Delft University of Technology

Delft Center for Systems and Control

Technical report 10-007

Model predictive control for urban traffic networks via MILP*

S. Lin, B. De Schutter, Y. Xi, and H. Hellendoorn

If you want to cite this report, please use the following reference instead:
S. Lin, B. De Schutter, Y. Xi, and H. Hellendoorn, "Model predictive control for urban traffic networks via MILP," *Proceedings of the 2010 American Control Conference*, Baltimore, Maryland, pp. 2272–2277, June–July 2010.

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.51.19 (secretary)
fax: +31-15-278.66.79
URL: `http://www.dcsc.tudelft.nl`

*This report can also be downloaded via `http://pub.deschutter.info/abs/10_007.html`

Model Predictive Control for Urban Traffic Networks via MILP

Shu Lin, Bart De Schutter, Yugeng Xi and Hans Hellendoorn

Abstract— **Model Predictive Control (MPC) is an advanced control strategy that can easily coordinate urban traffic networks. But, due to the nonlinearity of the traffic model, the optimization problem of the MPC controller will become intractable in practice when the scale of the controlled traffic network grows larger. To solve this problem, the nonlinear traffic model is reformulated into a model with only linear equations and inequalities. Mixed-Integer Linear Programming (MILP) algorithms can efficiently solve the reformulated optimization problem, and guarantee the global optimum at the same time. Moreover, the MILP optimization problem is further relaxed by model reduction and adding upper bound constraints.**

I. INTRODUCTION

Traffic congestion causes losses to both individuals and the society. Traffic control is one of the most efficient and also effective ways to alleviate traffic congestion, and thus reduce the losses caused by congestion. To reduce traffic congestion from a network-wide point of view, advanced traffic control strategies are needed to coordinate traffic lights of the intersections within throughout traffic networks.

There are already many control strategies [1] developed and implemented for urban traffic. Among them, model-based optimization methods (including Model Predictive Control, MPC) [2] are effective control approaches based on a similar control framework. The framework contains three classical steps: prediction, on-line optimization, and rolling time horizon. Due to this framework, model-based optimization control methods are capable of coordinating the traffic lights within urban traffic networks, of dealing with the uncertainty of real traffic, and of avoiding myopic control schemes. All these advantages make the model-based optimization methods very attractive.

Many model-based optimization strategies have already been developed. In the 1980s and 1990s, a number of model-based optimization control strategies emerged: OPAC, PRODYN, CRONOS, and RHODES [1]. These control strategies predict the future traffic demands at the intersections through the traffic flow data measured from the upstream detectors and the detectors in upstream links. They showed advantages compared with the conventional traffic-responsive strategies, but have a limitation on the prediction horizon. In recent years, some macroscopic urban traffic models, which can describe the traffic dynamic mechanics of the traffic flow, have been developed. Model-based optimization control strategies [2] based on these prediction models were developed, and overcame the drawbacks of previous strategies. However, the biggest challenge for implementing the model-based optimization control strategy is the high on-line computational complexity. Although these model-based control strategies have a lot of advantages, the real-time feasibility[1] is hard to guaranteed. Different methods have been applied to avoid this problem.

In this paper, the real-time feasibility of MPC controllers is increased further by improving the efficiency of solving optimization problems. The optimization of the MPC controller is reformulated into different optimization problems solved by different optimization algorithms. First, we rewrite the original nonlinear optimization problem into a Mixed-Integer Linear Programming (MILP) problem, which can be solved efficiently by an MILP solver. To achieve this, we propose an approach to reformulate the nonlinear urban traffic model (S model of [3], [4]) into a mixed-integer linear model, which only contains mixed-integer linear equations and inequalities. Next, we reduce the S model into an S* model by omitting one of the constraints. After reformulating the S* model, a more relaxed MILP optimization problem is obtained. Simulations are run to investigate and compare the different optimization approaches.

II. S MODEL

A simplified model (S model [3]) is derived from the macroscopic urban traffic network model (BLX model developed in [2], [5]). The S model is fast and also accurate enough as a prediction model of MPC controllers [3]. In the S model, a link is marked by its upstream node u and downstream node d as link (u,d). The sets of upstream and downstream nodes of link (u,d) are $I_{u,d}$ and $O_{u,d}$ (e.g., for the situation of Fig. 1 we have $I_{u,d} = \{i_1, i_2, i_3\}$ and $O_{u,d} = \{o_1, o_2, o_3\}$).

