Distributed Algorithm for Optimal Vehicle Coordination at Traffic Intersections

Yuning Jiang∗ Mario Zanon∗∗ Robert Hult∗∗ Boris Houska∗

∗ School of Information Science and Technology, ShanghaiTech University, Shanghai, China (jiangyn, borish@shanghaitech.edu.cn)
∗∗ Department of Signals and Systems, Chalmers University of Technology, Göteborg, Sweden (mzanon, robert.hult@chalmers.se)

Abstract: Automated vehicle coordination can be used to control vehicles across traffic intersections safely and efficiently. This paper proposes a novel parallelizable algorithm, which solves the coordination problem at traffic intersections under a given precedence order by using a tailored variant of the augmented Lagrangian based alternating direction inexact Newton method (ALADIN). Here, each vehicle solves its own optimal control problem and exchanges information about arrival and departure times at the intersection with its neighbors such that collisions are avoided. We illustrate the performance of ALADIN by analyzing two scenarios, one during rush hour and one at low-traffic conditions.

Keywords: Autonomous Vehicles, Traffic Control, Optimal Control, Distributed Optimization

1. INTRODUCTION

Traffic jams are a persistent problem in cities around the world as they account for up to 40% of the total accidents and 20% of the fatalities (Chen et al. (2016)). Automated coordination of vehicles at traffic intersection is a technology that aims at regulating the traffic in order to reduce traffic jams (Azimi et al. (2011); Dresner et al. (2004); Steinmetz et al. (2014); Campos et al. (2013); Lee et al. (2012)). Moreover, the coordination of automated vehicles has the potential to dramatically reduce the number of accidents, and improve both energy efficiency and infrastructure utilization. For these reasons, a large research effort has recently been devoted to intersection management and an excellent survey on the state of the art can be found in Chen et al. (2016). In this paper, we focus on fully automated intersections in a distributed framework. For the sake of simplicity, we consider the deterministic case, in which all agents communicate and cooperate with each other. Extensions of our approach to account for e.g. pedestrians with uncertain behaviour are left for future research.

By introducing communication between automated vehicles at an intersection, one obtains a cooperative multi-agent system where each vehicle is an agent, which can communicate with its neighbors. The overall coordination problem can be separated into two interdependent problems (Hult et al. (2016)). First, the precedence problem consists in deciding the order in which the vehicles should cross the intersection. Second, an optimal control problem is solved to compute the control actions guaranteeing that all vehicles cross the intersection optimally in the prescribed order and without collision. In this paper, we focus on the second problem assuming that the precedence order is given. The problem is then reduced to a structured optimal control problem.

For solving this structured control problem, in practice, a fast and efficient distributed nonlinear programming (NLP) solver is needed. Here, one way to construct such an NLP solver starts by using a sequential quadratic programming (SQP) (Nocedal et al. (2006)) solver. Most of the operations of an SQP method can be parallelized such as the function and derivative evaluation. The QPs are however coupled. In Hult et al. (2016) it is suggested to formulate a bilevel problem comprising decoupled finite horizon optimal control problem with linear dynamics in the lower level, as well as a non-convex upper level coordination problem. The upper level problem is then solved by using an SQP method. This framework has the advantage that the vehicles only communicate via the upper level, whose optimization variables are parameterizing the time schedule for passing the intersection. A technical disadvantage of this approach, however, is that the lower-level constrained optimization problems have non-differentiable objective values with respect to the upper-level parameters—a general problem in bilevel optimization. Of course, one could try to apply heuristics or systematic methods from the field of bilevel optimization or mathematical programming with equilibrium constraints (Bard (1999); Facchinei (1999)), which can deal with this issue. However, an empirical observation with such heuristics is that they are hard to tune often leading to quite non-reliable solvers. Another path, which is followed in the current paper, is to avoid bilevel structures completely by developing tailored distributed optimization solvers. More precisely, this paper uses a variant of the alternating direction augmented Lagrangian based inexact Newton method (ALADIN), which has originally been proposed in Houska et al. (2016). In contrast to Dual Decomposition (Neccara et al. (2008)) and ADMM (Boyd et al. (2011)), which can deal with convex optimization problems only, ALADIN can be applied to nonconvex optimization problems, too.

