Camera location optimisation for traffic surveillance in urban road networks with multiple user classes

Xiao-Shan Lu a, Hai-Jun Huang a & Jiancheng Long b

a School of Economics and Management, Beihang University, Beijing 100191, China
b School of Transportation Engineering, Hefei University of Technology, Hefei 230009, China

Published online: 08 May 2012.

To cite this article: Xiao-Shan Lu, Hai-Jun Huang & Jiancheng Long (2013) Camera location optimisation for traffic surveillance in urban road networks with multiple user classes, International Journal of Systems Science, 44:12, 2211-2222, DOI: 10.1080/00207721.2012.685776

To link to this article: http://dx.doi.org/10.1080/00207721.2012.685776
Camera location optimisation for traffic surveillance in urban road networks with multiple user classes

Xiao-Shan Lu, Hai-Jun Huang and Jiancheng Long

School of Economics and Management, Beihang University, Beijing 100191, China; School of Transportation Engineering, Hefei University of Technology, Hefei 230009, China
(Received 14 October 2011; final version received 13 March 2012)

New sensor technologies (e.g. surveillance cameras, loop detectors) enable the synthesis of disaggregated vehicle information from multiple locations. This article studies the camera location problem for traffic surveillance in urban road networks with multiple user classes. All users are differentiated by their own acceptance degree of camera monitoring and make their route choices in a logit-based stochastic user equilibrium manner. A bi-level programming model is proposed to formulate the problem and solved by the sensitivity analysis based branch and bound method. Numerical examples are presented to illustrate the model application and show the effectiveness of the solution method.

Keywords: traffic assignment; bi-level programming; camera location; traffic surveillance; sensitivity analysis; branch and bound method

1. Introduction

In recent years, there has been considerable interest in developing and evaluating intelligent transportation systems (ITS) from both transportation practical and academic circles (Wang and Niu 2011). As an important component of ITS, the traffic surveillance/control system observes the real time traffic conditions and provides information services for all travellers. Camera is one of the main equipments for surveillance. This technology has two primary advantages: on one hand, it can reduce the number of accidents caused by violation of traffic regulations due to the camera’s deterrent force through monitoring and recording the drivers’ behaviours; on the other hand, it can enhance the speed of handling various traffic accidents through sending policemen real time pictures.

In cities such as Beijing, a visible sign at every camera location is provided for all drivers to advise them to abide by traffic rules. However, the cameras are not installed on all road segments because of budget constraints. With the accumulation of driving experiences, most drivers become familiar with the distribution of cameras. Recently, it was reported that people were against traffic cameras due to the privacy issue or potential fine risk (News 13 2011; Total traffic Los Angeles 2011). Then, some drivers may choose the routes without or with fewer cameras for protecting their privacy against being recorded or avoiding from being fined once broking traffic regulations. This implies that the introduction of camera surveillance may result in potential costs for some travellers. Therefore, a trade-off between the normal travel cost and the potential cost caused by camera surveillance should be considered when travellers make their route decisions. An interesting problem arises naturally: how to optimise the camera locations for maximising the flow coverage by taking the drivers’ actions into account? Obviously, travellers who have different acceptance degrees of camera monitoring (ADCM) will behave differently in their route choices. To consider this kind of user heterogeneity, all drivers must be divided into limited groups according to their ADCMs, or a continuously distributed ADCM across the whole drivers must be assumed.

The transportation network design problem (NDP) has been extensively studied in the past few decades. For a comprehensive review, interested readers can refer to Boyce (1984), Bell and Iida (1997) and Yang and Bell (2001). Typically, the NDP can be classified into two categories: continuous NDP (e.g. Marcotte 1986; Davis 1994; Meng, Yang, and Bell 2001; Sadabadi, Zokaei-Aashtiani, and Haghani 2008; Meng and Liu 2011) and discrete NDP (e.g. Long, Gao, Zhang, and Szeto 2010). The continuous NDP mainly focuses on road pricing problem (Yang and Bell 1997), signal control problem (Gao and Song...
2002; Wong and Wong 2002) and link capacity improvement problem (Huang, Wang, and Bell 2001).

The discrete NDP usually deals with the selection of link additions to an existing road network. Yang and Zhou (1998) studied the optimisation problem of traffic counting points in a road network. They proposed a rule for excluding links without new information for traffic counting. Recently, Li and Ouyang (2011) developed a reliable facility location model to optimise both the flow coverage for traffic volume statistics and the path coverage for travel time estimation. The camera location problem is similar to the discrete NDP, which takes the individual behaviour into account. However, the difference between these two types of problems is significant due to the particular purpose of traffic surveillance and the responses from heterogeneous drivers.

In this article, a camera location problem with consideration of travellers’ ADCM is proposed. We develop a bi-level programming model to determine the optimal locations of cameras in urban road networks with multiple user classes. All users are differentiated by their own ADCMs and are assumed to choose their routes in a logit-based stochastic user equilibrium (SUE) manner. The upper level is for monitoring most traffic flow while the lower level is for depicting the drivers’ route choices under camera surveillance. A sensitivity analysis based branch and bound (B&B) method is designed to solve the proposed problem. We will numerically investigate the impacts of camera locations on traffic surveillance by examples.

This article is organised as follows. In Section 2, some notations are first introduced and the bi-level programming model is then formulated. The B&B method based on sensitivity analysis is presented in Section 3. In Section 4, numerical results in an example network are presented. Section 5 concludes the article.

