Presenting an Optimization Model for Signal Setting in the Network

Shahriar Afandizadeh, Seyed Bahman Moghimidarzi and Navid Kalantari
Department of Civil and Environmental Engineering, Iran University of Science and Technology, Tehran, Iran

Corresponding Author: Seyed Bahman Moghimidarzi, Department of Civil and Environmental Engineering, Iran University of Science and Technology, Tehran, Iran

ABSTRACT

In urban network, there is a realistic relationship between fixed-time signals setting and route perception. Because increasing the capacity of links in many districts of cities are impossible or make a lot of cost, signals setting can be helpful for network improvement. Using a suitable algorithm for signals setting is an important matter that is being attended in recent years. This matter can be considered as a continuous bi-level network design problem which signals are set in one level and traffic assignment is solved in another level. In this study, by using a Simulated Annealing (SA) algorithm, signals parameters were set and the suitable pattern from this parameters were solved. Moreover, in order to increase the rate of running, the Gradient Projection (GP) algorithm which is a path based traffic assignment method, was used. The results show that signal optimization in the network cause in improvement of the objective function.

Key words: Gradient projection method, simulated annealing, signal timing, bi-level programming

INTRODUCTION

Traffic signals have been used for reducing congestion and improving safety in urban road networks. Meanwhile many strategies such as public transportation preemption and reduction of environmental pollutions are possible by traffic signals. As traffic signals play an important role in the delay reduction strategies usually used in urban areas, traffic signal timing optimization was early recognized as one of the most important, yet challenging problem in transportation planning and traffic management.

Four types of signal optimization paradigms could be seen in the literature; early studies on traffic signal optimization have only considered an isolated signalized junction such as that of Webster (1958). In these strategies, a fixed time signal timing plan was developed based on historical traffic data on the intended junction. These methods did not consider the change in route flows which occur by the disturbance of the traffic equilibrium state, caused altering signal timings (Allsop, 1974). Later, more advanced fixed time strategies were emerged in which a group of signalized intersection was considered and their signal timing was simultaneously optimized (Cipriani and Fusco, 2004; Lee and Machemehl, 2005; Robertson, 1968; Sadabadi et al., 2008). These methods usually take into account the changed induced in the traffic equilibrium assignment. The so-called Demand Responsive strategies, has also been used in many metropolitan areas where real time traffic flow information is used for optimizing a single or a group of traffic signals (Hunt et al., 1981; Sun and Benekohal, 2005). More recently methods based on dynamic
traffic assignment and short term traffic flow prediction have been used to synchronize and optimize traffic signals (Abu-Lebdeh and Benekohal, 2003; Chow, 2003).

The focus of this paper was on the second paradigm of traffic signal timing. That is, as mentioned earlier, to optimize a group of fixed time traffic signals, while anticipating the traffic equilibrium state. This problem could be noticed as an instance of Network Design Problem (NDP) (LeBlanc et al., 1985; Yang and Bell, 1998; Ben-Ayed et al., 1988) NDPs could be characterized by a bi-level programming structure. The upper level problem is the planner’s objective which can be a network performance function such as total travel time, emission, total system cost, etc. The lower level problem is the user problem which, upon the planner’s decision, tries to minimize her/his own travel time. This behavior is modeled to be the user equilibrium traffic assignment model. Due to the non-smoothness and non-convexity of the objective function and the feasible region, respectively, such problems are among the most challenging problems in mathematical programming.

Different solution algorithms have been devised for solving the NDP. These methods can be classified as linearization methods (LeBlanc et al., 1985; Ben-Ayed, 1988), approximation methods (Danzig et al., 1976; Marcotte, 1981; Poorzahedy and Turnquist, 1982; Bergendorff et al., 1997; Hearn and Ramana, 1998), sensitivity analysis based methods (Friesz et al., 1990; Yang, 1997), heuristic and meta-heuristic methods (Friesz et al., 1992; Poorzahedy and Abulghasemi, 2005; Jahangiri et al., 2011). More recently some studies have used single level formulations, by the means of the so-called; marginal function method (Meng et al., 2001) and also methods based on Mathematical Problem with Equilibrium Constraint (MPEC) and Mathematical Problem with Complimentary Constraint (MPCC) formulations for solving NDPs.

In the present study, Simulated Annealing (SA) method was used for solving the fixed time signal setting problem. In this study, the network level effects on the signal timing in terms of route switching were taken into account using the UE traffic assignment. In order to expedite the solution and since the shortest path calculation is one of the most time consuming procedures in the UE traffic assignment problem (Sheffi, 1985), a path-based traffic assignment method was applied; namely, the Gradient Projection (GP) method was implemented. As the paths are not expected to change dramatically, the use of path-based traffic assignment problem would reduce the total run time as the paths are saved.

