Sensor location problems in path-differentiated congestion pricing

Mahmood Zangui a, Yafeng Yin a,*, Siriphong Lawphongpanich b

a Department of Civil and Coastal Engineering, University of Florida, Gainesville, FL, USA
b Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL, USA

A R T I C L E   I N F O
Article history:
Received 13 September 2014
Received in revised form 4 January 2015
Accepted 7 January 2015
Available online 10 February 2015

Keywords:
Congestion pricing
Sensor location
Path differentiation
Path observation

A B S T R A C T
Path-differentiated congestion pricing is a tolling scheme that imposes tolls on paths instead of individual links. One way to implement this scheme is to deploy automated vehicle identification sensors, such as toll tag readers or license plate scanners, on roads in a network. These sensors collect vehicles’ location information to identify their paths and charge them accordingly. In this paper, we investigate how to optimally locate these sensors for the purpose of implementing path-differentiated pricing. We consider three relevant problems. The first is to locate a minimum number of sensors to implement a given path-differentiated scheme. The second is to design an optimal path-differentiated pricing scheme for a given set of sensors. The last problem is to find a path differentiated scheme to induce a given target link-flow distribution while requiring a minimum number of sensors.

1. Introduction

Congestion pricing is a well-established instrument for congestion mitigation, in which tolls are imposed to affect travel choices of users and entice them to use road networks more efficiently. Previous attempts have been made to improve the performance and address implementation issues, e.g., public opposition and inequity, of congestion pricing by designing more efficient, equitable, or robust schemes. Among these is the idea of differentiating tolls (or prices) according to, e.g., user characteristics, vehicle types, and trip purposes. See, e.g., de Palma and Lindsey (2011) and Zangui et al. (2013) for recent reviews. This paper focuses on tolls that are differentiated by paths or path-differentiated tolls.

Traditionally, tolls are imposed on links and summing these link tolls along individual paths lead to one set of path tolls. Such path tolls are dependent in that a change in one link toll may affect tolls on several paths. Path-differentiated tolls relax this dependency by assigning tolls directly to individual paths, instead of links. As a result, path-differentiated tolls can be determined independently. In addition, paths are generally more numerous than links. Thus, path tolls are more flexible and can achieve better performance than link tolls. Zangui et al. (2014) illustrated the potentials of path-differentiated tolls by comparing their performance to their link-based counterpart in designing pricing schemes that impose the least financial burden on users.

To implement a path-differentiated pricing scheme, the toll collection system must be able to distinguish paths of individual vehicles. For example, a system relying on data from on-board tracking devices, such as GPS, possesses such a capability. However, such a system requires all vehicles to be equipped with tracking devices and doing so may be expensive and difficult to maintain. Alternatively, automated vehicle identification (AVI) sensors can be used to identify vehicles’ paths.
When AVI sensors detect nearby or passing vehicles, they can uniquely identify the vehicles and record the time of detection. When collected from a sufficient number of strategically located sensors, the time of detections and locations of the detecting sensors can be used to infer the path of individual vehicles. In this paper, we investigate the use of AVI sensors in identifying paths of vehicles for the purpose of implementing path-differentiated charging schemes.

In practice, a broad range of devices, including some of toll collection devices, can be categorized as AVI sensors. In fact, two of the basic tasks in any toll charging system are to identify vehicles and record their locations (or, more precisely, the locations of the detecting sensors), which are the functionalities of AVI sensors. Nowadays, most of the tolling facilities use electronic toll collection technologies such as toll-tag readers (e.g., SunPass in Florida, TxTag in Texas, and E-ZPass in 15 other states in the U.S.) or license plate scanners (e.g., Toll-By-Plate in Florida, and License Plate Toll in Colorado). These devices can detect and identify vehicles, and can thus serve as AVI sensors. See, e.g., de Palma and Lindsey (2011) for a review on sensor technologies used in toll collection.

In the literature, traffic sensors, including AVIs, have many applications in transportation systems and each application gives rise to a different sensor location problem (see, e.g., Gentili and Mirchandani (2012) for a comprehensive review). Among them, the most relevant one to this paper is to optimally locate sensors for the purpose of traffic flow inference, where the data from sensors is used to determine some of the unobserved flows. These sensor location problems can be categorized, with respect to the type of flow they attempt to obtain, as follows:

- **Origin–destination (OD) demand.** While most of the studies attempted to estimate OD demands by minimizing deviations from a historical trip matrix (Yang et al. (1991), Yang and Zhou (1998), Bianco et al. (2001), Ehlert et al. (2006), and Castillo et al. (2008c)), some, e.g., Liou and Hu (2009), obtained the exact value of OD demand using the least number of sensors on the network.
- **Flow on all paths.** Gentili and Mirchandani (2005) assumed that vehicles equipped with devices can transmit their path information to a nearby roadside sensor, and studied the location problem for this type of sensors. Others, e.g., Castillo et al. (2008c) and Minguez et al. (2010), attempted to find optimal locations of sensors for observing path flows and OD demands.
- **Flow on all links.** This problem has been introduced by Hu et al. (2009), who also proposed a solution algorithm that uses the link-path incidence matrix and thus requires path enumeration. Later, Ng (2012) proposed a solution method for the problem that avoids path enumeration. He (2013) introduced more variants of sensor location problem for obtaining link flows, and proposed solution algorithms for those problems.

In addition to the above, a more general form of flow observation problem has emerged, which attempts to infer different flows from a set of observed flow data, e.g., using link and path flow information collected by sensors to obtain flow on other paths. Castillo et al. (2008b), Castillo et al. (2010), and Castillo et al. (2011) discussed and proposed solution method for these types of problems.

The contributions of this paper are threefold. First, we introduce and explore the idea of using AVI sensors to implement path-differentiated tolls. Second, we develop mathematical models for three variations of the sensor location problem and investigate their properties, where two problems involve determining sensor locations and one assumes that sensor locations are given. Third, through examples, we demonstrate the advantages of using AVI sensors to implement path-differentiated tolls over traditional link-based schemes.

For the remainder, the next section introduces the notation and briefly reviews the concept of path-differentiated pricing. Section 3 defines and formulates conditions for a set of sensor locations, capable of identifying the path of every vehicle. Each of the next three sections discusses one problem in sensor locations for path-differentiated tolls. The first version aims at finding optimal sensor locations for implementing a given path-differentiated scheme, whereas the second one assumes the sensors are already located on the network and attempts to design an optimal path-differentiated scheme that is implementable using the available sensors. The third problem is to design a path-differentiated scheme that induces a given link-flow distribution and requires the least number of sensors to implement. Finally, Section 7 concludes the paper.

