City streets parking enforcement inspection decisions: The Chinese postman's perspective

Article *in* European Journal of Operational Research · April 2015

Impact Factor: 2.36 · DOI: 10.1016/j.ejor.2014.10.039

3 authors:

Nichalin S. Summerfield

University of Massachusetts Dartmouth

Moshe Dror

The University of Arizona

Morris A. Cohen

University of Pennsylvania

European Journal of Operational Research 242 (2015) 149–160

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Stochastics and Statistics

City streets parking enforcement inspection decisions: The Chinese postman's perspective

Nichalin S. Summerfield[a,*], Moshe Dror[b], Morris A. Cohen[c]

[a] Charlton College of Business, University of Massachusetts Dartmouth, North Dartmouth, MA 02747, USA
[b] Eller School of Management, University of Arizona, Tucson, AZ 85721, USA
[c] The Wharton School, University of Pennsylvania, Philadelphia, PA 19104, USA

ARTICLE INFO

Article history:
Received 17 February 2014
Accepted 17 October 2014
Available online 30 October 2014

Keywords:
Chinese postman
Routing
Simulation

ABSTRACT

We view an administrative activity of issuing parking tickets in a dense city street setting, like downtown Philadelphia or NYC, as a revenue collection activity. The task of designing parking permit inspection routes is modeled as a revenue collecting Chinese Postman Problem. After demonstrating that our design of inspection routes maximizes the expected revenue we investigate decision rules that allow the officers to adjust online their inspection routes in response to the observed parking permits' times. A simple simulation study tests the sensitivity of expected revenues with respect to the problem's parameters and underscores the main conclusion that allowing an officer to selectively wait by parked cars for the expiration of the cars' permits increases the expected revenues between 10% and 69 percent.

1. Introduction

Consider a large city street grid, like downtown Philadelphia or New York City, represented as a graph $G = (V, E)$, with parking segments along some streets in G and the common/familiar parking kiosk setting where the car owners buy parking time and place the receipt/permit on the dashboard of the car. The city administrators would like to maximize their street parking revenues by (a) collecting the parking fees from the legally parked cars – cars parked in designated parking spaces conforming to the parking times they purchased, and by (b) issuing parking tickets to cars parked in violation of the parking rules. Violation of the parking rules can take a number of forms. In our inquiry, we restrict the analysis to the time violations with respect to the parking times the car owners purchased and the parking tickets issued by the parking enforcement officers when observing parking time violations.

In order to collect the revenues from the parking violations, the city administrators usually resort to employing a crew of enforcement officers assigned to patrol the city parking areas at any given time of day and night. Consider a single parking enforcement officer's assignment. Without inferring any gender bias, we refer to the parking enforcement officer (from now on referred to as PEO) by the generic 'he'. As such, the officer is usually assigned a subgraph of city streets, say $G'(V', E') \subset G$, and he has to select (or is provided) an itinerary that traverses all the edges (it is edges since the PEO can traverse a street segment in either direction using the sidewalk) in G' where paid parking is allowed. We assume for simplicity that the subgraph G' is a connected component of G and all edges of E' have to be regularly inspected by the PEO during his patrol, both for public safety reasons and for the main function of parking permit enforcement. Again, the edges in E' correspond to the segments of the streets along which paid parking is allowed. The PEO inspects the cars' parking permits by walking along streets' sidewalks. Since a PEO can walk on any sidewalk in either direction the graphs G and G' are considered to be undirected. If paid parking is allowed on both sides of a given street segment the graph G' is a multigraph with two edges connecting the corresponding pair of nodes in V'. From now on we will refer to G' as a multigraph. In case it is desired to inspect the edges of the undirected multigraph G' with different frequencies, we add without loss of generality appropriate copies of these edges to G'. The PEO does not know in advance the number, density, and individual parking times purchased for the parked cars. As he traverses the multigraph G', he has to decide how, in what order of street segments (edges in G'), to traverse the streets' segments and at what rate; should he stop and wait next to a car whose parking time on the permit is about to expire or continue to the next car? Essentially, at each car a PEO has an option to wait, return to previously inspected cars, or continue walking to inspect the 'next' car. It corresponds to processing parking cars' information in real time and represents a real-time (online) optimization problem with the objective of collecting the maximum expected revenues from a PEO's patrol assignment. At least that

* Corresponding author. Tel.: +1 508 910 6436.
E-mail addresses: nsummerfield@umassd.edu, nichalin@yahoo.com, nichalin@email.ari (N. S. Summerfield), mdror@eller.arizona.edu (M. Dror), cohen@wharton.upenn.edu (M. A. Cohen).

http://dx.doi.org/10.1016/j.ejor.2014.10.039
0377-2217/

would be one of a number of the city administrators' objectives to consider.

Optimizing the traversal order of a multigraph's edges is not a new problem. Given a connected multigraph $G'(V', E')$ with 'length' weights $w(e)$ for each edge $e \in E'$, the problem of designing the shortest tour – a path or a circuit, that traverses each edge in E' at least once is well known under the heading of the *Chinese Postman Problem* (CPP) and dates back to 1962 (http://www.nist.gov/dads/HTML/chinesePostman.html).

In our problem of designing a traversal scheme for a single PEO, we need to define a few more concepts. We associate with each edge e in the street multigraph G' three weights; the weight $w(e)$ expresses the expected revenue collected from edge $e \in G'$, $\hat{w}(e)$ represents the expected traversal time of edge e while inspecting the cars parking permits, and $\underline{w}(e)$ is the dead-heading time for e (the traversal time with no inspections). The traversal minimization CPP problem on G' refers to the weights $\hat{w}(e)$ and dead-heading weights $\underline{w}(e)$. Our parking ticket revenue management problem is defined more formally below. Note that dead-heading edges might have to be added to E' when solving a CPP on G' as a necessary part of the CPP solution. We assume for now that in a planar graph such as a city street graph, we can construct an optimal CPP solution that traverses an edge in a dead-heading mode, when it is required, immediately after traversing it in a 'working' mode. We revisit this assumption in the paper's Summary section. Observe that with the assumption of triangle inequality time traversal matrix for G' for both 'working' mode times and dead-heading times, in an optimal CPP solution any edge in G' will have at most one dead-heading traversal.

Motivation: The potential of increased revenues due to more efficient issuing of parking tickets may constitute a non-negligible contribution to social economic well-being for many cities. Cities like New York, Philadelphia, Chicago, etc., are in great need for revenues and are desperately searching for innovative ways to raise additional revenues. The basic concept of CPP for an efficient traversal of city streets is well known to municipal managers from, for instance, planning of garbage collection operations (Beltrami & Bodin, 1974). Implementing some of the findings of this study in the daily routine operation of PEOs is rather straightforward.

The solution presented in this paper has the potential to increase revenue by about 10–69 percent. The significance of increasing parking ticket revenues by even 10 percent is invaluable for any city. Quoting from one source on parking fines (http://money.cnn.com/2004/05/03/news/parkingfinesup/): "A typical fine in Manhattan now can make your wallet $65 lighter. Parking at a fire hydrant or bus stop will run $115. The city's parking violations bureau expects to collect $562 million this year, up 48 percent from 2002. Los Angeles will collect $110 million in 2004, up 20 percent from two years ago. Angelenos endure some of the highest fines in the country; parking illegally in a disabled persons zone can draw a whopping $355 fine. Another beneficiary of higher parking fines is Chicago. Revenue has climbed 28 percent from 2002 to $141 million."

The significance of increasing parking related revenues can be illustrated by considering the city of Pittsburgh with population of about 0.3 million (http://www.census.gov/newsroom/releases/archives/2010_census/cb11-cn74.html). Based on the information from http://www.city.pittsburgh.pa.us/pghparkingauthority/assets/09_PPA_Annual_Report.pdf and http://www.post-gazette.com/pg/11248/1172336-53-0.stm?cmpid=localstate.xml, the total revenue of Pittsburgh Parking Authority from parking permit purchases and parking ticket revenues was $42 million. The city issued 280,000 tickets. The parking authority collected about $5 million from all parking-related fines with $2 million from expired meter fines. In terms of ticket-related variable cost, the facility and parking court management expenses were only $2 million (i.e. the variable cost is only 40 percent of all fines revenue). If we scale the number to match large cities, e.g. Manhattan, the 10 percent increase would translate to several million dollars.

In 2003, the city of Berkeley collected $6.9 million from parking citations, out of this amount, $2.3 million was attributed to tickets issued for expired meters (http://www.berkeleydailyplanet.com/article.cfm?archiveDate=05-14-04&storyID=18852). In 2009, the city of Milwaukee issued nearly 150,000 tickets for expired meters which would have brought in $3.3 million if all the tickets were paid (http://www.bizjournals.com/milwaukee/print-edition/2011/01/07/expired-downtown-parking-meters.html?page=all). In terms of cost information, citing the Seattle Parking Management Study of September 2002 (96 page report), "The parking ticket revenue generated by a Parking Enforcement Officer (PEO) is approximately three times the cost of labor and necessary equipment.... the average PEO generates $240 per hour in ticket revenue (collected revenue),...".

The outline of this paper is as follows: Section 2 discusses the structure of our problem relative to other problems examined within Operational Research. Section 3 describes our notation and introduces the problem of designing a PEO route over G' as a CPP. In Section 3.1 we start with the analysis of local inspection decisions regarding which car to inspect next as a function of the remaining times for the cars inspected so far on a given street segment. We begin by considering an option of waiting in front of a parked car with a valid permit anticipating its permit to expire before its owner's arrival. We denote this option as *memory size = 1*. This operational option can be extended by allowing to step back to the last previously inspected car (*memory size = 2*). The general case of allowing to step back to any previously inspected car or just inspecting a new car is examined in Section 3.1.3. A simple simulation study for the CPP routing with inspections with online local decisions is presented in Section 4. In Section 5 we present the results. In the summary and discussion sections (Section 6) we examine the related technological issues and the feasibility of implementation.

2. Related literature

In this section, we discuss two topics: previous research on related problems involving the Chinese Postman Problem, and the structure of our problem in relation to some other problems examined within Operational Research.