In order to describe the evolution of the models, we first define some variables (see also Fig. 1):

$I_{u,d}$	: set of input nodes of link (u,d),
$O_{u,d}$	: set of output nodes of link (u,d),
k	: simulation step counter,
$n_{u,d}(k)$	: number of vehicles in link (u,d) at step k,
$q_{u,d}(k)$	: queue length at step k in link (u,d), $q_{u,d,o}$ is the queue length of the sub-stream turning to o,

S. Lin and Y. Xi are with the Department of Automation, Shanghai Jiao Tong University, No. 800 Dongchuan Road, Minhang District, Shanghai, P.R. China; S. Lin is also with the Delft Center for Systems and Control, Delft University of Technology. lisashulin@gmail.com, ygxi@sjtu.edu.cn

B. De Schutter and J. Hellendoorn are with the Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands; B. De Schutter is also with the Marine & Transport Technology department of Delft University of Technology. b.deschutter@tudelft.nl, j.hellendoorn@tudelft.nl

[1]Real-time feasibility means that the on-line optimization problem can be solved fast enough to satisfy the real-time requirement of MPC controllers

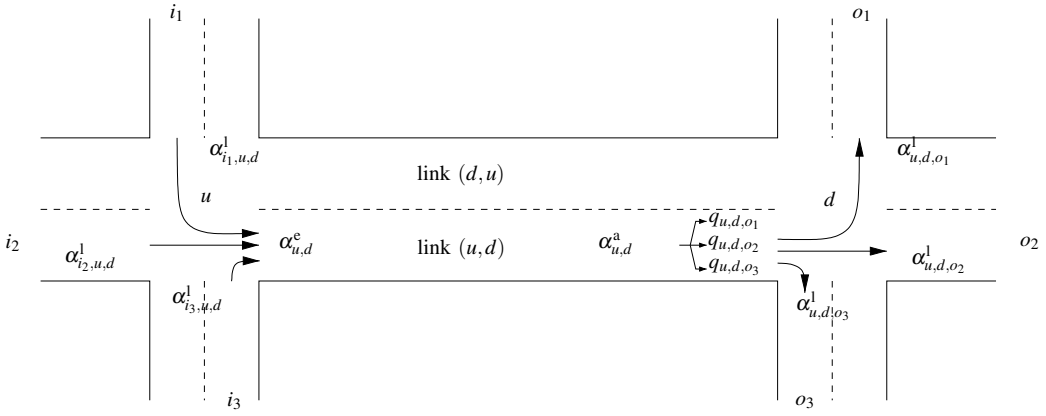

Fig. 1. A link connecting two traffic-signal-controlled intersections

$\alpha^{\mathrm{l}}_{u,d}(k)$: flow rate leaving link (u,d) at step k, $\alpha^{\mathrm{l}}_{u,d,o}(k)$ is the leaving flow rate of the sub-stream towards o,

$\alpha^{\mathrm{a}}_{u,d}(k)$: flow rate arriving at the end of the queue in link (u,d) at step k, $\alpha^{\mathrm{a}}_{u,d,o}(k)$ is the arriving flow rate of the sub-stream towards o,

$\alpha^{\mathrm{e}}_{u,d}(k)$: flow rate entering link (u,d) at step k, $\alpha^{\mathrm{e}}_{i,u,d}(k)$ is the flow rate entering link (u,d) from i,

$\beta_{u,d,o}(k)$: relative fraction of the traffic turning to o at step k,

$\mu_{u,d}$: saturated flow rate leaving link (u,d),

$g_{u,d,o}(k)$: green time length during step k for the traffic stream towards o in link (u,d),

$v^{\mathrm{free}}_{u,d}$: free-flow vehicle speed in link (u,d),

$C_{u,d}$: capacity of link (u,d) expressed in number of vehicles,

$N^{\mathrm{lane}}_{u,d}$: number of lanes in link (u,d),

$\Delta c_{u,d}$: offset between node u and node d, which represents the offset time between the cycle times of the upstream and the downstream intersections at the beginning of every control time step,

l_{veh} : average vehicle length.

In the S model, every intersection takes the cycle time as its simulation time interval. The cycle times for intersection u and d, denoted by c_u and c_d respectively, can be different from each other, as Fig. 2 illustrates. In this situation, the simulation step counters of different intersections are not same. The input and output flow rates of the link are averaged over the cycle times in the S model.

Taking the cycle time c_d as the length of the simulation time interval for link (u,d) and k_d as the corresponding time step counter, $n_{u,d}(k_d)$[2] is updated by the input and output average flow rate as

$$n_{u,d}(k_d+1) = n_{u,d}(k_d) + \left(\alpha^{\mathrm{e}}_{u,d}(k_d) - \sum_{o \in O_{u,d}} \alpha^{\mathrm{l}}_{u,d,o}(k_d) \right) \cdot c_d \ . \tag{1}$$

[2] $n_{u,d}(k_d)$ is the average number of the vehicles in link (u,d) at step k_d including the queues