Initial
attempts to apply ALADIN in the context of nonlinear optimal control can be found in Kouzoupis et al. (2016).

This paper is structured as follows. Section 2 discusses how to formulate and discretize optimal traffic control problems. The main contribution of this paper is presented in Section 3, which proposes a tailored ALADIN algorithm for solving this control problem in a distributed way.

Similar to the approach by Hult et al. (2016) the vehicles communicate their time schedules only, but the proposed algorithm avoids bilevel problems and, more importantly, reduces the communication overhead. Section 4 illustrates the performance of the proposed method for two scenarios one during rush hour the other during low-traffic conditions. Section 5 concludes the paper.

Notation: The notation $\mathbb{Z}_{z_1}^{z_2} = \{ z \in \mathbb{Z} | z_1 \leq z \leq z_2 \}$ is used to denote integer ranges.

2. OPTIMAL TRAFFIC CONTROL

This section discusses how to model and control the traffic at an intersection by using distributed optimization.

2.1 Continuous-time model

One of the simplest possible differential equations for modeling vehicles is a linear system of the form

$$\dot{x}_i(t) = Ax_i(t) + Bu_i(t), \quad A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

where $x_i(t) = (p_i(t), v_i(t))^T$ denotes the position and velocity of the $i$-th vehicle and $u_i(t)$ a control input at time $t$. Let $t_i^{in}$ denote the time at which vehicle $i$ enters the intersection and $t_i^{out}$ the time when it leaves the intersection. A simple collision avoidance strategy can be implemented by enforcing constraints of the form

$$p_i(t_i^{in}) = p_i^{in}, \quad p_i(t_i^{out}) = p_i^{out}, \quad t_i^{out} \leq t_{i+1}^{in}.$$

Here, as shown in Figure 1, $[d_i^{in}, d_i^{out}] \subseteq \mathbb{R}$ denotes the intersection interval. Note that this constraint structure is based on the assumption that the precedence order of the vehicles is pre-determined, i.e., vehicle $i$ has to pass the intersection before vehicle $i + 1$. The traffic control problem of our interest can now be written in the form

$$\min_{x,u,T} \quad \sum_{i=1}^{M} \frac{1}{\tau_i} \int_0^{\tau_i} \|x_i(t) - x_i^{ref}\|^2_{Q_i} + \|u_i(t)\|^2_{R_i} dt \quad \forall i \in \mathbb{Z}_{z_1}^{z_2}, \quad \forall t \in [0, \tau_i],$$

$$\dot{x}_i(t) = Ax_i(t) + Bu_i(t),$$

$$C_i x_i(t) + D_i u_i(t) + b_i \leq 0,$$

$$x_i(0) = \hat{x}_i,$$

$$\left[ x_i(t_i^{in}) \right]_1 = d_i^{in}, \quad \left[ x_i(t_i^{out}) \right]_1 = d_i^{out},$$

$$t_i^{out} \leq t_{i+1}^{in}.$$

Here, the time points $T = (T_1, \ldots, T_M), T_i = (t_i^{in}, t_i^{out})$, at which the vehicles enter or leave the intersection are free optimization variables. Moreover, throughout this paper, we consider quadratic tracking objectives with weighting matrices $Q_i \in \mathbb{R}_{+}^{n \times n}$ and $R_i \in \mathbb{R}_{+}^n$ and reference $x_i^{ref}$. Notice that the objective contribution of all vehicles is scaled by $\tau_i^{out}$ such that the vehicles have no incentive to speed-up in order to reduce their contribution to the overall objective value. As an alternative, one could also augment the problem formulation (2) with quadratic end-cost terms in order to take an approximation of the infinite-horizon cost into account. The initial positions and velocities of all vehicles, denoted by $\hat{x}_i$, are assumed to be given, too. The matrices $C_i \in \mathbb{R}^{n \times 2}$ and vectors $D_i, b_i \in \mathbb{R}^n$ can be used to model position, velocity, and acceleration constraints.