As mentioned before, the Chinese Postman Problem dates back to 1962 (see Dror, 2000, for the different aspects of related arc routing problems and for a more recent account see Corberán & Prins, 2010). Formally, a CPP is asking for a shortest closed circuit tour of a graph that visits (traverses) each edge and arc in that graph at least once. An optimizing solution for the CPP, if the multigraph G' is undirected (the adjacent nodes in V' are connected only by edges) or completely directed (has only arcs connecting the different nodes in V'), is obtainable in quadratic time in $|V'|$. Finding an optimal CPP solution in the case that G' is a mixed graph (some node pairs are connected by edges and some by arcs) is NP-hard (Papadimitriou, 1976).

The problem of designing an online algorithm that maximizes the expected total revenue collection in a CPP, like the one representing parking revenue collection, has not yet to our knowledge been introduced in the literature. However, a so-called *orienteering* version of a related problem has been examined by Feillet, Dejax, and Gendreau (2005). There are several other papers on similar problems. Archetti, Feillet, Hertz, and Speranza (2010) studied the Undirected Capacitated Arc Routing Problem with Profits (UCARPP). The UCARPP asks for a set of routes that maximize the total collected profit while satisfying the constraints on the route duration and on the vehicle capacity. Another capacitated variation of the orienteering CPP problem, called Maximum Benefit Chinese Postman Problem (MBCPP), was studied by Malandraki and Daskin (1993), Pearn and Wang (2003), Pearn and Chiu (2005), and recently by Corberán, Plana, Rodríguez-Chía, and Sanchis (2013). The MBCPP asks for a tour that maximizes net benefit while traversing some edges a certain number of times. Aráoz, Fernández, and Meza (2009) describe an algorithm for solving the Prize-collecting Rural Postman Problem (PRPP). Unlike CPP, the PRPP does not require traversing all edges and its profit collection on an edge is taken into account only the first time that the edge is traversed. Aráoz et al. (2009) modeled their problem version as a linear integer program.

In our problem, as a PEO traverses a road network, he has to decide how, in what order of street segments, to traverse the streets' segments and at what rate. At each car the PEO has an option to wait, return to previously inspected cars, or continue walking to inspect the next car. This setting is similar to the work of Dror and Stulman (1987) where a one dimensional robot movement control mechanism has been examined in a textile machine setting. A more elaborate examination was undertaken in the same robot movement setting by L'Ecuyer, Mayrand, and Dror (1991) modeling it as a Markov renewal decision process with a computational approach based on dynamic programming.

We note that, in the broader context of sequential resource allocation problems, the car parking enforcement inspection decisions might be viewed as a multi-armed bandit problem (see Katehakis & Veinott, 1987). This class of problem is concerned with allocating one or more resources among several alternative (competing) projects by making decisions that sacrifice current gains with the prospect of better future rewards. Such broad view is not attempted here because of the nature of the underlying combinatorial structure of the CPP.

3. Model description

Without significant loss of generality, we assume that the car parking spaces on any given street segment (segment capacity) are identical in terms of single length 'consumed' as a parking space, and we measure a parking segment's capacity in car units. Our analysis is restricted to a single PEO's subgraph G' and a time interval $[t_0, t_d]$, $t_d - t_0 = S > 0$, with S large enough to represent a PEO's working shift duration. We do not consider the problem of partitioning an entire city car parking inspection graph G into a set of single PEO assigned subgraphs G's, nor do we consider the selections of PEOs' shift durations. We note that the corresponding graph partitioning problem is an important and mathematically 'rich' topic that is left for future study.

We assume that it takes a constant time $a > 0$ to issue a parking ticket irrespective of the car location and that $a \ll S$. We use t as time value ($t \in [t_0, t_d]$). A parking space could be either empty or occupied by a car. At time t a parked car could be in either of two states: in compliance state with a valid parking permit, or in a parking violation state. Note that for a CPP it is immaterial to consider/introduce a starting node of a PEO's shift in V' and it is also immaterial if G' is a graph or a multigraph. For simplicity, we refer to G' as a graph.

Some notation and definitions:

- $G'(V', E')$: a connected subgraph with $|V'| = v > 1$ nodes and $|E'|$ edges.
- n_e: number of parking spaces, ordered from 0 to $n_e - 1$, available on edge $e \in E'$. ($n_e \in \mathbb{N}_0$.)
- N: total number of parking spaces in G'. $N = \sum_{e \in E'} n_e$.
- $Z_{ie}(t)$: an a priori likelihood (probability) that parking space $i \in \{0, 1, \ldots, n_e - 1\}$ on edge e is empty at time t. We assume for now that $Z_{ie}(t) = z$ for all t, i, and e. That is, every parking space has the same likelihood to be empty. $Z_{ie}(t)$ is a Bernoulli distribution. If we take a snapshot of $G'(V', E')$ at any time t, there will be zN empty parking spaces and $(1 - z)N$ occupied parking spaces.
- $P_{ie}(t)$: a probability that a car in a non-empty parking space i on edge e is in a parking violation state at time t. The rationale for i, e, and t dependencies is that certain parking spaces may be more likely to have a car in violation state in some times of day. We assume that all owners of cars in parking space i on edge e share the same probability distribution regarding a car owner's return time with respect to permit duration and all permits at any location are of the same time duration L. Note that $Z_{ie}(t)$ and $P_{ie}(t)$ are two independent probability density functions. $P_{ie}(t)$ can be calculated as shown below.

Consider car c in parking space i on edge e at time t. Let x be a continuous random variable representing an owner's return time after parking, and its probability density function is given by $v_{iet}(x)$ with a finite support $[0, X_{iet}]$. That is, X_{iet} is the latest possible time that this owner may return to her car. Let the parking time purchased by the owner (length of permit) be L. The permit expires after L. We assume that $L \le X_{iet}$. Note that x can map to t by adding the time the car arrived t_c to x.

Let y be a continuous random variable representing the time that the PEO arrives by the parked car for inspection. Because the PEO traverses the graph $G'(V', E')$ several times a day, and cars may arrive and leave many times in each parking space, we assume that the probability density function of y is a uniform distribution $u(y)$ between 0 and Y, and $Y \ge X_{ie}$. Probability density functions $v_{iet}(x)$ and $u(y)$ are independent. Note that y can also map to t by adding the time the car arrived t_c to y.

The event of the PEO's arrival is equivalent to drawing y randomly. To calculate $P_{ie}(t)$, we are only interested in the probability that is conditional upon the parking space being non-empty, i.e., $y < x$.

$$P_{ie}(t) = \Pr(y > L | y < x) = \frac{\Pr(y > L, y < x)}{\Pr(y < x)} = \frac{\int_L^{X_{iet}} \int_L^x v_{iet}(x) u(y) \mathrm{d}y \mathrm{d}x}{\int_0^{X_{iet}} \int_0^x v_{iet}(x) u(y) \mathrm{d}y \mathrm{d}x}.$$

Without loss of generality we assume that $v_{iet}(x) = v(x)$ and $X_{iet} = X$. Hence, $P_{ie}(t)$ is the same for all t, i, and e. $P_{ie}(t) = p$ where

$$p = \frac{\int_L^X \int_L^x v(x) u(y) \mathrm{d}y \mathrm{d}x}{\int_0^X \int_0^x v(x) u(y) \mathrm{d}y \mathrm{d}x}. \tag{1}$$

If we take a snapshot of $G'(V', E')$ at any time t, the expected number of cars in violation state is $(1 - z) \sum_{e \in E'} \sum_{i=0}^{n_e - 1} P_{ie}(t) = (1 - z)pN$.

More notation:

- $\xi_{ie}(t)$: the time information, observed at t, about how long has car i been parked in parking space i on edge e. If car c arrives for parking in parking space i on edge e at $t_c < t$, $\xi_{ie}(t) = t - t_c$. $\xi_{ie}(t) \ge 0$. If the parking space is empty, $\xi_{ie}(t) \equiv 0$.
- $s_{ie}(t)$: the 'observed' time at t until parking permit's expiration for a car in parking space i on edge e. If car c arrives for parking in parking space i on edge e at $t_c < t$, $s_{ie}(t_c) = L$ and $s_{ie}(t) = L - \xi_{ie}(t) \ge 0$. $s_{ie}(t) = 0$ implies that at time t the car in parking space i on e is in a parking violation state. If the parking space is empty, $s_{ie}(t) \equiv \infty$.

Given that a car is parked in parking space i on edge e at time t', with either compliance state or violation state, the probability that the car's owner will return to the car on or before time $t'' > t'$ is $H_{ie}(t', t'')$ where

$$H_{ie}(t', t'') = \Pr(x \le \xi_{ie}(t'') | x > \xi_{ie}(t')) = \frac{\Pr(x \le \xi_{ie}(t''), x > \xi_{ie}(t'))}{\Pr(x > \xi_{ie}(t'))} = \frac{\int_{\xi_{ie}(t')}^{\xi_{ie}(t'')} v(x) \mathrm{d}x}{\int_{\xi_{ie}(t')}^{X} v(x) \mathrm{d}x}. \tag{2}$$

Given that a car is in a compliance state at time t', the probability that the car's owner will return to the car on or before switching to aparking violation state is $\hat{H}_{ie}(t')$.

$$\hat{H}_{ie}(t') = H_{ie}(t', t' + s_{ie}(t')) = \Pr(x \le L | x > \xi_{ie}(t')) = \frac{\int_{\xi_{ie}(t')}^{L} v(x)\mathrm{d}x}{\int_{\xi_{ie}(t')}^{X} v(x)\mathrm{d}x}. \quad (3)$$

Proposition 1. *Assume that at a given time $t \in [t_0, t_d]$ the likelihood of an expired permit (a parking violation state) at any parked car in G is represented by probability $P_{ie}(t) = p$ and the likelihood of an empty parking space in G is represented by probability $Z_{ie}(t) = z$. Then, given that the PEO has no memory regarding the parking compliance time left for the cars he inspected so far, it is optimal for the PEO to follow the path found as the optimal solution to corresponding CPP on subgraph G'. That is, to traverse G' in the shortest time.*

Proof. The PEO has no memory regarding the parking compliance time left for the cars on the path that he has been traversing so far, $P_{ie}(t) = p$, and $Z_{ie}(t) = z$ for all t, i, and $e \in G'$. The PEO will traverse a segment of a CPP solution in a dead-heading mode as dictated by the optimal CPP solution after the segment has already been inspected for parking violations in that CPP cycle. When traversing segment e in a dead-heading mode the PEO does not inspect the permit status of cars parked along e.

