Technical report 15-001

Multi-agent model-based predictive control for large-scale urban traffic networks using a serial scheme[*]

Z. Zhou, B. De Schutter, S. Lin, and Y. Xi

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: https://www.dcsc.tudelft.nl

[*] This report can also be downloaded via https://pub.bartdeschutter.org/abs/15_001.html

Multi-Agent Model-Based Predictive Control for Large-Scale Urban Traffic Networks Using a Serial Scheme

Zhao Zhou[a,b], Bart De Schutter[a], Shu Lin[c], Yugeng Xi[b]

[a] *Delft Center for Systems and Control, Delft University of Technology, 2628 CD Delft, The Netherlands.*
[b] *Department of Automation, Shanghai Jiao Tong University, and Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai, 200240, China.*
[c] *School of Computer and Control Engineering, University of Chinese Academy of Science, Beijing, 100049, China.*

Abstract

Urban traffic networks are large-scale systems, consisting of many intersections controlled by traffic lights and interacting connected links. For efficiently regulating the traffic flows and mitigating the traffic congestion in cities, a network-wide control strategy should be implemented. Control of large-scale traffic networks is often infeasible by only using a single controller, i.e. in a centralized way, because of the high dimension, complicated dynamics, and uncertainties of the system. In this paper we propose a multi-agent control approach using a congestion-degree-based serial scheme. Each agent employs a model-based predictive control approach and communicates with its neighbors. The congestion-degree-based serial scheme helps the agents to reach an agreement on their decisions regarding traffic control actions as soon as possible. A simulation study is carried out on a hypothetical large-scale urban traffic network based on the presented control strategy. The results illustrate that this approach has a better performance with regard to computation time compared with the centralized control method and a faster convergence speed compared with the classical parallel scheme.

Keywords: Multi-agent control, model predictive control, large-scale networks, urban traffic networks.

1. Introduction

Traffic congestion in urban road networks creates a series of social-economic problems, such as the excessive consumption of energy, the environmental pollution, and the risk of car accidents. Although this problem can be solved by constructing new transportation infrastructures and extending the road networks, such a solution is both costly and time-consuming. Therefore, a potential solution is to apply network-wide traffic signal control on the basis of the existing transportation infrastructures.

Since the early 1980s, some TUC (Traffic-responsive Urban Control) strategies have been developed to address the congestion problem in road networks. SCOOT [1] and SCATS [2] are two well-known commercial traffic control systems and have been widely used in many

Email addresses: `zzhou553@gmail.com` (Zhao Zhou), `b.deschutter@tudelft.nl` (Bart De Schutter), `lisashulin@gmail.com` (Shu Lin), `ygxi@sjtu.edu.cn` (Yugeng Xi)

big cities around the world. These systems determine the signal timings based on an adaptive control approach responding to the current traffic states and they also use a simple hierarchical control scheme. OPAC [3] and RHODES [4] are real-time traffic-adaptive signal control systems. Their hierarchical structure has been designed to realize different functions at each level, such as coordination and synchronization of traffic signals, interaction control, and network flow control. From the experiment results in the literature, it is shown that these real-time signal control systems are more efficient than the fixed-time control plans and to some extent have a better performance in reducing traffic congestion.

More recently, with the development of computer techniques, a number of more precise model-based control approaches based on macroscopic traffic models that can describe the dynamics of the traffic flow of the whole urban traffic network in a sufficiently detailed way have been proposed, and some numerical solution algorithms are used to solve the optimization problems based on these models. Diakaki et al. [5] presented a multivariable regulator for traffic-responsive coordinated network-wide signal control based on the well-known store-and-forward urban traffic model. They formulated a linear-quadratic optimal control problem and designed the offline feedback regulator to calculate in real-time the signal splits using the measured traffic states from the detectors. Aboudolas et al. [6, 7, 8] further reformulated a quadratic-programming problem and embedded it into a rolling-horizon control scheme. This approach can predict the future traffic states based on the model, realize the rolling optimization and derive the optimal signal timing using the current states of the traffic system. Meanwhile, it can significantly reduce the on-line computational complexity because of the linear optimization problem, which is readily solved by use of the available tools within a few seconds. However, the models used in the approaches mentioned above are usually very simple, which limits the application of these systems. For example, in oversaturated traffic conditions, the performance of the urban traffic networks would deteriorate due to the mismatches between the reality and the models. Lin et al. [9] presented an efficient network-wide model predictive control (MPC) approach for urban traffic networks. MPC [18] is a model-based control strategy in which an optimal control sequence is determined by implementing numerical optimization over a given horizon based on a prediction model. Lin et al. [9] designed the MPC controllers based on a more accurate urban traffic model fully considering the various traffic scenarios. In order to further improve the computational efficiency and make the approach applicable in practice, they reformulated the nonlinear optimization problem into a mixed-integer linear programming (MILP) optimization problem [10]. Additional urban traffic control strategies have also been proposed in [11, 12, 13, 14].

The model-based control strategies developed in the literature can control and coordinate the urban traffic networks in real time; however, they also encounter on-line computation complexity problems when implemented in practice. For a large-scale urban network, the traffic flow system is a complex dynamic process. It is characterized by numerous intersections and roads, which means the dimension of traffic system is very high. The unpredictable activities of vehicles will lead to uncertainty in the traffic states of each road. There also exist complex interactions between connected roads. Hence, the urban traffic system is usually a dynamic large-scale system with multiple inputs and multiple outputs, the complexity of optimization problem increases with the growing scale of the networks, and it will bring high computational burden to design the real-time efficient centralized traffic control methods. Therefore, it is necessary to develop distributed or hierarchical coordination structures to reduce the on-line computational complexity, and make the traffic system more robust to unexpected disturbances.

In recent years, distributed coordination control of multi-agent systems has gained increasing attention in various fields because of its broad applications in e.g. electric power networks [15], sensor networks [16], road networks [17], and so on. Based on the decomposition of the entire network or system, several agents (local controllers) are developed and allocated to the corresponding non-overlapping subsystems, that each agent determines which traffic control actions how to be taken in the subsystem for which it is responsible. Each agent is able to make decisions by negotiating with its neighbors with the aim of achieving the best performance of the whole system.

In this paper, we mainly focus on the communication scheme among the agents. This paper contributes to the state-of-the-art by proposing a novel congestion-degree-based serial scheme for multi-agent control. First of all, the MPC methodology is used to design the local controllers. Then, a congestion-degree-based serial scheme for multi-agent decision making is proposed to coordinate the agents. In this scheme, the sequence of the agents implementing their own optimization depends on the severeness of the traffic conditions in each subnetwork. Through simulation experiment for a large-scale urban traffic network, we show the beneficial properties of the proposed approach compared with other control approaches.

The remainder of this paper is organized as follows. In Section II, a macroscopic traffic model from the literature is introduced and then used as the prediction model for network-wide control. Based on this model, the optimization problem of single-agent (centralized) MPC is formulated. In Section III, we propose a distributed multi-agent MPC approach fully considering the characteristics of the urban traffic network and the complex interactions between subsystems. An experiment using the presented method to control a typical traffic network is provided in Section IV. The comparison with other control methods is also given in this section. Section V concludes this paper and considers the future work.

2. MPC for a single urban traffic subnetwork

2.1. Urban traffic modeling

An urban traffic model is the basis of model-based traffic predictive controllers for urban networks. In this section, the *S* model [9] that can describe the dynamic process of traffic flow in a macroscopic way is briefly introduced, and then used as the prediction model of subnetwork MPC controllers.

An urban traffic network can be considered as a kind of complex network with links and intersections. As shown in Fig. 1, a typical urban road (link $(u,d) \in \mathscr{L}$) is represented by its upstream intersection u $(u \in I_{u,d})$ and downstream intersection d $(d \in O_{u,d})$. The input and output links of link (u,d) can also be indicated by the upstream intersections i_j and the downstream intersections o_j, $j = 1,2,3$. $\alpha_{u,d}^{\text{enter}}(k)$, $\alpha_{u,d}^{\text{arrive}}(k)$, $\alpha_{u,d}^{\text{leave}}(k)$ denote the flow rates of vehicles entering, arriving and leaving link (u,d) at step k, and $q_{u,d}(k)$ is the queue length in link (u,d).