MODEL FORMULATION

As mentioned previously in this paper, a bi-level network design problem is introduced for optimization network signal timing. All signals have been assumed to use fixed time plans. In this problem, green ratio of signals is modeled of the one decision into a signal mathematical programming problem as given in Eq. 1:

\[
(\psi) = \arg \min_{\psi} \{ f(\psi), t(\psi') \}
\]

Subject to:

\[
\psi \in \Lambda
\]

\[
f \in \Omega
\]
Where:
\( \phi = \) The vector of optimal signal control parameters (e.g., green time or green ratio)
\( f^* = \) The vector of optimal User Equilibrium (UE) link flows
\( t = \) The vector of UE link travel times
\( \Lambda = \) The set of feasible control parameters
\( \Omega = \) The set of feasible and equilibrium flows (actually the User Equilibrium constraint)

This problem could be also noticed as the Urban Network Design Problem (UNDP) since it considers one of the four major decisions usually encountered in urban transportation networks:

- The problem of limited capacity enhancement due to traffic engineering efforts such as efficiency improvement
- Signal timing problem at urban intersection
- Optimization of the direction and configuration of urban one-way and two-way streets
- Congestion pricing

This paper deals with the second component of the Urban Network Design Problem (UNDP). This problem could be easily reformulated as a bi-level Network Design Problem. Each level of this bi-level problem deals with a specific group and decision; the upper level problem could be known as the planner (leader) problem and the lower level problem could be noticed as the user (Follower) problem. The upper level problem of the NDP is the design problem which minimizes some network level performance measures. The lower level problem is the UE traffic assignment problem that estimates the UE flow on the network based on the planner decision in the upper level problem.

Consider the graph given as \( G(N,A) \) in which \( N \) is the set of nodes and \( A \) is the set of all links in the network. Let the set of signalized intersection be denoted as \( N_s \subset N \). For each \( i \in N_s \) a set of control parameters could be defined as \( \phi_i \). These control parameters may include phasing plan, green time, red time, cycle length and offset. Among the aforementioned control parameters, the green times were considered in this paper. As the traffic assignment method used in this paper was not dynamic and, meanwhile, demand responsive controls were not in the scope of this paper, cycle times and offset could not be taken into account in the optimization model. The traffic control signal is assumed to reduce the capacity proportional to the ratio of green time to cycle length. This ratio sums up to one for the all phases of a given signal. Thus, the inclusion of cycle length into this problem is redundant and could be omitted since it would not affect optimization. The formulation of this problem is given as follows:

\[
\text{Min} \sum_{i \in \Lambda} x^t \cdot \tau_a (X, \phi) \tag{2}
\]

Subject to:

\[
\frac{g_{ib}}{p_i} \leq \frac{g_{ib}}{p_i} - 1 \forall i \in N_s
\]

\[
\left( \frac{g}{p} \right)_{mn} \leq \frac{g_{ib}}{p_i} \leq \left( \frac{g}{p} \right)_{mn} \forall i \in N_s, \forall b \in \phi
\]
\[
\min \sum_{w=1}^{\infty} t(w)dw
\]

Subject to:

\[
\sum_{\rho \in \rho} f_{\rho}^r = q_r^s \quad \forall r \in W
\]

\[
\sum_{\rho \in \rho} \sum_{\alpha \in \alpha} f_{\rho}^r \delta_{\rho}^{rs} = x_\alpha \quad \forall \alpha \in \Lambda
\]

\[
f_{\rho}^r \geq 0 \quad \forall r \in W \quad \forall \rho \in \rho
\]

Where:

- \(gib\) = The amount of green time given to the b phase of signal i
- \(\rho_i\) = The cycle length of signalized intersection i
- \(f_{\rho}^r\) = The flow on route \(\rho\) between origin \(r\) and destination \(s\)
- \(q_r^s\) = The demand between \(r\) and \(s\)
- \(W\) = The set of origin destination pairs
- \(x_\alpha\) = Equilibrium flow in link \(\alpha\)
- \(\delta_{\rho}^{rs}\) = A 0-1 binary variable that is one whenever link \(\alpha\) is on route \(\rho\) between origin \(r\) and destination

\((g/p)_{\min}, (g/p)_{\max}\) = The minimum and maximum of green ratios which are considered 0.2 and 0.8, respectively

As mentioned previously this problem is a non-convex non-smooth optimization problem. Thus, stochastic optimization methods such as Meta-Heuristic method could be used to obtain the optimal or near optimal solution of this problem. Other methods used for solving such bi-level optimization problems, although deterministic, could only find the local optimal solution of the problem. Thus in this paper, Simulated Annealing (SA) was used to solve the NDP given in Eq. 2. SA is well-known for its superior robustness for solving non-convex optimization problems and has been analytically proven to find the optimal solution if used adequately (Kirkpatrick et al., 1983; Friesz et al., 1992; Jahangiri et al., 2011). The major short coming of SA is that it requires much iteration for finding the optimal solution. Meaning that, in this NDP, many UE traffic assignments which may be very time consuming should be solved.