Notice that at larger intersections, especially during rush hours, there may be hundreds of automated vehicles passing in a relatively short time window. In such scenarios $M$ is large and the collision avoidance constraints of the form $t_i^{out} \leq t_{i+1}^{in}$ introduce a coupling. This constraint requires communication and coordination between the vehicles. In this context, another principal challenge is that in a practical scenario there is an inflow of new vehicles entering the neighborhood of the intersection while other vehicles are leaving this region again. However, it would be unnatural to pick one selected vehicle to act as a central unit, since this vehicle might leave the neighborhood of the intersection soon afterwards. Consequently, the focus of this paper is on solving (2) in a distributed way, where each vehicle optimizes its own trajectory and communicates with its neighbors only.

2.2 Discretization

In order to discretize (2) a direct single shooting method with piecewise constant control discretization is applied (Sargent et al. (1978)). Here, we denote by

$$h_k(T_i) = \begin{cases} \frac{t_i^{in}}{N_1}, & \text{if } k < N_1^1 \\ \frac{t_i^{out} - t_i^{in}}{N_1^1}, & \text{otherwise} \end{cases}$$

the length of the shooting intervals with the number of discretization intervals $N_1^1$ and $N_1^1 - N_1^1$. Notice that the differential equations for the vehicles can be solved explicitly, i.e., no numerical integration is needed. The
solution of differential equation (1) at the shooting nodes is given by the explicit expression
\[ \xi_{i,k}(a_i,T_i) = g_{k,0}(T_i) \xi_i + \sum_{j=1}^{k} g_{k,j}(T_i) B_j(t) a_{i,j}, \]  
where \( a_{i,k} \) denotes the control discretization parameters, 
\[ g_{k+1,j}(T_i) = \begin{pmatrix} 1 & h_j(T_i) \\ 0 & 1 \end{pmatrix} g_{k,j}(T_i) \]  
with \( g_{j,j}(T_i) = I \), 
for all indices \( k, j, k \geq j \), and 
\[ B_j(T_i) = \begin{pmatrix} 0.5 h_j(T_i)^2 \end{pmatrix}. \]

A discrete-time problem (2) is now given by
\[
\begin{align*}
\min_{a,t} \quad & \sum_{i=1}^{M} \frac{1}{t_{i+1}^0} \sum_{k=0}^{N_i-1} \left\{ \xi_{i,k}(a_i,T_i) \right\} \\
\text{s.t.} \quad & C_i \xi_{i,k}(a_i,T_i) + D_i a_{i,k} + b_i \leq 0, \\
& [\xi_{i,N_i}(a_i,T_i)]_1 = d_i^{in}, \\
& [\xi_{i,N_i}(a_i,T_i)]_1 = d_i^{out}, \\
& f_i(a_i, \tau_i) = \begin{pmatrix} \xi_{i,k}(a_i,T_i) \\ a_{i,k} \\ 1 \end{pmatrix}, \\
& \text{with shorthands} \]
\[
Q_k(T_i) = \frac{h_k(T_i)^3}{3} A^T Q_i A + \frac{h_k(T_i)^2}{2} (A^T Q_i + Q_i A) + h_k(T_i) Q_i, \\
R_k(T_i) = \frac{h_k(T_i)^5}{20} B^T A^T Q_i A B + \frac{h_k(T_i)^4}{8} B^T A^T Q_i B \\
+ \frac{h_k(T_i)^3}{8} B^T Q_i A B + \frac{h_k(T_i)^2}{3} B^T Q_i B + h_k(T_i) R_i, \\
S_k(T_i) = \frac{h_k(T_i)^4}{8} A^T Q_i A B + \frac{h_k(T_i)^3}{6} Q_i A B \\
+ \frac{h_k(T_i)^3}{3} A^T Q_i B + \frac{h_k(T_i)^2}{2} Q_i B, \\
q_k(T_i) = -\frac{h_k(T_i)^2}{2} A^T Q_i x_i^{ref} - h_k(T_i) Q_i x_i^{ref}, \\
r_k(T_i) = -\frac{h_k(T_i)^3}{6} B^T A^T Q_i x_i^{ref} - \frac{h_k(T_i)^2}{2} B^T Q_i x_i^{ref},
\end{align*}
\]

Remark 1. The discrete-time mixed state-control constraint in (4) only enforces approximate feasibility of the original continuous-time optimal control problem in the sense that the trajectory \( x_i(t) \) might violate the constraint between the shooting nodes. However, the constraint violation converges to 0 if \( N_i^1 \to \infty \) and \( N_i - N_i^1 \to \infty \). Moreover, even if we use a broad discretization, we know that \( x_i(t) \) is for our control discretization a piecewise quadratic function for which an explicit expression can be found. Thus, in principle, it is possible to simply redefine the constraint vectors \( b_i \) by adding a small margin that depends on \( N_i^1 \) and \( N_i - N_i^1 \) such that feasibility of the continuous-time trajectory can be guaranteed in a mathematically rigorous way.