The leaving average flow rate over c_d is determined by the capacity of the intersection, the number of cars waiting and/or arriving, and the available space in the downstream link:

$$\begin{aligned} \alpha^{\mathrm{l}}_{u,d,o}(k_d) = \min\Big(& \beta_{u,d,o}(k_d) \cdot \mu_{u,d} \cdot g_{u,d,o}(k_d)/c_d, \\ & q_{u,d,o}(k_d)/c_d + \alpha^{\mathrm{a}}_{u,d,o}(k_d), \\ & \beta_{u,d,o}(k_d) \left(C_{d,o} - n_{d,o}(k_d) \right)/c_d \Big) \ . \end{aligned} \tag{2}$$

The number of vehicles waiting in the queue turning to o is updated as

$$\begin{aligned} q_{u,d,o}(k_d+1) = & q_{u,d,o}(k_d) + \left(\beta_{u,d,o}(k_d) \cdot \alpha^{\mathrm{a}}_{u,d}(k_d) - \right. \\ & \left. \alpha^{\mathrm{l}}_{u,d,o}(k_d) \right) \cdot c_d \ . \end{aligned} \tag{3}$$

The flow rate entered link (u,d) will arrive at the end of the queues after a time delay $\tau(k_d) \cdot c_d + \gamma(k_d)$, i.e.,

$$\begin{aligned} \alpha^{\mathrm{a}}_{u,d}(k_d) = & \frac{c_d - \gamma(k_d)}{c_d} \cdot \alpha^{\mathrm{e}}_{u,d}\left(k_d - \tau(k_d)\right) + \\ & \frac{\gamma(k_d)}{c_d} \cdot \alpha^{\mathrm{e}}_{u,d}\left(k_d - \tau(k_d) - 1\right), \end{aligned} \tag{4}$$

$$\begin{aligned} \tau(k_d) &= \mathrm{floor}\left\{ \frac{\left(C_{u,d} - q_{u,d}(k_d)\right) \cdot l_{\mathrm{veh}}}{N^{\mathrm{lane}}_{u,d} \cdot v^{\mathrm{free}}_{u,d} \cdot c_d} \right\}, \\ \gamma(k_d) &= \mathrm{rem}\left\{ \frac{\left(C_{u,d} - q_{u,d}(k_d)\right) \cdot l_{\mathrm{veh}}}{N^{\mathrm{lane}}_{u,d} \cdot v^{\mathrm{free}}_{u,d} \cdot c_d} \right\}. \end{aligned} \tag{5}$$

The flow rate entering link (u,d) is the sum of the flow rates entering from all the upstream links:

$$\alpha^{\mathrm{e}}_{u,d}(k_d) = \sum_{i \in I_{u,d}} \alpha^{\mathrm{e}}_{i,u,d}(k_d) = \sum_{i \in I_{u,d}} \alpha^{\mathrm{l}}_{i,u,d}(k_u). \tag{6}$$

Recall that different cycle times between the upstream and downstream intersections are defined, so the flow rates leaving from upstream links have to be synchronized before entering the current link.

In order to control the urban traffic network, a common control time interval T_{c} has to be defined for the network model (as in Fig. 2), so that intersections can communicate with each other and be synchronized. The control time

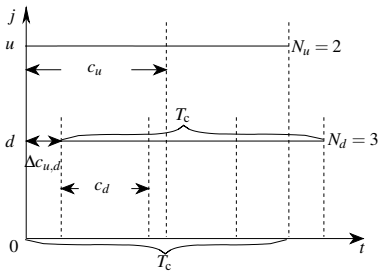

Fig. 2. Relationship between cycle times and control time interval

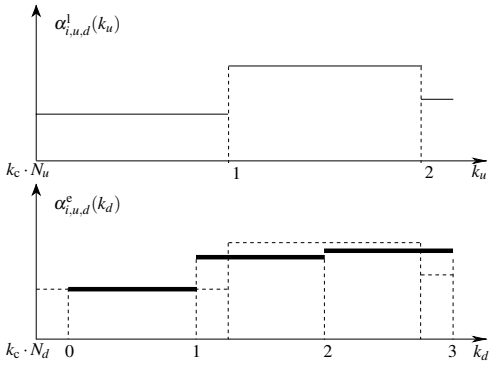

Fig. 3. Illustration for synchronizing flow rates

interval T_c is set to the Least Common Multiple of all the cycle times in the network, i.e. $T_c = N_j c_j$ (for all $j \in J$), with N_j an integer.

