Minimizing Total Delay in Fixed-Time Controlled Traffic Networks

Ekkehard Köhler, Rolf H. Möhring, and Gregor Wünsch

Technische Universität Berlin, Institut für Mathematik, MA 6-1, Straße des 17. Juni 136, D-10623 Berlin, Germany
{ekoehler,moehring,wuensch}@math.tu-berlin.de

Abstract. We present two different approaches to minimize total delay in signalized fixed-time controlled inner city traffic networks. Firstly, we develop a time discrete model where all calculations are done pathwise and vehicles move on "time trajectories" on their routes. Secondly, an idea by Gartner, Little, and Gabbay (GLG) is extended to a continuous, linkwise operating model using "Link Performance Functions" to determine delays. Both models are formulated as mixed-integer linear programs and are compared and evaluated by PTV AG's simulation tool VISSIM 3.70.

1 Introduction

Controlling the inner city traffic by a "good" setting of relevant traffic light parameters is an appropriate way of reducing congestions in networks and delays in general, respectively. We consider a scenario of dense but almost steady traffic, which appears for example in periods of morning peaks of rush hour traffic. In such cases often fixed-time controlling is used since vehicle actuated or adaptive controlling cannot accentuate their advantages, which lie in the ability of adjusting to different traffic situations. Before we start describing the two optimization models, we review some notation used in traffic engineering.

First of all, we consider networks where there is a *light-signal system* at each intersection, including various single *traffic lights* some of which are combined into so-called *signal groups*. Each of these signal groups controls traffic throughput of a different direction, for example inbound, outbound, or turning traffic. For each signal group there is a *signal timing plan* determining the beginning and ending of the green- and red phase, the so called *red-green split*. After a predetermined amount of time, the *cycle time*, patterns of red and green recur. The most important parameter is the so called *offset*, which describes how light-signal systems of different intersections, or their signal timing plans respectively, are set relative to a given zero-point (Sec. 2) or are set relative to each other (Sec. 3). For both approaches cycle time, red-green split, and the vehicles' *travel time* on the links are fixed and the offset acts as decision variable.

2 A discrete path-based approach

Since in this first approach all delay analysis is done pathwise, we refer to the model as *path-based model* and restrict ourselves in the following to a single path. The model's main characteristic is its discrete structure, see Fig. 1. The cycle time is divided into T trajectories. For example, let $T = 40$; then at the beginning of each route each trajectory carries 0.555 cars per unit of time in case of an assumed traffic volume of 1000 vehicles per hour and a cycle time of 80 seconds. Below, we identify the cycle time with the parameter T and establish a partition into one trajectory per second of cycle time, which will be $T = 80$ for all quoted examples.

For a further development of the model it is necessary to introduce the following parameters: the set of all paths in the network is denoted by $\mathcal{P}$ and the edge set[1] of a path $P \in \mathcal{P}$ is given by $E(P) = \{e_1^P, \cdots, e_{\alpha_P}^P\}$ where α_P is the number of edges of P. The set of intersections, i.e. the set of different traffic-signal systems, is denoted by $\mathcal{K}$, whereas $\mathcal{R}(K)$ is the set of the signal groups corresponding to intersection K. Canonically, the parameter $\tau(e)$ stands for the integral travel time needed to traverse edge $e \in E(P)$ of path $P \in \mathcal{P}$. We will formulate the problem as a MIP and use the following

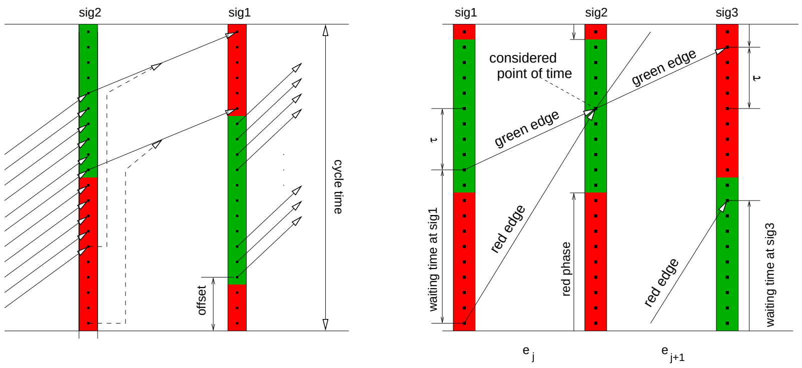

Fig. 1. Consecutive traffic signals of a particular path. At the beginning of each path each of the trajectories leads the same amount of vehicles.

