AN EMISSIONS-BASED USER EQUILIBRIUM MODEL AND ALGORITHM FOR LEFT-TURN PROHIBITION PLANNING

ABSTRACT

The left-turn of vehicles at intersections has significant impacts on urban traffic congestions and accidents, which have negative effect on vehicle emissions causing air pollution. Many urban traffic networks prohibit direct left-turns for transport planning to keep traffic moving efficiently on major roads. As such, this paper proposes a bi-level mathematical model for left-turn prohibition planning considering both travel times and traffic emissions. The lower-level and upper-level are respectively solved by using the Frank-Wolfe algorithm and an improved genetic algorithm. By numerical examples, this paper shows that the improved algorithm can effectively enhance the speed and accuracy of the calculation, and the traffic congestions and emissions can be alleviated by implementing the left-turn prohibition at some carefully selected intersections.

KEY WORDS

left-turn prohibition; traffic emissions; traffic assignment; user equilibrium (UE); bi-level programming;

1. INTRODUCTION

With rapid urbanization, more private cars enter the limited urban traffic network, which enhance the traffic congestion and worsen the performance of road networks, so that travellers waste a lot of travel time on the road, not only causing environmental contamination, but also leading to more traffic accidents [1].

Moreover, on crowded roads, frequent starting and braking make the exhaust gas density obviously higher than in other situations. Traffic emissions in urban traffic networks mainly contain carbon monoxide (CO), nitrogen oxides (NOx) and hydrocarbons (HC), suspended particulates and a little amount of sulphur dioxide (SO2), and aldehydes (RCHO) [2]. As such, traffic emissions are the major source of air pollution in urban areas and they contain many potential carcinogens. Many studies have found that exposure to traffic emissions may be associated with increased risk of cancers [3, 4], and several studies have confirmed that traffic emissions can be reduced under reasonable traffic control [5-9]. Therefore, reducing traffic congestion and controlling traffic pollution have become two major concerns for the urban transportation network planning and design.

There is a controversial topic on the left-turn vehicles. Owing to the conflict with the opposing through traffic flow, direct left-turn from driveways is considered as a contributor to the delay at intersections which may generate more accidents [10]. In order to avoid such collisions, the measures for prohibiting direct left-turns at intersections have been practiced in many states and cities [11-14].

However, because of the left-turn prohibitions, drivers cannot arrive at their destination directly. Drivers who intend to turn left initially at this intersection have to choose other alternatives, such as right-turn followed by U-turn [15]. Hence, it will lead to travelling additional distance and increasing traffic emissions. The ecological environment will be polluted and the adjacent roads may become more crowded, and in turn this will make the drivers adjust their route choice.

How do drivers choose their travel paths under left-turn prohibitions? Will the additional travelling distance of drivers' other alternatives increase total traffic emissions? Where should left-turn prohibitions be implemented, if we were to optimize the traffic network system from the view of environmental protection? For these questions, we propose a bi-level programming model considering both travel times and traffic emissions. The Frank-Wolfe algorithm is used to assign traffic demand for vehicles in the lower-level model and calculate the weighted combination of total travel
times and total traffic emissions, and then determine
where to implement left-turn prohibition in the up-
per-level model using an improved genetic algorithm,
such that the traffic network system is optimized. Us-
ing numerical examples it is shown that careful place-
ment of left-turn prohibitions can alleviate the traffic
congestion and emissions.

2. MODEL DESCRIPTION

The typical four-phase intersection is the most com-
mon intersection in urban traffic network, so the ma-
jority of traffic networks are regular lattice. As shown
in Figure 1, in order to adapt our model to the reality,
we have simplified the traffic network studied in this
paper into a lattice network \(G(N, A)\) with two-way roads,
where \(N\) and \(A\) are the sets of nodes and links, respec-
tively, and \(a \in A\). Each intersection is simplified as a
node. In addition, each node can be viewed as both
an origin and a destination and there is a trip demand
between any two nodes.

