# *Safety Verification and Control for Collision Avoidance at Road Intersections*



**Citation:** Ahn, Heejin, and Domitilla Del Vecchio. "Safety Verification and Control for Collision Avoidance at Road Intersections." IEEE Transactions on Automatic Control 63, no. 3 (March 2018): 630–642. © 2018 Institute of Electrical and Electronics Engineers (IEEE)

**As Published:** http://dx.doi.org/10.1109/TAC.2017.2729661

**Publisher:** Institute of Electrical and Electronics Engineers (IEEE)

**Persistent URL:** http://hdl.handle.net/1721.1/119135

**Version:** Original manuscript: author's manuscript prior to formal peer review

# Safety Verification and Control for Collision Avoidance at Road Intersections

Heejin Ahn and Domitilla Del Vecchio



***Abstract*—This paper presents the design of a supervisory algorithm that monitors safety at road intersections and overrides drivers with a safe input when necessary. The design of the supervisor consists of two parts: safety verification and control design. Safety verification is the problem to determine if vehicles will be able to cross the intersection without colliding with current drivers' inputs. We translate this safety verification problem into a jobshop scheduling problem, which minimizes the maximum lateness and evaluates if the optimal cost is zero. The zero optimal cost corresponds to the case in which all vehicles can cross each conflict area without collisions. Computing the optimal cost requires solving a Mixed Integer Nonlinear Programming (MINLP) problem due to the nonlinear second-order dynamics of the vehicles. We therefore estimate this optimal cost by formulating two related Mixed Integer Linear Programming (MILP) problems that assume simpler vehicle dynamics. We prove that these two MILP problems yield lower and upper bounds of the optimal cost. We also quantify the worst case approximation errors of these MILP problems. We design the supervisor to override the vehicles with a safe control input if the MILP problem that computes the upper bound yields a positive optimal cost. We theoretically demonstrate that the supervisor keeps the intersection safe and is non-blocking. Computer simulations further validate that the algorithms can run in real time for problems of realistic size.**

***Index Terms*—safety verification; approximation; hybrid systems; least restrictive control; supervisory control; collision avoidance; intersections; scheduling;**

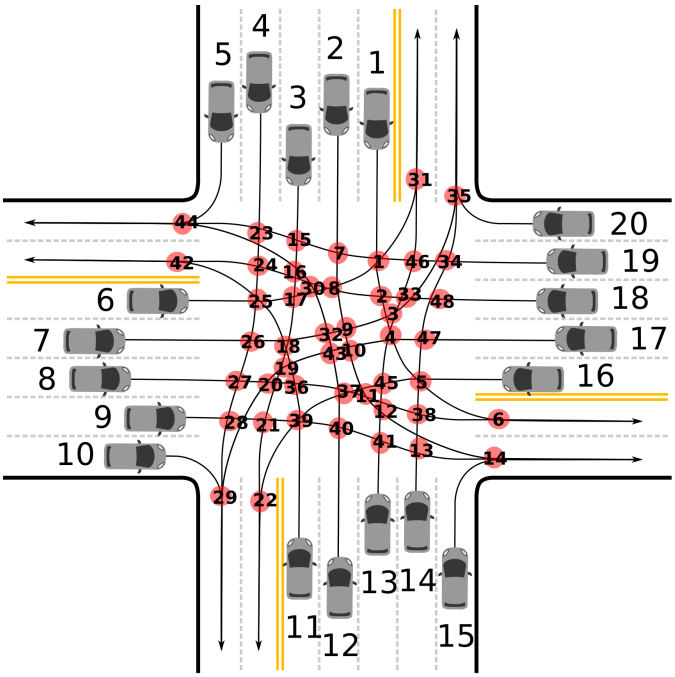


Fig. 1. General intersection. The safety verification problem in this scenario can be approximately solved within quantified bounds. This intersection is obtained from [15] to encompass 20 top crash locations in Massachusetts, USA.

## I. Introduction

THE first fatality caused by a self-driving technology has raised concerns about the safety of autonomous vehicles [1]. As an approach to ensuring safety particularly at a road intersection, this paper proposes the design of a safeguard, called a supervisory algorithm or supervisor. The supervisor monitors vehicles' current states and inputs through vehicle-to-vehicle and vehicle-to-infrastructure communications [2], and determines if their inputs will cause collisions. If this is the case, the supervisor intervenes to prevent collisions.

Informally, we state safety verification as a problem that determines if the state trajectory can be kept outside an unsafe set given an initial condition, where the unsafe set is defined as the set of states corresponding to a collision configuration. Reachability analysis has been used to solve this problem by calculating the reachable region of a system to find a set of initial states that can be controlled to avoid the unsafe set [3]–[5]. However, reachability analysis of dynamical systems with large state spaces is usually challenging due to the complexity of computing reachable sets. This motivates the development of several approximation approaches. One approximation approach is to consider a simpler dynamical model to compute reachable sets instead of using the original complex dynamical model, as studied in [6]–[8]. Another approximation approach is to approximate the original reachable set by employing various geometric representations, which include polyhedra [9], ellipsoids [10], or parallelotopes [11]. It has been shown in [12] that monotonicity of system dynamics, for which state trajectories preserve a partial ordering on states and inputs, makes the reachability analysis relatively simple. This is because for such systems, the boundary of a reachable set can be computed by considering only maximum and minimum states and inputs [13]. Indeed, an exact method for the reachability analysis is presented in [14] for piecewise continuous and monotone systems.

Our approach relies on the monotonicity of the system and on the approximation of vehicle dynamics. In this paper, we consider complex intersection scenarios in which vehicles follow predefined paths as shown in Figure 1. The longitudinal

This work was in part supported by NSF Award #1239182.

Heejin Ahn and Domitilla Del Vecchio are with the Department of Mechanical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, USA. Email: hjahn@mit.edu and ddv@mit.edu

dynamics of vehicles are piecewise continuous and monotone [14], which enables us to translate the safety verification problem to a scheduling problem. This scheduling problem minimizes the maximum lateness and determines if the optimal cost is zero. The zero optimal cost implies that every job (vehicle) can be processed on each machine (conflict area) in time, and equivalently, vehicles can cross the intersection without collision. However, because of the nonlinear second-order dynamics of vehicles, the scheduling problem is a Mixed Integer Nonlinear Programming (MINLP) problem, which is computationally difficult to solve. We therefore estimate the optimal cost of the scheduling problem by formulating two Mixed Integer Linear Programming (MILP) problems that assume first-order dynamics and nonlinear second-order dynamics on a restricted input set, respectively. We prove that these MILP problems yield lower and upper bounds of the optimal cost. These lower and upper bound problems are equivalent to computing over- and under-approximation of the reachable sets of the original problem. These approximation bounds are quantified in this paper.

The scheduling problem has been employed to solve the safety verification problem for collision avoidance at an intersection [16]–[21]. In [16]–[20], the safety verification problem is solved exactly with an assumption that the paths of vehicles intersect only at a single conflict point. Multiple conflict points are considered in [21] and the safety verification problem is solved exactly when vehicle dynamics are restricted to first-order linear dynamics. By contrast, the main contribution of this paper is to solve the safety verification problem on multiple conflict points for general longitudinal vehicle dynamics, which are nonlinear and second-order. A similar problem of robots following predefined paths is considered in [22], [23], but their approach is not designed for safety verification and thus restricted to zero initial speed with the double integrator dynamics. Our scheduling approach can deal with general vehicle dynamics and verify safety at any given state.

Recently, intersection management has been receiving considerable research attention. Most of the recent works concentrate on autonomous intersection management, where a controller takes control of vehicles at all times until they cross the intersection [24]–[30]. Our approach, instead, is to design a least restrictive supervisor in the sense that it overrides drivers only when they cannot avoid a collision.

Collision avoidance for multiple vehicles has been an active area of research mostly in air traffic management. Various approximation approaches are employed to solve collision avoidance problems, such as approximation of dynamics [31]–[36] or relaxation of the original problems [37], [38]. The controllers presented in these works are not least restrictive, as opposed to the controller considered in this paper. While least restrictive controllers are presented in [3], [4], they are applicable only to a small number of vehicles due to the computational complexity of safety verification. As a scalable approach with the number of vehicles, decentralized control is also employed in [39]–[41]. However, decentralized control usually terminates with suboptimal solutions or deadlock. In this paper, we present the design of a centralized controller and prove that it is non-blocking. We validate through computer simulations that this controller can run in real time for realistic size scenarios such as that illustrated in Figure 1.

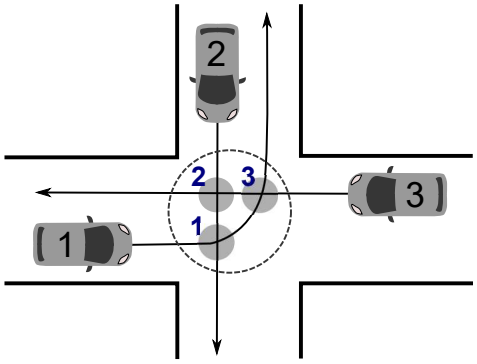



Fig. 2. An intersection is modeled as a set of conflict areas near which two longitudinal predefined paths intersect.

The rest of this paper is organized as follows. In Section II, we define an intersection model and a vehicle dynamic model. The safety verification problem is stated in Section III and translated into a scheduling problem in Section IV. To solve this problem, we formulate the lower and upper bound problems and quantify their approximation bounds in Section V. Based on the upper bound problem, a supervisory algorithm is introduced and proved to be non-blocking in Section VI. We present simulation results in Section VII and conclude the paper in Section VIII.

## II. System Definition

### A. Intersection Model

At a road intersection, vehicles tend to follow predefined paths, which intersect at several conflict points. We define an area around each conflict point accounting for the length of vehicles and call it a conflict area. In this paper, we model an intersection as a collection of all conflict areas. For example, in Figure 2, the intersection is modeled as a set of conflict areas 1-3.

The main focus of this paper is to prevent collisions among vehicles whose paths intersect at conflict areas. Thus, we assume that there is only one vehicle per lane and neglect rear-end collisions. These can be included using a similar approach as used in [17].

### B. Vehicle Dynamical Model

With a vehicle state $(x_j, \dot{x}_j)$ where $x_j \in X_j \subseteq \mathbb{R}$ is the position of vehicle $j$ on its longitudinal path and $\dot{x}_j \in \dot{X}_j := [\dot{x}_{j,min}, \dot{x}_{j,max}] \subset \mathbb{R}$ is the speed, the longitudinal dynamics of vehicle $j$ are described as follows:

$$\ddot{x}_j = f_j(x_j, \dot{x}_j, u_j). \tag{1}$$

The input $u_j$ is the throttle or brake input in the space $U_j := [u_{j,min}, u_{j,max}] \subset \mathbb{R}$.