### 2. Background

This section introduces the notation used throughout the paper and presents an overview of path-differentiated congestion pricing. To be concise, we assume that readers are familiar with the motivation and fundamental results in congestion pricing. Otherwise, readers can consult, e.g., Yang and Huang (2005), Lindsey and Verhoef (2000), and Tsekeris and VoB (2009).

#### 2.1. Notation

Let $G = (N, A)$ represent a transportation network, where $N$ and $A$ are the sets of nodes and links, respectively. The set of OD pairs is denoted by $W$, and $d_w$ represents the demand for OD pair $w \in W$. The set of paths connecting OD pair $w \in W$ is denoted by $P_w$, and the union of all these path sets by $P$. For each path $p$, its flow, travel time, and toll are represented by $f_p$, $t_p(f)$, and $\pi_p$, respectively. For simplicity, tolls are represented in the unit of time. (In this paper, vectors of variables are...
denoted by the same letters but without subscript, e.g., \( f \) is the vector of path flows \( f_p \), and double struck capital letters denote sets.) Also define \( \mathbb{F} \) as the set of all demand-feasible path flows, i.e.,
\[
\mathbb{F} = \left\{ f \in \mathbb{R}^{|\mathbb{P}|} : \sum_{p \in \mathbb{P}_w} f_p = d_w, \ \forall w \in \mathbb{W} \right\}
\]
where \(|\mathbb{P}|\) is the cardinality of set \( \mathbb{P} \) and \( \mathbb{R}^{|\mathbb{P}|} \) is the set of all non-negative vectors in \( \mathbb{R}^{|\mathbb{P}|} \).

2.2. Path-differentiated pricing

In the context of traffic assignment, user equilibrium (UE) is defined as a traffic condition where, for every OD pair, the costs of utilized paths are equal and no greater than the cost of any un-utilized path. The UE condition can be written as the following complementarity system:
\[
0 \leq f_p \perp (t_p(f) - u_w) \geq 0 \quad \forall p \in \mathbb{P}_w, \ w \in \mathbb{W}, \ f \in \mathbb{F}
\]
where for OD pair \( w \), \( u_w \) is the travel cost (or time) of a utilized path.

A traffic distribution, \( f^* \in \mathbb{F} \), that yields the least total travel time is said to be system optimal (SO). Mathematically, \( f^* \) solves the following optimization problem:
\[
f^* = \arg\min_{f \in \mathbb{F}} \{ f^t(t(f)) \}
\]

As formulated above, it is well-known that \( f^* \) is not unique.

In congestion pricing, tolls are used to decentralize a desired traffic distribution. It is well known in the literature that, if the desired distribution is a SO distribution or \( f^* \), then a vector of link tolls, \( \beta \), for such a purpose, or a valid \( \beta \), makes \( f^* \) be in a tolled user equilibrium, i.e., a valid \( \beta \) must satisfy the following complementarity condition:
\[
0 \leq f_p \perp \left( t_p(f^*) + \sum_{a \in \mathbb{A}} \beta_a \delta_a^p - u_w \right) \geq 0 \quad \forall p \in \mathbb{P}_w, \ w \in \mathbb{W} \quad (1)
\]
where \( \beta_a \) is the amount of toll on link \( a \) and the parameter \( \delta_a^p \) is 1 if path \( p \) contains link \( a \) and 0 otherwise. Thus, the summation \( (1) \) represents the total toll a motorist on path \( p \) must pay.

As stated, condition \( (1) \) requires every user traveling on link \( a \in \mathbb{A} \) to pay the same "anonymous" toll, namely \( \beta_a \). Summing these tolls on individual paths yields a set of path tolls. Such path tolls are typically dependent in the sense that changing one link toll may affect tolls on several paths. To avoid this dependency, Zangui et al. (2013) considered tolls based on trip characteristics, e.g., origin, destination, or entire path, and charging users accordingly. Among these, path-differentiated tolls (see Zangui et al. (2014)) is most flexible. Such tolls are directly determined for individual paths, not links. Similar to \( (1) \), a valid vector of path tolls, \( \pi \), satisfies the following:
\[
0 \leq f_p \perp \left( t_p(f^*) + \pi_p - u_w \right) \geq 0 \quad \forall p \in \mathbb{P}_w, \ w \in \mathbb{W}
\]
where \( \pi_p \) is the toll charged on path \( p \). Alternatively, \( \pi \) is valid if it is part of the pair \( (f^*, \pi) \) that solve the following optimization problem:

\[
\text{(PATH)} \quad \min \sum_{p \in \mathbb{P}} f_p t_p(f) \quad (2a)
\quad \text{s.t.} \quad f \in \mathbb{F} \quad (2b)
\quad f_p (t_p(f) + \pi_p - u_w) = 0 \quad \forall w \in \mathbb{W}, \ p \in \mathbb{P}_w \quad (2c)
\quad t_p(f) + \pi_p - u_w \geq 0 \quad \forall w \in \mathbb{W}, \ p \in \mathbb{P}_w \quad (2d)
\quad \pi_p, \ u_w \geq 0 \quad \forall w \in \mathbb{W}, \ p \in \mathbb{P}_w \quad (2e)
\]
where \( u_w \) is the minimum generalized cost, i.e., travel time plus toll, of traveling between OD pair \( w \). Constraint \( (2b) \) ensures that the path flows are feasible. The tolled user equilibrium conditions are represented by constraints \( (2c) \) and \( (2d) \). Finally, the last set of constraints guarantees that tolls and OD generalized costs are non-negative.

3. Path observation

This section discusses how AVI sensors can be used to identify paths of vehicles. In particular, the problem of determining the minimum number of sensor locations (i.e., links to install or equip with sensors) for the purpose of distinguishing a collection of paths is known as the path-observability problem in the literature. The formulation of this problem provides a foundation for our formulations of the three problems discussed in subsequent sections. Afterwards, we study the properties of the path-observability problem and illustrate them on a simple network.
3.1. Model formulation

Assume that some links are equipped with sensors. We say that path \( p \) is observable with a given set of sensors, if data from these sensors can identify vehicles that traverse path \( p \). This requires vehicles on path \( p \) to be detected by at least one sensor and these vehicles must also be distinguishable from vehicles traveling on any other path. Formally, path \( p \) is observable when the following conditions are satisfied:

(a) At least one link on path \( p \) must be equipped with a sensor.
(b) For every other path \( k \neq p \), at least one link on \( p \), that is not on \( k \), must be equipped with a sensor.