2. The bi-level programming model

2.1. Notations

Consider a network $G = (N, A)$, where $N$ denotes the set of nodes and $A$ the set of links in the network. Let $R$ and $S$ denote the sets of origins and destinations, respectively. The following notations are adopted throughout this article:

- $r$ an origin, $r \in R$;
- $s$ a destination, $s \in S$;
- $x_{ar}$ flow on link $a$, $a \in A$;
- $C_a$ capacity of link $a$, $a \in A$;
- $\mathbf{x}$ row vector of link flows, $\mathbf{x} = [x_{ar}]$;
- $K_{rs}$ set of all paths connecting $r$ and $s$;
- $[K_{rs}]$ = the number of paths in $K_{rs}$;
- $\delta_{ak}$ link-path incidence relationship, i.e. 1 if link $a$ is on path $k$ connecting $r$ and $s$, and 0 otherwise;
- $f_{mk}^r$ flow of user class $m$ on path $k$ connecting $r$ and $s$;
- $p_{mk}^r$ probability of choosing path $k$ by user class $m$;
- $c_{mk}^a$ travel cost of user class $m$ on path $k$;
- $q_{mk}^r$ travel demand of user class $m$ from $r$ to $s$;
- $t_a(X)$ travel cost function of link $a$;
- $y_a$ zero-one decision variable, $y_a = 1$ if a camera is installed on link $a$, $y_a = 0$ otherwise;
- $\mathbf{y}$ row vector of camera location variables, $\mathbf{y} = [y_a]$;
- $M$ set of user classes in the network, $|M| = $ the number of classes in $M$;
- $\beta_m$ additional cost of user class $m$ when recorded by camera, $m = 1, 2, \ldots, |M|$;
- $N_1$ number of cameras to be installed.

2.2. Logit-based SUE assignment

Travellers’ route choice behaviour plays a vital role in the camera location problem. In general, the installation of cameras definitely induces changes in traffic flow pattern throughout the network, since some drivers may shift to other routes without or with fewer cameras for protecting their privacies against being recorded or avoiding from refinement once breaking traffic regulations. This says, the fear to camera monitoring raises additional travel cost for these routes installed with cameras. Improper camera locations may bring a reduction in total observed flow and cut down the efficiency of surveillance. Therefore, a behaviourally sound model for an accurate prediction of traffic flow pattern is essential to the camera location process.

In analytical modelling approaches, two types of traffic assignment principles, namely user equilibrium (UE) and system optimum (SO) ones, are widely used to describe travellers’ route choice behaviour. The SO requires that all travellers can choose routes for minimising the total cost in the network. The UE assumes that all travellers choose routes for minimising their individual travel costs. The UE can be further defined as deterministic UE and SUE according to assumptions on travel cost perception. Davis (1994), Akamatsu (1996), Huang and Bell (1998), Yang, Meng, and Bell (2001), Leung and Lai (2002), Lo and Szeto (2002), Yin and Yang (2003), Huang and Li (2007) and Long et al. (2010) employed the logit-based SUE to formulate the route choice behaviours of drivers. This approach is also adopted in this study.
For simplicity and clarity, the travel demand of each class of users is supposed to be given and fixed. Hence, we are dealing with a multi-class traffic network equilibrium problem with fixed-demand.

The partial derivative of the satisfaction function with respect to route travel cost is the route choice probability (Sheffi 1985), i.e.

\[ p_{m}^{rs} = \frac{\partial s_{m}^{rs}}{\partial c_{mk}^{rs}} = \exp(-\theta c_{mk}^{rs}) \sum_{k' \in K^{rs}} \exp(-\theta c_{mk'}^{rs}), \]

\[ r \in R, s \in S, k \in K^{rs}, m \in M, \]

(1)

where \( s_{m}^{rs} \) is a so-called satisfaction function, defined as the expected minimal perceived travel cost of class \( m \) from \( r \) to \( s \). The positive parameter \( \theta \) measures the sensitivity of route choices to travel cost. The flow will be assigned to fewer paths when \( \theta \) becomes larger. For the logit-based SUE, the satisfaction function is

\[ s_{m}^{rs} = -\frac{1}{\theta} \ln \sum_{k \in K^{rs}} \exp(-\theta c_{mk}^{rs}). \]

(2)

Then, the path and link flows are given by

\[ f_{k}^{rs} = \sum_{m \in M} f_{mk}^{rs} = \sum_{m \in M} q_{m}^{rs} p_{m}^{rs}, \quad r \in R, \ s \in S, \ k \in K^{rs}, \]

(3)

and

\[ x_{a} = \sum_{r \in R} \sum_{s \in S} \sum_{k \in K^{rs}} f_{k}^{rs} s_{m}^{rs}, \quad a \in A. \]

(4)

According to Ying, Lu, and Shi (2007), the SUE can be characterised by the following nonlinear equations

\[ F_{a}(x, y) = x_{a} - \sum_{m \in M} \sum_{r \in R} \sum_{s \in S} \sum_{k \in K^{rs}} f_{mk}^{rs} \frac{\partial s_{m}^{rs}}{\partial c_{mk}^{rs}} = 0, \quad a \in A. \]

(5)

If a driver is recorded by a camera, he or she will endure an additional cost, which is explicitly reflected by a cost-associated positive parameter \( \beta_{m} \). Travellers who do not care about camera monitoring have higher ADCM. The higher ADCM, the smaller \( \beta_{m} \) and vice versa. Those drivers who very worry about privacy or punishment caused by breaking traffic regulations are valued by larger \( \beta_{m} \). Obviously, \( \beta_{m} \) is a crucial parameter which should be calibrated through questionnaire to drivers. Then, the travel cost of a specific user class \( m \) on path \( k \) between \( r \) and \( s \) can be expressed by

\[ c_{mk}^{rs} = \sum_{a \in A} (t_{a}(x) + y_{a} \beta_{m}) \frac{\partial s_{m}^{rs}}{\partial c_{mk}^{rs}}, \]

\[ r \in R, \ s \in S, \ k \in K^{rs}, \ m \in M. \]

(6)

Equation (6) means that the route potential cost is link-additive and not route-specific. So, it may give higher usage probability to paths containing fewer cameras. Considering the road congestion, we let the cost functions of all links follow the BPR type function (developed by the US Bureau of Public Road, Sheffi 1985), i.e.

\[ t_{a}(x) = t_{0}^{a}(1 + \chi_{1}(x_{a}/C_{a})^{\chi_{2}}), \quad a \in A, \]

(7)

where \( t_{0}^{a} \) is the free-flow travel cost on link \( a \), \( \chi_{1} \) and \( \chi_{2} \) the two parameters.

