

# CRUISING FOR PARKING AROUND A CIRCLE

Richard Arnott\* and Parker Williams\*\*

July 1, 2016

## *Abstract*

There has recently been considerable interest in cruising for curbside parking as a major contributor to traffic congestion in the downtown areas of major cities. The density of cars cruising for parking in the downtown area is related to the rate at which cars in transit in the downtown area start cruising for parking and the expected search time of a car that starts cruising for parking. This paper focuses on this expected search time. The literature has employed three different approaches to estimate expected cruising-for-parking time: direct measurement, inference based on the equilibrium condition that (for the marginal parker) the expected cost of curbside parking equals the expected cost of garage parking, and inference based on the observed occupancy rate of curbside parking and an assumed statistical relationship between expected cruising-for-parking time and the curbside parking occupancy rate. The last approach typically obtains estimates of expected cruising-for-parking times that are lower, and with high occupancy rates much lower, than those estimated using the other two approaches. This paper takes a step towards resolving this inconsistency by demonstrating, through computer simulation of cars cruising for parking around a circle in stochastic steady state, that an approximating assumption in the derived statistical relationship between expected cruising-for-parking time and the curbside parking occupancy rate leads to underestimation of average cruising-for-parking time, and at high occupancy rates very considerable underestimation.

\*Department of Economics, University of California, Riverside, Riverside CA 92521  
richard.arnott@ucr.edu

\*\*Department of Mathematics, University of California, Riverside, Riverside CA 92521  
parker.williams@ucr.edu

## **Cruising for Parking around a Circle<sup>1</sup>**

The pioneering work of Donald Shoup (2005) has stimulated considerable discussion of cruising for curbside parking as a major contributor to traffic congestion in the downtown areas of major cities. The literature contains estimates that the proportion of cars traveling on downtown city streets during the business day that are cruising for parking is 30% or even higher. Such estimates are not obtained from sidewalk observation since cars that are cruising for parking cannot be distinguished from cars in transit, but are instead obtained either by following a sample of cars or through model-based inference.

The density of cars cruising for parking in the downtown area is related to the rate at which cars in transit in the downtown area start cruising for parking and the expected search time of a car that searches for parking. This paper focuses on this expected search time. The literature has employed three different approaches to estimate expected cruising-for-parking time: direct measurement, inference based on the equilibrium condition that (for the marginal parker) the expected full price (which equals the money price plus the opportunity cost of time) of curbside parking equals that of garage parking,

---

<sup>1</sup> The authors would like to thank the U.S. Department of Transportation and Caltrans for their financial support of this research under a UCCONNECT grant (Department of Transportation Contract No. 65A0528), and Matthew Fitzgerald for excellent research assistance. Arnott would like to thank Tian Qiong for participating in earlier, preliminary joint analytical and simulation work on the topic, when, from April 2012 to April 2013, he was an academic visitor to UCR. In that work, he and the author treated the problem from the perspective of multi-server queuing theory. After Qiong returned to China, Derek Qu, then a graduate student in computer science at the University of California, Riverside, very ably continued the computer simulations for a short period of time. Arnott would also like to thank Amihai Glazer for having taken him to task in a seminar at the University of California, Irvine for the inconsistency in the approximations he employed in calculating expected cruising-for-parking times between Arnott and Rowse (1999) and Arnott and Rowse (2009).

and inference based on a derived statistical relationship between expected cruising-for-parking time and the curbside parking occupancy rate.

Most of the studies that employ direct measurement of cruising for parking are reviewed in Shoup (2005, Chapter 11). There are two reasons to be skeptical of the results. The first is that it is difficult to identify cars that are cruising for parking. One approach is to delineate a study area, follow random cars that enter the study area, identify them as searching for parking if they park curbside in the study area, and measure their cruising-for-parking times within the study area. This approach fails to identify cars that are indeed searching for parking in the study area but end up parking either outside the study area or in a parking garage. It also fails to identify when cars that park curbside in the study area initiate cruising for parking. The second reason to be skeptical of the results is that the study areas were not randomly selected, but were chosen instead because cruising for parking was perceived to be a severe problem there.

The second approach to estimate expected cruising-for-parking time was employed in Arnott and Rowse (2009, 2013). It is based on a model in which risk-neutral drivers choose between ubiquitous curbside and garage parking. Curbside parking is fully saturated, so that a car enters a curbside parking space immediately after it is vacated by the previous car parked there<sup>2</sup>, while garage parking can be obtained without search. In

---

<sup>2</sup> Arnott and Rowse assumed that each car cruising for parking experiences a vacant parking space according to a Poisson process with a rate equal to the turnover rate of parked cars per unit area divided by the stock of cars cruising for parking per unit area. When parking durations are negative exponentially distributed, the authors conjecture

equilibrium, drivers sort between curbside and garage parking such that their full prices are the same for "marginal" parkers -- those who are indifferent between curbside and garage parking. In most cities except those in the Netherlands, curbside parking is considerably cheaper than garage parking. An alternative statement of the parking equilibrium cost condition is then that, for a marginal parker, the curbside cruising-for-parking time cost equals the savings in the money cost from parking curbside. In the case of identical individuals, the expected cruising-for-parking time equals the savings in the money cost from curbside parking divided by the common value of time. When drivers differ according to visit length and the value of time, the calculation of expected cruising-for-parking time is more complicated. Consider an example with identical drivers in which the parking duration is one hour, the meter rate is<sup>3</sup> \$1.000/hr, the one-hour garage parking fee is \$10.00, and the value of time is \$30.00. Since the saving in the money cost of curbside parking is \$9.000, the equilibrium expected cruising-for-parking time is 0.3000 hrs.

The third approach considers a situation in which curbside parking is not saturated but is instead described by an expected occupancy rate. The central assumption is that the probability that each curbside parking space is occupied equals the expected occupancy rate, independent of history and of the occupancy status of neighboring curbside parking spaces. We term this the *binomial approximation*. It generates a geometric distribution for the number of parking spaces searched before finding a vacant space (including the

---

that the simulation model of this paper has this property in the limit as the expected occupancy rate approaches 100%.

<sup>3</sup> Throughout the paper, numbers are presented to the fourth significant digit.

vacant space). The number of parking spaces searched corresponds to the number of balls drawn from an urn with replacement (or with an infinitely large number of balls) before a "vacant" ball is drawn. Let  $q$  denote the probability that a ball is labeled "occupied", so that  $1 - q$  is the probability that a ball is labeled "vacant". The probability of finding the first vacant space on the first draw (i.e., the first parking space searched) is  $1 - q$ ; the probability of finding the first vacant space on the second draw is  $q(1 - q)$ , which is the probability that the first space searched is occupied times the probability that the second space searched is vacant, etc. The expected number of parking spaces searched before finding a vacant space (including the vacant space) is<sup>4</sup>  $1/(1 - q)$ . Thus, the expected numbers of parking spaces searched with curbside parking vacancy rates of 20%, 10%, 5%, and 1% are 5, 10, 20, and 100 respectively. Expected cruising-for-parking time can then be obtained by applying estimates of the average distance between parking spaces and of cruising-for-parking speed. As an example, assume that the distance between curbside parking spaces is 21.12 ft (1/250 mi) and that cruising-for-parking speed is 8.000 mph. Then the average cruising-for-parking time between parking spaces is 1/2000 hrs or 1.800 seconds. Shoup (2006) has proposed<sup>5</sup> that curbside meter rates be set to

---

<sup>4</sup> Let the expected number of draws before drawing a vacant ball (including the draw with the vacant ball) be  $S$ .  $S = (1)(1 - q) + (2)[q(1 - q)] + (3)[q^2(1 - q)] + \dots = (1 - q)\{1 + 2q + 3q^2 + \dots\}$ . Multiplying both sides by  $q$  yields  $qS = (1 - q)\{q + 2q^2 + 3q^3 + \dots\}$ . Subtracting  $qS$  from  $S$  yields  $(1 - q)S = (1 - q)\{1 + q + q^2 + \dots\}$ . Since the value of the infinite sum in the curly brackets is  $1/(1 - q)$ ,  $S = 1/(1 - q)$ .

The variance, (Pearson's) skewness, and (Pearson's) kurtosis of the geometric distribution are  $q/(1 - q)^2$ ,  $(1 + q)/q^{1/2}$ , and  $6 + (1 - q)^2/q$  (Wikipedia: Geometric distribution).

<sup>5</sup> Shoup's work has stimulated a number of downtown parking experiments. The best known is SFpark. The City of San Francisco has been adjusting curbside meter rates by block and by time of day to achieve a target curbside parking occupancy rate. Shoup (2006) originally proposed a target curbside parking occupancy rate of 85%. The City has been adjusting this rate by block and time of day to achieve what it judges to be optimal

achieve a curbside parking occupancy rate of 85%. Under the binomial approximation and the above parameter assumptions, applying the Shoup rule would generate expected cruising-for-parking time of only 12.00 seconds ( $1.800 \times 1/(1 - 0.8500)$ ).

All the recent papers that derive the expected number of parking spaces searched before finding a vacant space from the occupancy rate, including Arnott and Rowse (1999), Anderson and de Palma (2004), Geroliminis (2015), and Du and Gong (2016) have employed the binomial approximation.

Levy, Martens, and Benenson (2012) simulates a situation in which drivers search for parking in a residential neighborhood on their return from work, and in which therefore the occupancy rate increases as the evening proceeds. They compare the average realized number of parking spaces searched in their simulation model, PARKAGENT, as a function of the realized occupancy rate, to the expected number of parking spaces searched under the binomial approximation, as a function of the expected occupancy rate. When the realized occupancy rate in their simulation model is above 85%, the simulated average number of parking spaces searched is considerably higher than the expected number under the binomial approximation with that occupancy rate. Though their analysis is not steady state, and though their conclusions rest on the soundness of their simulation model, the discrepancy between their simulated numbers and those obtained under the binomial approximation is sufficiently large to cast doubt on the accuracy of the binomial approximation.

---

rates. They vary substantially but the average is considerably lower than 85% (Pierce and Shoup, 2013).

There are further reasons to doubt the accuracy of the binomial approximation.

1. The binomial approximation takes the occupancy rate as being constant over time. But even in a stochastic steady state, in which entry occurs according to a time-invariant Poisson process, the local occupancy rate fluctuates because of the stochasticity of demand at the local level. Adapting the binomial approximation to treat this complication only would lead to a higher expected number of parking spaces searched<sup>6</sup>.
2. The binomial approximation is based on the assumption that the occupancy probabilities of adjacent parking spaces are statistically independent. But common sense and experience suggest that, to the contrary, they are both spatially and temporally correlated. Suppose, for example, that, purely by chance, at a point in time parking on a block becomes fully occupied. This will have a ripple effect in space-time. Because that block is fully occupied, cars that have destinations on that block will have to continue their search to neighboring blocks, which will increase the expected occupancy rates on those blocks later in time.
3. When there are fluctuations in the entry rate, the expected number of parking spaces searched from the perspective of the user is different from that of an external observer, since users disproportionately travel at times when demand is high.
4. There are other, more practical reasons why the binomial approximation results in underestimation of expected cruising-for-parking time. First, there is history dependence in local demand, which amplifies stochastic fluctuations in the local occupancy rate. If

---

<sup>6</sup> The expected number of parking spaces searched before finding a vacant parking space would equal the reciprocal of the harmonic mean of the vacancy rate, which is greater than the reciprocal of the average vacancy rate.

the demand for parking on a block is higher than average, it may be due to a special event that will result in demand being higher than average over a period of time. Second, traffic is not in stochastic steady state but rather has systematic fluctuations over any period time, which increase the expected number of parking spaces searched. Third, practically an average curbside occupancy rate is measured over a period of time and over an area of space, with variation over time and over space.

5. The expected cruising-for-parking times generated by the binomial approximation square neither with experience nor with policy discussion. In particular, the expected cruising-for-parking times under the binomial approximation seem consistently too low. Experience suggests that on blocks where the average occupancy rate is, say, 80%, it may quite frequently be difficult to find a parking space on that block. In contrast, under the binomial approximation, the expected number of parking spaces searched before finding a vacant space is only five. Policy discussion in Shoup<sup>7</sup> (2005) and elsewhere indicates that cruising for parking is perceived to be a serious problem in downtown areas, and yet the expected cruising-for-parking times under the binomial assumption are modest, except as the occupancy rate approaches one.

In this paper, we consider a stylized model that abstracts from the practical complications considered under 4. above. It is depicted in Figure 1. Space is the circumference of a circle of finite length, which we refer to as "the track". Parking spaces are evenly spaced

---

<sup>7</sup> Table 11-5 in Shoup (2005) reports that average cruising-for-parking time over the 16 studies of cruising for parking that he located was 8.1 minutes, about 500 seconds. Our calibration of the numerical simulations, presented below, implies that it takes about 1.800 seconds to travel from one parking space to the next. Applying this figure implies that the expected number of parking spaces searched is 278. Under the binomial approximation, this corresponds to an occupancy rate of 99.64%.



points around the track. The arrival of drivers is generated by a time-independent and space-independent Poisson process. Each driver cruises clockwise around the track at an exogenous speed and takes the first vacant parking space she encounters, parks there for a period of time that is determined by a draw from a time- and space-independent probability distribution (which may or may not be negative exponential), and then exits the system. The expected curbside parking occupancy rate is calculated as the expected total time that cars are parked around the track per unit time divided by the maximum total time that cars can be parked around the track per unit time. The expected total time that cars are parked per unit time equals the Poisson arrival rate of drivers times the expected parking duration, and the maximum total time per unit time simply equals the number of parking spaces round the circle. The natural measure of a unit of time in the model, which we adopt, is the time it takes to travel from one parking space to the next.

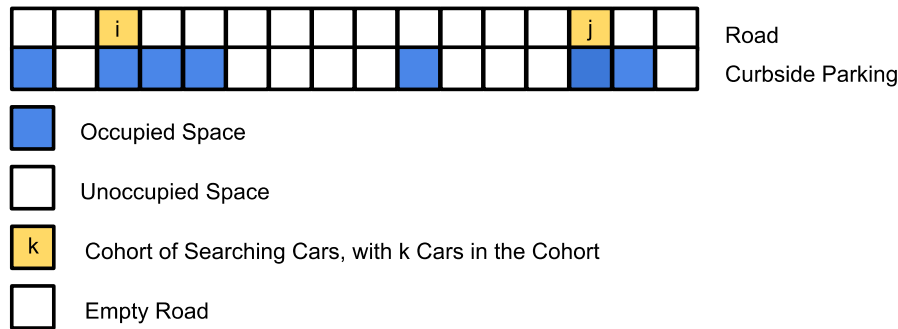


Figure 1: Depiction of a Sample State of the Simulation Model

Notes: Cars travel from left to right, one parking space per time unit. The space is in fact circular, so that a car that exits on the right simultaneously enters on the left.

The number in a yellow box is the number of cars that pass by the space during the current time unit. Together these cars are termed a cohort.

Our model incorporates three important simplifying assumptions. First, it is spatially symmetric (except that parking spaces are points). Second, its equilibrium describes a (stochastic) steady state with exogenous, time- and space-independent Poisson (and therefore history-independent) processes generating entry to the track and exit from a parking space, conditional on entry to it<sup>8</sup>. Third, as modeled, a driver's parking search "strategy" is trivial; she starts searching for parking as soon as she enters, and keeps on driving in the same direction until she encounters a vacant parking space. In reality, even one-dimensional parking search is more complicated than this. With an exact destination, a rational driver does not start cruising for parking until a certain distance from her destination (Arnott and Rowse, 1999; Arnott, 2014); also, if parking search is not unidirectional, a driver may decide to turn around and backtrack. Parking search strategy in two dimensions is considerably more complex.

We chose our assumptions to achieve a balance between comprehensibility, accuracy, and realism. We could have made the model even simpler. First, we could have described space more simply as a spatially ordered set of discrete parking spaces, and time as discrete, in which case cruising for parking would be modeled as a multi-server queuing system, with unserved users moving from one server to the next between time periods<sup>9</sup>. We decided not to do so since the discretization of time and space causes

---

<sup>8</sup> The exit rate from an individual parking space, conditional on entry to it, is generated by a time- and space-independent Poisson process. However, the process of entry into parking spaces depends on the history of the system.