In the *S* model, it is assumed that the cycle time c_{cycle} is equal to the sampling time interval for all intersections. Therefore, the number of vehicles in link (u,d) can be updated by the following conservation equation:

$$n_{u,d}(k+1) = n_{u,d}(k) + (\alpha_{u,d}^{\text{enter}}(k) - \alpha_{u,d}^{\text{leave}}(k)) \cdot c_{\text{cycle}} \tag{1}$$

where the flow rate entering link (u,d) is the sum of the flow rates leaving from its upstream links, i.e.

$$\alpha_{u,d}^{\text{enter}}(k) = \sum_{i \in I_{u,d}} \alpha_{i,u,d}^{\text{enter}}(k) \tag{2}$$

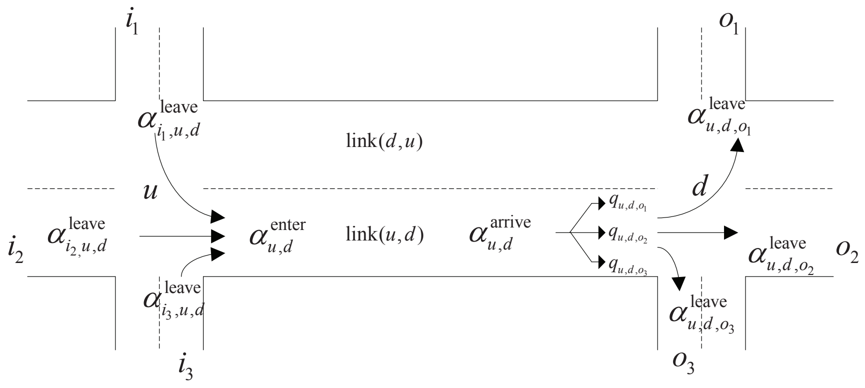

i_1
o_1
$\alpha^{\text{leave}}_{i_1,u,d}$
link(d,u)
$\alpha^{\text{leave}}_{u,d,o_1}$
u
d
q_{u,d,o_1}
i_2
$\alpha^{\text{leave}}_{i_2,u,d}$
$\alpha^{\text{enter}}_{u,d}$
link(u,d)
$\alpha^{\text{arrive}}_{u,d}$
q_{u,d,o_2}
$\alpha^{\text{leave}}_{u,d,o_2}$
o_2
q_{u,d,o_3}
$\alpha^{\text{leave}}_{i_3,u,d}$
$\alpha^{\text{leave}}_{u,d,o_3}$
i_3
o_3

Figure 1: A link between two adjacent intersections

Similarly, the leaving flow rate is equal to the sum of the flow rates leaving for its downstream links, i.e.

$$\alpha^{\text{leave}}_{u,d}(k) = \sum_{o \in O_{u,d}} \alpha^{\text{leave}}_{u,d,o}(k) \tag{3}$$

The leaving average flow rate over c_{cycle} is determined by

$$\begin{aligned}\alpha^{\text{leave}}_{u,d,o}(k) = &\min(\beta_{u,d,o}(k) \cdot \mu_{u,d} \cdot g_{u,d,o}(k)/c_{\text{cycle}},\\ &q_{u,d,o}(k)/c_{\text{cycle}} + \alpha^{\text{arriv}}_{u,d,o}(k),\\ &\beta_{u,d,o}(k)(C_{d,o} - n_{d,o}(k))/c_{\text{cycle}})\end{aligned} \tag{4}$$

where the three terms represent the capacity of the intersection, the number of vehicles waiting and arriving, and the available space in the downstream link, respectively. Moreover, $\beta_{u,d,o}$ is the relative fraction of the traffic turning to o, $\mu_{u,d}$ is the saturated flow rate leaving link (u,d), $g_{u,d,o}$ is the green time length for the traffic stream towards o in link (u,d), $C_{d,o}$ is the capacity of downstream links expressed in number of vehicles, $n_{d,o}$ is the number of vehicles in link (d,o).

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

$$q_{u,d,o}(k+1) = q_{u,d,o}(k) + (\alpha^{\text{arriv}}_{u,d,o}(k) - \alpha^{\text{leave}}_{u,d,o}(k)) \cdot c_{\text{cycle}} \tag{5}$$

Moreover, the S model explicitly considers the situation that vehicles that entered link (u,d) will arrive at the end of the queues after a time delay $\tau(k) \cdot c_{\text{cycle}} + \gamma(k)$, i.e.

$$\begin{aligned}\alpha^{\text{arriv}}_{u,d}(k) = &(1 - \frac{\gamma(k)}{c_{\text{cycle}}}) \cdot \alpha^{\text{enter}}_{u,d}(k - \tau(k))\\ &+ \frac{\gamma(k)}{c_{\text{cycle}}} \cdot \alpha^{\text{enter}}_{u,d}(k - \tau(k) - 1)\end{aligned} \tag{6}$$

$$\tau(k) = \text{floor}\left\{ \frac{(C_{u,d} - q_{u,d}(k)) \cdot l_{\text{veh}}}{N_{u,d}^{\text{lane}} \cdot v_{u,d}^{\text{free}} \cdot c_{\text{cycle}}} \right\},$$
$$\gamma(k) = \text{rem}\left\{ \frac{(C_{u,d} - q_{u,d}(k)) \cdot l_{\text{veh}}}{N_{u,d}^{\text{lane}} \cdot v_{u,d}^{\text{free}} \cdot c_{\text{cycle}}} \right\} \tag{7}$$

where $N_{u,d}^{\text{lane}}$ is the number of lanes in link (u,d), $v_{u,d}^{\text{free}}$ is the free-flow vehicle speed in link (u,d), l_{veh} is the average vehicle length, floor$\{x\}$ means the largest integer that is smaller than or equal to x, and rem$\{x\}$ is the remainder.

After entering the link (u,d), the flow rate of arriving vehicles will reach the tail of waiting queues depending on the turning rates

$$\alpha_{u,d,o}^{\text{arrive}}(k) = \beta_{u,d,o} \cdot \alpha_{u,d}^{\text{arrive}}(k) \tag{8}$$

2.2. Single-agent MPC

Assume that an urban traffic subnetwork is controlled by an agent individually and there is no communication among these agents. The aim of the agent is to generate a set of optimal traffic signal timings according to the current traffic conditions. The corresponding algorithm should be embedded in a rolling-horizon framework so that the optimal control problem can be solved on-line before every control cycle. To this end, MPC is employed by the agent since MPC enables it to implement optimal control repeatedly over a prediction horizon of several steps and to take various constraints into consideration in the optimization.

According to the S model presented in last subsection, the dynamic traffic model for each link can be described as

$$\mathbf{n}(k+1) = \mathbf{f}(\mathbf{n}(k), \mathbf{g}(k), \mathbf{d}(k)) \tag{9}$$

where $\mathbf{n}(k) = [n(k|k)^{\text{T}}, n(k+1|k)^{\text{T}}, \cdots n(k+N_{\text{p}}-1|k)^{\text{T}}]^{\text{T}}$ contains the number of vehicles in each link of the network for time step k up to $k+N_{\text{p}}-1$, $\mathbf{g}(k) = [g(k|k)^{\text{T}}, g(k+1|k)^{\text{T}}, \cdots g(k+N_{\text{p}}-1|k)^{\text{T}}]^{\text{T}}$ contains the future control inputs (the green times of the traffic signals), and $\mathbf{d}(k) = [d(k|k)^{\text{T}}, d(k+1|k)^{\text{T}}, \cdots d(k+N_{\text{p}}-1|k)^{\text{T}}]^{\text{T}}$ contains the predicted disturbances (the traffic demands). Since our goal is to regulate the traffic flows and improve the mobility of the subnetwork, the control objective is to minimize the risk of oversaturation and the number of vehicles in the subnetwork. Therefore, the TTS (Total Time Spent) is used as the objective function. Given the current traffic states at time step k measured from all links in the subnetwork as the initial local state $\mathbf{n}(k)$, and the local known traffic demand $\mathbf{d}(k) = [d(k|k)^{\text{T}}, d(k+1|k)^{\text{T}}, \cdots d(k+N_{\text{p}}-1|k)^{\text{T}}]^{\text{T}}$ over the prediction horizon N_{p}, the optimization problem of MPC solved by single agent can be formulated as follows

$$\begin{aligned} \min_{\mathbf{g}(k)} J =& J_{\text{local}}(\mathbf{n}(k), \mathbf{g}(k)) = \sum_{(u,d)\in\mathscr{L}} \sum_{k+1}^{k+N_{\text{p}}} n_{u,d}(k) \cdot c_{\text{cycle}} \\ \text{s.t.}\ & \mathbf{n}(k+1) = \mathbf{f}(\mathbf{n}(k), \mathbf{g}(k), \mathbf{d}(k)) \\ & \Phi(\mathbf{g}(k)) = 0; \\ & \mathbf{g}_{\min} \le \mathbf{g}(k) \le \mathbf{g}_{\max} \end{aligned} \tag{10}$$

where $\Phi(\mathbf{g}(k)) = 0$ represents the cycle time constraints for all intersections in the network, $\mathbf{g}(k) = [g(k|k)^{\text{T}}, g(k+1|k)^{\text{T}}, \cdots g(k+N_{\text{p}}-1|k)^{\text{T}}]^{\text{T}}$ is the future control input sequence (the splits