2.3. The upper level optimisation problem

Usually, a classical NDP involves two groups of players: the network planner and the network users. In our study, the route choice of network users follows the logit-based SUE principle and the duty of network planner is to determine the camera positions for a specific purpose. Li and Ouyang (2011) assumed that the sensor at a node can detect all traffic passing that node from different directions. Since a camera has a limited effective range for flow inspection, we here assume all cameras are installed on road sides and detect single-direction flows only.

Let \( y_{a} \) be a zero-one decision variable, \( y_{a} = 1 \) if a camera is installed on link \( a \) and \( y_{a} = 0 \) otherwise. The upper level is to solve \{\( y_{a} \)\} for monitoring traffic flow as most as possible, i.e.

\[ \max_{(x, y)} G(x, y) = \sum_{a \in A} x_{a}(y) y_{a}, \]

subject to

\[ \sum_{a \in A} y_{a} \leq N_{1}, \]

(9)

\[ y_{a} \in \{0, 1\}, \quad a \in A, \]

(10)

where \( x(y) = [x_{a}(y)] \) is an implicit function of \( y \), which can be obtained by solving the logit-based SUE assignment problem (5). Constraint (9) requires the total number of cameras installed to be less than a predetermined value due to budget limit. Constraint (10) is the binary decision variable for indicating camera location.

3. Solution algorithm

In order to solve the bi-level programming problem, we must evaluate the equilibrium flow pattern for a given vector of camera installation decision variables. It is difficult to directly evaluate the equilibrium link flow pattern because each link flow is a nonlinear implicit function of the decision variables. For the continuous
NDPs (i.e. all decision variables are continuous), the sensitivity analysis based method is known to be efficient (Patrïksson 2004; Ying and Yang 2005; Ying et al. 2007). In our study, however, \( y \) is a vector of binary decision variables in the upper level problem, the \( \text{B&B} \) method is a preferential approach to solve the nonlinear mixed integer bi-level programming problem. Next, we implement the sensitivity analysis into the \( \text{B&B} \) framework for developing an efficient solution algorithm.

### 3.1. Solving the lower level SUE problem

The algorithms for solving the lower level SUE problem can be classified into two categories: link- and path-based algorithms. Although the link-based algorithms do not require explicit enumeration of paths, the implementation of implicit path choice sets may be unrealistic from a behavioural viewpoint (Bekhor and Toledo 2005). However, the path choice set can be changed during the solution procedure, which may lead to non-convergence of the solution algorithm. Even if the column generation method is adopted, solution convergence may not be guaranteed (Meissner, Strauss, and Talluri 2011). In addition, the definition of efficient path set may lead to the problem that some feasible and shorter paths are omitted. On the contrary, the choice sets generated by the path-based algorithms can accommodate to more realistic scenarios. With the rapid development of computer technologies, it is possible to enumerate the path choice set for large-scale networks. Thus, the path-based algorithm is more suitable to solve the lower level SUE problem.

Generating the path choice set is the first step to solve the lower level SUE problem by a path-based algorithm. Bekhor, Toledo, and Prashke (2008) discussed the implementation of determining the size of route set and pointed out that for real-size networks, the generation of a large number of alternative routes is needed. Chen and Alfa (1991) and Huang (1995) used the STOCH method to generate the path set. Bekhor and Toledo (2005) realised a combination of the link elimination method and the \( k \)-shortest path method. The link elimination method which consists of successively removing links and finding the shortest path on the remaining links of the network, can overcome a high degree of similarity of the routes generated by the \( k \)-shortest method only. Han (2007) defined a reasonable path set in the sense of Dial (1971). In this article, a combination of the \( k \)-shortest path method and the Dial’s STOCH method is used to generate a path set (Long et al. 2010). Since the path set generated by the STOCH method may omit some reasonable paths, the \( k \)-shortest path method is used to make the path set complete to some extent.

A simple descent direction of the logit-based SUE problem at current solution point \( \mathbf{x}^{(n)} \), denoted by \( d = [d_a] \), can be obtained by solving the problem (1) for a given \( y \), as follows:

\[
d_a = \sum_{m \in M} \sum_{r \in R} \sum_{s \in S} \sum_{k \in K^r} q_{mk}^{rs} s_{ab}^{rs} - \lambda_a^{(n)}, \quad a \in A.
\] (11)

The method of successive averages (MSA) proposed by Powell and Sheffi (1982) has successfully been applied to solve various SUE problems (e.g. Sheffi 1985; Huang and Bell 1998; Han 2003). Huang and Li (2007) recently employed this method to solve a multiclass and multicriteria logit-based SUE assignment problem. Liu, He, and He (2009) proposed the method of successive weighted averages (MSWA), which is much faster than the original MSA. This solution scheme of the MSWA is adopted in this study (Appendix gives the procedure of MSWA in detail).

### 3.2. Solving the relaxed upper level location problem

Before designing the \( \text{B&B} \) method, we should relax the integer variables (i.e. replace the constraint (10) by \( 0 \leq y_a \leq 1, a \in A \)) and then employ the sensitivity analysis based Frank–Wolfe algorithm to determine the upper bound (UB) and lower bound (LB) of the upper level’s objective function (8).