Then, the flow rates expressed in the timing of intersection u can be recast into the timing of intersection d, as Fig. 3 shows. First, we transform the discrete time leaving flow rates from the upstream links into continuous time using the zero-order hold strategy, as

$$\alpha^{l,\mathrm{cont}}_{i,u,d}(t) = \alpha^{l}_{i,u,d}(k_u), \quad k_u \cdot c_u \le t < (k_u+1)\cdot c_u, \quad (7)$$

and then sample them again to obtain the average flow rates in time step k_d so as to be used by the downstream link as

$$\alpha^{e}_{i,u,d}(k_d) = \frac{\int_{k_d \cdot c_d + \Delta c_{u,d}}^{(k_d+1)\cdot c_d + \Delta c_{u,d}} \alpha^{l,\mathrm{cont}}_{i,u,d}(t)}{c_d} \mathrm{d}t \ . \quad (8)$$

III. MPC FOR URBAN TRAFFIC NETWORKS

Model Predictive Control [6] is a methodology that implements and repeatedly applies Optimal Control over a prediction horizon in a rolling horizon way. MPC has the ability to deal with the uncertainty of the process, which can be caused by the unpredictable disturbances, the (slow) variation over time of the parameters, and model mismatches in the prediction model. MPC can also easily deal with multi-input and multi-output problems with constraints.

Therefore, MPC is a suitable advanced control strategy for controlling and coordinating large urban traffic networks. The S model can be used as the prediction model of the MPC controller. However, due to the nonlinear nature of the S model, only nonlinear optimization algorithms (e.g. Sequential Quadratic Programming (SQP) and Pattern Search (PS)) can solve the optimization problem of the MPC controller. Nonlinear optimization algorithms usually search for the optimal solution numerically by computing the nonlinear model repeatedly. Thus, the MPC controller will become real-time infeasible when the scale of the controlled traffic network grows.

IV. REFORMULATION OF THE S MODEL

As shown in Section II, the nonlinear features of the S model are caused by the Max-Min-Plus-Scaling (MMPS) formula (2), and the nonlinear formulas (4), (7), and (8). In order to increase the real-time feasibility of the MPC controller, the nonlinear optimization problem can be rewritten into a Mixed-Integer Linear Programming (MILP) problem by reformulating the S model. Compared with nonlinear optimization algorithms, MILP is an efficient optimization algorithm guaranteeing the global optimum at the same time. An MMPS model can be equivalently transformed into a Mixed Logical Dynamical (MLD) model, which can be expressed as mixed-integer equations and inequalities. Therefore, after the S model is reformulated into an MLD model, an efficient MILP solver can be used to solve the optimization problem of the MPC controller. The nonlinear formulas (4), (7), and (8) can be easily linearized by making an assumption. Here are the two assumptions made to assist establishing the MILP-based MPC controller:

- **Assumption 1:** We assume that the traffic state $n(k)$ by the traffic model stays constant during the simulation time interval T_s. Then the objective function TTS can be approximated as in (27).
- **Assumption 2:** We assume the time delay of the vehicles traveling from the beginning of the link to the end of the queues in the link to be constant and independent of k. Then (4) becomes linear as

$$\alpha^{a}_{u,d}(k_d) = (1-\gamma_{\mathrm{const}})\cdot \alpha^{e}_{u,d}(k_d - \tau_{\mathrm{const}}) + \gamma_{\mathrm{const}} \cdot \alpha^{e}_{u,d}(k_d - \tau_{\mathrm{const}} - 1), \quad (9)$$

where τ_{const} and γ_{const} are constant values obtained by (5) with the queue length fixed. This queue length can be taken as the average queue length based on the historical data.

A. Rules for Equivalent Transformation to MLD System

According to [7], consider the statement $f(x) \le 0$, where $f : \mathbb{R}^n \to \mathbb{R}$. Assume that $x \in \mathscr{X}$, where $\mathscr{X}$ is a given bounded set, and select M, m such that

$$M \ge \max_{x\in\mathscr{X}} f(x), \quad m \le \min_{x\in\mathscr{X}} f(x) \ . \quad (10)$$

Then, by introducing in $\delta \in \{0,1\}$, the following equivalence holds true

$$[f(x) \le 0] \Leftrightarrow [\delta = 1], \text{ iff } \begin{cases} f(x) \le M(1-\delta) \\ f(x) \ge \varepsilon + (m-\varepsilon)\delta \end{cases}, \quad (11)$$

where ε is a small tolerance, typically the machine precision.

Moreover, $\delta f(x)$ can be replaced by auxiliary real variable $z = \delta f(x)$ which satisfies $[\delta = 0] \Rightarrow [z = 0]$, $[\delta = 1] \Rightarrow [z = f(x)]$. Then $z = \delta f(x)$ is equivalent to

$$\begin{cases} z \le M\delta \\ z \ge m\delta \\ z \le f(x) - m(1-\delta) \\ z \ge f(x) - M(1-\delta) \end{cases} . \tag{12}$$

