Pricing of parking games with atomic players

Fang He, Yafeng Yin, Zhibin Chen, Jing Zhou

Article info

Article history:
Received 18 May 2014
Received in revised form 4 December 2014
Accepted 5 December 2014

Keywords:
Atomic parking game
Equilibrium
System optimum
Pricing

Abstract

This paper considers a parking competition game where a finite number of vehicles from different origins compete for the same number of parking spaces located at various places in a downtown area to minimize their own parking costs. If one vehicle reaches a desired vacant parking space before another vehicle, it will occupy the space and the other vehicle would have to search elsewhere. We first present a system of nonlinear equations to describe the equilibrium assignment of parking spaces to vehicles, and then discuss optimal pricing schemes that steer such parking competition to a system optimum assignment of parking spaces. These schemes are characterized by a union of polyhedrons. Given that the equilibrium state of parking competition is not unique, we further introduce a valid price vector to ensure that the parking competition outcome will always be system optimum. A sufficient condition is provided for the existence of such a valid price vector. Lastly, we seek for a robust price vector that yields the best worst-case outcome of the parking competition.

1. Introduction

Parking is a growing problem in many large cities around the world. Cruising for parking is time consuming and further contributes to traffic congestion and pollution. For example, Arnott and Rowe (1999) pointed out that over 50% of the cars driving in rush hour are cruising for parking in the downtown areas of Boston and some major European cities. Assuming a three-minute search time, Shoup (2006) estimated that the cruising for a curbside parking space yields 1,825 vehicle-miles travelled each year. For a city like Chicago with over 35,000 curbside parking spaces (Ayala et al., 2012a), this translates into approximately 64 million vehicle-miles travelled, 3 million gallons of gasoline consumed and 30 thousand tons of CO₂ emitted every year (EPA, 2012).

The provision of real-time availability and prices of parking spaces will help drivers find parking spaces quickly. In many cities, this information is being distributed via, e.g., dynamic message signs or Internet. The proliferation of advanced smartphones offers new opportunities for parking information to be more widely disseminated and easily accessed in real time. The adoption of smartphones has been increasing rapidly. The number of smartphones is predicted to triple to 5.6 billion globally by 2019 (Ericsson, 2013). By communicating with, e.g., wireless sensors embedded on parking spaces, smartphone-based parking applications, such as SFpark (http://sfpark.org/), can allow drivers to easily access the information of availability and prices of parking spaces. It is envisioned that smartphone-based parking information systems will be widely...
deployed in the near future, which may intensify the competition of parking spaces and strengthen strategic interactions among drivers in the competition.

Although the pace of parking research has increased markedly in recent years, only a dozen of attempts have been made to capture strategic interactions among drivers when competing for parking and investigate how parking competition affects travel patterns. One group of prior studies integrate parking into bottleneck models to explore how parking competition shapes travel patterns of the morning commute, and then investigate parking policies to improve social welfare (e.g., Arnott et al., 1991; Zhang et al., 2008; Qian et al., 2011; Yang et al., 2013; Liu et al., 2014a,b; Qian and Rajagopal, 2014). See Fosgerau and de Palma (2013) for a recent review of this research direction. Another group of studies examine parking competition in a spatial setting and investigate how parking search strategies affect spatial travel patterns in downtown areas (e.g., Bifulco, 1993; Arnott and Rowse, 1999; Anderson and de Palma, 2004; Arnott and Inci, 2006). Among them, the most relevant to this paper are those that consider a finite number of drivers competing for a limited number of parking spaces. For example, Guo et al. (2013) studied a static game where a group of drivers with the same destination choose between two parking lots to minimize their walking distances to the final destination. The drivers are assumed to know the capacity of both lots and the probability of finding a parking space in either lot. A Nash equilibrium would arise if drivers make decisions simultaneously and have perfect knowledge about the strategies of their fellow drivers. The percentage of drivers using either lot can thus be calculated. Kokolaki et al. (2012) examined a similar parking selection game where drivers compete for scarce low-cost public parking spaces and those who do not win the competition have to drive further for more costly private parking spaces. Considering different levels of knowledge that drivers have on the overall parking demand, the authors derived multiple equilibria and compared them with the system optimum state. Note that both studies aggregate parking spaces into two sets and thus do not capture drivers’ choices of individual parking spaces within each set. Both also assume that each vehicle has the same probability of winning the parking competition, while in reality vehicles often compete on a first-arrive-first-served basis. It is thus the cursing time of each competing vehicle to a particular parking space that determines the winner. In contrast, Ayala et al. (2011, 2012a,b) considered a parking competition game where a finite number of drivers from different origins compete for a finite number of parking spaces that are located at different places. If one driver reaches a vacant parking space before another driver, he or she will occupy the space and the other driver will have to search elsewhere. The authors discussed the system optimum and Nash equilibrium states of the game, and explored the design of a parking pricing scheme to drive the system from Nash equilibrium to system optimum. Considering both vehicles and parking spaces are indivisible and the one-to-one assignments between vehicles and spaces are examined in the proposed game, we hereinafter refer to it as a parking game with atomic players or an atomic parking game.

Recognizing that it may become more practically relevant due to the growing adoption of smartphone-based parking service applications, this paper provides a systematic account of an uncooperative static atomic parking game with complete information. Compared with Ayala et al. (2011, 2012a,b), the contributions of this paper include (i) specifying the normal-form representation of the parking competition game by defining an explicit payoff function; (ii) formulating the equilibrium state of the atomic parking game as a system of nonlinear equations, which is equivalent to depicting all the equilibrium assignments of parking spaces; (iii) investigating the price of anarchy for the game of a given size; (iv) characterizing all the pricing schemes that can make the system optimum assignments of parking spaces satisfy the equilibrium conditions of the parking game; (v) establishing a sufficient condition for the existence of valid price vectors, which ensure that the parking competition outcome will always be system optimum; and (vi) formulating a robust pricing problem and developing a global optimum solution procedure for finding a pricing scheme that yields the best worst-case outcome of the game.

