure strategy can be added to improve the solution, when the solution obtained must be optimal against the entire strategy space.

The problem thus reduces to how to find a new pure strategy to improve the solution of the restricted problem. To accomplish this, FLORA-A takes advantage of the concavity and differentiability of the objective function. Let $\mathbf{x}^*$ be the optimal mixed strategy of the restricted problem defined over $\mathcal{S}'$. FLORA-A evaluates the gradient of the objective function U at $\mathbf{p}^* = \mathbf{p}(\mathbf{x}^*)$ in the PoW space, and searches for a point $\mathbf{s} \in \mathcal{S}$ such that $(\mathbf{s} - \mathbf{p}^*)^{\mathsf{T}} \nabla U(\mathbf{p}^*) > 0$ (For ease of description, we write $U(\mathbf{f}, \mathbf{p})$ as $U(\mathbf{p})$, since $\mathbf{f}$ is actually fixed in the lower level program). In such a way:

- If a satisfying point $\mathbf{s}$ is found, $\mathcal{S}'$ is expanded to $\mathcal{S}'' = \mathcal{S}' \cup \{\mathbf{s}\}$, and correspondingly, the PoW space is expanded from the convex polytope of $\mathcal{S}'$, say $\mathcal{P}'$, to that of $\mathcal{S}''$, say $\mathcal{P}''$. The optimal value of U over $\mathcal{P}''$ must be strictly larger than that over $\mathcal{P}'$, because the optimality criterion (Theorem 3) does not hold for $\mathbf{p}^*$ at $\mathbf{s} \in \mathcal{P}''$. Therefore, the solution of the restricted problem is improved when the pure strategy set expands from $\mathcal{S}'$ to $\mathcal{S}''$.
- If no satisfying point exists, then $(\mathbf{s} - \mathbf{p}^*)^{\mathsf{T}} \nabla U(\mathbf{p}') \leq 0$ holds for all $\mathbf{s} \in \mathcal{S}$, and in turn for all points in the convex polytope of $\mathcal{S}$, say polytope $\mathcal{P}$. According to the optimality criterion, $\mathbf{p}^*$ must be optimal over $\mathcal{P}$, which means that the solution of the current restricted problem is already optimal against the entire strategy space.

Theorem 2 (Carathéodory's theorem [8]). *If point $\mathbf{p} \in \mathbb{R}^n$ lies in the convex hull of a set $\mathcal{S}$, there is a subset $\mathcal{S}'$ of $\mathcal{S}$ consisting of $n+1$ or fewer points such that $\mathbf{p}$ lies in the convex hull of $\mathcal{S}'$.*

Theorem 3 (Optimality criterion [6]). *Suppose in a concave maximization problem (i.e., a program of maximizing a concave objective function over a convex feasible set) the objective function $f(\mathbf{x})$ is differentiable. Let X be the convex feasible set. Then $\mathbf{x}^*$ is optimal in X if and only if $(\mathbf{x} - \mathbf{x}^*)^{\mathsf{T}} \nabla f(\mathbf{x}^*) \leq 0$ for all $\mathbf{x} \in X$.*

4.2 FLORA-A with Linear Constraints

Notably, when all the scheduling constraints can be expressed as linear constraints, a *mixed-integer programming* (MILP) structured as follows can be employed to facilitate the process of searching for new pure strategies.

$$\max_{\mathbf{s} \in \{0,1\}^n} \quad (\mathbf{s} - \mathbf{p}^*)^{\mathsf{T}} \nabla U(\mathbf{p}^*)$$
$$\text{s.t.} \quad \textit{scheduling constraints as linear constraints}$$

If the optimal solution of the above MILP has a positive objective value, then it can be taken as the new pure strategy; otherwise, we can conclude that no satisfying pure strategy exists. Indeed, constraints **C1**–**C5** can all be expressed as linear constraints:

- **C1 and C2:** As we have shown in Section 3, schedules violating **C1** and **C2** can be eliminated by linear constraints defined in the third and the fourth parts of Eq. (10), respectively.
- **C3:** A schedule where continuous working of $m < \check{n}_c$ periods occurs has the following structure:

$$(\,\dots,\quad 0,\ \underbrace{1,\ 1, \dots, 1,}_{m}\quad 0,\ \dots\,)^{\mathsf{T}}$$

Thus we can define the following linear constraints for all $m = 1, \dots, \check{n}_c - 1$ to eliminate schedules violating **C3**:

$$\begin{pmatrix} 1 & \cdots & 1 & -1 & & & \\ -1 & 1 & \cdots & 1 & -1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & -1 & 1 & \cdots & 1 & -1 \\ & & & -1 & \underbrace{1 \quad \cdots \quad 1}_{m} & & \end{pmatrix} \mathbf{s} < \begin{pmatrix} m \\ m \\ \vdots \\ m \\ m \end{pmatrix}$$

It can be easily verified that a schedule satisfying **C3** satisfies all of the above linear constraints, while one violating **C3** should violates at least one of them. There are $O(n^2)$ such constraints in total.

- **C4:** Similarly, a schedule where a taxi driver rests for $m < \check{n}_r$ periods has the following structure:

$$(\,\dots,\ 1,\quad \underbrace{0,\quad 0, \dots,\quad 0,}_{m}\ 1,\ \dots\,)^{\mathsf{T}}$$

We can define the following constraints for $m = 1, \dots, \check{n}_r - 1$ to eliminate schedules violating **C4**:

$$\begin{pmatrix} 1 & -1 & \cdots & -1 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & \underbrace{-1 \quad \cdots \quad -1}_{m} & & 1 \end{pmatrix} \mathbf{s} < \begin{pmatrix} 2 \\ \vdots \\ 2 \end{pmatrix}$$

There are $O(n^2)$ such constraints in total.

- **C5:** A schedule where taxi drivers switch off during peak time has the following structure:

$$(\,\dots,\ \underbrace{\dots,\ 1,\ 0, \dots,}_{\text{peak periods}}\ \dots\,)^{\mathsf{T}}$$

We can eliminate schedules violating **C5** with the following linear constraints:

$$\begin{pmatrix} & \underbrace{\begin{matrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \end{matrix}}_{\text{Peak periods}} & \end{pmatrix} \mathbf{s} < \begin{pmatrix} 2 \\ \vdots \\ 2 \end{pmatrix}$$

The number of such constraints is less than the number of peak periods.

5. EXPERIMENTAL EVALUATION

Government data on Beijing's transportation [5] are used for the experiments, and are listed in Tables 1 and 2. The data have also been used by existing work [13]. It provides statistics for 18 one-hour-long periods ranging from 5:00 a.m. to 11:00 p.m. We use the non-linear programming solver KNITRO (8.0.0) to solve all the standard form optimization programs; and we use the solver CPLEX (12.0.4) to solve all the MILPs. All results are obtained on a machine with a 3.10GHz quad core CPU and a 4.00GB memory.

5.1 Scalability of FLORA

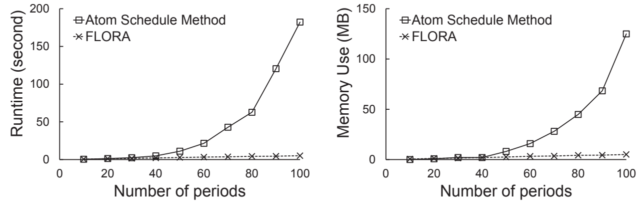
The first set of experiments compare the scalability of FLORA with the existing atom schedule method. Models with up to 100 periods are generated with the 18-period data. For an n-period setting, we fix the horizon T to 18 hours, and fix the maximum total and continuous working time to 9 hours and 5 hours, respectively. We use $\tau = 18/n$ (hour) as the duration for each period, and use $\hat{n}_w = \lceil 9 \times \tau \rceil$ and $\hat{n}_c = \lceil 5 \times \tau \rceil$ for **C1** and **C2**. We use linear interpolation to generate period-dependent parameters from the 18-period raw data. Other period-independent parameters remain the same as the 18-period setting.