To study the traffic situation at the intersection, the
virtual nodes and virtual links need to be introduced.
Adding the virtual links to the network will not change
the original structure, but it just increases the size of the
network. As illustrated in Figure 1, a single intersection
is denoted by a network of four virtual nodes and 12 vir-
tual links, where \(N'\) and \(A'\) are the sets of virtual nodes
and virtual links, respectively, and \(a' \in A'\). Each permis-
sible movement through the intersection is represented
by a separate virtual link and each intersection move-
ment can be associated with the appropriate delay.

Before the formulation is discussed, this paragraph
presents the network notations used throughout this
paper. Let \(x_i\) (or \(x_j\)) represent the flow on real link \(a\) (or
virtual link \(a'\)). \(t_i\) is the travel time on real link \(a\). The
set of origin-destination (OD) pairs in the network is
denoted by \(W\) and \(\omega \in W\). The set of paths connecting
OD pair \(\omega\) is denoted by \(K^\omega\) and \(k \in K^\omega\). Let \(f_k^\omega\) be
the flow on path \(k\) connecting OD pair \(\omega\). \(q^\omega\) represents
the trip demand between OD pair \(\omega\). \(\delta_{a,k}\) (or \(\delta_{a',k}\)) is an
indicator variable. If real link \(a\) (or virtual link \(a'\)) is on path
\(k\), then \(\delta_{a,k}=1\) (or \(\delta_{a',k}=1\)), otherwise \(\delta_{a,k}=0\) (or \(\delta_{a',k}=0\)).

2.1 Lower-level Model

The real link performance function used here is the
equation developed by the U.S. Bureau of Public Road
(BPR) [16]. The equation is given by
\[
\begin{align*}
t_i(x_i) &= t_i^0 + \left(1 + \frac{x_i}{C_a}\right)^\alpha, \quad (1)
\end{align*}
\]
where \(t_i^0\) is the free-flow travel time on real link \(a\), \(C_a\) is
the capacity of real link \(a\), and \(\alpha\) and \(\beta\) are two positive
parameters.

Travel time \(d_{a'}\) on virtual link \(a'\) represents the time
of traffic flow going through the intersection, including
queuing time and running time. The running time affect-
ing the intersection can usually be assumed away, when
compared to the magnitude of the queuing time. Con-
sequently, for a signalized intersection, \(d_{a'}\) can be seen
as the queuing time on virtual link \(a'\) in a signal period.

Signal period \(c\) can be divided into green time \(g\) and red time \(r\) (including amber time). Obviously,
\(c = g + r\). Assume that vehicles arrive at the intersection
at the rate of \(\lambda = x_i\) and depart at the rate of \(\mu = C_a\)
during the green time (\(C_a\) is the capacity of virtual link
\(a'\)). The process of vehicle arrivals to and departures

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{A simplified network}
\end{figure}
from a signalized intersection is exactly a queuing system. Due to the assumption that $\lambda < \mu$, when the signal turns green, the queue clears after $g_0$, from the start of the green time. At this point, the arriving vehicles depart from the intersection in the rest of the green time $(g - g_0)$, until the signal turns red again. According to $(r + g_0)\lambda = \mu g_0$, $g_0$ is given by

$$g_0 = \frac{r}{1 - \frac{\mu}{\lambda}}. \quad (2)$$

The total vehicle time of queuing on virtual link $a$ at an intersection in each signal period is

$$d_a = \frac{\lambda}{2} (r + g_0) = \frac{r^2}{2(1 - \frac{\lambda}{\mu})}. \quad (3)$$

In a signal period $c$, the total number of vehicles is $\lambda c$, and thus the average queuing time per vehicle is given by

$$\bar{d}_a(x) = d_a / \lambda c = \frac{r^2}{2c(1 - \frac{\lambda}{\mu})} = \frac{r^2}{2c(1 - x_a / C_a)}. \quad (4)$$