2.3 Reformulation as a Distributed Problem
In order to solve (4) by a distributed optimization algorithm, the auxiliary variables \( t_i^0 \) are introduced such that
\[ t_i^0 = t_{i+1}^0 \quad \forall i \in \mathbb{Z}_{1}^{M-1}. \]

The coupled time constraints can be then be written in the form
\[ t_i^0 = t_{i+1}^0 \quad \forall i \in \mathbb{Z}_{1}^{M-1}. \]

In the following, we introduce new stacked time variables \( \tau_i = (t_i^0, t_i^{in}, t_i^{out}) \). Next, we introduce the shorthands
\[
L_i(a_i, \tau_i) = \frac{1}{t_i^{out}} \sum_{k=0}^{N_i-1} \Phi_{i,k}(a_i, (t_i^{in}, t_i^{out})), \\
G_i(a_i, \tau_i) = \begin{pmatrix} \xi_{i,N_i}(a_i, (t_i^{in}, t_i^{out})) \\ a_{i,k} \\ 1 \end{pmatrix}, \\
F_i(a_i, \tau_i) = \begin{pmatrix} \xi_{i,N_i}(a_i, (t_i^{in}, t_i^{out}))_1 + d_i^{in}, \\
- \xi_{i,N_i}(a_i, (t_i^{in}, t_i^{out}))_1 - d_i^{out} \end{pmatrix}, \\
\text{as well as} \]
\[
\Phi_i(\tau_i) = \min_{a_i} L_i(a_i, \tau_i) \quad \text{s.t.} \quad F_i(a_i, \tau_i) = 0, \]
Here, an extended value notation is used; that is, \( \Phi_i \) takes the value \(+\infty\) if the constraints in the above minimization problem are infeasible. By using this notation NLP (4) can be written in the equivalent form
\[
\min_{\tau} \quad \sum_{i=1}^{M} \Phi_i(\tau_i) \quad \text{s.t.} \quad t_i^{in} = t_{i+1}^{in} \quad \forall i \in \mathbb{Z}_{1}^{M-1}. \]

Here, \( \lambda_i \in \mathbb{R} \) denotes the multiplier of the \( i \)-th coupling constraint. Note that the above problem is written in the
form of a bilevel optimization problem, since the separable objective functions $\Phi_i$ can only be evaluated by solving (decoupled) optimization problems.

3. DISTRIBUTED TRAFFIC CONTROL ALGORITHM

This section proposes a distributed traffic control algorithm based on a variant of ALADIN (Houska et al. (2016)). Algorithm 1 outlines the main steps for solving problem (13). Notice that the main primal and dual iterates are denoted by $(z_i, a_i)$ and $\lambda_i$. In practice, we initialize $\lambda$ with 0 while the initialization of $z$ is based on an initial estimate of the time schedule. Notice that Step 1 and 2 are completely parallelizable, but Step 3 requires communication. However, fortunately, the QP in Step 3 can be solved by communicating between neighboring vehicles only as explained further below.

Algorithm 1 Distributed Traffic Control Algorithm

Input: Initial guess $z$ and $\lambda$, tuning parameter $\rho$.

Repeat:

(1) Each vehicle solves $z$ and $\lambda$

\[
\min_{a_i, \tau_i} L_i(a_i, \tau_i) - \lambda_i - t_i^\text{in} + \lambda_i t_i^c + \frac{\rho}{2} ||\tau_i - z_i||^2
\]

s.t. \[G_i(a_i, \tau_i) \leq 0 \quad \text{and} \quad F_i(a_i, \tau_i) = 0.\]

(2) Each vehicle chooses $H_i > 0$ and computes $g_i = \rho(z_i - \tau_i) + (\lambda_i - 1, 0, -\lambda_i)^T$.

