Comparing algorithms for solving the Local Optimisation of the Signal Settings (LOSS) problem under different supply and demand configurations

Mariano Gallo\textsuperscript{a,}\textsuperscript{*}, Luca D’Acierno\textsuperscript{b}

\textsuperscript{a}University of Sannio, Department of Engineering, piazza Roma 21, Benevento 82100, Italy
\textsuperscript{b}‘Federico II’ University of Naples, Department of Civil, Architectural and Environmental Engineering, via Claudio 21, Naples 80125, Italy

Abstract

In this paper we study the Local Optimisation of Signal Settings (LOSS) problem that arises when signal control parameters of an urban road network are locally optimised and have to be consistent with equilibrium traffic flows. This problem can be formulated with an (asymmetric) equilibrium assignment model. In this paper we study the problem and the corresponding equilibrium model and compare several solution algorithms proposed in the literature. All algorithms were tested on a real-scale network with different demand levels and a different number of signalised intersections. Numerical results show that differences in computing times among algorithms are significant when the network is highly congested and there are many signalised intersections.

Keywords: local optimisation of signal settings; network design; traffic lights

1. Introduction

Traffic-light systems play a fundamental role in urban network management and effective optimisation of signal settings may produce significant improvements in network performance. The problem of optimising signal settings is a particular case of the more general Equilibrium Network Design Problem (ENDP), where signal settings assume the role of decision variables; this problem is also known as the Signal Setting Design Problem and for solving it two different approaches can be identified (Cascetta et al., 2006): a global approach and a local approach.

\textsuperscript{*} Corresponding author. Tel.: +39-0824-305565; fax: +39-0824-325246.
\textit{E-mail address:} gallo@unisannio.it
In the first case, the problem is actually an ENDP, formulated with a (non-linear constrained) optimisation model, and is also known as Global Optimisation of Signal Settings (GOSS). In the second case, instead, it is assumed that the signal settings of each junction are designed so as to minimise only the total delay at the same junction according to a specific local control policy. This problem is known also as Local Optimisation of Signal Settings (LOSS) and is the focus of this paper.

The distinction between the global and local approach can be found in Marcotte (1983), Fisk (1984), Cantarella et al. (1991), Cantarella and Sforza (1995), and Cascetta et al. (1999, 2006).

The LOSS problem can be formulated as a fixed/point problem, where we have to search for equilibrium traffic flows congruent with link costs and signal settings, and the signal settings are obtained according to a local control policy. This problem was studied, amongst others, by Allsop (1977), Smith (1979a, 1979b), Dafermos (1980, 1982), Fisk and Nguyen (1982), Florian and Spiess (1982), Gartner (1983), Meneguzzo (1990, 1995), Cantarella and Improta (1991), Smith and Van Vuren (1993), Al-Malik and Gartner (1995), Lee and Hazelton (1996), Cascetta et al. (1999, 2006), and D’Acierno et al. (2012). The above authors examined the theoretical properties of the problem and/or proposed several solution algorithms.


In this paper the LOSS problem, known also as the combined assignment-control problem, is studied by formulating an asymmetric equilibrium model, proposing some solution algorithms and testing them on a real-scale network. Even if the GOSS approach leads to better solutions with respect to the LOSS, the latter allows the problem to be solved in a much lower computing times, giving a solution not so far from the global optimum obtained by solving the GOSS. Indeed, in a comparison between LOSS and GOSS solutions on a real-scale network (Cascetta et al., 2006) the GOSS vs. LOSS improvements in objective function values were always under 6% while the computing time decreased from about 2 hours (GOSS) to about 2 minutes (LOSS). Due to such reasons and the greater affordability of the LOSS problem, which can also be solved with commercial software by updating the signal settings after the traffic equilibrium solution with a so-called “external approach”, the LOSS can be more suitable for the problem to be solved by the traffic departments of small-medium towns.

The focus of the paper is mainly devoted to numerical tests that concern several scenarios corresponding to different demand levels and different numbers of signalised intersections, in order to examine when this approach to signal settings design is actually useful with respect to designing each intersection independently of one another. Compared with the previous papers proposed by the authors, here we test a real-scale case based on real data obtained with specific demand and supply surveys performed while drawing up the Urban Traffic Plan of the City of Benevento. Moreover, in addition to the real case, with the actual number of signalised intersections and the actual level of demand, we test the proposed algorithms also for the assumptions of more signalised intersections, appropriately chosen where useful, and for different demand levels, in order to test the advantages of solving the LOSS problem in more and less congested cases. The contribution to the literature lies in the utility of the paper for those wishing to apply the LOSS to a real case for designing the signal settings: our findings may be used to ascertain whether it would be useful to adopt the method, depending on the level of demand and the number of signalised intersections.