For the remainder, Section 2 describes equilibrium and system optimum assignments of the atomic parking game, followed by a discussion on optimal parking pricing in Section 3. Section 4 formulates robust parking pricing as a min–max problem and proposes a global optimum solution procedure. Lastly, Section 5 concludes the paper.

2. Equilibrium and system optimum

2.1. Equilibrium assignment

We consider a static non-cooperative game where there are a finite number of available parking spaces and a finite number of drivers or vehicles (hereinafter we use “drivers” and “vehicles” interchangeably) compete for them. Considering that in reality drivers can always find a place to park if searching sufficiently far away their final destinations, the number of available parking spaces is thus no less than that of vehicles. Without loss of generality, we assume that the numbers of drivers and spaces are the same. Let $I$ and $J$ denote, respectively, the sets of vehicles searching for parking and available parking spaces. Both sets have the same cardinality of $N$. Let $t_{ij}$ represent the time it takes for vehicle $i \in I$ to travel from its current location to space $j \in J$. It is assumed that $t_{ij}$ is given and fixed, and $t_{ij} = t_{kj}, \forall j, k \neq i$. The latter implies that winners can be indisputably determined. When two vehicles compete for the same space, the one with a smaller $t_{ij}$ will win the competition. If their cursing times are the same, the competition outcome can be random. We leave this scenario to our future study.

---

1 If there are more parking spaces than drivers, the problem can be converted to the one where drivers are limited to select among those attractive parking spaces occupied in a system optimum assignment. Consequently, the numbers of drivers and spaces are the same and the results of this paper thus apply.
With smartphone-based parking applications, it is assumed that drivers know the locations of available parking spaces and then attempt to compete to minimize their individual parking costs. We use $e_{ij}$ to denote the cost for vehicle $i$ to park at space $j$, and $e_{ij}$ is a weighted sum of the driving time $t_{ij}$ and the walking time from space $j$ to the final destination of vehicle $i$. We thus use $PG(e, t)$ to represent a parking game with a cost vector $e = (e_1, \ldots, e_n)$ and a driving time vector $t = (t_1, \ldots, t_n)$, as shown in Fig. 1. We further assume that drivers have perfect knowledge of $(e, t)$ and the strategies of their fellow drivers, and they simultaneously make choices on where to park. This is a strong assumption, which may be more justifiable with the proliferation of smartphone-based parking applications and drivers’ enhanced learning from their day-to-day parking experiences.

Below we summarize the assumptions made so far before proceeding to the investigation of the parking competition problem:

i. The numbers of drivers and parking spaces are finite and the same.
ii. The driving time, i.e., $t_{ij}$, is given and fixed, and $t_{ij} \neq t_{kj}$ $\forall j, i \neq k$.
iii. Drivers have complete information on the driving times, parking costs and the payoffs of their fellow drivers, and simultaneously make choices on where to park.

As drivers compete against each other to minimize their own parking costs, the competition for parking spaces will lead to an equilibrium state or assignment defined as follows:

**Definition 1** (Ayala et al., 2011). For $PG(e, t)$, an assignment of vehicles to spaces is at equilibrium if and only if for any vehicle $i$ and its assigned space $j$, there is no $j' \in J$ such that $e_{ij'} < e_{ij}$ and $t_{ij'} < t_{ij}$, where $i$ is the vehicle assigned to space $j'$.

Define that space $j$ is available to vehicle $i$ if and only if vehicle $i$ can win the space when competing against the vehicle currently assigned to space $j$. Thus, Definition 1 essentially suggests that at equilibrium, no vehicle can further decrease its parking cost by unilaterally switching to one of the parking spaces available to it.

We can also formally represent the parking competition problem as a game of $N$ drivers (players) and interpret the equilibrium assignment as a Nash equilibrium. To see this, consider each driver's strategy space is $J$, the set of parking spaces. Given other drivers’ choices, the payoff for driver $i$ to choose space $j$, i.e., $C_{ij}$, can be defined as $-|e_{ij} + M(t_{ij} - \min_{k \in G(j)} t_{kj})|$, where $M$ is a sufficiently large constant and $G(j)$ is the set of all the vehicles choosing space $j$ to park their vehicles (vehicle $i$ is included). Thus, Definition 1 essentially suggests a Nash equilibrium where no driver can further increase his or her payoff by unilaterally switching to another parking space.

![Fig. 1. The conceptual setting of an atomic parking game.](image-url)
Using a vector \( \mathbf{x} = (\cdots, x_{ij}, \cdots) \) to represent a feasible assignment of \( PG(\mathbf{e}, t) \), below we formulate the problem of finding an equilibrium assignment, as defined in Definition 1, as a system of nonlinear equations.

**Proposition 1.** For \( PG(\mathbf{e}, t) \), assignment \( \mathbf{x} \) is at equilibrium if and only if \( \mathbf{x} \) satisfies the following nonlinear equation system:

\[
\begin{align*}
\sum_{i} x_{ij} &= 1 \quad \forall j \\
\sum_{j} x_{ij} &= 1 \quad \forall i \\
x_{ij}(1 - x_{ij}) &= 0 \quad \forall ij \\
x_{ij} \left[ e_{ij} + \gamma_{ij} \left( t_{ij} - \sum_{k} t_{kj} x_{kj} \right) - \mu_{l} \right] &= 0 \quad \forall ij \\
\sum_{j} x_{ij} \left[ e_{ij} + \gamma_{ij} \left( t_{ij} - \sum_{k} t_{kj} x_{kj} \right) - \mu_{l} \right] &\geq 0 \quad \forall i \\
\gamma_{ij} &\geq 0 \quad \forall ij
\end{align*}
\]

where \( \mu_{l} \) and \( \gamma_{ij} \) are auxiliary variables.