Now, we propose a virtual link performance function $x_a \rightarrow t_a$ after left-turn prohibition and introduce a left-turn switching indicator $\theta_a$, which indicates whether the left-turn at virtual node $n'$ is prohibited. If the left turn is prohibited, then $\theta_a = 1$, otherwise $\theta_a = 0$. The vector of left-turn switching indicator is denoted as $\Theta = (\theta_{a(n')} : n' \in N')$. Then, the travel time of virtual link in left-turn direction after left-turn prohibition is given by

$$t_{\text{AB}}(x_a) = \max \left\{ d_{\text{AB}}(x_{\text{AB}}), \theta_{\text{AB}} M \right\}, \quad (5)$$

where $d_{\text{AB}}(x_{\text{AB}})$ is the average queuing time in left-turn direction $AB$ and $M$ is a very large positive number. If the left-turn direction $AB$ is prohibited, the switching indicator $\theta_{\text{AB}} = 1$ and $t_{\text{AB}}$ is a very large positive number, and as a result, the vehicles will not turn left. If the left-turn direction is not prohibited, then $\theta_{\text{AB}} = 0$ and $t_{\text{AB}}$ equals the average queuing time.

When the left-turn direction $AB$ is prohibited, the vehicles going straight can occupy the left lanes, thus increasing the number of straight lanes and indicates the capacity of straight direction $AC$. In this way, the travel time of virtual link in straight direction after left-turn prohibition is given by

$$t_{\text{AC}} = \frac{r^2}{2c(1 - x_a / C_{\text{AC}}^\text{new})}, \quad (6)$$

$$C_{\text{AC}}^\text{new} = C_a + \theta_{\text{AB}} \Delta C_a, \quad (7)$$

where $C_{\text{AC}}^\text{new}$ is the new capacity of straight direction after left-turn prohibition and $\Delta C_a$ is the growth of the capacity.

The travel time of virtual link in right-turn direction after left-turn prohibition remains unchanged and is written as

$$t_{\text{AD}} = \bar{d}_{\text{AD}}. \quad (8)$$

The user equilibrium (UE) assignment problem under left-turn prohibition can be formulated as the following mathematical programming:

$$\min \sum_{x_a} t_a(x_a) \omega_a + \sum_{x_a} t_a(x_a) \omega_a, \quad (9)$$

s.t. $\sum_{x_a} t_a(x_a) = q^a$, $\forall \omega \in W$,

$$f^* > 0, \quad \forall k \in K^*, \omega \in W,$$

$$x_a = \sum_{x_a} t_a(x_a) a, \quad \forall a \in A,$$

$$x_a = \sum_{x_a} t_a(x_a) a, \quad \forall a \in A'.$$

In this formulation, the objective value $z(x)$ consists of two parts. The first part is the sum of the integrals of the real link performance functions and the other part is the sum of the integrals of the virtual link performance functions.

### 2.2 Upper-level Model

On the one hand, the objective function in the upper-level programming model needs to consider the total travel time in traffic network, which includes the total travel time on real links and on virtual links, i.e.,

$$\min \sum_{x_a} t_a(x_a) + \sum_{x_a} t_a(x_a) . \quad (10)$$

On the other hand, among the varieties of traffic emissions, CO is considered as an important indicator for the level of air pollution [17]. For the sake of simplification, we have chosen CO as the only evaluation factor of traffic emissions in urban areas. Yin and Lawphongpanich [18] proposed the following function to estimate the CO per vehicle discharged on link $a$:

$$e_a(x_a) = 0.2038 \times t_a(x_a) \times e^{0.7962 / l_a}, \quad (11)$$

where $l_a$ is the length of link $a$ and is equal to the product of free-flow travel time and average free-flow speed:

$$l_a = t_a^0 \times v_a^0. \quad (12)$$

Here, $t_a^0$ and $v_a^0$ are two given parameters, and hence the vehicular CO emissions $e_a$ only depend on traffic flow $x_a$ on link $a$.

Here, the emissions of both real links and virtual links are considered. The objective of the upper-level programming model is to minimize the weighted combination of total travel time and total traffic emissions:

$$\min_{x_a} \gamma \left( \sum_{x_a} t_a(x_a) + \sum_{x_a} t_a(x_a) \right) +$$

$$(1 - \gamma) \phi \left( \sum_{x_a} e_a(x_a) \right), \quad (13)$$

where $\gamma \in [0, 1]$ is the weight of total travel time and $\phi$ is unit conversion factor, which is to match the units of the travel time and the traffic emissions.
All in all, in the lower-level model, the UE minimization program describes the flow pattern resulting from each driver's choice of the shortest travel-time route from their own interests. In the upper-level model, the planners will set the left-turn prohibitions according to the total travel time and total traffic emissions, and the drivers will accept the unified dispatching to change their routes so as to optimize the traffic system rather than their own.