<sup>9</sup> Indeed, this is how Qiong and Arnott, and Qu and Arnott, modeled the problem in their work.

artificial difficulties in the between period modeling of events<sup>10</sup>. Second, we could have ignored that servers are spatially ordered, or even have collapsed the multi-server queue into a single-server queuing, but both simplifications might affect the qualitative properties of equilibrium.

The central issue that the paper addresses is whether, in stochastic steady state, the interaction between the time- and space-independent Poisson entry process, and the time- and space-independent process of exit from a parking space, conditional on entry to it, and cruising for parking generates a time- and space-independent vacancy generation process. If it does, the binomial approximation is sound. But the process of curbside parking search may make entry into a particular parking space dependent on the state of the system, which makes the unconditional exit process history dependent, which in turn may cause the distribution of the expected number of curbside parking spaces searched not to be geometric.

We originally explored exact analytical solution of the model, but had no substantive success. In the next section, we cast our model in the context of Markov chains, and then explain the difficulties in exact analytical solution. We then had the choice between

---

<sup>10</sup> For example, the modeler needs to make the choice as to which happens first between time periods (between the "current" period and the "next" period), the vacation of parking spaces that occurred during the current period or the assignment of drivers who were waiting in a queue at the beginning of the current period. The modeler also needs to make the choice as to whether entrants to the system between periods are treated in the same way as drivers who were waiting in a queue at the beginning of the period. Since these sequencing decisions are not present in a continuous time model, they are an artifact of discretization. They may affect the qualitative properties of equilibrium, and even if they do not they invite confusion.

investigating analytical solution under simplifying assumptions or proceeding to simulation. We decided to employ simulation since, without an analytical solution of the proper model, we would have no way of judging how accurate were the analytical solutions under the approximations.

*Our basic finding is that the binomial approximation is a bad one.* The base case parameter values are recorded in Table 1 below. There are 100 parking spaces around the track. The Poisson entry rate to the system is  $1/30$  per time unit, the distribution of parking times (stay lengths) is negative exponential with mean 2000 (or put alternatively the Poisson exit rate from an occupied parking space is  $1/2000$  per time unit). The expected total parking duration per unit of time equals the entry rate times expected parking duration, which equals 66.67. The maximum parking duration per unit of time is simply equal to the number of parking spaces, 100. Thus, the expected occupancy rate is  $2/3$ . A unit of time is the period it takes to travel from one parking space to the next. Accordingly, the expected time it takes for an entering car to reach the first parking space is 0.5000 time units, the second parking space is 1.500, ----. Thus, expected cruising-for-parking time equals the expected number of curbside parking spaces searched (including the last, successful search) minus 0.500. Under the binomial approximation, with an occupancy rate of  $2/3$ , the expected number of curbside parking spaces searched is 3.000 (see fn. 2), which corresponds to an expected cruising-for-parking time of 2.500. In contrast, for the central base case run the simulated mean cruising-for-parking time is 3.602, so that the ratio of the simulated mean cruising-for-parking time to the expected value obtained under the binomial approximation is 1.441. The variance of the cruising-

for-parking time under the binomial approximation is 6.000 (see fn. 2) and in the simulation is 23.92, for a ratio of 3.987.

Number of parking spaces	P	100
Distance between parking spaces		1 (normalized distance unit)
Travel time between parking spaces		1 (normalized time unit)
Poisson entry rate to track	$\mu$	1/30 (per normalized time unit)
Poisson exit rate from occupied parking space	$\lambda$	1/2000 (per normalized time unit)
(Implied) expected occupancy rate	Q	2/3

Table 1: Base Case Parameter Values

Notes: We have taken a normalized distance unit to be 1/250 ml (21.12 ft), and a normalized time unit to be 1/2000 hr (1.800 seconds)

As the expected occupancy rate increases (generated by a proportional increase in the entry rate), so too does the ratio of the simulated mean cruising-for-parking time to the expected value obtained under the binomial approximation. With an expected occupancy rate of 11/12, the expected cruising-for-parking time under the binomial assumption is 11.50 and the corresponding simulated mean 97.62, giving a ratio of 8.489. The corresponding variances are 132.0 and 86560, for a ratio of 651.5. *Thus, the binomial approximation gets worse as the expected occupancy rate increases.*

Since we were unsuccessful in obtaining analytical results, we can only go so far in explaining why the probability distribution of simulated cruising-for-parking differs from the distribution obtained under the binomial approximation. Consequently, the paper focuses on describing the simulation results and the ways in which the simulated results differ from those obtained under the binomial approximation, from a variety of statistical perspectives.

Section 2 places the model in the context of queuing and Markov process theory. Section 3 presents the simulation model, section 3.1 presenting the simulation algorithm and section 3.2 the model's parameterization. Section 4 records the quantitative results of the central base-case simulation and compares them to the results obtained under the binomial approximation, using a variety of statistical approaches. Section 5 undertakes a variety of comparative stochastic steady-state exercises, examining how the simulated probability distributions of search times change with changes in exogenous parameters. Section 6 discusses directions for future research, and presents some concluding remarks, including comments on the policy insights from the research.

## 2. *Placing the Model in the Context of Queuing and Markov Process Theory*

This section draws heavily on Gross, Shortle, Thompson, and Harris (2008).

### 2.1 *Queuing Theory*

There is a standard notation used in the queuing theory literature. A queuing process is described by a series of symbols and slashes, such as  $A/B/X/Y/Z$ , where A indicates the arrival-time distribution, B the probability distribution of service time, X the number of parallel service channels, Y the restriction of system capacity, and Z the queue discipline. In our parking model: i) since arrivals at the track are generated by a time-independent Poisson process, the arrival-time distribution is negative exponential with the exponent equal to the Poisson arrival rate, so that, according to queuing theory notation,  $A = M$  (for

Markovian); ii) since each parking space is a separate service channel,  $X = P$ , where  $P$  is the number of parking spaces round the track; iii) in the base case, the probability distribution of service time is negative exponential with mean equal to the expected parking stay time or duration, so that  $B = M$ ; iv) since there is no restriction on capacity,  $Y = \infty$ ; and v) the queue discipline is not a conventional one<sup>11</sup>, so that we set  $Z = ?$ . Thus, the parking model is similar to a  $M/M/P/\infty/?$  queuing process. In the standard multi-server queuing process, the servers are assumed to be parallel, by which is meant there is a single queue for all the servers together. However, the parking model here has a different multi-server queuing process<sup>12</sup>. Thus, though our parking model can be viewed as a queuing model, it is not a standard queuing model, and indeed we have found no queuing models in the literature that describe it.

## 2.2 *Markov Process Theory*

A Markov process is a memoryless stochastic process, in the sense that the stochastic evolution of the system after time  $t$  is completely determined by the state of the system at time  $t$ . Markov processes are classified according to the index set of the process (whether

---

<sup>11</sup> Among cars that are currently between a particular parking space and its clockwise neighbor, the car that is closest to the clockwise neighbor parking space will have priority in parking; the queue discipline is FCFS in this respect. However, if a car enters the system between that parking space and its clockwise neighbor, its queuing priority is determined by the location where it enters relative to the location of other cars in the clockwise neighbor's queue; since the location of the entering car is random, the queue discipline is in this respect random access.

<sup>12</sup> From the perspective of a driver cruising for parking, the servers are moving in a counter-clockwise direction and the driver is served by the first vacant server that passes him by. From the perspective of a server, the cars cruising for parking are moving in a clockwise direction. If the server is full, the cars cruising for parking just pass on by, while, if the server is vacant, the first car to reach the server takes the vacant parking space.

time is discrete or continuous) and the nature of the state space of the process (whether there is a finite or infinite number of states of the system). When the parking stay time is a negative exponential distribution, our parking model describes a continuous-time, infinite Markov process. Since there are no absorbing states, we conjecture but have not proved that our model has a limiting distribution<sup>13</sup> (hence, the model has a stationary distribution and the Markov process is ergodic). We are interested in the distribution of curbside parking search times associated with the limiting distribution of states of the system. Since our simulations are finite, we do not observe the limiting distribution but rather a sample of the probability distribution of states conditional on the starting state.

The dimension of the state space in our model is very high. When the parking stay time is a negative exponential distribution, as it is in our base case, the state of the system is described by the occupied/vacant status of the spatially ordered parking spaces (which entails  $2^P$  permutations) and the positions of all the cars that are cruising for parking (which may be infinite), each of which is described over the continuous space of the track.

As we indicated earlier, we could reduce the number of states of the system by discretizing space and time. A state would then be indexed by the spatially ordered parking spaces, and then, for each parking space, by its occupancy status and the number of cars queued at it. The state transition from one period to the next would then be

---

<sup>13</sup> Let  $s$  index the possible states. Starting with state  $s_0$ , let  $p_{s,s_0}(t)$  denote the probability that the system will be in state  $s$  at time  $t$ . If in the limit as time approaches infinity, these probabilities are independent of the initial state of the system, these probabilities describe the limiting distribution of the system.

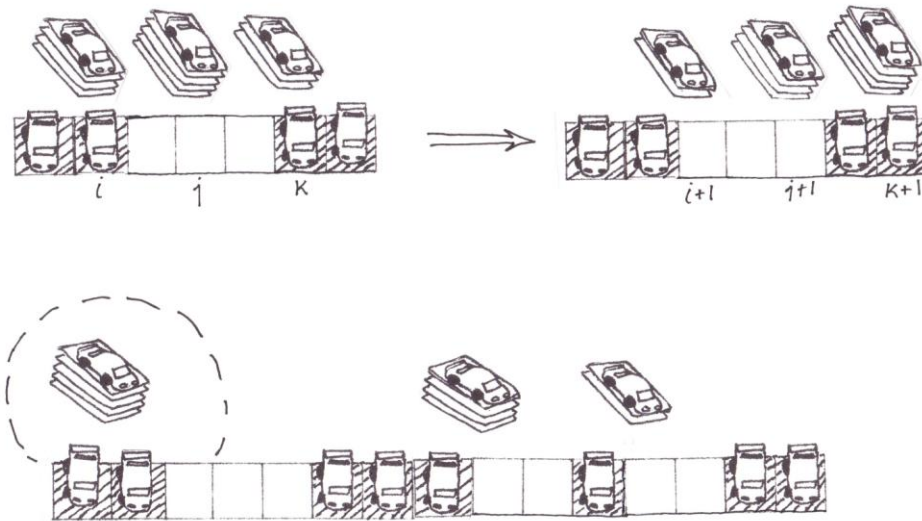


determined by the number of entries at each parking space, change in the occupancy status of each parking space, and the movement of cars queued at each space to its rightmost neighbor (representing the cruising of cars). The limiting distribution of the probabilities of the various states and of the cruising-for-parking search times would then be calculated from the corresponding transition matrix. We have not done this in the paper since the discretization applies approximations that might alter the qualitative properties of the limiting distributions.

The following cartoon has three parts. The top part illustrates one of many possible state transitions that can occur moving forward in time. The middle part and the bottom part display the very large number of different state transitions that may occur at a particular parking space over a single time step. The dashed circle in the middle part of the cartoon indicates the particular parking space whose possible set of state transitions are illustrated in the bottom part. The left side of the bottom panel shows the situation at time  $t$ . There are  $k$  cruising for parking between parking spaces  $i$  and  $i + 1$  (and which therefore pass space  $i + 1$  between  $t$  and  $t + 1$ ), and parking space  $i + 1$  may be occupied or vacant.

The right side of the bottom panel shows four qualitatively different states at time  $t + 1$ . In the first, all  $k$  searching cars find parking space  $i + 1$  occupied, and no entering cars pass by space  $i + 1$  between  $t$  and  $t + 1$ . Thus, at time  $t + 1$ , there are  $k$  cars cruising for parking between spaces  $i + 1$  and  $i + 2$ . In the second, one of the  $k$  cars (not necessarily the first) finds parking space  $i + 1$  vacant and occupies it at least until time  $t + 1$ , and no entering cars pass by space  $i + 1$  between  $t$  and  $t + 1$ , so that at time  $t + 1$  there are  $k - 1$

cars cruising for parking between spaces  $i + 1$  and  $i + 2$ . In the third, all  $k$  searching cars find parking space  $i + 1$  occupied, and  $p$  (one, two, three, ---) entering cars pass by space  $i + 1$  between  $t$  and  $t + 1$ , all finding it occupied, so that at time  $t + 1$  there are  $k + p$  cruising cars for parking between spaces  $i + 1$  and  $i + 2$ . In the fourth, one of the  $k + p$  cars that pass space  $i + 1$  finds it vacant and occupies it at least



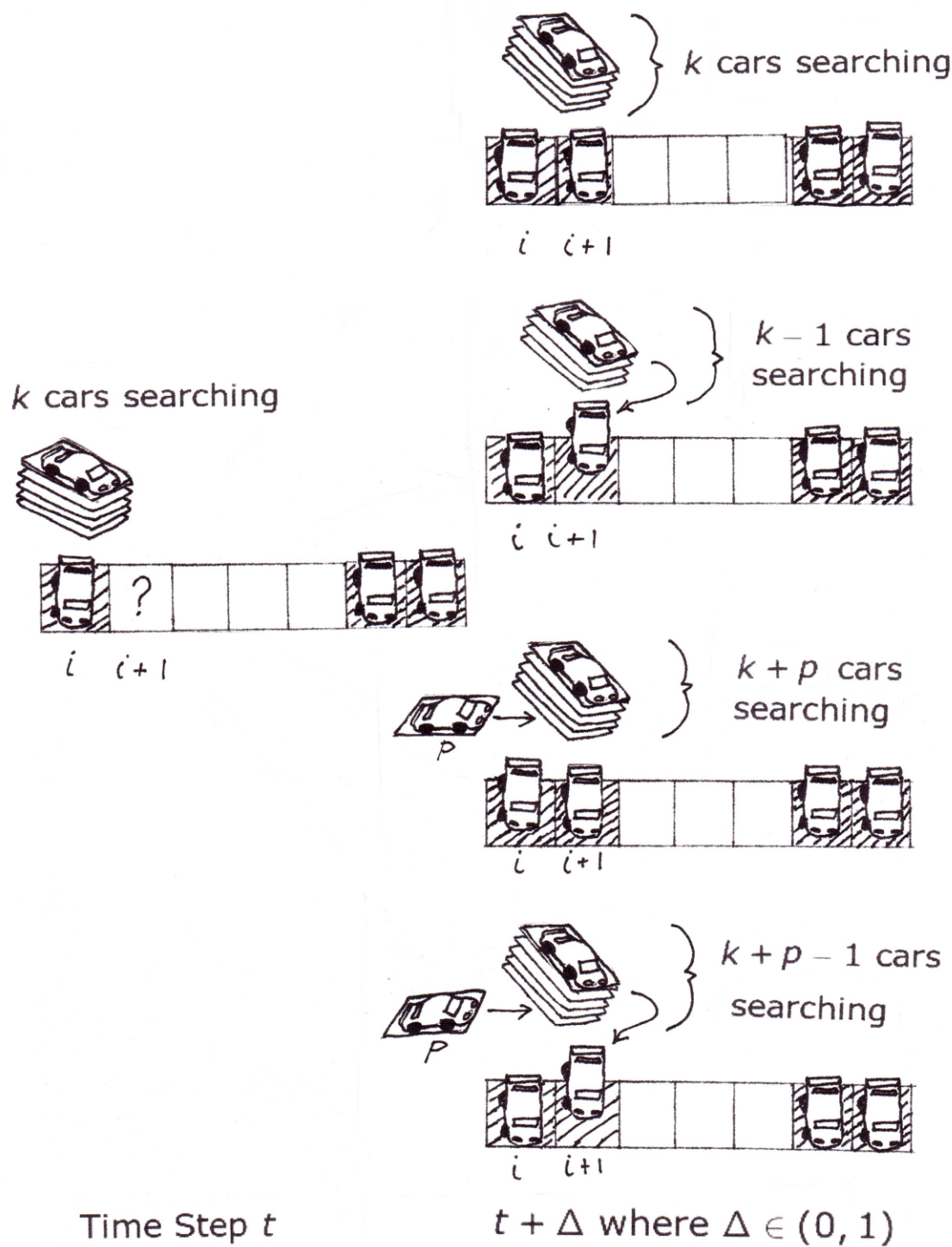


Figure 2: Cartoon of Possible State Transitions from One Time Unit to the next

until time  $t + 1$ , so that at time  $t + 1$  there are  $k + p - 1$  cars cruising for parking between spaces  $i + 1$  and  $i + 2$ . The above enumeration of state transitions is not exhaustive, since a parking space may become occupied more than once during a time period.