B. Reformulation from MMPS Model to MLD Model

With the equivalences above, the MMPS formula (2) can be reformulated as an MLD formula. Let

$$\begin{aligned} a &= \beta_{u,d,o}(k_d) \cdot \mu_{u,d} \cdot g_{u,d,o}(k_d)/c_d \\ b &= q_{u,d,o}(k_d)/c_d + \alpha^{\mathrm{a}}_{u,d,o}(k_d) \\ c &= \beta_{u,d,o}(k_d)\left(C_{d,o} - n_{d,o}(k_d)\right)/c_d \\ d &= \min(a,b), \end{aligned} \tag{13}$$

then (2) becomes

$$\alpha^{\mathrm{l}}_{u,d,o}(k_d) = \min(a,b,c) = \min(d,c) \ . \tag{14}$$

Let

$$f_1 = b - a, \tag{15}$$

and define

$$\delta_1 = \begin{cases} 1 & \text{if } f_1 \le 0 \\ 0 & \text{if } f_1 > 0 \end{cases} \tag{16}$$

then we have

$$d = a + (b-a)\cdot\delta_1 = a + f_1 \cdot \delta_1 \ . \tag{17}$$

Similarly, let

$$f_2 = c - d, \tag{18}$$

and define

$$\delta_2 = \begin{cases} 1 & \text{if } f_2 \le 0 \\ 0 & \text{if } f_2 > 0 \end{cases} \tag{19}$$

then we have

$$\min(d,c) = d + (c-d)\cdot\delta_2 = d + f_2 \cdot \delta_2 \ . \tag{20}$$

Let

$$z_1 = f_1 \cdot \delta_1 \tag{21}$$

$$z_2 = f_2 \cdot \delta_2 \tag{22}$$

and substitute (17) into (20), then (14) becomes a linear expression

$$\alpha^{\mathrm{l}}_{u,d,o}(k_d) = a + z_1 + z_2 \ . \tag{23}$$

Then, (16), (19), (21) and (22) can be equivalently rewritten into inequality constraints as in (11) and (12).The maximum value and the minimum value of f_1 are $M_1 = C_{u,d}/c_d$ and $m_1 = -\mu_{u,d}$, and the maximum value and the minimum value of f_2 are $M_2 = C_{d,o}/c_d$ and $m_2 = -\min(\mu_{u,d}, C_{u,d}/c_d)$.

Therefore, by introducing the additional auxiliary binary variables δ_1 and δ_2, and the auxiliary real variables f_1, f_2, z_1, and z_2, the original formula (2) in the urban traffic model is equivalently reformulated as mixed-integer linear equations (15), (18) and (23), and inequalities.

With the reformulation and the assumptions above, the urban traffic model is reformulated from an MMPS model into an MLD model.

C. Model Synchronization Reformulation

Consider (8) for fixed i,u,d and k_d, we will now show that this result in

$$\alpha^{\mathrm{e}}_{i,u,d}(k_d) = \mathscr{F}\left(\alpha^{\mathrm{l}}_{i,u,d}(\ell), \cdots, \alpha^{\mathrm{l}}_{i,u,d}(\ell + N_\ell)\right), \tag{24}$$

with ℓ, N_ℓ integers, and $\mathscr{F}$ is a linear function.

In (8), $\alpha^{\mathrm{l,const}}_{i,u,d}(t)$ is a piecewise continuous function. Let $\xi_\ell, \cdots, \xi_{\ell+N_\ell}$ be the length of the intervals and $\alpha^{\mathrm{l}}_{i,u,d}(\ell), \cdots, \alpha^{\mathrm{l}}_{i,u,d}(\ell + N_\ell)$ be the function values of this function on $[k_d c_d + \Delta c_{u,d}, (k_d+1)c_d + \Delta c_{u,d}]$. Hence, the flow rate entering link (u,d) is synchronized from time step k_u to k_d by input synchronization function

$$\alpha^{\mathrm{e}}_{i,u,d}(k_d) = \mathscr{F}_{\mathrm{in}}\left(\alpha^{\mathrm{l}}_{i,u,d}(k_u)\right) = \frac{1}{c_d}\sum_{j=0}^{N_\ell} \xi_{\ell+j}\alpha^{\mathrm{l}}_{i,u,d}(\ell + j), \tag{25}$$

which is a linear expression in the $\alpha^{\mathrm{l}}_{i,u,d}$ values.

When the flow rate leaving link (u,d) is computed in the S model, the number of vehicles in downstream links $n_{d,o}(k_o)$ are used to calculate the number of vehicles that the downstream links can accept at most. The simulation time step counter of intersection o is k_o. If k_o is different from k_d, an output synchronization function is needed for synchronizing the original number of vehicles in the downstream link of link (u,d), $n^{\mathrm{origin}}_{d,o}(k_o)$, from time step k_o to k_d, as

$$n_{d,o}(k_d) = \mathscr{F}_{\mathrm{out}}\left(n^{\mathrm{origin}}_{d,o}(k_o)\right), \tag{26}$$

which is also a linear expression derived using the same rules as deriving the input synchronization function.