Condition (a) guarantees that every vehicle on path \( p \) is detected by at least one sensor. Condition (b) ensures that data from the sensors are sufficient to distinguish vehicles on path \( p \) from those traveling on every other path. When condition (b) does not hold, we say paths \( p \) and \( k \) are indistinguishable.

In the path-observability problem, we want to find a minimum number of sensor locations to observe or distinguish a collection of paths, denoted below as \( P \). The following is a formulation of the path-observability problem from Castillo et al. (2008a):

\[
\text{(OBSV)} \quad \min \sum_{a \in A} z_a \quad \text{subject to} \quad \sum_{a \in A} z_{a | p_k^p} \geq 1 \quad \forall p \in P \tag{3a} \\
\sum_{a \in A} z_{a | p_k^p} \geq 1 \quad \forall p, k \in P, \; p \neq k \tag{3b} \\
z_a \in \{0, 1\} \quad \forall a \in A \tag{3c}
\]

where \( z_{a | p_k^p} = |a | p_k^p - \delta_{a | k}^p \) is a parameter equal to one if link \( a \) is uncommon among \( p \) and \( k \), i.e., \( z_{a | p_k^p} = 1 \) if link \( a \) is on path \( p \) or \( k \), but not both, and is zero otherwise. Also, \( z_a \) is a binary variable that equals one if link \( a \) has a sensor, and zero otherwise. The objective function minimizes the number of sensors. Constraint (3b) requires the number of sensors on path \( p \) be at least 1. While, condition (3c) ensures at least one sensor is installed on an uncommon link among \( p \) and \( k \), for every pair of paths \( p \neq k \).

The following theorem shows this sensor location problem is an instance of the set covering problem, which is known to be NP-hard (Karp, 1972). Given a universal set \( \mathcal{U} \) and another set \( \mathcal{S} \), comprised of subsets of \( \mathcal{U} \) that satisfy \( \bigcup_{a \in \mathcal{S}} a = \mathcal{U} \); set covering problem is to find the minimum number of elements in \( \mathcal{S} \) whose union is \( \mathcal{U} \).

**Theorem 1.** OBSV is an instance of the set covering problem.

**Proof.** Define \( \mathcal{U}_1 = P \), \( \mathcal{U}_2 = \bigcup_{p \in k} \{p, k\} \), and \( \mathcal{U} = \mathcal{U}_1 \cup \mathcal{U}_2 \) as the universal set. Also, define \( \mathcal{S}_1 = \bigcup_{p \in k} \{p\} \), \( \mathcal{S}_2 = \bigcup_{p \in k} \{p, k\} \), and \( \mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2 \). In words, \( \mathcal{S}_1 \) is the set of all paths that will be detected by the sensor on link \( a \), and \( \mathcal{S}_2 \) is the set of all pairs of paths that will be distinguished by the sensor on link \( a \). Then the minimum sensor location problem can be stated as finding \( \mathcal{A} \subseteq \mathcal{A} \) with the minimum cardinality, such that \( \bigcup_{a \in \mathcal{S}} a = \mathcal{U} \).

Although this sensor location problem is NP-hard, some researchers, e.g., Castillo et al. (2008b) and Ng (2012), proposed algorithms that can solve the problem for large networks in numerical experiments. In theory, these algorithms can take an exponential amount of CPU time to find an optimal solution.

3.2. Illustration

Fig. 1 shows a network, a simplified version of the one in Hearn and Ramana (1998), with four OD pairs: (1, 3), (1, 4), (2, 3) and (2, 4). Each pair is connected by six paths, as shown in Table 1. An optimal solution to OBSV indicates that the minimum
number of sensors to distinguish the 24 paths in the preceding table is eight. The optimal sensor locations are marked by ✷ in Fig. 1 and the links with sensors are printed in boldface in Table 1. For every path, we define its signature as a subset of its links that are equipped with sensors. Path signatures for the nine-node network are presented in Table 1. Since the sensors in Fig. 1 produce a unique signature for each path, we can determine the path of any vehicle traveling on these 24 paths.

4. Implementing path-differentiated tolls using AVI sensors

This section assumes that path tolls in a network are given or pre-specified. The goal is to find the least number of sensors required to collect the predetermined amounts of tolls from the paths in the network. Clearly, an optimal solution to the path observability problem in Section 3 would ensure that every tolled path can be distinguished and tolls on these paths can be collected accurately. However, paths with the same toll need not be distinguished from each other. Below, we group paths with the same tolls together and paths in the same group are indistinguishable with respect to toll price. To further reduce the number of required sensors, it may also make sense to adjust individual path tolls to reduce the number of groups with distinctive tolls.

4.1. Model formulation

A mathematical model for the implementation problem can be derived from model OBSV by partially relaxing constraints (3b) and (3c) as follows:

\[
\begin{align*}
\text{(IMPL)} & \quad \min \sum_{a \in \mathcal{A}} z_a \\
\text{s.t.} & \quad \sum_{a \in \mathcal{A}} z_{ap}^{p} \geq 1 \quad \forall p \in \mathcal{P}, \quad \pi_p > 0 \quad (4b) \\
& \quad \sum_{a \in \mathcal{A}} z_{ak}^{k} \geq 1 \quad \forall p, k \in \mathcal{P}, \quad \pi_p \neq \pi_k \quad (4c) \\
& \quad z_a \in \{0, 1\} \quad \forall a \in \mathcal{A} \quad (4d)
\end{align*}
\]

where \(\pi_p\) is a given toll for path \(p\). Constraint (4b) is defined only for paths with positive toll. So, toll-free paths are not required to have equipped links. Similarly, constraints (4c) are defined for path pairs with different amounts of tolls, because paths that have equal tolls do not have to be distinguishable from each other.

4.2. Illustration

We solve model IMPL for the nine-node network, and the results are provided in Table 2. The path tolls we attempt to implement, as presented in Table 2, are the ones that minimize the financial burden on users, i.e., toll revenue, while replicating the SO flow distribution.