### 3. *The Simulation Algorithm and Parameterization of the Model*

Section 3.1 describes the solution algorithm and section 3.2 the parameterization of the model.

#### 3.1 *The Solution Algorithm*

The algorithm simulates a circular street (the "track") of fixed length with a fixed number of evenly-spaced parking spaces, cars arriving at a time- and location-independent Poisson rate, cruising for a vacant parking space at an exogenous speed, taking the first vacant space that becomes available, and then exiting that parking space and leaving the track at a different time- and location-independent Poisson rate.

Since it was written to conserve on storage space, the actual algorithm is quite complex. To make it easier to understand, we proceed in three steps. The first step presents a cartoon of the algorithm, treating time and space as being discrete, though in the actual algorithm both are treated as continuous. The second step presents pseudocode for the algorithm, and the third step is the actual python code<sup>14</sup>, which is presented in Appendix

---

<sup>14</sup> Floating point values in python are stored in double precision floating point format and follow the IEEE 754 standard, which is documented here [https://en.wikipedia.org/wiki/Double-precision\\_floating-point\\_format](https://en.wikipedia.org/wiki/Double-precision_floating-point_format)

1, along with a hyperlink to the source code,

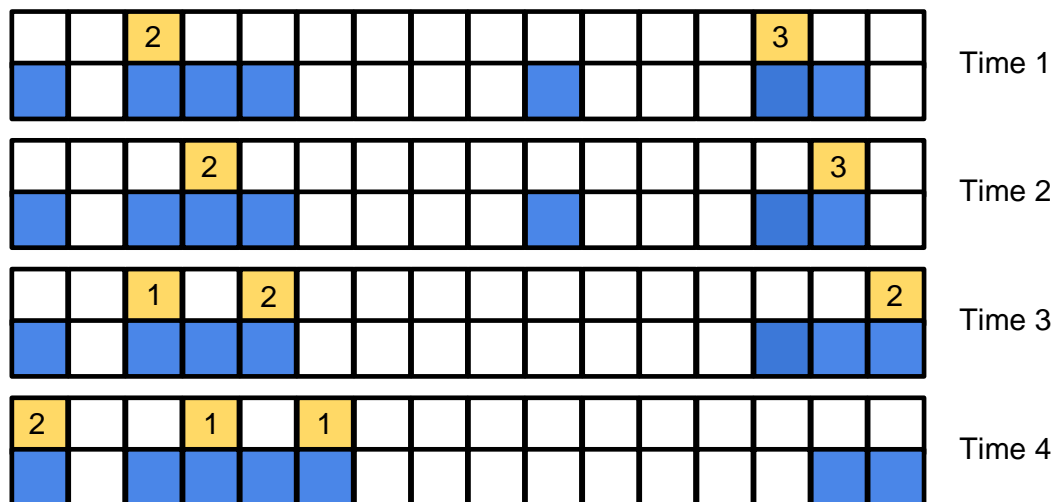
<http://math.ucr.edu/~parker/CruisingForParking/>.

- *The cartoon*

Figure 1 displayed a cartoon in the spirit of the algorithm, imagining a pair of lanes divided into boxes, the top lane being the road and the bottom being the curbside. A white box in the bottom lane is a vacant parking space, and a blue box in the bottom lane is an occupied parking space. A cohort is a group of cars that are queued at a particular parking space. A white box in the top lane indicates that there are no cars queued at the corresponding parking space. A yellow box in the top lane indicates that there are cars queued at the corresponding parking space, with the number of cars queued indicated by the number in the yellow box. Figure 3 illustrates how the state of the system (the pattern of cruising cars and occupied/vacant spaces) might evolve over four time units. The snapshots are taken at the end of each time unit. Between time units:

- Each yellow box advances to the right one unit space.
- Some of the parked cars may exit their parking spaces. Each of the spaces that was occupied becomes vacant (switching from blue to white).
- Having advanced, if the cohort is above an occupied space, the cars in the cohort queue at that space for the unit of time. If, alternatively, the cohort is above a vacant space, one of the cars in the cohort takes the vacant space, and the remaining cars in the cohort queue at that space for a unit of time.
- Some cars may enter the system. A car that enters the system at a vacant parking space and is the only car to enter at that vacant space occupies that

parking space. If more than one car enters the system at a vacant parking space, one of the cars occupies that parking space and the rest become a cohort of cars at that space (so that the corresponding road box becomes a yellow box with a number equal to the size of the rest of the cohort). A car that enters the system at an occupied space either becomes a cohort of one (so that space becomes a yellow box with 1 in it) or joins the existing cohort at that space (so that the number of cars in the cohort increases by one).



Notes: The space is in fact circular, so that a car that exits on the right simultaneously enters on the left.

The number in a yellow box is the number of cars in the cohort that are queued at the corresponding occupied parking space during the time unit.

During time unit 1, there are 7 occupied spaces on the track, which has 16 spaces, with two cars in a cohort queued at occupied space 3 and three cars in a cohort at occupied space 14. Between times 1 and 2, both of these cohorts advance, no cars exit, and no cars enter, so that during time unit 2, there are again 7 occupied spaces on the track, with two cars in a cohort queued at space 4 and three cars in a cohort at space 15. Between times 2 and 3, both of these cohorts advance, one car exits at space 10, and one car enters at occupied space 3, so that during time unit 3, there are 7 occupied spaces, with one car queued at space 3, two cars queued at space 5, and two cars queued at space 16 (between times 2 and 3 the cohort at space 15 advanced to vacant space 16 and one car

in the cohort occupied the vacant space, reducing the number of cars in the cohort to 2). Between times 3 and 4, all three cohorts advance one space (with the cohort that) was at space 16 advancing to space 1), one car exits at space 14, and no cars enter, so that during time unit 4, there are 7 occupied spaces, with two cars queued at space 1, one car queued at space 4, one car queued at space 6 (between times 3 and 4 the cohort at space 5 advanced to vacant space 6, one of which occupied the vacant space, reducing the number of cars in the cohort to 1).

Note that the evolution of the system from one period to the next depends on the sequencing of events between periods, which is an awkward artifact of the discretization of time. In contrast, in the actual program the location of parking spaces is discrete, but entry can occur anywhere on the circle, and time is continuous. The resolution of the state of the system is done one time step at a time.

- *The pseudocode*

1 Create a list of parking space objects with a linear order, occupancy status and vacancy time, initialized with respective parameters for the model, called "spaces".

2  $T = 1$

3 Make sure the Poisson process governing the arrival times and locations has generated enough arrivals that go beyond the next time step. Then add car objects to a list called "cars" with these arrival times and locations.

4 Resolve all events that occur between time  $T$  and  $T-1$  using the lists cars and spaces of objects in "continuous" time, comparing times at which a car in the cars list passes by a space and at which time a space is vacant. Cars cruise for parking by travelling in one direction around a circle taking the first vacant parking space they encounter. Record (T-Arrival time) for any car that finds parking.

5 Update all vacancy flags, and generate new vacancy times for spaces that have become occupied.

6 Record ambient statistics at time  $T$ , including occupancy rate and number of cars currently searching.

7 Should the desired number of observations not be met, set  $T = T + 1$  and go to 3.

### 3.2 *Parameterization of the Model*

The parameters of the base case simulation model were reported in Table 1. For the initial conditions of each simulation, all parking spaces are initialized as occupied, with a time until available that is exponentially distributed with the mean stay length of 2000 time units (the same as for the entering cars). We start recording simulation results when 10,000 time units (5 hours in unnormalized time units) have passed since the start time. We made this choice early in the research, on the assumption that, after five hours of run time, the distribution of cruising-for-parking times should be little affected by the assumed starting point. In most of the simulations, we ran  $10^6$  cars through the system, which corresponds to approximately  $10^6 \times 30 \times 1.8/3600 = 15,000$  hrs of traffic. This might seem like overkill, but as we shall see, it was not.

We have undertaken a large number of simulation runs. To avoid a tedious cataloguing of results, we have decided to proceed by examining in the next section the central base case simulation from several different statistical perspectives, and then in the subsequent section by presenting the effects of varying one parameter at a time using a more limited set of statistics.

## 4. *The Central Base Case Simulation: Different Statistical Perspectives*

We start in Table 2 by comparing the moments of the distribution of cruising-for-parking times for the central base case simulation to the theoretical distribution based on the



binomial approximation. All the numbers presented are in normalized time units. The simulated results are based on a simulation length of  $10^6$  cars, which corresponds to 15,000 hours, with the results from the first five hours dropped in order to ensure that the recorded results are little affected by the assumed initial conditions.

	Mean	Variance	Pearson Skewness	Fisher Kurtosis
Simulated	3.602	23.92	4.031	42.20
Binomial Approximation	2.500	6.000	2.041	6.167

Table 2: Moments of the Probability Distribution of the Cruising-for-Parking Time for the Central Base Case in Normalized Time Units

Notes: Pearson skewness equals the third moment of the distribution, normalized by dividing by the second moment of the distribution to the  $3/2$  power. Fisher kurtosis equals the fourth moment of the distribution, normalized by dividing by the second moment of the distribution squared.

Thus, as was noted earlier, the mean from the simulations is 1.441 times as large as the mean calculated under the binomial assumption, and the variance, Pearson skewness, and Fisher kurtosis are all larger too, indicating that the simulated distribution has a fatter tail.

We could now compare the cdf's for the simulated distribution and the theoretical distribution. We have chosen not to do so since the probability mass is so concentrated for just a few parking spaces searched. The cdf's show that the simulated distribution first-order stochastically dominates the theoretical distribution, consistent with the results in Table 2. We found a "ratio plot" of the number of occupied parking spaces searched before finding a vacant space (excluding the vacant space) to be a more informative way of comparing the two distributions. A ratio plot plots a ratio against the number of

curbside parking spaces searched. For  $n$  curbside parking spaces searched, the numerator in the ratio is the realized proportion of cars that encountered  $n$  occupied parking spaces before finding a vacant parking space, while the denominator is the corresponding expected proportion based on the binomial approximation<sup>15</sup>. Figure 4 displays the ratio plot for the central base case simulation. It gives a vivid representation of how much fatter the tail is in the distribution obtained from the simulation than in the theoretical distribution. Consider for example  $n = 17$ . Under the binomial approximation, the probability that  $n = 17$  is  $(2/3)^{17}(1/3) = 3.383 \times 10^{-4}$  implying that, out of  $10^6$  cars, the expected number of cars to encounter 17 occupied parking spaces before finding a vacant one is 338.3. The number obtained from the simulation is about 10 times as large. With  $n = 25$ , the corresponding numbers are 13.20 and a number that is about 100 times as large.

---

<sup>15</sup> As was noted earlier, where  $q$  is the expected occupancy rate, the expected proportion of cars that encounter 0 occupied parking spaces before finding a vacant space is  $(1 - q)$ , 1 space is  $q(1 - q)$ , 2 spaces is  $q^2(1 - q)$ , etc., which with an occupancy rate of  $1/3$  gives the proportions  $1/3$  for  $n = 0$ ,  $2/9$  for  $n = 1$ ,  $4/27$  for  $n = 2$ , etc.

Note the distinction between the number of occupied curbside parking spaces searched before finding a vacant space, and the number of curbside parking spaces searched before finding a vacant space (including the vacant space). They differ by one. We have shown that with a curbside occupancy rate of  $2/3$ , the mean of the latter distribution is  $3(1/(1 - q))$  while that of the former is  $2(1/(1 - q) - 1)$ .

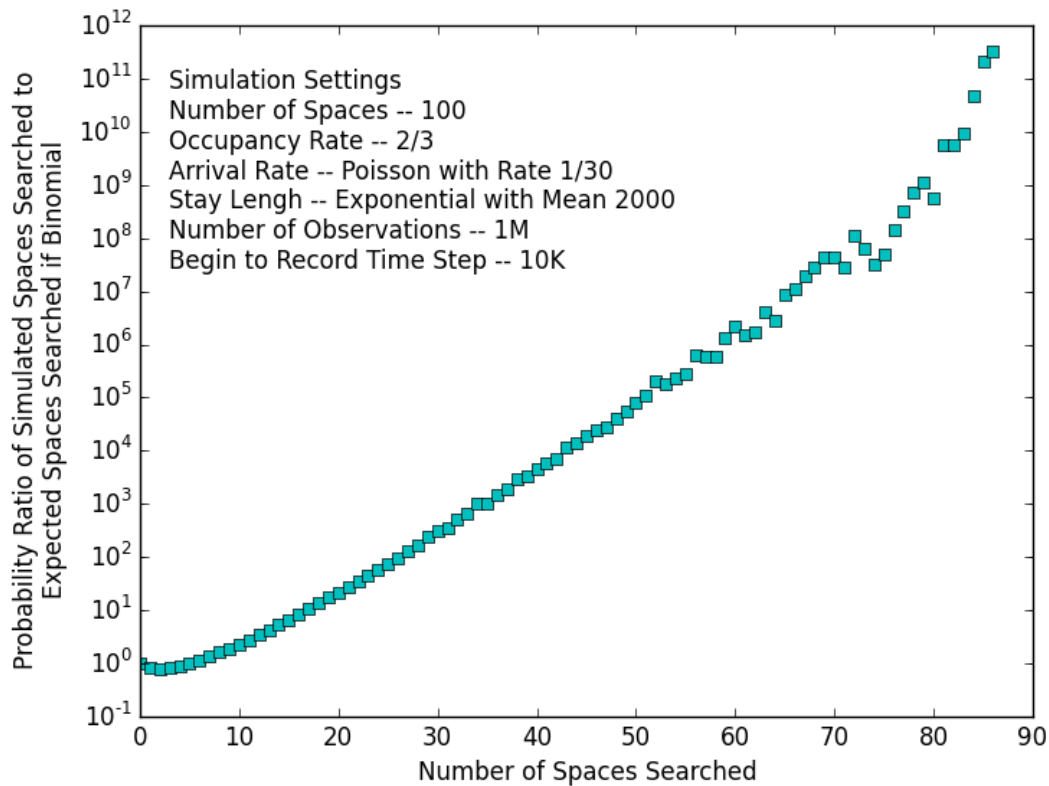


Figure 4: Ratio Plot for Occupancy Rate 2/3

Notes: The number on the y-axis is the ratio of the probability for the simulation run to the corresponding probability under the binomial approximation.

Whatever the probability distribution of the number of occupied parking spaces searched before finding a vacant space, the probability of  $n = 0$  equals the vacancy rate. When the expected occupancy rate is  $2/3$ , so that the expected vacancy rate is  $1/3$ , a car that has just entered the track should find the first parking space to be vacant with probability  $1/3$ . All of our simulation runs are consistent with this observation.

What is causing the simulated density function for cruising-for-parking time to have tails that are so much fatter than the tails of the density function implied by the binomial

approximation? Unfortunately, as explained earlier, even though our model specifies the stochastic process generating occupancies and vacancies, we have been unable to solve analytically for the implied stationary distribution of cruising-for-parking times. We can, however, identify several possible effects, each of which tells part of the story.