(3) All vehicles solve the equality constrained QP

\[
\min_{\Delta\tau} \sum_{i=1}^M \frac{1}{2} \Delta\tau_i^\top H_i \Delta\tau_i + g_i^\top \Delta\tau_i
\]

s.t. \[t_i^c + \Delta t_i^c = t_i^\text{in} + \Delta t_i^\text{in} \quad \text{for } i \in Z \setminus \{1\}, i \neq 0 \quad \text{and} \quad t_i^\text{out} + \Delta t_i^\text{out} = t_i^c + \Delta t_i^c, \quad i \in A\]

together by communicating with each other.

(4) Each vehicle updates $z_i \leftarrow \tau_i + \Delta\tau_i$ and $\lambda_i \leftarrow \lambda_i^{QP}$.

3.1 Decoupled Optimal Control Problems

In Step 1 of Algorithm 1 the decoupled discrete-time optimal control problems

\[
\min_{a_i, \tau_i} L_i(a_i, \tau_i) - \lambda_i - t_i^\text{in} + \lambda_i t_i^c + \frac{\rho}{2} ||\tau_i - z_i||^2
\]

s.t. \[G_i(a_i, \tau_i) \leq 0 \quad \text{and} \quad F_i(a_i, \tau_i) = 0.\]

have to be solved, which can be done separately onboard each vehicle and in parallel by using any suitable NLP solver, e.g., an SQP method.

Notice that if (for a given vehicle $i$) the function $\Phi_i$ is differentiable at the solution $\tau_i$ of the above decoupled NLP, then the vector $g_i$ in Step 2 is such that $g_i = \nabla \Phi_i(\tau_i)$, which can be proven by working out the first order KKT optimality condition of the above optimization problem. Thus, Algorithm 1 coincides with the original ALADIN algorithm from Houska et al. (2016) applied to (13). However, notice that the function $\Phi_i$ is not differentiable in general and, consequently, the above derivative-free ALADIN variant is preferable in such scenarios. In addition, if $\Phi_i$ is twice differentiable at $\tau_i$, we may choose a positive definite $H_i$ such that $H_i \approx \nabla^2 \Phi_i(\tau_i)$ in order to improve the local convergence rate. However, also other scaling matrices can be used. The tuning parameter $\rho$ should be sufficiently large as discussed in more detail in Houska et al. (2016).

The local convergence proof from Houska et al. (2016) can be applied one-to-one in order to establish local convergence of Algorithm 1 under the assumption that the minimizer is regular KKT point. On the other hand, Algorithm 1 might not converge, if it is initialized far away from a locally optimal solution, but in this case ALADIN can be augmented with line search routines that guarantee convergence to local minimizers from any starting point for sufficiently large $\rho$ as discussed in Houska et al. (2016), too.

3.2 Communication Step

The arguably most critical step of Algorithm 1 is Step 3, where a coupled QP of the form

\[
\min_{\Delta\tau} \sum_{i=1}^M \frac{1}{2} \Delta\tau_i^\top H_i \Delta\tau_i + g_i^\top \Delta\tau_i
\]

s.t. \[t_i^c + \Delta t_i^c = t_i^\text{in} + \Delta t_i^\text{in} \quad \text{for } i \in Z \setminus \{1\}, i \neq 0 \quad \text{and} \quad t_i^\text{out} + \Delta t_i^\text{out} = t_i^c + \Delta t_i^c, \quad i \in A\]

has to be solved. Here, $A$ denotes the set of indices for which the constraint $t_i^\text{out} \leq t_i^c$ is strictly active at the optimal solution of the decoupled optimal control problems.

Notice that the above QP is always feasible and band-structured, i.e., the solution of this QP can be found by a dynamic programming recursion (Bertsekas (2012)) starting with last vehicle $i = N$ and passing a quadratic cost-to-go function backwards, from vehicle $i$ to vehicle $i - 1$ until the first vehicle is reached. Next, the first vehicle knows the solution and passes it to second vehicle, and so on, until all vehicles know the solution of the above QP. By working out the dynamic programming recursion in detail, it turns out that in total only 2 floating point numbers need to be passed from vehicle $i$ to vehicle $i - 1$ and another single floating point number is passed back from vehicle $i - 1$ to vehicle $i$. Thus, the communication overhead compared to centralized methods such as SQP is reduced significantly.