D. Objective Function

Recall that $n_{u,d}(k_d)$ equals to the average number of vehicles in link (u,d) including the queues. Hence, according to Assumption 1, the objective function, Total Time Spent (TTS), can be approximated by linear function

$$J_{\mathrm{TTS}} = \sum_{k_d = N_d k_{\mathrm{c}}+1}^{N_d(k_{\mathrm{c}}+N_{\mathrm{p}})} \sum_{(u,d)\in L} c_d \cdot n_{u,d}(k_d). \tag{27}$$

V. MILP BASED MPC CONTROLLER

For intersection d, the control time interval and the simulation time interval satisfy $T_{\mathrm{c}} = N_d c_d$. Then, for a given control time step k_{c}, the corresponding simulation time steps are $k_d = N_d k_{\mathrm{c}}, N_d k_{\mathrm{c}} + 1, \cdots, N_d(k_{\mathrm{c}}+1) - 1$.

After the model reformulation, the optimization problem of the MPC controller can be expressed as an MILP problem of the following form:

$$\begin{aligned} &\min_{\mathbf{u}(k_{\mathrm{c}})} J_{\mathrm{TTS}} = \mathbf{c}^{\mathrm{T}} \cdot \mathbf{u}(k_{\mathrm{c}}) \\ s.t. \quad &\mathbf{A}\mathbf{u}(k_{\mathrm{c}}) \le \mathbf{b} \\ &\mathbf{A}_{\mathrm{eq}}\mathbf{u}(k_{\mathrm{c}}) = \mathbf{b}_{\mathrm{eq}} \\ &\mathbf{u}_{\min} \le \mathbf{u}(k_{\mathrm{c}}) \le \mathbf{u}_{\max} \end{aligned} \tag{28}$$

for appropriately defined matrices $\mathbf{A}$, $\mathbf{A}_{\text{eq}}$, and vectors $\mathbf{c}$, $\mathbf{b}$, $\mathbf{b}_{\text{eq}}$, $\mathbf{u}_{\min}$ and $\mathbf{u}_{\max}$, where $\mathbf{u}(k_{\text{c}})$ contains all the optimized variables including the control inputs, the states, and the auxiliary variables for control time steps $k_{\text{c}},\cdots,k_{\text{c}}+N_{\text{p}}-1$ (see expressions (29) and (30)); the constraints contain the linear inequality constraints, the linear equality constraints, and the lower and upper bounds of the optimization variables.

The inequality constraints in (28) are the mixed-integer inequality constraints obtained as in Section IV-A and IV-B for all the traffic streams in link (u,d) ($\in L$) within the network and for all the predicted simulation time steps in the future (i.e. for $k_d,k_d+1,\cdots,k_d+N_dN_{\text{p}}-1$). The equality constraints contain the linear traffic states update equations (1) and (3), the linear equations (15), (18) and (23) obtained from the model reformulation, the linearized equation (9), the pre-specified reformulated synchronization equations (25) and (26), and the cycle time constraints for the green times of every intersection.

The vector of optimized variables at control time step k_{c} in optimization problem (28) is

$$\mathbf{u}(k_{\text{c}}) = [u^{\text{T}}(k_{\text{c}}|k_{\text{c}})\ u^{\text{T}}(k_{\text{c}}+1|k_{\text{c}})\ \cdots\ u^{\text{T}}(k_{\text{c}}+N_{\text{p}}-1|k_{\text{c}})]^{\text{T}}\ , \tag{29}$$

where $u(k_{\text{c}})$ at any control time step k_{c} consists of control variables (i.e. green time splits), state variables, and auxiliary variables for all the nodes and links in the traffic network as:

$$u(k_{\text{c}}) = [\ \overbrace{g^{\text{T}}(k_{\text{c}})}^{\text{Control variables}} \overbrace{q^{\text{T}}(k_{\text{c}})\ n^{\text{T}}(k_{\text{c}})\ n^{\text{T}}_{\text{downLink}}(k_{\text{c}})\ \alpha_{\text{l}}^{\text{T}}(k_{\text{c}})\ \alpha_{\text{a}}^{\text{T}}(k_{\text{c}})\ \alpha_{\text{e}}^{\text{T}}(k_{\text{c}})}^{\text{State variables}} \overbrace{\delta_1^{\text{T}}(k_{\text{c}})\ \delta_2^{\text{T}}(k_{\text{c}})\ f_1^{\text{T}}(k_{\text{c}})\ f_2^{\text{T}}(k_{\text{c}})\ z_1^{\text{T}}(k_{\text{c}})\ z_2^{\text{T}}(k_{\text{c}})]^{\text{T}}}^{\text{Auxiliary variables}}\ . \tag{30}$$

where $n_{\text{downLink}}(k_{\text{c}})$ represents the vector of the state variables which gives the number of vehicles in the downstream links. All the optimized variables are real values except the binary variables $\delta_1(k_{\text{c}})$ and $\delta_2(k_{\text{c}})$.