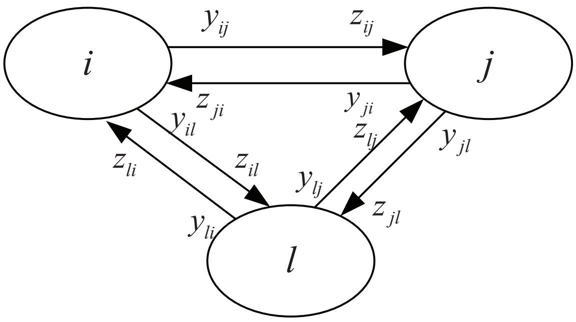


Figure 2: The interactions between subnetworks

of the traffic signals). Once the single agent has solved the optimization problem and obtained the control input sequence over the horizon, it will implement the first sample to the traffic signals. At the next sample step, the agent will receive the new measured values from traffic subnetwork, move the prediction horizon forward with one sample step, and start the optimization again. The moving horizon scheme of MPC does not only guarantee the system to reach a better performance due to the feedback, but it also enhances the robustness of the system in the face of the uncertainties and disturbances of the various urban traffic scenarios.

3. Multi-agent MPC with a serial scheme for large-scale urban traffic networks

3.1. Control problem formulation

For a large-scale urban traffic network, single-agent MPC (centralized MPC) will bring high computational complexity and low robustness. Therefore, multi-agent MPC is increasing more and more attention and it is being applied to many different fields. Based on partition of the network, the whole system is first decomposed into several subsystems [19, 20]. The neighbors of each subnetwork are determined to be the agents of the subnetworks directly connected to the given subnetwork. In a multi-agent MPC scheme each subsystem will be assigned an agent, and each agent employs MPC to determine the control actions for its subsystem by solving a low-dimensional optimization problem. Meanwhile, each agent can receive information from its neighbors through communication. This information influences the decision making of each agent. This scheme makes all agents reach an agreement on taking actions that yield a better performance for the whole system. In this paper, a novel multi-agent MPC scheme so-called congestion-degree-based serial scheme is proposed to deal with the large-scale urban traffic networks control problem.

For illustration purpose, we consider an urban traffic network that is divided into three subnetworks, i, j, and l, as shown in Fig. 2 (this approach can be extended easily for four or more subnetworks). For subsystems i, j, l, the optimization problem can be expressed as

- Subsystem i:

$$
\begin{aligned}
\min_{\mathbf{g}_i(k)} & J_i = J_{\text{local},i}(\mathbf{n}_i(k), \mathbf{g}_i(k)) \\
\text{s.t. } & \mathbf{n}_i(k+1) = \mathbf{f}(n_i(k), \mathbf{g}_i(k), \mathbf{d}_i(k), \mathbf{z}_{ji}(k), \mathbf{z}_{li}(k)); \\
& \Phi_i(\mathbf{g}_i(k)) = 0; \\
& \mathbf{g}_{i,\min} \leq \mathbf{g}_i(k) \leq \mathbf{g}_{i,\max}; \\
& \mathbf{y}_{ij}(k) = \mathbf{f}_{i,j}(\mathbf{n}_i(k), \mathbf{g}_i(k), \mathbf{d}_i(k)); \\
& \mathbf{y}_{il}(k) = \mathbf{f}_{i,l}(\mathbf{n}_i(k), \mathbf{g}_i(k), \mathbf{d}_i(k));
\end{aligned}
\tag{11}
$$

- Subsystem j:

$$
\begin{aligned}
\min_{\mathbf{g}_j(k)} & J_j = J_{\text{local},j}(\mathbf{n}_j(k), \mathbf{g}_j(k)) \\
\text{s.t. } & \mathbf{n}_j(k) = \mathbf{f}(n_j(k), \mathbf{g}_j(k), \mathbf{d}_j(k), \mathbf{z}_{ij}(k), \mathbf{z}_{lj}(k)); \\
& \Phi_j(\mathbf{g}_j(k)) = 0; \\
& \mathbf{g}_{j,\min} \leq \mathbf{g}_j(k) \leq \mathbf{g}_{j,\max}; \\
& \mathbf{y}_{ji}(k) = \mathbf{f}_{j,i}(\mathbf{n}_j(k), \mathbf{g}_j(k), \mathbf{d}_j(k)); \\
& \mathbf{y}_{jl}(k) = \mathbf{f}_{j,l}(\mathbf{n}_j(k), \mathbf{g}_j(k), \mathbf{d}_j(k));
\end{aligned}
\tag{12}
$$

- Subsystem l:

$$
\begin{aligned}
\min_{\mathbf{g}_l(k)} & J_l = J_{\text{local},l}(\mathbf{n}_l(k), \mathbf{g}_l(k)) \\
\text{s.t. } & \hat{\mathbf{n}}_l(k) = \mathbf{f}(n_l(k), \mathbf{g}_l(k), \mathbf{d}_l(k), \mathbf{z}_{il}(k), \mathbf{z}_{jl}(k)); \\
& \Phi_l(\mathbf{g}_l(k)) = 0; \\
& \mathbf{g}_{l,\min} \leq \mathbf{g}_l(k) \leq \mathbf{g}_{l,\max}; \\
& \mathbf{y}_{li}(k) = \mathbf{f}_{l,i}(\mathbf{n}_l(k), \mathbf{g}_l(k), \mathbf{d}_l(k)); \\
& \mathbf{y}_{lj}(k) = \mathbf{f}_{l,j}(\mathbf{n}_l(k), \mathbf{g}_l(k), \mathbf{d}_l(k));
\end{aligned}
\tag{13}
$$

where $\mathbf{y}_{ij}, \mathbf{y}_{il}, \mathbf{y}_{ji}, \mathbf{y}_{jl}, \mathbf{y}_{li}, \mathbf{y}_{lj}, \mathbf{z}_{ij}, \mathbf{z}_{il}, \mathbf{z}_{ji}, \mathbf{z}_{jl}, \mathbf{z}_{li}, \mathbf{z}_{lj}$ are the interaction variables among subnetworks i, j, l. More specifically, $\mathbf{y}_{ji}$, represents the vector of the traffic flows running out of subnetwork j and then into subnetwork i, which can be considered as the influence that agent j has on the control problem of agent i. Moreover, $\mathbf{z}_{ji}$ represents the vector of the traffic flows exiting subnetwork j and entering subnetwork i, which can be seen as the input caused by agent j on the control problem of agent i. It is obvious that the interaction input traffic flow $\mathbf{z}_{ji}$ must be equal to the interaction output traffic flow $\mathbf{y}_{ji}$. Therefore, the interactions between subnetworks will be guaranteed by the following interaction constraints

$$
\begin{aligned}
\mathbf{y}_{ij}(k) = & \quad \mathbf{z}_{ij}(k) \\
\mathbf{y}_{il}(k) = & \quad \mathbf{z}_{il}(k) \\
& \vdots \\
\mathbf{y}_{lj}(k) = & \quad \mathbf{z}_{lj}(k)
\end{aligned}
\tag{14}
$$

However, the interaction constraints cannot be added into the optimization problem of any of the individual agents directly, since each interaction constraint includes two variables from the optimization problem of different agents. Therefore, in order to make sure that the interaction constraints among subnetworks are satisfied, the coordination methodology of multi-agent MPC is developed. Let $\mathscr{N}_m$ be the set of neighbors of agent m. In our case, we have e.g. $\mathscr{N}_i = \{j,l\}$. Moreover, we define

$$\mathbf{y}_{i\mathscr{N}_i}(k) = \left[\begin{array}{c} \mathbf{y}_{ij}(k) \\ \mathbf{y}_{il}(k) \end{array} \right].$$

In a similar way, we can also define $\mathbf{y}_{\mathscr{N}_m m}(k)$, $\mathbf{z}_{m\mathscr{N}_m}(k)$, and $\mathbf{z}_{\mathscr{N}_m m}(k)$ for $m \in \{i,j,l\}$.