Let us consider $n$ vehicles approaching an intersection. The whole system dynamics can be obtained by combining the individual dynamics (1) and written as follows:

$$\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{u}), \tag{2}$$

where $\mathbf{x} = (x_1, \dots, x_n) \in \mathbf{X} \subseteq \mathbb{R}^n$ and similarly, $\dot{\mathbf{x}} \in \dot{\mathbf{X}} = [\dot{\mathbf{x}}_{min}, \dot{\mathbf{x}}_{max}] \subset \mathbb{R}^n$, $\mathbf{u} \in \mathbf{U} = [\mathbf{u}_{min}, \mathbf{u}_{max}] \subset \mathbb{R}^n$.

We define an input signal $u_j(\cdot) : t \in \mathbb{R} \mapsto u_j(t) \in U_j$ in the input signal space $\mathcal{U}_j$. Let $x_j(t, u_j(\cdot), x_j(0), \dot{x}_j(0))$ denote the position reached at time $t$ starting from $(x_j(0), \dot{x}_j(0))$ using an input signal $u_j(\cdot) \in \mathcal{U}_j$. We also use the aggregate position $\mathbf{x}(t, \mathbf{u}(\cdot), \mathbf{x}(0), \dot{\mathbf{x}}(0))$ with the aggregate input signal $\mathbf{u}(\cdot) \in \mathcal{U}$. Similarly, we use $\dot{\mathbf{x}}(t, \mathbf{u}(\cdot), \mathbf{x}(0), \dot{\mathbf{x}}(0))$ to denote the speed at time $t$ evolving with $\mathbf{u}(\cdot)$. Regarding these, we make the following assumption.

**Assumption 1.** For all $j \in \{1, \dots, n\}$, the position $x_j(t, u_j(\cdot), x_j(0), \dot{x}_j(0))$ depends continuously on $u_j(\cdot) \in \mathcal{U}_j$, and the input signal space $\mathcal{U}_j$ is path-connected.

We say $u_j(\cdot) \leq u'_j(\cdot) \in \mathcal{U}_j$ if $u_j(t) \leq u'_j(t) \in U_j$ for all $t \geq 0$. We consider $\dot{x}_{j,min} > 0$ to exclude a trivial scenario in which vehicles come to a full stop before an intersection and do not cross it. Most importantly, we assume that the individual dynamics (1) are monotone, that is, they satisfy the following property.

**Assumption 2.** For all $j \in \{1, \dots, n\}$, if $u_j(\cdot) \leq u'_j(\cdot)$, $x_j(0) \leq x'_j(0)$, $\dot{x}_j(0) \leq \dot{x}'_j(0)$, and $t \leq t'$,

$$x_j(t, u_j(\cdot), x_j(0), \dot{x}_j(0)) \leq x_j(t', u'_j(\cdot), x'_j(0), \dot{x}'_j(0))$$

for all $t \geq 0$.

## III. Problem Statement

Let us consider $n$ vehicles approaching an intersection that is modeled as a collection of $m$ conflict areas. We denote the location of conflict area $i$ on the longitudinal path of vehicle $j$ as an open interval $(\alpha_{ij}, \beta_{ij}) \subset \mathbb{R}$. We say a collision occurs at an intersection if two vehicles stay inside the same conflict area simultaneously. This configuration is referred to as a *bad set*, which is denoted by $\mathcal{B}$ and defined as follows:

$$\begin{aligned}\mathcal{B} := \{\mathbf{x} \in X : x_j \in (\alpha_{ij}, \beta_{ij}) \text{ and } x_{j'} \in (\alpha_{ij'}, \beta_{ij'}) \\ \text{for some } i \in \{1, \dots, m\} \text{ and } j \neq j' \in \{1, \dots, n\}\}\end{aligned} \tag{3}$$

The main interest of this paper is safety verification, that is, verifying whether the system can avoid entering the bad set at all future time. We approach this by stating a mathematical problem, called the *safety verification* problem, as follows.

**Problem 1** (safety verification)**.** Given an initial condition $(\mathbf{x}(0), \dot{\mathbf{x}}(0))$, determine if there exists $\mathbf{u}(\cdot) \in \mathcal{U}$ such that $\mathbf{x}(t, \mathbf{u}(\cdot), \mathbf{x}(0), \dot{\mathbf{x}}(0)) \notin \mathcal{B}$ for all $t \geq 0$.

To answer this problem, we need to evaluate all possible input signals, which are functions of time, until finding one satisfying the condition. To avoid this exhaustive and infinite set of computations, we translate this problem to a scheduling problem, which is to find feasible schedules, non-negative real numbers, for vehicles to cross the conflict areas. The rationale behind this translation is that real numbers are computationally less complicated to manipulate than functions. While this scheduling problem is still not easy to solve, we can provide its approximate solutions efficiently.

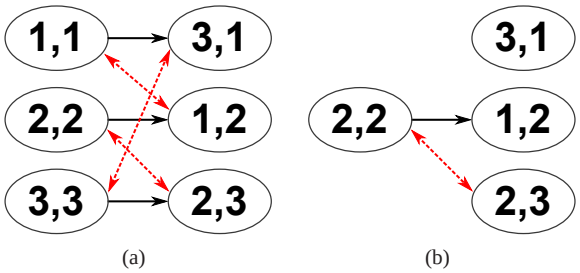



Fig. 3. (a) All operations of the scenario in Figure 2. (b) Operations when $\beta_{11} \leq x_1(0) < \beta_{31}, x_2(0) < \beta_{22}$, and $\beta_{33} \leq x_3(0) < \beta_{23}$. See Example 1 for more details.

## IV. Scheduling: Equivalent Problem to Problem 1

In this section, we formulate a scheduling problem and present the theorem stating that this problem is equivalent to the safety verification problem (Problem 1).

A scheduling problem can be described by a graph representation [42]. For a node, let $(i, j)$ be an operation of vehicle $j$ processing on conflict area $i$. The collection of all operations is denoted by $\bar{\mathcal{N}}$.

$$\begin{aligned}\bar{\mathcal{N}} := \{(i, j) \in \{1, \dots, m\} \times \{1, \dots, n\} : \\ \text{conflict area } i \text{ is on the route of vehicle } j\}.\end{aligned}$$

We also define a set $\mathcal{N} \subseteq \bar{\mathcal{N}}$ that contains operations to be processed given an initial position $\mathbf{x}(0)$.

$$\mathcal{N} := \{(i, j) \in \bar{\mathcal{N}} : x_j(0) < \beta_{ij}\}.$$

Recall that $(\alpha_{ij}, \beta_{ij}) \subset \mathbb{R}$ denotes the location of conflict area $i$ on the longitudinal path of vehicle $j$. Notice that $\mathcal{N}$ is the set of all the operations of interest given $\mathbf{x}(0)$ because $\beta_{ij} \leq x_j(0)$ indicates that vehicle $j$ has exited conflict area $i$.

A first operation set $\mathcal{F} \subseteq \mathcal{N}$ and a last operation set $\mathcal{L} \subseteq \mathcal{N}$ are defined as follows:

$$\begin{aligned}\mathcal{F} &:= \{(i, j) \in \mathcal{N} : \ (i, j) \text{ is the first operation of vehicle } j\}, \\ \mathcal{L} &:= \{(i, j) \in \mathcal{N} : \ (i, j) \text{ is the last operation of vehicle } j\}.\end{aligned}$$

Now, let us define arcs in the graph. We define sets of conjunctive arcs $\mathcal{C}$ and disjunctive arcs $\mathcal{D}$, which connect two operations in $\mathcal{N}$ as follows:

$$\begin{aligned}\mathcal{C} := \{(i, j) \to (i', j) : \text{vehicle } j \text{ crosses conflict area } i \\ \text{and then conflict area } i' \text{ for some } i, j, j'\}, \\ \mathcal{D} := \{(i, j) \leftrightarrow (i, j') : \text{ vehicles } j \text{ and } j' \text{ share} \\ \text{the same conflict area } i \text{ for some } i, j, j'\}.\end{aligned}$$

That is, an element in $\mathcal{C}$ represents the sequence of operations on the path of a vehicle, and an element in $\mathcal{D}$ represents the undetermined sequence of operations on the same conflict area.

**Example 1.** In the scenario in Figure 2, suppose $\beta_{11} \leq x_1(0) < \beta_{31}, x_2(0) < \beta_{22}$, and $\beta_{33} \leq x_3(0) < \beta_{23}$. The operations are illustrated in Figure 3, where the operation sets $\bar{\mathcal{N}}$ and $\mathcal{N}$ become as follows:

$$\begin{aligned}\bar{\mathcal{N}} &= \{(1, 1), (3, 1), (2, 2), (1, 2), (3, 3), (2, 3)\}, \\ \mathcal{N} &= \{(3, 1), (2, 2), (1, 2), (2, 3)\}.\end{aligned}$$

Here, the first and last operation sets are $\mathcal{F} = \{(3,1),(2,2),(2,3)\}, \mathcal{L} = \{(3,1),(1,2),(2,3)\}$. The sets of conjunctive and disjunctive arcs are $\mathcal{C} = \{(2,2) \to (1,2)\}$ and $\mathcal{D} = \{(2,2) \leftrightarrow (2,3)\}$.

We now introduce scheduling parameters, release times, deadlines, and process times, to formulate a jobshop scheduling problem [42].

Given an initial condition $(x_j(0), \dot{x}_j(0))$, let $T_{ij}$ be the time that vehicle $j$ will enter conflict area $i$, that is, $x_j(T_{ij}, u_j(\cdot), x_j(0), \dot{x}_j(0)) = \alpha_{ij}$ for some $u_j(\cdot) \in \mathcal{U}_j$. Let $\mathbf{T}$ be the set of $T_{ij}$ for all $(i,j) \in \mathcal{N}$. A jobshop scheduling problem is to find this set, called a schedule, such that vehicles never meet inside a conflict area.

Given $\mathbf{T}$, the release time $R_{ij}(\mathbf{T})$ is the soonest time at which vehicle $j$ can enter conflict area $i$ under the constraint that it enters the previous conflict area $i'$ at $T_{i'j}$. The deadline $D_{ij}(\mathbf{T})$ is the latest such time. The process time $P_{ij}(\mathbf{T})$ is the minimum time that vehicle $j$ takes to exit conflict area $i$ under the constraint that it enters the same conflict area at time $T_{ij}$ and the next conflict area $i''$ at time $T_{i''j}$. We omit the argument $\mathbf{T}$ if it is clear from context.