1. Intuitively, the directed nature of search for a vacant parking space may lead to spatial autocorrelation, which we term the *bunching effect*. One measure of this effect is the expected number of occupied parking spaces in a row before a vacant space is encountered<sup>16</sup>. To illustrate how this effect works, compare the number of occupied parking spaces searched before finding a vacant parking space under the repeated pattern VOOVOOVOO -- with that under VVOOOOVVOOOO ---, both of which have an occupancy rate of  $2/3$ . In the former situation, the distribution of the number of occupied parking spaces searched before finding a vacant space is 0 with probability  $1/3$ , 1 with probability  $1/3$ , and 2 with probability  $1/3$ , for an expected value of 1. In the latter situation, the corresponding probabilities are 0 with probability  $1/3$ , 1 with probability  $1/6$ , 2 with probability  $1/6$ , 3 with probability  $1/6$ , and 4 with probability  $1/6$ , for an expected value of  $10/6$ . Another measure of this effect is the expected number of occupied spaces in a bunch. In the former example, this number is 2; in the latter example, it is 4; and under the binomial approximation, it is 3.<sup>17</sup> In the central base case

---

<sup>16</sup> Note that this is a measure of bunching from the perspective of a stationary observer, which is different from a measure from the perspective of a searching car. To illustrate the difference, suppose that bunches moved at the same speed as cars. Then an entering car that found the first parking space it encountered to be occupied would continue to encounter occupied spaces.

<sup>17</sup> Start with an occupied space whose leftmost neighbor is vacant. The probability that there is one occupied space in the bunch is the probability that the space after the

simulation, the mean size of a bunch of occupied spaces is 3.887 (and the variance is 19.24), which indicates that bunches are significantly more clustered than under the binomial approximation.

To get an idea of the importance of the bunching effect, assume for the sake of argument that the mean number of parking spaces searched (including the vacant space) equals the expected number of occupied parking spaces in a bunch, which holds under the binomial approximation. Then the expected cruising-for-parking time in the central base case simulation would be  $3.887 - 0.5000 = 3.387$ , and the bunching effect would explain a proportion  $(3.387 - 2.500)/(3.602 - 2.500) = 0.8049$  of the difference between the mean cruising-for-parking time between the central base case simulation and that obtained under the Poisson assumption. While the validity of the assumption is certainly open to question, this back-of-the-envelope calculation does indicate the potential importance of the bunching effect in explaining the discrepancies between the simulation results and those obtained under the binomial approximation.

Above we have documented the bunching effect for the central base case simulation, and taken a first pass at gauging its importance in explaining the discrepancy between the simulated mean cruising-for-parking time and that generated under the Poisson approximation. But we have not investigated how the bunching effect differs across

---

occupied space is vacant, which is  $1/3$ ; the probability that there are two occupied spaces in the bunch is  $2/3(1/3)$ , ---. Thus, the expected value is  $1/3 + 2(2/9) + 3(4/27) + 4(8/81)$  ---, which can be shown to equal 3.

simulation runs. Nor have we developed theory that explains either how the bunching effect occurs or how it affects the discrepancy<sup>18</sup>.

2. Intuitively, if the track has an infinite number of parking spaces around it, by some law of large numbers the occupancy rate should remain at  $2/3$ . With a finite track, however, the occupancy rate is stochastic. Suppose, for example, that the occupancy rate is  $q_0 = 1/2$  half the time and  $q_1 = 5/6$  the other half. If the stochastic process otherwise satisfies the binomial approximation, then the expected number of occupied parking spaces encountered before finding a vacant space is  $1/2(1/(1 - q_0) - 1) + 1/2(1/(1 - q_1) - 1) = 3$ , whereas it would be  $1/(1 - q) - 1 = 2$  with an occupancy rate of  $2/3$  all the time. This result is an application of Jensen's Inequality<sup>19</sup> since, under the binomial approximation, the function relating the expected number of occupied parking spaces searched to the occupancy rate is convex. Intuition suggests that this convexity property should hold for the simulations too. Thus, we term this effect the *Jensen's Inequality effect*.

To get an idea of the quantitative importance of the Jensen's Inequality effect, assume for the sake of argument that the current mean cruising-for-parking time depends on the contemporaneous vacancy rate. Then the expected number of parking spaces searched

---

<sup>18</sup> A useful place to start might be to define the *strength of the bunching effect* as the ratio of the simulated mean bunch length to the corresponding number obtained under the binomial approximation, and then to measure how the strength of the bunching effects is related to the simulation model's parameters. If there is no bunching effect, the strength of the bunching effect, so defined, equals 1.0, which is appropriate for the multiplicative decomposition of effects.

<sup>19</sup> In the probabilistic context, Jensen's Inequality states that if  $X$  is a random variable and  $f$  is a convex function, the expected value of  $f$  is greater than  $f$  evaluated at the expected value of  $X$ ; here  $X$  is the occupancy rate, and  $f$  is the function relating the expected number of occupied parking spaces searched to the occupancy rate.

before finding a vacant space (including the vacant space) would equal the reciprocal of the harmonic mean of the vacancy rate<sup>20</sup>, and the expected cruising-for-parking time would equal this number minus 0.5000. Figure 5 displays the frequency distribution of the occupancy rate for the central base case simulation.

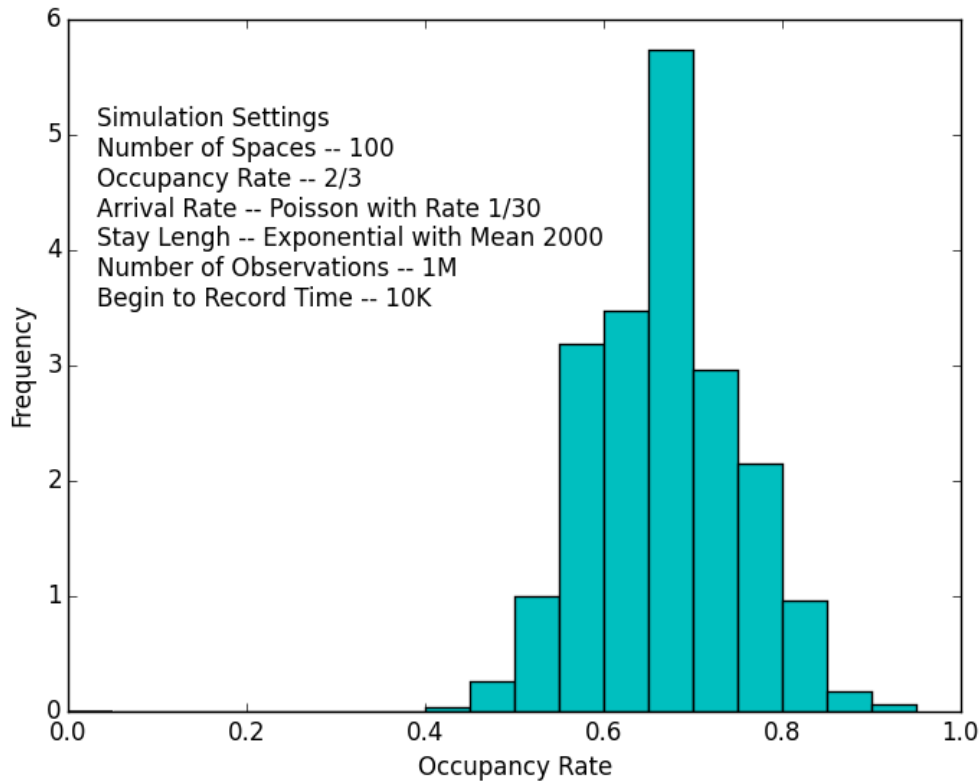


Figure 5: Frequency Distribution of the Occupancy Rate: Central Base Case

The corresponding harmonic mean of the vacancy rate for this simulation is 0.3082.

Adjusting the binomial approximation by replacing the expected vacancy rate with the harmonic mean of the realized vacancy rate gives an expected number of parking spaces

<sup>20</sup> Thus, the strength of the Jensen's Inequality effect may be measured as the ratio of the arithmetic mean of the vacancy rate and the harmonic mean. If indeed expected cruising-for-parking time is a convex function of the occupancy rate, this ratio has a minimum value of 1, which is appropriate for multiplicative decomposition, and a higher number represents a stronger effect.

searched before finding a vacant parking space (including the vacant parking space) of  $1/0.3082 = 3.245$ , and an expected cruising-for-parking time of 2.745. Thus, this interpretation of the Jensen's Inequality effect would explain a proportion  $(2.745 - 2.500)/(3.602 - 2.500) = 0.2223$  of the difference between the simulated mean cruising-for-parking time and that obtained via the binomial approximation. This interpretation of the Jensen's Inequality effect likely overestimates its importance since it assumes that each car faces an occupancy rate over its entire search equal to the occupancy rate at the time it entered the track. If, to the contrary, fluctuations in the occupancy rate are of very high frequency, then each car would face an occupancy rate over its entire search equal to the expected occupancy rate, in which case the Jensen's Inequality effect would disappear. Thus, the importance of the Jensen's Inequality effect depends negatively on the frequency of fluctuations in the vacancy rate.

Figure 6 gives some insight into the importance of short relative to long waves in determining the difference between the arithmetic and harmonic means of the occupancy rates, which is at the heart of the Jensen's Inequality effect. Panel A plots a moving average of the occupancy rate for the central base case simulation run, where the average is over the 100 time units (3 minutes); panels B and C do the same but with an average over 1000 time units (30 minutes) and 10,000 time units (5 hours). Together the panels give an idea of the periodicity of fluctuations in the occupancy rate. Recall that the average cruising-for-parking time for the central base case simulation run is 3.602 time units (6.484 seconds). Over such a small time period, during which, on average, only 0.1201 cars



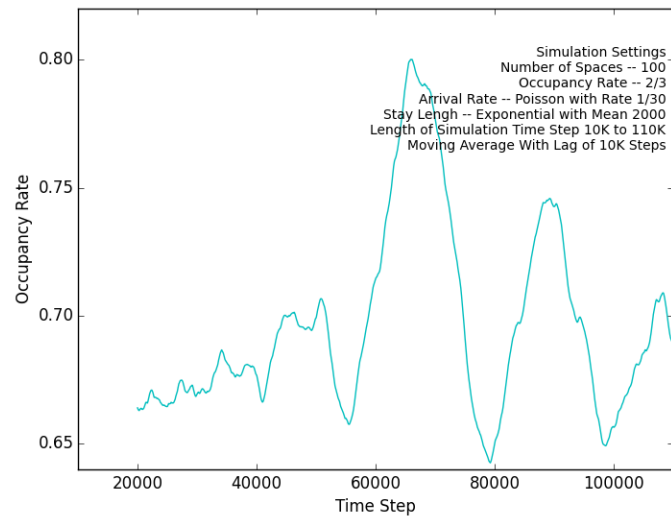
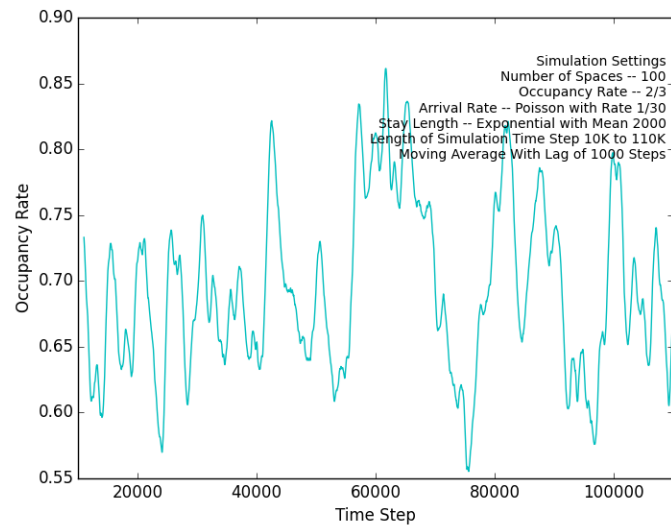
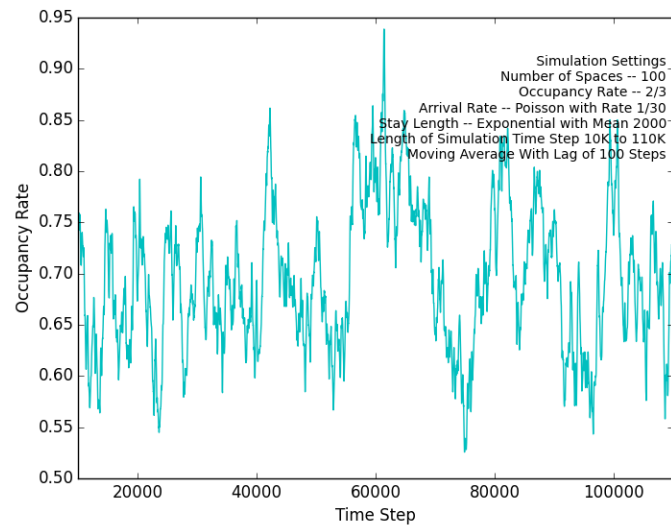


Figure 6: Different Moving Averages of the Occupancy Rate

Notes: Panel A: moving average over 100 time units; Panel B: moving average over 1000 time units; Panel C: moving average over 10,000 time units, each subsampled at 100 time units

enter the track, with the same average number leaving the track, the occupancy rate will only change marginally. Since the average variation in the occupancy rate over a trip is small, it appears that the Jensen's Inequality effect explains some 20% of the discrepancy between the mean cruising-for-parking time in the central base case simulation and the expected cruising-for-parking time obtained under the binomial approximation.

3. A searching car's probability of encountering a parking space that has recently been vacated is inversely proportional to the number of cars circling the track. We term this the *competition effect*. The binomial approximation for the expected cruising-for-parking time ignores this effect.

Figure 7 displays a time series of the number of cars on the track, sampled at intervals of 100.

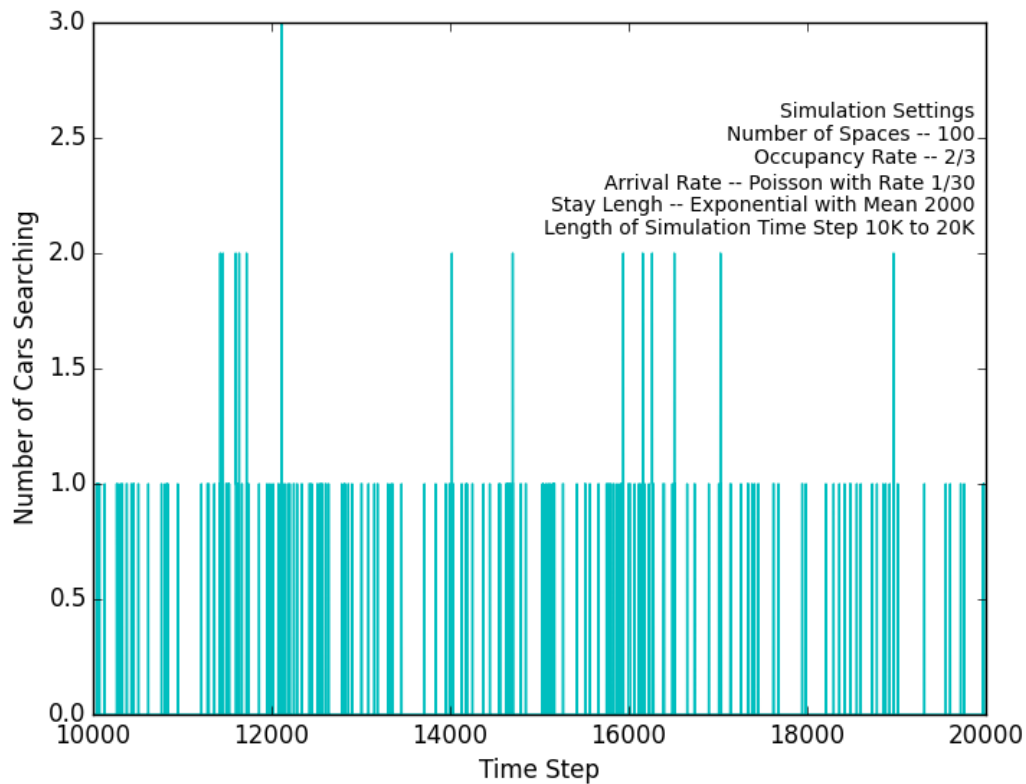


Figure 7: Time Series for the Number of Cars Currently Searching at Occupancy Rate 2/3  
 Notes: The simulation run has 10,000 time steps.