It is worth mentioning that there are some related studies on left-turn prohibition [12-15,19-21]. Most of these existing studies focus on the microscopic level and study the vehicle running status from the view of signal design or security analysis. In contrast, this paper proposes a UE assignment model from the macroscopic perspective and also considers the traffic emissions in addition to the traditional travel time for the left-turn prohibition traffic management. Thus, the emphases of these existing studies are different from that of this paper.

3. MODEL ALGORITHM

3.1 Algorithm for Lower-level Model

The upper-level model gives the vector of left-turn switching indicator Θ, namely the specific locations of the left-turn prohibition. Applying to the lower-level model, the Frank-Wolfe algorithm was used for UE to assign trip demand for vehicles to get the link flows, and then calculate the weighted combination of total travel time and total traffic emissions were calculated in the upper-level model to be minimum.

The detailed steps of the Frank-Wolfe algorithm are given as follows:

- **Step 1**
  Initialization. Set initial feasible solutions \( x^1_a \) (a ∈ A) and \( x^1_{a'} \) (a′ ∈ A′). Perform the all-or-nothing assignment based on \( t^1_a = t_a(x^1_a) \) (a ∈ A) and \( r^1_{a'} = r_{a'}(x^1_{a'}) \) (a′ ∈ A′).
  Obtain the set of real link flows \( \{ x^1_a \} \) and virtual link flows \( \{ x^1_{a'} \} \). Set iteration counter n = 1.

- **Step 2**
  Update of travel time. Set
  \[
  t^\ast_a = t_a(x^\ast_a), \forall a;
  \]  
  \[
  r^\ast_{a'} = r_{a'}(x^\ast_{a'}), \forall a'.
  \]  

- **Step 3**
  Direction finding. Perform the all-or-noting assignment based on \( \{ t^1_a \} \) and \( \{ r^1_{a'} \} \). This yields the set of auxiliary real link flows \( \{ y^1_a \} \) and auxiliary virtual link flows \( \{ y^1_{a'} \} \).

- **Step 4**
  Line search. Find the optimal step length \( \lambda^{(1)} \) by solving
  \[
  \min Z(\lambda) = \sum_{\omega \in \Omega} \int_{t_0}^{t_1} t_\omega(s) d\omega + \sum_{\omega \in \Omega} \int_{t_0}^{t_1} r_\omega(s) d\omega.
  \]  

- **Step 5**
  Update. Set
  \[
  x^{n+1}_a = x^n_a + \lambda^{(n)} (y^n_a - x^n_a), \forall a;
  \]
  \[
  x^{n+1}_{a'} = x^n_{a'} + \lambda^{(n)} (y^n_{a'} - x^n_{a'}), \forall a'.
  \]

- **Step 6**
  Convergence test. The algorithm stops if the convergence criterion
  \[
  \sqrt{\sum_a (x^{n+1}_a - x^n_a)^2} + \sum_{a'} (x^{n+1}_{a'} - x^n_{a'})^2 \left/ \left( \sum_a x^n_a + \sum_{a'} x^n_{a'} \right) \right. \leq \varepsilon
  \]  

is met, where \( \varepsilon \) is a preset precision parameter. The current solution \( \{ x^{n+1}_a \} \) and \( \{ x^{n+1}_{a'} \} \) can be seen as the set of equilibrium link flows; set \( n = n + 1 \) and go to step 2, otherwise.