Formally, the release time and deadline are defined as follows. Given an initial condition $(\mathbf{x}(0), \dot{\mathbf{x}}(0))$ and $\mathbf{T}$, for all $(i,j) \in \mathcal{N} \setminus \mathcal{F}$, there is a preceding operation $(i',j)$ such that $(i',j) \to (i,j) \in \mathcal{C}$.

$$\begin{aligned} R_{ij}(\mathbf{T}) := \min_{u_j(\cdot)\in\mathcal{U}_j} \{&t : x_j(t, u_j(\cdot), x_j(0), \dot{x}_j(0)) = \alpha_{ij} \\ &\text{with constraint } x_j(T_{i'j}, u_j(\cdot), x_j(0), \dot{x}_j(0)) = \alpha_{i'j}\}, \\ D_{ij}(\mathbf{T}) := \max_{u_j(\cdot)\in\mathcal{U}_j} \{&t : x_j(t, u_j(\cdot), x_j(0), \dot{x}_j(0)) = \alpha_{ij} \\ &\text{with constraint } x_j(T_{i'j}, u_j(\cdot), x_j(0), \dot{x}_j(0)) = \alpha_{i'j}\}. \end{aligned} \tag{4}$$

If the constraint is not satisfied, set $R_{ij} = \infty$ and $D_{ij} = -\infty$. For all $(i,j) \in \mathcal{F}$, such a preceding operation $(i',j)$ does not exists. If $x_j(0) < \alpha_{ij}$,

$$\begin{aligned} R_{ij} &:= \min_{u_j(\cdot)\in\mathcal{U}_j} \{t : x_j(t, u_j(\cdot), x_j(0), \dot{x}_j(0)) = \alpha_{ij}\}, \\ D_{ij} &:= \max_{u_j(\cdot)\in\mathcal{U}_j} \{t : x_j(t, u_j(\cdot), x_j(0), \dot{x}_j(0)) = \alpha_{ij}\}. \end{aligned} \tag{5}$$

If $\alpha_{ij} \le x_j(0)$, set $R_{ij} = D_{ij} = 0$. Notice that for $(i,j) \in \mathcal{F}$, the release time and deadline are independent of $\mathbf{T}$.

The process time is defined as follows. Given an initial condition $(\mathbf{x}(0), \dot{\mathbf{x}}(0))$ and $\mathbf{T}$, for all $(i,j) \in \mathcal{N} \setminus \mathcal{L}$, there is a succeeding operation $(i'',j)$ such that $(i,j) \to (i'',j) \in \mathcal{C}$. If $x_j(0) < \alpha_{ij}$,

$$\begin{aligned} P_{ij}(\mathbf{T}) := \min_{u_j(\cdot)\in\mathcal{U}_j} \{&t : x_j(t, u_j(\cdot), x_j(0), \dot{x}_j(0)) = \beta_{ij} \\ &\text{with constraints } x_j(T_{ij}, u_j(\cdot), x_j(0), \dot{x}_j(0)) = \alpha_{ij} \\ &\qquad \text{and } x_j(T_{i''j}, u_j(\cdot), x_j(0), \dot{x}_j(0)) = \alpha_{i''j}\}. \end{aligned} \tag{6}$$

Set $P_{ij}(\mathbf{T}) := \min_{u_j(\cdot)\in\mathcal{U}_j}\{t : x_j(t, u_j(\cdot), x_j(0), \dot{x}_j(0)) = \beta_{ij}$ with constraint $x_j(T_{i''j}, u_j(\cdot), x_j(0), \dot{x}_j(0)) = \alpha_{i''j}\}$ if $\alpha_{ij} \le x_j(0) < \beta_{ij}$. For all $(i,j) \in \mathcal{L}$, such a succeeding operation $(i'',j)$ does not exist. If $x_j(0) < \alpha_{ij}$,

$$\begin{aligned} P_{ij}(\mathbf{T}) := \min_{u_j(\cdot)\in\mathcal{U}_j} \{&t : x_j(t, u_j(\cdot), x_j(0), \dot{x}_j(0)) = \beta_{ij} \\ &\text{with constraint } x_j(T_{ij}, u_j(\cdot), x_j(0), \dot{x}_j(0)) = \alpha_{ij}\}. \end{aligned} \tag{7}$$

If $\alpha_{ij} \le x_j(0) < \beta_{ij}$, set $P_{ij} := \min_{u_j(\cdot)\in\mathcal{U}_j}\{t : x_j(t, u_j(\cdot), x_j(0), \dot{x}_j(0)) = \beta_{ij}\}$. If $\beta_{ij} \le x_j(0)$, operation $(i,j)$ is not of interest since vehicle $j$ has already crossed conflict area $i$. If the constraints are not satisfied, set $P_{ij} = \infty$.

Using the definitions above, a jobshop scheduling problem is formulated as follows.

**Problem 2** (jobshop scheduling problem)**.** Given an initial condition $(\mathbf{x}(0), \dot{\mathbf{x}}(0))$, determine if $s^* = 0$:

$$s^* := \underset{\mathbf{T},\mathbf{k}}{\text{minimize}} \max_{(i,j)\in\mathcal{N}} (T_{ij} - D_{ij}(\mathbf{T}), 0)$$

subject to

$$\text{for all } (i,j) \in \mathcal{N}, \qquad R_{ij}(\mathbf{T}) \le T_{ij}, \tag{P2.1}$$

$$\text{for all } (i,j) \leftrightarrow (i,j') \in \mathcal{D}, \quad \begin{cases} P_{ij}(\mathbf{T}) \le T_{ij'} + M(1 - k_{ijj'}), \\ P_{ij'}(\mathbf{T}) \le T_{ij} + M(1 - k_{ij'j}), \\ k_{ijj'} + k_{ij'j} = 1. \end{cases} \tag{P2.2}$$

where $\mathbf{T} = \{T_{ij} : (i,j) \in \mathcal{N}\}$, $\mathbf{k} = \{k_{ijj'} \in \{0,1\} :$ for all $(i,j) \leftrightarrow (i,j') \in \mathcal{D}\}$, and $M > 0$ is a large number.

In the scheduling literature, $\max(T_{ij} - D_{ij}(\mathbf{T}), 0)$ is called the maximum lateness. This cost indicates the existence of a schedule that violates the deadline, and thus its minimization is one of the most studied scheduling problems [42].

If $s^* = 0$, we have $T_{ij}$ that satisfies $T_{ij} \le D_{ij}(\mathbf{T})$ for all $(i,j) \in \mathcal{N}$. The fact that $T_{ij}$ is bounded by $R_{ij}(\mathbf{T})$ and $D_{ij}(\mathbf{T})$ encodes the bounds of the input. Constraint (P2.2) says that for two vehicles $j$ and $j'$ that share the same conflict area $i$, either vehicle $j'$ enters it after vehicle $j$ exits (when $k_{ijj'} = 1$) or the other way around (when $k_{ij'j} = 1$). By this constraint, each conflict area is exclusively used by one vehicle at a time. Thus, the existence of such $\mathbf{T}$ and $\mathbf{k}$ that yield $s^* = 0$ is equivalent to the existence of $\mathbf{u}(\cdot) \in \mathcal{U}$ to avoid the bad set. This is the essence of the proof of the following theorem.

**Theorem 1** ( [21])**.** *Problem 1 is equivalent to Problem 2.*

In [21], Problem 2 was introduced as a feasibility problem and the above theorem was proved.

Theorem 1 implies the following: given an initial condition $(\mathbf{x}(0), \dot{\mathbf{x}}(0))$,

$$\begin{aligned} s^* = 0 &\implies \text{a safe input signal exists to avoid } \mathcal{B}, \\ s^* > 0 &\implies \text{no safe input signal exists to avoid } \mathcal{B}. \end{aligned}$$

That is, $s^*$ is the indicator of the vehicles' safety.

While this theorem holds for general dynamics (1), Problem 2 can be difficult to solve depending on which vehicle dynamics are considered. In [21], vehicle dynamics are assumed to be first-order and linear in which case Problem 2 becomes a Mixed Integer Linear Programming (MILP) problem, which can be easily solved by a commercially available solver, such as CPLEX [43]. In this paper, due to the nonlinear and higher order vehicle dynamics, the constraints are nonlinear in $T_{ij}$ and therefore, Problem 2 is a Mixed Integer Non-Linear Programming (MINLP) problem, which is notorious for its computational intractability. To approximately solve

Problem 2, we formulate two MILP problems that yield lower and upper bounds of $s^*$, respectively. The MILP problem that computes the lower bound is a reformulation of the MILP problem given in [21], and the MILP problem that computes the upper bound is based on nonlinear second-order dynamics with a limited input space.

## V. Approximate Solutions to Problem 2

In this section, we provide two MILP problems that yield lower and upper bounds of the optimal cost of Problem 2. Using these bounds, we can quantify the approximation error between the approximate solution and the exact solution to Problem 2.

### A. Lower bound problem

Let us consider first-order vehicle dynamics, that is,

$$\dot{\chi} = \mathbf{v},$$

where $\chi \in \mathbf{X}$ is the vector representing the position of vehicles on their longitudinal paths, and $\mathbf{v}$ is the input. The input $\mathbf{v}$ lies in the space $\dot{\mathbf{X}} = [\dot{\mathbf{x}}_{min}, \dot{\mathbf{x}}_{max}]$ with $\dot{\mathbf{x}}_{min} > \mathbf{0}$. We also define an input signal $v_j(\cdot) : \mathbb{R}_+ \to \dot{X}_j$ in the space $\mathcal{V}_j$ for vehicle $j$.

We define release times and deadlines for the lower bound problem. Process times are considered as decision variables.

**Definition 1.** Given an initial condition $(\mathbf{x}(0), \dot{\mathbf{x}}(0))$ and $\chi(0) = \mathbf{x}(0)$, release times $r_{ij}$ and deadlines $d_{ij}$ are defined as follows.

For all $(i,j) \in \mathcal{F}$, if $x_j(0) < \alpha_{ij}$,

$$r_{ij} := \min_{u_j(\cdot)\in\mathcal{U}_j} \{t : x_j(t, u_j(\cdot), x_j(0), \dot{x}_j(0)) = \alpha_{ij}\},$$
$$d_{ij} := \max_{u_j(\cdot)\in\mathcal{U}_j} \{t : x_j(t, u_j(\cdot), x_j(0), \dot{x}_j(0)) = \alpha_{ij}\}.$$

If $\alpha_{ij} \le x_j(0)$, then $r_{ij} = d_{ij} = 0$.