The partial derivatives of Equation (5) are required for the process of sensitivity analysis of the logit-based SUE flow. It follows

\[
\frac{\partial F_a}{\partial x_b} = \delta_{ab} - \sum_{m \in M} \sum_{r \in R} \sum_{s \in S} \sum_{k \in K^r} q_{mk}^{rs} s_{ab}^{rs} \frac{\partial \rho_{mk}^{rs}}{\partial \epsilon_{ml}^{rs}} \frac{\partial \epsilon_{ml}^{rs}}{\partial x_b},
\] (12)

and

\[
\frac{\partial F_a}{\partial y_b} = - \sum_{m \in M} \sum_{r \in R} \sum_{s \in S} \sum_{k \in K^r} q_{mk}^{rs} s_{ab}^{rs} \sum_{l \in K^r} \frac{\partial \rho_{mk}^{rs}}{\partial \epsilon_{ml}^{rs}} \frac{\partial \epsilon_{ml}^{rs}}{\partial y_b},
\] (13)

where \( \delta_{ab} = 1 \) if links \( a = b \), and \( \delta_{ab} = 0 \) otherwise.

According to the definition of route travel cost in Equation (6), we have

\[
\frac{\partial \epsilon_{mk}^{rs}}{\partial x_b} = \sum_{a} q_{mk}^{rs} \frac{\partial \tau_a(x)}{\partial x_b},
\] (14)

and

\[
\frac{\partial \epsilon_{mk}^{rs}}{\partial y_b} = \beta_{mk} q_{mk}^{rs} s_{ab},
\] (15)
From Equation (1), we have
\[
\frac{\partial p_{mk}^{rs}}{\partial c_{ml}^{rs}} = \begin{cases} 
\theta \exp\left(-\theta c_{mk}^{rs} + c_{mk}^{rs}\right) / \left(\sum_k \exp\left(-\theta c_{mk}^{rs}\right)\right), & \text{if } l \neq k, \\
-\theta \exp\left(-\theta c_{mk}^{rs}\right) + \theta \left(\sum_k \exp\left(-\theta c_{mk}^{rs}\right)\right)^2, & \text{if } l = k.
\end{cases}
\]

Equation (16) can be simplified as
\[
\frac{\partial p_{mk}^{rs}}{\partial c_{ml}^{rs}} = \begin{cases} 
\theta p_{mk}^{rs} \frac{\partial p_{mk}^{rs}}{\partial c_{ml}^{rs}}, & \text{if } l \neq k, \\
-\theta p_{mk}^{rs} (1 - p_{mk}^{rs}), & \text{if } l = k.
\end{cases}
\]

Substituting Equation (17) into Equations (12) and (13), we obtain
\[
\frac{\partial F_a}{\partial x_h} = \delta_{ab} - \sum_{m \in M} \sum_{r \in R} \sum_{s \in S} \sum_{k \in K^s} q_{mk}^{rs} p_{mk}^{rs} \delta_{mk}^{rs} \times \left(\frac{\partial c_{mk}^{rs}}{\partial x_h} - \sum_{l \in K^s} p_{ml}^{rs} \frac{\partial c_{ml}^{rs}}{\partial x_h}\right),
\]

and
\[
\frac{\partial F_a}{\partial y_b} = \theta \sum_{m \in M} \sum_{r \in R} \sum_{s \in S} \sum_{k \in K^s} q_{mk}^{rs} p_{mk}^{rs} \delta_{mk}^{rs} \times \left(\frac{\partial c_{mk}^{rs}}{\partial y_b} - \sum_{l \in K^s} p_{ml}^{rs} \frac{\partial c_{ml}^{rs}}{\partial y_b}\right).
\]

Equations (18) and (19) form the matrices \( M(y) \) and \( N(y) \), respectively. The partial derivative of \( x(y) \) with respect to \( y \) can be calculated using the method proposed in Yang and Chen (2009) and Long et al. (2010)
\[
\nabla_y x(y) = -M(y)^{-1} N(y).
\]

The relaxed upper level problem has a nonlinear objective function with linear constraints. It can be solved by the Frank–Wolfe algorithm efficiently. The gradient of the objective function (8) at \( y^{(n)} \) is
\[
\left. \frac{\partial G}{\partial y_b} \right|_{y=y^{(n)}} = \sum_{a \in A} \left( \frac{\partial x_a}{\partial y_b} \right)_{y=y^{(n)}} y_a^{(n)} + \lambda^{(n)} \delta_{ab},
\]

where the partial derivatives on the right hand side are given by Equation (20). The ascending direction can be obtained by solving the following linear optimisation problem:
\[
\max_{y \in \Omega} Z = [\nabla_y G]^{T} (y - y^{(n)}),
\]

where \( \Omega \) is the feasible domain of the relaxed upper level problem, given by
\[
\Omega = \left\{ y | 0 \leq y_a \leq 1, \quad \forall a \in A \right\}.
\]

Let \( y \) be the solution of the above linear programming problem (22), the ascending direction is \( w = y - y^{(n)} \). The step-by-step procedure of the sensitivity analysis based Frank–Wolfe algorithm for solving the relaxed upper level problem is given as follows:

**Step 0:** Initialisation. Select an initial vector of the camera installation decision variables \( y^{(0)} \) in the feasible domain \( \Omega \). Set the iteration counter \( n = 0 \) and the convergence tolerance \( \sigma > 0 \).

**Step 1:** SUE assignment. For a given \( y^{(n)} \), use the MSWA to solve the lower level problem and get the link flow \( x^{(n)} \).

**Step 2:** Sensitivity analysis. Compute \( \nabla_y x^{(n)} \) by Equation (20).

**Step 3:** Computing ascending direction. Compute the objective function’s gradient by Equation (21), and then get the ascending direction \( w^{(n)} \) by solving the problem (22).