Supplied with initial traffic states and traffic demands of the network, the optimization problem can be solved at the control time step k_{c} by an MILP solver.

VI. S* MODEL-BASED MPC CONTROLLER VIA MILP

A. S* Model

For the S model described in Section II, the formula (2) computing the average flow rate leaving link (u,d) is the minimum of three parts. Each part gives the possible leaving flow rate under a traffic scenario. Under the saturated scenario, the average leaving flow rate depends on the saturated flow rate and the green time of the link; under the unsaturated scenario, the average flow rate is calculated according to the waiting and arriving flow rate at the intersection; under the over-saturated scenario, the average flow rate depends on the flow rate that the downstream link can accept. The traffic is always in the scenario which has the minimal average flow rate that could possible leave the link. As an urban traffic model, the S model is capable of describing all the situations that may happen in reality. However, when the S model is taken as a control model of the MPC controller, the third part of (2) can be removed from the S model to leave the over-saturated scenario out if extra constraints are added. Therefore, the S model can be rewritten into S* model by substituting (2) into

$$\alpha^{\text{l}}_{u,d,o}(k_d) = \min\Big(\beta_{u,d,o}(k_d)\cdot\mu_{u,d}\cdot g_{u,d,o}(k_d)/c_d,\ q_{u,d,o}(k_d)/c_d + \alpha^{\text{a}}_{u,d,o}(k_d)\Big)\ , \tag{31}$$

and adding upper bound constraint $0\le n_{u,d}(k_d)\le C$ to traffic state $n_{u,d}(k_d)$ (number of vehicles in a link) to make sure that the number of vehicles inside a link will never exceed its storage capacity C, i.e. no more vehicles can enter the link when it is already totally congested.

B. S* model-based MPC controller

An MPC controller can be established based on the S* model using the same method as shown in Section V. A similar MILP optimization problem as (28) can be built through reformulating the S* model into an MLD model. But, for the new MILP optimization problem, the number of the auxiliary variables is reduced by half because of the reduction of the S* model. Although the S* model does not take the over-saturated scenario into consideration, it can still be easily guaranteed due to the constraints added. Instead of constraining the average traffic flow rates leaving links, the maximum number of vehicles that the downstream link can accept is constrained by the upper bound. The traffic state $n(k)$, which is the number of vehicles in a link, is already an optimized variable of the MILP optimization problem. Hence, no extra effort is needed to add constraints to the traffic states $n(k)$ of all the links within the network at every simulation time step k. In fact, the key idea of this approach lies in relaxing the optimization problem by reducing one constraint in the control model (the S model), and adding upper bounds to the optimized state variables $n(k)$ instead. The S* model-based optimization problem focuses on the same problem as the S model-based optimization problem, but the S model-based optimization problem has tighter constraints. So the optimization problem of MPC controllers is relaxed and obtains more flexibility in the optimization space.

VII. SIMULATIONS

As a macroscopic urban traffic model, the BLX model of [5] is relatively accurate enough, which is demonstrated in [5]. Therefore, we use the BLX model to simulate the real traffic process, and design MPC controllers to derive control inputs for the BLX model. In order to improve the real-time feasibility of the MPC controller for urban traffic networks, the on-line optimization of the MPC controller is reformulated into different optimization problems, and

these optimization problems are solved by different optimization methods. Multi-start Sequential Quadratic Programming (SQP) and Pattern Search (PS) are applied to solve the S model-based nonlinear optimization problem. To increase the real-time feasibility, both the S model and the S* model are reformulated into MLD models, and optimization problems are constructed and solved by an MILP solver.

The network investigated is a grid network with 4 intersections. The cycle times of the intersections are 120 s or 60 s respectively. The control time interval is set to the Least Common Multiple of all the cycle times in the network, i.e. T_c is 120 s. The prediction horizon is 10 control intervals. The control simulations run for the same time period of 1200s for all the experiments. Constant input traffic flow rates to the network origins are specified as 2000 veh/h. The free-flow speed on the link is 50 km/h. In Table I, "t_{avrg}" is the average optimization CPU time over all the control steps, and "$t_{\max}$" is the maximum optimization CPU time. The cost function is TTS. The number of random initial starting points for both the SQP algorithm and the PS algorithm is 5.