variables: first, we have binary offset variables h, which are defined for each intersection and each point in time $\{1, \ldots, T\}$. For a $K \in \mathcal{K}$ only one variable $h[K, t]$ equals 1. The t_0 with $h[K, t_0] = 1$ corresponds to the offset value at intersection K. A second group of binary variables indicates whether a certain point of time $t \in \{1, \ldots, T\}$ at intersection $K \in \mathcal{K}$ and signal group $R \in \mathcal{R}(K)$ belongs to a green phase or not. These variables are denoted by

[1] We refer to links as edges although we do not actually use an underlying graph structure.

z. So, $z[K,R,t] = 1$ indicates a red phase and $z[K,R,t] = 0$ a green phase, respectively. The variables h and z are connected via

$$\sum_{i \in l[K,R,t]} h[K,i] = z[K,R,t] \qquad \forall K \in \mathcal{K}, \forall R \in \mathcal{R}(K), \forall t \in T, \tag{1}$$

where $l[K,R,t]$ is a preprocessed list of numbers in $\{1,\ldots,T\}$ such that t belongs to a red phase at intersection K and signal group R iff the offset at K is an element of $l[K,R,t]$.

As mentioned above, vehicles move on trajectories on their routes. Each trajectory leads to a certain point of time at an intersection and depending on the traffic light cars on that trajectories can pass the intersection or have to wait because of a red signal. For both cases (a red or a green signal) outgoing trajectories are provided, always one of which is blocked by a constraint and vehicles have to follow "their right" trajectory, as a result. See Fig. 1. The variables x_r and x_g measure the amount of vehicles following those *red edges* and *green edges*. Such blocking is realized by

$$x_g[P,e,t] \leq M \cdot (1 - z[K_{(P,e)}, R_{(P,e)}, t]) \qquad \forall P \in \mathcal{P}, \forall e \in E(P) \backslash \{e_1^P\}, \forall t \in T \tag{2}$$

and

$$x_r[P,e,t] \leq M \cdot z[K_{(P,e)}, R_{(P,e)}, t] \qquad \forall P \in \mathcal{P}, \forall e \in E(P) \backslash \{e_1^P\}, \forall t \in T, \tag{3}$$

where M denotes a big constant and $K_{(P,e)}, R_{(P,e)}$ the obvious dependences. Of course, the approach of traffic flow on trajectories only works, if the cars are "forced" to stay on the trajectories and do not get lost. This is ensured by the following flow conservation equation

$$\begin{aligned} x_g[P,e+1,t] + x_r[P,e+1,t] &= x_g[P,e,t-\tau(e)] \\ &+ x_r[P,e,t-\tau(e) - \mathbf{WT(P,e)}] \quad \forall P, t, \forall e \in E(P) \backslash \{e_1^P, e_{\alpha_P}^P\}. \end{aligned} \tag{4}$$

The waiting time, which has to be constant and integral, is denoted by the parameter WT and calculated with WEBSTER's classical formula [3].

2.1 The MIP formulation

Together with an equation ensuring that only one offset is adjusted and an equation corresponding to an initial flow conservation, constraints (1)-(4) now form our mixed-integer linear program. This MIP models our optimization problem well, however, it has to be mentioned that the model's size, namely the large amount of integral variables, leads to large computation times. For example, a computation time of more than 9 hours[2] is needed for a 9-intersection test network and a discretization of the cycle time into 80 units.