The equipped links in the optimal solution are printed in boldface in Table 2. These sensor locations support the implementation of the given path tolls. For example, if a vehicle is only detected by sensor 5, it will be charged 4.0 for using path 7; if it is detected by sensors 5 and 9, then it has to pay 8 for using path 1; and if detected by sensors 3, 5, and 9, then it will be charged 15.2 for using path 13. By inspecting the paths in Table 2, one can tell that all paths with positive toll are observable. Paths 8, 9, and 11 do not traverse any equipped link and thus travelers on those paths are not detected by any sensors. However, since they are toll-free, one can implement the given path tolls without installing any sensor on those links. Also, notice

\[\text{1 A path signature has been called a “scanning map” in Gentili and Mirchandani (2012).}\]

<table>
<thead>
<tr>
<th>OD Path</th>
<th>Links</th>
<th>Signature</th>
<th>OD Path</th>
<th>Links</th>
<th>Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 7 11</td>
<td>15 3 6 13 9</td>
<td>4 6 14 11</td>
<td>16 3 6 14 11</td>
<td>5 2 8 13 9</td>
<td>17 4 8 13 9</td>
</tr>
<tr>
<td>3 1 6 9</td>
<td>15 3 6 13 9</td>
<td>6 2 8 14 11</td>
<td>18 4 8 14 11</td>
<td>6 2 8 14 12</td>
<td>2 8 12</td>
</tr>
</tbody>
</table>
| [1, 4] 7 10 | 2 7 12 | [2, 4] 19 3 5 10 |}
that implementing this scheme requires installing six sensors, which is two less than the number of sensors needed for full observability.

4.3. Adjusting a given path-differentiated scheme

Although we assumed path tolls are predetermined, it remains possible to alter them without affecting the performance of the scheme. For example, if a path is so costly that it will never be utilized, then increasing its toll will not have any impact on any performance measure of the network. Given the path tolls, path set $P$ can be partitioned into two sets of usable, $P^+$, and unusable paths, $P^0$, as defined below:

$$P^0 = \bigcup_{w \in W} \{ p \in P^0 | t_p + \pi_p > u_w \}$$

$$P^+ = P - P^0$$

where $t_p$ and $u_w$ are the travel time of path $p$ and generalized cost of OD pair $w$, under the implementation of given path tolls. Thus, the first condition defines set $P^0$ as the set of paths with travel costs greater than the OD generalized cost. Using these sets, constraint (4c) can be divided into the following four sets of constraints:

$$\sum_{a \in A} z_{a, p}^{pk} \geq 1 \quad \forall p, k \in P^+, \text{ and } \pi_p \neq \pi_k$$  \hspace{1cm} (5a)

$$\sum_{a \in A} z_{a, p}^{pk} \geq 1 \quad \forall p \in P^+, k \in P^0, \text{ and } \pi_p < \pi_k$$  \hspace{1cm} (5b)

$$\sum_{a \in A} z_{a, p}^{pk} \geq 1 \quad \forall p \in P^+, k \in P^0, \text{ and } \pi_p > \pi_k$$  \hspace{1cm} (5c)

$$\sum_{a \in A} z_{a, p}^{pk} \geq 1 \quad \forall p, k \in P^0, \text{ and } \pi_p \neq \pi_k$$  \hspace{1cm} (5d)

If adjusting tolls is allowed, it might be possible to eliminate constraints (5c) and (5d) by increasing toll on unusable paths. For example, if $p \in P^+, k \in P^0,$ and $\pi_p > \pi_k,$ we can increase $\pi_k$ to the value of $\pi_p$ to eliminate the corresponding constraint. Although one can increase the tolls on unusable paths without affecting the equilibrium state, decreasing its toll may make an unusable path usable, and thus impact the performance of the pricing scheme, which we will avoid.

To implement the above idea, we first modify model IMPL by replacing constraint (4c) by (5a) and (5b), which is equivalent to relaxing constraints (5c) and (5d). So, the optimal solution to the relaxed model may not satisfy constraints (5c) and (5d). Suppose $p, q, k,$ and $l$ are four paths, such that $p, q \in P^0$ and $k, l \in P^+.$ Also, assume that $p$ is indistinguishable from $q,$ $k,$ and $l,$ by the optimal sensor locations. In the following, Theorem 2 proves indistinguishability of paths is an equivalence relation and subsequently Corollary 1 shows that every pairs of these paths are indistinguishable.

**Lemma 1.** If $z_{q, a}^{pk} = z_{p, a}^{pq} = 0,$ then $z_{a, q}^{pk} = 0$
Proof. Consider three paths, \( p, k, \) and \( q \), and let \( \sim \) represent the indistinguishability relation. This relation has the following properties:

1. (Reflexive): \( p \sim p \) \( \iff \forall a \in A, z_a = 0 \Rightarrow z_{ap} = 0 \Rightarrow p \sim p \).
2. (Symmetric): \( p \sim k \) \( \iff \sum_{a \in A} z_a = 0 \Rightarrow \sum_{a \in A} z_a = 0 \Rightarrow k \sim q \).
3. (Transitive): Using Lemma 1, we have:

   \[
   \begin{align*}
   p \sim k \Rightarrow \sum_{a \in A} z_a = 0 \Rightarrow \sum_{a \in A} z_a = 0 \Rightarrow k \sim q.
   \end{align*}
   \]

Since the indistinguishability relation is reflexive, symmetric, and transitive, it is an equivalence relation.

\[\square\]

Corollary 1. The indistinguishability relation partitions the set of paths into equivalence classes, where every pair of paths from the same class are indistinguishable.

Since paths \( q, k, \) and \( l \) are indistinguishable from \( p \), all four paths belong to the same indistinguishable class. Thus, \( \pi_p \) and \( \pi_q \) should be less than or equal to both \( \pi_k \) and \( \pi_l \), because otherwise the sensor locations do not satisfy constraint \((5b)\). Similarly, \( \pi_k \) should be equal to \( \pi_l \) because otherwise constraint \((5a)\) will be violated. Increasing \( \pi_p \) and \( \pi_q \) to be equal to \( \pi_k \) would eliminate the corresponding distinguishability constraints. Thus, the optimal sensor locations for the relaxed problem would become feasible to model IMPL if we increase the toll on unusable paths to the level of usable paths in the same class.

The results of implementing the above modification on the nine-node network are presented in Table 3. For this example, adjusting the tolls on unusable paths results in a scheme that needs five sensors, a reduction of one sensor or 17%, compared to model IMPL that requires six sensors.

5. Designing path-differentiated scheme for given sensor locations

In this section it is assumed that AVI sensors have been deployed for other applications, and we attempt to utilize these sensors to implement path-differentiated pricing. Thus, we discuss how to design path tolls that can be implemented by the given sensor locations.

