A Bi-level Model to Optimize the Tolling Level and Period on a Bottleneck Road with Multi-class Users

Hua-ling REN, Rong-hui ZHAO, and Dong WANG

MOE Key Laboratory for Urban Transportation Complex Systems Theory and Technology, Beijing Jiaotong University, Beijing, 100044

Keywords: Bi-level programming, Step-tolling; Multi-user; Bottleneck model.

Abstract. This paper analyzes the multi-user step-tolling problem on a bottleneck road, and optimizes the tolling level and tolling period using a bi-level model. Firstly, based on the Greenshields model, the multi-user travel behaviors are simulated and the flow propagation process is obtained on the bottleneck road. The queuing length and the speed of the moving part can be calculated at the stable state. Based on these, a bi-level programming model is established to optimize the tolling level and tolling period. The bi-level model is solved using an improved genetic algorithm (GA). Finally, a simple example is given to illustrate the application of the model and its algorithm.

Introduction

Nowadays, due to rapidly economic development, the rapidly growth in car ownership in most cities in the world has made them experience serious traffic congestion. In 1975, Singapore firstly implemented congestion pricing policy. However, due to technical limits, it was only a manual pricing. Then in 1998, Singapore became the first city that implemented electronic fee collection. In 2003, London applied the congestion pricing policy, and the pricing areas were mainly concentrated in the urban center. In 2006, Stockholm began to implement the congestion pricing policy, and after trial operation of six months, it was proved having eliminated the traffic congestion [1].

In our country, along with the rapid growth of car ownership, Beijing, Shanghai, Chengdu, and other cities have suffered different degrees of traffic congestion [2]. Means of limiting the number of travelling and buying cars are used in Beijing to control the traffic congestion. Congestion pricing methods are also discussed to be used to control the traffic congestion.

In most dynamic traffic congestion models, congestion tolls vary continuously over time to achieve an absolutely social optimum. Yang [3] mathematically analyzed the dynamic pricing problem based on a bottleneck model, for both single-class users case and multi-class users case. Yet no existing road pricing scheme has such sophisticated, continuously time-varying tolls [4].

All the above single-step-toll models are analytical and are based on the assumption that the vehicles have no lengths, which is unpractical and difficult to be used in pricing on the networks. On the other hand, few references consider multi-class users in single-step-toll models. This paper will establish a multi-class step-tolling model based on the Greenshields model [5], and then optimize the tolling level and tolling period by a bi-level model, minimizing the maximal queue length and the minimal speed.

The rest of the paper is organized as follows: In Section 2, we derive a multi-class step-tolling model based on Greenshields model. The tolling level and the tolling period are optimized using a bi-level programming model in section 3. In Section 4, a numerical example is given. Finally, our summary and conclusions are provided in Section 5.

A Multi-class Step-tolling Model Based on Greenshields Model

Symbols in the Model

In the bottleneck model there is a continuum of D identical commuters who travel alone by car (one traveler per car) from a common origin to a common destination connected by a single road which
Multi-class Greenshields Model for the Bottleneck Road

Consider a certain period of time $[0, T]$ and it is discretely divided in to $N$ equal time intervals for the convenience of simulation and each time interval is $[(t - 1)T / N, tT / N]$ and $t \in \{1, 2, \ldots, N\}$. The working time is set as $[w^l, w^r]$. Commuters arriving at the bottleneck before $w^l$ will have early penalty, and arriving at the bottleneck after $w^r$ will have late penalty. The commuters are divided into two classes: class 1, who have high value of time and low value of penalty time (early or late); class 2, who have low value of time and high value of penalty time. To control the serious traffic congestion, a step toll is collected during a certain period of time $[t^+, t^-]$. The dynamic link travel time is divided into two parts: moving time and queuing delay time. When the vehicles depart on the link at time $t$, the number of vehicles on the link at this time is decided by the inflow rate at $t - 1$, the capacity of the bottleneck and the number of vehicles at $t - 1$ [6]:

$$X(t) = X(t - 1) + u(t - 1) - v. \quad (1)$$

Similarly, the number of queuing vehicles at this time is decided by the queuing rates at $t - 1$, the capacity of the bottleneck and the number of queuing vehicles at $t - 1$:

$$X^2(t) = X^2(t - 1) + r(t - 1) \cdot k^1(t - 1) - v. \quad (2)$$

The the queue length at time $t$ can be calculated:

$$L^2(t) = \frac{X^2(t)}{k^1}, \quad \forall t. \quad (3)$$

According to the above equations, we can get the density of the moving part at time $t$:

$$k^1(t) = \frac{X(t) - X^2(t)}{L(t) - \frac{X^2(t)}{k^1}}, \quad \forall t. \quad (4)$$

Table 1. The meaning of symbols.

| $L$ : | the length of the link; |
| $u(t)$ : | the inflow rate at time $t$; |
| $u_1(t)$ : | the inflow rate of user class $w$ at time $t$; |
| $u_2(t)$ : | the inflow rate of user class $b$ at time $t$; |
| $v$ : | the outflow rate capacity of bottleneck; |
| $X(t)$ : | the number of vehicles on the link at time $t$; |
| $X^2(t)$ : | the number of vehicles on the queuing section at time $t$; |
| $r(t)$ : | the speed of moving vehicles on the link at time $t$; |
| $r_{\text{max}}$ : | the upper speed limit on the link; |
| $r_{\text{min}}$ : | the lower speed limit on the link; |
| $k_j$ : | the jam density on the link; |
| $C(t)$ : | the travel cost at time $t$; |
| $C_1(t)$ : | the travel cost of user class $w$ at time $t$; |
| $C_2(t)$ : | the travel cost of user class $b$ at time $t$; |
| $C_{\text{max}}$ : | the minimal travel cost; |
| $C_{\text{roll}}$ : | the toll level on the link; |
| $T(t)$ : | the travel time of vehicles entering the link at time $t$; |
| $r^1(t)$ : | the travel time of the moving vehicles on the link at time $t$; |
| $r^2(t)$ : | the delaying time of queuing vehicles at time $t$; |
| $[t^+, t^-]$ : | the tolling period; |
| $[w^l, w^r]$ : | the time window of working; |

is subject to bottleneck queuing congestion. Without a queue, an individual thus departs “from home”, passes the bottleneck, and completes his/her trip by arriving “at work” all at the same moment. The symbols are as follows:
So, using the Greenshields speed-density relationship, the speed of moving part at time $t$ can be described by the density of the moving part, that is,

$$r(t) = r_{\text{max}} + (r_{\text{max}} - r_{\text{min}})\left[1 - \left(\frac{k_j(t)}{k_j}\right)^\alpha\right]^\beta, \quad \forall t.$$  

We have

$$t^1(t) = \left[\frac{L - X^2(t)}{k_j}\right]/r(t), \quad \forall t,$$  

$$t^2(t) = \left[\frac{X^2(t) + [r(t) \cdot k^1(t) - v \cdot t^1(t)]}{v}\right], \quad \forall t.$$  

Then travel time can be computed as follows:

$$T(t) = t^1(t) + t^2(t), \quad \forall t.$$  

The travel costs of the two classes at time $t$ are different and can be obtained as (9) and (10), respectively [7, 8, 9]:

$$C_1(t) =\begin{cases} \alpha_1 T(t) + \beta_1[w^l - t - T(t)] & t + T(t) < w^l \\ \alpha_1 T(t) + \gamma_1[T(t) - w^l] & t + T(t) > w^l \end{cases} + \begin{cases} C_{\text{toll}} & t \in [t^+, t^-] \\ 0 & t \notin [t^+, t^-] \end{cases},$$  