TABLE I

COMPARISON OF COST FUNCTIONS, CPU TIMES, AND THE NUMBER OF OPTIMIZATION VARIABLES FOR DIFFERENT OPTIMIZATION ALGORITHMS

Algorithm	TTS (veh·h)	CPU time (s)		# variables	
		t_{avrg}	$t_{\max}$	Real	Boolean
SQP	1741.5	516.6	736.5	120	-
PS	1743.8	2266.5	2533.6	120	-
S MILP	1736.9	2.7	3.0	6880	1440
S* MILP	1735.0	1.7	2.5	4480	720

Table I shows the simulation results for different optimization algorithms. `fmincon` and `patternsearch` from the optimization toolbox of Matlab are chosen as the solvers for SQP and PS respectively. The SQP algorithm is superior in both performance and CPU time. Two MILP approaches for the reformulated S model and the reformulated S* model are called respectively "S MILP" and "S* MILP" here. CPLEX, implemented through the `cplex` interface function of the Matlab Tomlab toolbox, is used as the MILP solver. Both S MILP and S* MILP are much faster, and also result in lower costs than the nonlinear optimization algorithms. Because the problem at hand is a nonlinear non-convex optimization problem, which has many local optima. SQP searches for the local optimum only, which results in a sub-optimal solution. A multi-start method can be applied to help select a better sub-optimal solution. However, the multi-start procedure also requires more CPU time. On the contrary, an MILP problem can be solved efficiently by existing solvers, which are able to find the global optimum. As Table I shows, S* MILP has less optimization variables than S MILP, where the number of auxiliary variables (real and boolean) is reduced by half because of the model adaptation. Thus, S* MILP takes less CPU time than S MILP. Moreover, due to the constraint relaxation, S* MILP gains more flexibility in the optimization space, and thus has better performance. As a result, MILP methods, especially S* MILP, are suitable optimization approaches for the urban traffic MPC controllers.

VIII. CONCLUSIONS AND FUTURE WORK

Model Predictive Control provides many advantages for controlling and coordinating urban traffic networks. But, due to the nonlinearity of the urban traffic model, the MPC controller will become real-time infeasible, i.e. the optimization cannot be solved on-line, when the network grows larger. Thus, a method is given to reformulate the nonlinear urban traffic model into an MLD model, which only contains linear equations and inequalities, by introducing auxiliary integer variables. The S model and the S* model are both reformulated according to this method. An efficient optimization approach using MILP is applied to solve the reformulated MPC optimization problem. Two MILP optimization problems based on the reformulated S model and S* model are established. For the S* model-based MILP problem, the constraints are relaxed.

The simulation results show that the CPU time is dramatically reduced if the nonlinear optimization problem of the MPC controller is reformulated into an MILP problem. Moreover, the MILP approaches also guarantee a better control performance. Due to the constraints relaxation, the S* model-based MILP approach obtains better optimization results, and is also faster than the S model-based MILP approach.

In the future, extensive simulations will be run to further test the MPC controllers using a microscopic model instead of the macroscopic BLX model to represent or to simulate the real traffic network. This will result in an extensive assessment of the performance of the MILP-based MPC controllers.

IX. ACKNOWLEDGEMENTS

This research is supported by a Chinese Scholarship Council (CSC) grant, the National Science Foundation of China (Grant No. 60674041, 60934007), the European COST Action TU0702, the 7th framework European STREP project "Hierarchical and distributed model predictive control (HD-MPC)" (contract number INFSO-ICT-223854), the BSIK projects "Transition to Sustainable Mobility (TRANSUMO)" and "Next Generation Infrastructures (NGI)", the Delft Research Center Next Generation Infrastructures, and the Transport Research Centre Delft.

REFERENCES

[1] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, "Review of road traffic control strategies," *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2043–2067, 2003.

[2] M. van den Berg, A. Hegyi, B. De Schutter, and J. Hellendoorn, "Integrated traffic control for mixed urban and freeway networks: A model predictive control approach," *Eur. J. Transp. Infrastructure Res.*, vol. 7, no. 3, pp. 223–250, 2007.

[3] S. Lin, B. De Schutter, Y. Xi, and J. Hellendoorn, "A simplified macroscopic urban traffic network model for model-based predictive control," in *Proc. 12th IFAC Symp. Ctrl. Transp. Syst.*, Redondo Beach (CA), USA, 2009, pp. 286–291.

[4] ——, "Study on fast model predictive controllers for large urban traffic networks," in *Proc. 12th Int. IEEE Conf. Intell. Transp. Syst.*, St. Louis (MO), USA, 2009, pp. 691–696.

[5] S. Lin and Y. Xi, "An efficient model for urban traffic network control," in *Proc. 17th IFAC World Congress*, Seoul, Korea, 2008, pp. 14 066–14 071.

[6] E. Camacho and C. Bordons, *Model Predictive Control in the Process Industry*. Berlin, Germany: Springer-Verlag, 1995.

[7] D. Christiansen, *Electronics Engineers' Handbook*, 4th ed. New York: IEEE Press/McGraw Hill, 1997.