5.1. Model formulation

If there are already enough sensors on the network that make all paths observable, any path-differentiated pricing scheme can be implemented. Otherwise, a scheme has to satisfy the two conditions mentioned in Section 3 to be implementable by the available sensors. These conditions can be enforced by the following constraints:

<table>
<thead>
<tr>
<th>Path</th>
<th>Links</th>
<th>Init. toll</th>
<th>Adj. toll</th>
<th>Path</th>
<th>Links</th>
<th>Init. toll</th>
<th>Adj. toll</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 5 9</td>
<td>8.0</td>
<td>8.0</td>
<td>13</td>
<td>3 5 9</td>
<td>15.2</td>
<td>15.2</td>
</tr>
<tr>
<td>2</td>
<td>2 7 11</td>
<td>0.0</td>
<td>3.2</td>
<td>14</td>
<td>4 7 11</td>
<td>3.2</td>
<td>3.2</td>
</tr>
<tr>
<td>3</td>
<td>1 6 13</td>
<td>0.0</td>
<td>3.2</td>
<td>15</td>
<td>3 6 13</td>
<td>7.2</td>
<td>7.2</td>
</tr>
<tr>
<td>4</td>
<td>1 6 14</td>
<td>0.0</td>
<td>0.0</td>
<td>16</td>
<td>3 6 14</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>5</td>
<td>2 8 13</td>
<td>0.0</td>
<td>3.2</td>
<td>17</td>
<td>4 8 13</td>
<td>3.2</td>
<td>3.2</td>
</tr>
<tr>
<td>6</td>
<td>2 8 14 11</td>
<td>0.0</td>
<td>0.0</td>
<td>18</td>
<td>4 8 14 11</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>7</td>
<td>1 5 10</td>
<td>4.0</td>
<td>4.0</td>
<td>19</td>
<td>3 5 10</td>
<td>8.0</td>
<td>8.0</td>
</tr>
<tr>
<td>8</td>
<td>2 7 12</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
<td>4 7 12</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>9</td>
<td>1 6 13 10</td>
<td>0.0</td>
<td>0.0</td>
<td>21</td>
<td>3 6 13 10</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>10</td>
<td>1 6 14 12</td>
<td>0.0</td>
<td>0.0</td>
<td>22</td>
<td>3 6 14 12</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>11</td>
<td>2 8 13 10</td>
<td>0.0</td>
<td>0.0</td>
<td>23</td>
<td>4 8 13 10</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>12</td>
<td>2 8 14 12</td>
<td>0.0</td>
<td>0.0</td>
<td>24</td>
<td>4 8 14 12</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>
\[ \pi_p = 0 \quad \forall p \in \mathbb{P}, \quad \sum_{a \in X} z_{ap} = 0 \]  
\[ \pi_p = \pi_k \quad \forall p, k \in \mathbb{P}, \quad \sum_{a \in A} z_{ap} = 0 \]  

where unlike the preceding sections, \( z_a \) is not a variable, but a known constant. The first set of constraints require paths without equipped links to be toll-free, and the second one requires indistinguishable paths to be charged the same toll amounts.

Adding the above constraints to model PATH yields a formulation for finding an optimal scheme that can be implemented with available sensors. Since constraint (6b) is defined over pairs of paths, its size grows approximately proportional to the square of the number of paths, \(|\mathbb{P}|^2\), which can be very large. However, using the concept of equivalence classes, from Corollary 1, allows a more concise formulation of the problem.

Paths in each equivalence class are indistinguishable and can only charge the same amount of toll. Hence, we will determine the amounts of tolls for equivalence classes rather than individual paths, and call them toll classes. This helps to remove constraint (6b) and make the model formulation more concise.

Given the sensor locations, we can classify the paths accordingly and formulate the following optimization problem to find the path-differentiated tolling that minimizes the total system travel time:

\[
\begin{align*}
\text{(DES)} & \quad \min \sum_{p \in \mathbb{P}} f_p t_p(f) \\
\text{s.t.} & \quad f \in \mathbb{F} \quad (7a) \\
& \quad f_p(t_p(f) + \pi_{cl(p)} - u_w) = 0 \quad \forall w \in \mathbb{W}, \quad p \in \mathbb{P}_w \quad (7b) \\
& \quad t_p(f) + \pi_{cl(p)} - u_w \geq 0 \quad \forall w \in \mathbb{W}, \quad p \in \mathbb{P}_w \quad (7c) \\
& \quad \pi_{cl(p)} \geq 0 \quad \forall p \in \mathbb{P} \quad (7d) \\
& \quad \pi_0 = 0 \quad (7e) 
\end{align*}
\]

where \( cl(p) \) is the toll class of path \( p \), and thus \( \pi_{cl(p)} \) is the amount of toll for path \( p \). We dedicate class 0 to undetected paths, which should be toll-free. Similar to model PATH, model DES is difficult to solve, because it contains complementarity constraints. Effective solution algorithms for such mathematical programs with complementarity constraints can be found in, e.g., Lawphongpanich and Yin (2010), Wu et al. (2011) and Wu et al. (2012).

5.2. Illustration

To illustrate the idea, assume that links 3, 6, 10, and 12 of the nine-node network are equipped with sensors. Table 4 shows the indistinguishable paths, and assigns an ID to each class of paths. The paths are partitioned into 11 classes, each charging a different amount of toll.

Table 4 also presents an optimal solution to model DES. This path-differentiated scheme reduces the total travel time of the network from 2455.9, to 2253.92, a reduction of 201.98 units. In comparison, an optimal anonymous link-based scheme, which charges toll on links 3, 6, 10, and 12, will reduce the travel time to 2326.59, a reduction of 129.31. This example illustrates how our new scheme outperforms the anonymous link-based counterpart.