For $(i,j) \in \mathcal{N}\setminus\mathcal{F}$, there exists a preceding operation $(i',j)$ such that $(i',j) \to (i,j) \in \mathcal{C}$. Given $p_{i'j} \ge 0$ such that $\chi_j(p_{i'j}, v_j(\cdot), \chi_j(0)) = \beta_{i'j}$ for some $v_j(\cdot) \in \mathcal{V}_j$,

$$r_{ij} := p_{i'j} + \frac{\alpha_{ij} - \beta_{i'j}}{\dot{x}_{j,max}}, \quad d_{ij} := p_{i'j} + \frac{\alpha_{ij} - \beta_{i'j}}{\dot{x}_{j,min}}. \tag{8}$$

Notice that for the first operations, $(i,j) \in \mathcal{F}$, release times and deadlines consider general dynamics (1) and thus $r_{ij} = R_{ij}$ and $d_{ij} = D_{ij}$ by (5). This is a different definition from that in [21] and results in a tighter constraint than $r_{ij} = (\alpha_{ij} - \chi_j(0))/\dot{x}_{j,max}$ and $d_{ij} = (\alpha_{ij} - \chi_j(0))/\dot{x}_{j,min}$.

Using Definition 1, the lower bound problem is formulated as a decision problem in which the maximum lateness is minimized.

**Problem 3** (lower bound problem)**.** Given an initial condition $(\mathbf{x}(0), \dot{\mathbf{x}}(0))$, determine if $s_L^* = 0$:

$$s_L^* := \underset{\mathbf{t},\mathbf{p},\mathbf{k}}{\text{minimize}} \max_{(i,j)\in\mathcal{N}} (t_{ij} - d_{ij}, 0)$$

subject to

$$\text{for all } (i,j) \in \mathcal{N}, \quad r_{ij} \le t_{ij}, \tag{P3.1}$$
$$\text{for all } (i,j) \in \mathcal{N}, \quad \frac{\beta_{ij} - \alpha_{ij}}{\dot{x}_{j,max}} \le p_{ij} - t_{ij} \le \frac{\beta_{ij} - \alpha_{ij}}{\dot{x}_{j,min}}, \tag{P3.2}$$
$$\text{for all } (i,j) \leftrightarrow (i,j') \in \mathcal{D},$$
$$\begin{cases} p_{ij} \le t_{ij'} + M(1 - k_{ijj'}), \\ p_{ij'} \le t_{ij} + M(1 - k_{ij'j}), \\ k_{ijj'} + k_{ij'j} = 1. \end{cases} \tag{P3.3}$$

where $\mathbf{t} = \{t_{ij} : (i,j) \in \mathcal{N}\}$, $\mathbf{p} = \{p_{ij} : (i,j) \in \mathcal{N}\}$, $\mathbf{k} := \{k_{ijj'} \in \{0,1\} : \text{for all } (i,j) \leftrightarrow (i,j') \in \mathcal{D}\}$ and $M$ is a large number in $\mathbb{R}_+$.

From (8), we know that the objective function and constraint (P3.1) are linear with the decision variables. Thus, this problem is a Mixed Integer Linear Programming (MILP) problem.

**Theorem 2.** $s_L^* \le s^*$

*Proof:* Suppose Problem 2 finds $\mathbf{T}^*$ and $\mathbf{k}^*$ with the corresponding cost $s^*$, whether or not $s^* = 0$. We will show that $\tilde{\mathbf{t}} = \mathbf{T}^*$ and $\tilde{\mathbf{k}} = \mathbf{k}^*$ become a feasible solution for Problem 3 with some $\tilde{\mathbf{p}}$.

Given $\mathbf{T}^*$, we have $P_{ij}(\mathbf{T}^*)$ for all $(i,j) \in \mathcal{N}$. Consider $\tilde{p}_{ij} = P_{ij}(\mathbf{T}^*)$. From (6) and (7), $P_{ij}(\mathbf{T}^*) - T_{ij}^* = \tilde{p}_{ij} - \tilde{t}_{ij}$ is the time to reach $\beta_{ij}$ from $\alpha_{ij}$ and thus satisfies (P3.2). Constraint (P3.3) is the same as constraint (P2.2) in Problem 2.

We will show that $r_{ij} \le R_{ij}(\mathbf{T}^*)$ and $D_{ij}(\mathbf{T}^*) \le d_{ij}$. As mentioned earlier, for $(i,j) \in \mathcal{F}$, $r_{ij} = R_{ij}$ and $d_{ij} = D_{ij}$. For $(i,j) \in \mathcal{N}\setminus\mathcal{F}$, we have a preceding operation $(i',j)$ such that $(i',j) \to (i,j) \in \mathcal{C}$. By (4), $R_{ij}$ is equal to $T_{i'j}^*$ plus the minimum time to reach $\alpha_{ij}$ from $\alpha_{i'j}$. Considering this definition with (6), $R_{ij}$ is again equal to $P_{i'j}$ plus the minimum time to reach $\alpha_{ij}$ from $\beta_{i'j}$, thereby $R_{ij} \ge P_{i'j} + (\alpha_{ij} - \beta_{i'j})/\dot{x}_{j,max}$. Also, since $P_{i'j} = \tilde{p}_{i'j}$, we have $P_{i'j} + (\alpha_{ij} - \beta_{i'j})/\dot{x}_{j,max} = r_{ij}$. Thus $r_{ij} \le R_{ij}$. Similarly, $D_{ij} \le d_{ij}$. By these inequalities, $R_{ij} \le T_{ij}^*$ implies $r_{ij} \le T_{ij}^* = \tilde{t}_{ij}$, which is constraint (P3.1).

Therefore, $\tilde{\mathbf{t}}, \tilde{\mathbf{p}}$, and $\tilde{\mathbf{k}}$ is a feasible solution of Problem 3. Since $D_{ij} \le d_{ij}$, $s_L^* \le \max(\tilde{t}_{ij} - d_{ij}, 0) \le \max(T_{ij}^* - D_{ij}, 0) = s^*$. ∎

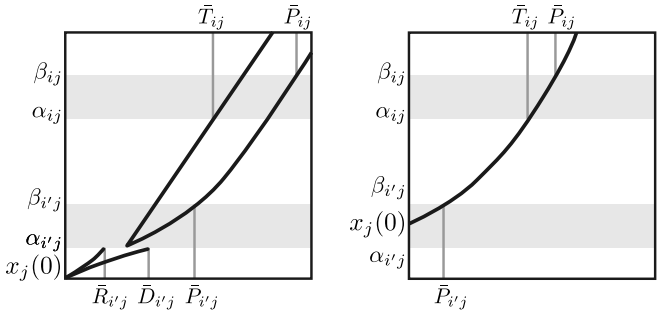
### B. Upper bound problem

In this section, we relax Problem 2 to a MILP problem by considering general dynamics (1) on a restricted input space. This problem is to find a set of times at which vehicles enter their *first conflict area*, assuming that to reach the following conflict areas all vehicles apply maximum input. The rationale here is that once a vehicle enters an intersection, the driver tries to exit as soon as possible.

We define $\alpha_{j,min}$ to denote the first conflict area as follows:

$$\alpha_{j,min} := \min_{(i,j)\in\bar{\mathcal{N}}} \alpha_{ij}.$$

Recall that $\bar{\mathcal{N}}$ is a set of all operations independent of an initial condition.




(a) $x_j(0) < \alpha_{j,min} = \alpha_{i'j}$ (b) $\alpha_{i'j} \le x_j(0) < \beta_{i'j}$

Fig. 4. Illustration of Definitions 2 and 3. Suppose $(i',j) \in \mathcal{F}$ and $(i',j) \to (i,j) \in \mathcal{C}$. We can compute $\bar{P}_{i'j}, \bar{T}_{ij}, \bar{P}_{ij}$ by considering the maximum input inside the intersection.

In the upper bound problem, the time to reach the first conflict area is a decision variable, as opposed to Problems 2 and 3 whose decision variables are the entering times for all conflict areas. This decision variable, called a schedule, is denoted by $\mathbf{T}^{\mathcal{F}} = \{T_{ij}^{\mathcal{F}} : \text{for all } (i,j) \in \mathcal{F}\}$ and defined as $x_j(T_{ij}^{\mathcal{F}}, u_j(\cdot), x_j(0), \dot{x}_j(0)) = \alpha_{ij}$ for some $u_j(\cdot) \in \mathcal{U}_j$ if $x_j(0) < \alpha_{ij}$ and otherwise $T_{ij}^{\mathcal{F}} = 0$.

The release times and deadlines are defined only for the first operation as follows.

**Definition 2.** Given an initial condition $(\mathbf{x}(0), \dot{\mathbf{x}}(0))$, release times $\bar{\mathbf{R}} = \{\bar{R}_{ij} : \text{for all } (i,j) \in \mathcal{F}\}$ and deadlines $\bar{\mathbf{D}} = \{\bar{D}_{ij} : \text{for all } (i,j) \in \mathcal{F}\}$ are defined as follows.

For all $(i,j) \in \mathcal{F}$, if $x_j(0) < \alpha_{j,min}$,

$$\begin{aligned} \bar{R}_{ij} &:= \min_{u_j(\cdot)\in\mathcal{U}_j} \{t : x_j(t, u_j(\cdot), x_j(0), \dot{x}_j(0)) = \alpha_{j,min}\}, \\ \bar{D}_{ij} &:= \max_{u_j(\cdot)\in\mathcal{U}_j} \{t : x_j(t, u_j(\cdot), x_j(0), \dot{x}_j(0)) = \alpha_{j,min}\}. \end{aligned} \tag{9}$$

If $\alpha_{j,min} \le x_j(0) < \alpha_{ij}$,

$$\begin{aligned} \bar{R}_{ij} &:= \min_{u_j(\cdot)\in\mathcal{U}_j} \{t : x_j(t, u_j(\cdot), x_j(0), \dot{x}_j(0)) = \alpha_{ij}\}, \\ \bar{D}_{ij} &= \bar{R}_{ij}. \end{aligned} \tag{10}$$

If $\alpha_{ij} \le x_j(0)$, set $\bar{R}_{ij} = \bar{D}_{ij} = 0$.

Notice that $\bar{R}_{ij} = R_{ij}$ and $\bar{D}_{ij} = D_{ij}$ if $x_j(0) < \alpha_{j,min}$, and $\bar{D}_{ij} = \bar{R}_{ij} = R_{ij}$ if $x_j(0) \ge \alpha_{j,min}$. The release time $\bar{R}_{ij}$ and the deadline $\bar{D}_{ij}$ depend only on an initial condition $(x_j(0), \dot{x}_j(0))$, not on the decision variable $T_{ij}^{\mathcal{F}}$.