The paper is organised as follows: Section 2 formulates the model; the solution algorithms are described in Section 3; numerical results on a real-scale network are summarised in Section 4; Sections 5 concludes the paper and identifies possible research prospects.
2. Mathematical model

On a road network, the link flow vector, \( f \), can be estimated as follows:

\[
f(c) = A \ P (A^T \ c) \ d
\]

where:
- \( c \) is the link cost vector;
- \( A \) is the link-path incidence matrix, whose components, \( a_{lp} \), are equal to 1 if link \( l \) belongs to path \( p \) and 0 otherwise;
- \( P \) the path choice probability matrix, with a column for each OD pair and a row for each path \( p \); the generic element, \( P_{p,od} \), of this matrix represents the probability that a user will use path \( p \) from \( o \) to \( d \);
- \( d \) is the demand vector, whose components are the demand values \( d_{od} \) for each O-D pair.

Assuming that the demand, \( d \), and network topology, summarised in \( A \), are invariable the previous equation may be simplified as follows:

\[
f = f(c)
\] (1)

Link costs, \( c \), depend on traffic flows, \( f \), and signal settings, \( g \), if we consider the latter as variables in the problem:

\[
c = c(f; g)
\] (2)

Substituting eqn (2) in eqn (1) we obtain:

\[
f = f(c(f; g))
\] (3)

which relates flows, \( f \), costs, \( c \), and signal settings, \( g \).

Having fixed the vector \( g \), we can calculate the equilibrium traffic flows, \( f^* \), with one of the algorithms available in the literature (see for instance Cantarella, 1997). Formally, we can write:

\[
f^* = f(c(f^*; g))
\] (4)

In this case, since the transportation supply is given, it may be stated, under some assumptions, that the solution of the fixed-point problem (4) exists and is unique (Cantarella, 1997). Therefore, eqn (4) is an application: one, and only one, equilibrium traffic flow vector, \( f^* \), corresponds to each signal settings vector, \( g \).

Now, we assume that the parameters of each signalised intersection are optimised according to a local control policy, that we can formally express as:

\[
g = g(f^*)
\] (5)

Under this assumption, the following fixed-point mathematical model can be formulated (see also Cascetta et al., 2006):

\[
f^* = f(c(f^*; g(f^*)))
\] (6)

The local control policy, \( g(.) \), relates the signal settings, \( g \), to traffic flows, \( f \); in detail, the signal settings of each intersection of the network are locally optimised according only to the flows approaching the intersection.
Several control policies can be adopted: those most widely used are the Webster (1958) method, SIGSET (Allsop, 1971), SIGCAP (Allsop, 1976), and Smith’s $P_0$ policy (Smith, 1980, 1981).

In terms of theoretical properties, the link cost functions are non-separable, since at each intersection the cost of a link depends on the flows of all concurring links (the control policy recalculates signal settings as a function of all flows at the intersection). Therefore, the Jacobian is not positive definite and the uniqueness of the fixed-point solution cannot be stated (Charlesworth, 1977, showed that more than one equilibrium solution can be found). Instead, the existence of a solution is ensured by the continuity of the functions (a condition that is satisfied for stochastic route choice models, continuous cost-flow functions and continuous local control policy functions). The theoretical properties have received extensive attention from Smith (1979a), Meneguzzer (1990, 1995), Al-Malik and Gartner (1995) and Cascetta et al. (2006).

3. Solution algorithms

For solving the fixed-point (1) we propose three algorithms based on an MSA framework (Powell and Sheffi, 1982; Sheffi and Powell, 1982). The MSA (Method of Successive Averages) is widely used for solving traffic assignment problems. For solving the traffic assignment problem three MSA algorithms are available: the MSA-FA (Flow Averaging), which is the original version proposed by Sheffi and Powell (1982); the MSA-CA (Cost Averaging), which was proposed by Cantarella (1997); and, the MSA-ACO (Ant Colony Optimisation), which was proposed by D’Acierno et al. (2006). Even if these algorithms could also be used under the assumptions of deterministic route choice models, in the following we refer only to stochastic route choice models, that work better for simulating user behaviour on urban networks.