3.2. Distributed multi-agent MPC approach

We combine the control problems of three agents (11,12,13) mentioned above and the interaction constraints (14), and then obtain the overall control problem, i.e.,

$$\begin{aligned} &\min_{\mathbf{g}_m(k)} J = \sum_{m\in\{i,j,l\}} J_{\text{local},m}(\mathbf{n}_m(k), \mathbf{g}_m(k)) \\ &\text{s.t. } \mathbf{n}_m(k+1) = \mathbf{f}(n_m(k), \mathbf{g}_m(k), \mathbf{d}_m(k), \mathbf{z}_{\mathscr{N}_m m}(k)); \\ &\quad \Phi_m(\mathbf{g}_m(k)) = 0; \\ &\quad \mathbf{g}_{m,\min} \leq \mathbf{g}_m(k) \leq \mathbf{g}_{m,\max}; \\ &\quad \mathbf{y}_{m\mathscr{N}_m}(k) = \mathbf{f}_{m,\mathscr{N}_m}(\mathbf{n}_m(k), \mathbf{g}_m(k), \mathbf{d}_m(k)); \\ &\quad \mathbf{y}_{\mathscr{N}_m m}(k) = \mathbf{z}_{\mathscr{N}_m m}(k). \end{aligned} \tag{15}$$

Due to the interaction constraints $\mathbf{y}_{\mathscr{N}_m m}(k) = \mathbf{z}_{\mathscr{N}_m m}(k)$, the overall control problem (15) is not separable into three optimization subproblems using only local information of individual agent. In order to handle these interaction constraints, the dual decomposition method (the augmented Lagrangian method) [21, 22, 23] has been introduced to move the interaction constraints into the objective function in the form of adding the Lagrangian multipliers to guarantee the satisfaction of interaction terms as well as additional quadratic terms. Therefore, the Lagrangian function of the overall optimization problem can be written as

$$\begin{aligned} L = &\sum_{m\in\{i,j,l\}} J_{\text{local},m}(\mathbf{n}_m(k), \mathbf{g}_m(k)) \\ &+ \sum_{m\in\{i,j,l\}} \Big(\lambda^{\mathrm{T}}_{\mathscr{N}_m m}(k)(\mathbf{z}_{\mathscr{N}_m m}(k) - \mathbf{y}_{\mathscr{N}_m m}(k)) \\ &+ \frac{c}{2} \|\mathbf{z}_{\mathscr{N}_m m}(k) - \mathbf{y}_{\mathscr{N}_m m}(k)\|_2^2 \Big) \end{aligned} \tag{16}$$

where $\lambda_{\mathscr{N}_m m}(k)$ is the Lagrangian multiplier vector corresponding to the interaction constraint $\mathbf{y}_{\mathscr{N}_m m}(k) = \mathbf{z}_{\mathscr{N}_m m}(k)$, and c is a positive constant.

According to the theory of duality [23], the optimization problem of whole system is equivalent to its dual problem

$$
\begin{aligned}
& \max_{\lambda(k)} (\min_{\mathbf{g}(k)} L) \\
\text{s.t.}\ & \mathbf{n}_m(k+1) = \mathbf{f}(n_m(k), \mathbf{g}_m(k), \mathbf{d}_m(k), \mathbf{z}_{\mathcal{N}_m m}(k)); \\
& \Phi_m(\mathbf{g}_m(k)) = 0; \\
& \mathbf{g}_{m,\min} \leq \mathbf{g}_m(k) \leq \mathbf{g}_{m,\max}; \\
& \mathbf{y}_{m\mathcal{N}_m}(k) = \mathbf{f}_{m,\mathcal{N}_m}(\mathbf{n}_m(k), \mathbf{g}_m(k), \mathbf{d}_m(k)).
\end{aligned} \tag{17}
$$

Once the Lagrangian multipliers $\lambda(k)$ are fixed, the overall dual control problem can be divided into several subproblems. Since the formulation (16) including quadratic terms is non-separable, we approximate it with the following equation (see [24] for more details and a motivation of this approach):

$$
\begin{aligned}
\widetilde{L} &= \sum_{m\in\{i,j,l\}} L_m \\
&= \sum_{m\in\{i,j,l\}} \Big(J_{\text{local},m}(\mathbf{n}_m(k), \mathbf{g}_m(k)) \\
&+ J_{\text{inter},m}(\mathbf{z}_{\mathcal{N}_m m}(k), \mathbf{y}_{m\mathcal{N}_m(k)}, \lambda_{\mathcal{N}_m m}(k), \lambda_{m\mathcal{N}_m}(k)) \Big)
\end{aligned} \tag{18}
$$

where $J_{\text{inter},m}(\cdot)$ is the cost function associated with the interaction variables (see (23) below for the detailed definition).

In order to reduce the number of optimization variables of the control problem of each agent, the interaction prediction principle is used to coordinate the agents. In this principle, the interactions among subnetworks are not disconnected, but are estimated using the information from their neighbors. At every iteration, the input traffic flow of subnetwork m, i.e. $\mathbf{z}_{\mathcal{N}_m m}(k)$, is not considered as a variable that needs to be optimized by agent m, but as a known variable based on the information received from its neighboring subnetworks through communication, e.g. $\mathbf{z}^{s+1}_{\mathcal{N}_m m}(k) = \mathbf{y}^{s}_{\mathcal{N}_m m}(k)$. Therefore, taking the network illustrated in Fig. 2 as an example, the distributed multi-agent MPC approach for large-scale urban traffic networks at each control step k can be described as follows:

I Initialization

Get the current traffic states of the road subnetwork as the initial states for each agent and estimate the expected traffic demand.

II Iteration

The iteration optimization process to make the interaction constraints satisfied is illustrated as follows:

1. Set the iteration step $s = 1$, the Lagrange multipliers $\lambda^{s}_{\mathcal{N}_m m}(k), \lambda^{s}_{m\mathcal{N}_m}(k)$ and the input traffic flow $z^{s}_{\mathcal{N}_m m}(k)$ for each agent $m \in \{i, j, l\}$.
2. Since the coordination operators and the interaction variables are given, agent m can

solve the following optimization problem over the prediction horizon

$$\begin{aligned}
&\min_{\mathbf{g}_m(k)} J_{\text{local},m}(\mathbf{n}_m(k),\mathbf{g}_m(k)) \\
&\qquad + \sum_{\mathscr{N}_m} J_{\text{inter},m}(\mathbf{z}^s_{\mathscr{N}_m m}(k),\mathbf{y}_{m\mathscr{N}_m}(k),\lambda^s_{\mathscr{N}_m m}(k),\lambda^s_{m\mathscr{N}_m}(k)) \\
&\text{s.t. } \mathbf{n}_m(k+1)=\mathbf{f}(n_m(k),\mathbf{g}_m(k),\mathbf{d}_m(k), \\
&\quad \mathbf{z}^s_{\mathscr{N}_m m}(k)); \\
&\quad \Phi_m(\mathbf{g}_m(k))=0; \\
&\quad \mathbf{g}_{m,\min} \leq \mathbf{g}_m(k) \leq \mathbf{g}_{m,\max}; \\
&\quad \mathbf{y}_{m\mathscr{N}_m}(k)=\mathbf{f}_{m,\mathscr{N}_m}(\mathbf{n}_m(k),\mathbf{g}_m(k),\mathbf{d}_m(k));
\end{aligned} \tag{19}$$

and obtain the green time split $\mathbf{g}^{s+1}_m$ and the output traffic flow $\mathbf{y}^{s+1}_{m\mathscr{N}_m}$.