In order to overcome the aforementioned short coming of a path-based traffic assignment method, Gradient Projection (GP) algorithm was used in this study. Path enumeration, that was previously known as a disadvantage in path based traffic assignment method, is not a limitation today because ad hoc high speed memory of computers which have emerged. Conversely, this property could be promising for the considered problem. The paths could be enumerated once in the first UE traffic assignment and could only be checked in the following UE assignments performed in the successive iteration of SA. This property was used in this paper in order to expedite the UE assignment by implementing the favorable properties of both SA and GP, as a path-based traffic assignment method which will be seen in the following section.

**SOLUTION ALGORITHM**

The solution algorithm proposed in this paper consisted of two major components; the GP traffic assignment and SA. As follows, first, the GP traffic assignment algorithm
is presented then the implementation of SA to the signal timing network design is illustrated.

GP has been originally proposed by Bertsekas Jayakrishnan et al. (1994) for mathematical programming. It has been successfully applied to the UE traffic assignment problem by Jayakrishnan et al. (1994); then, the effects of flow update policies are implemented (Chen and Jayakrishnan, 1998). The steps of the GP traffic assignment method could be summarized as follows (interested readers are referred to the original paper for more details on the GP traffic assignment):

**Step 1:** Path enumeration: for each origin destination pair do:

- Let \( n = 0 \) and:

\[
{x}_n = 0, \quad t_n = t_n(x_n) \quad \forall \alpha \in A
\]  
(3)

- Find the shortest path, \( k_n \), between the origin destination pair \( r \) and \( s \)

\[
k_n = \{k_n\}
\]  
(4)

- Perform an all or nothing assignment

\[
f^0_{k_n} = q_0 \quad \forall r, s
\]  
(5)

- Update link flows and travel times based on the all or nothing assignment performed in 1-3

**Step 2**

*Column generation:* Find the shortest path between each origin destination pair based on the current flows on the network and add the new path to the path set \( K_n \):

- let \( n = n+1 \)
- Update link travel times based on the current flows:

\[
t_\alpha = t_\alpha(x_n)
\]  
(6)

- Find the shortest path, \( k_n \), between each origin destination pair
- If \( k_n \notin k_n \), let \( k_n = k_n \cup \{k_n\} \); Otherwise let the shortest path in \( k_n \) be \( \bar{k}_n \)

**Step 3**

*Traffic assignment:* Find \( \xi^*_k \) and \( \xi^*_k \) the travel time and travel time derivative, respectively for all paths, other than the shortest path:

\[
\xi^*_k = \sum_{\alpha \in A} \nabla t_\alpha(x_n) \delta_\alpha^0 \quad \forall k \in K_n
\]

\[
\xi^*_k = \sum_{\alpha \in A} \nabla t_\alpha(x_n) (\delta_\alpha^0 - \delta_\alpha^k) \quad \forall k \in K_n, k \neq \bar{k}_n
\]  
(7)
• Update the flows on all nonshortest paths:

\[
f^r_k = \max \left\{ f^r_k - \frac{\alpha}{s_k^r} \left( d^w_k - d^r_k \right) \right\}; \forall w \in W, \forall k \in k_r, k \neq k_w
\]  

(8)

• If \( f^r_k = 0 \) remove path \( k \) from path set \( k_r \)
• Update the shortest path flow:

\[
f^r_k = q_{r_k} - \sum_{k \in k_r} f^r_k
\]  

(9)

• Update link flows:

\[
x^r_k = \sum_i \sum_{k \in k_r} f^r_k \lambda^r_{ai} \quad \forall a \in A
\]  

(10)

Step 4

Convergence check: if the error is less than the allowable level of error \( \varepsilon \) (e.g. 0.001) stop; otherwise go to Step 2:

\[
E_{fr} = \max_{a} \sum_{k \in k_a} f^r_k \left( \frac{d^w_k - d^r_k}{d^w_k} \right)
\]  

(11)

• This procedure was called GPA in this paper and was incorporated in the SA in order to solve the proposed NDP. The SA used in this paper could be summarized as follows:

Step 1

Initiation:

• Define initial temperature, \( T_{int} \), final temperature, \( T_{fin} \), the number of iteration, \( L \) and the temperature reduction factor \( \alpha \)
• Assign the ratio of 0.5 to all the signalized intersection and name the initial solution vector \( i \)
• Perform GPA and find the objective function values denoted by \( E_i \) and set \( E_{best} = E_i \)
• Put the value of \( T_{int} \) into \( T_n \)