All these algorithms are based on recursive equations and will stop when the link traffic flows are equal (in practice, a stop threshold is used) to the uncongested network loading traffic flows. The MSA-FA averages at each iteration the uncongested link flows, $f_{SUN}^k$, with the results of the previous iteration $f^{k-1}$, as follows:

\[
k = k + 1 \\
c^k = c(f^{k-1}) \\
f_{SUN}^k = f_{SUN}(c^k) \\
f^k = f^{k-1} + 1/k (f_{SUN}^k - f^{k-1})
\]

Cascetta et al. (2006) proposed an algorithm for solving the fixed-point problem (1) based on the MSA-FA; this algorithm updates, according to local control policies, the signal settings at each iteration, as follows:

\[
k = k + 1 \\
c^k = c(f^{k-1}, g^{k-1}) \\
f_{SUN}^k = f_{SUN}(c^k) \\
f^k = f^{k-1} + 1/k (f_{SUN}^k - f^{k-1}) \\
g^k = g(f^k)
\]

The MSA-CA is based on the same general framework of the MSA-FA but the costs, instead of flows, are averaged, as follows:

\[
k = k + 1 \\
f^k = f_{SUN}(c^{k-1}) \\
y^k = c(f^k) \\
c^k = c^{k-1} + 1/k (y^k - c^{k-1})
\]
Also this algorithm can be modified in order to solve the fixed point problem (1), as follows (see D’Acierno et al., 2012):

\[
\begin{align*}
  k & = k + 1 \\
  f^k & = f_{\text{SUN}}(c^{k-1}) \\
  g^k & = g(f^k) \\
  y^k & = c(f^k, g^k) \\
  c^k & = c^{k-1} + 1/k (y^k - c^{k-1}) \\
\end{align*}
\]

Finally, D’Acierno et al. (2006) proposed the MSA-ACO algorithm (see Appendix for details), based on Ant Colony Optimisation (Dorigo, 1992; Dorigo and Stützle, 2004), which uses the general framework of the MSA as follows:

\[
\begin{align*}
  k & = k + 1 \\
  c^k & = c(f^{k-1}) \\
  \Delta \tau^k & = \tau(c^k) \\
  \tau^k & = \tau^{k-1} + 1/k (\Delta \tau^k - \tau^{k-1}) \\
  f^k & = f_{\text{SUN}}(\tau^k) \\
\end{align*}
\]

where \( \tau^k \) and \( \Delta \tau^k \) represent, respectively, the pheromone trail and the related increase at iteration \( k \). This algorithm was adapted to solve the fixed point problem (1) by D’Acierno et al. (2012) as follows:

\[
\begin{align*}
  k & = k + 1 \\
  c^k & = c(f^{k-1}, g^{k-1}) \\
  \Delta \tau^k & = \tau(c^k) \\
  \tau^k & = \tau^{k-1} + 1/k (\Delta \tau^k - \tau^{k-1}) \\
  f^k & = f_{\text{SUN}}(\tau^k) \\
  g^k & = g(f^k) \\
\end{align*}
\]

These three algorithms are tested on a real case in the following section in order to compare their performance in different scenarios with different demand levels and number of signalised intersections.

4. Numerical results

The model and algorithms were tested on the urban network of Benevento, a town in the south of Italy with about 61,000 inhabitants. The transportation model (demand and supply) was built during the design of the Urban Traffic Plan of the town. The network graph has 1,577 oriented links, which represents about 216 kms of roads, and 678 nodes. The zoning of the study area is very dense, with 66 internal zones; the cordon sections are 14, so the total centroids are 80 (66 internal and 14 external). Figure 1 shows the graph of the network: different colours for links and nodes indicate different kinds of roads and intersections.

The OD matrix was estimated by a system of random utility models (see Cascetta, 2009) calibrated for other urban networks and adapted to the specific case; the OD matrix thereby calculated was improved with a correction procedure using traffic surveys (see Cascetta, 2009): traffic data were collected at 139 count sections, giving a good coverage of the network (see Figure 2).