Let each parking ticket issued to a car in violation state generate a revenue of r dollars. Hence, the expected revenue collected from any single parking space is $(1 - z)pr$ dollars per tour cycle. Thus, by strictly executing the optimal CPP traversal solution, the PEO generates an expected revenue of $(1 - z)prN$ from each tour cycle.

Assume that to observe the parking status of a car (inspecting the parking permit) takes a constant time $b > 0$, it takes a constant time $a > 0$ to issue a parking ticket, and a time $\underline{w}(e)$ to dead-head through the edge $e \in E'$. Thus, $\hat{w}(e) = \underline{w}(e) + (1 - z)(pa + b)n_e$ is the time required to inspect and issue parking tickets along the edge $e \in G'$. Assume also that the PEO executes an optimal CPP tour that may have dead-heading edges. Let $\alpha_e^* \ge 1$ represent the number of times edge e is used in one optimal CPP tour. Thus, the expected tour time is $\sum_{e\in E'} \alpha_e^* \underline{w}(e) + (1 - z)(pa + b)N$. During the PEO's shift duration S, he would expect to execute $\frac{S}{\sum_{e\in E'} \alpha_e^* \underline{w}(e) + (1-z)(pa+b)N}$ tour cycles. Therefore the expected revenue per shift would be $\frac{(1-z)SprN}{\sum_{e\in E'} \alpha_e^* \underline{w}(e) + (1-z)(pa+b)N}$ dollars.

Consider the case of the PEO deciding to inspect the graph G by traversing G in a suboptimal CPP fashion of a longer time duration than the optimal CPP tour. For simplicity assume that the suboptimal tour requires additional dead-heading edges with $\alpha_e \ge \alpha_e^*$ representing the number of times edge e is traversed in the suboptimal tour versus the optimal one. The expected revenue for each tour cycle is still $(1 - z)prN$. The expected suboptimal tour time is $\sum_{e\in E'} \alpha_e \underline{w}(e) + (1 - z)(pa + b)N$. Thus, during the PEO's shift duration S, he would expect to generate $\frac{(1-z)SprN}{\sum_{e\in E'} \alpha_e \underline{w}(e) + (1-z)(pa+b)N}$ dollars per shift.

Since $\alpha_e \ge \alpha_e^*$, the revenue of generated by the suboptimal CPP tour is lower than or equal to the optimal CPP. □

For notation simplicity, let $E_{r/t} = \frac{(1-z)prN}{\sum_{e\in E'} \alpha_e^* \underline{w}(e) + (1-z)(pa+b)N}$ represent the expected revenue per unit time given an expected duration of optimal CPP solution.

3.1. Local decisions with limited record of remaining compliance times

3.1.1. Memory size = 1

Now assume that the PEO observes the compliance time left for the car just inspected and has the option of waiting by the car for the parking permit to expire before its owner returns or continue to the next car (Fig. 1, left). (A PEO may remember the compliance time left on more than one car. However, in this section, we assume a memory of 1 thus not allowing the PEO's return to a previously inspected car.)

Consider the case that the PEO inspects car c in parking space i on edge e at time t' where $s_{ie}(t') > 0$ time left until its parking permit expires. Once the parking permit expires, the PEO spends $a > 0$ time issuing a ticket. The expected revenue for waiting $s_{ie}(t')$ unit time and issuing the ticket is $r(1 - \hat{H}_{ie}(t'))$.

Clearly, if the owner of the car comes back before the parking permit expires and the PEO is waiting in front of it, the PEO would at that point continue to the next car on his CPP tour. But if the owner of the car comes back while the PEO is still issuing the ticket, the owner may not convince the PEO to discard the ticket. The waiting time function can be defined as

$$w_{iet'}(x) = \begin{cases} s_{ie}(t') + a & \text{if } x > L \\ x - \xi_{ie}(t') & \text{if } x \le L \end{cases} \quad (4)$$

where x is a continuous random variable representing the car owner's return time after parking.

Hence, the expected time for waiting in front of car c is

$$(s_{ie}(t') + a)(1 - \hat{H}_{ie}(t')) + \int_{\xi_{ie}(t')}^{L} (x - \xi_{ie}(t'))v(x|x > \xi_{ie}(t'))\mathrm{d}x$$

where $v(x|x > \xi_{ie}(t'))$ is essentially a truncated distribution of $v(x)$ with support $(\xi_{ie}(t'), X]$. Thus, we divide the resulting distribution by 1 minus the integral up to the truncation point $\xi_{ie}(t')$ to 'normalize' the distribution results.

For instance, let $V(x)$ be a cumulative probability distribution of $v(x)$.

$$v(x|x > \xi_{ie}(t')) = \begin{cases} \dfrac{2x}{(1 - V(\xi_{ie}(t')))mX} & \text{if } 0 \le x \le m \\ \dfrac{2(X - x)}{(1 - V(\xi_{ie}(t')))(X - m)X} & \text{if } m < x \le X \end{cases} \quad (5)$$

We consider the expected revenue per unit time in comparison to the expected revenue per unit time from traversing the CPP path without waiting at car c, which is $E_{r/t}$. Therefore, the PEO should wait in front of car c only if

$$\frac{r(1 - \hat{H}_{ie}(t'))}{(s_{ie}(t') + a)(1 - \hat{H}_{ie}(t')) + \int_{\xi_{ie}(t')}^{L} (x - \xi_{ie}(t'))v(x|x > \xi_{ie}(t'))\mathrm{d}x} \ge E_{r/t} \quad (6)$$

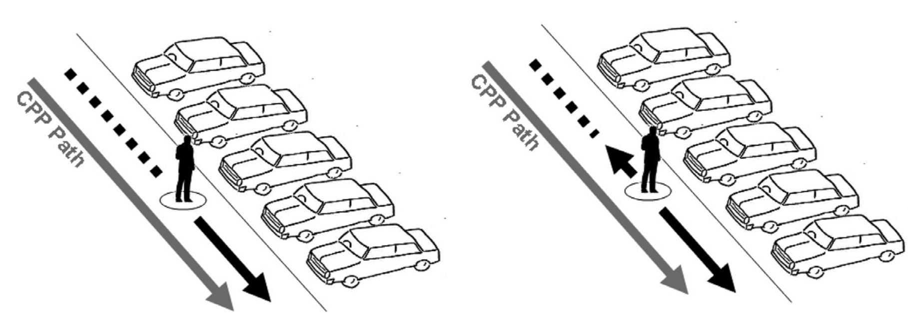


Fig. 1. (Left) Options to wait or continue. (Right) Options to wait, step back to the previous car, or continue.

By assuming the same distribution of $v(x)$ for all cars, we can calculate the time s_{limit} such that the PEO should wait by the car at i only if $s_{ie}(t) < s_{\text{limit}}$.

Consistency check. Suppose that it is optimal for the PEO to wait at time t'. Then, it is also optimal for him to wait at time $t' + \delta$ where $0 < \delta < s_{ie}(t')$ and $s_{ie}(t' + \delta) > 0$. Because $s_{ie}(t') > s_{ie}(t' + \delta)$,

$$\hat{H}_{ie}(t') > \hat{H}_{ie}(t' + \delta) \tag{7}$$

$$\int_{\xi_{ie}(t')}^{L} (x - \xi_{ie}(t'))v(x|x > \xi_{ie}(t'))\mathrm{d}x > \int_{\xi_{ie}(t'+\delta)}^{L} (x - \xi_{ie}(t' + \delta))v(x|x > \xi_{ie}(t' + \delta))\mathrm{d}x \tag{8}$$

Note that the expected revenue per shift of duration S given the option of waiting by cars with $s_{ie}(t) > 0$ is greater or equal to the expected revenue per shift without the option of waiting.

3.1.2. Memory size = 2

Next, consider the case when the PEO remembers the parking permit's expiration time for the car inspected last before the present car (Fig. 1, right). In this case, the PEO has the choice to wait at the current car, to step back to the previous car, or to move to the next car. For simplicity, assume that once he moves on to the next parking segment (different edge in G'), he cannot recall the compliance times left for any of the cars on the previous segment.

Denote the time taken for stepping back to the previous car as $\lambda > 0$ (approximately, $\lambda = \underline{w}(e)/n_e$ unit time) and consider that the PEO inspects car i on edge e at time t'. Suppose that cars at i and $i - 1$ have $s_{ie}(t') > 0$ and $s_{i-1,e}(t')$ time left until the parking permits expire, respectively. Should the PEO wait in front of car i or car $i - 1$, or not wait at all? If he waits at car i and issues a ticket, should he still consider waiting at car $i - 1$ afterward?

In our online heuristic rules we assume that the PEO makes two sequential decisions. The first decision is to choose among cars $i - 1$, i, and $i + 1$, the next parking space that has not been inspected yet. If he walks to car $i - 1$, waits, and issues a ticket, then he will make a second decision choosing between the remaining two cars, i and $i + 1$. If he walks to car i, waits, and issues a ticket, then he will have to decide between the remaining two cars, $i - 1$ and $i + 1$. Each of the two decisions will attempt to maximize the expected revenue per unit time.

To calculate the expected revenue per unit time for the option of pursuing car $i - 1$, we divide the expected revenue by the expected time necessary for pursuing car $i - 1$. At time t', if the remaining time on car $i - 1$'s permit $s_{i-1,e}(t')$ is greater than λ which is the time to reach $i - 1$, then the expected revenue is the ticket revenue $r(1 - \hat{H}_{i-1,e}(t'))$ or $r(1 - H_{i-1,e}(t', t''))$ where $t'' = t' + s_{i-1,e}(t')$. If the remaining time on the permit is less than the time the PEO takes to walk to it, then the expected revenue is the ticket revenue r times 1 minus the probability of the car owner arrival between t' and $t'' = t' + \lambda$ which is $r(1 - H_{i-1,e}(t', t''))$. Hence, the expected revenue can be written as $r(1 - H_{i-1,e}(t', t''))$ where $t'' = t' + \max\{s_{i-1,e}(t'), \lambda\}$.