3.2 Algorithm for Upper-level Model

The traditional genetic algorithm has been always used in this class of problems, but its ability to explore global space is limited, so it is easy to converge to the local optimal solution. As such, according to the features of our left-turn prohibition model, the traditional genetic algorithm was improved by combining the enumerative algorithm with the genetic algorithm in upper-level model, which effectively improved the speed and accuracy of calculation. The procedures of the algorithm are described as follows:

- **Step 1**
  Generate initial population. The population is initialized based on the left-turn switching indicator Θ. Decompose vector Θ into fragments with each intersection, that is \( \Theta = (\Theta_1, \Theta_2, ..., \Theta_N) \), where \( N \) is the number of intersections in the traffic network. Use the enumerative algorithm to calculate the optimal solution of each \( \Theta \), with other left-turn switching indicator set to zero, to make the weighted combination of total travel time and total traffic emissions be minimized. Combine all optimal solutions \( \Theta^* \) into \( \Theta^* \), namely \( \Theta^* = (\Theta^*_1, \Theta^*_2, ..., \Theta^*_N) \) and \( \Theta^* \) can be seen as an excellent individual. Generate m chromosomes randomly, each of which has \( n \) genes, where \( n \) is the number of virtual nodes in traffic network and \( n = 4N \). Hybridize each individual with the excellent individual \( \Theta^* \) to form an excellent initial population.

- **Step 2**
  Fitness value calculation. Apply the initial population in step 1 into the lower-level model to get the link flows, and then calculate the weighted combination
Z of total travel time and total traffic emissions. The fitness of individual chromosome in the population is computed by

\[ F(i) = 1 - \frac{Z}{\text{sum}(Z)}. \]  

(20)

That is to say, the fitness increases as the objective value gets smaller.

Step 3

Selection. In order to let the excellent individuals have more chance to survive and maintain the diversity of the population, replace the individual with the smallest fitness by the individual with the largest fitness. The individual with the largest fitness enters the next generation directly and the rest individuals are selected by using the roulette method.

Step 4

Crossover. Perform crossover operation to generate offspring individuals according to the crossover probability \( P_c \).

Step 5

Mutation. Mutation operation is performed on each selected individual by mutation probability \( P_m \).

Step 6

Termination condition test. If the generation is \( K \) or the fitness value no longer increases for several continuous generations, then terminate the algorithm; otherwise, go to step 2.

Figure 2 illustrates the solving process of the algorithm in the upper-level.

4. NUMERICAL EXAMPLES

Consider the network shown in Figure 1. The trip demands and free-flow travel times are given in Tables 1 and 2, respectively. Set signal period \( c = 120 \) s, red time \( r_L = 90 \) s of left turn, red time \( r_D = 90 \) s of straight

<table>
<thead>
<tr>
<th>OD</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>0</td>
<td>50</td>
<td>50</td>
<td>100</td>
<td>50</td>
<td>50</td>
<td>100</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>50</td>
<td>50</td>
<td>0</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>4</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>0</td>
<td>100</td>
<td>100</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>100</td>
<td>50</td>
<td>100</td>
<td>0</td>
<td>100</td>
<td>50</td>
<td>100</td>
<td>50</td>
</tr>
<tr>
<td>6</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>7</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>0</td>
<td>50</td>
</tr>
<tr>
<td>8</td>
<td>50</td>
<td>100</td>
<td>50</td>
<td>50</td>
<td>100</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( t^*_i )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>34</td>
<td>0</td>
<td>46</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>36</td>
<td>0</td>
<td>33</td>
<td>0</td>
<td>48</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>42</td>
<td>0</td>
<td>0</td>
<td>35</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>37</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>46</td>
<td>0</td>
<td>39</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>48</td>
<td>0</td>
<td>40</td>
<td>0</td>
<td>35</td>
<td>0</td>
<td>41</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>37</td>
<td>0</td>
<td>45</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>45</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>41</td>
<td>0</td>
<td>0</td>
<td>31</td>
<td>0</td>
<td>45</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>40</td>
<td>0</td>
<td>31</td>
<td>0</td>
<td>45</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>32</td>
<td>0</td>
<td>48</td>
<td>0</td>
</tr>
</tbody>
</table>
direction, and red time $r_o = 50$ s of right turn. Capacities $C_R = C_L = 1000$ pcu/h and $C_D = 2000$ pcu/h of right turn, left turn and straight going, respectively, owing to the numbers of their lanes being different. The capacity of straight-going increases to 3,000 pcu/h, when the left turn on the same side is prohibited. The model parameters $\alpha = 0.15$ and $\beta = 4$, the average free-flow speed $v_{\text{av}} = 15$ m/s, the weight of total travel time $\gamma = 0.7$, and the unit conversion factor $j = 10^{-4}$ s/g. In the upper-level model, set the number of populations to 50, the crossover probability to 0.7, the mutation probability to 0.3 and the number of iterations to 200.