The network has only eight signalised intersections and the demand level is not high. In this paper, in order to test and compare the algorithms, we generate 35 different scenarios, considering seven different demand levels and five different supply models. Over the available OD matrix, six other matrices were generated, multiplying
all cells of the OD matrix by the following factors: 0.8, 1.0 (real case), 1.2, 1.4, 1.6, 1.8 and 2.0. The seven matrices are identified with OD08, OD10 (real case), OD12, OD14, OD16, OD18 and OD20, respectively. Moreover, we increased the number of signalised intersections, introducing into the model other traffic lights where the traffic flows were higher. We thus tested five networks with: 8 (real case), 18, 28, 38 and 48 signalised intersections, the networks being labelled SIG08, SIG18, SIG28, SIG38 and SIG48, respectively.

It is useful to point out that the actual supply configuration of the network has only eight signalised intersections (real case) and that the other supply configurations, with more traffic-lights, are “created” only in order to test how the proposed algorithms perform in different cases.

The cost functions adopted were the Festa and Nuzzolo (1990) and Doherty (1977) for determining respectively running and waiting times in urban contexts.

In the tests we adopted the Webster (1958) method as the local control policy, but other policies could be adopted in the same algorithms without loss of generality. Indeed, several local control policies can lead to different solutions and hence to different values of the objective functions. A comparison between the different local control policies needs a dedicated paper and will be subject of further research by the authors.

The three algorithms, MSA-FA, MSA-CA and MSA-ACO, were implemented in Visual Basic code and all tests were conducted using a PC Intel Core i7-2600 (3.40 GHz).

The three algorithms are tested for all 35 scenarios; Table 1 reports the number of iterations and the corresponding computing times. The comparison shows that MSA-ACO and MSA-CA algorithms perform better than MSA-FA for almost all scenarios. Between MSA-ACO and MSA-CA the differences are less substantial, although MSA-ACO seems to work slightly better. The differences between algorithms are very significant when the demand is high (OD18 and, above all, OD20) and when the signalised intersections are numerous. In three scenarios, MSA-FA algorithm does not converge in acceptable computing times: the algorithm is stopped after 100,000 iterations but the convergence test always tends to decrease. Therefore the algorithm does not reach the fixed error threshold in the maximum prefixed number of iterations since the decreasing trend is too slow but tends to converge also in these cases. Examining in detail the iterations of the algorithm in these cases, we noted that there is only one signalised intersection whose traffic flows (on 4 links) are very slowly convergent.

Moreover, it can be noted that in several cases the number of iterations for convergence does not present monotonicity with respect to demand level and/or number of signalised intersections. Indeed, there are no theoretical reasons that imply monotonicity properties, since convergence can be reached before or after independently of congestion levels and/or signalised junctions.

Figures 3 and 4 show the convergence of the algorithms in scenarios SIG48-OD16 and SIG48-OD20; in the diagrams only the first 200 iterations are represented.

These results suggest that MSA-FA should not be used when the network is highly congested, in particular if the signalised intersections are numerous, while MSA-CA and MSA-ACO always seem to be applicable.

The reduction in computing times of MSA-CA and MSA-ACO vis-à-vis MSA-FA is also significant for the other scenarios where MSA-FA converges in some tens of iterations. In particular, it is very important when the combined assignment-control problem is used as a subroutine of an algorithm for solving the Urban Network Design Problem (UNDP). In this case, indeed, the LOSS problem has to be solved at each iteration of the algorithm for solving the UNDP. Usually, the corresponding number of iterations is huge: saving about 90 % of the time concerned (see Table 1) for each iteration can significantly change the total time for solving the problem.

Another important point to test concerns the utility of solving the combined assignment-control problem for designing the signal settings, with respect to designing them without updating traffic flows until convergence. Indeed, even if the LOSS problem cannot be assumed as an actual network design problem, it can be used for designing the signal settings: the LOSS solution will be a feasible (but sub-optimal) solution of the more complex (and time consuming) GOSS problem. This procedure is often used inside urban network design solution algorithms (see for instance Gallo et al., 2010) for updating signal settings during topological interventions.
Fig. 1. Supply model
Fig. 2. Count sections
Table 1. Comparison among MSA-FA, MSA-CA and MSA-ACO in terms of iterations and computing times