Given a schedule $\mathbf{T}^{\mathcal{F}}$, we define $\bar{\mathbf{T}} = \{\bar{T}_{ij} : \text{for all } (i,j) \in \mathcal{N}\}$ and $\bar{\mathbf{P}} = \{\bar{P}_{ij} : \text{for all } (i,j) \in \mathcal{N}\}$ as illustrated in Figure 4. Suppose vehicle $j$ has the first operation $(i',j)$. When vehicle $j$ is outside the intersection ($x_j(0) < \alpha_{j,min}$), $\bar{T}_{ij}$ and $\bar{P}_{ij}$ represent the minimum times at which vehicle $j$ can enter and exit conflict area $i$, respectively, *no matter what speed* it has at $T_{i'j}^{\mathcal{F}}$. When vehicle $j$ is inside the intersection ($\alpha_{j,min} \le x_j(0)$), $\bar{T}_{ij}$ and $\bar{P}_{ij}$ are the minimum times at which it enters and exits conflict area $i$, respectively. These are formally defined as follows.

**Definition 3.** Given an initial condition $(\mathbf{x}(0), \dot{\mathbf{x}}(0))$ and a schedule $\mathbf{T}^{\mathcal{F}}$, we define $\bar{\mathbf{T}} = \{\bar{T}_{ij} : \text{for all } (i,j) \in \mathcal{N}\}$ and $\bar{\mathbf{P}} = \{\bar{P}_{ij} : \text{for all } (i,j) \in \mathcal{N}\}$ as follows.

If $x_j(0) < \alpha_{j,min}$, for $(i,j) \in \mathcal{F}$,

$$\begin{aligned} \bar{T}_{ij} &= T_{ij}^{\mathcal{F}}, \\ \bar{P}_{ij} &= T_{ij}^{\mathcal{F}} + \min_{u_j(\cdot)\in\mathcal{U}_j} \{t : x_j(t, u_j(\cdot), \alpha_{ij}, \dot{x}_{j,min}) = \beta_{ij}\}, \end{aligned} \tag{11}$$

and for $(i,j) \in \mathcal{N} \setminus \mathcal{F}$, there exists the first operation $(i',j)$ such that $(i',j) \in \mathcal{F}$ and $\alpha_{i'j} = \alpha_{j,min}$.

$$\begin{aligned} \bar{T}_{ij} &= T_{i'j}^{\mathcal{F}} + \min_{u_j(\cdot)\in\mathcal{U}_j} \{t : x_j(t, u_j(\cdot), \alpha_{j,min}, \dot{x}_{j,max}) = \alpha_{ij}\}, \\ \bar{P}_{ij} &= T_{i'j}^{\mathcal{F}} + \min_{u_j(\cdot)\in\mathcal{U}_j} \{t : x_j(t, u_j(\cdot), \alpha_{j,min}, \dot{x}_{j,min}) = \beta_{ij}\}. \end{aligned} \tag{12}$$

If $\alpha_{j,min} \le x_j(0)$, for $(i,j) \in \mathcal{F}$,

$$\begin{aligned} \bar{T}_{ij} &= T_{ij}^{\mathcal{F}}, \\ \bar{P}_{ij} &= T_{ij}^{\mathcal{F}} - \bar{R}_{ij} + \\ &\qquad \min_{u_j(\cdot)\in\mathcal{U}_j} \{t : x_j(t, u_j(\cdot), x_j(0), \dot{x}_j(0)) = \beta_{ij}\}. \end{aligned} \tag{13}$$

and for $(i,j) \in \mathcal{N} \setminus \mathcal{F}$, there exists the first operation $(i',j)$ such that $(i',j) \in \mathcal{F}$.

$$\begin{aligned} \bar{T}_{ij} &= T_{i'j}^{\mathcal{F}} - \bar{R}_{i'j} + \\ &\qquad \min_{u_j(\cdot)\in\mathcal{U}_j} \{t : x_j(t, u_j(\cdot), x_j(0), \dot{x}_j(0)) = \alpha_{ij}\}, \\ \bar{P}_{ij} &= T_{i'j}^{\mathcal{F}} - \bar{R}_{i'j} + \\ &\qquad \min_{u_j(\cdot)\in\mathcal{U}_j} \{t : x_j(t, u_j(\cdot), x_j(0), \dot{x}_j(0)) = \beta_{ij}\}. \end{aligned} \tag{14}$$

In (13) and (14), we have $\bar{R}_{ij} = \bar{D}_{ij}$ by (10) and thus for $(i',j) \in \mathcal{F}$, $T_{i'j}^{\mathcal{F}} - \bar{R}_{i'j} = 0$ if $T_{i'j}^{\mathcal{F}} \in [\bar{R}_{i'j}, \bar{D}_{i'j}]$.

Using these definitions, we formulate the upper bound problem.

**Problem 4** (upper bound problem)**.** Given an initial condition $(\mathbf{x}(0), \dot{\mathbf{x}}(0))$, determine if $s_U^* = 0$:

$$s_U^* := \underset{\mathbf{T}^{\mathcal{F}}, \mathbf{k}}{\text{minimize}} \max_{(i,j)\in\mathcal{F}} (T_{ij}^{\mathcal{F}} - \bar{D}_{ij}, 0)$$

subject to

$$\text{for all } (i,j) \in \mathcal{F}, \qquad \bar{R}_{ij} \le T_{ij}^{\mathcal{F}}, \tag{P4.1}$$

$$\text{for all } (i,j) \leftrightarrow (i,j') \in \mathcal{D}, \quad \begin{cases} \bar{P}_{ij} \le \bar{T}_{ij'} + M(1 - k_{ijj'}), \\ \bar{P}_{ij'} \le \bar{T}_{ij} + M(1 - k_{ij'j}), \\ k_{ijj'} + k_{ij'j} = 1. \end{cases} \tag{P4.2}$$

where $\mathbf{T}^{\mathcal{F}} = \{T_{ij}^{\mathcal{F}} : \text{for all } (i,j) \in \mathcal{F}\}$, $\mathbf{k} = \{k_{ijj'} \in \{0,1\} : \text{for all } (i,j) \leftrightarrow (i,j') \in \mathcal{D}\}$, and $M$ is a large number in $\mathbb{R}_+$.

The constraints in the problem are written in linear forms with the decision variable $\mathbf{T}^{\mathcal{F}}$ as noticed in (11)-(14). Thus, the problem is a MILP problem.

We will show that $s_U^*$ in Problem 4 can be considered as an upper bound of $s^*$ in Problem 2 in the sense that $s^* \le M s_U^*$ for a large number $M > 0$. This inequality is not trivial if $s_U^* = 0$. In the following thereom, therefore, we will show that $s_U^* = 0$ implies $s^* = 0$ for any initial condition $(\mathbf{x}(0), \dot{\mathbf{x}}(0))$.

**Theorem 3.** $s_U^* = 0 \Rightarrow s^* = 0$.

*Proof:* Suppose Problem 4 finds an optimal solution $\mathbf{T}^{\mathcal{F}*} = \{T_{ij}^{\mathcal{F}*} : (i,j) \in \mathcal{F}\}$ and $\mathbf{k}^* = \{k^*_{ijj'} : (i,j) \leftrightarrow (i,j') \in \mathcal{D}\}$ that yields $s^*_U = 0$.

We define $\tilde{\mathbf{T}}$ and $\tilde{\mathbf{P}}$ as follows. For $(i,j) \in \mathcal{F}$, if $x_j(0) < \alpha_{ij}$,

$$\begin{aligned} \tilde{T}_{ij} =& T_{ij}^{\mathcal{F}*}, \\ \tilde{P}_{ij} =& T_{ij}^{\mathcal{F}*} + \min_{u_j(\cdot)\in\mathcal{U}_j}\{t : x_j(t, u_j(\cdot), \alpha_{ij}, \dot{x}_j^0) = \beta_{ij}\}, \end{aligned} \tag{15}$$

where $\dot{x}_j^0 = \dot{x}_j(T_{ij}^{\mathcal{F}*}, u_j(\cdot), x_j(0), \dot{x}_j(0))$ with $\alpha_{ij} = x_j(T_{ij}^{\mathcal{F}*}, u_j(\cdot), x_j(0), \dot{x}_j(0))$. If $x_j(0) \geq \alpha_{ij}$, let $\tilde{T}_{ij} = T_{ij}^{\mathcal{F}*} = 0$ and $\tilde{P}_{ij} = \min\{t : x_j(t, u_j(\cdot), x_j(0), \dot{x}_j(0)) = \beta_{ij}\}$.

For $(i,j) \in \mathcal{N} \setminus \mathcal{F}$, there exists the first operation $(i',j) \in \mathcal{F}$, and

$$\begin{aligned} \tilde{T}_{ij} =& T_{i'j}^{\mathcal{F}*} + \min_{u_j(\cdot)\in\mathcal{U}_j}\{t : x_j(t, u_j(\cdot), \alpha_{i'j}, \dot{x}_j^0) = \alpha_{ij}\}, \\ \tilde{P}_{ij} =& T_{i'j}^{\mathcal{F}*} + \min_{u_j(\cdot)\in\mathcal{U}_j}\{t : x_j(t, u_j(\cdot), \alpha_{i'j}, \dot{x}_j^0) = \beta_{ij}\}, \end{aligned} \tag{16}$$

where $\dot{x}_j^0 = \dot{x}_j(T_{i'j}^{\mathcal{F}*}, u_j(\cdot), x_j(0), \dot{x}_j(0))$ with $\alpha_{i'j} = x_j(T_{i'j}^{\mathcal{F}*}, u_j(\cdot), x_j(0), \dot{x}_j(0))$.

Notice that $\tilde{P}_{ij}$ is $P_{ij}(\tilde{\mathbf{T}})$ by (6) and (7). We will show that $\tilde{\mathbf{T}}$ and $\tilde{\mathbf{k}} = \mathbf{k}^*$ is a feasible solution to Problem 2.

First, we will show that $\tilde{\mathbf{T}}$ satisfies constraint (P2.1) in Problem 2. For all $(i,j) \in \mathcal{F}$, $\bar{R}_{ij} = R_{ij}$ and $\bar{D}_{ij} \leq D_{ij}$. Since $T_{ij}^{\mathcal{F}*}$ satisfies constraint (P4.1), we have $R_{ij} \leq \tilde{T}_{ij}$. For $(i,j) \in \mathcal{N} \setminus \mathcal{F}$, we define $\tilde{T}_{ij}$ as the minimum time to reach conflict area $i$, thereby $\tilde{T}_{ij} = R_{ij}(\tilde{\mathbf{T}})$ by (4). This establishes that $R_{ij} \leq \tilde{T}_{ij}$ for all $(i,j) \in \mathcal{N}$.