6. Simultaneous design of path-differentiated tolls and sensor locations

The goal of this section is to design a path-differentiated scheme that induces a target link-flow distribution with the least number of sensors. The flow distribution can be the one that minimizes total travel time, air pollution, or any other objective that is a function of link flows. We first discuss the link-based version of the problem, and then present the path-differentiated counterpart. A solution algorithm and a numerical example are presented afterwards to illustrate the idea.
6.1. Model formulation

6.1.1. Minimum sensor locations for link-based tolls

A similar minimum sensor location problem for link-based tolls has been introduced by Hearn and Ramana (1998), and studied by others such as Bai (2004), Bai and Rubin (2009), and Bai et al. (2010). The problem defined in Hearn and Ramana (1998) is to find a minimum number of toll collection facilities, their locations, and toll amounts such that the resulting link flows replicate SO. Below is a mathematical formulation of this problem:

\begin{align}
\text{(MSL-L)} & \quad \min \sum_{a \in A} z_a \\
\text{s.t.} \quad f & \in F \\
\sum_{p \in F} \delta_p f_p & = \bar{\lambda}_a \quad \forall a \in A \tag{8b} \\
\bar{t}_p + \sum_{a \in A} \beta_a \delta_a f_p - u_w & \geq 0 \quad \forall w \in \mathcal{W}, \ p \in \mathcal{P} \tag{8c} \\
\sum_{w \in \mathcal{W}} \left( \bar{t}_p + \sum_{a \in A} \beta_a \delta_a f_p \right) f_p & = \sum_{w \in \mathcal{W}} u_w q_w \tag{8d} \\
0 & \leq \beta_a \leq z_a M \quad \forall a \in A \tag{8e} \\
z_a & \in \{0, 1\} \quad \forall a \in A \tag{8f} \\
u_w & \geq 0 \quad \forall w \in \mathcal{W}, \ p \in \mathcal{P} \tag{8g}\end{align}

where \( M \) is the largest possible toll; \( \bar{\lambda} \) is the vector of SO link flows that is the target distribution, and \( \bar{t}_p \) is the path travel time for the target distribution, i.e., \( \bar{t}_p = \sum_{a \in A} \delta_a t_a(\bar{\lambda}_a) \).

The MSL-L formulation as presented above is a linear mixed-integer program. For every link \( a \), denoted by \( z_a \), is a binary variable that is equal to 1 if a sensor is located on link \( a \), and 0 otherwise. The objective function minimizes the number of sensors. Constraints (8d) and (8e) represent the tolled UE conditions. Constraint (8f) ensures all tolled links are equipped with sensors.

Bai (2004) showed this problem is NP-complete. Bai (2004), Bai and Rubin (2009), and Bai et al. (2010) proposed different analytic and heuristic solution algorithms for this problem. Despite these developments, solving this problem for large size networks is still challenging.

6.1.2. Minimum sensor locations for path-differentiated tolls

In the MSL-L model presented above, a sensor has a single functionality of collecting toll. However, as discussed in Section 1, sensors deployed in toll collection systems can work as AVI sensors, and support the implementation of path-differentiated pricing. We consider this approach to be an improvement over MSL-L, i.e., the model presented here generally requires a less number of sensors to achieve the same target link flow distribution. This improvement will be illustrated with numerical examples later in this section.

The following is a linear integer formulation for this problem:

\begin{align}
\text{(MSL-P)} & \quad \min \sum_{a \in A} z_a \tag{9a} \\
\text{s.t.} \quad f & \in F \tag{9b} \\
\sum_{p \in F} \delta_p f_p & = \bar{\lambda}_a \quad \forall a \in A \tag{9c} \\
f_p & \leq (1 - y_p) M \quad \forall p \in \mathcal{P} \tag{9d} \\
\bar{t}_p + \pi_p - u_w & \leq y_p M \quad \forall w \in \mathcal{W}, \ p \in \mathcal{P} \tag{9e} \\
\bar{t}_p + \pi_p - u_w & \geq 0 \quad \forall w \in \mathcal{W}, \ p \in \mathcal{P} \tag{9f} \\
\pi_p & \leq M \sum_{a \in A} z_a \delta_a \tag{9g} \\
\pi_p - \pi_k & \geq -M \sum_{a \in A} z_a \delta_{ap} \quad \forall p, k \in \mathcal{P} \tag{9h} \\
y_p, z_a & \in \{0, 1\} \quad \forall a \in A, \ p \in \mathcal{P} \tag{9i}\end{align}

In MSL-P, the first two sets of constraints are restricting path flows to the ones that are feasible and yield the target aggregate link flows. The next three constraints ensure the tolled UE conditions. The summation on the right hand side of constraint (9g) is the number of sensors on path \( p \). When this summation is equal to zero, path \( p \) has to be toll-free. The summation on the right hand side of constraint (9h) indicates the number of sensors located on uncommon links of the paths \( p \) and \( k \). If this summation is greater than zero, then paths \( p \) and \( k \) are distinguishable and can charge different amounts of toll. Otherwise, constraint (9h) indicates \( \pi_p - \pi_k \geq 0 \) and \( \pi_k - \pi_p \geq 0 \), which means these two paths will have the same amounts of toll.
It is important to point out that model MSL-P, can be too large to solve. One major reason for that is the number of paths grows exponentially with the size of the network, and the number of constraints \((9h)\) is proportional to the square of number of paths, i.e., \(|P|^2\). In Section 6.3 we present a heuristic approach that can find good solutions within limited time.

### 6.2. Illustration

We solve model MSL-P for the nine-node network with the SO distribution being the target link flow. Table 5 shows all 24 paths in the nine-node network and the corresponding travel time, toll, total cost and flow.

The equipped links, which are printed in boldface in the column of links, are 3, 5, and 12. Notice that each tolled path includes at least one equipped link, and each pair of paths with different toll have at least one uncommon equipped link. The minimum number of sensors required to replicate the target link-flows is 3, which represents a 40% improvement compared to model MSL-L that requires 5 sensors to achieve the same. Notice that the cost column shows some paths with minimum cost that are not utilized, suggesting that there might be multiple path flow solutions to this problem. However, as discussed in Zangui et al. (2014), all of them yield the same target link flows.

Table 6 presents the tolling scheme. For example, if a vehicle is only detected by one of sensors 3 or 12, it will be charged 4 units of toll. If it is detected by both sensors, the toll would be 0.8 units. Vehicles detected on link 5 will be charged 8, and those detected on both links 3 and 5 will be charged 12. Also, notice that some combinations of sensors, e.g., 5 and 12, are not relevant, because none of the paths contain both of these links.

### 6.3. Heuristic solution algorithm

As mentioned earlier, the number of constraints \((9h)\) is proportional to the square of the number of paths, which can be very large. In addition to the large number of constraints, model MSL-P has numerous variables, proportional to the number of paths. The solution method proposed here addresses both of these issues.