**Step 4:** Optimising step size. Solve the one-dimensional search problem \( \max G(1 - \xi) x^{(n)} + \xi w^{(n)} \) and obtain the optimal step size \( \xi^{(n)} \).

**Step 5:** Updating decision variables. Generate a new vector, \( y^{(n+1)} = (1 - \xi^{(n)}) y^{(n)} + \xi^{(n)} w^{(n)} \).

**Step 6:** Checking convergence. If \( \max |y_a^{(n+1)} - y_a^{(n)}| \leq \sigma \) holds, stop; otherwise let \( n = n + 1 \) and go to Step 1.

### 3.3. B&B method for the bi-level problem

It is well known that the mixed integer bi-level programming problem is a NP-hard discrete optimisation problem. A satisfactory method for solving this problem should achieve a balance between solution quality and time consumption. The original B&B method can definitely obtain the optimal solution, but it is time-consuming. Thus, the original B&B method should be improved according to specific solution environment. The basic framework of the B&B method will not be described here again for saving space but some modifications for improving the method. In this study, we apply the approximate integer solution method to determine a better LB so as to prune off some redundant branches. In the branch strategy, we design an approximate regulation strategy to reduce the number of nodes in the search tree.
Note that the optimal solutions of decision variables by the sensitivity analysis based Frank–Wolfe algorithm may not be integers. Here, an approximate method is introduced to get the nearly optimal integer solutions:

$$\tilde{y}_a^* \in \begin{cases} 1, & \text{if } y_a^* \in \text{top}_{N_1}(y^*), \\ 0, & \text{otherwise}, \end{cases}$$  \hspace{1cm} (24)$$

where $\tilde{y}_a^*$ is the nearly optimal integer solution, $\text{top}_{N_1}(y^*)$ defines a set containing elements with values from the largest to the $N_1$-largest in vector $y^*$ and $|\text{top}_{N_1}(y^*)| = N_1$ holds.

We now discuss the bound strategy. The LB and the UB of the upper level problem can be obtained by

$$LB = \max_{v \in UN \cup BN} \{LB_v\},$$  \hspace{1cm} (25)$$

and

$$UB = \max_{v \in UN} \{UB_v\},$$  \hspace{1cm} (26)$$

where $v$ is a node in the search tree, UN and BN the sets of unbranched and branched nodes in the search tree, respectively. $LB_v$ and $UB_v$ the lower bound and upper bound of the optimal value of upper level objective function for node $v$. They are equal to the objective function value with respect to the nearly optimal integer solution of this node and the optimal objective function value of the relaxed upper level problem, respectively.

In our B&B method, an unbranched node but having the maximal UB is chosen as a branch node. So, the index of the branch node can be obtained as follows:

$$\mu = \arg \max_{v \in UN} \{UB_v\}. \hspace{1cm} (27)$$

If the solution of the relaxed upper level problem at branch node $\mu$ is non-integer, a widely used method is to branch non-integer elements of the solution directly, yet branches may grow up to a large number when the number of decision variables increases. In order to reduce the number of branches and speed up the calculation, we fix camera installation variables using an approximate regulation strategy, while constraint (9) is exerted. The so-called approximate regulation strategy is

$$\tilde{y}_a^* = \begin{cases} 1, & \text{if } y_a^* > 0.5 + \omega \\ 0, & \text{if } y_a^* < 0.5 - \omega \end{cases} \hspace{1cm} (28)$$

where $\tilde{y}_a^*$ is the nearly optimal solution after implementing the approximate regulation strategy, $\omega$ a critical value satisfying $0 < \omega < 0.5$. Once the elements are regulated by this strategy, the links corresponding to them are definitely installed with cameras or not.

For links satisfying $0.5 - \omega \leq y_a^* \leq 0.5 + \omega$, we choose the following element for further branching:

$$\tilde{y}_a^* = \min_{a \in A} \{|y_a^* - 0.5|\}. \hspace{1cm} (29)$$

We adopt an example to show the newly defined variables. The number of links in the network is six and the value of $\omega$ is 0.3. Let the optimal solution of an unbranched node $v$ is $y^* = [1, 0.5, 0.9, 0.8, 0.4, 0.1]$ and the total number of cameras to be installed is $N_1 = 3$. According to Equation (24), we can obtain the nearly optimal integer solution $\tilde{y}_v^* = [1, 0, 1, 1, 0, 0]$, which is used to estimate the LB of the upper level objective function for node $v$. If node $v$ is selected as the branch node, we fix camera installation vector $\tilde{y}_v^* = [1, 0.5, 1, 1, 0.4, 0]$ using Equation (28). Hence Links 2 and 5 remain undetermined. According to Equation (29), the second element will be selected for further branching.

**Remark 1:** In principle, the B&B method is terminated if one of the following conditions is satisfied: (i) LB $\geq$ UB; (ii) there are no nodes to be branched; (iii) the computational time exceeds a reasonable limit. Our experience shows that condition (i) is always satisfied.

**Remark 2:** The critical value in approximate regulation strategy is a key parameter since it can reduce not only the computational time, but also improve the nearly optimal solution. Besides the critical value, the properties of functions in the relaxed upper level problem also influence the solution quality. Though it is difficult to check the convexity of the functions, the Frank–Wolfe algorithm can converge to a local optimal solution at least. In addition, the solution quality by the B&B method can be evaluated by comparing the solution with that by a complete enumeration method when the number of combinations is not large.