For constraint (P2.2) in Problem 2, let us focus on (15), (16), and Definition 3 given $\mathbf{T}^{\mathcal{F}*}$. If $x_j(0) < \alpha_{j,min}$, we have $\bar{T}_{ij} \leq \tilde{T}_{ij}$ and $\bar{P}_{ij} \leq \tilde{P}_{ij}$ because $\bar{T}_{ij}$ and $\bar{P}_{ij}$ are computed with the maximum and minimum speed, respectively. If $x_j(0) \geq \alpha_{j,min}$, for $(i',j) \in \mathcal{F}$ we have $T_{i'j}^{\mathcal{F}*} = \bar{R}_{i'j}$ since $s^*_U = 0$ and $\bar{R}_{i'j} = \bar{D}_{i'j}$. This implies that $\bar{T}_{ij} = \tilde{T}_{ij}$ and $\bar{P}_{ij} = \tilde{T}_{ij}$. Thus, constraint (P4.2) becomes

$$\begin{aligned} &\tilde{P}_{ij} \leq \bar{P}_{ij} \leq \bar{T}_{ij'} + M(1 - k^*_{ijj'}) \leq \tilde{T}_{ij'} + M(1 - k^*_{ijj'}), \\ &\tilde{P}_{ij'} \leq \bar{P}_{ij'} \leq \bar{T}_{ij} + M(1 - k^*_{ij'j}) \leq \tilde{T}_{ij} + M(1 - k^*_{ij'j}). \end{aligned}$$

That is, $\tilde{\mathbf{T}}$ and $\tilde{\mathbf{k}} = \mathbf{k}^*$ satisfy constraint (P2.2) in Problem 2.

Now we have a feasible solution $\tilde{\mathbf{T}}$ and $\tilde{\mathbf{k}}$. For $(i,j) \in \mathcal{N} \setminus \mathcal{F}$, we have $\tilde{T}_{ij} = R_{ij} \leq D_{ij}$, and thus, $\max_{(i,j)\in\mathcal{N}\setminus\mathcal{F}}(\tilde{T}_{ij} - D_{ij}, 0) = 0$. For $(i,j) \in \mathcal{F}$, we have $\bar{D}_{ij} \leq D_{ij}$ and the following inequalities complete the proof.

$$\begin{aligned} s^* &\leq \max_{(i,j)\in\mathcal{N}}(\tilde{T}_{ij} - D_{ij}, 0) = \max_{(i,j)\in\mathcal{F}}(\tilde{T}_{ij} - D_{ij}, 0) \\ &\leq \max_{(i,j)\in\mathcal{F}}(\tilde{T}_{ij} - \bar{D}_{ij}, 0) = s^*_U. \end{aligned}$$

Therefore, if $s^*_U = 0$, we have $s^* = 0$. ∎

By Theorems 2 and 3, we have

$$s^*_L > 0 \Rightarrow s^* > 0 \quad \text{and} \quad s^*_U = 0 \Rightarrow s^* = 0.$$

That is, from Problems 3 and 4, which can be solved with a commercial solver such as CPLEX, we can find the solution for Problem 2 as shown in Table I. However, when $s^*_L = 0$ and $s^*_U > 0$, represented by Case II in the table, $s^*$ is not exactly determined. In this case, we will provide approximation bounds.

TABLE I
SUMMARY OF THEOREMS 2 AND 3

| | $s^*_L$ | $s^*_U$ | $s^*$ |
|---|---|---|---|
| Case I | · | 0 | **0** |
| Case II | 0 | + | **?** |
| Case III | + | · | **+** |

### C. Approximation bounds

We exactly solve Problem 2 in cases I and III as noted in Table I, but have to approximate the solution in case II. In this section, we will focus on the latter case when $s^*_L = 0$ and $s^*_U > 0$ and quantify the approximation bounds. Notice that $s^*_L > 0$ and $s^*_U = 0$ cannot occur.

We will prove the following statement: if $s^*_L = 0$ and $s^*_U > 0$, an input exists that makes the system avoid a *shrunk bad set* but not an *inflated bad set*. These bad sets are defined independent of time and thus we consider $\mathcal{N} = \bar{\mathcal{N}}$.

A shrunk conflict area and an inflated conflict area are defined in the following definitions. See Figure 5.

**Definition 4.** A shrunk conflict area $(\hat{\alpha}_{ij}, \hat{\beta}_{ij})$ for all $(i,j) \in \mathcal{N}$ is defined as follows. For all $(i,j) \in \mathcal{F}$,

$$\begin{aligned} &\hat{\alpha}_{ij} = \alpha_{ij}, \\ &\hat{\beta}_{ij} = \min_{u_j(\cdot)\in\mathcal{U}_j} x_j\left(\frac{\beta_{ij} - \alpha_{ij}}{\dot{x}_{j,max}}, u_j(\cdot), \alpha_{ij}, \dot{x}_{j,min}\right). \end{aligned} \tag{17}$$

For all $(i,j) \in \mathcal{N} \setminus \mathcal{F}$, there exists the first operation $(i',j)$ for $i \neq i'$ such that $(i',j) \in \mathcal{F}$.

$$\begin{aligned} \hat{\alpha}_{ij} &= \max_{u_j(\cdot)\in\mathcal{U}_j} x_j\left(\frac{\alpha_{ij} - \alpha_{i'j}}{\dot{x}_{j,min}}, u_j(\cdot), \alpha_{i'j}, \dot{x}_{j,max}\right), \\ \hat{\beta}_{ij} &= \min_{u_j(\cdot)\in\mathcal{U}_j} x_j\left(\frac{\beta_{ij} - \alpha_{i'j}}{\dot{x}_{j,max}}, u_j(\cdot), \alpha_{i'j}, \dot{x}_{j,min}\right). \end{aligned} \tag{18}$$

If $\hat{\alpha}_{ij} \geq \hat{\beta}_{ij}$, set $(\hat{\alpha}_{ij}, \hat{\beta}_{ij})$ as an empty set.

**Definition 5.** An inflated conflict area $(\check{\alpha}_{ij}, \check{\beta}_{ij})$ for all $(i,j) \in \mathcal{N}$ is defined as follows. For all $(i,j) \in \mathcal{F}$,

$$\check{\alpha}_{ij} = \alpha_{ij}, \quad \check{\beta}_{ij} = \max_{u_j(\cdot)\in\mathcal{U}_j} x_j(t^*, u_j(\cdot), \alpha_{ij}, \dot{x}_{j,max}), \tag{19}$$

where $t^* = \min_{u_j(\cdot)\in\mathcal{U}_j}\{t : x_j(t, u_j(\cdot), \alpha_{ij}, \dot{x}_{j,min}) = \beta_{ij}\}$ and $\alpha_{ij} = \alpha_{j,min}$.

For all $(i,j) \in \mathcal{N} \setminus \mathcal{F}$,

$$\begin{aligned} \check{\alpha}_{ij} &= \min_{u_j(\cdot)\in\mathcal{U}_j} x_j(t^{**}, u_j(\cdot), \alpha_{j,min}, \dot{x}_{j,min}), \\ \check{\beta}_{ij} &= \max_{u_j(\cdot)\in\mathcal{U}_j} x_j(t^{***}, u_j(\cdot), \alpha_{j,min}, \dot{x}_{j,max}), \end{aligned} \tag{20}$$

where $t^{**} = \min_{u_j\in\mathcal{U}_j}\{t : x_j(t, u_j(\cdot), \alpha_{j,min}, \dot{x}_{j,max}) = \alpha_{ij}\}$ and $t^{***} = \min_{u_j\in\mathcal{U}_j}\{t : x_j(t, u_j(\cdot), \alpha_{j,min}, \dot{x}_{j,min}) = \beta_{ij}\}$. Notice that $t^*, t^{**}$, and $t^{***}$ are the same as the added times in (11) and (12).

A shrunk bad set $\hat{\mathcal{B}}$ and an inflated bad set $\check{\mathcal{B}}$ are defined as follows:

$$\begin{aligned} \hat{\mathcal{B}} := \{\mathbf{x} \in \mathbf{X} : &\text{for some } (i,j) \leftrightarrow (i,j') \in \mathcal{D} \\ &x_j \in (\hat{\alpha}_{ij}, \hat{\beta}_{ij}) \text{ and } x_{j'} \in (\hat{\alpha}_{ij'}, \hat{\beta}_{ij'})\}. \end{aligned} \tag{21}$$

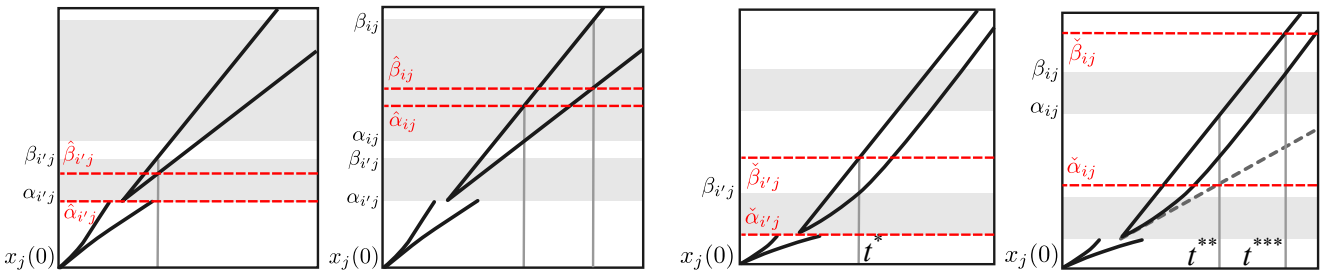



Fig. 5. Shrunk and Inflated conflict areas for $(i',j) \in \mathcal{F}$ and $(i',j) \to (i,j) \in \mathcal{C}$. Figures (a)-(d) illustrate (17)-(20), respectively. By definition, $(\hat{\alpha}_{i'j}, \hat{\beta}_{i'j}) \subseteq (\alpha_{i'j}, \beta_{i'j}) \subseteq (\check{\alpha}_{i'j}, \check{\beta}_{i'j})$ and $(\hat{\alpha}_{ij}, \hat{\beta}_{ij}) \subseteq (\alpha_{ij}, \beta_{ij}) \subseteq (\check{\alpha}_{ij}, \check{\beta}_{ij})$.

$$\check{\mathcal{B}} := \{\mathbf{x} \in \mathbf{X} : \text{for some } (i,j) \leftrightarrow (i,j') \in \mathcal{D} \\ x_j \in (\check{\alpha}_{ij}, \check{\beta}_{ij}) \text{ and } x_{j'} \in (\check{\alpha}_{ij'}, \check{\beta}_{ij'})\}. \quad (22)$$

It can be checked that

$$\hat{\mathcal{B}} \subseteq \mathcal{B} \subseteq \check{\mathcal{B}}$$

by showing that $(\hat{\alpha}_{ij}, \hat{\beta}_{ij}) \subseteq (\alpha_{ij}, \beta_{ij}) \subseteq (\check{\alpha}_{ij}, \check{\beta}_{ij})$ for all $(i,j) \in \mathcal{N}$.