Table 1: Period-independent parameters

	Value	Description
N_t	6.66×10^4	Total number of licensed taxis
$\bar{N}$	1.00×10^6	Road network capacity
$\bar{d}$	$7.2\ km$	Average trip distance
τ	$1\ hour$	Duration of a period (in hour)
n	18	Total number of periods
n_w	9	Maximum number of working periods
n_c	5	Maximum number of continuous working periods
V_0	$50\,km/hour$	Free-flow speed
β	0.06	Demand sensitive parameter
ϕ_1	¥$20/hour$	Parameter for converting travel time to monetary cost
ϕ_2	¥$40/hour$	Parameter for converting waiting time to monetary cost
ω	400	Parameter for calculating waiting time
γ	1.5	Average number of passengers per trip
c_f	¥$20/hour$	Fuel cost per hour

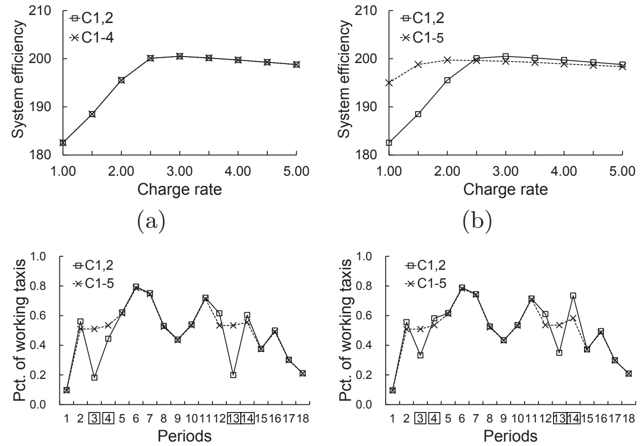
Table 2: Period-dependent parameters

i	$\bar{D}^i$ $(\times10^4)$	N_v^i $(\times10^4)$	i	$\bar{D}^i$ $(\times10^4)$	N_v^i $(\times10^4)$	i	$\bar{D}^i$ $(\times10^4)$	N_v^i $(\times10^4)$
1	8.99	8.53	7	64.70	59.53	13	159.44	81.53
2	49.43	54.64	8	49.27	47.14	14	112.40	75.95
3	151.93	81.75	9	41.45	43.13	15	34.97	44.31
4	106.87	78.14	10	50.46	46.38	16	45.96	47.95
5	55.42	53.34	11	63.72	54.96	17	28.88	37.42
6	68.76	58.99	12	52.90	59.48	18	21.13	21.99



(a) Runtime (b) Memory use

Figure 4: Runtime and memory scalability



(a) (b) (c) (d)

Figure 5: Market conditions under different constraints: (a) and (b), variations in system efficiency under different constraint settings; (c) and (d), PoW under current and optimal fare, respectively (peak periods are labelled with boxes).

Experimental results show that FLORA returns exactly the same solution as the existing algorithm, verifying the theoretical correctness of FLORA. In Figure 4, FLORA shows orders of magnitude improvement in speed and memory efficiency over the existing atom schedule method. Each result of the experiments is averaged over 50 runs.

5.2 Effects of Additional Constraints

The second set of experiments uses FLORA-A to examine the necessity of considering other constraints.