**Proof.** We first prove if \( \mathbf{x} \) satisfies (1)–(6), then \( \mathbf{x} \) is an equilibrium assignment. From (1)–(3), \( \mathbf{x} \) is a feasible assignment. Define set \( G_{x} = \left\{ (i, j) | x_{ij} = 1 \right\} \). Because of (1), for \( \forall j \in J \), we have \( \sum_{i} x_{ij} = t_{ij}, (i, j) \in G_{x} \). Therefore, for \( \forall (i, j) \in G_{x} \), (4) implies that \( e_{ij} - \mu_{l} = 0 \) and thus \( \mu_{l} \) is the equilibrium parking cost for vehicle \( i \). For \( \forall (i, j) \not\in G_{x} \), if \( t_{ij} < t_{kj}, (i, j) \in G_{x} \), from (5), we have \( e_{ij} = \mu_{l} + \gamma_{ij} (-t_{ij} + \sum_{j} t_{ij} x_{ij}) = \mu_{l} + \gamma_{ij} (-t_{ij} + t_{ij}) \geq \mu_{l} \), implying vehicle \( i \) has no incentive to switch to space \( j \) even if space \( j \) is available to vehicle \( i \), i.e., \( t_{ij} < t_{kj}, (i, j) \in G_{x} \). Therefore, \( \mathbf{x} \) is an equilibrium assignment.

We then prove if \( \mathbf{x} \) is at equilibrium, \( (\cdots, \mu_{l}, \cdots) \) and \( (\cdots, \gamma_{ij}, \cdots) \) can be found such that (1)–(6) are satisfied. For \( \forall (i, j) \in G_{x} \), set \( \mu_{l} = e_{ij} \). It can be verified that (1)–(4) are satisfied. We now show how to find \( (\cdots, \gamma_{ij}, \cdots) \) to satisfy (5) and (6). For \( \forall (i, j) \not\in G_{x} \), if \( t_{ij} < t_{kj}, (i, j) \in G_{x} \), we have \( e_{ij} = \mu_{l} \) from Definition 1. Therefore, (5) can be satisfied by simply setting \( \gamma_{ij} = 0 \). If \( t_{ij} > t_{kj}, (i, j) \in G_{x} \), (5) will always hold if we make \( \gamma_{ij} \) sufficiently large. Therefore, it is proved that if \( \mathbf{x} \) is at equilibrium, the conditions specified by (1)–(6) hold. \( \Box \)

Below, we provide a small example with three vehicles, i.e., \( V_{1}, V_{2} \) and \( V_{3} \), and three spaces, i.e., \( S_{1}, S_{2} \) and \( S_{3} \). Table 1 specifies \( (\mathbf{e}, t) \) with the first number in each cell representing \( e_{ij} \) and the second \( t_{ij} \). Two assignments, i.e., \( (V_{1}, S_{1}), (V_{2}, S_{2}) \) and \( (V_{3}, S_{3}), (V_{2}, S_{2}) \) and \( (V_{3}, S_{1}) \), can be verified to be consistent with Definition 1, and are thus equilibrium assignments. To satisfy the equilibrium conditions of (1)–(6), for the former assignment, \( \mu_{1} = \mu_{3} = 4, \mu_{2} = 2, \gamma_{12} = \gamma_{13} = \gamma_{31} = \gamma_{32} = 10 \) and others are zero; for the latter, \( \mu_{1} = 3, \mu_{2} = \mu_{3} = 2, \gamma_{12} = 10 \) and others equal zero. Note that \( \gamma \) is not unique.

Ayala et al. (2012b) pointed out that the above atomic parking game is equivalent to the stable marriage problem first introduced by Gale and Shapley (1962), which considers a community consisting of the same number of men and women, with each ranking those of the opposite sex based on his or her preferences for a marriage partner. A matching between a group of women and men is called unstable if there exist two men \( Q_{1} \) and \( Q_{2} \) marrying woman \( q_{1} \) and \( q_{2} \), respectively, although \( Q_{1} \) prefers \( q_{2} \) to \( q_{1} \) and \( q_{2} \) prefers \( Q_{1} \) to \( Q_{2} \). For any \( PG(\mathbf{e}, t) \), we can construct a corresponding stable marriage problem where the preference orders of vehicles over spaces are determined by cost \( \mathbf{e} \), i.e., vehicles prefer spaces with smaller costs, and the preference orders of spaces over vehicles are by driving time \( t \), i.e., spaces "prefer" vehicles that are closer to them. Consequently, an assignment of spaces to vehicles is at equilibrium if and only if it is a stable matching of the corresponding marriage problem. For example, the marriage problem corresponding to the above parking game has preference orders shown in Table 2 where the first number in each cell represents the vehicle’s preference rank for the space, and the second is the space’s preference rank for the vehicle. It can be verified that stable matchings in the marriage problem pair vehicles and spaces in the same way as equilibrium assignments in the atomic parking game.

Gale and Shapley (1962) proved the existence of a stable matching for any marriage problem, and proposed a deferred acceptance algorithm to find a stable matching. Recognizing the equivalence of the atomic parking game and stable marriage problem, we thus conclude that for any \( PG(\mathbf{e}, t) \), there always exists at least one equilibrium assignment, which can be identified by using Gale and Shapley’s deferred acceptance algorithm, rather than directly solving the nonlinear system of (1)–(6). Moreover, since equilibrium assignments of parking games are generally not unique, we can implement other algorithms

<table>
<thead>
<tr>
<th>Table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parking cost and driving time.</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>( V_{1} )</td>
</tr>
<tr>
<td>( V_{2} )</td>
</tr>
<tr>
<td>( V_{3} )</td>
</tr>
</tbody>
</table>
proposed for the stable marriage problem, e.g., McVitie and Wilson (1971) and Gusfield (1987), to enumerate all the possible equilibrium assignments. Lastly, if we denote by \( f(N) \) the maximum number of equilibrium assignments for a parking game with \( N \) spaces and \( N \) vehicles, \( f(N) \) is a strictly increasing function of \( N \). Various lower bounds of \( f(N) \) have been proposed in the literature of the stable marriage problem (e.g., Irving and Leather, 1986; Benjamin et al., 1995; Thurber, 2002). For instance, Thurber (2002) showed \( f(N) > \frac{\log N}{1 + \sqrt{3}} \), implying that \( f(N) \) increases at least exponentially.