Consider the first decision. The function for the PEO's time spent on car $i - 1$, corresponding to the owner's return time after parking, can be defined as:

$$w_{i-1,et'}(x) = \begin{cases} 2\lambda & \text{if } x \le \xi_{i-1,e}(t' + \lambda) \\ 2\lambda + x - \xi_{i-1,e}(t' + \lambda) & \text{if } \xi_{i-1,e}(t' + \lambda) < x \text{ and } x \le L \\ 2\lambda + s_{i-1,e}(t' + \lambda) + a & \text{if } \xi_{i-1,e}(t' + \lambda) < x \text{ and } x > L \end{cases} \tag{9}$$

That is, if the owner of car $i - 1$ walks back to the car before the PEO walks to the car ($x \le \xi_{i-1,e}(t' + \lambda)$), the PEO would have spent in total of 2λ unit time which is the time walking to car $i - 1$ and from car $i - 1$ back to car i. If the owner of car $i - 1$ arrives while the PEO is waiting in front of the car ($\xi_{i-1,e}(t' + \lambda) < x \le L$), the PEO would have spent walking time plus the waiting time until the owner comes back. If the owner of car $i - 1$ arrives after the permit expired, the PEO would have spent walking time plus waiting time and plus the ticket issuing time.

The expected revenue per unit time of car $i - 1$ is

$$E_{i-1} = \frac{r(1 - H_{i-1,e}(t', t''))}{E_x[w_{i-1,et'}(x)]}$$

where $E_x[w_{i-1,et'}(x)] = 2\lambda + \int_{\xi_{i-1,e}(t'+\lambda)}^{L} (x - \xi_{i-1,e}(t' + \lambda))v(x|x > \xi_{i-1,e}(t'))\mathrm{d}x + (s_{i-1,e}(t' + \lambda) + a)(1 - H_{i-1,e}(t', t''))$.

We compare it with the expected revenues per unit time at car i which is

$$E_i = \frac{r(1 - \hat{H}_{ie}(t'))}{(s_{ie}(t') + a)(1 - \hat{H}_{ie}(t')) + \int_{\xi_{ie}(t')}^{L} (x - \xi_{ie}(t'))v(x|x > \xi_{ie}(t'))\mathrm{d}x}.$$

Hence, the PEO should wait at car i if

$E_i \ge E_{r/t}$
and $E_i \ge E_{i-1}$.

On the other hand, the PEO should wait at car $i - 1$ if

$E_{i-1} \ge E_{r/t}$
and $E_{i-1} \ge E_i$.

Once the above outcome occurs (the PEO waits) then the next decision will be considered/evaluated when either a parking violation ticket has been issued or the car owner returns to the corresponding car before its parking permit expiration time. Clearly, the availability of the option of stepping back leads to higher expected revenue than the previous case with the options to wait or continue since it subsumes the former.

3.1.3. Memory size = n_e. Remembering expiration times for all the cars on a given segment

Now consider the case of a PEO who remembers the expiration times for all the cars inspected on edge e. The PEO can also visually observe which of the inspected cars are still parked and which have left their parking spaces. In this case, he has to choose whether to wait at the current car, to step back to any of the previous cars on that edge, or to walk to the next car on the current edge. As before, assume that it takes $\lambda = \underline{w}(e)/n_e$ to walk to the next car on the same edge.

At each car in a compliance state, the PEO can go back to any of the previously inspected cars, stay put at the current car, or continue to inspect the next car. This setting has a very similar flavor to the work of Dror and Stulman (1987).

Consider the following case. Say the PEO is currently positioned at car j. Let car at $i < j$ have $s_{ie}(t') > 0$ time left before its permit expires. The PEO should walk back to car i only if the expected revenue per unit time of pursuing car i is higher than the expected revenue per unit time earned when following the CPP path.

To calculate the expected revenue per unit time for the option of pursuing car i, we divide the expected revenue by the expected time necessary for pursuing car i. At time t', if the remaining time on car i's permit $s_{ie}(t')$ is greater than $\lambda|j - i|$ which is the time to reach i, then the expected revenue is the ticket revenue $r(1 - \hat{H}_{ie}(t'))$ or $r(1 - H_{ie}(t', t''))$ where $t'' = t' + s_{ie}(t')$. If the remaining time on the permit is less than the time the PEO takes to walk to it, then the expected revenue is the ticket revenue r times 1 minus the probability of the car owner arrival between t' and $t'' = t' + \lambda|j - i|$ which is $r(1 - H_{ie}(t', t''))$. Hence, the expected revenue can be written as $r(1 - H_{ie}(t', t''))$ where $t'' = t' + \max\{s_{ie}(t'), \lambda|j - i|\}$.

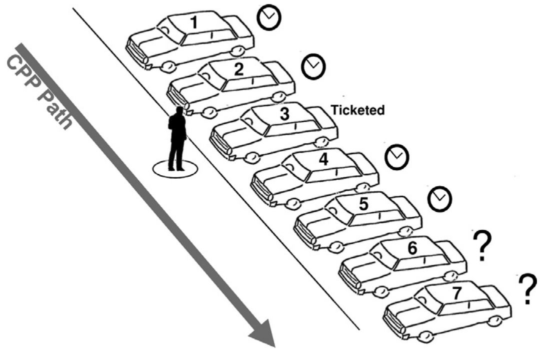


Fig. 2. Options to step to previously inspected cars, or continue.

Given that the PEO is at car j at time t', the function for the PEO's time spent on car i, corresponding to the owner's return time after parking, can be defined as:

$$w_{iet'}(x) = \begin{cases} 2\lambda|j-i| & \text{if } x \le \xi_{ie}(t'+\lambda|j-i|) \\ 2\lambda|j-i| + x - \xi_{ie}(t'+\lambda|j-i|) & \text{if } \xi_{ie}(t'+\lambda|j-i|) < x \text{ and } x \le L \\ 2\lambda|j-i| + s_{ie}(t'+\lambda|j-i|) + a & \text{if } \xi_{ie}(t'+\lambda|j-i|) < x \text{ and } x > L \end{cases} \tag{10}$$

The expected time spent for the option of pursuing car i while the PEO is at j is

$$2\lambda|j-i| + \int_{\xi_{ie}(t'+\lambda|j-i|)}^{L} (x - \xi_{ie}(t'+\lambda|j-i|))v(x|x > \xi_{ie}(t'))\mathrm{d}x \\ + (s_{ie}(t'+\lambda|j-i|) + a)(1 - H_{ie}(t', t' + \max\{s_{ie}(t'), \lambda|j-i|\})).$$

Hence, the PEO should walk back to car i if the expected revenue per unit time spent is greater than or equal to $E_{r/t}$.

Now, consider when the PEO has more options than just walking to a certain car as shown in Fig. 2.

Previously, the PEO was positioned at car 5 (he had inspected cars 1, 2, 3, 4, and 5, but no ticket was issued) but has decided to walk back to car 3. Say he issued a ticket to car 3. At this time t', he knows of the time left on cars 1, 2, 4, and 5. He may choose to walk to any of them or continue to car 6, which he has not yet inspected. The simplest rule is for him to go to the car that promises the highest expected revenue per unit time. The calculation has to take care not to double count the walking time. That is, when he was at car 5 and decided to walk back to car 3, he has already accounted for the time walking from car 5 to car 3 and back from car 3 to car 5. Therefore, when he is at car 3 and evaluating this rule for car 5, he should not include the time walking from car 3 to car 5. Hence, at time t', he can calculate the expected time spent at cars 1, 2, 4, and 5 relative to his current location (car 3) as:

$$\begin{aligned} \text{car 1:}\quad & 2\lambda|3-1| + \int_{\xi_{1e}(t'+\lambda|3-1|)}^{L} (x - \xi_{1e}(t'+\lambda|3-1|))v(x|x > \xi_{1e}(t'))\mathrm{d}x \\ & + (s_{1e}(t'+\lambda|3-1|) + a)(1 - H_{1e}(t', t' + \max\{s_{1e}(t'), \lambda|3-1|\})). \\ \text{car 2:}\quad & 2\lambda|3-2| + \int_{\xi_{2e}(t'+\lambda|3-2|)}^{L} (x - \xi_{2e}(t'+\lambda|3-2|))v(x|x > \xi_{2e}(t'))\mathrm{d}x \\ & + (s_{2e}(t'+\lambda|3-2|) + a)(1 - H_{2e}(t', t' + \max\{s_{2e}(t'), \lambda|3-2|\})). \\ \text{car 4:}\quad & 2\lambda|3-4| + \int_{\xi_{4e}(t'+\lambda|3-4|)}^{L} (x - \xi_{4e}(t'+\lambda|3-4|))v(x|x > \xi_{4e}(t'))\mathrm{d}x \\ & + (s_{4e}(t'+\lambda|3-4|) + a)(1 - H_{4e}(t', t' + \max\{s_{4e}(t'), \lambda|3-4|\})). \\ \text{car 5:}\quad & 2\lambda|3-5| + \int_{\xi_{5e}(t'+\lambda|3-5|)}^{L} (x - \xi_{5e}(t'+\lambda|3-5|))v(x|x > \xi_{5e}(t'))\mathrm{d}x \\ & + (s_{5e}(t'+\lambda|3-5|) + a)(1 - H_{5e}(t', t' + \max\{s_{5e}(t'), \lambda|3-5|\})). \end{aligned} \tag{11}$$

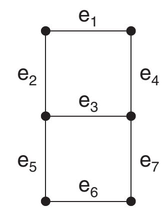


Fig. 3. Selected graph.

Using our heuristic rules, the PEO will choose the car with the highest expected revenue per unit time. Below we describe a numerical experiment – a simple simulation study – that 'calculates' the expected revenue per shift when implementing our heuristic rules.

4. Numerical experiment

4.1. Selected graph and parameters

We chose the graph depicted in Fig. 3 for our numerical experiment. This graph could be translated into a street grid that surrounds two city blocks and cars are allowed to park on one side of each street. The graph in Fig. 3 is not Eulerian. We have to dead-head e_3 to transform this graph into an Eulerian graph to allow for the construction of an optimal CPP solution. The optimal CPP solution requires the PEO to walk from the top-left node and follow the tour, $e_1 \to e_4 \to e_3 \to e_3$ (reverse) $\to e_7 \to e_6 \to e_5 \to e_2$. He traverses e_3 in reverse direction in a dead-heading mode.

In the experiment we assume a constant number of cars on each edge. Specifically, we tested with n_e equals 25. That is, there are 175 parking spaces in total in the graph. The parameters that we used in our numerical experiment are as follows:

- Each parking permit is valid for exactly 60 minutes ($L = 60$).
- Each violation ticket generates a revenue of $r = 30$ dollars.
- The PEO's shift length S is 8 hours.
- The PEO spends $b = 0.5$ minutes inspecting the time left on each parking permit.
- The PEO spends $a = 5$ minutes issuing a ticket.
- Dead-heading time $\underline{w}(e_k)$ on any edge of the graph is 6.25 minutes ($\lambda = 0.25$).