Step 2

Neighborhood search:

• Randomly choose a signalized intersection from the candidate intersections set and name the selected intersection \( h \)
• Find the new control parameter for it as:

\[
\frac{g_{hb}}{p_{h}} = 1 - \frac{g_{hb}}{p_{h}} = \text{Rand} \left[ \frac{g}{p_{\text{min}}}, \frac{g}{p_{\text{max}}} \right]
\]  

(12)
Then go to next
- Perform GPA and find the objective function value denoted by $E_i$
- Find the amount of change in the objective function $\Delta E_i = E_i - E_i$

**Step 3:** Update to current and the current best solution. If $\Delta E_i \geq 0$ put $i = j$ and also if $E_i \leq E_{\text{best}}$ put $E_j$ into $E_{\text{best}}$ and $F_{\text{best}} = j$; otherwise find a random number $\alpha$ and if $\exp(\Delta E_i/Tn) \geq \alpha$, then $i = j$ and go to step 4; otherwise go to step 2

**Step 4:** If $n < L$ then let $n = n + 1$ go to step 2; otherwise $n = 0$ and go to step 5

**Step 5**

**Convergence check:**
- Reduce the temperature $T_{n+1} = \alpha T_n$
- If $T_{n+1} = T_{\text{fin}}$ go to the next step; otherwise go to step 2
- Output $E_{\text{best}}$ and $F_{\text{best}}$ as the best objective value and the solution vector, respectively

The model and solution algorithm was applied to the city networks.

**MODEL IMPLEMENTATION**

The network used in this study is Sioux Falls network which consists of 76 links, 24 nodes that 12 of them are signalized intersections and with 428 Origin Destination (O/D) pairs. Sioux Falls network is shown in Fig. 1 and the O/D matrix is given in Table 1. The darker nodes in Fig. 1 are the signalized intersections and their cycle lengths are considered 120 sec. Moreover, to set the dimension of problem from day to hour, the given O/D matrix is multiply by 0.11. As a note, due to four approaches format which are used in this programming, a dummy node, node 25, is made to correct difficulties of node 10.

The cost function used to model the travel time of the link was the changed form of BPR function and its parameters, $a$ and $b$ measures, are given in the study of LeBlanc et al. (1985). In addition to the BPR cost function, the Webster delay function as in Eq. 14 should be used for all the links entering signalized intersections:

$$t(x_a) = a + b(x_a)^4$$  \hspace{1cm} (13)

$$w = 0.9 \left[ \frac{1 - \left( \frac{\rho_{sa}}{\rho_a} \right)^2}{2 \left( 1 - \frac{x_s}{S_s} \right)} \right] + \frac{\frac{x_s}{S_s} \left( \frac{\rho_{sa}}{\rho_a} \right)}{2 \left( 1 - \frac{x_s}{S_s} \right)} \left( \frac{\frac{x_s}{S_s} \left( \frac{\rho_{sa}}{\rho_a} \right)}{2 \left( 1 - \frac{x_s}{S_s} \right)} \right)$$  \hspace{1cm} (14)

In this equation all the used variables are as defined previously and $S_s$ is the practical capacity of link $a$. As the cost function given by Webster is not continuous near the practical capacity of the
Fig. 1: Sioux Falls network

Table 1: The best calculated results of \( g / b \) for all signalized intersections

<table>
<thead>
<tr>
<th>Origin</th>
<th>Destination</th>
<th>Green ratio</th>
<th>Origin</th>
<th>Destination</th>
<th>Green ratio</th>
<th>Origin</th>
<th>Destination</th>
<th>Green ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4</td>
<td>0.69</td>
<td>5</td>
<td>9</td>
<td>0.56</td>
<td>14</td>
<td>15</td>
<td>0.44</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>0.69</td>
<td>8</td>
<td>9</td>
<td>0.44</td>
<td>19</td>
<td>15</td>
<td>0.44</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>0.31</td>
<td>10</td>
<td>9</td>
<td>0.56</td>
<td>22</td>
<td>15</td>
<td>0.56</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>25</td>
<td>15</td>
<td>0.56</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>0.71</td>
<td>9</td>
<td>10</td>
<td>0.64</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>0.71</td>
<td>11</td>
<td>10</td>
<td>0.36</td>
<td>8</td>
<td>16</td>
<td>0.23</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>0.29</td>
<td>16</td>
<td>10</td>
<td>0.36</td>
<td>10</td>
<td>16</td>
<td>0.77</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>25</td>
<td>10</td>
<td>0.64</td>
<td>17</td>
<td>16</td>
<td>0.23</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>0.77</td>
<td></td>
<td></td>
<td></td>
<td>18</td>
<td>16</td>
<td>0.77</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>0.23</td>
<td>4</td>
<td>11</td>
<td>0.52</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>0.77</td>
<td>10</td>
<td>11</td>
<td>0.48</td>
<td>12</td>
<td>16</td>
<td>0.38</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>12</td>
<td>11</td>
<td>0.48</td>
<td>16</td>
<td>17</td>
<td>0.62</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>0.6</td>
<td></td>
<td></td>
<td></td>
<td>14</td>
<td>17</td>
<td>0.62</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>0.4</td>
<td>11</td>
<td>14</td>
<td>0.73</td>
<td>15</td>
<td>19</td>
<td>0.28</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>0.4</td>
<td>15</td>
<td>14</td>
<td>0.27</td>
<td>17</td>
<td>19</td>
<td>0.72</td>
</tr>
<tr>
<td>16</td>
<td>8</td>
<td>0.6</td>
<td>23</td>
<td>14</td>
<td>0.73</td>
<td>20</td>
<td>19</td>
<td>0.72</td>
</tr>
</tbody>
</table>