3.3 Termination

Algorithm 1 should be terminated as soon as all vehicles agree on a collision-free and locally optimal time schedule. Here, a suitable termination criterion is given by

\[
\max_i \left| t_i^c - t_i^\text{in} \right| \leq \epsilon \quad \text{and} \quad \|\tau - z\|_{\infty} \leq \epsilon
\]

for a user-specified tolerance $\epsilon$, see Houska et al. (2016). Checking this condition requires additional communication between the vehicles, however, as they need to agree on when to stop iterating.
In this section, we illustrate the performance of Algorithm 1 for two scenarios, one during rush hour and one at low-traffic conditions. Actually, the proposed algorithm can solve traffic coordination problems within milliseconds and scales up easily to hundreds of vehicles, as it is fully distributed. However, for illustration purposes, we show in the current paper scenarios that involve 10 vehicles only, such that we can reasonably plot the solution. In both scenarios, the intersection is defined by $d_{in} = 0$ m and $d_{out} = 10$ m. Control constraints of the form $-2.0 \frac{m}{s^2} \leq u(t) \leq 2.0 \frac{m}{s^2}$ are enforced. The weighting matrices are set to

$$Q_i = \begin{bmatrix} 0 & 1 & 0 & s \\ 0 & 0 & 1 & \frac{s^2}{m^2} \\ \end{bmatrix}, \quad R_i = 1 \frac{s^4}{m^4},$$

such that there are no position references needed, as the corresponding entries in the $Q$s are set to 0. The reference for the velocities of the vehicles is set to

$$v^{ref} = (80, 80, 65, 70, 70, 60, 70, 75, 80, 85)^T \, \text{km/h}$$
during Scenario 1 and to

$$v^{ref} = (80, 80, 75, 70, 70, 75, 80, 75, 75, 75)^T \, \text{km/h}$$
during Scenario 2. Additionally, we set $N^i_1 = 20$ and $N_i = 25$ for all vehicles. The termination tolerance is set to $\epsilon = 10^{-8}$ and we use $\rho = 250$ for Scenario 1 (rush hour) and $\rho = 1$ for Scenario 2 (low traffic conditions). The algorithm is implemented in Julia-0.4.6. The primal iterates are initialized with the decoupled optimal solution that is obtained without any communication by not enforcing the collision avoidance constraints.

4.1 Optimal trajectories

Figure 2 and Figure 3 show the numerical results for the position trajectories for Scenario 1 and 2, respectively. For the rush hour scenario (Scenario 1) the initial states are set in such a way that all vehicles would collide in the intersection area if collision avoidance constraints would not be enforced. In Scenario 2 the situation is somewhat more relaxed and only a few of the collision constraints are active in the optimal solution.

4.2 Convergence

The upper left part of Figure 4 shows the progress of the residuals of the coupling constraints, the difference between the local primal solution and primal iterates, as well as the $\infty$-norm of the KKT residuals. First row: Scenario 1 (rush hour); second row: Scenario 2 (low-traffic conditions). The upper right part of Figure 4 shows the $\infty$-norm of the violation of the first order necessary KKT optimality conditions of problem (13), denoted by $||r||_{\infty}$. The lower left and the lower right parts of Figure 4 show the same but for Scenario 2. Notice that Algorithm 1 needs more iterations during rush hour (Scenario 1) as the collision avoidance constraints are much more difficult to satisfy. However, in both case the proposed algorithm converges.
reliably to a locally optimal solution. Also notice that a locally quadratic convergence rate can be observed for all scenarios, as exact Hessian approximations were used. Notice that for accuracies less than $10^{-10}$ the algorithm iterates on rounding errors that are due to the finite precision arithmetic.

### 4.3 Comparison versus SQP

![Fig. 5. Convergence of the difference between the primal iterates and optimal solutions: SQP versus Algorithm 1. Left: Scenario 1; Right: Scenario 2.](image)

Figure 5 compares the convergence of $\|z - \tau^\ast\|_\infty$ of Algorithm 1 with a standard SQP algorithm indicating that the proposed method needs fewer iterations to converge. However, the main advantage of Algorithm 1 compared to SQP is that it is fully distributed and that the communication overhead is reduced significantly.