These parameters were selected based on observing parking rules in various cities.

We experiment with a PEO remembering expiration times for various numbers of parked cars, Bernoulli probability of empty parking spaces parameter z and time length of permits L.

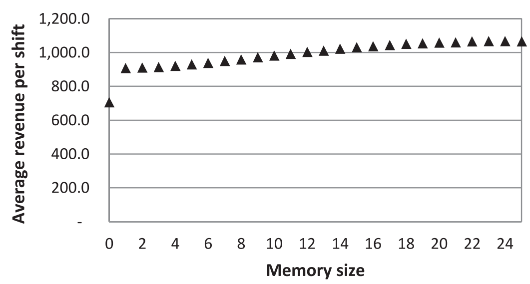
For simplicity, we chose a triangle distribution to represent the distribution of the time between the parking of a car and the return of its owner, with a mode m as 55, which is 5 minutes before the car turns into violation state and the end point X is 90, which is 30 minutes after the car turns into violation state.

5. Results and discussion

We simulated 1000 one-shift rounds for each parameter's setting. In each round, a PEO traverses the graph for 8 hours (one shift). We examine the effects of memory size, Bernoulli probability of empty spaces on average revenue per shift z and permit length L. The revenue per parking ticket is set to \$30.

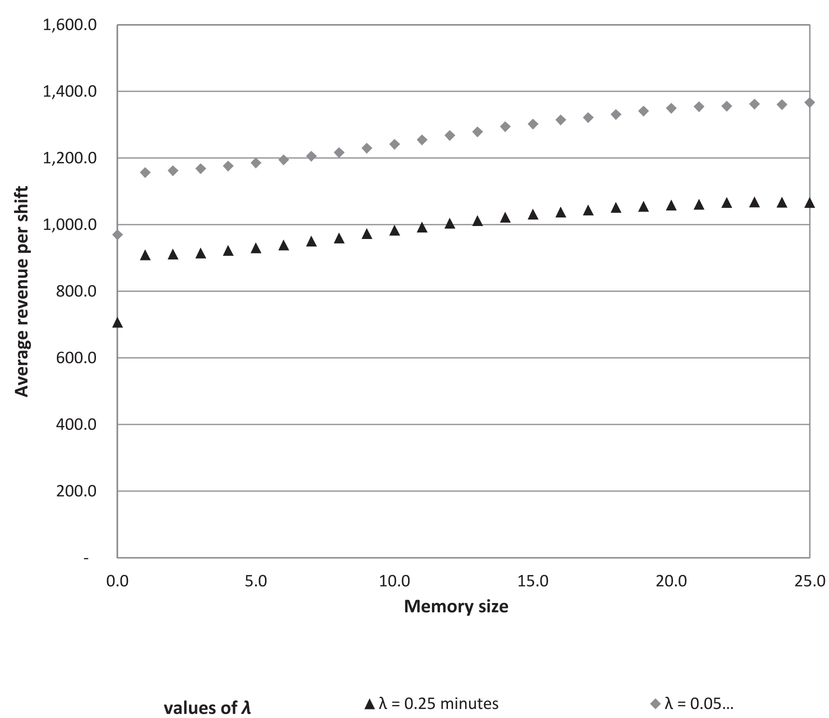
5.1. Effect of the memory size

To observe the effects of memory size, we chose $Z_{ie}(t) = z = 0.30$. The computed p is 0.0591. We varied the memory size from 0 (no



Memory size	Average Revenue per shift	Std. Dev	Memory size	Average Revenue per shift	Std. Dev	Memory size	Average Revenue per shift	Std. Dev
0	708.0	110.2	10	984.2	96.9	20	1,059.5	99.7
1	910.3	96.1	11	993.7	98.9	21	1,062.1	98.9
2	912.7	92.8	12	1,005.1	98.5	22	1,067.5	97.5
3	915.8	90.9	13	1,012.8	100.8	23	1,068.9	97.0
4	923.3	92.1	14	1,023.3	98.8	24	1,068.5	96.9
5	931.2	92.7	15	1,032.4	100.3	25	1,067.6	100.3
6	939.6	94.8	16	1,038.8	100.1			
7	951.1	96.7	17	1,045.1	100.8			
8	961.0	98.7	18	1,052.7	100.7			
9	974.1	98.1	19	1,055.9	102.2			

Fig. 4. Effect of the memory size.



	Average Revenue per shift for different λ	
Memory size	λ = 0.25 minutes	λ = 0.05 minutes
0.0	708.0	969.8
1.0	910.3	1,156.9
2.0	912.7	1,162.6
3.0	915.8	1,168.4
4.0	923.3	1,176.6
5.0	931.2	1,185.4
6.0	939.6	1,195.2
7.0	951.1	1,206.0
8.0	961.0	1,217.1
9.0	974.1	1,230.1
10.0	984.2	1,241.9
11.0	993.7	1,255.1
12.0	1,005.1	1,268.3
13.0	1,012.8	1,279.0
14.0	1,023.3	1,294.7
15.0	1,032.4	1,302.6
16.0	1,038.8	1,314.5
17.0	1,045.1	1,322.0
18.0	1,052.7	1,331.5
19.0	1,055.9	1,341.3
20.0	1,059.5	1,349.9
21.0	1,062.1	1,354.9
22.0	1,067.5	1,356.2
23.0	1,068.9	1,362.7
24.0	1,068.5	1,360.5
25.0	1,067.6	1,366.7

Fig. 5. Effect of λ.

memory, no option to wait) to 25 (the PEO can memorize the parking permit time of all 25 cars on an edge.) We ran 1000 rounds of experiment for each one of the 25 memory options. Given our heuristic rules, we expected to see an increase in average revenue per shift as the memory size increases.

Fig. 4 shows the result of our experiment. When the memory size is 0, the simulation shows the average (over a thousand rounds of simulation) revenue per shift to be $708.00 which is very close to the computed expected revenue per shift of $707.16. The simulated PEO went through an average of 572.64 parking spaces in an 8-hour shift, out of these, 172.16 parking spaces were empty (30 percent). Among the occupied spaces, 23.60 cars were in violation state (approx. 5.9 percent). The simulated PEO took approximately 3.26 tour cycle per shift.

Increasing the memory size to 1 (the PEO can wait at a car but cannot walk back to previous cars), we noted a 28.57 percent improvement ($202.26) in the average revenue per shift. The subsequent increases did not show significant improvement in the average revenue per shift. To understand this result, we observed the number of times the PEO waited at a car or walked back to previous cars.

Table 1 shows the average number of times per shift that the PEO waited at a car or walked back to previous cars for each given memory size. For instance, when the memory size was three cars, the PEO waited at the car he was inspecting 20.34 times per shift, walked back one space to a previous car 2.21 times per shift, and walked back two spaces 2.04 times per shift. In total, the PEO waited or walked back 24.58 times per shift. The average revenue per shift was $915.84.

Notice that the PEO rarely walked back far. That is, the expected revenue per unit time on walking back further is rarely better than the expected revenue per unit time of traversing the normal CPP path. This is because the longer it takes for the PEO to walk back, the higher the expected time spent; and the higher the chance that the car owner comes back before the PEO reaches the car, and the lower the expected revenue. One may ask how can we improve with respect to the average revenue as the memory increase. What happens if the PEO walks faster or rides a Segway? In this case, λ decreases. See Fig. 5 for the result with $\lambda = 0.05$. We don't see a significant changes in the improvement pattern. The increase in the revenue seems to come from the effect of *lambda*, but not the memory size.

Table 1
Average number of times per shift that the PEO walked back to previous cars.

	Number of parking spaces the officer walked back																										
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Total	Average revenue
Memory size	Average number of times that the officer walked back per shift																										
0	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	708.00
1	22.19	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	22.19	910.26
2	21.23	2.21	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	23.44	912.72
3	20.34	2.21	2.04	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	24.58	915.84
4	19.56	2.19	2.02	1.93	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	25.70	923.25
5	18.74	2.21	2.03	2.02	1.87	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	26.86	931.17
6	18.06	2.21	2.06	2.01	1.88	1.74	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	27.96	939.63
7	17.45	2.21	2.06	2.05	1.95	1.73	1.62	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	29.06	951.09
8	16.86	2.19	2.09	2.03	1.89	1.72	1.67	1.56	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	30.01	960.96
9	16.32	2.16	2.07	2.03	1.93	1.79	1.69	1.61	1.40	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	30.99	974.07
10	15.81	2.18	2.08	2.08	1.93	1.81	1.66	1.55	1.42	1.36	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	31.87	984.24
11	15.37	2.20	2.08	2.05	1.90	1.79	1.68	1.55	1.474	1.381	1.175	–	–	–	–	–	–	–	–	–	–	–	–	–	–	32.66	993.69
12	15.05	2.17	2.07	2.04	1.92	1.76	1.68	1.58	1.400	1.382	1.203	1.083	–	–	–	–	–	–	–	–	–	–	–	–	–	33.32	1,005.09
13	14.71	2.18	2.12	2.07	1.89	1.78	1.66	1.57	1.442	1.376	1.176	1.097	0.887	–	–	–	–	–	–	–	–	–	–	–	–	33.94	1,012.83
14	14.45	2.17	2.06	2.07	1.87	1.82	1.68	1.59	1.468	1.392	1.173	1.114	0.931	0.851	–	–	–	–	–	–	–	–	–	–	–	34.64	1,023.33
15	14.18	2.17	2.06	2.11	1.86	1.81	1.75	1.59	1.471	1.392	1.192	1.097	0.904	0.840	0.672	–	–	–	–	–	–	–	–	–	–	35.09	1,032.42
16	14.01	2.12	2.07	2.12	1.93	1.81	1.69	1.57	1.469	1.337	1.178	1.077	0.920	0.843	0.690	0.615	–	–	–	–	–	–	–	–	–	35.44	1,038.81
17	13.86	2.15	2.07	2.03	1.90	1.80	1.73	1.57	1.460	1.329	1.196	1.119	0.932	0.847	0.724	0.667	0.557	–	–	–	–	–	–	–	–	35.94	1,045.08
18	13.69	2.14	2.07	2.05	1.95	1.79	1.67	1.54	1.471	1.391	1.207	1.108	0.948	0.876	0.740	0.693	0.560	0.43	–	–	–	–	–	–	–	36.32	1,052.70
19	13.57	2.16	2.08	1.98	1.90	1.76	1.69	1.56	1.462	1.378	1.235	1.164	0.974	0.861	0.740	0.707	0.563	0.44	0.35	–	–	–	–	–	–	36.59	1,055.91
20	13.48	2.16	2.06	2.04	1.95	1.82	1.69	1.57	1.488	1.375	1.252	1.094	0.957	0.858	0.744	0.659	0.554	0.43	0.35	0.25	–	–	–	–	–	36.78	1,059.45
21	13.44	2.20	2.09	2.03	1.92	1.76	1.69	1.55	1.471	1.380	1.243	1.117	0.992	0.855	0.745	0.631	0.549	0.42	0.37	0.28	0.19	–	–	–	–	36.92	1,062.06
22	13.39	2.12	2.07	2.00	1.87	1.78	1.67	1.58	1.471	1.352	1.265	1.124	0.978	0.884	0.762	0.639	0.568	0.45	0.40	0.30	0.20	0.17	–	–	–	37.03	1,067.49
23	13.38	2.15	2.08	1.99	1.87	1.77	1.69	1.59	1.466	1.389	1.259	1.093	0.979	0.881	0.756	0.644	0.561	0.46	0.39	0.31	0.21	0.17	0.06	–	–	37.12	1,068.90
24	13.36	2.15	2.06	1.99	1.87	1.79	1.67	1.59	1.444	1.405	1.232	1.093	1.001	0.860	0.768	0.654	0.560	0.45	0.38	0.30	0.21	0.15	0.06	0.04	–	37.09	1,068.45
25	13.33	2.16	2.08	2.01	1.90	1.79	1.65	1.57	1.425	1.394	1.237	1.108	0.980	0.859	0.772	0.666	0.556	0.46	0.37	0.32	0.21	0.15	0.06	0.04	0.02	37.11	1,067.58