We now explore the distribution of the number of cars circling the track under the approximating assumption that the underlying stochastic process determining the number of cars cruising the track is a birth-and-death process with birth rate  $\mu$  and death rate  $n(1 - q)$ , where  $n$  is now the number of cars cruising for parking. Cars enter the track at Poisson rate  $\mu$ . Under the binomial approximation, each of the  $n$  cars cruising for parking finds a vacant parking space at the Poisson rate  $1 - q$ , and since, also under the binomial

approximation, these Poisson rates are independent, the rate at which cars cruising for parking find a vacant parking space is  $n(1 - q)$ . Let  $p_n(t)$  be the probability that the number of cars cruising for parking at time  $t$  is  $n$ . Then we have that

$$p_n(t + dt) = p_n(t)(1 - n(1 - q)dt - \mu dt) + p_{n-1}(t)\mu dt + p_{n+1}(t)(n + 1)(1 - q)dt \text{ for } n = 1, \dots, \infty$$

$$p_0(t + dt) = p_0(t)(1 - \mu dt) + p_1(t)(1 - q)dt \quad (1)$$

The interpretation of the second equation is as follows. During the increment of time  $dt$ , it is infinitely more likely that one state transition will occur than more than one state transition. The probability that the number of cars cruising for parking is zero at time  $t + dt$  is therefore (the probability that zero cars were cruising for parking at time  $t$  multiplied by the probability that there was no entry to the track during the increment of time,  $1 - \mu dt$ ) plus (the probability that one car was cruising for parking at time  $t$  multiplied by the probability that the car found a vacant parking space,  $(1 - q)dt$ ). The interpretation of the first equation is similar. Letting  $p_n$  be the steady-state probability that the number of cars cruising for parking is  $n$ , (1) implies the recursion

$$p_n = (\mu/[n(1 - q)])p_{n-1} \text{ for } n = 1, \dots, \infty. \quad (2)$$

From this, we obtain the infinite series

$$\sum_{n=0}^{\infty} p_n = 1 = p_0(1 + z + z^2/2! + \dots) \text{ where } z \equiv \mu/(1 - q). \quad (3)$$

The value of the infinite series is  $e^z$ . Thus,  $p_0 = e^{-z}$ ,  $p_1 = ze^{-z}$ ,  $p_2 = z^2e^{-z}/2!$ , etc. The mean number of cars cruising for parking is therefore

$$m = 0e^{-z} + 1(ze^{-z}) + 2(z^2e^{-z}/2!) + 3(z^3e^{-z}/3!) + \dots = z. \quad (4)$$

In the base case,  $z = \mu/(1 - q) = (1/30)/(1/3) = 0.1$ , which implies that  $m = 0.10$ ,  $p_0 = 0.9048$ ,  $p_1 = 0.9048E-1$ ,  $p_2 = 0.4524E-2$ ,  $p_3 = 0.1508E-3$ ,  $p_4 = 0.377E-5$ , .... The variance is

$$\text{var} = z^2 e^{-z} + (1-z)^2 (ze^{-z}) + (2-z)^2 (z^2 e^{-z}/2!) \dots,$$

which after some tedious algebra can be shown to reduce to  $z$ , which equals 0.1 in the base case.

The above approximation ignores that, on average, a car travels 0.5000 distance units before encountering a parking space, taking 0.5000 time units. With a Poisson entry rate of 1/30 cars per time unit, the expected number of cars on the track that have not yet reached the first parking space is  $0.5000/30 = 0.01667$ . Taking this into account, the adjusted binomial approximation of the expected number of cars cruising for parking is 0.1167.

The corresponding mean for the central base case simulation run is 0.1038. Thus, at least for the central base case simulation, the birth-and-death/Poisson approximation for the expected number of cars cruising for parking is quite accurate. As we shall note later, however, this is not the case for the simulation run with an expected occupancy rate of 5/6, and may not be the case for other simulation runs with an expected occupancy rate of 2/3.

The above calculations may be employed to estimate the number of other cars on the track when a car is cruising for parking. Under the Poisson/birth-and-death approximation, this equals the expected number of cars cruising for parking conditional on at least one car cruising for parking, minus one, which can be shown to equal 0.05083. Under the binomial approximation, from the perspective of a car cruising for parking, the

probability that the next parking space will be vacant is  $1/3$ , so that the probability that the next parking space will be vacant and that no other cruising car reaches it before the car in question is  $(1/3)(1 - 0.05083/100) = 0.3332$ . Under these assumptions, therefore, with an occupancy rate of  $2/3$ , the competition effect is quantitatively unimportant.

Applying the same logic but with the observed distribution of the number of cars cruising for parking (recorded in Table 3) in the central base simulation gives an expected number of other cars on the track when a car is cruising for parking of 0.07474, for which the competition effect is also quantitatively unimportant. Again, as we shall note later, the competition effect does become important for high occupancy rates, where the mean number of cars cruising for parking is considerably higher, and may be significant for some simulation runs with an expected occupancy rate of  $2/3$ .

4. To illustrate the next effect, suppose that there are only 10 parking spaces round the track, that the expected occupancy rate is  $2/3$ , that a driver has circled the track without finding a vacant space, and that the driver is and remains the only car cruising for parking. Under our assumed distribution of parking stay times, conditional on his having circled the track without having found a vacant space, the probability that the first parking space he encountered upon entering the track is still occupied on his second circuit is one minus the probability that it was vacated during his first circuit (that probability is  $\int_0^{10} \lambda e^{-\lambda t} dt = 0.004988$ , where  $\lambda = 1/2000$  is the Poisson rate at which a parking space is vacated), which equals  $e^{-1/200} \approx 0.9950$ . The same argument applies to the second and so on occupied spaces that he encountered upon entering the track, and also to his subsequent circuits of the track. The obvious term for this effect is the

*circling-the-block effect*. Since, by assumption, the rates at which occupied parking spaces are vacated are statistically independent, we can proceed further. The probability that a car makes a second circuit of the track without encountering a vacant parking space, contingent on its not having found a vacant space on the first circuit, is approximately  $(0.995)^{10} = 0.9512$ . The probability that it makes an  $n+1^{\text{st}}$  circuit of the track without encountering a vacant parking space on the  $n^{\text{th}}$  circuit is the same.

In the central base case simulation, only 7 cars out of the one million circled the block, and only 1 of the 7 circled twice. Thus, in this simulation the cruising-the-block effect is of negligible importance. As we shall see, however, this may not be the case for other simulations with the same parameter values. The circling-the-block effect is more important the higher is the expected occupancy rate and the smaller the number of parking spaces.

5. Thus far we have identified four different effects, each of which goes part of the way towards explaining why the simulated mean cruising-for-parking time significantly exceeds that obtained under the binomial approximation: the bunching effect, the Jensen's Inequality effect, the competition effect, and the cruising-the-block effect. We considered each of these effects in isolation. But there may also be important interaction effects.

These interaction effects may cause the combination of pairs of effects on mean cruising-for-parking time to be subadditive or superadditive. There are six pairs of effects. Here we shall consider only three.

The most obvious of the pairwise interaction effects is that between the competition effect and the cruising-the-block effect. Suppose that there is and continues to be a single car cruising for parking, and that the track has all 10 of its parking spaces occupied. We argued above that, if parking stay times are negative exponentially distributed with mean 2000, the probability that the cruising car will make a full circuit of the track without finding a vacant space is  $(0.995)^{10} = 0.9512$ , which is the probability that none of the 10 spaces is vacated in one circuit. Now, suppose that there are two cars cruising for parking with all of the track's parking spaces occupied, and that no cars enter on the next circuit. The probability that a particular car will make a full circuit of the track without finding a vacant space equals the probability that none of the 10 spaces is vacated in one circuit, plus one-half the probability that one is vacated:  $0.9512 + (0.5)(10)(0.995)^9(0.005) = 0.9751$ . Put alternatively, when the number of cars cruising for parking increases from one to two, and when initially all parking spaces are occupied, the probability that a particular car will find a vacant space on a full circuit of the track falls from 0.04877 to 0.02488.

We term the interaction between these two effects the *multiplication* effect because, if the number of cars cruising for parking were to persist, expected cruising-for-parking time



would equal the expected cruising-for-parking-time with one car cruising for parking multiplied by the number of cars cruising for parking.

The next pairwise interaction effect we consider is that between the Jensen's Inequality effect and the competition effect. Both intuition and the earlier discussion suggest that the number of cars cruising for parking and the realized occupancy rate are positively correlated. This is demonstrated in Table 3, which for the central base case run (which has an expected occupancy rate of  $2/3$ ) displays the mean occupancy rate and the realized probability distribution of the number of cars cruising for parking, conditional on the number of cars cruising for parking. The positive correlation between the number of cars cruising for parking and the occupancy rate conditional on the number of cars cruising for parking fattens the right tail of the distribution of cruising-for-parking times. For want of a better term, we term this the *correlation effect*. A statistical accident in which fewer than the expected number of cars vacate their parking spaces has a direct positive effect on the realized occupancy rate, which in turn increases the expected number of cars cruising for parking. A statistical accident in which more than the expected number of cars enter the track has a direct effect on both the realized occupancy rate and the number of cars cruising for parking. Thus, the Jensen's Inequality effect and the competition effect on the expected cruising-for-parking time are super-additive.

Number of Cars Searching, k	0	1	2	3	4	5
Probability of k Cars	0.9034	0.08983	0.006322	0.0004220	0.0000193	0.0000023
Mean Occupancy	0.6620	0.6988	0.7425	0.8074	0.8343	0.8871

Table 3: Probability of k Cars Searching and Mean Occupancy Rate with k Cars  
Searching with Occupancy Rate 2/3

The final interaction effect we consider is that between the two effects that appear to be of dominant importance when the occupancy rate is 2/3, the bunching effect and the Jensen's inequality effect. Earlier we defined the strength of the bunching effect to be the ratio of the simulated mean number of cars in a bunch to the corresponding expected number under the binomial approximation, and the strength of the Jensen's Inequality effect to be the ratio of the arithmetic mean of the vacancy rate to the harmonic mean. We now undertake a multiplicative decomposition of the effects. Letting A denote the ratio of the simulated mean cruising-for-parking time to the corresponding expected number under the binomial approximation, B denote the strength of the bunching effect as defined earlier, J denote the strength of the Jensen's inequality effect as defined earlier, and R denote the strength of the residual effect, defined to equal 1 when the residual effect is neither positive nor negative. It can be shown that  $A \equiv BJR$ . Applying this formula for the central base case simulation gives  $R = (3.602/2.5) \div [(0.3333/0.3082)](3.887/3.000) = 1.02816$ . Since the competition effect, the circling-the-block effect, and their interaction effects, are negligible for the central base case simulation, then the residual effect reduces to the interaction effect between the bunching effect and the Jensen's Inequality effect. That the residual effect is only slightly greater than 1.000 suggests that the bunching effect and the Jensen's inequality effect are only very slightly super-additive.

When we started our simulations, we fixed the simulation time at  $10^6$  time units. But then, unexpectedly, we found that the mean cruising-for-parking time was substantially different from run to run, so we switched to simulation runs in which  $10^6$  cars were simulated, which as noted earlier corresponds in unnormalized time units to about 15,000 hours. To investigate how different the mean cruising-for-parking time is across runs, we reran the base case simulation for  $10^5$  observations (not the  $10^6$  observations that were used in the central base case simulation) 1000 times. The probability density function of both the mean and the variance of the means are presented in Figure 8. The only explanation that we can think of for why the results differ substantially from run to run, despite the large number of cars on each run, is "disasters" -- very low probability but extreme events. Even for one hundred thousand observations, there may not even be a minor disaster, but a major disaster may increase mean cruising-for-parking time substantially. A major disaster is "a perfect storm" that comes about purely by chance,

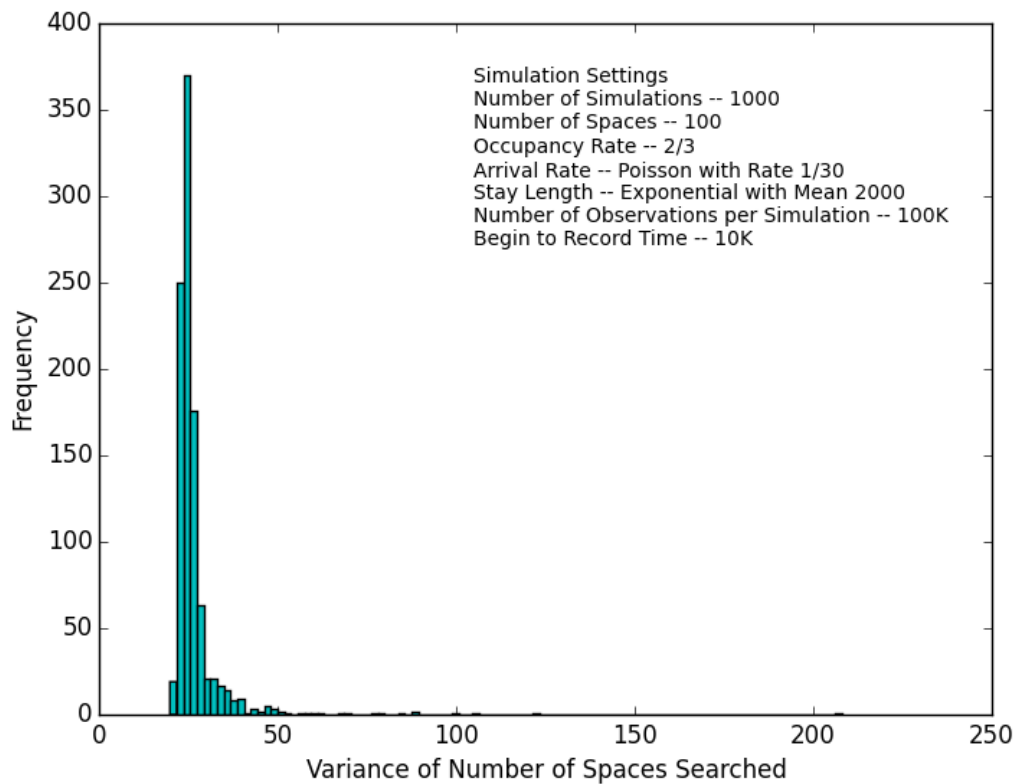
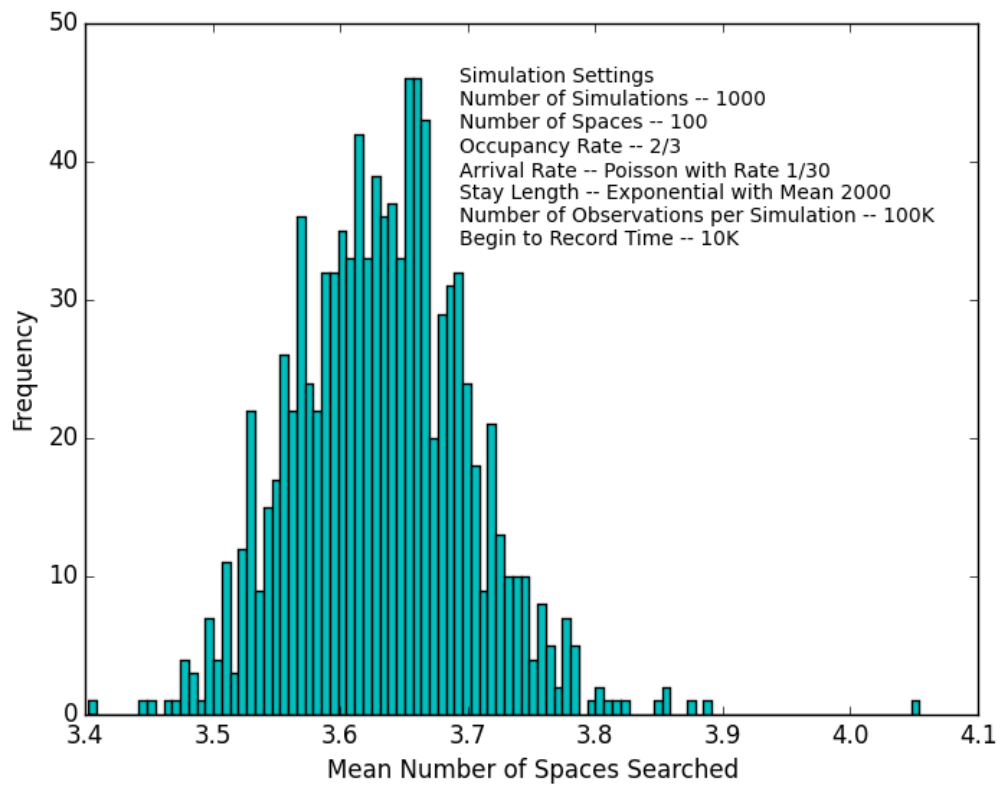


Figure 8, Panel A: Distribution of the Number of Parking Spaces Searched for 1000 Simulations with Occupancy Rate =  $2/3$

Panel B: Distribution of the Variance of the Number of Parking Spaces

Notes: Each simulation is for  $10^5$  time units.