### 2.2. System optimum and price of anarchy

The system optimum assignment of the atomic parking game can be defined as the one that minimizes the total parking cost. Below we formulate a linear program to seek for such an assignment:

**SO:**

\[
\begin{align*}
\text{min} \sum_{ij} x_{ij}c_{ij} \\
\text{s.t.} & \quad \sum_{i} x_{ij} \leq 1 \quad \forall j \\
& \quad x_{ij} \geq 0 \quad \forall i, j
\end{align*}
\]

(7)

(8)

Compared to (1)–(3), the constraints in the above formulation do not require decision variables to be binary and each parking space assigned one vehicle. Below we prove the validity of the formulation.

**Proposition 2.** The integer optimal solution to SO is a system optimum assignment.

**Proof.** The matrix associated with constraints (2) and (7) is totally unimodular, and SO admits integer optimal solutions. We now prove constraint (7) is always binding at optimality via contradiction. Assume that there exists \( j \in J \) in an optimal solution such that \( \sum_{i} x_{ij} < 1 \). Because the optimal solution is integer, we have \( \sum_{i} x_{ij} = 0 \), implying \( x_{ij} = 0 \). Considering \( |I| = |J| = N \), to satisfy (2), there must exist \( j' \in J, j' \neq j \), such that \( \sum_{i} x_{ij'} > 1 \), which is contradictory to (7). Therefore, constraint (7) is always binding at optimality. The solution is thus a system optimum assignment. \( \square \)

The price of anarchy of atomic parking games is defined as the maximal ratio between the total cost of an equilibrium assignment and that of a system optimum assignment (e.g., Roughgarden, 2005). Ayala et al. (2011) proved that the price of anarchy of atomic parking games is unbounded because it increases with the size of the games. Below we investigate the price of anarchy for parking games of a given size.

**Proposition 3.** Suppose PG(\( e, t \)) includes \( N \) drivers and \( N \) parking spaces and assume \( e_{ij} \in [c^-, c^+] \), \( \forall i, j \). The price of anarchy for the game is \( \frac{N}{N-1} \). Moreover, if there exists a nonempty set \( U = \{ (i,j) | t_{ij} < t_{i'j'} \forall i' \in I, i \neq i', e_{ij} < e_{i'j'} \forall j' \in J, j \neq j' \} \), then the price of anarchy is \( \frac{N-|U|}{N-1} \), where \( |U| \) is the cardinality of the set.

**Proof.** It is straightforward to observe that the ratio between the total costs of equilibrium and optimum assignments is less than \( \frac{N}{N-1} \), because the total parking cost of an equilibrium assignment is no greater than \( Nc^+ \) while the total cost of a system optimum assignment is no less than \( Nc^- \). We can easily construct a game whose ratio is exactly \( \frac{N}{N-1} \), e.g., a game with \( t_{ij} < t_{i'j'} \), \( i' \neq i, j = j \); \( e_{ij} = c^+ \), \( \forall ij, i \neq j \) and \( e_{ij} = c^- \), \( \forall ij, i \neq j \). Thus the first statement holds.

If there exists a nonempty set \( U \) as defined above, for \( (i, j) \in U \), space \( j \) should be always assigned to vehicle \( i \) at any equilibrium assignment. Denote by \( l_i \in J \) the assigned space for vehicle \( i \) at equilibrium and by \( l_i \in J \) as its assigned space at optimum. Without loss of generality, we index the vehicles in \( U \) as 1 to \( |U| \), and then have

---

2 Another objective can be defined to represent a different managing goal.

<table>
<thead>
<tr>
<th>( S_1 )</th>
<th>( S_2 )</th>
<th>( S_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V_1 )</td>
<td>3, 1</td>
<td>1, 2</td>
</tr>
<tr>
<td>( V_2 )</td>
<td>2, 3</td>
<td>1, 1</td>
</tr>
<tr>
<td>( V_3 )</td>
<td>1, 2</td>
<td>2, 3</td>
</tr>
</tbody>
</table>
Similarly, we can construct another game to demonstrate that this upper bound is tight, and thus $\frac{\max - \min}{\min}$ is proved to be the price of anarchy. □

3. Optimal pricing

We now assume a parking management agency has complete information of available parking spaces, and current locations and final destinations of vehicles searching for parking, and attempts to set the price of each available parking space to induce drivers to choose parking spaces in a way that minimizes the total parking cost. In other words, the agency wishes to replicate or decentralize the system optimum assignment. Below we investigate how to determine such parking pricing strategies.

3.1. Optimal price vectors

Let $X^*$ denote the set of all system optimum assignments, i.e., $X^* = \arg\min \left\{ \sum_p x_p e_p \right\}$. We define an optimal price vector as the one that makes at least one system optimum assignment satisfy the equilibrium conditions of (1)–(6). To simplify the notation, parking price is expressed in the same unit of parking cost. Denote by $R_p$ the set of all possible equilibrium assignments under a price vector $p$. Define $W(x) = \{ p | x \in R_p \}$. Below we prove that the set of all optimal price vectors is a union of polyhedrons.