3. Update the Lagrange multipliers by

$$\lambda^{s+1}(k)=\lambda^s(k)+c\mathbf{e}^{s+1}(k) \tag{20}$$

where c is the positive constant representing the update step length, and $\mathbf{e}^{s+1}(k)$ are the errors between the desired traffic flow input $\mathbf{z}^{s+1}_{\mathscr{N}_m m}(k)$ and the real traffic flow supply $\mathbf{y}^{s+1}_{\mathscr{N}_m m}(k)$ from the neighboring subnetworks

$$\begin{aligned}
\mathbf{e}^{s+1}(k) &= \begin{bmatrix} \mathbf{z}^{s+1}_{\mathscr{N}_i i}(k)-\mathbf{y}^{s+1}_{\mathscr{N}_i i}(k) \\ \mathbf{z}^{s+1}_{\mathscr{N}_j j}(k)-\mathbf{y}^{s+1}_{\mathscr{N}_j j}(k) \\ \mathbf{z}^{s+1}_{\mathscr{N}_l l}(k)-\mathbf{y}^{s+1}_{\mathscr{N}_l l}(k) \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{y}^{s}_{\mathscr{N}_i i}(k)-\mathbf{y}^{s+1}_{\mathscr{N}_i i}(k) \\ \mathbf{y}^{s}_{\mathscr{N}_j j}(k)-\mathbf{y}^{s+1}_{\mathscr{N}_j j}(k) \\ \mathbf{y}^{s}_{\mathscr{N}_l l}(k)-\mathbf{y}^{s+1}_{\mathscr{N}_l l}(k) \end{bmatrix}
\end{aligned} \tag{21}$$

4. Move to the next iteration $s+1$ and repeat step (II) (1)-(3) until the interaction balance constraints are satisfied, or the termination condition is reached, e.g. $\|\mathbf{e}^s(k)\|_2 < \varepsilon$, for some $\varepsilon > 0$.

III Implement actions

The agents implement the green time splits to the traffic signals in their subsystems of the road network.

IV Start the process for the next control step

3.3. *Congestion-degree-based serial scheme*

Generally speaking, distributed multi-agent MPC uses a parallel scheme [25, 26, 27] to implement the step II.2. In the parallel scheme, each agent solves its own optimization problem at the same time using the information provided by its neighbors during the last iteration step. In order to improve the rate of convergence, Negenborn et al. [15] proposed a serial scheme and verified that in some cases it has preferable properties in terms of the convergence speed and the quality of the solution. In contrast with the parallel scheme, in the serial scheme one agent after another solves its own control problem by using the local information and the interaction variables that is updated immediately while the other variables are fixed.

In this paper, we develop a serial scheme so-called congestion-degree-based serial scheme to deal with distributed multi-agent MPC for large-scale urban traffic networks. Considering the characteristics of urban traffic networks, the distribution of traffic densities is heterogeneous, i.e. the congestion degree (see (22) below for a formal definition) for each subnetwork is different. Therefore, the priority of the serial sequence for solving the control problems of the agents can be determined by the congestion degree of the subnetworks. Because if one traffic subnetwork is more congested than the others, it is important to execute the optimization by the corresponding agent firstly. The interaction variables are then transmitted to the next agent of the less congested subnetworks. The reason for this is to regulate the traffic flow so as to mitigate the traffic congestion problems in the network as soon as possible and to provide the more accurate interaction information to the other agents.

Hence, it is necessary to design the serial scheme based on the congestion degree of road networks, which is a scalar to reflect the traffic states in real time. Recently, it has been verified by Daganzo et al. [28] and Geroliminis et al. [29, 30] that there exists a well-defined macroscopic fundamental diagram (MFD) with a unimodal and low-scatter relationship between the network vehicle density and the space-mean flow in urban traffic networks. The maximum value of traffic flow appears at the critical point of MFD. With the increase of vehicles in road network, the flow decreases and the traffic conditions become more congested. So the critical point of MFD indicates the optimal condition of traffic flow. Therefore, we can define the congestion degree $d_m(k)$ for the subnetwork m at time step k as follows

$$\begin{aligned} d_m(k) &= \frac{w_m(k)}{w_{\text{critical},m}} \\ w_m(k) &= \frac{\sum_{(u,d)\in\mathscr{L}_m} \frac{n_{u,d}(k)}{C_{u,d}}}{\#\mathscr{L}_m} \end{aligned} \tag{22}$$

where $w_m(k)$ is the current average traffic density measured from the detectors, $w_{\text{critical},m}$ is the critical traffic density, which can be determined off-line, and $\#\mathscr{L}_m$ is the number of links in subnetwork m. The value of d_m decides the sequence of the control problem of the agents, i.e. the higher the congestion degree, the higher the optimization priority.

Given the sequence of the implementation of the agents, for example, $j \rightarrow i \rightarrow l$, the congestion-degree-based serial scheme can be used to realize the coordination in the distributed multi-agent MPC scheme. For the sake of compactness we drop the index k here for variables. At the current iteration step s, agent j then solves its local control problem before agent i, computes the interaction variables $\mathbf{y}_{ji}^{s+1}$ and $\mathbf{y}_{jl}^{s+1}$, and sends the corresponding information to the agent i and l. Then, agent i uses the new information from the agent j and the previous information of the last iteration $s-1$ from agent l to solve the problem (19) using the following additional objective function:

$$\begin{aligned}
J_{\text{inter},i}(\cdot) &= \begin{bmatrix} \lambda_{ji}^{s} \\ \lambda_{li}^{s} \\ -\lambda_{ij}^{s} \\ -\lambda_{il}^{s} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathbf{z}_{ji} \\ \mathbf{z}_{li} \\ \mathbf{y}_{ij} \\ \mathbf{y}_{il} \end{bmatrix} + \frac{c}{2} \left\| \begin{bmatrix} \mathbf{z}_{ij} - \mathbf{y}_{ij} \\ \mathbf{z}_{il} - \mathbf{y}_{il} \end{bmatrix} \right\|_2^2 \\
&= \begin{bmatrix} \lambda_{ji}^{s} \\ \lambda_{li}^{s} \\ -\lambda_{ij}^{s} \\ -\lambda_{il}^{s} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathbf{y}_{ji}^{s+1} \\ \mathbf{y}_{li}^{s-1} \\ \mathbf{y}_{ij} \\ \mathbf{y}_{il} \end{bmatrix} + \frac{c}{2} \left\| \begin{bmatrix} \mathbf{y}_{ij}^{s-1} - \mathbf{y}_{ij} \\ \mathbf{y}_{il}^{s-1} - \mathbf{y}_{il} \end{bmatrix} \right\|_2^2
\end{aligned} \tag{23}$$

where c is the positive scalar appearing in (16) that penalizes the deviation from the interaction variables iterates that were computed by the other agents during the last iteration and by agent i at the current iteration. Then, agent l uses the new information from agents i and j to solve its own optimization problem. From the description above, it is obviously that the congestion-degree-based serial scheme uses both information from the current iteration and from the last iteration. Therefore, we can add one step into the initialization step I and revise the step II.2 as follows:

I.2 Compute the congestion degree for each subnetwork and determine the sequence of the implementation of control problem of the agents.

II.2 According to the sequence, one agent after another, the agent m determines the green time split $\mathbf{g}_m^{s+1}$ and the output traffic flow $\mathbf{y}_{m\mathcal{N}}^{s+1}$ by solving the problem (19) and sends the new values of the output traffic flow to its neighbors.

4. Simulation-based Experiments

To evaluate the effectiveness of the proposed congestion-degree-based serial distributed multi-agent MPC method in urban traffic management, we construct a hypothetical test urban traffic network to assess the performance of the proposed approach and to compare it with other existing control approaches, namely the fixed-time control, the centralized MPC control and the parallel distributed control.

4.1. Scenario

The test network is shown in Fig. 3. There are 55 nodes including 21 source nodes and 34 intersections, and 133 two-way links with length varying from 213 to 366 meters in it. All the links have two lanes. We carry out the simulation by using CORSIM (CORridor SIMulation), C++, and MATLAB. CORSIM is a microscopic traffic simulation software developed by the FHWA (Federal Highway Administration) for analyzing traffic operations. It is able to simulate the dynamics of an urban traffic network consisting of several intersections and allows the use of an external control algorithm. MATLAB is used to solve the rolling-horizon optimization problem. C++ provides the interface between CORSIM and MATLAB.