## 5. CONCLUSIONS

This paper has presented a distributed optimization algorithm for solving traffic coordination problems at intersections under the assumption that the precedence order is given. The algorithm is based on a variant of ALADIN that has been tailored for traffic coordination problems, which are non-convex with respect to the parameterization of the time schedule. The performance of the method has been illustrated by applying it to a problem with 10 vehicles for different scenarios finding that the proposed algorithm converges reliably. Generally, in more strongly coupled scenarios, e.g., during rush hour, many collision avoidance constraints may become active and the optimization problem is more challenging to solve. Future work will investigate the robustness of the method with respect to communication failures as well as extensions of the proposed algorithm for real-time model predictive control scenarios.

## ACKNOWLEDGEMENTS

The work of Yuning Jiang and Boris Houska was supported by the National Science Foundation China (NSFC), Nr. 61473185, as well as ShanghaiTech University, Grant-Nr. F-0203-14-012. The work of Mario Zanon and Robert Hult was supported by the Swedish Research Council (VR) under grant number 2012-4038 and the European Commission Seventh Framework under the AdaptIVe, grant agreement number 610428.

## REFERENCES

S.R. Azimi, G. Bhatia, R. Rajkumar, and P. Mudalige. Vehicular networks for collision avoidance at intersections. SAE International Journal of Passenger Cars - Mechanical Systems, 4(1):406–416, 2011.

J.F. Bard. Practical Bilevel Optimization: Algorithms and Applications. Kluwer Academic Publishers, Boston MA, 1999.

D. Bertsekas. Dynamic Programming and Optimal Control. Athena Scientific Dynamic Programming and Optimal Control, 2012.

S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine Learning, 3(1):1–122, 2011.

G. R. de Campos, P. Falcone, and J. Sjoberg. Autonomous cooperative driving: A velocity-based negotiation approach for intersection crossing. In Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on, pages 1456–1461, 2013.

L. Chen and C. Englund. Cooperative Intersection Management: A Survey. In IEEE Transactions on Intelligent Transportation Systems, pages 570–586, Feb 2016.

K. Dresner and P. Stone. Multiagent traffic management: A reservation-based intersection control mechanism. In The Third International Joint Conference on Autonomous Agents and Multiagent Systems, pages 530–537, July 2004.

F. Facchinei, H. Jiang, L. Qi. A smoothing method for mathematical programs with equilibrium constraints. Mathematical Programming, 85(1):107–134, 1999.

R. Hult, M. Zanon, S. Gros, and P. Falcone. Primal Decomposition of the Optimal Coordination of Vehicles at Traffic Intersections. In Proceedings of the 55th IEEE Conference on Decision and Control, 2016.

B. Houska, J. Frasch, M. Diehl. An Augmented Lagrangian Based Algorithm for Distributed Non-Convex Optimization. SIAM Journal on Optimization, Volume 26(2), pages 1101–1127, 2016.

D. Kouzoupis, R. Quiryuen, B. Houska, M. Diehl. A block based ALADIN scheme for highly parallelizable direct optimal control. In Proceedings of the 2016 American Control Conference, Boston, USA, 2016.

J. Lee and B. Park. Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment. IEEE Transactions on Intelligent Transportation Systems, 13(1):81–90, 2012.

I. Necsoara, J.A.K. Suykens. Application of a smoothing technique to decomposition in convex optimization. IEEE Transactions on Automatic Control, 53(11):2674–2679, 2008.

J. Nocedal and S. J. Wright. Numerical Optimization. Springer, 2nd edition, 2006.

E. Steinmetz, R. Hult, G. Rodrigues de Campos, M. Wildemeersch, P. Falcone, and H. Wymeersch. Communication analysis for centralized intersection crossing coordination. In Wireless Communications Systems (ISWCS), 2014 11th International Symposium on, pages 813–818, Aug 2014.

R.W.H. Sargent and G.R. Sullivan. The development of an efficient optimal control package. In J. Stoor, editor, Proceedings of the 8th IFIP Conference on Optimization Techniques (1977), Part 2, Heidelberg, 1978. Springer