<table>
<thead>
<tr>
<th></th>
<th>OD08</th>
<th>OD10</th>
<th>OD12</th>
<th>OD14</th>
<th>OD16</th>
<th>OD18</th>
<th>OD20</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Iterations</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIG08</td>
<td>7</td>
<td>16</td>
<td>31</td>
<td>49</td>
<td>51</td>
<td>110</td>
<td>62</td>
</tr>
<tr>
<td>SIG18</td>
<td>7</td>
<td>16</td>
<td>32</td>
<td>49</td>
<td>51</td>
<td>131</td>
<td>59</td>
</tr>
<tr>
<td>SIG28</td>
<td>7</td>
<td>12</td>
<td>28</td>
<td>46</td>
<td>49</td>
<td>137</td>
<td>&gt;100,000</td>
</tr>
<tr>
<td>SIG38</td>
<td>6</td>
<td>9</td>
<td>28</td>
<td>46</td>
<td>50</td>
<td>137</td>
<td>&gt;100,000</td>
</tr>
<tr>
<td>SIG48</td>
<td>6</td>
<td>9</td>
<td>28</td>
<td>45</td>
<td>50</td>
<td>83</td>
<td>&gt;100,000</td>
</tr>
<tr>
<td>SIG08</td>
<td>7</td>
<td>8</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>SIG18</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>11</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>SIG28</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>9</td>
<td>10</td>
<td>13</td>
</tr>
<tr>
<td>SIG38</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>10</td>
<td>13</td>
</tr>
<tr>
<td>SIG48</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>SIG08</td>
<td>4</td>
<td>7</td>
<td>12</td>
<td>11</td>
<td>8</td>
<td>14</td>
<td>13</td>
</tr>
<tr>
<td>SIG18</td>
<td>4</td>
<td>7</td>
<td>12</td>
<td>16</td>
<td>7</td>
<td>15</td>
<td>12</td>
</tr>
<tr>
<td>SIG28</td>
<td>5</td>
<td>7</td>
<td>12</td>
<td>16</td>
<td>11</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>SIG38</td>
<td>5</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>SIG48</td>
<td>5</td>
<td>7</td>
<td>8</td>
<td>14</td>
<td>8</td>
<td>14</td>
<td>13</td>
</tr>
<tr>
<td><strong>Computing times (s)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIG08</td>
<td>12</td>
<td>25</td>
<td>47</td>
<td>72</td>
<td>94</td>
<td>160</td>
<td>87</td>
</tr>
<tr>
<td>SIG18</td>
<td>11</td>
<td>25</td>
<td>48</td>
<td>73</td>
<td>95</td>
<td>191</td>
<td>84</td>
</tr>
<tr>
<td>SIG28</td>
<td>13</td>
<td>19</td>
<td>42</td>
<td>68</td>
<td>91</td>
<td>209</td>
<td>&gt;150,000</td>
</tr>
<tr>
<td>SIG38</td>
<td>14</td>
<td>15</td>
<td>42</td>
<td>68</td>
<td>92</td>
<td>199</td>
<td>&gt;150,000</td>
</tr>
<tr>
<td>SIG48</td>
<td>13</td>
<td>15</td>
<td>42</td>
<td>67</td>
<td>91</td>
<td>121</td>
<td>&gt;150,000</td>
</tr>
<tr>
<td>SIG08</td>
<td>11</td>
<td>13</td>
<td>12</td>
<td>14</td>
<td>23</td>
<td>23</td>
<td>23</td>
</tr>
<tr>
<td>SIG18</td>
<td>12</td>
<td>13</td>
<td>13</td>
<td>15</td>
<td>22</td>
<td>22</td>
<td>23</td>
</tr>
<tr>
<td>SIG28</td>
<td>12</td>
<td>14</td>
<td>14</td>
<td>15</td>
<td>19</td>
<td>16</td>
<td>20</td>
</tr>
<tr>
<td>SIG38</td>
<td>16</td>
<td>12</td>
<td>12</td>
<td>15</td>
<td>25</td>
<td>17</td>
<td>21</td>
</tr>
<tr>
<td>SIG48</td>
<td>15</td>
<td>12</td>
<td>12</td>
<td>15</td>
<td>22</td>
<td>17</td>
<td>23</td>
</tr>
<tr>
<td>SIG08</td>
<td>8</td>
<td>13</td>
<td>19</td>
<td>18</td>
<td>13</td>
<td>22</td>
<td>20</td>
</tr>
<tr>
<td>SIG18</td>
<td>8</td>
<td>14</td>
<td>19</td>
<td>25</td>
<td>11</td>
<td>24</td>
<td>19</td>
</tr>
<tr>
<td>SIG28</td>
<td>9</td>
<td>12</td>
<td>19</td>
<td>24</td>
<td>17</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>SIG38</td>
<td>9</td>
<td>12</td>
<td>13</td>
<td>15</td>
<td>13</td>
<td>14</td>
<td>6</td>
</tr>
<tr>
<td>SIG48</td>
<td>9</td>
<td>13</td>
<td>14</td>
<td>21</td>
<td>13</td>
<td>21</td>
<td>21</td>
</tr>
</tbody>
</table>
Fig. 3. Convergence of algorithms for the scenario SIG48-OD16