**Proposition 4.** For $PG(e, t)$, the set of all optimal price vectors is a union of polyhedrons, i.e., $T^1 = \bigcup_{x \in X^*} W(x)$, where polyhedron $W(x)$ is defined as follows:

\[
\begin{align*}
\hat{x}_i & \left[ e_i + p_j + \gamma_i \left( t_{ij} - \sum_k t_{kj} \hat{x}_k \right) \right] = 0 \quad \forall i, j \\
e_i & + p_j + \gamma_i \left( t_{ij} - \sum_k t_{kj} \hat{x}_k \right) - \mu_i \geq 0 \quad \forall i, j \\
\gamma_i & \geq 0 \quad \forall i, j
\end{align*}
\]

Proof. For a given system optimum assignment $\hat{x}$, the equilibrium conditions of (1)–(6) with a price vector reduce to (9)–(11) and define a polyhedron $W(x)$. Based on the definition of optimal price vectors, it is straightforward to observe that $T^1 = \bigcup_{x \in X^*} W(x)$. □

For the numerical example introduced in Section 2, the system optimum assignment is $S_0 = (V_1, S_1), (V_2, S_2)$ and $(V_3, S_1)$, which is unique, and actually one of the equilibrium assignments. The set of all optimal price vectors is a polyhedron written as follows:

\[
\begin{align*}
3 + p_3 - \mu_1 &= 0 \\
2 + p_2 - \mu_2 &= 0 \\
2 + p_1 - \mu_3 &= 0 \\
4 + p_1 - 3\gamma_{11} - \mu_1 &= 0 \\
2 + p_2 + 4\gamma_{12} - \mu_1 &= 0 \\
3 + p_1 - 3\gamma_{21} - \mu_2 &= 0 \\
4 + p_2 + 2\gamma_{22} - \mu_2 &= 0 \\
3 + p_2 + 2\gamma_{23} - \mu_3 &= 0 \\
4 + p_3 - 2\gamma_{33} - \mu_3 &= 0 \\
\end{align*}
\]

\[
\begin{align*}
\gamma_{11}, \gamma_{12}, \gamma_{21}, \gamma_{22}, \gamma_{23}, \gamma_{33} &\geq 0
\end{align*}
\]

Note that in the formulation of SO, the Lagrangian multiplier associated with constraint (7) represents the marginal cost of parking a vehicle at space $j$. Inspired by the concept of marginal-cost congestion pricing (e.g., Pigou, 1920), we conjecture that charging drivers the marginal cost of parking at each space, i.e., the Lagrangian multiplier of constraint (7), can make the system optimum assignment satisfy the equilibrium conditions. In other words, the Lagrangian multipliers of constraint (7) are an optimal price vector. We prove this conjecture by examining the KKT conditions of SO, which are written below:
The KKT conditions yield (9)–(11), suggesting that the multiplier vector \((\gamma_1, \gamma_2, \cdots)\) is an optimal price vector. This thus guarantees that the set of optimal price vectors \(T^1\) is not empty. In addition, we note that besides this multiplier vector, there may be other optimal vectors in the set. For instance, \(p = 0\) is an optimal price vector for the numerical example in Section 2, but one can verify that it violates the KKT conditions of \(SO\) and is thus not a vector of Lagrangian multipliers.

Ayala et al. (2012a) proposed an interesting auction procedure to search for an optimal price vector. In the auction, each round starts with an assignment and a set of prices. If all drivers are happy with the current assignment, the procedure terminates. Otherwise, an unhappy driver will be selected. The driver will increase the price of his or her first preferred space, which eliminates. Pointed out by the authors, the auction can be viewed as an iterative procedure to solve the dual problem of \(SO\). Similar auction algorithms have been discussed in the literature, e.g., Bertsekas (1990), for assignment problems. As such, the optimal price vector prescribed in Ayala et al. (2012a) is the multiplier vector of the \(SO\) problem (see a formal proof in Appendix A). Other optimal price vectors may exist for the same parking game.

### 3.2. Valid price vectors

As the equilibrium outcome of the parking competition may not be unique, an optimal price vector may yield multiple equilibrium assignments. At least one of them is the system optimum assignment while the other may not, which can compromise the effectiveness of parking pricing. A parking scheme originally designed for achieving system optimum may yield an undesirable outcome and actually increase total parking cost. To overcome this limitation, we introduce the concept of valid price vectors, similar to valid toll vectors discussed in Hearn and Ramana (1998) for congestion road pricing. A parking pricing scheme \(p\) is called a valid price vector if and only if it satisfies \(\Phi \neq R_p \subseteq X^\ast\), implying that all the equilibrium assignments resulted by a valid price vector are system optimum. The set of all the valid price vectors, i.e., \(T^2\), can be defined as follows:

\[
T^2 = \bigcup_{x \in X^\ast} W(x) \setminus \bigcup_{x \in X} W(x)
\]

where \(x \notin X^\ast\) denotes any non-system-optimum feasible assignment. Obviously, \(T^2 \subseteq T^1\).

A valid price vector does not necessarily exist for a parking game, i.e., \(T^2\) can be an empty set. For example, consider a parking game with three vehicles and three spaces, and \(e\) and \(t\) reported in Table 3. Again, the first number in each cell represents the parking cost and the second one is the driving time. It can be verified that the optimum assignment of this example is unique, and pairs \((V_1, S_1), (V_2, S_3)\) and \((V_3, S_2)\). Therefore, \(T^1 = \bigcup_{x \in X} W(x)\) is a polyhedron specified by Proposition 4.

Consider a non-system-optimum feasible assignment \(x\) with \(x_{11}, x_{12}\) and \(x_{13}\) being one and others zero. We now show \(W(x) = R^1\), i.e., for any price vector, there always exist \((\cdots, \mu_i, \cdots)\) and \((\cdots, \gamma_j, \cdots)\) that make \(x\) satisfy (1)–(6). Specifically, for \(\forall p \in R^1\), setting \(\mu_1 = e_{11} + p_1\), \(\mu_2 = e_{22} + p_2\) and \(\mu_3 = e_{33} + p_3\) make (4) hold. For \(\forall ij \notin G_x\), from Table 3, \(t_{ij} - \sum_k t_{kj} x_{kj} > 0\). Thus, if \(\gamma_j\) is sufficiently large, (5) can always be satisfied as well as (6). Thus \(W(x) = R^1\). Because \(T^2 = T^1 \setminus \bigcup_{x \in X^\ast} W(x)\), \(T^2\) is empty.