First of all, the entire urban traffic network should be divided into several subnetworks appropriately. We consider the partition method proposed by Zhou et al. [20] and divide the whole network into three subnetworks, each controlled by an agent, as shown in Fig. 3. The cycle times of the traffic signal are 60s for all intersections and the offsets between two adjacent intersections

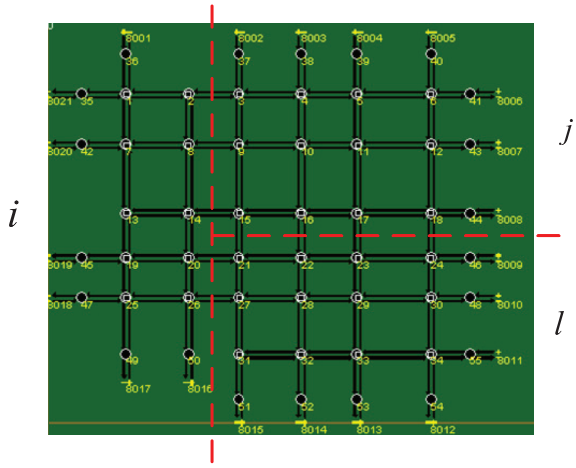


Figure 3: Urban traffic network used for simulation

are 0s during the simulation. These two parameters are constant in our simulations. The total simulation time is 5400s. The control time interval is 180s, thus yielding 30 control steps. The average vehicle length is 5 meters, and the free-flow speed for each link is 50 km/h. At each intersection, the turning rate for each direction is 33.33%. The input traffic flow rates of all the source nodes to the network are equal, and the demand variation is illustrated in Table 1.

Table 1: Network inflow for each source node

Simulation time (s)	Traffic demand flow (veh/h)
0-1800	2000
1800-3600	2500
3600-5400	3000

In the following we compare four control methods:

1. Fixed-time control method, which is a signal control plan that the green time split has been designed for the intersections.
2. A single agent use centralized MPC to control the whole network;
3. The agents of the subnetworks use the parallel distributed multi-agent MPC scheme;
4. The agents of the subnetworks use the congestion-degree-based serial distributed multi-agent MPC scheme.

In order to compare the results and assess the performance of each control plan, five different estimation criteria are considered. The TTS is the total amount of time spent by all the vehicles inside the road network since the beginning of the simulation, including both the vehicles freely running on a link and the vehicles slowing down or waiting in queues:

$$\mathrm{TTS} = \sum_{k=0}^{K_{\mathrm{step}}} \sum_{(u,d)\in\mathscr{L}} T \cdot n_{u,d}(k). \tag{24}$$

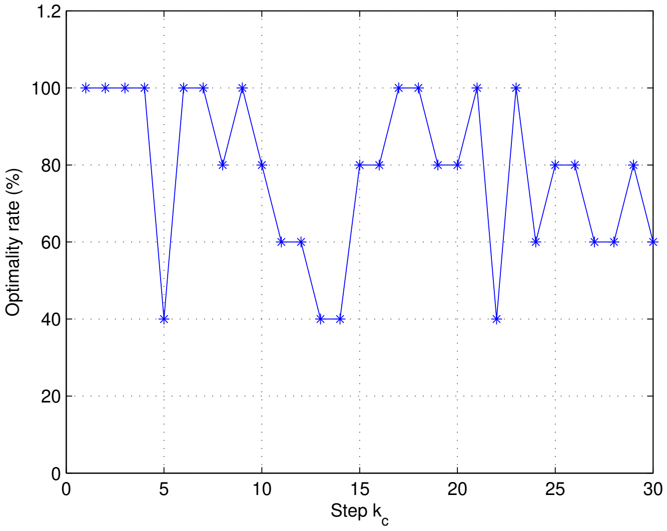


Figure 4: The rate of finding optimal values by using 5 initial points for centralized control

where K_{step} is the time horizon. The TDT (Total Delay Time) is the difference between the total travel time of all vehicles inside the road network since the beginning of the simulation and the total free-flow travel time. So the TDT is actually the total amount of time that the vehicles are delayed. The number of congested links denotes the number of oversaturated links, where a link is considered to be congested if $n_{u,d} \geq 0.7C_{u,d}$.

The parameters we choose for all the control approaches are: the prediction horizon $N_{\text{p}} = 7$, the positive scalar $c = 1$, and the error threshold $\varepsilon = 0.05$.

Since the optimization problem is the non-linear, non-convex problem, the function of *fmincon* of the optimization toolbox of MATLAB is used to calculate the optimal control inputs. Moreover, in order to avoid the optimization ending up on a local minimum, we use the multistart technique to search for a global optimal solution. Different feasible solutions are given as initial starting points, and we run the solver for each initial and record the results. After that, the one corresponding to the lowest objective function value is selected as the optimal solution and applied to the traffic network. With respect to the selection of the number of initial points, Fig. 4 shows the results of we use five initial feasible solutions for each step. The average rate of finding the optimal value is 78%, which is acceptable in our experiment. Hence, we always use 5 initial points.

The selection of the prediction horizon is another important aspect in the MPC. It is noted that an increase of the prediction horizon will improve the performance of the system, but at the same time it will increase the computational complexity, especially for the non-linear, non-convex optimization problem. In a multi-agent MPC setting, an increase of N_{p} will cause more information uncertainties for the other agents. Therefore, more iteration steps will be required. The computation time will increase correspondingly with the increase of the number of iteration steps. For large-scale urban traffic networks, an appropriate prediction horizon N_{p} should be determined to

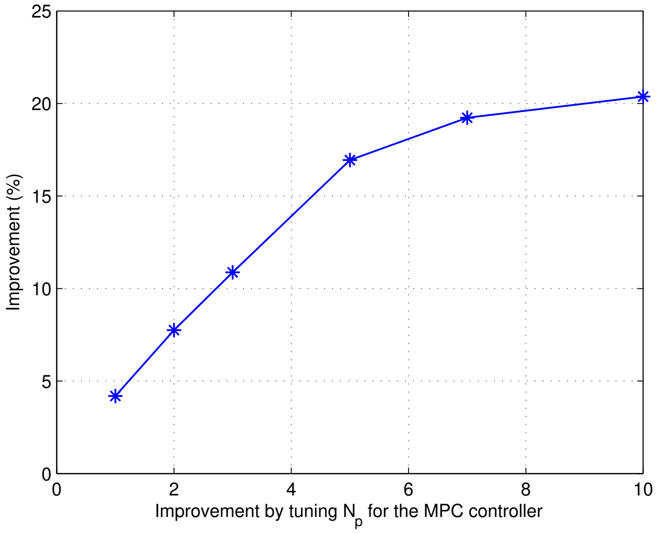


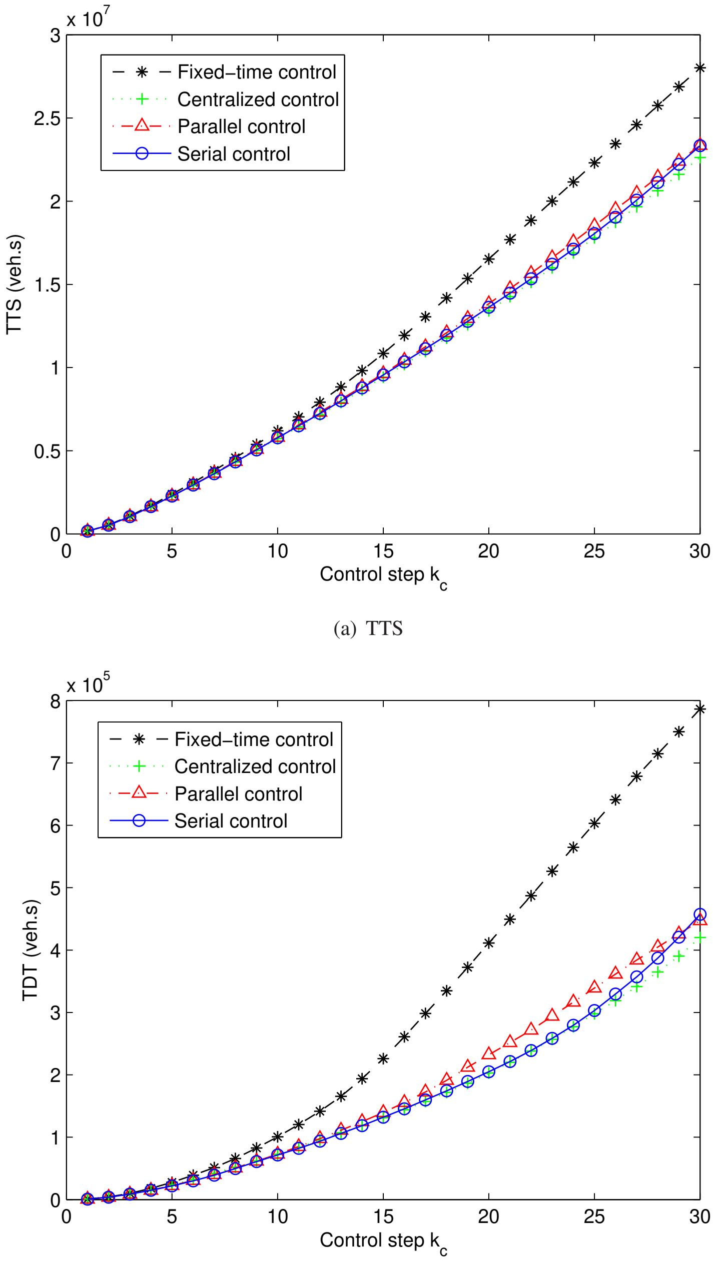
Figure 5: Tuning parameter N_p for the MPC controller