$$C_2(t) =\begin{cases} \alpha_2 T(t) + \beta_2[w^l - t - T(t)] & t + T(t) < w^l \\ \alpha_2 T(t) + \gamma_2[T(t) - w^l] & t + T(t) > w^l \end{cases} + \begin{cases} C_{\text{toll}} & t \in [t^+, t^-] \\ 0 & t \notin [t^+, t^-] \end{cases},$$

where $\alpha_i (\beta_i)$ is the unit cost value of travel time of user class 1 (user class 2), $\beta_i (\beta_i)$ is the unit cost value of early penalty of user class 1 (user class 2), and $\gamma_i (\beta_i)$ is the unit cost value of late penalty of user class 1 (user class 2). Empirically, we have $\alpha_1 / \beta_1 > \alpha_2 / \beta_2 ; \gamma_1 / \beta_1 = \gamma_2 / \beta_2 = \eta$ [3].

Algorithm for the Multi-class Model

The commuters of these two classes choose their departure time so that the following user equilibrium (UE) state is satisfied: at any chosen departure time, the commuters have the same minimal travel cost, while at any other time, the travel cost are no less than the minimal travel cost. In order to get the UE state, Algorithm A is proposed.

Step 1: Set OD demand as $D$. The OD demand of user class 1 is $D_1 = 2/5 D$.

Step 2: Set the tolling level $C_{\text{toll}}$ and the tolling period $[t^+, t^-]$.

Step 3: Set the initial inflow rate $u_1(t)^{(1)}$ and $u_2(t)^{(1)}$. The iterative number is $n = 1$.

Step 4: Calculate the number of moving vehicles and queuing vehicles by Eqs. (1) and (2).

Step 5: According to Eq. (8), the travel time can be calculated.

Step 6: According to Eqs. (9) and (10), the travel costs of the two classes of commuters at each time can be calculated. The minimal travel cost $C_{\text{min}}^{1,i,n}$ and $C_{\text{min}}^{2,i,n}$ can be found.

Step 7: The inflow rates for each time can be updated as follows,

If $C_i(t)^{(n)} \neq C_{\text{min}}^{1,i,n}$, set

$$u_i(t)^{(n+1)} = u_i(t)^{(n)} - \rho \cdot u_i(t)^{(n)} \cdot [C_i(t)^{(n)} - C_{\text{min}}^{1,i,n}], i = 1, 2;$$

If $C_i(t)^{(n)} = C_{\text{min}}^{1,i,n}$, set
\[ u_i(t)^{(n+1)} = u_i(t)^{(n)} + \sum_{i=1}^{N} \rho \cdot u_i(t)^{(n)} \cdot [C_i(t)^{(n)} - C_{i,n, \min}]/C_{i,n, \min}, \quad i = 1, 2, \]

where \( C_{i,n, \min} \) is the number of time intervals at which the travel cost equals the minimal travel cost.

**Step 7:** If \( \sum_{i=1}^{N} u_i(t)^{(n)} \cdot [C_i(t)^{(n)} - C_{i,n, \min}]/C_{i,n, \min} \leq \epsilon \), stop; otherwise, set \( n = n + 1 \), and return to Step 4 [6, 7].

**Numerical Example of the Multi-class Model**

In this sub-section, a simple numerical example is given to show the application. The link length is set as 5.25 km, and the traffic demand is 625 vehicles. The free-flow speed and the minimum velocity are set as 0.7 km/min and 0.3 km/min. The crowded density and critical density equal to 160 veh/km and 56 veh/km. The traffic capacity at the downstream outlet of the bottleneck section is 15 veh/min by the calculation. The work starting time window is \([w', w'] = [75, 85]\) min; Other parameters are set as follows: \( \alpha_1 = 1, \beta_1 = 0.2, \gamma_1 = 2; \alpha_2 = 1, \beta_2 = 0.22, \gamma_2 = 2.2; \eta = 10; \rho = 0.0001; \ a = 1, \ b = 3. \) The tolling time is \([55, 75]\) min with 2 Yuan. Number of cycles \( n = 55000 \) and \( \epsilon = 0.0004 \).