Considering valid price vectors do not always exist, below we provide a sufficient condition for the existence of such vectors.

**Proposition 5.** A valid price vector exists for \(PG(e, t)\) if there exist for the game a system optimum assignment, i.e., \(x^\ast\), and ordered arrays of vehicles and spaces, i.e., \((\cdots, i - 1, i, i + 1, \cdots)\) and \((\cdots, j - 1, j, j + 1, \cdots)\), with \(x^\ast_{ij} = 1\) and \(t_{ij} < t_{k,j}\) for \(\forall i = j\) and \(k > i\).

<table>
<thead>
<tr>
<th>(S_1)</th>
<th>(S_2)</th>
<th>(S_3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(V_1)</td>
<td>2, 0</td>
<td>4, 1</td>
</tr>
<tr>
<td>(V_2)</td>
<td>3, 4</td>
<td>4, 0</td>
</tr>
<tr>
<td>(V_3)</td>
<td>4, 1</td>
<td>3, 2</td>
</tr>
</tbody>
</table>
Proof. Our proof is built upon the sufficient condition for the uniqueness of stable marriage matching proposed by Eeckhout (2000). With the ordered sets \((\ldots, i - 1, i, i + 1, \ldots)\) and \((\ldots, j - 1, j, j + 1, \ldots)\) where \(t_{ij} < t_{k,j} \forall i, k > i\), a price vector \(p\) can be constructed to ensure \(e_{ij} + p_j < e_{ij} + p_i \forall i, l > j\) (see Appendix B for a procedure). Consequently, the sequential preference condition proposed by Eeckhout (2000) is satisfied, which ensures that the system optimum assignment \(x^*\) is the unique equilibrium assignment under \(p\). As such, \(p \in \bigcup_{x \in \mathcal{X}} \mathcal{W}(x)\) and \(p \notin \bigcup_{x \in \mathcal{X}} \mathcal{W}(x)\). Therefore, \(p \in T^2\). □

From the perspective of the stable marriage problem, the sequential preference condition sets the requirements for both preference orders of vehicles and spaces. In the above, we essentially utilize parking pricing to affect the preference order of vehicles over spaces, while the preference order of spaces over vehicles remains intact. Thus, Proposition 5 needs to specify the requirement on the driving time to ensure the preference order of spaces over vehicles to satisfy the sequential preference condition. It is also worth noting that the uniqueness of equilibrium of a parking game depends on the parking pricing scheme. In contrast, for congestion pricing of traffic networks (e.g., Hearn and Ramana, 1998), the uniqueness of user equilibrium flow distribution is often not affected by the tolling scheme. Consequently, if equilibrium is unique, we will have \(T^2 = T^1\). However, this does not apply to pricing of atomic parking games.

Below is a small example to demonstrate the above proposition. Consider a three-to-three assignment with the same parking costs as Table 3. The new driving times are shown in Table 4.

It can be verified that the ordered array of vehicles, i.e., \((V_1, V_2, V_3)\), and the ordered array of spaces, i.e., \((S_1, S_2, S_3)\), and the system optimum assignment, i.e., \((V_1, S_1), (V_2, S_2)\) and \((V_3, S_3)\), satisfy the conditions in Proposition 5. Using the procedure in Appendix B, if \(\gamma = 0.5\), a valid price vector can be obtained as \(p_1 = p_2 = 0, p_3 = 1.5\). Note that in this example, all spaces have the same preference order for vehicles, i.e., \((V_1, V_2, V_3)\). Such a preference profile satisfies the conditions for Proposition 5. Indeed, for any parking game, if all spaces have the same preference order for vehicles, Proposition 5 holds. This is similar to the vertical heterogeneity discussed in Eeckhout (2000).

4. Robust pricing

It is likely that valid price vectors do not exist in a significant number of parking games in practice. Consequently, the challenge imposed by the non-uniqueness of equilibrium still remains. This sections offer another remedy. In the same spirit of robust pricing of traffic networks (e.g., Lou et al., 2010; Ban et al., 2009), we aim at designing a parking pricing scheme that minimizes the total parking cost of the worst-case equilibrium assignment incurred by the scheme. In other words, we seek for a parking price vector whose worst-case performance is the best. Below is a robust pricing formulation to identify such a price vector:

**RP:**

\[
\min_{p} \max_{x, l: p} \sum_{ij} x_{ij} e_{ij}
\]

\[
\text{s.t.} (1)-(3), (6)
\]

\[
x_{ij} \left[ e_{ij} + p_j + \gamma_{ij} \left( t_{ij} - \sum_k t_{ij} x_{kj} \right) - \mu_i \right] = 0 \ \forall ij
\]

\[
e_{ij} + p_j + \gamma_{ij} \left( t_{ij} - \sum_k t_{ij} x_{kj} \right) - \mu_i \geq 0 \ \forall ij
\]

As formulated, RP is a min–max problem with complementarity constraints, a class of problems difficult to solve. Below we develop a solution procedure to seek for a global optimal solution. The procedure involves solving a sequence of a master problem, a relaxed RP because it limits the search for the worst equilibrium assignment within a given subproblem. For a particular \(x^*\), \(s \in S\), we introduce two new sets, i.e., \(A^*_1 = \{(i,j) | x_{ij}^* = 1\}\) and \(A^*_2 = \{(i,j) | x_{ij}^* = 0, \exists k, t_{ij} < t_{kj}, x_{kj} = 1\}\). The master problem is formulated as follows:

<table>
<thead>
<tr>
<th>(t_i)</th>
<th>(S_1)</th>
<th>(S_2)</th>
<th>(S_3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(V_1)</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>(V_2)</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>(V_3)</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>
\[
\begin{align*}
\min_{\psi} & \quad \psi \\
\text{s.t.} & \quad z^e M + \psi \geq \sum_s x^e_s e_{ij} \quad \forall s \in S \quad (17) \\
& \quad e_{ij} + p_j - \mu_i^t = 0 \quad \forall ij \in A^t_1 \quad (18) \\
& \quad - (e_{ij} + p_j - \mu_i^t) + (1 - y^t_{ij}) M \geq \varepsilon \quad \forall ij \in A^t_2 \quad (19) \\
& \quad z^e = \sum_y y^e_{ij} \quad \forall s \in S \quad (20) \\
& \quad y^t_{ij} \in \{0, 1\} \quad \forall s \in S, ij \quad (21) \\
& \quad z^e \in \{0, 1\} \quad \forall s \in S \quad (22) \\
& \quad \psi \geq 0 \quad (23)
\end{align*}
\]

where \( \psi \) equals the maximum parking cost of all the possible equilibrium assignments within set \( H \) under pricing scheme \( p \); \( y^t_{ij} \) and \( z^e \) are two binary variables and \( M \) is a sufficiently large constant.

Constraints (18)–(21) dictate that if \( x^e \) is an equilibrium assignment under \( p \), \( z^e \) equals zero. To achieve a better computational efficiency, we only enforce “approximately” or “almost” equilibrium conditions. More specifically, given an equilibrium assignment \( x \), for \( x_q = 1, x_q = 1 \), if \( t_q < t_{q'} \), Definition 1 requires \( e_q - e_{q'} \geq 0 \). Instead, we now relax this condition to be \( e_q - e_{q'} > -\varepsilon \) where \( \varepsilon \) is a small positive constant. Therefore, (19) suggests that if \( e_q + p_j - \mu_i^t > -\varepsilon: y^t_{ij} \) equals zero. If \( x^e \) is an approximately equilibrium assignment under \( p \), the binary variable of \( z^e \) equals zero. Considering the objective function is to minimize \( \psi \), (17) essentially states that \( z^e \) equals one if \( x^e \) does not satisfy the approximately equilibrium conditions, and \( \psi \) thus equals the maximum parking cost of all the approximately equilibrium assignments within \( H \) under \( p \). Constraint (18) sets the parking cost for each parking space. Lastly, constraint (23) requires \( \psi \) to be non-negative. Note that the optimal objective function value of the master problem provides a lower bound for RP.

Suppose that \( p^* \) is an optimal solution to the master problem. Our sub problem then aims at finding the worst equilibrium assignment under \( p^* \). We treat \( x_q \) as a binary variable and then introduce another new binary variable \( s_{ikj} \) to linearize a non-linear term \( \gamma_{ij} \). The sub problem is thus formulated as the following mixed integer program:

\[
\begin{align*}
\max_{x^e, s_{ikj}} & \quad \sum_{s} x^e_s e_{ij} \\
\text{s.t.} & \quad (1), (2), \text{ and } (11) \\
& \quad e_{ij} + p_j + \gamma_{ij} t_{ij} - \sum_{ikj} t_{ik}s_{ikj} - \mu_i \geq -\varepsilon + \varepsilon x_{ij} \quad \forall ij \quad (24) \\
& \quad e_{ij} + p_j + \gamma_{ij} t_{ij} - \sum_{ikj} t_{ik}s_{ikj} - \mu_i \leq M(1 - x_{ij}) \quad \forall ij \quad (25) \\
& \quad s_{ikj} \leq Mx_{ikj} \quad \forall ikj \quad (26) \\
& \quad s_{ikj} \geq 0 \quad \forall ikj \quad (27) \\
& \quad s_{ikj} \leq \gamma_{ij} \quad \forall ikj \quad (28) \\
& \quad s_{ikj} \geq Mx_{ikj} + \gamma_{ij} - M \quad \forall ikj \quad (29) \\
& \quad x_{ij} \in \{0, 1\} \quad \forall ij \quad (30)
\end{align*}
\]

Constraints (26)–(29) is equivalent to setting \( s_{ikj} = \gamma_{ij} x_{ikj} \). Therefore, constraints (24), (25) and (30) are corresponding to (4) and (5). Once again, we enforce approximately equilibrium conditions. Lastly, the optimal objective function value is the maximum equilibrium parking costs under \( p^* \) and thus provides an upper bound for RP.

After introducing the master and sub problems, we are ready to present the procedure, called as GO, to solve RP. GO solves the master and sub problems iteratively to update the upper and lower bounds for RP until these two bounds are equal. Let \( \theta_l \) and \( \theta_u \) be the lower and upper bounds of RP and the procedure of GO is shown as follows:

**Step 0:** Solve the sub problem with \( p^* = 0 \).

Set \( \theta_l = \sum_s x^e_s e_{ij} \) and \( \theta_l = 0 \). Add \( x^e \) into \( H \)

**Step 1:** Solve the master problem with \( H \) and \( p^* \) is the optimal pricing scheme.

If \( \psi^* > \theta_l \), set \( \theta_l = \psi^* \).

**Step 2:** Solve the sub problem with \( p^* \).

If \( \sum_s x^e_s e_{ij} = \theta_u \), stop and \( p^* \) is the optimal solution to RP.

Otherwise, add \( x^e \) into \( H \). If \( \sum_s x^e_s e_{ij} < \theta_u \), set \( \theta_u = \sum_s x^e_s e_{ij} \). Go to Step 1.