Mean of means: 3.633. 95% confidence interval of means<sup>21</sup> (3.527, 3.745)

Mean of variances: 26.79. 95% confidence interval of variances (22.32, 36.53).

when stochastic realizations are such that not only does parking become almost completely blocked but also it takes an exceptionally long period for parking to unblock (either because the realized entry rate is exceptionally high, or because the rate at which parking spaces are vacated is exceptionally low, or because the number of cars cruising for parking is exceptionally high). The few major disasters that occurred in this set of simulation runs can be identified as the right-tail outliers in the Figures. Thus, even though the competition effect and the cruising-the-block effect were of negligible importance in the central base case simulation, the same may not be the case for other simulations with the same exogenous parameter values.

We conclude this section by further exploring the bunching effect. We started with the broad intuition that, if the parking on an entire block becomes fully occupied, whether by statistical accident or by some special event, it will have ripple effects in space-time. Cars that were intending to park on that block will move to neighboring blocks in their search for curbside parking. This got us to thinking about how bunches of occupied parking spaces evolve over space-time. We started with a snapshot of the track, such as that shown in Figure 9 below.

---

<sup>21</sup> Computed so that 2.5% of the probability mass lies to the left of the interval and 2.5% of the mass lies to the right of the interval.



Figure 9: Snapshot of Street with Mean Occupancy Rate  $2/3$

Notes: White spaces are occupied, black spaces are vacant. There are 70 occupied spaces and 30 vacant spaces. The vacant spaces are 2, 3, 4, 6, 7, 14, 17, 21, 23, 24, 25, 26, 27, 28, 29, 31, 33, 35, 37, 38, 48, 50, 54, 60, 63, 70, 71, 73, 78, 87.

We then aimed to examine how bunches evolve -- expanding, contracting, forming, and dissolving. Examining this analytically, we uncovered one result. Bunches tend to move clockwise over time since there is a higher than average probability that the first vacant space after a bunch of occupied spaces becomes occupied, and a lower than average probability that the first vacant space before a bunch of occupied spaces becomes occupied. Under the binomial approximation, the unconditional arrival rate of a car at a particular parking space is Poisson with arrival rate at the entire track,  $1/30$ , divided by the number of parking spaces around the track, which is 100, times the expected number of parking spaces searched before finding a vacant space (including the vacant space), which is 3.0, for a total of  $1/1000$ . But conditional on an initial situation where at time  $t$  there is a bunch of six occupied spaces followed by a vacant space, the rate at which cars arrive at the vacant parking space is different. The probability that a car that entered the track in the previous time unit passes that vacant space in time interval  $[t, t + dt]$  is  $dt/3000$ ; the probability that a car that entered the track in the time unit before that

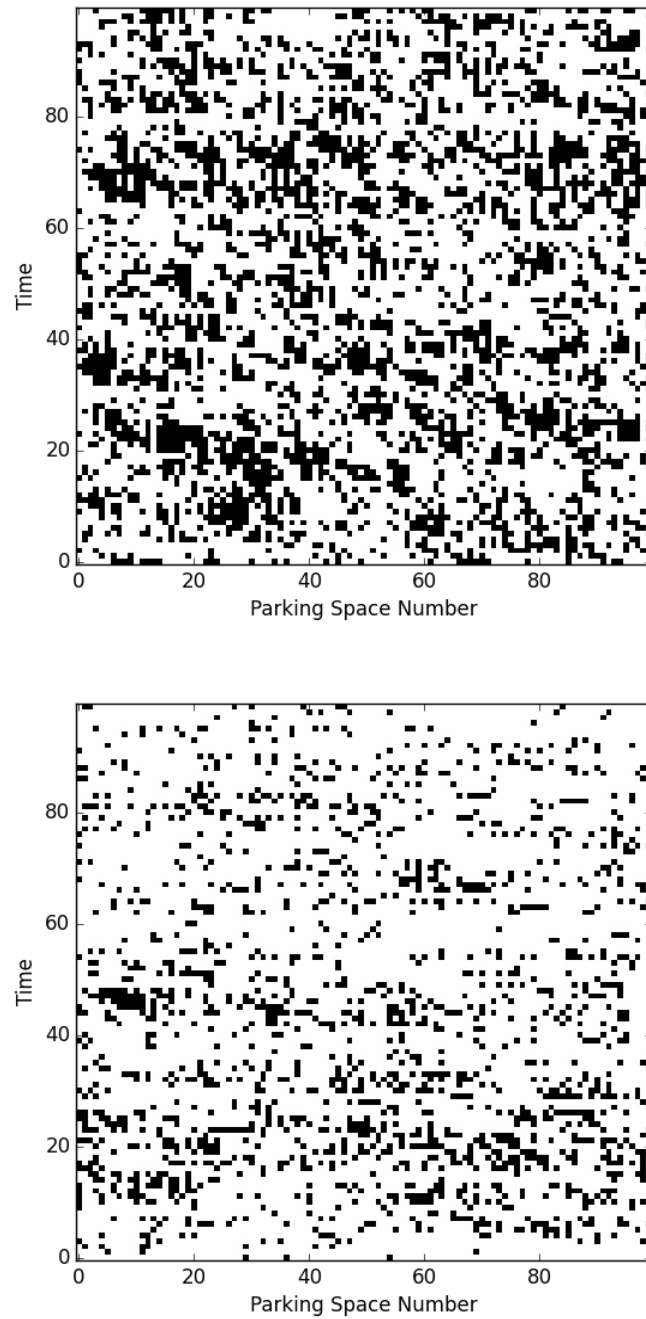


Figure 10, Top Panel: Snapshots of the Circle Every 1000 Time Steps with Occupancy Rate =  $2/3$

Bottom Panel: Snapshots of the Circle Every 1000 Time Steps with Occupancy Rate =  $5/6$

Notes: White is occupied, black is vacant

passes that vacant spot in that increment of time is  $dt/3000$  times that the probability that the left-side neighbor was occupied at time  $t - 1$  (conditional on its being occupied at time  $t$ ), which is approximately  $1999/2000$ ; ---. Thus, the expected rate at which cars arrive at the vacant parking space is somewhat less than  $7/3000$ . This is the rate at which the front edge of a bunch of seven occupied spaces moves forward. By an analogous argument, if there is a string of vacant parking spaces before the first of a bunch of occupied parking spaces, the rate at which the rightmost of the vacant spaces is occupied is lower than  $1/1000$ .

Unfortunately, we were unsuccessful in uncovering a body of literature on spatio-temporal statistics that address the spatio-temporal evolution of bunches. The best we could do was to take snapshots of the track every so many time steps. Figure 10 displays such a snapshot, taken every 1000 time steps (every half hour) with occupancy rate  $2/3$  (top panel) and  $5/6$  (bottom panel). Bunching shows up as positive horizontal correlation between white spaces (or black spaces); standard autocorrelation as positive vertical correlation; and the type of spatio-temporal autocorrelation discussed in the previous paragraph as positive correlation in a northeast direction. To the naked eye, bunching and clumping of vacant spaces are evident, but not standard autocorrelation nor positive correlation in the northeast direction<sup>22</sup>.

We could have undertaken considerably more statistical analysis of the central base case run, and we could have proceeded more formally. Nevertheless, we judge that the

---

<sup>22</sup> While we have not investigated this, at high occupancy rates intuitively bunching should also occur among cars cruising for parking.



statistical evidence we have accumulated makes a compelling case that the binomial approximation is a poor one, and that the expected cruising-for-parking time calculated using it significantly underestimates the true expected cruising-for-parking time. The next section investigates the effects of parameter changes one at a time. Among other things, it will: i) show the binomial approximation becomes increasingly poor as the occupancy rate increases; ii) provide strong evidence of the circling-the-block effect; and iii) present evidence that the behavior of the parking system is insensitive to the turnover rate, holding constant the occupancy rate, and is somewhat sensitive to the distribution of stay times, holding constant the average stay time.

## 5. *Comparative Stochastic Steady States*

We start off by examining how the stochastic steady state changes as the Poisson entry rate changes, *ceteris paribus*. A change in the Poisson entry results in a proportional increase in the expected occupancy rate. Table 4 shows the moments of the distribution of cruising-for-parking times when the vacancy rate is successively halved. In the central base case, the occupancy rate is  $2/3$  and the vacancy rate is  $1/3$ ; a halving of the vacancy rate to  $1/6$  results in an increase in the occupancy from  $2/3$  to  $5/6$ ; and a further halving of the vacancy rate to  $1/12$  results in an increase in the occupancy rate from  $5/6$  to  $11/12$ . For each occupancy rate, the results are for only single simulation runs of a million cars, and, as we have seen, the results for each case can be quite different across simulation runs. The number in each cell is computed from the corresponding simulation run. The number in parenthesis in a cell gives the corresponding number according to the binomial approximation.

Occupancy Rate	Mean	Variance	Pearson Skewness	Fisher Kurtosis
2/3 (base case)	3.614 (2.500)	23.74 (6.000)	3.685 (2.041)	27.87 (6.167)
5/6	16.84 (5.500)	3575 (30.00)	15.25 (2.640)	423.1 (6.033)
11/12	97.63 (11.50)	86600 (132.0)	8.248 (2.760)	117.6 (6.008)

Table 4: Moments of the Distribution of Cruising-for-Parking Times in Normalized Time Units

Notes: For each case, the principal numbers are from a single simulation with  $10^6$  cars; the numbers in parenthesis are those generated by the binomial approximation.

Table 4 demonstrates that the ratio of the mean cruising-for-parking time for a simulation to the corresponding expected value obtained from the binomial approximation becomes increasingly large as the expected occupancy rate increases. In other words, the binomial approximation underestimates the mean cruising-for-parking time by an increasingly large proportion, the higher is the expected occupancy rate.

We were curious concerning the accuracy of the binomial approximation when the expected rate is low. Figure 11 displays our now-standard ratio plot when the occupancy rate is 0.1. The mean cruising-for-parking time is 0.6172. The expected cruising-for-parking time according to the binomial approximation is  $1/0.9 - 0.5 = 0.6111$ . Even though the binomial approximation provides an accurate approximate of the actual mean cruising-for-parking time, it still sharply underestimates the fatness of the right tail of the distribution.

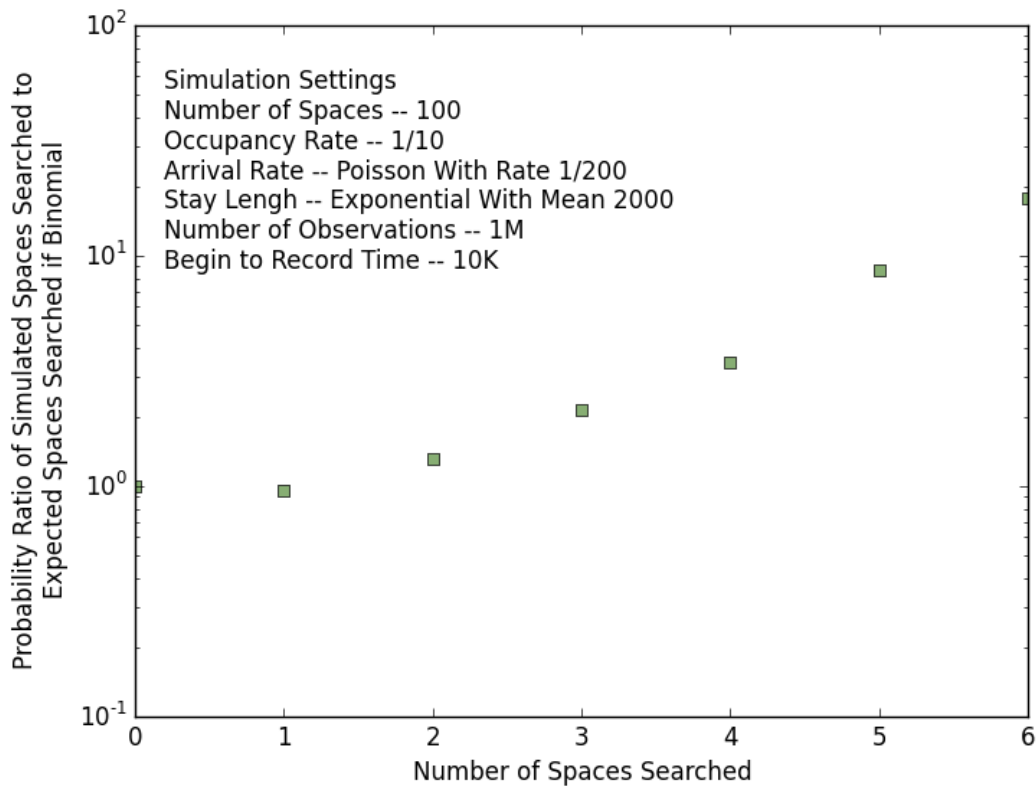


Figure 11: Ratio Plot with a Mean Occupancy Rate = 0.1

Figure 12 presents two pairs of panels. The upper panel on the left shows a smoothed (moving average over 100 time steps or 3 minutes) time series for the occupancy rate and the bottom panel the corresponding time series for the number of cars on the track, both for the case where the expected occupancy rate is  $2/3$ , between 30,000 and 40,000 time units into the simulation (15 hours and 20 hours, respectively). The corresponding panels on the right are for the case where the expected occupancy rate is  $5/6$ . With an expected occupancy rate of  $2/3$ , over the entire simulation the mean cruising-for-parking time was 3.614, the mean number of cars searching for parking was shown earlier to be 0.1038 (in contrast to 0.1167 according to the Poisson approximation), with only 7 cars out a million making more than a complete circle of the track. With an expected occupancy rate of

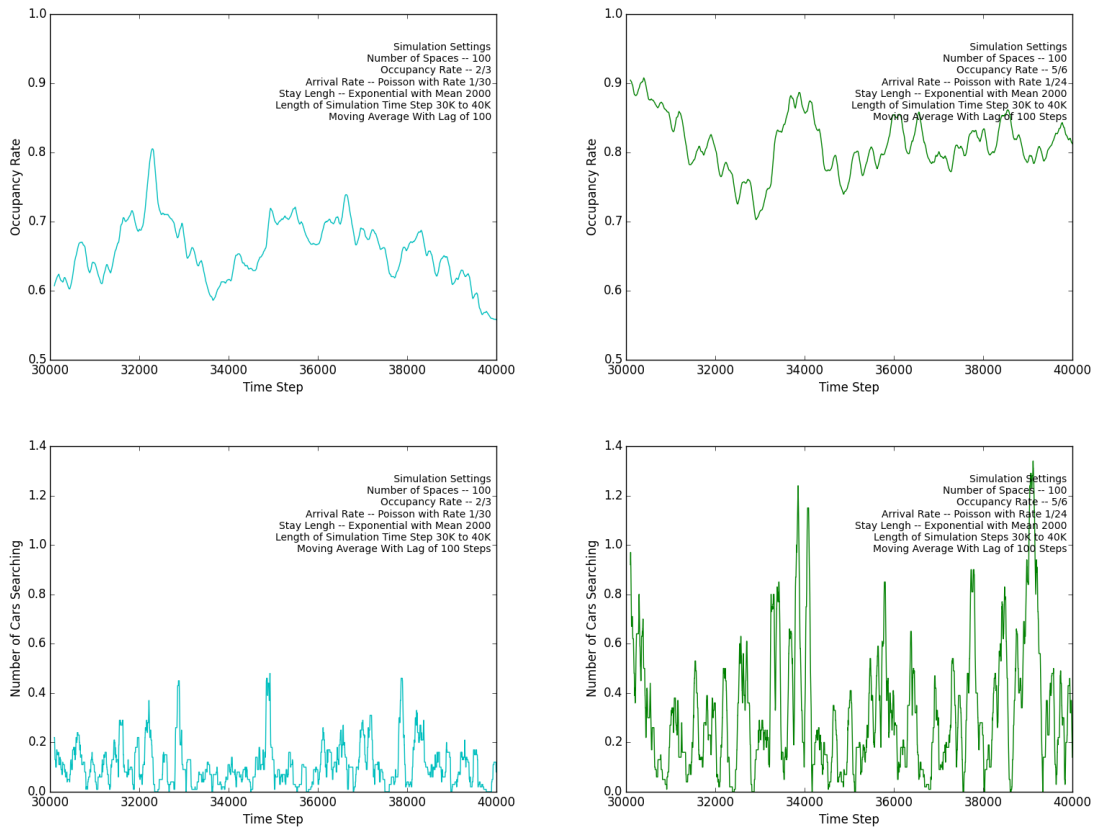


Figure 12: Smoothed Time Series for Number of Cars on the Track against the Occupancy Rate

Notes: The upper panel on the left shows the smoothed time series for the number of cars on the track, and the lower panel on the left shows the time series for the occupancy rate, both with the occupancy rate =  $2/3$ . The panels on the right show the same with the occupancy rate =  $5/6$ .