find the trade-off between the performance and the computational cost. We choose several values for N_p and computed the relative improvement in TTS corresponding to the MPC controller compared with the fixed-time control according to the formula $[\%] = (J_\mathrm{MPC} - J_\mathrm{fixed-time})/J_\mathrm{fixed-time}$, where J is computed according to (24). The results are shown in Fig. 5. We can see that the improvement of the performance of the system tends to minor along with the increase of the prediction horizon N_p. Considering the computational cost, we set $N_\mathrm{p} = 7$ for the MPC controller in this paper.

4.2. Simulation results

In this section, the simulation results are presented to explore the efficiency of the multi-agent distributed MPC controllers compared with the fixed-time and centralized control, especially in the congested conditions.

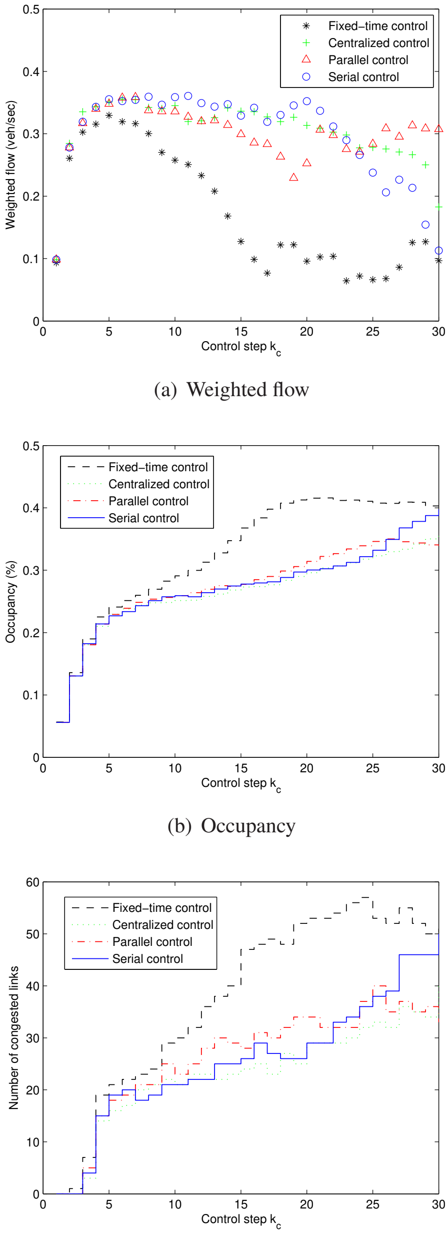
Fig. 6 shows the TTS and the TDT for all the control approaches. From Fig. 6 (a), we can see that the centralized control and the two distributed control approaches yield a greater decrease in the TTS compared with fixed-time control. Before the simulation step 20, the traffic network is in the undersaturated situation. The difference between the three control approaches is not obvious. When the network becomes more congested because of the increasing number of vehicles, the centralized control exhibits a better performance than the other two distributed control approaches. The mean error between the parallel scheme and centralized control is 2.29%, and the maximal error is 4.16%. The mean error between the proposed method and centralized control is 1.20%, and the maximal error is 3.13%. Comparing the typical parallel scheme and the proposed method, our method yields a 1.05% improvement in TTS in the mean error and a 2.56% in the maximal error. This means that the proposed method has a slightly better performance than the parallel scheme. The reason is that our proposed method considers the congestion degree



(a) TTS

(b) TDT

Figure 6: TTS and TDT comparison for fixed-time, centralized, parallel and serial MPC scheme



(a) Weighted flow

(b) Occupancy

(c) Number of congested links

Figure 7: Three indexes under four control strategies

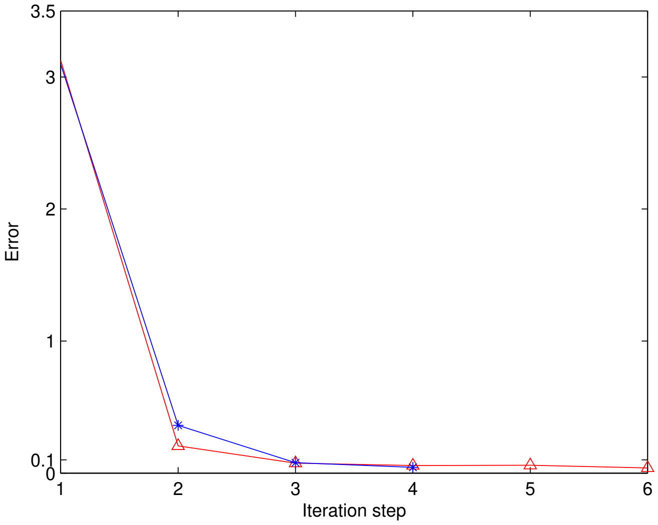


Figure 8: Comparison of the iteration steps

of each subnetwork and determines the implementation sequence of the optimization problem. On the one hand, it can increase the speed of convergence; on the other hand, the agent for the most congested subnetwork executes the optimization firstly to mitigate the traffic congestion and improve the mobility, and through communication, its neighbors will receive more accurate interaction information and perform their own optimization resulting in a more efficient decision-making process. This can be verified in Fig. 7 (a), the weighted flow for the network, which is the weighted average flow of all links in the network, reflecting the degree of mobility in the whole network. We can see that the serial scheme is able to maintain a relative high weighted flow until the later part of the simulation. After the step 25, the traffic network becomes very congested because there is a high demand from the source node, and as a consequence all subnetworks become congested. The performance of our method is worse than the other methods. This can be solved in the future by adding the term of weighted flow into the objective function.

From Fig. 6 (b), we can see that all three methods can reduce the value of TDT. Moreover, the performance of the proposed method is very close to that of the centralized control compared with the parallel scheme. Fig. 7 (b) and Fig. 7 (c) show the evolution of the two indexes: the occupancy and the number of congested links of the network. Comparing to the parallel scheme, the proposed method can approach the performance of the centralized MPC. The detailed performance comparison of the four control methods for entire network is given in Table 2.

The optimization problems are solved in MATLAB 7.11 environment on a computer with Intel Core (TM) I5-2410M 2.3 GHz processor. Finally, we investigate the convergence speed of the two multi-agent distributed control approaches, see Fig. 8. The traffic data collected from the same simulation step is used to evaluate the convergence. The results show that the proposed method converges faster than the parallel scheme. Since the scale of the optimization problem for each agent in the multi-agent distributed control method is reduced, the CPU time for

Table 2: Performance comparison of controllers for entire network

Index	Fixed-time	Centralized	Parallel	Serial
TTS (veh.s)	2.8×10^7	2.26×10^7	2.34×10^7	2.33×10^7
TDT (veh.s)	7.8×10^5	4.2×10^5	4.5×10^5	4.5×10^5
Mean occupancy	33.22%	26.83%	27.69%	27.67%
Mean weighted flow (veh/s)	0.17	0.30	0.29	0.29
# of congested links	38	24	27	26

solving optimization problem is much less than the centralized control. Table 3 gives the average CPU time spent for solving the optimization problem once by the three control approaches, which illustrates the computational complexity of multi-agent distributed control is significantly reduced.