[2] on a 1.7GHz Linux computer with 512MB memory

Note that for the objective, only vehicles on red edges must be taken into account.

$$minimize \sum_{P\in\mathcal{P}} \sum_{j=2}^{\alpha_P} \sum_{t\in T} \mathbf{WZ}[K_{(P,e^P_{j-1})}, R_{(P,e^P_{j-1})}] \cdot x_r[P, e^P_j, t]$$

$$\sum_{t\in T} h[K,t] = 1 \qquad \forall K \in \mathcal{K}, \tag{5}$$

$$\sum_{i\in l[K,R,t]} h[K,i] = z[K,R,t] \tag{6}$$

$$\forall K \in \mathcal{K}, \forall R \in \mathcal{R}(K), \forall t \in T,$$

$$x_g[P,e,t] \leq M \cdot (1 - z[K_{(P,e)}, R_{(P,e)}, t]) \tag{7}$$

$$\forall P \in \mathcal{P}, \forall e \in E(P)\backslash\{e^P_1\}, \forall t \in T,$$

$$x_r[P,e,t] \leq M \cdot z[K_{(P,e)}, R_{(P,e)}, t] \tag{8}$$

$$\forall P \in \mathcal{P}, \forall e \in E(P)\backslash\{e^P_1\}, \forall t \in T,$$

$$x_g[P,e^P_2,t] + x_r[P,e^P_2,t] = flowrate[P] \tag{9}$$

$$\forall P \in \mathcal{P}, \forall t \in T,$$

$$x_g[P,e^P_{j+1},t] + x_r[P,e^P_{j+1},t] =$$

$$x_g[P,e^P_j,t-\tau(e^P_j)] + x_r[P,e^P_j,t-\tau(e^P_j) - \mathbf{WT}(P,e^P_j)] \tag{10}$$

$$\forall P \in \mathcal{P}, \forall t \in T, j = 2 \cdots \alpha_P,$$

$$h, z \quad \in \{0,1\},$$

$$x_g, x_r \quad \geq 0.$$

3 A continuous model

This second model is based on an approach by Gartner, Little, and Gabbay (GLG)[1]. In 1975 they formulated a "Network Coordination Problem" (NCP) where cycle time and red-green split are fixed and total waiting time is to be minimized by finding optimal offsets. Later, they included cycle time and red-green split to the set of variables and formulated a "Network Synchronization Problem" [2]. We extend their NCP model in a canonical way and introduce some adjustments in the objective function.

Again, some parameters have to be introduced first: The cycle time is again denoted by T (in our examples it is set to 80 seconds). The traffic network is represented by a directed multi-graph $G = (V, A)$, where the nodes $v \in V$ are the intersections and the edges $e \in E$ stand for the traffic carrying streets/links between intersections. Thus one can see that the graph has to be directed, since traffic may move in both directions of a link. Moreover there maybe more than one copy of an arc because we have to distinguish between different vehicle flows of a link according to which signal group they come from and go to. The set of all cycles in G will be denoted by $\mathcal{C}$ and for a single $\ell \in \mathcal{C}$ the set of forward edges is $F(l)$ and the set of reverse edges is $R(l)$, whereas a cycle is traversed clockwise.

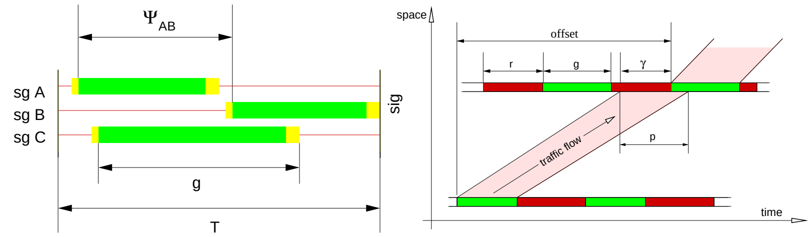

Fig. 2. The left side shows a sketch of a signal timing plan with three signal groups. The parameter Ψ describes the adjustment of the beginning of the green phases of the different signal groups. Of course, it holds that $\Psi_{AA} = 0$ and $\Psi_{AB} = 1 - \Psi_{BA}$. The right side illustrates the traffic flow in a platoon at one link. Because this structure allows linkwise definitions of e.g. the offset, all calculations and waiting time analysis is done linkwise. Note that the platoon length p does not necessarily equal the length of its green phase.