$5/6$ , over the entire simulation the mean cruising-for-parking time is 16.84, with a maximum of around 403, and the expected number of cars searching for parking is 0.5067 (in contrast to 0.2167 according to the Poisson approximation). Also, while not shown in the Figure, there are rare events in which the number of cars simultaneously on the track exceeds twenty, and one point in time at which the number exceeds 40. Our conjecture is that these rare events occur when parking around the track becomes almost

completely occupied and an accumulation of cars cruising for parking develops, resulting in a significant number of cars circling the block.

Table 5 gives the moments for three cases that differ only in the number of parking spaces around the track. The first row repeats the results for the central base case simulation in which the number of parking spaces is 100, the second row gives the results when the number of parking spaces equals 1000, and the third row the results when the number equals 10.

Parameters	Mean	Variance	Pearson Skewness	Fisher Kurtosis
Central Base Case P = 100	3.614	23.74	3.684	27.87
Base Case except P = 1000 and Arrival Rate = 1/3	3.609	23.72	3.589	23.31
Base Case except P = 10 and Arrival Rate = 1/300	108.6	1681	7.353	85.01

Table 5: Moments of the Distribution of Cruising-for-Parking Times in Normalized Time Units with Occupancy Rate 2/3

Increasing the number of parking spaces around the track by a factor of 10 relative to the base case, holding their spacing the same has little effect on the moments. However, decreasing the number of parking spaces around the track by a factor of 10 strongly increases both the mean and variance of cruising-for-parking times. The obvious conjecture is that this is due to the circling-the-block effect being important with  $P = 10$  but not with  $P = 100$  (as we have shown) or with  $P = 1000$ . The ratio plots for these

cases, which are shown in Figure 13, confirm this conjecture. For  $P = 10$ , the circling-the-block effect would result in a sharp drop in the ratio between space 9 (since the number of spaces searched excludes the successful space searched, this corresponds to successful search on the last space before circling the block) and space 10 (successful search on the first space after circling the block), and that is what is observed.

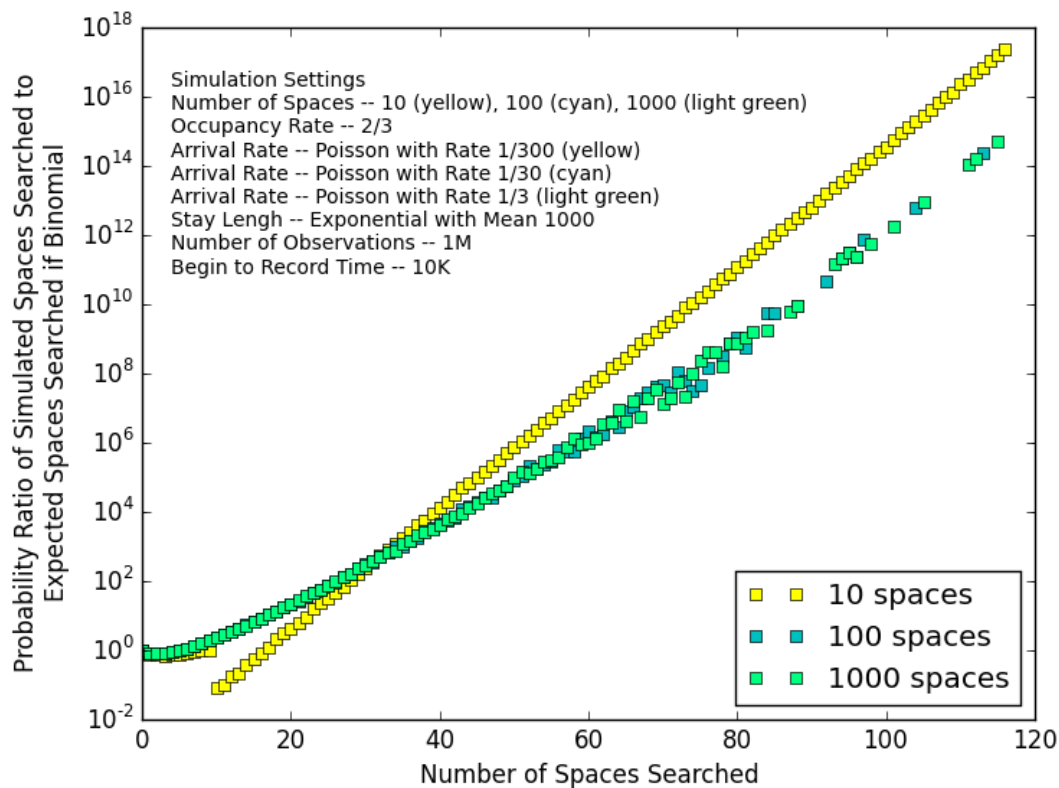


Figure 13: Ratio Plots: Number of Parking Spaces = 10, 100, and 1000 and Expected Occupancy Rate =  $2/3$

Let us consider the parking system when, after a long series of statistical accidents, it finds itself in a situation where all 10 parking spaces are occupied and 5 cars are cruising for parking. Each of the 10 parking spaces is vacated at the rate  $1/2000$ , for a Poisson death rate of occupied parking spaces of  $1/200$ . The Poisson entry rate is  $1/300$ . Thus,

the parking logjam will eventually break up, but it may take a long time. From the perspective of a driver, the Poisson rate at which she will be the first to pass by a recently vacated parking space is  $1/1000$ . If the situation were to persist, the expected cruising-for-parking time for the driver would be 1000 (1/2 hour). But the situation will likely not persist. In the next increment of time,  $dt$ , the probability is  $dt/200$  that the system will make a transition from 5 cars circling the block to 4, and  $dt/300$  that it will make a transition from 5 cars circling the block to 6. We refer to parking as being saturated when a vacated parking space is taken almost immediately by one of the cars that is cruising the block. In this saturated state, since each of the cars cruising round the circle has the same probability of being the first to encounter the single parking space that has just been vacated, the queue discipline is random access, so that the parking system behaves like an  $M/M/1/\infty/RSS$  queuing system. The common probability of encountering a parking space in an increment of time  $dt$  is  $(P\lambda/n)dt$ , where  $P\lambda dt$  is the probability that some space is vacated so that  $P\lambda/n$  is the probability that a recently-vacated space is taken by a particular car among the  $n$  that are cruising for parking<sup>23</sup>.

Table 6 compares the moments of the cruising-for-parking time distribution for three situations. The first row displays the results for the central base case. The second row

---

<sup>23</sup> The reader might be inclined to dismiss the "small- $P$ " case as unrealistic. In most real-world situations,  $P$  is large, but also parking spaces are differentiated according to distance to the parker's destination, and many parkers are averse to walking long distances from their parking space to their destination. Consequently, many parkers do follow a circling-the-block strategy (often supplemented with a decision to garage park after a certain number of unsuccessful circles of the block), even though they should recognize that the probability of finding a parking space on the second circuit of the block is typically much smaller than on the first circuit.

displays the results for a case that is identical to the base case except that the entry rate is doubled and the mean stay rate halved (while maintaining the negative exponential distribution of stay times), resulting in no change in the expected occupancy rate but a doubling of the turnover rate. The mean is not significantly different from that of the base case. Nor either is the corresponding ratio plot, which is displayed in Figure 14, Panel A. This result is qualitatively consistent with the binomial approximation in which the expected cruising-for-parking time is independent of the turnover rate, holding constant the occupancy rate.

Parameters	Mean	Variance	Pearson Skewness	Fisher Kurtosis
Base Case	3.614	23.74	3.685	27.87
Base Case but Doubling the Turnover Rate	3.597	24.42	5.422	123.3
Base Case but Constant Stay Time	4.188	34.25	4.096	60.08

Table 6: Moments of the Distribution of Cruising-for-parking Times in Normalized Time Units

Notes: For each case, the numbers are from a single simulation with  $10^6$  cars.



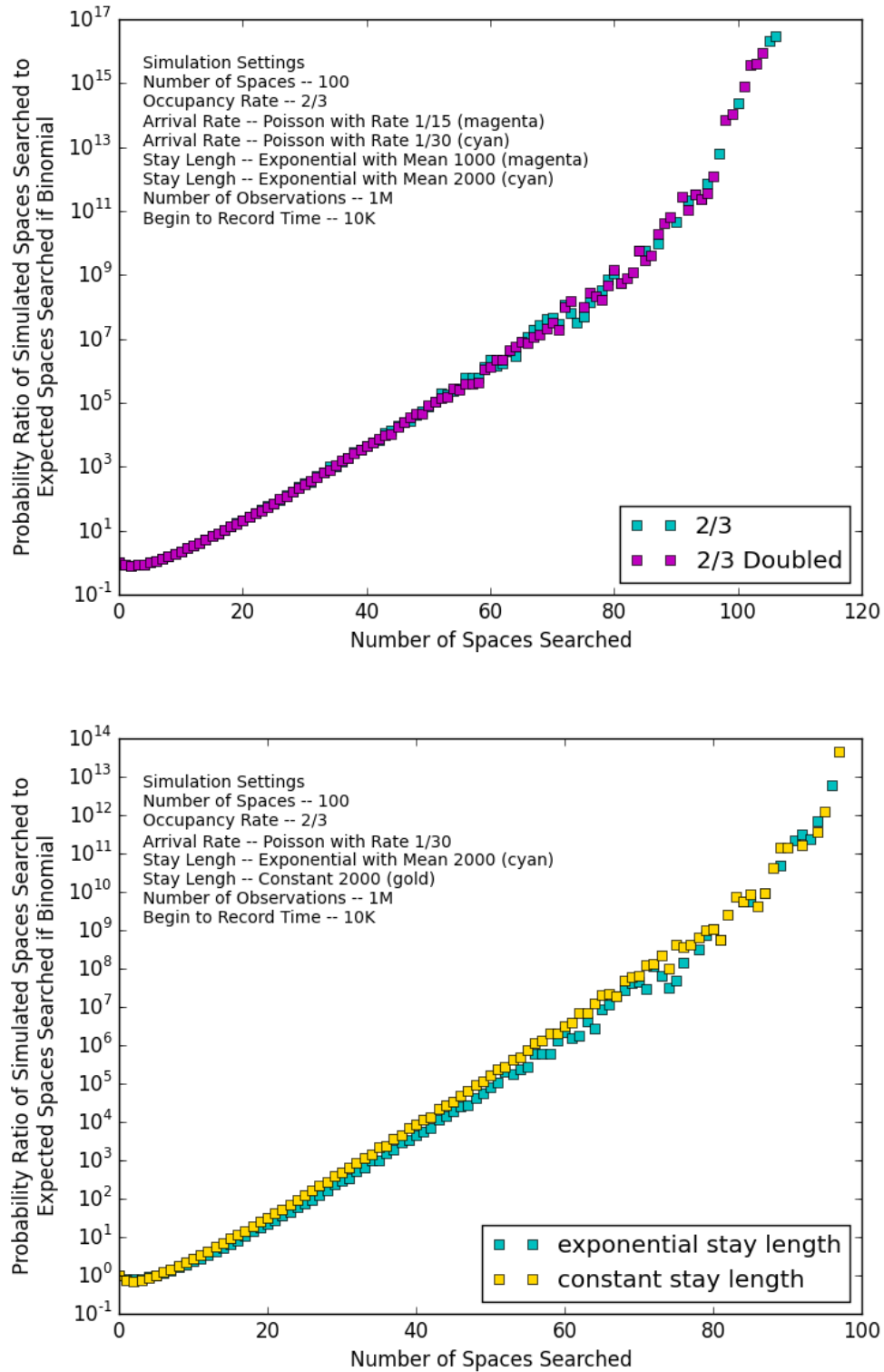


Figure 14: Ratio Plots

Notes: Panel A: Base case compared with base case but double the turnover rate.

Panel B: Base case compared with base case but constant parking duration.

The third row displays the results for the case that is identical to the base case except that the stay length is constant at one hour rather than being negative exponentially distributed with a mean of one hour (in terms of the queuing theoretic notation introduced earlier,  $B = D$  (deterministic) rather than  $M$  (Markovian)). The mean cruising-for-parking time with a constant stay time is significantly higher than that with a stay time that is negatively exponentially distributed, which is visually evident in the corresponding ratio plot in Figure 14, Panel B. Our conjecture is that short stay times of some of the cars with the negative exponential distribution allow a blockage to unlock more quickly.

At the risk of oversimplification, it appears that the parking system we have examined has four phases. In the first phase, the expected parking occupancy rate is modest and the system behaves in much the way predicted by the binomial approximation. The binomial approximation still severely underestimates the fatness of the right tail of the distribution of cruising-for-parking times but there is little probability weight there. In the second phase, the expected occupancy rate rises to the point that, via Jensen's Inequality, stochastic fluctuations in arrivals and exits generate stochastic fluctuations in the realized vacancy rate that raise the expected cruising-for-parking time substantially above that predicted according to the binomial approximation. Bunching is a related phenomenon, reflecting localized variations in the realized vacancy rate. In this phase, the circling-the-block effect and the competition effect are unimportant. In the fourth phase, by statistical accident, parking becomes saturated, with an accumulation of cars circling the track hoping to be the first to pass by a parking space that has just been vacated. In this fourth

phase, the circling-the-block effect and the competition effect dominate the behavior of the parking system, which is relatively easy to describe. The third phase, which lies between the second and the fourth phases, is the most complicated. Even though parking is not saturated, there is still a sufficient accumulation of cars cruising for parking that circling the block becomes significant. If stochastic realizations are favorable, the system moves out of this phase into phase two; if they are unfavorable, the system moves into phase four.

## 6. *Concluding Comments*

This paper reported on a voyage of discovery into cruising for parking. It examined cruising for parking in about the simplest context possible -- cruising for parking at a constant speed around a circle with evenly spaced parking spaces in a stochastic steady state, with a temporally and spatially invariant Poisson entry rate and negative exponentially distributed parking stay times. Even such a simple model appears to be analytically intractable. To investigate its properties, we employed stochastic simulation modeling without resort to any *ad hoc* assumptions. This is in contrast to most previous work on cruising for parking which has employed what we termed the "binomial approximation", that the probability that each parking space is vacant equals the mean vacancy rate, independent of history and the current state of parking around the track.

The exploration generated considerable data. Even though we analyzed them with only crude statistical tools, our results make a compelling case that, except in low occupancy rate situations where cruising for parking is not a problem, the binomial approximation

causes significant, and in high occupancy situations severe, underestimation of expected cruising-for-parking times. The exploration also uncovered many results that to us at least were unexpected, and raised more questions than it answered.