Both class users depart not two early or too late to avoid the penalty. Traveler who departs at \( t = 53 \) min just becomes the first charged one when he/she reaches the outlet of the bottleneck. So, the inflow rate decreases rapidly at this time interval to avoid congestion charges (Figure 1), and the travel costs of the three time intervals after this time point are higher (Figure 2).

Under these values of the parameters, the early penalty of class user 2 is higher than the tolling value, so many commuters of class 2 are willing to depart after the toll starting time (Figure 1). The number of queuing vehicles increases and the speed of the moving part decreases (Figure 3 and 4).
Change some values of the parameters as: $\alpha_1 = 1$, $\beta_1 = 0.2$, $\gamma_1 = 2$; $\alpha_2 = 0.48$, $\beta_2 = 0.16$, $\gamma_2 = 1.6$. They still satisfy $\alpha_1/\beta_1 > \alpha_2/\beta_2$, $\gamma_1/\beta_1 = \gamma_2/\beta_2 = \eta$, but with $\beta_2 < \beta_1$ while $\beta_2 > \beta_1$ in the study above. The numerical results are shown in Figure 5-6.

Under the last set of values of the parameters, the commuters of class 1 are more willing to depart earlier or later, because they want to avoid the heavy congestion (from Figure 3 and Figure 5, we can see that the congestion of Figure 4 is heavier). Under the second set of values of the parameters, the commuters of class 1 choose to depart during the middle time period, because the congestion is not so heavy and these commuters are more willing to pay the toll to get the good travel condition. The result is the same with the mathematical conclusion of Yang [3].

The Bi-level Optimization Model and the Solution Algorithm

The Bi-level Optimization Model

Greenshields model with multi-class users shows that the number of queuing vehicles on the link $X^2(t)$ and the speed of the moving vehicles $r(t)$ are all the function of the tolling level and the tolling period. Using these relationships in the Greenshields model, the number of queuing vehicles can be minimized and the speed of moving vehicles can be maximized by optimizing the tolling level and the tolling period. So a bi-level programming is proposed in this section, and the upper objective is to both minimize the number of queuing vehicles and maximize the speed of moving vehicles. The multi-class Greenshields model is the lower level model to describe the travel behaviors under certain tolling level and tolling period.

The upper planning model is as follows:

$$
\begin{align*}
\min_{C_\text{toll}, t^+, t^-} & \quad Z = \theta \max X^2(t) - (1 - \theta) \delta \min r(t) \\
\text{st.} & \quad C_\text{toll}^\min \leq C_\text{toll} \leq C_\text{toll}^\max, \\
& \quad t^\min \leq t^+ \leq t^+ \max, \\
& \quad t^- \min \leq t^- \leq t^- \max. 
\end{align*}
$$

(13)

For any given $C_\text{toll}$, $t^+$, and $t^-$, we can calculate $X^2(t)$ and $r(t)$ by the Greenshields model, which will be reflected to the upper model in the iterative algorithm, and then the upper model can decide the adjust direction of the tolling level and the tolling period.

Solution Algorithm

As it is known, the solution of the bi-level programming is very difficult, because it is a NP-hard problem. In this paper, a genetic algorithm [9,10] is presented to solve the proposed bi-level programming, which has the characteristics of high applicability, simple program, and high searching efficiency. The summary of the algorithm, defined as Algorithm B, is as follows.

Step 1: Set the parameters in the genetic algorithm.

405
Step 2: Randomly generate \( m \) individuals of a population \( C_{toll}^{(0)} \), \( t^{+}(0) \) and \( t^{-}(0) \). Set the initial evolution generation number \( M=0 \).

Step 3: Define the upper objective function as the fitness function. Encode the upper decision variables and solve the lower model with braking behaviors described in subsection 2.2.

Step 4: Select and copy the next generation populations \( C_{toll}^{(M)} \), \( t^{+}(M) \) and \( t^{-}(M) \). For each population, use Algorithm A to get the UE flow state with braking behaviors.