5.2. Effect of the Bernoulli probability of empty spaces

To observe the effect of the Bernoulli probability of empty spaces, we varied z from 0.00 (all parking spaces are occupied) to 0.90 (10 percent of parking spaces are occupied) and the memory size from 0 to 25. Given our heuristic rules, we expected to see a linear decrease in average revenue per shift as the Bernoulli probability of empty spaces increases.

Fig. 6 shows the result of our experiment. Consider changes in the average revenue when the memory size is 0. When $z = 0.0$, the average revenue per shift was $782.9. The average revenue per shift decreased by 2.85 percent ($22.3) when z increased to 0.1. The rate of decrease in average revenue grew as z increased. The same effect can be observed for other memory sizes.

With constant p, in one tour cycle, the PEO encountered a linearly decreasing number of cars with an expired permit as z linearly increased. However, the more the empty spaces, the faster the PEO completes a tour cycle because the PEO does not need to spend b unit time inspecting empty spaces. Thus, in an 8-hour shift, the PEO can complete more tour cycles when $z = 0.5$ than when $z = 0.0$. Recall that for each tour cycle, the probability of parking violation is constant. Hence, the 10 percent decrease in the number of empty spaces when most parking spaces were occupied generated less effect than the 10 percent decrease when most parking spaces were empty.

5.3. Effect of the parking permit time length L

To observe the effect of the parking permit time length, we varied L from 30 minutes to 120 minutes, and the memory size from 0 to 25. We kept the mode and the upper bound of the triangle distribution at 5 minutes before L and 30 minutes after L, respectively. Given our heuristic, we expected to see a decrease in average revenue per shift as L increases.

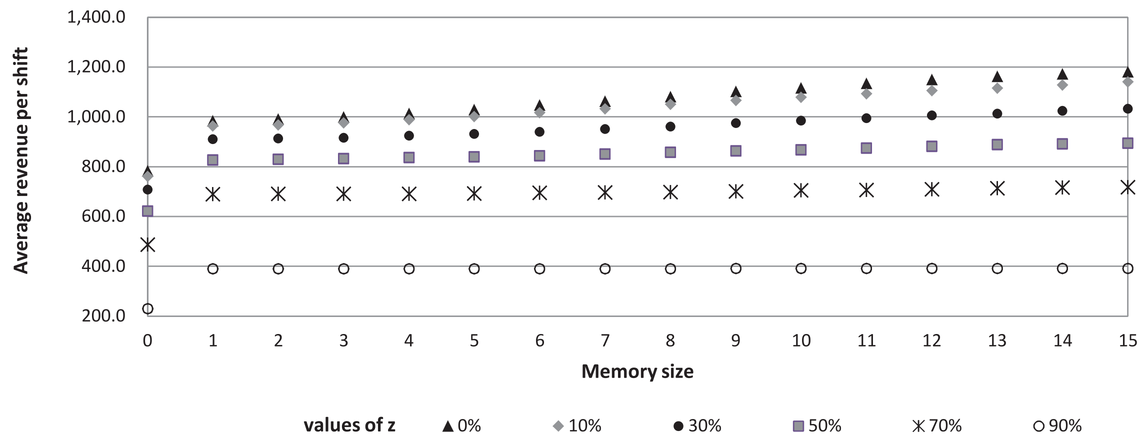
Fig. 7 shows the result of our experiment. Consider changes in the average revenue when the memory size is 0. When $L = 30$ minutes, $p = 15.13$ percent, the average revenue per shift was $1313.7. The average revenue per shift decreased by 46 percent ($605.7) when L increased to 60 minutes (thus p decreased to 5.91 percent).

Essentially, the PEO encounters a decreasing number of cars with an expired permit in each tour cycle, as p decreases. However, the less the parking violations, the shorter the PEO takes to complete a tour cycle because the PEO needs to spend a unit time issuing each ticket. So, in an 8-hour shift, the PEO can complete fewer tour cycles when $p = 15.13$ percent than when $p = 5.91$ percent. Notice also that the improvement in the revenue from memory size of 0 to memory size of 1 is greatest when p is small.

5.4. Effect of the distribution of the owner's return time $v(x)$

In the previous section, we experimented with a triangle distribution. In this section, we experiment with Kumaraswamy distribution, which is similar to the Beta distribution. The original probability density function of this distribution is $abx^{a-1}(1 - x^a)^{b-1}$ for $x \in [0, 1]$ before we generalized it to our time range. We kept the parking time L at 60 minutes and with $z = 0.3$. We select Kumaraswamy parameters $a = 4$ and $b = 5.6275$ so that the mode m is still at 55, which is 5 minutes before the car turns into violation state and the end point X is still 90, which is 30 minutes after the car turns into violation state. Fig. 8 shows the selected distribution.

The computed p is 0.03946. We varied the memory size from 0 to 25. We ran 1000 rounds of experiment for each one of the 25 memory options. Fig. 9 shows the result of our experiment. When the memory size is 0, the simulation shows the average revenue per shift to be $511.8 which is very close to the computed expected revenue per shift of $513.97. The simulated PEO went through an average of 620.31 parking spaces in an 8-hour shift, out of these, 186.09 parking



Memory size	Average Revenue per shift for different values of z: 0% Average Revenue per shift	0% Std. Dev	10% Average Revenue per shift	10% Std. Dev	30% Average Revenue per shift	30% Std. Dev	50% Average Revenue per shift	50% Std. Dev	70% Average Revenue per shift	70% Std. Dev	90% Average Revenue per shift	90% Std. Dev
0	782.9	112.5	760.6	110.8	708.0	110.2	621.3	108.9	486.3	104.2	229.8	74.5
1	984.4	93.2	963.2	91.9	910.3	96.1	826.7	93.0	688.9	88.5	390.0	73.6
2	991.5	95.1	968.1	91.9	912.7	92.8	829.9	92.8	690.2	87.1	389.9	75.5
3	1,000.1	94.7	975.5	93.3	915.8	90.9	833.0	91.5	690.1	86.4	389.6	75.3
4	1,014.5	92.7	988.8	93.9	923.3	92.1	837.1	93.5	690.5	87.4	389.7	74.6
5	1,029.1	94.2	1,001.7	92.6	931.2	92.7	840.1	92.0	691.9	85.8	389.3	73.5
6	1,047.0	94.4	1,016.7	93.9	939.6	94.8	844.3	90.7	694.4	85.4	390.0	72.9
7	1,063.1	92.0	1,032.6	93.8	951.1	96.7	851.0	89.4	696.6	87.8	390.4	73.7
8	1,081.7	91.6	1,050.3	94.6	961.0	98.7	857.4	90.2	698.0	87.3	389.9	73.0
9	1,101.8	97.2	1,065.5	96.6	974.1	98.1	863.6	89.7	700.6	89.0	390.9	73.1
10	1,116.7	98.5	1,078.0	93.4	984.2	96.9	867.8	90.8	704.3	89.1	390.8	72.2
11	1,134.5	100.1	1,092.2	97.3	993.7	98.9	875.2	93.0	706.7	88.4	391.6	73.3
12	1,150.7	102.9	1,105.7	96.1	1,005.1	98.5	881.6	94.4	708.9	88.5	391.9	70.9
13	1,163.2	100.6	1,115.7	97.3	1,012.8	100.8	888.2	93.5	713.5	88.0	391.3	73.6
14	1,173.3	101.8	1,127.6	97.2	1,023.3	98.8	892.1	93.7	715.6	89.6	391.5	72.3
15	1,181.3	101.2	1,139.7	99.6	1,032.4	100.3	895.0	96.2	717.7	90.4	391.7	73.0
16	1,190.2	100.7	1,145.0	99.2	1,038.8	100.1	899.6	96.5	720.3	88.2	391.8	73.0
17	1,193.4	102.7	1,152.5	101.5	1,045.1	100.8	903.5	97.3	721.3	90.9	391.9	73.9
18	1,200.4	98.7	1,160.5	98.2	1,052.7	100.7	906.9	95.7	722.5	90.2	392.3	73.0
19	1,203.8	101.4	1,165.1	101.3	1,055.9	102.2	914.0	97.0	725.6	89.1	391.7	72.7
20	1,204.4	99.5	1,165.7	100.7	1,059.5	99.7	918.4	98.0	727.0	90.5	392.3	71.9
21	1,207.7	101.5	1,171.2	102.9	1,062.1	98.9	921.6	99.8	726.5	91.0	393.4	73.0
22	1,207.5	99.2	1,173.5	100.8	1,067.5	97.5	923.3	96.8	728.6	90.6	393.8	73.9
23	1,209.1	101.4	1,174.1	99.7	1,068.9	97.0	923.8	98.2	729.1	88.9	393.9	72.2
24	1,210.4	100.3	1,175.9	102.0	1,068.5	96.9	926.6	99.2	729.2	89.3	393.8	73.2
25	1,212.2	101.1	1,173.2	102.3	1,067.6	100.3	924.1	99.6	730.4	87.7	393.6	73.6

Fig. 6. Effect of the Bernoulli probability of empty spaces.

spaces were empty (30 percent). Among the occupied spaces, 17.13 cars were in violation state (approx. 3.9 percent). The simulated PEO took approximately 3.54 tour cycle per shift.