Fig. 4. Convergence of algorithms for the scenario SIG48-OD20
Table 2. Comparison among starting and final solution in terms of total travel times

<table>
<thead>
<tr>
<th></th>
<th>OD08</th>
<th>OD10</th>
<th>OD12</th>
<th>OD14</th>
<th>OD16</th>
<th>OD18</th>
<th>OD20</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIG08</td>
<td>110,456</td>
<td>158,614</td>
<td>248,457</td>
<td>410,218</td>
<td>662,313</td>
<td>1,023,507</td>
<td>1,521,819</td>
</tr>
<tr>
<td>SIG18</td>
<td>110,412</td>
<td>156,502</td>
<td>242,116</td>
<td>395,744</td>
<td>638,237</td>
<td>984,243</td>
<td>1,468,881</td>
</tr>
<tr>
<td>SIG28</td>
<td>109,412</td>
<td>148,394</td>
<td>226,692</td>
<td>366,471</td>
<td>591,821</td>
<td>919,602</td>
<td>1,385,807</td>
</tr>
<tr>
<td>SIG38</td>
<td>105,518</td>
<td>148,959</td>
<td>228,788</td>
<td>369,840</td>
<td>593,438</td>
<td>915,727</td>
<td>1,377,422</td>
</tr>
<tr>
<td>SIG48</td>
<td>104,681</td>
<td>148,959</td>
<td>228,788</td>
<td>369,840</td>
<td>593,438</td>
<td>915,727</td>
<td>1,377,422</td>
</tr>
<tr>
<td>SIG08</td>
<td>109,968</td>
<td>152,151</td>
<td>224,507</td>
<td>354,985</td>
<td>567,568</td>
<td>876,225</td>
<td>1,301,436</td>
</tr>
<tr>
<td>SIG18</td>
<td>108,943</td>
<td>150,127</td>
<td>219,523</td>
<td>342,824</td>
<td>545,542</td>
<td>838,175</td>
<td>1,247,938</td>
</tr>
<tr>
<td>SIG28</td>
<td>108,943</td>
<td>144,309</td>
<td>206,360</td>
<td>320,056</td>
<td>510,916</td>
<td>790,713</td>
<td>1,189,227</td>
</tr>
<tr>
<td>SIG38</td>
<td>105,517</td>
<td>144,180</td>
<td>206,678</td>
<td>318,313</td>
<td>502,405</td>
<td>770,677</td>
<td>1,159,057</td>
</tr>
<tr>
<td>SIG48</td>
<td>104,707</td>
<td>143,153</td>
<td>204,220</td>
<td>312,260</td>
<td>492,506</td>
<td>755,738</td>
<td>1,134,521</td>
</tr>
</tbody>
</table>