As already mentioned, there are two main characteristics of the model. Firstly, it is assumed that vehicles move in platoons with a constant rate of flow within. Secondly, this approach does not use any discretization but the parameters and variables can take continuous values. The arrival time of the platoon is denoted by γ and has to be within the interval $[-r, g]$. So, the offset ϕ is defined as the distance (in time) between the platoon's starting and the beginning of the green phase that defines the relevant interval for the platoon's γ. For example, in Fig. 2, γ is negative. Because of the linear dependency $\phi = \tau - \gamma$ one can use the arrival time γ as well as ϕ for calculating the delays.

Now, all necessary parameters and variables have been introduced. The big advantage of this "link-based" approach is its compactness. Only one group of constraints has to be formulated, namely the so called cycle-equations

$$\sum_{e \in F(\ell)} \phi_e - \sum_{e \in R(\ell)} \phi_e + \sum_{r=1}^{k_\ell} \Psi^{v_r^\ell}_{P[r,e^\ell_{r-1}],P[r,e^\ell_r]} = n_\ell T \qquad \forall \ell \in \mathcal{C}.$$

The parameter $P[\cdot,\cdot]$ indicates the particular signal groups. These equations are essential, since all calculations shall be done modulo the cycle time, because all cycles shall be regarded as equal[3]. Therefore a particular point of time during one circulation of the cycle has to correspond to the *same* point of time in *all* cycles. For this reason, the sum of all offsets around a cycle[4] in the graph must be an integral multiple of T.

[3] Therefore traffic load values are fixed over all cycles.
[4] A cycle may also consist of only a pair of edges!

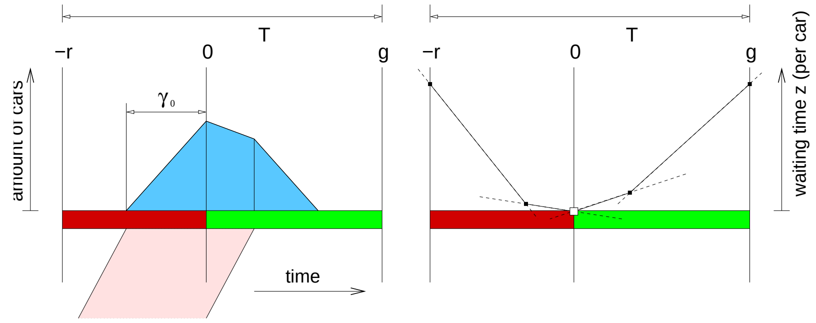

Fig. 3. A sketch of the arising waiting queues of a platoon arriving at approx. $\gamma = -r/2$. Note the break in the curve due to different service rates, first $s - q$ and after the platoon's tail s. The curve on the right represents the Link Performance Function of platoon's arrival time γ.

Now, the only open question is how to calculate delays. This is done using the arrival time γ. This is, however, quadratically related to the waiting time. To obtain a linear model each link's delay function is evaluated at some intermediate points and then approximated by a piecewise linear function, see Fig. 3. The choice of such intermediate points determines the shape of the link's delay function. At this point, we significantly simplified the GARTNER, LITTLE, and GABBAY method by ensuring the convexity of the curve. This is important for the following reason. Suppose the curve is not convex, the inequalities $z \geq g_{1,2,\ldots}(\gamma)$ where g denotes the approximation lines, do not force waiting time z to be equal to an actual waiting time. But instead the variable z probably takes too high values.

3.1 The MIP formulation

Also this model is represented as a mixed-integer linear program.

$$\mathit{minimize} \sum_{(i,j)\in A} f_{ij} z_{ij}$$

$$\sum_{e\in F(\ell)} \phi_e - \sum_{e\in R(\ell)} \phi_e + \sum_{r=1}^{k_\ell} \Psi^{v_r^\ell}_{P[r,e_{r-1}^\ell],P[r,e_r^\ell]} = n_\ell T \qquad \forall \ell \in \mathcal{C},$$

$$z_{ij} \geq g_r^e(\tau_{ij} - \phi_{ij}) \qquad \forall e = (i,j) \in A, \quad r = 1 \ldots k_e,$$

$$\underline{n_\ell} \leq n_\ell \leq \overline{n_\ell} \qquad \forall \ell \in \mathcal{C},$$

$$n_\ell \in \mathbb{Z} \qquad \forall \ell \in \mathcal{C},$$

$$\phi_{ij} \in [\tau_{ij} - g_{ij}, \tau_{ij} + r_{ij}], \quad \forall (i,j) \in A.$$

One can see immediately the compact formulation compared to the first model. This also leads to faster computation times. E.g. a test network of 25 intersections could be optimally solved in about 5 seconds.