link and so the KKT condition of this problem cannot be satisfied, the cost function given by Van Vuren and Van Vliet (1992) was used in this paper which is given in Eq. 15.

\[
\text{for } x < \bar{x}_{\text{link}} = s \lambda - \frac{s \lambda}{V}.
\]

Then:

\[
w = w_i + w_j = \frac{C L s (1 - \lambda)^3}{2(s - x)} + \frac{x}{2s \lambda (s \lambda - x)}
\]

\[
\text{for } x \geq \bar{x}_{\text{link}}
\]

37
Table 2: Origin Destination matrix for Sioux Falls network (per thousand per day)

<table>
<thead>
<tr>
<th></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>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0.1</td>
<td>0.1</td>
<td>0.5</td>
<td>0.2</td>
<td>0.3</td>
<td>0.5</td>
<td>0.8</td>
<td>0.5</td>
<td>1.3</td>
<td>0.5</td>
<td>0.2</td>
<td>0.5</td>
<td>0.3</td>
<td>0.5</td>
<td>0.5</td>
<td>0.4</td>
<td>0.1</td>
<td>0.3</td>
<td>0.3</td>
<td>0.1</td>
<td>0.4</td>
<td>0.3</td>
<td>0.1</td>
</tr>
<tr>
<td>2</td>
<td>0.1</td>
<td>0</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
<td>0.4</td>
<td>0.2</td>
<td>0.4</td>
<td>0.2</td>
<td>0.6</td>
<td>0.2</td>
<td>0.1</td>
<td>0.3</td>
<td>0.1</td>
<td>0.1</td>
<td>0.4</td>
<td>0.2</td>
<td>0.1</td>
<td>0.1</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0.1</td>
<td>0</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
<td>0.3</td>
<td>0.1</td>
<td>0.3</td>
<td>0.2</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.5</td>
<td>0.2</td>
<td>0.2</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>0.4</td>
<td>0.4</td>
<td>0.7</td>
<td>0.7</td>
<td>1.2</td>
<td>1.4</td>
<td>1.6</td>
<td>0.5</td>
<td>0.6</td>
<td>0.5</td>
<td>0.8</td>
<td>0.5</td>
<td>0.1</td>
<td>0.2</td>
<td>0.3</td>
<td>0.2</td>
<td>0.4</td>
<td>0.5</td>
</tr>
<tr>
<td>5</td>
<td>0.2</td>
<td>0.1</td>
<td>0.1</td>
<td>0.5</td>
<td>0</td>
<td>0.2</td>
<td>0.2</td>
<td>0.5</td>
<td>0.8</td>
<td>1</td>
<td>0.5</td>
<td>0.2</td>
<td>0.2</td>
<td>0.1</td>
<td>0.2</td>
<td>0.5</td>
<td>0.2</td>
<td>0.0</td>
<td>0.1</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>6</td>
<td>0.3</td>
<td>0.4</td>
<td>0.3</td>
<td>0.4</td>
<td>0</td>
<td>0.2</td>
<td>0</td>
<td>0.4</td>
<td>0.8</td>
<td>0.4</td>
<td>0.8</td>
<td>0.4</td>
<td>0.8</td>
<td>0.4</td>
<td>0.2</td>
<td>0.2</td>
<td>0.1</td>
<td>0.2</td>
<td>0.9</td>
<td>0.5</td>
<td>0.1</td>
<td>0.2</td>
<td>0.3</td>
<td>0.1</td>
</tr>
<tr>
<td>7</td>
<td>0.5</td>
<td>0.2</td>
<td>0.1</td>
<td>0.4</td>
<td>0</td>
<td>0.2</td>
<td>0</td>
<td>0.4</td>
<td>0</td>
<td>1</td>
<td>0.6</td>
<td>1.9</td>
<td>0.5</td>
<td>0.7</td>
<td>0.4</td>
<td>0.2</td>
<td>0.5</td>
<td>1.4</td>
<td>1</td>
<td>0.2</td>
<td>0.4</td>
<td>0.5</td>
<td>0.2</td>
<td>0.5</td>
</tr>
<tr>
<td>8</td>
<td>0.8</td>
<td>0.4</td>
<td>0.2</td>
<td>0.7</td>
<td>0.5</td>
<td>0.8</td>
<td>1</td>
<td>0</td>
<td>0.8</td>
<td>1.6</td>
<td>0.8</td>
<td>0.6</td>
<td>0.6</td>
<td>0.4</td>
<td>0.6</td>
<td>0.6</td>
<td>2.2</td>
<td>1.4</td>
<td>0.3</td>
<td>0.7</td>
<td>0.9</td>
<td>0.4</td>
<td>0.5</td>
<td>0.3</td>
</tr>
<tr>
<td>9</td>
<td>0.5</td>
<td>0.2</td>
<td>0.1</td>
<td>0.7</td>
<td>0.8</td>
<td>0.4</td>
<td>0.6</td>
<td>0.8</td>
<td>0</td>
<td>2.8</td>
<td>1.4</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