Step 5: Replication, crossover, Mutation;

Step 6: Set \( M = M + 1 \), and return to step 3.

**Numerical Example**

This section provides a simple example to apply the presented model and its solution algorithm, and seeks the optimal solution based on the example given in subsection 2.4.

Firstly, the input parameters are the same as those in subsection 2.4. The working starting time is during \([75, 85] \) and there is no early or late arrival penalty during this period. In addition, \([C_{toll}^{\text{min}}, C_{toll}^{\text{max}}]=[1,10] \) Yuan, \([t_{\text{min}}^{+}, t_{\text{max}}^{+}]=[40,60] \) min and \([t_{\text{min}}^{-}, t_{\text{max}}^{-}]=[70,90] \) min.

Based on the above data, this section considers three kinds of charging optimization schemes:

(a) Minimizing the maximal number of queuing vehicles;
(b) Maximizing the minimal speed of the moving part;
(c) Both minimizing the maximal number of queuing vehicles and maximizing the minimal travel speed.

Figure 7 and Figure 8 depict the curve graphs of the numbers of queuing vehicles and the speeds of the moving part under these three different optimizing schemes respectively. From Figure 8, it can be found that both in scheme (b) and scheme (c), the moving speeds are relative higher, which is what we want. But, from Figure 10, the number of queuing vehicles in scheme (b) is larger than that in scheme (c). So scheme (c) is preferred.
There is a weight coefficient $\theta$ in the upper level of the bi-level programming model. The weight coefficient is to adjust the importance of the maximal number of queuing vehicles and the minimal travel speed of the moving part in the optimization. With the increase of the weight coefficient $\theta$, more importance is put on minimizing the maximal number of vehicles (Figure 9 and Figure 10).

The optimal solution is obtained as follows when the weight coefficient takes different values:

1. $\theta = 0.1$: $C_{toll} = 6.23$, $t^+ = 52.44$, $t^- = 75.22$;
2. $\theta = 0.3$: $C_{toll} = 4.3$, $t^+ = 54.09$, $t^- = 88.15$;
3. $\theta = 0.5$: $C_{toll} = 1.52$, $t^+ = 49$, $t^- = 84.93$;
4. $\theta = 0.7$: $C_{toll} = 1.82$, $t^+ = 50.27$, $t^- = 82.64$;
5. $\theta = 0.9$: $C_{toll} = 6$, $t^+ = 50.07$, $t^- = 76$.

From the above calculation results we can know that when the weight coefficient changes, the results of the optimal tolling level and the tolling period also change. When the weight coefficient $\theta = 0.5$ and $\theta = 0.7$, the optimal tolling level are relatively low, while the optimal tolling period are relatively short and the optimal travel costs of each travelers are low. It should be noted that the optimal tolling levels have no regularities with the increasing of weight coefficient, so do the optimal tolling periods. The reason is that the tolling level and the tolling period are optimized at the same time, and they will interact with each other in the optimizing process.

**Conclusions**

In this paper, a Greenshields model is applied to obtain the queue length and the speed of the moving vehicles on a bottleneck link with multi-class users, which is more realistic and simple to describe the travel behaviors in practice. Furthermore, a bi-level programming model is proposed, in which the upper level is to minimize the maximal number of queuing vehicles and the minimal speed of moving vehicles, and the lower programming is the single-step-toll model based on the Greenshields model.

We find that, in the bottleneck model, the multi-class users have different travel behaviors when step-tolling is used on the congested link. Their behaviors are also related to the values of the parameters, especially the values of the time and the penalty. Three optimized tolling schemes are considered respectively. We find it more effective to take both the number of queuing vehicles and the speed of moving vehicles into consideration in the optimal model, compared to considering only one of them. When considering both, the optimal tolling level is the lowest, the optimal tolling period is the shortest, and the travel cost is the smallest.

**Acknowledgements**

The work was supported by the National Natural Science Foundation of China (71621001), the National Natural Science Foundation of China (71371026, 71471104, 71171124).

**References**