In the above, solving the master problem provides a non-decreasing lower bound for RP. The procedure proceeds until the upper and lower bounds match, suggesting that the solution is a global optimum to RP (with approximately equilibrium conditions). At Step 2, solving the sub problem will generate a feasible assignment. If it is already included in \( H \), the best
obtained lower and upper bounds will be equal and the procedure would terminate. Only if it is new, the procedure may continue. Therefore, the assignments added to $H$ will never repeat. Because there are a finite number of feasible assignments, the procedure will thus terminate within a finite number of steps.

To test the performance of GO, we solve a parking game with six vehicles and six parking spaces. We randomly generate five scenarios where the parking cost and driving time vary between 10 and 200. We solve these five instances with GO (setting $c$ being one) and a pattern search algorithm proposed by Custódio and Vicente (2007) and Custódio et al. (2010). A personal computer with Intel(R) Core(TM) i7-2640M Duo 2.8 GHz CPU, 6 GB RAM, and Windows 7 Home Premium operating system is used for all the tests. When implementing GO, we use CPLEX 12.2 to solve the master and sub problems, setting the absolute stopping tolerance as zero.

Table 5 compares the CPU times of the two algorithms, and lists the relative gap between the optimal total parking costs. It can be observed that for scenarios 2 and 4, the pattern search algorithm can find global optimum but take much longer time. Its CPU time is 9–18 times that of GO. In addition, for scenarios 1, 3 and 5, the pattern search algorithm yields worse solutions with an 11–30% relative gap. Except scenario 1, the CPU time of the pattern search algorithm is more than two times that of GO.

In order to further demonstrate GO’s potential for solving larger games, we solve another parking game with 50 vehicles and 50 parking spaces. Three scenarios are randomly generated where the parking cost and driving time vary between 10 and 200. We solve these three instances with GO, setting the iteration limit to be 100. Moreover, the approximate equilibrium conditions are only implemented for the master problem, and thus the assignments generated by the sub-problem are exact equilibrium solutions. Table 6 lists the CPU times, the maximal total costs under the best-obtained pricing scheme and the maximal costs without parking pricing for the three cases. The stopping criteria, i.e., $h_l = h_u$, is not satisfied for any case. However, it is observed that in all three scenarios, CPU times are about 20 min, and approximately 30% reduction in total costs is achieved compared to the scenario without parking pricing.

5. Conclusions

We have analyzed and priced a parking game where a finite number of drivers compete for the same number of parking spaces initiated by Ayala et al. (2011, 2012a,b). Specifically, we have formulated the equilibrium and system optimum assignments of parking spaces to vehicles and investigated the price of anarchy for the atomic parking games of a given size. We have further characterized optimal price vectors as a union of polyhedrons, which make system optimum assignments satisfy the equilibrium conditions. However, because the equilibrium of a parking game may not be unique, imposing those price vectors do not necessarily lead to a better system performance, not mentioning the system optimum. We have then offered two remedies. One is to search for a valid price vector that guarantees the resulting equilibrium assignments will always be system optimum. Unfortunately, such a valid price vector does not always exist for a particular game. Having provided a sufficient condition to ensure its existence, we have further suggested applying a robust pricing design to optimize the worst-case performance. A global optimum solution procedure has been proposed to solve the robust price design problem. Numerical experiments have demonstrated that the procedure works well with small-size problems, and has the potential of solving realistic problems.

We conclude this paper by pointing out that in this paper drivers are assumed to have complete information on the curing times, parking costs, and strategies of their fellow drivers. Our future study will examine more realistic settings with incomplete information. A dynamic version of the game is also more of practical relevance. It is interesting to investigate how to determine a time-dependent pricing strategy to effectively manage parking spaces, considering the uncertainty associated with parking demand and supply.
Acknowledgements

The research is partially supported by grants from the U.S. National Science Foundation (CMMI-1362631 and CNS-1239364) and the National Natural Science Foundation of China (71228101). The paper benefits substantially from discussions with Profs. Ouri Wolfson and Jie Lin at University of Illinois at Chicago, Hai Yang at Hong Kong University of Science and Technology and William Lam at Hong Kong Polytechnic University.

Appendix A

We show that the optimal price vector prescribed in Ayala et al. (2012a) is the multiplier vector of the SO problem. Specifically, the dual of the SO problem is written as follows:

\[
\max \sum_i \left[ \min_j (\beta_j + e_{ij}) \right] - \sum_j \beta_j
\]

s.t. \( \beta_j \geq 0 \)

Denote the optimal objective values of the SO and its dual problems as \( \pi_1 \) and \( \pi_2 \) respectively. Let \( \bar{x} \) and \( \bar{p} \) represent the assignment and price vector with which the auction procedure terminates. We thus have \( \sum_j e_{ij} \bar{x}_{ij} = \sum_j (e_{ij} + \beta_j) \) \( \bar{x}_{ij} = \sum_j \beta_j = \sum_j \min_j (e_{ij} + \beta_j) - \sum_j \beta_j \leq \pi_2 \) where the second equality is from the fact that the auction terminates only if all the drivers are happy with the assignment, and the first inequality holds because \( \bar{x} \) is feasibility to SO. Based on the strong duality of linear programming, \( \pi_1 = \pi_2 \). Therefore, \( \sum_j \min_j (e_{ij} + \beta_j) - \sum_j \beta_j = \pi_2 \), implying that \( \bar{p} \) solves the dual problem, i.e., \( \bar{p} \) is the vector of Lagrangian multipliers associated with (2).

Appendix B

A procedure to find a valid price vector \( \bar{p} \).

**Step 1.** Set \( \bar{p}_j = 0 \), \( \forall j = 1, \ldots, N \)

**Step 2.** Loop \( i, i = 1, \ldots, N \)

Loop \( j = i + 1, \ldots, N \)

If \( e_{ij} + \bar{p}_j > e_{ij} + \bar{p}_i \), do nothing.

Otherwise, increase \( \bar{p}_j = e_{ij} + \bar{p}_i - e_{ij} + \epsilon \)

End if

End loop

End loop

where \( \epsilon \) is a small positive number.

References