5.2.1 Effects of C3 *and* C4

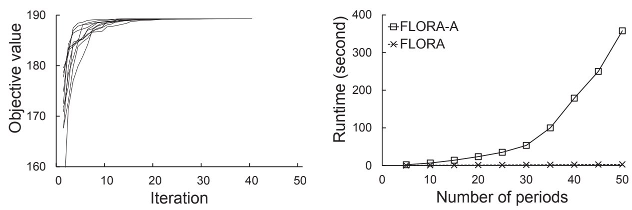
We first consider the effects of **C3** and **C4**. Experiments conducted by the previous work have only considered **C1** and **C2** (**C1,2** for short) in a 18-period problem where each period lasts for one hour [13]. Under this setting, it is reasonable to not consider **C3** and **C4**, since a working or resting section would naturally last for at least one hour (i.e., one period), which is quite realistic. To examine the effects of **C3** and **C4**, we use a more fine-grained 36-period data generated in the first set of experiments, where each period lasts for only 0.5 hour, and set the minimum durations in **C3** and **C4** as $\check{n}_c$=2 and $\check{n}_r$=2, which amounts to one hour and is in line with the 18-period setting. We fix the charge rate of non-peak periods, and optimize the peak periods (i.e., periods 5–8 and period 25–28) charge rate within $\{¥1.00, ¥1.20, \ldots, ¥5.00\}$. The results show that market conditions under **C1,2** and **C1–4** are the same. As depicted in Figure 5(a), the system efficiency (i.e., total number of served customers ($\times10^3$)) are exactly the same such that the two curves overlap completely. This suggests that even if **C3** and **C4** are considered, the taxi drivers still have the flexibility to implement the best strategy under **C1,2** as the limitation of **C3** and **C4** are not likely to be reached. We can thus relax **C3** and **C4** and use FLORA, which is more efficient, to compute the approximately optimal solution.

5.2.2 Effects of C5

Next, we consider constraint **C5** to check the impact of market regulations. We use the 18-period data and consider all **C1** to **C5** to optimize the peak-time fare price as existing work did [13]. 07:00–09:00 a.m. and 05:00–07:00 p.m. are identified as peak time. Figure 5(b) depicts the variations in system efficiency under different constraint settings, where differences as a result of additional constraints can be easily observed. The **C1–5** curve tends to be flatter because the taxi drivers have less flexibility in arranging their schedule in the presence of market regulation defined by **C5**, and are thus less sensitive to the price change. Different optimal charge rates are also obtained (¥2.80 as for **C1,2**, and ¥2.20 as for **C1–5**). In addition, Figure 5(c) depicts PoW in each period with different constraint settings under the current charge rate (f=¥2.00). The valleys in peak periods in the **C1,2** curve are leveled up in the **C1–5** curve when taxi drivers are not allowed to switch off during peak time. Similar patterns can be observed under optimal charge rates in Figure 5(d). These differences suggest inaccuracy as a result of not considering **C5** in scenarios where market regulation takes effect.

5.2.3 *Performance of FLORA-A*

In addition, we show the performance of FLORA-A in the above experiments. Figure 6(a) shows the convergence of 20 runs, where each run solves an 18-period problem with **C1–5** being considered. All the runs end in less than 50 iterations, and converge very quickly to the optimal value. Figure 6(b) depicts the scalability of FLORA-A in comparison with that of FLORA. FLORA-A is less efficient than FLORA as a tradeoff for being more general.



(a) Convergence (b) Runtime

Figure 6: Performance of FLORA-A

5.3 Discussions

As we can see from the above experiments, while FLORA-A is more general and can be used to solve TEMPs with arbitrary scheduling constraints, FLORA is more efficient in dealing with problems with constraints **C1** and **C2**. Even if in the presence of some other constraints, such as **C3** and **C4**, FLORA provides good approximate solutions as shown in Figure 5(a). Therefore, FLORA can be used for solving TEMPs in cities with normal constraints (e.g., **C1** and **C2**) and constraints that FLORA could provide good approximate solutions (e.g., **C3** and **C4**). For TEMPs in cities where other constraints exists (e.g., regulation constraint **C5**), the more general algorithm FLORA-A has to be used.

6. CONCLUSIONS

Our key contributions in this paper are two novel algorithms: 1) FLORA provides a fully compact representation of the taxi drivers' strategy space, which scales up the model far more efficiently than existing algorithms as is shown both theoretically and experimentally; 2) FLORA-A is the first known method for solving taxi system efficiency optimization problems with arbitrary scheduling constraints. It allows us to examine more extensive market conditions, and we believe that it can also be applied to similar large scale optimization problems in other domains.