<td>0.9</td>
<td>1.4</td>
<td>0.9</td>
<td>0.2</td>
<td>0.4</td>
<td>0.6</td>
<td>0.3</td>
<td>0.7</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1.3</td>
<td>0.6</td>
<td>0.3</td>
<td>1.2</td>
<td>1</td>
<td>0.8</td>
<td>1.9</td>
<td>1.6</td>
<td>2.8</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1.9</td>
<td>2.1</td>
<td>4</td>
<td>4.4</td>
<td>3.9</td>
<td>0.7</td>
<td>1.8</td>
<td>2.5</td>
<td>1.2</td>
<td>2.6</td>
<td>1.8</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0.5</td>
<td>0.2</td>
<td>0.3</td>
<td>1.5</td>
<td>0.5</td>
<td>0.4</td>
<td>0.5</td>
<td>0.8</td>
<td>1.4</td>
<td>3.9</td>
<td>0</td>
<td>1.4</td>
<td>1</td>
<td>1.6</td>
<td>1.4</td>
<td>1</td>
<td>1</td>
<td>0.1</td>
<td>0.4</td>
<td>0.6</td>
<td>0.4</td>
<td>1.1</td>
<td>1.3</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0.2</td>
<td>0.1</td>
<td>0.2</td>
<td>0.6</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.7</td>
<td>0.6</td>
<td>0.6</td>
<td>2</td>
<td>1.4</td>
<td>0</td>
<td>1.3</td>
<td>0.7</td>
<td>0.7</td>
<td>0.7</td>
<td>0.6</td>
<td>0.2</td>
<td>0.3</td>
<td>0.4</td>
<td>0.3</td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>0.5</td>
<td>0.3</td>
<td>0.1</td>
<td>0.6</td>
<td>0.2</td>
<td>0.2</td>
<td>0.4</td>
<td>0.6</td>
<td>0.6</td>
<td>1.9</td>
<td>1</td>
<td>1.3</td>
<td>0</td>
<td>0.6</td>
<td>0.7</td>
<td>0.6</td>
<td>0.5</td>
<td>0.1</td>
<td>0.3</td>
<td>0.6</td>
<td>0.6</td>
<td>1.3</td>
<td>0.8</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0.3</td>
<td>0.1</td>
<td>0.1</td>
<td>0.5</td>
<td>0.1</td>
<td>0.1</td>
<td>0.2</td>
<td>0.4</td>
<td>0.6</td>
<td>1.6</td>
<td>0.7</td>
<td>0.6</td>
<td>0</td>
<td>0.6</td>
<td>0.7</td>
<td>0.1</td>
<td>0.3</td>
<td>0.5</td>
<td>0.4</td>
<td>1.2</td>
<td>1.1</td>
<td>0.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>0.5</td>
<td>0.1</td>
<td>0.1</td>
<td>0.5</td>
<td>0.2</td>
<td>0.2</td>
<td>0.5</td>
<td>0.5</td>
<td>1</td>
<td>4</td>
<td>1.4</td>
<td>0.7</td>
<td>0.7</td>
<td>0.1</td>
<td>1.3</td>
<td>0</td>
<td>1.2</td>
<td>1.5</td>
<td>0.2</td>
<td>0.8</td>
<td>1.1</td>
<td>0.8</td>
<td>2.6</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>0.5</td>
<td>0.4</td>
<td>0.2</td>
<td>0.8</td>
<td>0.5</td>
<td>0.9</td>
<td>1.4</td>
<td>2.2</td>
<td>2.1</td>
<td>4.4</td>
<td>1.4</td>
<td>0.7</td>
<td>0.6</td>
<td>0.7</td>
<td>1.2</td>
<td>0</td>
<td>2.8</td>
<td>0.5</td>
<td>1.3</td>
<td>1.6</td>
<td>0.6</td>
<td>1.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>0.4</td>
<td>0.2</td>
<td>0.1</td>
<td>0.5</td>
<td>0.2</td>
<td>0.5</td>
<td>1</td>
<td>1.4</td>
<td>0.9</td>
<td>3.9</td>
<td>1</td>
<td>0.5</td>
<td>0.5</td>
<td>0.7</td>
<td>1.5</td>
<td>2.8</td>
<td>0</td>
<td>0.6</td>
<td>1.7</td>
<td>1.7</td>
<td>0.6</td>
<td>1.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>0.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>0.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>0.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>0.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>0.4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>0.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>0.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Then:

\[
w = w_1 + w_2 + w_3 + w_{st} = (\bar{x} - \bar{x}_{in}) \frac{T}{2k\lambda} + w(\bar{x}_{in})
\]

where, \( s \) is the practical capacity or saturation capacity of the intersection, \( \lambda \) is the green time ratio and \( T \) is the time period for traffic studies (the reasonable time interval of 60 min is used in this study).

The initial temperature, final temperature and the cooling rate for SA algorithm were 0.05, 0.001 and 0.97, respectively. At each temperature level 20 iteration of SA was solved. Table 2 shows the results of applying the SA algorithm to the network. It is necessary to say that SA parameters were chosen after sensitivity analysis. Also, 2580 iterations of SA shown in Fig. 2 were solved in order to find the optimal solution of Sioux Falls network. In this study the algorithm solved by using C++ programming and a computer with CPU of 2.5 GHz and the RAM of 4 Giga bytes. After 2580 iterations, the best objective function was received to 4.1213. Table 3 shows the results of best link flows and delays after signal optimization.

The use of the proposed algorithms reduced the total network travel time by 5.96%. As a note, implementing the GP algorithm and the dynamic memory in the coding process, this bi-level network design problem was solved in 111.26 sec by a GPA with a convergence rate of 0.0001.
Fig. 2: Change of objective function through 2580 iterations