Figure 3(a) shows the result of traffic assignment before left-turn prohibition. Figure 3(b) displays the specific locations of left-turn prohibition and the flow assignment pattern after left-turn prohibition. The flows of real links and left turns are also marked in the figure, where symbol (●) represents that the vehicles in this direction are prohibited to turn left.

It can be seen from Figure 3(b) that the trip demands have been assigned reasonably after left-turn prohibition, and the left-turn prohibition is implemented at only these locations with low left-turn volume. As a result, when the left-turn volume is high, the left-turn movement should be allowed at the intersection.

---

**Figure 3** – Traffic assignment before and after left-turn prohibition

**Figure 4** – Values of objective before and after improving the algorithm

(a) Total travel time

(b) Total Traffic Emissions
For the upper-level model, Figure 4(a) compares the iterative processes obtained from the traditional genetic algorithm and improved algorithm when the weight \( \gamma = 1 \) (i.e. we just consider the total travel time of the network). One can see that the traditional genetic algorithm needs about 35 minutes to reach the optimal value 5.9163×10^5s with 138 iterations. However, the improved algorithm just needs about 20 minutes to reach the optimal value 5.9131×10^5s with 87 iterations, which is even more optimized than the former. Compared with the total travel time 5.9849×10^5s before left-turn prohibition, the optimal value is reduced by 1.20%.

Figure 4(b) just considers the total traffic emissions of the network, namely the weight \( \gamma = 0 \). It can also be seen that the performance of the improved genetic algorithm is much better than the traditional genetic algorithm. The total traffic emission is 9.8963×10^5s before left-turn prohibition, and it converges to 9.8353×10^5s after left-turn prohibition, being reduced by 0.62%. It may be because the increase of traffic emissions caused by the alleviation of traffic congestion exceeds the increase of emissions caused by other alternatives' additional distance.

Thus, the improved algorithm can not only enhance the computing speed and reduce the iteration steps, but also advance the accuracy of the optimal solution. As shown in Figure 4(a), the initial value of the iteration is larger than the value before left-turn prohibition, so it may be concluded that unsuitable placement of left-turn prohibition may result in an increase of the system cost.

It can be also learned from the results that the effect of left-turn prohibition simply considering the total travel time is much better than simply considering the total traffic emissions. Therefore, how to balance the weight of travel time and traffic emissions is particularly important.

5. CONCLUSION

With the optimal objective considering both travel time and traffic emissions, a bi-level programming model for left-turn prohibition planning is proposed. The vehicles' queuing model at intersections before and after left-turn prohibition is given, the left-turn switching indicator \( \theta \) is put forward, and the traffic flow is assigned utilizing the UE rule. The genetic algorithm for the upper-level model has been also improved. The improved model can be used to obtain the concrete locations of left-turn prohibition with the information of network structure, free-flow travel time and trip demand, to make the total network cost minimum.

Numerical results show that careful placement of left-turn prohibition can reduce both traffic congestion and emissions. The improved algorithm can effectively enhance the speed and accuracy of the calculation. Therefore, the research and analysis in this paper are expected to be useful in designing and operating the left-turn prohibition in urban traffic network and to be of value in alleviating traffic congestion and reducing the pollutant emissions.

ACKNOWLEDGEMENTS

We extend our sincere thanks to the anonymous reviewers for their constructive comments. The work described in this paper was jointly supported by Grants from the National Natural Science Foundation of China (71261016), the Program for New Century Excellent Talents in University (NCET-12-1016), and the Natural Science Foundation of Inner Mongolia of China (2014JQ03).

REFERENCES