In addition to improvements within the links' delay functions we also extended the GLG approach such that we distinguish traffic flow on one edge not only according to its direction but also according to its origin's and destination's signal group. To this end, we simply allocate multiple link copies and analyze them separately. This only needs a few more cycle equalities and therefore some more integral variables, but it admits to consider a more complex traffic flow. Note that adding good lower and upper bounds on the integral variable n may lead to better computation times.

4 Simulation using VISSIM 3.70

Both optimization models have been tested and validated with PTV's simulation tool VISSIM 3.70. It provides microscopic traffic flow simulation and various analysis tools. One of them, the so called *lost time* corresponds best to our criteria *waiting time*.

In addition to test runs on simple instances, such as arterials or 2×2 grid networks, we constructed a 3×3 as well as a 4×4 grid network and defined 14 respectively 19 routes on which motorized individual traffic moves. In detail we have 12(16) routes of traffic following the grid on straight paths without any turns, divided into 6(8) "horizontal paths" and 6(8) vertical traffic streams. In addition, we added 2 non-conventional paths to the 3×3 grid that include right-turn traffic and 3 paths to the 4×4 grid containing left-turn traffic.

Results of simulation runs for both of these instances are listed in the tables below. They are compared to the results of a heuristic that sets a progressive signal system to some of the paths. In this way, lost time is minimized for 43% of all cars in the 3×3 grid and for 46% of the vehicles in the larger grid, respectively.

Table 1. Comparison between lost time measurements for the 3×3 grid's traffic with VISSIM 3.70, using the offsets of both approaches and a heuristic. The discrete model is denoted by A, whereas B stands for the continuous approach.

Model	Solution	Lost time in sec/veh	Rate of improvement in %
A	non opt.	21.2	29.3%
B	opt.	23.4	22%
heuristic		30.0	-
	random	32.5	-8.3%

As it can be seen in the tables only suboptimal offsets could be used for evaluating the path-based model, since its size did not permit to calculate an optimal solution within a reasonable amount of time, which was a few hours.

Table 2. Same measurements as listed in Table 1 but for the 4×4 grid instance.

Model	Solution	Lost time in sec/veh	Rate of improvement in %
A	non opt.	42.6	1.8%
B	opt.	39.7	8.5%
B^{*} [a]	opt.	38.8	10.6%
heuristic		43.4	-
	random	55.9	-28.8%

[a] Model with small changes at the LPFs, namely different interpolation points.

Table 2 also contains average simulation results for 11 randomly chosen offset sets. Although a more realistic comparison to results of, say, actual offsets of an actual traffic network are missing, it can be seen that both models calculate better offsets than the heuristic. We consider this to be a promising start and we will extend and advance both models as well as design further non-grid real world instances and compare results of our models with results of the actual adjusted offset set.

Acknowledgement

We would like to thank PTV AG, Karlsruhe, and Klaus Nökel for providing the simulation tool VISSIM 3.70 and for inspiring discussions.

References

1. Gartner, N. H., Little, J. D. C., Gabbay, H. (1975) Optimization of Traffic Signal Settings by Mixed-Integer Linear Programming, Part I: The Network Coordination Problem. Transportation Science **9**, 321–343
2. Gartner, N. H., Little, J. D. C., Gabbay, H. (1975) Optimization of Traffic Signal Settings by Mixed-Integer Linear Programming, Part II: The Network Synchronization Problem. Transportation Science **9**, 344–363
3. Webster, F. V. (1958) Traffic Signal Settings. Road Research Technical Paper **39**