Assuming total travel time as a performance index of the network, we compare, for all 35 scenarios, the solutions obtained by solving the combined assignment-control problem with the solution that can be obtained by applying the local control policy without updating flows and signal settings until convergence. In Table 2 the total travel times on the network are compared; we report only the results obtained with MSA-CA algorithms, since the results obtained with the other MSA algorithms are similar (a slight difference in total travel times is produced by the approximation due to the stop threshold but the final solutions are in practice the same). The results (see Table 2) show that great advantages of applying the methodology are obtained when the network is very congested and the signalised intersections are numerous; in this case, travel time reduction may be as much as 17%. Instead, if there is little congestion on the network, the improvements are negligible and the results obtained without updating traffic flows and signal settings are not significantly different with respect to solving the combined assignment-control problem. Note that in most cases the percent reduction of total travel time is almost insensitive to the number of signalised intersections; only for high demand levels can a sensitive variation be found. This is probably due to the fact that the scenarios with more signalised intersections have objective function values which are already lower than those of the corresponding scenario with fewer signalised intersections. Indeed, the starting solution of each scenario assumes signal settings equal to those calculated with the Webster method, assuming equilibrium traffic flows resulting from an assignment with constant signal settings (equally distributed between approaches). Therefore, when we introduce more signalised intersections they are “almost” optimised according to the traffic flows. Assuming this starting solution is more correct than assuming a starting solution where the signal settings are equally distributed between approaches, since if we do
not use the LOSS for designing the parameters at least we have to design the parameters at each intersection according to current traffic flows.

5. Conclusions and research prospects

This paper focused on the Local Optimisation of Signal Settings (LOSS) problem that arises when we assume that the signal settings of each junction are designed so as to minimise only the total delay at the same junction, according to a specific local control policy. This problem, known also as the combined assignment-control problem, can be formulated with a fixed-point model and solved with some algorithms based on an MSA framework. In this paper we tested three different algorithms on 35 scenarios based on a real case.

Our numerical results showed that MSA-CA and MSA-ACO, modified in order to solve the LOSS problem, are more efficient than the MSA-FA in terms of computing times, mainly if the network is highly congested and there are many signalised intersections. Moreover, other results show that solving the LOSS problem for designing signal settings is actually useful only if the network is highly congested.

Future research could involve the testing of other local control policies inside the same algorithms. Other algorithms could be proposed to solve the problem for reducing computing times further, mainly if the combined assignment-control problem is used within methods and algorithms for solving the (topological) Urban Network Design Problem.

Acknowledgements

Partially supported by the Italian MIUR under PRIN2009 grant no. 2009EP3S42_002.

References


Appendix A. The Ant Colony Optimisation (ACO) approach

The aim of this appendix is to provide a short description concerning algorithms based on Ant Colony Optimisation (ACO) in order to enhance the reader’s familiarity with the topic.

Since the solution of the asymmetric assignment problem requires more computing time than a classic (symmetric) assignment problem due to the fact that signal settings have to be updated at each iteration and there is a double circular dependence (local optimal signal settings depend on flows that, in turn, depend on costs, that depend on both flows and signal settings), our proposal is to steer our research into ACO-based, i.e. an algorithm based on the food source search of ant colonies, which have in many cases shown their efficiency in terms of calculation times (see for instance Dorigo, 1992; Dorigo & Stützle, 2004; D’Acierno et al., 2006).

In order to understand the underlying principles of this nature-inspired approach, it is first necessary to describe the behaviour of real ants in the search for food sources which is based on the following simple rules: each ant provides a pheromone trail along its path; each ant follows a path if there is a pheromone trail; if there is no pheromone trail, ants choose their paths randomly; there is evaporation of pheromone trails, which causes short paths to have a more intense pheromone trail; if there is a diversion point (i.e. where different paths start), ants follow the path with the most intense pheromone trail. Hence, if a path is blocked by an obstacle, ants initially choose their path randomly. Thus, due to evaporation, they choose to follow the path with the most intense pheromone trail, which is the shortest path.

The abilities of real ants have been used to develop suitable algorithms that model some ant features integrated with other properties, allowing these new kinds of ants, termed artificial ants, to solve many problems. In particular, a generic meta-heuristic algorithm based on Ant Colony Optimisation can be formalised as:

```
procedure ACO_Algorithm()
    while (termination_criterion_not satisfied)
        schedule_activities
            ant_generation_and_activity();
            pheromone_evaporation();
            demon_actions(); {optional}
        end_schedule_activities
    end while
end procedure
```

The first phase, indicated as ant_generation_and_activity, can be described by means of three tasks: ant generation, transition probability calculation and the pheromone trail increase.

Ant generation consists in defining the number of colonies and/or the number of ants per colony in order to perform the algorithm optimally.