4. Numerical examples

In this section, we present numerical results to illustrate the model application and show the effectiveness of the solution method. The numerical results are from the Sioux Falls test network, as shown in Figure 1. This network includes 24 nodes, 76 links and 528 origin-destination (OD) pairs. The free-flow link travel times, link capacities and OD demands can be found in Leblanc, Morlok, and Pierskalla (1975). All solution algorithms are coded by Matlab and implemented on a personal computer with 3.20 GHz CPU and 2 GB memory. In the preliminary stage, 1942 paths are generated. The average number of paths per OD pair is 3.7 and the maximal number of paths between an OD pair is 13.
methods are denoted by $G^p$ and $G^q$ (GA), respectively; and the CPU times required by these two methods are denoted by $T^p$ and $T^q$, respectively. The outputs from 30 instances, which corresponds to some combinations of $N_1 = (3, 8, 20)$, $\theta = (0.5, 1.0)$, $\beta_1 = (0.1, 0.5)$ and $\alpha = (0.3, 0.5, 0.7, 1.0)$, are summarised in Table 1.

In order to identify whether the solution by the proposed B&B method is optimal, we implemented the enumeration method to find the optimal solutions for the instances 1–20 in Table 1. Except instance 10, the global optimal solutions were indeed obtained by the B&B method for other 19 instances. We can observe from Table 1 that the values of the upper level objective function of the B&B method are not less than that of the GA method. This implies the B&B method can obtain solutions with higher quality. We can also observe from Table 1 that the B&B method outperforms the GA method for most of the tested instances in terms of both CPU time and solution quality. Therefore, the proposed B&B method can generate satisfactory and qualified camera location solutions within reasonable CPU times, though it is hard to guarantee the method can always lead to global optimum. As special cases in instances 15 and 20, the CPU time of GA method is less than the B&B method. This is because the B&B method may need lots of CPU time due to a large number of branches.

Table 2 shows the impacts of travel demand on camera location when other parameters are valued by $N_1 = 8$, $\theta = 0.5$, $\alpha = 0.5$, $\beta_1 = 0.5$ and $\beta_2 = 0.1$. Here, all base OD demands are multiplied by an adjustment factor for reflecting the changes of travel demands. We let the factor increase from 0.5 to 1.5 with an increment of 0.01 and check the solutions. Table 2 illustrates that the optimal camera location is somewhat sensitive to the demand adjustment factor, especially when the factor is between 0.9 and 1.1.

Figure 2 shows the total flows monitored by the cameras which are installed at the optimal locations when the demand adjustment factor increases from 0.5 to 1.5. The base camera location is the solution solved when the adjustment factor equals 1. Comparing the two curves, we can see that different camera locations bring small differences on the monitored flows, particularly when the factor is between 0.8 and 1.2. Thus, the camera location, once determined, is unnecessarily updated when travel demand slightly fluctuates. It is worth noting that the results may not be robust when a non-uniform increase is applied to each OD pair.

Figure 3 illustrates how the demand proportion and additional cost of class 1 affect the total monitored flow. First, if $\beta_1 = \beta_2 = 0$, the monitored flow is not affected by the demand composition. Second, the monitored flow when $\beta_1 = \beta_2 = 0.1$ is less than that
when $\beta_1 = \beta_2 = 0$. This means the consideration of additional cost caused by ADCM leads to a decrease in the flow monitored by cameras. Third, when $\beta_2$ is fixed at 0.1, a larger $\beta_1$ results in a larger decrease of the monitored flow. This is because when the users of class 1 are more willing to choose the routes without or with fewer cameras, the monitored flow naturally decreases. Fourth, the monitored flow decreases with the demand proportion of class 1. Thus, the more travellers who are averse to cameras and the less the monitored flow.

Table 1. Outputs of 30 instances on the Sioux Falls test network.