Recall that if an uncommon link of paths \(p\) and \(k\) is equipped with a sensor, these two paths are distinguishable and their tolls can be different, i.e., the corresponding constraint \((9h)\) will be automatically satisfied. Therefore, a pair of paths with many uncommon links have a high probability of being distinguishable with an arbitrary set of sensor locations such as the optimal sensor locations. For example, assume paths \(p\) and \(k\) have 4 uncommon links and sensors are to be installed on 30 percent of the links. If sensor locations are completely random, there is a probability of \(1 - (1 - 0.3)^3 = 0.7\) for the two paths to become distinguishable. Our proposed method thus initially relaxes constraint \((9h)\) for path pairs with high number of uncommon links, and then add them back to the model iteratively, as needed. This approach makes the solution process more efficient by eliminating some of the constraints, but regardless of which constraints are removed initially, the final solution satisfies all the constraints.

To facilitate the presentation of our solution algorithm, we define three new sets: \(Z\) as the set of equipped links at the current iteration, \(U\) as the set of paths with no equipped link, and \(S\) as the set of path pairs that, with a high probability,
will not be distinguishable in the final solution. The method initializes \( Z \) as an empty set, and then add sensors to it iteratively. For a given set \( Z \), \( \sum_{a \in \mathcal{A}} \delta_a^p \) would be the number of equipped links on path \( p \). So, set \( U \) can be represented as \( U(Z) = \{ p \in \mathcal{P} | \sum_{a \in \mathcal{A}} \delta_a^p = 0 \} \). In addition, \( \sum_{a \in \mathcal{A}} \delta_a^p \) and \( \sum_{a \in \mathcal{A}} \delta_a^{pk} \) are the number of uncommon links and sensors installed on uncommon links of paths \( p \) and \( k \), respectively. As before, a pair of paths with uncommon equipped links can have different tolls. If a path pair have many uncommon links, e.g., greater than \( \bar{n} \), then they have a high probability of becoming distinguishable with the final set of sensors. So, the set of paths that are indistinguishable in the current iteration and have a low probability of being distinguishable in the final iteration can be shown as follows:

\[
S(Z) = \left\{ (p, k) \in \mathcal{P}^2 | \sum_{a \in \mathcal{A}} \delta_a^{pk} \leq \bar{n}, \text{ and } \sum_{a \in \mathcal{A}} \delta_a^{pk} = 0 \right\}
\]

For a given set \( Z \), we relax model MSL-P as follows and solve it with an iterative approach.

(MSL-R) \quad \text{min} \sum_{a \in \mathcal{A}} z_a \tag{10a}

\[ \text{s.t.} \quad f \in \mathcal{F} \quad \sum_{p \in \mathcal{P}^f} \delta_{pf} = \bar{x}_a \quad \forall a \in \mathcal{A} \tag{10b} \]

\[ f_p \leq (1 - y_p)M \quad \forall p \in \mathcal{P} \tag{10c} \]

\[ t_p + \pi_p - u_w \leq y_p M \quad \forall w \in \mathcal{W}, \ p \in \mathcal{P}_w \tag{10d} \]

\[ t_p + \pi_p - u_w \geq 0 \quad \forall w \in \mathcal{W}, \ p \in \mathcal{P}_w \tag{10e} \]

\[ \pi_p \leq M \sum_{a \in \mathcal{A}} z_a \delta_a^p \quad \forall p \in \mathcal{U} \tag{10f} \]

\[ \pi_p - \pi_k \geq -M \sum_{a \in \mathcal{A}} z_a \delta_a^{pk} \quad \forall (p, k) \in \mathcal{S} \tag{10g} \]

\[ y_p, \ z_a \in \{0, 1\} \quad \forall a \in \mathcal{A}, \ p \in \mathcal{P} \tag{10i} \]

where constraints (10g) and (10h) are the partially relaxed versions of (9g) and (9h).

Set \( Z \) in the above model will be initialized as an empty set. After every iteration the newly equipped links are added to \( Z \), i.e., \( Z = Z \cup \{ a \in \mathcal{A} | z_a = 1 \} \). The value of \( \bar{n} \) should be chosen such that the cardinality of set \( \mathcal{S} \) is small enough to make model MSL-R solvable. A lower value of \( \bar{n} \) results in a smaller \( |\mathcal{S}| \) and more relaxed constraints. As sensors are added to \( Z \) at every iteration, more path pairs become distinguishable, shrinking the size of \( \mathcal{S} \). As a result, the value of \( \bar{n} \) can be increased at every iteration, without drastically increasing the size of the problem. At the final iteration, the value of \( \bar{n} \) should be large enough to avoid excluding any pair of paths from \( \mathcal{S} \). So, in the final iteration, only the constraints that are already satisfied with the sensors in set \( Z \) are relaxed.

Since our formulation is path-based, we need a method to generate paths instead of enumerating all of them. Hence, we also adopt a path generation method that adds the required paths iteratively, instead of generating all the possible paths. Here, we use model PATH-GEN to generate new paths for every OD pair. This model is similar to the shortest path problem, but has some extra constraints, i.e., constraints (11c), which are called canonical cuts, ensuring that the generated path does not already exist in set \( \mathcal{P}_w \). Then, we find the class of the newly generated path and its associated toll. If the total cost, i.e., travel time plus toll, of the path is less than its corresponding OD generalized travel cost, it will be added to the path set. For each OD pair, we continue generating new paths until the travel time of the new path is greater than its OD generalized travel cost.

(PATH-GEN) \quad \text{min} \sum_{i,j \in \mathcal{A}} h_{ij} t_{ij}(x_{ij}) \tag{11a}

\[ \text{s.t.} \quad \sum_{j \in \mathcal{A}} h_{ij} - \sum_{k \in \mathcal{A}} h_{ik} = E_i^k \quad \forall k \in \mathcal{N} \tag{11b} \]

\[ \sum_{j \in \mathcal{P}} h_{ij} - \sum_{j \in \mathcal{P}} h_{ij} \leq n_p - 1 \quad \forall p \in \mathcal{P}_w \tag{11c} \]

\[ h_{ij} \in \{0, 1\} \quad \forall ij \in \mathcal{A} \]
where \( r \) is origin, \( s \) is destination, \( E^r_s = 1 \), \( E^s_r = -1 \), and all other elements of \( E^r \) are zero. Also, \( n_p \) is the number of links comprising path \( p \). A decision variable, \( h_{ij} \), is equal one if the generated path traverses link \( ij \), and zero otherwise.