In the following theorems, we prove that the shrunk and inflated bad sets can represent the approximation errors of the solutions of Problems 3 and 4, respectively. More precisely, we prove that 1) if the solution of Problem 3 is *yes*, that is, $s_L^* = 0$, then there exists an input that makes the system avoid the shrunk bad set, and 2) if the solution of Problem 4 is *no*, that is, $s_U^* > 0$, then there is no input that makes the system avoid the inflated bad set.

**Theorem 4.** *Given an initial condition* $(\mathbf{x}(0), \dot{\mathbf{x}}(0))$*, if* $s_L* = 0$*, then there exists an input signal* $\mathbf{u}(\cdot) \in \mathcal{U}$ *such that* $\mathbf{x}(t, \mathbf{u}(\cdot), \mathbf{x}(0), \dot{\mathbf{x}}(0)) \notin \hat{\mathcal{B}}$ *for all* $t \geq 0$.

*Proof:* If $s_L^* = 0$, there exists an optimal solution $\mathbf{t}^*, \mathbf{p}^*$ and $\mathbf{k}^*$ that satisfies $t_{ij}^* \leq d_{ij}$ for all $(i,j) \in \mathcal{N}$ and constraints (P3.1)-(P3.3) in Problem 3.

For $(i,j) \in \mathcal{F}$, $r_{ij} \leq t_{ij}^* \leq d_{ij}$ implies that there exists $u_j^*(\cdot) : [0, t_{ij}^*) \to U_j$ such that

$$x_j(t_{ij}^*, u_j^*(\cdot), x_j(0), \dot{x}_j(0)) = \alpha_{ij} = \hat{\alpha}_{ij} \quad (23)$$

since the flow of $x_j$ is a continuous function of $u_j^*(\cdot)$ and the input space is path connected by Assumption 1. Let $\dot{x}_j(t_{ij}^*)$ denote $\dot{x}_j(t_{ij}^*, u_j^*(\cdot), x_j(0), \dot{x}_j(0))$. Constraint (P3.2), $(\beta_{ij} - \alpha_{ij})/\dot{x}_{j,max} \leq p_{ij}^* - t_{ij}^* \leq (\beta_{ij} - \alpha_{ij})/\dot{x}_{j,min}$, implies that for any input $u_j^*(\cdot) : [t_{ij}^*, p_{ij}^*) \to U_j$,

$$x_j(p_{ij}^* - t_{ij}^*, u_j^*(\cdot), \alpha_{ij}, \dot{x}_j(t_{ij}^*)) \geq \hat{\beta}_{ij}, \quad (24)$$

because $\hat{\beta}_{ij}$ defined in (17) is the minimum distance from $\alpha_{ij}$ traveled for time $(\beta_{ij} - \alpha_{ij})/\dot{x}_{j,max}$.

For $(i,j) \in \mathcal{N} \setminus \mathcal{F}$, there exists the first operation $(i',j) \in \mathcal{F}$. By the definition of $r_{ij}$ and $d_{ij}$ in (8), we have

$$t_{i'j}^* + \frac{\alpha_{ij} - \alpha_{i'j}}{\dot{x}_{j,max}} \leq r_{ij} \leq t_{ij}^* \leq d_{ij} \leq t_{i'j}^* + \frac{\alpha_{ij} - \alpha_{i'j}}{\dot{x}_{j,min}}.$$

Since $\hat{\alpha}_{ij}$ is the maximum distance from $\alpha_{i'j}$ traveled for time $(\alpha_{ij} - \alpha_{i'j})/\dot{x}_{j,min}$, for any input $u_j^*(\cdot) : [t_{i'j}^*, t_{ij}^*) \to U_j$,

$$x_j(t_{ij}^* - t_{i'j}^*, u_j^*(\cdot), \alpha_{i'j}, \dot{x}_j(t_{i'j}^*)) \leq \hat{\alpha}_{ij}. \quad (25)$$

Similarly for any input $u_j^*(\cdot) : [t_{i'j}^*, p_{ij}^*) \to U_j$,

$$x_j(p_{ij}^* - t_{i'j}^*, u_j^*(\cdot), \alpha_{i'j}, \dot{x}_j(t_{i'j}^*)) \geq \hat{\beta}_{ij}. \quad (26)$$

Thus, there exists an input $u_j^*(\cdot) \in \mathcal{U}_j$ that satisfies (23)-(26).

Now we will show that $\mathbf{x}(t, \mathbf{u}^*(\cdot), \mathbf{x}(0), \dot{\mathbf{x}}(0)) \notin \hat{\mathcal{B}}$ for all $t \geq 0$. By (P3.3), we have for $(i,j) \leftrightarrow (i,j') \in \mathcal{D}$ either $p_{ij}^* \leq t_{ij'}^*$ or $p_{ij'}^* \leq t_{ij}^*$. Without loss of generality, we consider $p_{ij}^* \leq t_{ij'}^*$. Then, at time $t_{ij'}^*$, vehicle $j'$ has not yet entered shrunk conflict area $i$ as shown in (23) and (25). At the same time, vehicle $j$ has already exited the shrunk conflict area as shown in (24) and (26). Thus, two vehicles never meet inside the same shrunk conflict area, and thus the system avoids the shrunk bad set with the input signal $\mathbf{u}^*(\cdot)$. ■

**Theorem 5.** *Given an initial condition* $(\mathbf{x}(0), \dot{\mathbf{x}}(0))$*, if* $s_U^* > 0$*, then for all input signal* $\mathbf{u}(\cdot) \in \mathcal{U}$*,* $\mathbf{x}(t, \mathbf{u}(\cdot), \mathbf{x}(0), \dot{\mathbf{x}}(0)) \in \check{\mathcal{B}}$ *for some* $t \geq 0$.

*Proof:* We prove the contra-position: if there exists an input signal $\mathbf{u}^*(\cdot) \in \mathcal{U}$ such that $\mathbf{x}(t, \mathbf{u}^*(\cdot), \mathbf{x}(0), \dot{\mathbf{x}}(0)) \notin \check{\mathcal{B}}$ for all $t \geq 0$, then $s_U^* = 0$.

For all $(i,j) \in \mathcal{N}$, let us define

$$T_{ij}^* := \{t : x_j(t, u_j^*(\cdot), x_j(0), \dot{x}_j(0)) = \check{\alpha}_{ij}\},$$

if $x_j(0) < \check{\alpha}_{ij}$. Otherwise, set $T_{ij}^* = 0$. Also, if $x_j(0) < \check{\beta}_{ij}$,

$$P_{ij}^* := \{t : x_j(t, u_j^*(\cdot), x_j(0), \dot{x}_j(0)) = \check{\beta}_{ij}\}.$$

Otherwise, set $P_{ij}^* = 0$.

Since $\mathbf{x}(t, \mathbf{u}^*(\cdot), \mathbf{x}(0), \dot{\mathbf{x}}(0)) \notin \check{\mathcal{B}}$ for all $t \geq 0$, we have for all $(i,j) \leftrightarrow (i,j') \in \mathcal{D}$,

$$(T_{ij}^*, P_{ij}^*) \cap (T_{ij'}^*, P_{ij'}^*) = \emptyset, \quad (27)$$

which indicates that each inflated conflict area is occupied by only one vehicle at a time.

Let us define a schedule $\tilde{\mathbf{T}}^{\mathcal{F}} = \{\tilde{T}_{ij}^{\mathcal{F}} : \text{for all } (i,j) \in \mathcal{F}\}$ as follows: if $x_j(0) < \alpha_{j,min}$,

$$\tilde{T}_{ij}^{\mathcal{F}} := \{t : x_j(t, u_j^*(\cdot), x_j(0), \dot{x}_j(0)) = \alpha_{j,min}\}.$$

If $\alpha_{j,min} \leq x_j(0)$, set $\tilde{T}_{ij}^{\mathcal{F}} = \bar{R}_{ij}$. By Definition 2, $\tilde{T}_{ij}^{\mathcal{F}} \in [\bar{R}_{ij}, \bar{D}_{ij}]$. Thus, $\tilde{\mathbf{T}}^{\mathcal{F}}$ satisfies constraint (P4.1) and $\tilde{T}_{ij}^{\mathcal{F}} - \bar{D}_{ij} \leq 0$ for all $(i,j) \in \mathcal{F}$. If $\tilde{\mathbf{T}}^{\mathcal{F}}$ satisfies constraint (P4.2), $\tilde{\mathbf{T}}^{\mathcal{F}}$ is a feasible solution to Problem 4 with a corresponding binary variable $\tilde{\mathbf{k}} = \{\tilde{k}_{ijj'} : \text{for all } (i,j) \leftrightarrow (i,j') \in \mathcal{D}\}$ and thus $s_U^* = 0$.

Given $\tilde{\mathbf{T}}^{\mathcal{F}}$, $\bar{\mathbf{T}} = \{\bar{T}_{ij} : \text{for all } (i,j) \in \mathcal{N}\}$ and $\bar{\mathbf{P}} = \{\bar{P}_{ij} : \text{for all } (i,j) \in \mathcal{N}\}$ are defined according to Definition 3.

First, consider $x_j(0) < \alpha_{j,min}$. For $(i,j) \in \mathcal{F}$, we have by (11), $\bar{T}_{ij} = \tilde{T}^{\mathcal{F}}_{ij}$ and $\bar{P}_{ij} = \tilde{T}^{\mathcal{F}}_{ij} + t^*$ where $t^*$ is introduced in (19). Since $\check{\alpha}_{ij} \leq \alpha_{j,min}$ by definition, $T^*_{ij} \leq \bar{T}_{ij}$. Also, since $\check{\beta}_{ij}$ represents the maximum distance from $\alpha_{j,min}$ that vehicle $j$ can travel during $t^*$, traveling from $\alpha_{j,min}$ to $\check{\beta}_{ij}$ takes no less time than $t^*$. Thus, $\bar{P}_{ij} \leq P^*_{ij}$. For $(i,j) \in \mathcal{N} \setminus \mathcal{F}$, there exists $(i',j) \in \mathcal{F}$. By (12), $\bar{T}_{ij} = \tilde{T}^{\mathcal{F}}_{i'j} + t^{**}$ and $\bar{P}_{ij} = \tilde{T}^{\mathcal{F}}_{i'j} + t^{***}$, where $t^{**}$ and $t^{***}$ are introduced in (20). Since $\check{\alpha}_{ij}$ and $\check{\beta}_{ij}$ are the minimum and maximum distance traveled from $\alpha_{j,min}$ for time $t^{**}$ and $t^{***}$, respectively, we have $T^*_{ij} \leq \bar{T}_{ij}$ and $\bar{P}_{ij} \leq P^*_{ij}$.