<table>
<thead>
<tr>
<th>Instances</th>
<th>$N_1$</th>
<th>$\theta$</th>
<th>$\beta_1$</th>
<th>$\alpha$</th>
<th>$G^*$</th>
<th>$G^b$</th>
<th>$G^\dagger$</th>
<th>$T^b$ (s)</th>
<th>$T^\dagger$ (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>0.5</td>
<td>0.5</td>
<td>1</td>
<td>79,494</td>
<td>79,494</td>
<td>79,494</td>
<td>24</td>
<td>296</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0.5</td>
<td>0.5</td>
<td>0.3</td>
<td>77,770</td>
<td>77,770</td>
<td>77,670</td>
<td>24</td>
<td>202</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>76,610</td>
<td>76,610</td>
<td>76,610</td>
<td>24</td>
<td>269</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0.5</td>
<td>0.5</td>
<td>0.7</td>
<td>75,440</td>
<td>75,440</td>
<td>75,440</td>
<td>23</td>
<td>207</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>0.5</td>
<td>0.5</td>
<td>1</td>
<td>73,670</td>
<td>73,670</td>
<td>71,632</td>
<td>30</td>
<td>252</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>1</td>
<td>0.1</td>
<td>1</td>
<td>78,642</td>
<td>78,642</td>
<td>78,600</td>
<td>45</td>
<td>628</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1</td>
<td>0.5</td>
<td>0.3</td>
<td>75,861</td>
<td>75,861</td>
<td>75,861</td>
<td>36</td>
<td>640</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>1</td>
<td>0.5</td>
<td>0.5</td>
<td>73,955</td>
<td>73,955</td>
<td>73,955</td>
<td>37</td>
<td>417</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>1</td>
<td>0.5</td>
<td>0.7</td>
<td>72,009</td>
<td>72,009</td>
<td>72,009</td>
<td>104</td>
<td>564</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>1</td>
<td>0.5</td>
<td>1</td>
<td>69,121</td>
<td>69,020</td>
<td>69,020</td>
<td>81</td>
<td>410</td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td>0.5</td>
<td>0.1</td>
<td>1</td>
<td>195,772</td>
<td>195,772</td>
<td>194,313</td>
<td>19</td>
<td>599</td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td>0.5</td>
<td>0.5</td>
<td>0.3</td>
<td>191,312</td>
<td>191,312</td>
<td>182,987</td>
<td>89</td>
<td>526</td>
</tr>
<tr>
<td>13</td>
<td>8</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>188,302</td>
<td>188,302</td>
<td>185,060</td>
<td>246</td>
<td>566</td>
</tr>
<tr>
<td>14</td>
<td>8</td>
<td>0.5</td>
<td>0.5</td>
<td>0.7</td>
<td>185,264</td>
<td>185,264</td>
<td>181,113</td>
<td>568</td>
<td>853</td>
</tr>
<tr>
<td>15</td>
<td>8</td>
<td>0.5</td>
<td>0.5</td>
<td>1</td>
<td>181,549</td>
<td>181,549</td>
<td>180,878</td>
<td>1009</td>
<td>599</td>
</tr>
<tr>
<td>16</td>
<td>8</td>
<td>1</td>
<td>0.1</td>
<td>1</td>
<td>192,710</td>
<td>192,710</td>
<td>192,710</td>
<td>115</td>
<td>1183</td>
</tr>
<tr>
<td>17</td>
<td>8</td>
<td>1</td>
<td>0.5</td>
<td>0.3</td>
<td>185,671</td>
<td>185,671</td>
<td>185,671</td>
<td>335</td>
<td>1251</td>
</tr>
<tr>
<td>18</td>
<td>8</td>
<td>1</td>
<td>0.5</td>
<td>0.5</td>
<td>181,532</td>
<td>181,532</td>
<td>180,923</td>
<td>733</td>
<td>944</td>
</tr>
<tr>
<td>19</td>
<td>8</td>
<td>1</td>
<td>0.5</td>
<td>0.7</td>
<td>177,648</td>
<td>177,648</td>
<td>177,345</td>
<td>947</td>
<td>1225</td>
</tr>
<tr>
<td>20</td>
<td>8</td>
<td>1</td>
<td>0.5</td>
<td>1</td>
<td>171,883</td>
<td>171,883</td>
<td>171,437</td>
<td>1299</td>
<td>1274</td>
</tr>
<tr>
<td>21</td>
<td>20</td>
<td>0.5</td>
<td>0.1</td>
<td>1</td>
<td>–</td>
<td>444,219</td>
<td>443,462</td>
<td>40</td>
<td>1240</td>
</tr>
<tr>
<td>22</td>
<td>20</td>
<td>0.5</td>
<td>0.5</td>
<td>0.3</td>
<td>–</td>
<td>436,161</td>
<td>436,142</td>
<td>51</td>
<td>1307</td>
</tr>
<tr>
<td>23</td>
<td>20</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>–</td>
<td>431,052</td>
<td>431,052</td>
<td>34</td>
<td>1589</td>
</tr>
<tr>
<td>24</td>
<td>20</td>
<td>0.5</td>
<td>0.5</td>
<td>0.7</td>
<td>–</td>
<td>425,900</td>
<td>425,900</td>
<td>40</td>
<td>1137</td>
</tr>
<tr>
<td>25</td>
<td>20</td>
<td>0.5</td>
<td>0.5</td>
<td>1</td>
<td>–</td>
<td>418,049</td>
<td>418,049</td>
<td>68</td>
<td>1801</td>
</tr>
<tr>
<td>26</td>
<td>20</td>
<td>1</td>
<td>0.1</td>
<td>1</td>
<td>–</td>
<td>431,443</td>
<td>429,807</td>
<td>91</td>
<td>2843</td>
</tr>
<tr>
<td>27</td>
<td>20</td>
<td>1</td>
<td>0.5</td>
<td>0.3</td>
<td>–</td>
<td>417,516</td>
<td>417,516</td>
<td>76</td>
<td>3009</td>
</tr>
<tr>
<td>28</td>
<td>20</td>
<td>1</td>
<td>0.5</td>
<td>0.5</td>
<td>–</td>
<td>408,350</td>
<td>405,598</td>
<td>401</td>
<td>2612</td>
</tr>
<tr>
<td>29</td>
<td>20</td>
<td>1</td>
<td>0.5</td>
<td>0.7</td>
<td>–</td>
<td>399,817</td>
<td>399,431</td>
<td>234</td>
<td>2402</td>
</tr>
<tr>
<td>30</td>
<td>20</td>
<td>1</td>
<td>0.5</td>
<td>1</td>
<td>–</td>
<td>387,447</td>
<td>387,074</td>
<td>502</td>
<td>2413</td>
</tr>
</tbody>
</table>

Note: $G^*$ represents the optimal value of the upper level objective function generated by the enumeration method.

Table 2. Optimal solutions against demand adjustment factor.

<table>
<thead>
<tr>
<th>Adjustment factor</th>
<th>Optimal camera location $y^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.50–0.84</td>
<td>111100010100100100000000</td>
</tr>
<tr>
<td>0.85–0.91</td>
<td>111101010101001001000000</td>
</tr>
<tr>
<td>0.92–0.97</td>
<td>111111010100000000000000</td>
</tr>
<tr>
<td>1.03–1.26</td>
<td>111111110000000000000000</td>
</tr>
</tbody>
</table>

Notes: These elements in $y^*$ corresponds to links 9, 11, 25, 26, 27, 28, 32, 43, 45, 46, 49, 50, 52, 53, 55, 56, 57, 58, 60 and 67 in Figure 1, respectively. Elements not listed in the table take zero.