**Algorithm 1.** Heuristic algorithm for sensor location problem.

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:</td>
<td>For each OD pair, construct the initial set of paths ( P_w ).</td>
</tr>
<tr>
<td>2:</td>
<td>Set ( Z = S = U = \emptyset ) and solve model MSL-L.</td>
</tr>
<tr>
<td>3:</td>
<td>Choose a value for ( n ), construct ( S ) accordingly, and set ( U = P ).</td>
</tr>
<tr>
<td>4:</td>
<td>Solve model MSL-R. Let ( LB ) and ( Z ) represent the optimal value of objective function, and optimal sensor locations.</td>
</tr>
<tr>
<td>5:</td>
<td>repeat</td>
</tr>
<tr>
<td>6:</td>
<td>Fix the sensor locations in the solution, ( Z = Z \cup \emptyset ).</td>
</tr>
<tr>
<td>7:</td>
<td>Update value of ( n ) if needed.</td>
</tr>
<tr>
<td>8:</td>
<td>for all OD-pairs ( w ) do</td>
</tr>
<tr>
<td>9:</td>
<td>repeat</td>
</tr>
<tr>
<td>10:</td>
<td>Solve model (PATH-GEN) to find the shortest path that is not already in ( P_w ), call it ( p_{new} ), and denote the optimal value of objective function by ( l ).</td>
</tr>
<tr>
<td>11:</td>
<td>Find the toll class of ( p_{new} ) and the corresponding toll, ( \pi_{cl(p_{new})} ).</td>
</tr>
<tr>
<td>12:</td>
<td>if ( l + \pi_{cl(p_{new})} \leq u_w ) then</td>
</tr>
<tr>
<td>13:</td>
<td>Add ( p_{new} ) to ( P_w ), ( P_w = P_w \cup { p_{new} } ).</td>
</tr>
<tr>
<td>14:</td>
<td>end if</td>
</tr>
<tr>
<td>15:</td>
<td>until ( l &gt; u_w )</td>
</tr>
<tr>
<td>16:</td>
<td>end for</td>
</tr>
<tr>
<td>17:</td>
<td>Re-solve model MSL-R and let ( Z ) represent the optimal sensor locations.</td>
</tr>
<tr>
<td>18:</td>
<td>until no new path is added to any ( P_w ) and ( Z = \phi ).</td>
</tr>
</tbody>
</table>

At every iteration of **Algorithm 1**, except the final one, at least one path is being generated. Since the number of paths is finite, the algorithm will stop after a finite number of iterations. The convergence to an optimal solution is contingent upon the proper way of updating \( n \). If the value of \( n \) at the final iteration is sufficiently large, e.g., equal to the number of nodes minus one, when the algorithm stops, the path tolls induce the target link-flow and are implementable by the sensors in \( Z \). The algorithm thus converges to a feasible solution.

**6.4. Numerical example**

This section presents the results of solving the minimum sensor location problem for path-differentiated tolls for the Sioux Falls network, Fig. 2, using the solution algorithm proposed in the preceding subsection. Results of models MSL-L and MSL-P are also compared to show the advantage of implementing path-differentiated tolls over their anonymous link-based counterpart.

For this example, we choose to replicate the link-flows that minimize total system travel time, i.e., SO. The initial path set for each OD pair is constructed by adding all the system optimum paths, i.e., paths with minimum marginal travel time, and paths that are not longer (in terms of travel time) than the longest SO path. The size of the initial path set is 1096, i.e., \( |P| = 1096 \).

We relax constraints (9h) for the path pairs with 3 or more uncommon links, i.e., \( \bar{n} = 2 \). Model MSL-R is then solved, by calling Gurobi Optimization (2013) 5.5 solver in C# environment. We terminate the solution process after finding an integer feasible solution, when the best obtained solution is 21, and the best lower-bound is 17. We then fix the location of these 21 sensors on the network, by adding them to set \( Z \). The violated constraints are also added back to model MSL-R. A total of 199 new paths are generated by solving model PATH-GEN and added to set \( P \).

For the second iteration, the value of \( \bar{n} \) is increased to 23, one less than the number of nodes, so that none of the constraints (9h) is relaxed except for the paths that are already distinguishable. Because set \( Z \) has already 21 sensors, most of the path pairs are distinguishable and the size of problem does not increase drastically.

Generating paths and resolving model MSL-R are repeated until no new path is added to the path set, \( P \). For this example, we have to repeat this loop six times. The final path set has 1601 paths, and the best obtained solution contains 24 sensors. The sensor locations are marked with a \( \Diamond \) in Fig. 2.

The number of sensors that are needed to achieve the same performance using the anonymous link-based schemes is 32. As discussed earlier, the type of sensors that are used for implementing link-based schemes often belong to the AVI category, which are also the type of sensors we propose for implementing path-differentiated tolls. Thus, our proposed scheme has reduced the number of toll facilities, required to achieve the same level of congestion, by more than 30%.
7. Conclusion

In our previous studies, we have shown that path-differentiated schemes can offer a much better performance compared to traditional link-based schemes. Recognizing that toll collection devices, e.g., toll-tag readers, or license plate scanners, can identify vehicles and collect location information required to obtain their paths, this paper proposed an innovative method to implement path-differentiated pricing using infrastructures similar to link-based anonymous pricing.

We have investigated different versions of the sensor location problem (see Table 7 for a summary) and illustrated them on a small nine-node network. We have also used a heuristic method to solve our most general model for the Sioux Falls network. The results of the proposed scheme are superior to the link-based counterpart. More specifically, for the Sioux Falls network, our method can reduce the infrastructure required to achieve the same level of performance by more than 30%.

While this paper assumes that sensors have to be located on links, it can be more efficient to locate them on nodes so that one sensor can detect multiple roads. In fact, the problem of optimally locating counting sensors on the nodes of networks
for observing traffic flows has been studied by, e.g., Bianco et al. (2001), Bianco et al. (2006), and Morrison (2008). Implementing path-differentiated pricing schemes by locating AVI sensors on nodes can be an interesting extension.

Acknowledgements

This research was partly funded by the U.S. National Science Foundation (CNS-1239364 and CMMI-1362631) and National Natural Science Foundation of China (71228101).

Appendix A

It is worth mentioning that model MSL-P should include all the usable paths, whether they contain loops or not. To explain this, consider the network in Fig. 3, where the travel time of every link is equal to 1. The toll for vehicles detected only on link 1 is 3 and for those detected on both links 1 and 2 is 0. In this case, a vehicle traveling from origin a to destination b would be better-off if it passes node b, go to node c and comes back to node b. This example shows the need to consider any path, whether it has loop or not, in model MSL-P. For this reason, model PATH-GEN does not have any constraint for avoiding loops.

References