The transition probability expresses the probability of choosing a path as a function of the intensity of pheromone trails and the visibility term (i.e. a term which expresses a sort of distance that could affect probability choices), that is:

\[
p_p^k(l \mid i) = \left(\tau_p^l\right)^\alpha \cdot \left(\eta_p^l\right)^\beta \cdot \sum_{l' \in FS(i)} \left(\tau_p^{l'}\right)^\alpha \cdot \left(\eta_p^{l'}\right)^\beta
\]

where \( p_p^k(l \mid i) \) is the probability of choosing link \( l \), with \( l=(i,j) \), at diversion node \( i \); \( \tau_p^k \) is the intensity of the pheromone trail on link \( l \) at iteration \( k \); \( \eta_p^k \) is the visibility term on link \( l \) at iteration \( k \); \( FS(i) \) is the set of links belonging to the forward star of node \( i \); \( \alpha \) and \( \beta \) are model parameters which express the relevance respectively of the pheromone trail and visibility term in ant choices.
The pheromone trail increase consists in defining the quantity of pheromone produced by each ant on the generic link \( l \), that is:
\[
\Delta \tau^k_l = \lambda^i_l(p^k, X)
\]
where \( \Delta \tau^k_l \) is the pheromone trail produced by each ant on link \( l \) at iteration \( k \); \( p^k \) is the transition probability matrix at iteration \( k \) whose generic element is \( p^k(l|i) \); \( X \) is the matrix of model parameters; \( \lambda^i_l \) is the function that expresses the increase in the pheromone trail on link \( l \) depending on \( p^k \) and \( X \).

Pheromone evaporation is a task that expresses how the pheromone trail evaporates. In general, this task is performed jointly with the calculation of the increase in pheromone due to contributions of each ant, and is generally formulated as:
\[
\tau^k_i = (1 - \rho) \cdot \tau^{(k-1)}_i + \rho \cdot \Delta \tau^k_i \quad \text{or} \quad \tau^k_i = (1 - \rho) \cdot \tau^{(k-1)}_i + \Delta \tau^k_i
\]
where \( \rho \) is the evaporation term.

Finally, the demon action is a generally optional task that introduces some local optimisation procedures in order to improve algorithm performance. However, algorithms proposed for solving the (symmetric and/or asymmetric) traffic assignment problem do not adopt this task.

Adopting an ACO-based paradigm, D’Acierno et al. (2006) proposed an algorithm for solving the (symmetric) traffic assignment problem based on the following assumption:

- the initial intensity of the pheromone trail on each link \( l \), associated to ant colony \( od \), indicated as \( \tau^0_{od,l} \), is a function of path costs, that is:
  \[
  \tau^0_{od,l} = \sum_{p: od \rightarrow p} T^0_{od,p}
  \]
  with:
  \[
  T^0_{od,p} = \begin{cases} 
  \exp\left(-C^0_p / \theta\right) & \text{if } p \in I_{od} \\
  0 & \text{if } p \notin I_{od}
  \end{cases}
  \]
  where \( T^0_{od,p} \) is the initial intensity of the pheromone trail on path \( p \); \( C^0_p \) is the initial cost of path \( p \); \( \theta \) is the parameter of the path choice model; \( I_{od} \) is the set of all available (or considered) paths that join origin node \( o \) with destination node \( d \);

- the increase in the pheromone trail, indicated as \( \Delta \tau^k_{od,l} \), can be expressed by a function of path costs, that is:
  \[
  \Delta \tau^k_{od,l} = \sum_{p: od \rightarrow p} \Delta T^k_{od,p}
  \]
  with:
  \[
  \Delta T^k_{od,p} = \begin{cases} 
  \exp\left(-C^k_p / \theta\right) & \text{if } p \in I_{od} \\
  0 & \text{if } p \notin I_{od}
  \end{cases}
  \]
  where \( \Delta T^k_{od,p} \) is the increase in the pheromone trail at iteration \( k \); \( C^k_p \) is the path \( p \) cost at iteration \( k \);

- updating of the pheromone trail can be expressed as:
  \[
  \tau^{k+1}_{od,l} = (1 - \rho) \cdot \tau^k_{od,l} + \rho \cdot \Delta \tau^k_{od,l}
  \] (A.1)
In particular, D’Acierno et al. (2006) showed that the proposed ACO-based algorithm should be considered an application of an MSA algorithm where successive averages are applied to the pheromone trail, as shown by formulation (A.1).

Finally, D’Acierno et al. (2006) stated the convergence of the proposed ACO algorithm by means of the extension of Blum’s theorem (Blum, 1954) provided by Cantarella (1997).