Acknowledgments

This work is supported by NSFC grant No. 61202212 and Singapore MOE AcRF Tier 1 grant MOE RG33/13. This research is supported by the National Research Foundation, Prime Ministerąŕs Office, Singapore under its IDM Futures Funding Initiative and administered by the Interactive and Digital Media Programme Office.

REFERENCES

[1] D. Avis, D. Bremner, and R. Seidel. How good are convex hull algorithms? *Computational Geometry*, 7(5):265–301, 1997.

[2] A. L. Bazzan. Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 18:342–375, 2009.

[3] M. E. Beesley. Regulation of taxis. *The Economic Journal*, 83(329):150–172, 1973.

[4] M. E. Beesley and S. Glaister. Information for regulating: the case of taxis. *The Economic Journal*, 93(371):594–615, 1983.

[5] Beijing Transportation Research Center (BTRC). Annual transportation report of Beijing (2011), 2011.

[6] S. P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[7] R. D. Cairns and C. Liston-Heyes. Competition and regulation in the taxi industry. *Journal of Public Economics*, 59(1):1–15, 1996.

[8] G. Danninger-Uchida. Carathéodory theorem. In C. Floudas and P. Pardalos, editors, *Encyclopedia of Optimization*, pages 236–237. Springer US, 2001.

[9] G. W. Douglas. Price regulation and optimal service standards: The taxicab industry. *Journal of Transport Economics and Policy*, pages 116–127, 1972.

[10] K. Dresner and P. Stone. A multiagent approach to autonomousintersection management. *Journal of Artificial Intelligence Research*, 31:591–656, 2008.

[11] T. Fleiner, V. Kaibel, and G. Rote. Upper bounds on the maximal number of facets of 0/1-polytopes. *European Journal of Combinatorics*, 21(1):121–130, 2000.

[12] J. Gan and B. An. Game theoretic considerations for optimizing efficiency of taxi systems. In *AAAI-15 Workshop on Computational Sustainability*, 2015.

[13] J. Gan, B. An, H. Wang, X. Sun, and Z. Shi. Optimal pricing for improving efficiency of taxi systems. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, pages 2811–2818, 2013.

[14] B. Grunbaum, V. Klee, M. A. Perles, and G. C. Shephard. *Convex polytopes*, volume 2. Springer, 1967.

[15] Y.-J. Kim and H. Hwang. Incremental discount policy for taxi fare with price-sensitive demand. *International Journal of Production Economics*, 112(2):895–902, 2008.

[16] N. Pulter, H. Schepperle, and K. Böhm. How agents can help curbing fuel combustion: a performance study of intersection control for fuel-operated vehicles. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'11)*, pages 795–802, 2011.

[17] B. Schaller. Issues in fare policy: Case of the new york taxi industry. *Transportation Research Record:*

Journal of the Transportation Research Board, 1618(1):139–142, 1998.

[18] J. M. Smith and F. Cruz. State dependent travel time models, properties, & kernerąŕs three-phase traffic theory. 2012.

[19] P. Varakantham, S.-F. Cheng, G. J. Gordon, and A. Ahmed. Decision support for agent populations in uncertain and congested environments. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI'12)*, pages 1471–1477, 2012.

[20] X.-F. Xie, S. Smith, and G. Barlow. Schedule-driven coordination for real-time traffic network control. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, pages 323–331, 2012.

[21] H. Yang, C. Fung, K. Wong, and S. Wong. Nonlinear pricing of taxi services. *Transportation Research Part A: Policy and Practice*, 44(5):337–348, 2010.

[22] H. Yang, S. C. Wong, and K. Wong. Demand–supply equilibrium of taxi services in a network under competition and regulation. *Transportation Research Part B: Methodological*, 36(9):799–819, 2002.

[23] H. Yang, M. Ye, W. H.-C. Tang, and S. C. Wong. A multiperiod dynamic model of taxi services with endogenous service intensity. *Operations research*, 53(3):501–515, 2005.