There are a number of obvious directions for future research. We proceeded on the principle that it is important to understand the basics before adding complications in the direction of realism. Along these lines, an obvious direction is to apply more sophisticated statistical tools to analyze the data generated by both the simulation runs we undertook (which are accessible via the hyperlink <http://math.ucr.edu/~parker/CruisingForParking/>) and similar runs. For queuing theorists, an obvious direction is to apply the full arsenal of queuing theory to our model, or to simplified variants of it, attempting to obtain analytical results regarding the properties of our model and better analytical approximations for the distribution of cruising-for-parking times.

The next bold step forward will be to generalize the paper's model to an isotropic, two-dimensional space. This generalization is challenging because it qualitatively changes the nature of the problem. In the one-dimensional problem we analyzed, a driver is simply a cellular automaton who keeps on driving round the circle until he finds a vacant parking space. In the two-dimensional problem, in contrast, the driver can adopt a wide variety of strategies in his search for a curbside parking space. Attempting to solve for a Bayesian Nash equilibrium is unrealistically ambitious. A simple approach is to compare the equilibria when all drivers adopt the same strategy. A more sophisticated approach is

to apply evolutionary game theory, looking for an evolutionarily stable mix of strategies. Following Arnott (2014), taking into account that most drivers have a specific destination and a desired arrival time, and will decide when to depart and when to start cruising for parking trading off in-transit travel time, cruising-for-parking time, walking time, schedule delay, and parking costs, would improve the model's realism.

Because of the difficulty of treating non-stationary dynamics, the analysis of most queuing models focuses on the steady state. But it is important to analyze cruising for parking in the context of rush-hour congestion dynamics since the expected occupancy rate varies systematically over the rush hour (indeed Geroliminis, 2015, does this under the binomial approximation). Practically, the best that can be hoped for is simulation models that endow cellular automata with some degree of sophistication rather than full rationality.

The paper's model assumes that drivers just keep on cruising for curbside parking until they find a vacant space. But realistically since cruising becomes increasingly frustrating and costly, cruisers develop stopping rules. What those stopping rules are depends on the alternatives -- balking (just returning home), changing the sequencing of the day's activities, parking in a garage<sup>24</sup>, and changing the search location. If stopping rules are not considered, cruising-for-parking models will consistently overestimate the probabilities of extreme cruising-for-parking times.

---

<sup>24</sup> While the paper has cast cruising for parking in the context of curbside parking, cruising for parking in parking lots and in parking garages is also important.

It is important to take account of spatial inhomogeneity. Much of the spatial inhomogeneity most relevant to cruising for parking is highly localized, pointing to a network simulation approach using complete street networks.

Another obvious direction for future research is to provide an integrated treatment of cruising for parking and traffic congestion. Cruising for parking affects traffic congestion, and traffic congestion affects the speed at which cars cruise for parking. Some work along these lines has been undertaken in Gerolominis (2015).

Thus far the paper has made no mention of economics. In cruising for parking, a driver not only increases the expected cruising-for-parking time of other cruisers but also contributes to traffic congestion. Parking pricing should account for both these externalities. Capacity is the other main category of parking policy. How should the rules proposed (e.g., in Arnott, Inci, and Rowse, 2016) to choose the amounts of curbside, public garage, and private garage parking be modified to account for the more sophisticated treatment of cruising for parking that this paper investigates?

The simulation results reported in the paper indicate how much extreme stochastic events can affect expected cruising-for-parking times. An analogous phenomenon in the context of traffic congestion has recently been receiving considerable attention, with policy discussion focusing on how hypercongestion (situations in which traffic become so congested that traffic flow falls as traffic density rises) should be nipped in the bud through enlightened traffic management policies, such as diverting traffic via signal

phase timing around locations where severe hypercongestion is imminent. How can the analogous phenomenon of near-gridlock in curbside parking in a downtown locale (in which a stock of cruisers for parking has accumulated) be avoided? The problem is made more difficult by the inability to distinguish between cars in transit and cars cruising for parking. Over sixty years ago Vickrey (1954) proposed (truly<sup>25</sup>) responsive curbside parking pricing, in which the price of curbside parking on a block is continuously adjusted so that at least one parking space is almost always available. At the time the policy was impractical, but now, with cell phone apps permitting curbside parking reservations (Du and Gong, 2016), such a policy might be implementable without causing drivers excessive frustration.

In the context of cruising for parking, Levy *et al.* (2013) characterized the binomial approximation as "working with averages", in contrast to their simulation model, which took explicit account of stochastic fluctuations. They emphasized the importance of taking stochasticity into account, and this paper's simulations have underscored their message. The enlightened management of curbside parking needs to take into account temporally and spatially localized stochasticity. The same is no doubt true of downtown traffic congestion more generally. How to accommodate such stochasticity in the design of downtown transportation policy is an important challenge for transportation researchers.

---

<sup>25</sup> Vickrey used the term "responsive pricing" to pricing that adjusts automatically and in real time to stochastic realizations. Shoup uses the term differently, to refer to *ex ante* pricing that is adjusted periodically on the basis of average performance but not in response to stochastic realizations

## REFERENCES

- Anderson, S. and de Palma, A. 2004. The economics of curbside parking. *Journal of Urban Economics* 55: 1-20.
- Arnott, R. 2014. On the optimal target curbside occupancy rate. *Economics of Transportation* 3:133-144.
- Arnott, R., and Rowse, J. 1999. Modeling parking. *Journal of Urban Economics* 45: 97-124.
- Arnott, R., and Rowse, J. 2009. Downtown parking in auto city. *Regional Science and Urban Economics* 39: 1-14.
- Arnott, R., and Rowse, J. 2013. Curbside parking time limits. *Transportation Research A: Policy and Practice* 55: 89-110.
- Arnott, R., Inci, E., and Rowse, J. 2015. Downtown curbside parking capacity. *Journal of Urban Economics* 86: 83-97.
- Du, L., and Gong, S. 2016. Stochastic Poisson game for an online decentralized and coordinated parking mechanism. *Transportation Research B* 87: 44-63.
- Geroliminis, N. 2015. Cruising-for-parking in congested cities with an MFD representation. *Economics of Transportation* 4: 156-165.
- Gross, D., Shortle, J., Thompson, J. and Harris, C. 2008. *Fundamentals of Queuing Theory*. Hoboken, NJ: Wiley.
- Levy, N., Martens, K. and Benenson, I. 2013. Exploring cruising using agent-based and analytical models of parking. *Transportmetrica* 9: 773-797 doi: 10.1080/18128602.2012.664575.
- Pierce, G. and Shoup, D. Getting the prices right: an evaluation of pricing parking by demand in San Francisco. *Journal of the American Planning Association* 79: 67-81.
- Shoup, D. 2006. Cruising for parking. *Transport Policy* 13: 479-486.
- Shoup, D. 2005. *The High Cost of Free Parking*. Chicago, IL: Planners Press.
- Vickrey, W. 1954. The economizing of curb parking space. *Traffic Engineering* 29: 62-67.



## Appendix 1: Source Code

Source code, available at <http://math.ucr.edu/~parker/CruisingForParking/>

```
from __future__ import division
import numpy as np

#model variables
#length of block
n = 100
#arrival rate
arrivalRate = 30
#stay duration
S = 2000
#stabilization cutoff. Dont record data before this timestep
sCutoff = 10000
#desired number of observations
numObs = 1000000

#car object
class car:

    def __init__(self,
                  startingLocation,
                  startingTime,
                  currentTime,
                  carId,
                  startedSearching,
                  foundParking
                  ):

```

```

self.startingLocation = startingLocation
self.presentLocation = startingLocation
self.startingTime = startingTime
self.currentTime = currentTime
self.carId = carId
self.startedSearching = False
self.foundParking = False
def startinglocation(self):
    return self.startingLocation
def presentLocation(self):
    return self.presentLocation
def startedSeartching(self):
    return self.startedSearching
def foundParking(self):
    return self.foundParking
def currentTime(self):
    return self.currentTime
def __repr__(self):
    return "Car id " \
        + str(self.carId) \
        + " started at " \
        + str(int(round(self.startingLocation))) \
        + " at time " \
        + str(self.startingTime) \
        + " Currently at " \
        + str(int(round(self.presentLocation))) \
        + " with found parking status " \
        + str(self.foundParking) \
        + " and started searching status " \
        + str(self.startedSearching)

```

```

#parking space object
class parkingSpace:

    def __init__(self,
                  location,
                  occupied,
                  emptyBy
                  ):
        self.location = location
        self.occupied = True
        self.emptyBy = np.random.exponential(S)

#objects required for simulation that are not model parameters
#array of searchtimes.
searchTimes = []
occupancyRate = []
cars = []
currentlySearching = []
parkingSpaces = []
nextArrival = np.random.exponential(arrivalRate)

#array of cars start it with 15
for i in range(0,15):
    cars.append(car(np.random.uniform(0,n),nextArrival,nextArrival,i,False,False))
    i +=1
    nextArrival = nextArrival + np.random.exponential(arrivalRate)

#array of parking spaces
for l in range(0,n):
    parkingSpaces.append(parkingSpace(l,True,np.random.exponential(S)))
    print('starting with space '

```

```

    + str(l)
    + ' filled, it will be empty by '
    + str(parkingSpaces[l].emptyBy))
step = 0
while( len(searchTimes) < numObs ):

    print("Currently on step "
          + str(step)
          + " with "
          + str(len(searchTimes))
          + " observations")
    #if we run low on cars add more
    if(len(cars) < 10):
        cars.append(car(np.random.uniform(0,n),nextArrival,nextArrival,i,False,False))
        i+=1
        nextArrival = nextArrival + np.random.exponential(arrivalRate)

    #move cars that begin to search from cars array to currentlySearching
    for c in cars:
        if((c.startingTime < step) and c.startedSearching == False):
            c.startedSearching = True
            currentlySearching.append(c)
            print("Car "
                  + str(c.carId)
                  + " has begun to search ")

    #see if the cars in current cars are on empty spaces
    for cs in currentlySearching:
        if(cs.foundParking == False) and (parkingSpaces[int(np.floor(cs.presentLocation))].emptyBy < cs.currentTime):
            #found parking
            cs.foundParking = True

```

```

#fill the space
parkingSpaces[int(np.floor(cs.presentLocation))].occupied = True
print("parking space "
      + str(np.floor(cs.presentLocation))
      + " is now taken ")
#get a new exit time
parkingSpaces[int(np.floor(cs.presentLocation))].emptyBy = (np.random.exponential(S) + cs.currentTime)
#if we are far enough along record it
if(step > sCutoff):
    searchTimes.append(step - cs.startingTime)
#output for console
print("Car "
      + str(cs.carId)
      + " found parking at time "
      + str(step)
      + " in location "
      + str(np.floor(cs.presentLocation))
      + " having searched for time "
      + str(step - cs.startingTime))
else:
    #space was full, advance the car
    print("Car "
          + str(cs.carId)
          + " advanced from location "
          + str(cs.presentLocation)
          + " to location "
          + str((cs.presentLocation + 1)%n))
    cs.presentLocation = (cs.presentLocation + 1)%n
    cs.currentTime += 1

```

```

#cleanup - remove cars that have found parking, allow new cars to merge onto street in order
currentlySearching[:] = [x for x in currentlySearching if (x.foundParking == False)]
cars[:] = [y for y in cars if (y.startedSearching == False)]
if(len(currentlySearching) != 0):
    currentlySearching = sorted(currentlySearching,key = car.presentLocation)

currentSpacesTaken = 0
#empty out spaces that were not taken
for p in parkingSpaces:
    if(p.emptyBy < step) and (p.occupied == True) :
        p.occupied = False
        print("Space "
              + str(p)
              + " is now available")
    if(p.occupied == True):
        currentSpacesTaken +=1
#reord ambient data
if(step > sCutoff):
    occupancyRate.append([currentSpacesTaken/100,len(currentlySearching)])

step +=1

searchTimesToBeSaved = np.array(searchTimes)
np.save('1Mobobservatoins23S',searchTimesToBeSaved)
occupancyRateToBeSaved = np.array(occupancyRate)
np.save('1Mobobservations23TimeSeries',occupancyRateToBeSaved)

```

## Appendix 2: Generated Simulation Data

Directory of files

On <http://www.math.ucr.edu/~parker/CruisingForParking/> there are the following files

`Cruising_For_Parking.py` This is the simulation file. Simulation parameters are at the top of the file with a comment section indicating as such. The parameters you may change are the arrival rate, the stay length, the number of spaces on the street, and the begin to record time.

`Ratio_Plot.py` This is an example of the ratio plots

The Table is a list of the data from which we worked.

Code

Filename	Output
<code>Cruising_For_Parking.py</code>	Two numpy arrays, Search times and Time Series. Contents are explained in the Data Contents Table
<code>Ratio_Plot.py</code>	An image of a ratio plot

## Data

Filename	Figure/Table in Paper	Number of Observations	Occupancy Rate	Stay Length	Arrival Rate
SearchTimes110.npy	Figure 11	1M	1/10	Exponential, Mean 2000	Poisson Rate 30
SearchTimes1112.npy	Table 4	1M	11/12	Exponential, Mean 2000	Poisson Rate 30
SearchTimes23.npy	Table 2 Figure 4	1M	2/3	Exponential, Mean 2000	Poisson Rate 30
SearchTimes23100spaces.npy	Table 5 Figure 13	1M	2/3	Exponential, Mean 2000	Poisson Rate 30
SearchTimes2310spaces.npy	Table 5 Figure 13	1M	2/3	Exponential, Mean 2000	Poisson Rate 30
SearchTimes23ConstantStay2000.npy	Table 6 Figure 14	1M	2/3	Constant 2000	Poisson Rate 30
SearchTimes23DoubleEntryHalfStay.npy	Table 6 Figure 14	1M	2/3	Exponential, Mean 1000	Poisson Rate 15
SearchTimes56.npy	Table 4	1M	5/6	Exponential, Mean 2000	Poisson Rate 30
Parameters23.npy	Figure 8	1K	2/3	Exponential, Mean 2000	Poisson Rate 30
Snapshots23.npy	Figure 10	100	2/3	Exponential, Mean 2000	Poisson Rate 30
Snapshots56.npy	Figure 10	100	5/6	Exponential, Mean 2000	Poisson Rate 24
OccupancyRate23Series.npy	Figure 5	29976647	2/3	Exponential, Mean 2000	Poisson Rate 30
NumberOfCarsSearchingVacancyRate23.npy	Table 3	100k	2/3	Exponential, Mean 2000	Poisson Rate 30



Data Content  
Data

Filename	Figure/Table in Paper	Description	Columns	Data
SearchTimes110.npy	Figure 11	Search times	1	Step – Starting time
SearchTimes1112.npy	Table 4	Search times	1	Step – Starting time
SearchTimes23.npy	Table 2 Figure 4	Search times	1	Step – Starting time
SearchTimes231000spaces.npy	Table 5 Figure 13	Search times	1	Step – Starting time
SearchTimes2310spaces.npy	Table 5 Figure 13	Search times	1	Step – Starting time
SearchTimes23ConstantStay2000.npy	Table 6 Figure 14	Search times	1	Step – Starting time
SearchTimes23DoubleEntryHalfStay.npy	Table 6 Figure 14	Search times	1	Step – Starting time
SearchTimes56.npy	Table 4	Search times	1	Step – Starting time
Parameters23.npy	Figure 8	1000 runs of the 2/3 model with 100k observations each	2	First column is mean of simulation. Second column is variance of simulation
Snapshots23.npy	Figure 10	100 snapshots of the road in the 2/3 occupancy rate model	100	Column i corresponds to parking space i, 1 if occupied 0 if unoccupied
Snapshots56.npy	Figure 10	100 snapshots of the road in the 5/6 occupancy rate model	100	Column i corresponds to parking space i, 1 if occupied 0 if unoccupied
OccupancyRate23Series	Figure 5	In the process of getting 1M observations occupancy rates were saved at each step	2	First column is occupancy rate, second is time. Note time begins after record cutoff. (10K)
NumberOfCarsSearchingVacancyRate23.npy	Table 3	Separate from the central base case, this records number of cars searching and vacancy rate for 100K time steps	3	Number of cars searching, vacancy rate, time step