Table 3: The best link flows and delays after signal optimization

<table>
<thead>
<tr>
<th>Origin</th>
<th>Destination</th>
<th>Travel time</th>
<th>Flow</th>
<th>Origin</th>
<th>Destination</th>
<th>Travel time</th>
<th>Flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>0.06</td>
<td>0.895</td>
<td>14</td>
<td>11</td>
<td>0.058</td>
<td>1.102</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>0.04</td>
<td>0.664</td>
<td>14</td>
<td>15</td>
<td>0.053</td>
<td>0.679</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.06</td>
<td>0.344</td>
<td>14</td>
<td>23</td>
<td>0.04</td>
<td>1.137</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>0.021</td>
<td>0.913</td>
<td>15</td>
<td>25</td>
<td>0.06</td>
<td>2.002</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0.04</td>
<td>0.973</td>
<td>15</td>
<td>14</td>
<td>0.078</td>
<td>0.335</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>0.041</td>
<td>0.903</td>
<td>15</td>
<td>19</td>
<td>0.087</td>
<td>2.079</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>0.04</td>
<td>0.84</td>
<td>15</td>
<td>22</td>
<td>0.03</td>
<td>2.504</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0.04</td>
<td>1.013</td>
<td>16</td>
<td>8</td>
<td>0.052</td>
<td>1.282</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>0.021</td>
<td>1.212</td>
<td>16</td>
<td>10</td>
<td>0.112</td>
<td>0.9</td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>0.002</td>
<td>0.906</td>
<td>16</td>
<td>17</td>
<td>0.03</td>
<td>1.548</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>0.021</td>
<td>1.808</td>
<td>16</td>
<td>18</td>
<td>0.03</td>
<td>1.128</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>0.107</td>
<td>0.357</td>
<td>17</td>
<td>25</td>
<td>0.08</td>
<td>1.121</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>0.052</td>
<td>1.378</td>
<td>17</td>
<td>15</td>
<td>0.112</td>
<td>0.523</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>0.05</td>
<td>0.696</td>
<td>17</td>
<td>19</td>
<td>0.027</td>
<td>1.086</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>0.041</td>
<td>1.39</td>
<td>18</td>
<td>7</td>
<td>0.02</td>
<td>1.853</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>0.023</td>
<td>1.234</td>
<td>18</td>
<td>16</td>
<td>0.03</td>
<td>1.918</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>0.037</td>
<td>1.402</td>
<td>18</td>
<td>20</td>
<td>0.04</td>
<td>1.69</td>
</tr>
<tr>
<td>7</td>
<td>18</td>
<td>0.02</td>
<td>2.049</td>
<td>19</td>
<td>15</td>
<td>0.033</td>
<td>2.382</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>0.03</td>
<td>1.967</td>
<td>19</td>
<td>17</td>
<td>0.022</td>
<td>1.193</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>0.03</td>
<td>1.596</td>
<td>19</td>
<td>20</td>
<td>0.04</td>
<td>1.386</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>0.103</td>
<td>0.355</td>
<td>20</td>
<td>18</td>
<td>0.04</td>
<td>2.292</td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>0.08</td>
<td>0.202</td>
<td>20</td>
<td>10</td>
<td>0.041</td>
<td>1.202</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>0.056</td>
<td>0.933</td>
<td>20</td>
<td>21</td>
<td>0.06</td>
<td>0.672</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>0.103</td>
<td>0.228</td>
<td>20</td>
<td>22</td>
<td>0.05</td>
<td>0.788</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>0.031</td>
<td>2.461</td>
<td>21</td>
<td>20</td>
<td>0.06</td>
<td>0.806</td>
</tr>
<tr>
<td>10</td>
<td>9</td>
<td>0.032</td>
<td>1.886</td>
<td>21</td>
<td>22</td>
<td>0.02</td>
<td>1.574</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>0.053</td>
<td>2.125</td>
<td>21</td>
<td>24</td>
<td>0.03</td>
<td>1.751</td>
</tr>
<tr>
<td>10</td>
<td>16</td>
<td>0.067</td>
<td>2.193</td>
<td>22</td>
<td>15</td>
<td>0.032</td>
<td>2.495</td>
</tr>
<tr>
<td>10</td>
<td>17</td>
<td>0.097</td>
<td>0.621</td>
<td>22</td>
<td>20</td>
<td>0.05</td>
<td>1.031</td>
</tr>
<tr>
<td>10</td>
<td>25</td>
<td>0</td>
<td>2.008</td>
<td>22</td>
<td>21</td>
<td>0.02</td>
<td>1.584</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>0.074</td>
<td>0.333</td>
<td>22</td>
<td>22</td>
<td>0.04</td>
<td>1.232</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>0.057</td>
<td>1.748</td>
<td>22</td>
<td>14</td>
<td>0.041</td>
<td>0.891</td>
</tr>
<tr>
<td>11</td>
<td>12</td>
<td>0.06</td>
<td>1.196</td>
<td>23</td>
<td>22</td>
<td>0.04</td>
<td>1.444</td>
</tr>
</tbody>
</table>
Table 3: Continued

<table>
<thead>
<tr>
<th>Origin</th>
<th>Destination</th>
<th>Travel time</th>
<th>Flow</th>
<th>Origin</th>
<th>Destination</th>
<th>Travel time</th>
<th>Flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>14</td>
<td>0.042</td>
<td>1.688</td>
<td>23</td>
<td>24</td>
<td>0.03</td>
<td>0.685</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>0.04</td>
<td>1.119</td>
<td>24</td>
<td>23</td>
<td>0.04</td>
<td>1.590</td>
</tr>
<tr>
<td>12</td>
<td>11</td>
<td>0.066</td>
<td>0.952</td>
<td>24</td>
<td>21</td>
<td>0.03</td>
<td>1.074</td>
</tr>
<tr>
<td>12</td>
<td>13</td>
<td>0.03</td>
<td>1.599</td>
<td>24</td>
<td>23</td>
<td>0.02</td>
<td>0.691</td>
</tr>
<tr>
<td>13</td>
<td>12</td>
<td>0.03</td>
<td>1.546</td>
<td>25</td>
<td>10</td>
<td>0.002</td>
<td>3.723</td>
</tr>
<tr>
<td>13</td>
<td>24</td>
<td>0.04</td>
<td>1.655</td>
<td>25</td>
<td>15</td>
<td>0.06</td>
<td>2.008</td>
</tr>
</tbody>
</table>

CONCLUSION

Traffic management strategies are among the most important and common congestion relief strategies nowadays. In this paper, a bi-level programming of traffic signal timing was proposed. The model optimizes the split of the traffic signal, the green time, of the network. In order to solve the NP-Complete problem the Simulated Annealing (SA) was used. As the number of traffic assignment which was required for solving the master problem using SA was very high, a path-based Gradient Projection method was implemented. By reducing the number of path enumeration procedures needed for the UE traffic assignment, the run times was substantially reduced. In continuing of this study, the simultaneous use of traffic signal timing and capacity improvement and optimizing traffic signals with micro-simulation software were proposed.

REFERENCES