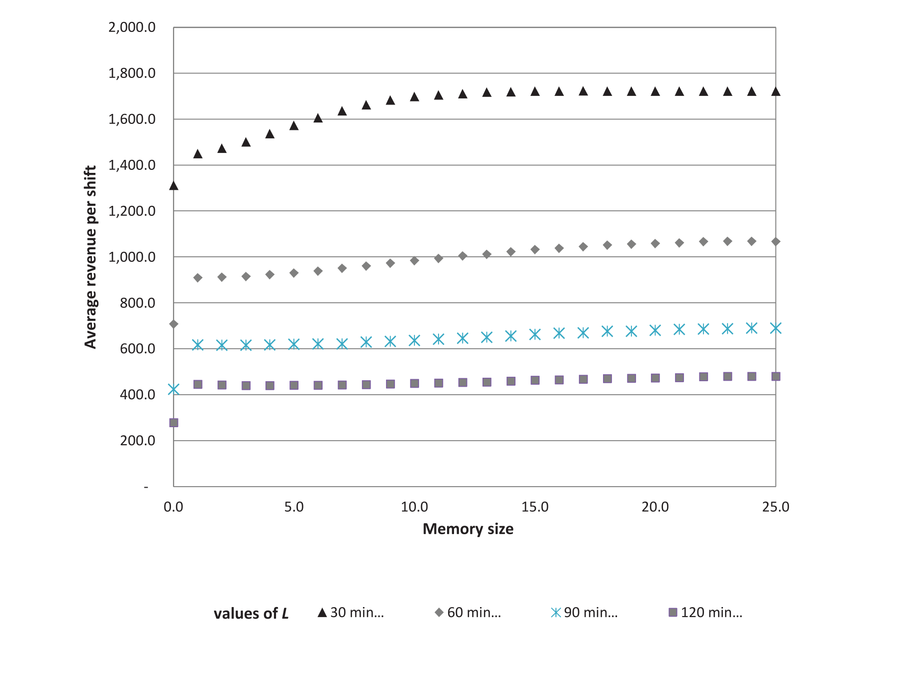
Since the percentage of cars in violation state when using Kumaraswamy distribution is less than the similar setting when using triangle distribution as shown in Fig. 4, the average revenues are less for all memory sizes.

Increasing the memory size to 1, we noted a 46.37 percent improvement ($237.3) in the average revenue per shift. The subsequent increases did not show significant improvement in the average revenue per shift. This shows that our heuristic is robust for the change in the distribution.

6. Summary

6.1. Technical summary

In this paper we show that revenue collection from car parking violations can be modeled as a Chinese Postman Problem (CPP) on a street graph and prove that the car inspection order is optimal when the PEO automatically follows a preset route. We assume throughout the analysis in the paper that the number of cars whose permit expiration times a PEO can memorize does not impact the traversing order of the graph's edges. We discard the case of the PEO re-optimizing his CPP inspection route when retracing his route to a previously



	Average Revenue per shift for different permit duration time			
Memory size	30 min p = 15.13%	60 min p = 5.91%	90 min p = 3.14%	120 min p = 1.94%
0.0	1,313.7	708.0	423.8	278.6
1.0	1,450.7	910.3	617.1	446.0
2.0	1,474.5	912.7	615.7	443.1
3.0	1,502.9	915.8	616.7	441.2
4.0	1,538.0	923.3	617.1	440.7
5.0	1,573.5	931.2	619.6	441.8
6.0	1,607.8	939.6	621.0	441.5
7.0	1,637.7	951.1	622.0	443.7
8.0	1,663.3	961.0	629.3	445.0
9.0	1,684.5	974.1	632.4	447.3
10.0	1,700.2	984.2	636.6	449.9
11.0	1,707.2	993.7	642.0	451.6
12.0	1,712.5	1,005.1	646.2	454.2
13.0	1,719.2	1,012.8	650.9	456.1
14.0	1,721.0	1,023.3	656.6	459.9
15.0	1,723.6	1,032.4	662.8	463.8
16.0	1,722.9	1,038.8	668.0	466.0
17.0	1,724.3	1,045.1	669.4	468.3
18.0	1,723.6	1,052.7	677.3	470.6
19.0	1,723.7	1,055.9	677.3	472.5
20.0	1,723.6	1,059.5	680.4	474.0
21.0	1,723.5	1,062.1	685.5	475.8
22.0	1,723.6	1,067.5	686.1	479.6
23.0	1,723.6	1,068.9	687.6	480.9
24.0	1,723.7	1,068.5	690.4	480.8
25.0	1,723.7	1,067.6	690.4	481.3

Fig. 7. Effect of the parking permit time length *L*.

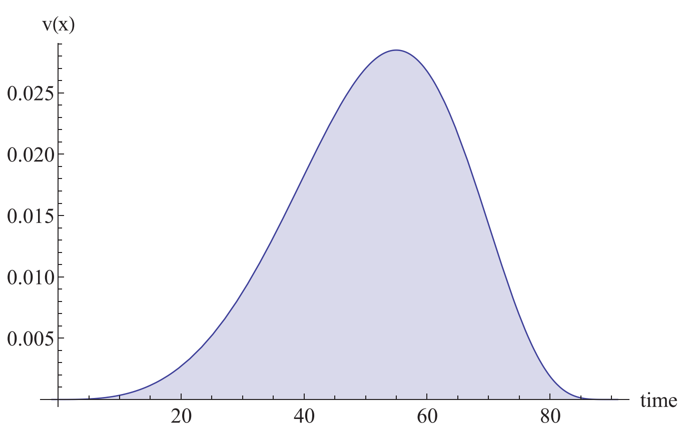
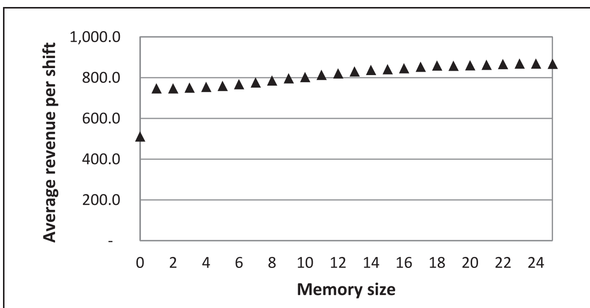


Fig. 8. Kumaraswamy distribution.

inspected car since the likelihood of such an event is small. The PEO commits to traversing the assigned street subgraph by following the optimal CPP path.

We focus on the PEO's real-time online decisions at each inspected car. The PEO has the options to wait at the current car, to step back to previously inspected cars on that edge, or to walk to the next car. We propose a myopic set of rules that help the PEO to make a locally best decision aimed at maximizing the expected revenue per unit time. We tested our heuristic rules by running a simulation experiment using various parameters. The main finding is that by allowing for the PEO to wait by a car (for some cars) with a valid parking permit until the permit's time expires raises the average collected revenue significantly (10–69 percent). Increasing the number of inspected cars whose permit expiration times the PEO can remember beyond one does not significantly increase the average revenue and therefore is unlikely to happen. We also validated the 'obvious' that the average revenue depends on the probability of empty spaces and the probability of parking violations.

We tested the effect of changing dead-heading time λ. Decreasing λ will increase the average revenue per shift. We did not test the effect of inspecting time b, and ticket issuing time a. Increasing a and b will definitely decrease the average revenue per shift regardless of the probability distributions of empty spaces and parking violations. We did, however, test the effect of the number of parking spaces per



Memory size	Average Revenue per shift	Std. Dev	Memory size	Average Revenue per shift	Std. Dev	Memory size	Average Revenue per shift	Std. Dev
0	511.8	105.0	10	804.5	93.3	20	862.1	93.9
1	749.1	88.9	11	815.0	93.8	21	864.3	92.7
2	748.8	88.2	12	821.8	92.3	22	868.3	94.5
3	752.0	90.0	13	831.2	94.5	23	869.7	95.7
4	755.8	88.7	14	839.5	93.0	24	869.8	94.6
5	760.8	89.1	15	842.2	92.9	25	869.4	92.9
6	769.6	90.9	16	847.6	91.8			
7	778.0	91.3	17	854.3	92.5			
8	787.7	89.9	18	860.4	95.0			
9	798.0	91.7	19	859.2	92.4			

Fig. 9. Experiment result for Kumaraswamy distribution.

edge and found that it does not effect the average revenue per shift at all because the PEO would encounter the same number of parking violations in 8 hours.

Our simulation experiments were based on several assumptions regarding the respective probability distributions. We assumed that the probability distributions of empty spaces is Bernoulli. We assume that the probability distribution of the PEO arrival at each car is uniform. Moreover, we assume that the probability distribution of the car owner's return time counting from the moment the car owner parks her car is a triangle distribution. (Note, however, that when we replaced the triangle distribution with Kumaraswamy distribution, we reached similar results.) These plausible assumptions might need to be validated with real data before our online heuristic rules are to be implemented. However, once the data is collected and the maximum time worth waiting for a valid permit to expire is calculated, the actual implementation into the daily routine operation of the PEOs is straightforward.

One might ask what would happen if the probability distribution of empty spaces differs among different edges and varies over a period of time. For instance, parking spaces near a popular restaurant are usually full around lunch time and dinner time. Should the PEO follow the optimal CPP path, or should the PEO traverse the edge with denser parking cars more often? This question is outside the scope of this paper as we do not test for different probability distributions of cars' turnover rates on edges.