Table 3: Comparison of CPU time spent for three control strategies

Control strategy	CPU time (s)
Centralized	1637.5
Parallel	948.6
Serial	891.2

5. Conclusions

Network-wide traffic control plays an important role in the mitigating and avoiding congestion in urban traffic networks. However, the increasing scale of the traffic networks requires to address some critical issues of such large-scale systems, such as high dimension, multiple objectives, weak robustness, and so on. Multi-agent distributed control strategy is a good way to address these problems. In this paper, based on a partition of the network, we designed an MPC controller for each agent and proposed a congestion degree-based serial scheme to deal with the interactions between agents. A case study was investigated using this approach, which has also been compared with the centralized scheme and the traditional parallel scheme. The simulation results show that the proposed multi-agent distributed control approach can coordinate the agents, make them to reach an agreement on decision making through negotiations, and yield an efficient performance that is comparatively close to the results of the centralized control approach.

In the future, some global optimization algorithms will be explored for solving the non-linear MPC control problem of the agent to obtain a better performance of the traffic system. Possible approaches to reduce the computation time, such as implementation in object code, fast MPC [10], parallel computing and parameterized MPC [32], should also be investigated in order to further reduce the computation time and to make the proposed approach applicable in practice. In addition, other traffic performance objectives such as L^2-norm and L^∞-norm [31] need to be taken into account in the optimization problem to regulate the traffic flow and make the traffic network more homogeneous.

Acknowledgments

This work is supported in part by the China Scholarship Council, the National Science Foundation of China (Grant No. 61203169, 61104160), Shanghai Education Council Innovation Research Project (Grant No. 12ZZ024), Chinese International Cooperation Project of National Science Committee (Grant No. 71361130012), and the NWO-NFSC project 'Multi-level predictive traffic control for large-scale urban networks' (629.001.011), which is partly financed by the Netherlands Organization for Scientific Research (NWO).

[1] Hunt, P.B., Robertson, D.I., Bretherton, R.D., *et al.*: 'The SCOOT on-line traffic signal optimisation technique', Traffic Engineering & Control, 1982, 23, (4), pp. 190-192

[2] Lowrie, P.R.: 'The Sydney coordinated adaptive traffic system-principles, methodology, algorithms', Proceedings of the International Conference on Road Traffic Signalling, London, United Kingdom, March 1982, pp. 67-70

[3] Gartner, N.H.: 'OPAC: A demand-responsive strategy for traffic signal control', Transportation Research Record, 1983, (906), pp. 75-81

[4] Mirchandani, P., Head, L.: 'A real-time traffic signal control system: architecture, algorithms, and analysis', Transportation Research Part C: Emerging Technologies, 2001, 9, (6), pp. 415-432

[5] Diakaki, C., Papageorgiou, M., Aboudolas, K.: 'A multivariable regulator approach to traffic-responsive network-wide signal control', Control Engineering Practice, 2002, 10, (2), pp. 183-195

[6] Aboudolas, K., Papageorgiou, M., Kosmatopoulos, E.: 'Control and optimization methods for traffic signal control in large-scale congested urban road networks', Proceedings of the American Control Conference, New York, USA, July 2007, pp. 3132-3138

[7] Aboudolas, K., Papageorgiou, M., Kosmatopoulos, E.: 'Store-and-forward based methods for the signal control problem in large-scale congested urban road networks', Transportation Research Part C: Emerging Technologies, 2009, 17, (2), pp. 163-174

[8] Aboudolas, K., Papageorgiou, M., Kouvelas, A., *et al.*: 'A rolling-horizon quadratic-programming approach to the signal control problem in large-scale congested urban road networks', Transportation Research Part C: Emerging Technologies, 2010, 18, (5), pp. 680-694

[9] Lin, S., De Schutter, B., Xi, Y., *et al.*: 'Efficient network-wide model-based predictive control for urban traffic networks', Transportation Research Part C: Emerging Technologies, 2012, 24, pp. 122-140

[10] Lin, S., De Schutter, B., Xi, Y., *et al.*: 'Fast model predictive control for urban road networks via MILP', IEEE Transactions on Intelligent Transportation Systems, 2011, 12, (3), pp. 846-856

[11] Lo, H.K.: 'A cell-based traffic control formulation: strategies and benefits of dynamic timing plans', Transportation Science, 2001, 35, (2), pp. 148-164

[12] Papageorgiou, M., Diakaki, C., Dinopoulou, V., *et al.*: 'Review of road traffic control strategies', Proceedings of the IEEE, 2003, 91, (12), pp. 2043-2067

[13] Dotoli, M., Fanti, M.P., Meloni, C.: 'A signal timing plan formulation for urban traffic control', Control Engineering Practice, 2006, 14, (11), pp. 1297-1311

[14] Wang, F.: 'Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications', IEEE Transactions on Intelligent Transportation Systems, 2010, 11, (3), pp. 630-638

[15] Negenborn, R.R., De Schutter, B., Hellendoorn, J.: 'Multi-agent model predictive control for transportation networks: Serial versus parallel schemes', Engineering Applications of Artificial Intelligence, 2008, 21, (3), pp. 353-366

[16] Stankovic, J.A., Abdelzaher, T.F., Lu, C., *et al.*: 'Real-time communication and coordination in embedded sensor networks', Proceedings of the IEEE, 2003, 91, (7), pp. 1002-1022

[17] de Oliveira, L.B., Camponogara, E.: 'Multi-agent model predictive control of signaling split in urban traffic networks', Transportation Research Part C: Emerging Technologies, 2010, 18, (1), pp. 120-139

[18] Mayne, D.Q., Rawlings, J.B., Rao, C.V., *et al.*: 'Constrained model predictive control: Stability and optimality', Automatica, 2000, 36, (6), pp. 789-814

[19] Ji, Y., Geroliminis, N.: 'On the spatial partitioning of urban transportation networks', Transportation Research Part B: Methodological, 2012, 46, (10), pp. 1639-1656

[20] Zhou, Z., Lin, S., Xi, Y.: 'A fast network partition method for large-scale urban traffic networks', Journal of Control Theory and Applications, 2013, 11, (3), pp. 359-366

[21] Bertsekas, D.P.: 'Constrained Optimization and Lagrange Multiplier Methods' (Academic Press, 1982)

[22] Bertsekas, D.P., Tsitsiklis, J.N.: 'Parallel and Distributed Computation: Numerical Methods' (Prentice-Hall Press, 1989)

[23] Boyd, S.P., Vandenberghe, L.: 'Convex Optimization' (Cambridge University Press, 2004)

[24] Batut, J., Renaud, A.: ’Daily generation scheduling optimization with transmission constraints: a new class of algorithms’, IEEE Transactions on Power Systems, 1992, 7, (3), pp. 982-989

[25] Camponogara, E., Jia, D., Krogh, B.H., *et al*.: ’Distributed model predictive control’, IEEE Control Systems, 2002, 22, (1), pp. 44-52

[26] Fawal, H.E., Georges, D., Bornard, G.: ’Optimal control of complex irrigation systems via decomposition-coordination and the use of augmented Lagrangian’, Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, San Diego, USA, October 1998, pp. 3874-3879

[27] Georges, D.: ’Decentralized adaptive control for a water distribution system’, Proceedings of the Third IEEE Conference on Control Applications, Glasgow, UK, August 1994, pp. 1411-1416

[28] Daganzo, C.F.: ’Urban gridlock: macroscopic modeling and mitigation approaches’, Transportation Research Part B: Methodological, 2007, 41, (1), pp. 49-62

[29] Geroliminis, N., Daganzo, C.F.: ’Macroscopic modeling of traffic in cities’, Proceedings of the Transportation Research Board 86th Annual Meeting, Washington DC, USA, January 2007, no. 07-0413

[30] Geroliminis, N., Daganzo, C.F.: ’Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings’, Transportation Research Part B: Methodological, 2008, 42, (9), pp. 759-770

[31] Lin, S., Zhou, Z., Xi, Y.: ’Analysis of Performance Criteria for Model-Based Traffic Congestion Control in Urban Road Networks’, Proceedings of the Transportation Research Board 92nd Annual Meeting, Washington DC, USA, January 2013, no. 13-1775

[32] Zegeye, S., De Schutter, B, Hellendoorn, J., *et al*.: ’Parameterized MPC to reduce dispersion of road traffic emissions’, Proceedings of the American Control Conference, San Francisco, USA, June 2011, pp. 4428-4433