Figure 2. Effects of travel demand on the monitored flow.
Table 3 shows the impact of the class 1’s demand proportion on the optimal camera location, when other parameters are valued by $N_1 = 8$, $\theta = 0.5$, $\beta_1 = 0.5$ and $\beta_2 = 0.1$. We can see that the optimal camera location is rather robust to the proportion. Thus, there is no need to frequently change the camera location although the demand composition has changed a lot. However, a critical value of the demand composition which results in a new optimal solution should be noted. In Table 3, the critical value is $\alpha = 0.8$.

Finally, we depict two optimal camera locations in Figure 4, as examples for direct illustration. These two solutions correspond to two extreme cases of the class 1’s demand proportion, namely $\alpha = 0$ and $\alpha = 1$, respectively. When $\alpha = 0$, all travellers belong to class 2 and have identical $\beta$-value 0.1; when $\alpha = 1$, the $\beta$-value of all travellers is 0.5. It can be seen that all cameras are located on links having heavy traffic regardless of the $\alpha$-value. But, the location details are somewhat different in the two cases because, all in all, drivers with different $\beta$-values have different criteria for route choice. In all later 20 instances in Table 1, the four links 28, 43, 27 and 32 are always selected for camera installation. This is because they are the first four links, concerning the link flow assigned.

5. Conclusions

This article formulates a novel camera location problem with multiple user classes in the context of traffic surveillance. All users are differentiated by their own acceptance degree of camera monitoring and make their route choices in a logit-based SUE manner. A bilevel programming model is proposed in which the upper level aims at maximising the total monitored traffic flow for traffic surveillance. A sensitivity analysis based B&B method is developed for solving the
bi-level programming model and numerically compared with the GA on an example network. We found that the B&B method has good performance for most of the tested instances. The GA can yield satisfactory solutions but is time-consuming. The experimental results clearly show that the total monitored flow is sensitive to the travellers’ additional costs reflecting their ADCM, the demand composition and the budget limitation for camera installation. The camera location solution is somewhat robust to the demand level and composition.

This article, through considering the cost-associated parameter reflecting acceptance degree of camera monitoring, aims at improving the insights into evading behaviour against camera monitoring and the corresponding optimal camera location strategy. However, the model parameters should be calibrated through questionnaire to drivers in the next empirical study.

The work can be further extended along four lines. First, it is essential and realistic to use a continuously distributed ADCM across all travellers to replace the limited groups of users adopted in this article. Second, it is of interest and meaningful to use the data collected by cameras to estimate OD trip (Marzano, Papola, and Simonelli 2009). In this regard, the difficulty is how to identify the influence caused by the drivers’ evading behaviour from camera monitoring on real OD demand. Third, the problem could be extended with other ITS strategies. Indeed, it is possible to use location data of bus or taxi fleets for monitoring road vehicle flows via Automatic Vehicle Location (AVL) system (D’Acierno, Carteni, and Montella 2009). Therefore, the bi-level model proposed in this article could be adopted for validating or integrating with AVL data.

Acknowledgements
The authors would like to thank the anonymous referees for their useful comments on this article. The research described in this article was jointly supported by grants from the National Basic Research Program of China (2012CB725401), the National Natural Science Foundation of China (71001001) and the Innovation Funds for Doctorial Candidates of Beihang University (300528).

Notes on contributors

Xiao-Shan Lu received her BS and MS in Operations Research from Beijing University of Chemical Technology. She is currently working toward the PhD in transportation engineering from the School of Economics and Management, Beijing University of Aeronautics and Astronautics (Beihang University). Her research interests include traffic modelling, intelligent transportation systems and operations research.

Hai-Jun Huang received his PhD in Systems Engineering from the School of Economics and Management, Beihang University (1992). He is currently the Cheng Kong professor in Beihang. His research interests include road traffic flow models, transport network modelling, travel behaviour analysis and congested road-use pricing. He has published more than 100 papers in international journals. He co-authored the book Mathematical and Economic Theory of Road Pricing (Elsevier, 2005). He is now on the editorial boards of Transportation Research Part B, Journal of Advanced Transportation, Transportmetrica and others.

Jiancheng Long received his PhD in Traffic Engineering from the School of Traffic and Transport, Beijing Jiaotong University (2009). He then worked for 2 years a postdoctor in Beihang before joining Hefei University of Technology. His research interests include traffic modelling, traffic simulation and intelligent transportation systems.

References

Appendix: MSWA procedure

The solution scheme of the MSWA is outlined below:

**Step 0:** Initialisation. Use Equation (1) to calculate the route choice probability on the basis of free-flow travel cost, get the initial link flow \( x^{(1)} \) by Equations (3) and (4), and set two weight parameters \( \eta > 1 \) and \( 0 < \gamma < 1 \). Let \( \rho^{(0)} = 1 \), the iteration index \( n = 1 \) and the convergence tolerance \( \varepsilon > 0 \).

**Step 1:** Updating path travel cost. According to the current link flows \( x^{(n)} \), update link travel time \( t^{(n)} = [\ell^{(n)}_{mk}] \) and path travel cost \( c^{(n)}_{mk} \) by Equations (7) and (6), respectively.

**Step 2:** Updating descent direction. Compute the descent direction \( d^{(n)} \) by Equation (11).

**Step 3:** Step size determination. Set the step size, \( \lambda^{(n)} = 1/\rho^{(n)} \), where
\[
\rho^{(n)} = \begin{cases} 
\rho^{(n-1)} + \eta, & \text{if } \|d^{(n)}\| \geq \|d^{(n-1)}\| \\
\rho^{(n-1)} + \gamma, & \text{otherwise}
\end{cases}
\]

**Step 4:** Updating link flow. Let \( x^{(n+1)} = x^{(n)} + \lambda^{(n)}d^{(n)} \).

**Step 5:** Checking convergence. If \( \|d^{(n)}\| \leq \varepsilon \) holds, stop; otherwise, let \( n = n + 1 \) and go to Step 1.