Another issue pertains to the construction of an optimal CPP path. There could be multiple optimal CPP paths which result in the same distance travel per tour cycle. In our experiment with the graph in Fig. 3, we chose the CPP path that traverses edge e_3 in a dead-heading mode right after inspecting all cars on that edge. If a different path was selected, e.g. $e_1 \to e_4 \to e_3 \to e_5 \to e_6 \to e_7 \to e_3$ (reverse) $\to e_2$, edge e_3 would be repeated long after the PEO inspected the cars on that edge. In this case, some cars' permits might have expired and some new cars might have been parked in previously empty parking spaces on edge e_3. In that case it might be better for the PEO to reinspect edge e_3, instead of traversing it in a dead-heading mode. The problem of selecting an optimal CPP solution that allows for traversing an edge in a dead-heading mode immediately after its regular traversal is outside the scope of this paper. We are not aware of any work on this topic or on the question of establishing an existence of such optimal CPP solution for any nontrivial family of graphs.

One might also ask what would happen if we do not assume that the PEO commits to traversing the subgraph following the selected optimal CPP path. For instance, consider the following scenario. The PEO was able to memorize the parking time left of all parked cars in the subgraph while following CPP tour: $e_1 \to e_4 \to e_3 \to e_3$ (reverse) $\to e_7 \to e_6 \to e_5 \to e_2$. After he inspected the last (top most) car on edge e_5 in Fig. 3, he recalled that the left most car on edge e_3 would have 1 minutes left. He might want to change the course of his traversal to inspect the left most car on edge e_3 before continuing on to edge e_2. Such deviation from the original CPP path would require very little time and might result in larger expected revenue per unit time. Again, we do not examine such deviations in the current work.

6.2. Discussion

Our study has broader implication beyond revenue collection from illegally parked cars. Clearly, any inspection activity over edges of graphs that involves revenues, say ice-cream vendors, etc. would have a similar flavor and could use similar CPP based logic and analysis as the one used in this paper. The results of this study reveals the existence of a 'bang for the buck' phenomena; pay attention to the remaining time for the currently inspected car – it matters. In the broader context it translates to paying attention to the dynamic nature of events along the arcs of the graph.

The revenues collected by parking enforcement operations are significant as demonstrated by the available data. Thus, increasing such revenues even more, say by 10 percent, by implementing a simple rule prescribing when a PEO ought to wait in front of a car with a valid permit (without going back to previous cars) and when to continue on his CPP route (or any preset inspection route) has important revenue implications. We argue that the implementation of such a rule is feasible in practice. Calculating (or even estimating by trial and error) a threshold time value and setting a policy that PEOs wait by the car upon inspection of a valid parking ticket if the time left on the ticket is below the threshold value and continue otherwise, represents a reasonable task. The threshold time value can be calculated by a city manager/analyst prior to communicating to PEOs.

Another feasibility question relates to the technology of issuing/purchasing a parking permit. Since there are many different parking permit schemes implemented in the various cities across the globe and the technological frontier of issuing short term parking permits is evolving, is our car parking permit inspection procedure with the rule that tells a PEO when to wait in front of a car with a valid permit restricted by a specific technology? The clear answer is that the procedure is independent of the technology. Regardless if the technology used is the 'old' coin operated parking meters, central kiosk parking ticket purchasing machines, electronically (remotely adjusted) purchase of parking permits, all these technologies require human inspection and the ticketing of parking violations unless a expensive automated violations monitoring system is installed. We note (Seattle Parking Management Study, 2002) that having parking inspection PEOs patrolling city streets generates not only considerable revenues but also as its by-product it generates considerable societal benefits in terms of maintaining order and crime prevention.

Acknowledgments

The authors express their thanks to Professor Lerzan Ormeci for her contribution in the early stages of the project.

Appendix A. Description of the simulation study

The scheme for our numerical analysis is outlined in the pseudo code below. We use the graph in Fig. 3, with the inspection CPP tour always starting at the top left corner of the graph and executing the following CPP tour: $e_1 \to e_4 \to e_3 \to e_3$ (reverse) $\to e_7 \to e_6 \to e_5 \to e_2$. In the first step, the subroutine selects the first parking space on edge e_1 for inspection. The walking time is increased by λ. We 'call' a random number generator (randomizer) to determine if this parking space is empty, occupied by a car with a valid permit, or occupied by a car with an expired permit.

For instance, the randomizer generates the first number between 0 and 1 from a uniform distribution; if the number falls between 0 and z, the parking space is empty, if the number falls between z and 1, the parking space is occupied. The second random number between 0 and X is generated from a triangle distribution for the car owner's return time. The third random number also between 0 and X is generated from a uniform distribution for the PEO arrival time. If the third random number is greater than the second number, we regenerate both numbers. If the third number is smaller than the second number and greater than L, the parking space is occupied by a car in violation state. If the third number is smaller than the second number and smaller than L, the parking space is occupied by a car with a valid permit.

Main routine

1 Set the current position of the officer to the top left corner of the graph.
2 Determine CPP path including direction and the dead-heading arcs.
3 While clock time $t < S$:
4 Determine the target position i (Subroutine).
5 If this is the first car on an edge to get inspected, reset the memory of all cars' time left $s_{ie}(t)$.
6 Walk to the target position i.
7 Increase t by walking time and deadhead traversing time if any.
8 If the time left of i is not previously known,
9 generate three random numbers from [0,1] uniform distribution (for z), $[0,X]$ triangle distribution for $\xi_{ie}(t)$, and $[0,X]$ uniform distribution for y to determine if i is empty, occupied by a car with a valid permit, or occupied by a car with an expired permit.
10 If i is occupied by a car with an expired permit, mark i as ticketed, increase t by $a + b$ and increase revenue by r.
11 If i is occupied by a car with a valid permit, increase t by b. The amount of permit time left is $L - y$.
12 Else
13 If owner comes back before the permit expires ($\xi_{ie}(t) \le L$), mark i as empty. Increase t by walking time and waiting time, if any.
14 Else, mark i as ticketed. Increase t by walking time and waiting time, and ticket issuing time a, and increase revenue by r.
15 End if.
16 End if.
17 End while.

Subroutine: Determine the target position

S1 Set the next parking space on the CPP path that has not been inspected as the default target position i
S2 For each car j in the memory (including the current car if not ticketed)
S3 Determine if car j's owner has returned to the car using $\xi_{ie}(t)$. If so, skip to the next car in the memory.
S4 Calculate the expected profit per expected time spent of car j.
S5 If the calculated amount is greater than the default target position, set car j as the new target position.
S6 End for.
S7 Return the target position.

Discussion: If the parking space is empty, the PEO moves on to determine the next target car (line 4). If the parking space is occupied by a car with an expired permit, the PEO immediately issues a ticket and increases the revenue by r. The clock time t is increased by $a + b$ which is the inspecting time b plus the ticket issuing time a. If the parking space is occupied by a car with a valid permit, we determine the amount of parking permit time left by subtracting the PEO arrival time y (generated by the randomizer) from L. The clock time t is increased by b. Then, the PEO moves on by determining the next target car (line 4 calling the subroutine).

Assume that the first parking space is occupied by car 1 with a valid permit. The subroutine sets the second parking space on the CPP path as the default target position. We have one car in memory, which is car 1. Hence, we determine whether the expected profit per expected time of waiting at car 1 is greater than the expected revenue per unit time of traversing CPP path without waiting (expected revenue per unit time of the second parking space).

Assume that it is more profitable to wait at car 1. We would not have to increase the walking time as the PEO is already at this parking space. Using the random number previously generated for $\xi_{ie}(t)$, and y, we determine whether the owner will return to the car before the permit expires. If the owner returns to the car before the permit expires, we increase the waiting time upto the time when the owner returns. If the owner returns to the car after the permit expires, we mark car 1 as ticketed, increase the revenue by r, and increase the waiting time by $L - y + a$.

Throughout the numerical experiment, the memory size is interpreted as the number of cars that the PEO can remember their parking permit time left. For instance, if the furthest parking space that the PEO has inspected is car 8 and he has four car memory option, he would be able to remember the time left of cars 5, 6, 7, and 8.

At the end of one tour (the PEO reaches the starting point), we reset every car's time. We assume that they are all new cars. This assumption is reasonable because one tour takes longer than 60 minutes – the available permit time.

References

Aráoz, J., Fernández, E., & Meza, O. (2009). Solving the prize-collecting rural postman problem. *European Journal of Operational Research*, *196*, 886–896.

Archetti, C., Feillet, D., Hertz, A., & Speranza, M. G. (2010). The undirected capacitated arc routing problem with profits. *Computers & Operations Research*, *37*, 1860–1869.

Beltrami, E. J., & Bodin, L. D. (1974). Networks and vehicle routing for municipal waste collection. *Networks*, *4*, 65–94.

Corberán, A., Plana, I., Rodríguez-Chía, A. M., & Sanchis, J. M. (2013). A branch-and-cut algorithm for the maximum benefit Chinese postman problem. *Mathematical Programming*, *141*, 21–48.

Corberán, A. & Prins, C. (2010). Recent results on arc routing problems: An annotated bibliography. *Networks*, *56*(1), 50–69.

Dror, M. (Ed.). (2000). *ARC ROUTING: Theory, solutions, & applications*. Norwell, MA: Kluwer Academic Publishers.

Dror, M. & Stulman, A. (1987). Optimizing robot's service movement: A one dimensional case. *Computers & Industrial Engineering*, *12*, 39–46.

Feillet, D., Dejax, P., & Gendreau, M. (2005). The profitable arc tour problem: Solution with branch-and-price algorithm. *Transportation Science*, *39*, 539–552.

Katehakis, M. & Veinott, Jr., A. F., (1987). The multi-armed bandit problem: Decomposition and computation. *Mathematics of Operations Research*, *12*(2), 262–268.

L'Ecuyer, P., Mayrand, M., & Dror, M. (1991). Dynamic scheduling of a robot servicing machines on a one-dimensional line. *IIE Transactions*, *23*, 371–382.

Malandraki, C. & Daskin, M. S. (1993). The maximum benefit Chinese postman problem and the maximum benefit traveling salesman problem. *European Journal of Operational Research*, *65*, 218–234.

Papadimitriou, C. H. (1976). On the complexity of edge traversing. *Journal of the Association for Computing Machinery*, *23*, 544–554.

Pearn, W. L. & Chiu, W. C. (2005). Approximate solutions for the maximum benefit Chinese postman problem. *International Journal of Systems Science*, *36*, 815–822.

Pearn, W. L. & Wang, K. H. (2003). On the maximum benefit Chinese postman problem. *OMEGA*, *31*, 269–273.