Next, consider $\alpha_{j,min} \leq x_j(0)$. In this case, $\bar{R}_{ij} = \bar{D}_{ij}$ for all vehicle $j$'s operations by (10) and thus for $(i',j) \in \mathcal{F}$, we have $\tilde{T}^{\mathcal{F}}_{i'j} = \bar{R}_{i'j}$. By (13), $\bar{T}_{i'j} = \tilde{T}^{\mathcal{F}}_{i'j}$ and $\bar{P}_{i'j}$ is the minimum time to reach $\beta_{ij}$ since $\tilde{T}^{\mathcal{F}}_{i'j} - \bar{R}_{i'j} = 0$. The fact that $\check{\alpha}_{ij} \leq \alpha_{ij}$ and $\beta_{ij} \leq \check{\beta}_{ij}$ implies $T^*_{ij} \leq \bar{T}_{ij}$ and $\bar{P}_{ij} \leq P^*_{ij}$, respectively. For $(i,j) \in \mathcal{N} \setminus \mathcal{F}$, $\bar{T}_{ij}$ and $\bar{P}_{ij}$ are the minimum time to reach $\alpha_{ij}$ and $\beta_{ij}$, respectively, starting from $x_j(0)$. Since $\bar{T}_{ij} \geq (\alpha_{ij} - x_j(0))/\dot{x}_{j,max}$ and $\check{\alpha}_{ij}$ is the minimum distance traveled in $(\alpha_{ij} - x_j(0))/\dot{x}_{j,max}$, $T^*_{ij} \leq \bar{T}_{ij}$. Also, $\beta_{ij} \leq \check{\beta}_{ij}$ implies $\bar{P}_{ij} \leq P^*_{ij}$.

Therefore, $(\bar{T}_{ij}, \bar{P}_{ij}) \subseteq (T^*_{ij}, P^*_{ij})$ for all $(i,j) \in \mathcal{N}$. By (27), we derive $(\bar{T}_{ij}, \bar{P}_{ij}) \cap (\bar{T}_{ij'}, \bar{P}_{ij'}) = \emptyset$ for all $(i,j) \leftrightarrow (i,j') \in \mathcal{D}$. This is equivalent to constraint (P4.2), and thus there exists $\tilde{k}_{ijj'}$ and $\tilde{k}_{ij'j}$ satisfying the constraint.

Therefore, $\tilde{\mathbf{T}}^{\mathcal{F}}$ and $\tilde{\mathbf{k}}$ is a feasible solution with $s^*_U = 0$. ■

By Theorem 4, if Problem 3 returns *yes*, there is an input to make the system avoid the shrunk bad set. Otherwise, no input exists to make the system avoid the bad set. Similarly for Problem 4, if it returns *yes*, there is an input to make the system avoid the bad set. Otherwise, no input exists to avoid the inflated bad set by Theorem 5. Thus, the shrunk and inflated bad sets represent the over-approximation and under-approximation of the reachable set from an initial condition $(\mathbf{x}(0), \dot{\mathbf{x}}(0))$, respectively.

### *D. Other upper bound solutions*

In Section V-B, we formulate an MILP problem that yields an upper bound by considering the maximum input inside an intersection. Different MILP formulations are possible, for example, by considering the minimum input inside an intersection. To obtain a tighter upper bound, various combinations of the maximum and minimum inputs can be considered inside an intersection with a binary variable associated with each combination. At the expense of computational complexity, this approach is less conservative since more choices of inputs are allowed.

## VI. Control design

Based on the results of Section V, we can design a supervisor that is activated when a future collision is detected inside the inflated conflict areas.

Let ApproxVerification$(\mathbf{x}(0), \dot{\mathbf{x}}(0))$ be an algorithm solving Problem 4 given an initial condition $(\mathbf{x}(0), \dot{\mathbf{x}}(0))$. Let ApproxVerification return $(s^*_U, \mathbf{T}^{\mathcal{F}*})$ where $\mathbf{T}^{\mathcal{F}*}$ is the optimal solution.

The supervisory algorithm runs in discrete time with a time step $\tau$. At time $k\tau$, it receives the measurements of the state $(\mathbf{x}(k\tau), \dot{\mathbf{x}}(k\tau))$ and the desired input $\mathbf{u}^k_d \in \mathbf{U}$, which is a vector of inputs that the drivers are applying at the time. These measurements are used to predict the desired state at the next time step, which is denoted by $(\mathbf{x}^k_d, \dot{\mathbf{x}}^k_d)$. We solve Problem 4 to see if the desired state has a safe input signal within the approximation bound.

If ApproxVerification$(\mathbf{x}^k_d, \dot{\mathbf{x}}^k_d)$ returns $\mathbf{T}^{\mathcal{F}*}$ that makes $s^*_U = 0$, we can find a safe input signal by defining an input generator function $\sigma : \mathbf{X} \times \dot{\mathbf{X}} \times \mathbb{R}^n \to \mathcal{U}$ as follows:

$$\sigma(\mathbf{x}^k_d, \dot{\mathbf{x}}^k_d, \mathbf{T}^{\mathcal{F}*}) \in \{\mathbf{u}(\cdot) \in \mathcal{U} : \text{ for all } (i,j) \in \mathcal{F},$$
$$x_j(T^{\mathcal{F}*}_{ij}, u_j(\cdot), x^k_{j,d}, \dot{x}^k_{j,d}) = \alpha_{ij} \text{ and } u_j(t) = u_{j,max}\, \forall\, t \geq T^{\mathcal{F}*}_{ij}\},$$

where $x^k_{j,d}$ and $\dot{x}^k_{j,d}$ denote the $j$-th entries of $\mathbf{x}^k_d$ and $\dot{\mathbf{x}}^k_d$, respectively. The supervisor stores this safe input restricted to time $(0, \tau)$ for a possible use at the next time step. Since there is an input signal that makes the system avoid entering the bad set from $(\mathbf{x}^k_d, \dot{\mathbf{x}}^k_d)$, the supervisor allows the desired input.

If ApproxVerification$(\mathbf{x}^k_d, \dot{\mathbf{x}}^k_d)$ returns $s^*_U > 0$, the supervisor overrides the drivers with the safe input stored at the previous step. This safe input is used to predict the safe state, denoted by $(\mathbf{x}^k_{safe}, \dot{\mathbf{x}}^k_{safe})$, and this safe state is used to generate a safe input for the next time step. We will prove in the next theorem that ApproxVerification$(\mathbf{x}^k_{safe}, \dot{\mathbf{x}}^k_{safe})$ always returns $s^*_U = 0$ and thus a safe input signal is defined.

This supervisor is provided in Algorithm 1. An input signal with superscript $k$, such as $\mathbf{u}^k(\cdot)$, denotes an input function from $[0, \tau)$ to $\mathbf{U}$. We define the desired input signal as $\mathbf{u}^k_d(\cdot) : t \mapsto \mathbf{u}^k_d$ given $\mathbf{u}^k_d \in \mathbf{U}$. Also, an input signal with superscript $k, \infty$ indicates that the domain of the input signal is $[0, \infty)$.

**Algorithm 1** Supervisory control algorithm at $t = k\tau$

**procedure** Supervisor$(\mathbf{x}(k\tau), \dot{\mathbf{x}}(k\tau), \mathbf{u}^k_d)$
  $\mathbf{x}^k_d \leftarrow \mathbf{x}(\tau, \mathbf{u}^k_d(\cdot), \mathbf{x}(k\tau), \dot{\mathbf{x}}(k\tau))$
  $\dot{\mathbf{x}}^k_d \leftarrow \dot{\mathbf{x}}(\tau, \mathbf{u}^k_d(\cdot), \mathbf{x}(k\tau), \dot{\mathbf{x}}(k\tau))$
  $(s^*_U, \mathbf{T}^{\mathcal{F}*}_1) =$ ApproxVerification$(\mathbf{x}^k_d, \dot{\mathbf{x}}^k_d)$
  **if** $s^*_U = 0$ **then**
    $\mathbf{u}^{k+1,\infty}_{safe}(\cdot) \leftarrow \sigma(\mathbf{x}^k_d, \dot{\mathbf{x}}^k_d, \mathbf{T}^{\mathcal{F}*}_1)$
    $\mathbf{u}^{k+1}_{safe}(\cdot) \leftarrow \mathbf{u}^{k+1,\infty}_{safe}(t)$ for $t \in [0, \tau)$
    **return** $\mathbf{u}^k_d(\cdot)$
  **else**
    $\mathbf{x}^k_{safe} \leftarrow \mathbf{x}(\tau, \mathbf{u}^k_{safe}(\cdot), \mathbf{x}(k\tau), \dot{\mathbf{x}}(k\tau))$
    $\dot{\mathbf{x}}^k_{safe} \leftarrow \dot{\mathbf{x}}(\tau, \mathbf{u}^k_{safe}(\cdot), \mathbf{x}(k\tau), \dot{\mathbf{x}}(k\tau))$
    $(\cdot, \mathbf{T}^{\mathcal{F}*}_2) =$ ApproxVerification$(\mathbf{x}^k_{safe}, \dot{\mathbf{x}}^k_{safe})$
    $\mathbf{u}^{k+1,\infty}_{safe}(\cdot) \leftarrow \sigma(\mathbf{x}^k_{safe}, \dot{\mathbf{x}}^k_{safe}, \mathbf{T}^{\mathcal{F}*}_2)$
    $\mathbf{u}^{k+1}_{safe}(\cdot) \leftarrow \mathbf{u}^{k+1,\infty}_{safe}(t)$ for $t \in [0, \tau)$
    **return** $\mathbf{u}^k_{safe}(\cdot)$

By Theorem 5, $s^*_U > 0$ indicates that no input signal exists to avoid the inflated bad set, that is, there may exist an input to avoid the bad set. Thus, we say this supervisor overrides vehicles when a future collision is detected within the approximation bound.

**Theorem 6.** *Algorithm 1 is non-blocking, that is, if* SUPERVISOR$(\mathbf{x}(0), \dot{\mathbf{x}}(0), \mathbf{u}_d^0) \neq \emptyset$, *then for any* $\mathbf{u}_d^k \in \mathbf{U}$, SUPERVISOR$(\mathbf{x}(k\tau), \dot{\mathbf{x}}(k\tau), \mathbf{u}_d^k) \neq \emptyset$.

*Proof:* We prove this by induction on $k$. For the base case, assume SUPERVISOR$(\mathbf{x}(0), \dot{\mathbf{x}}(0), \mathbf{u}_d^0) \neq \emptyset$ and $\mathbf{u}_{safe}^{1,\infty}(\cdot)$ is well-defined, that is, $\mathbf{u}_{safe}^{1,\infty}(\cdot) \in \mathcal{U}$. Suppose at time $(k-1)\tau$, SUPERVISOR$(\mathbf{x}((k-1)\tau), \dot{\mathbf{x}}((k-1)\tau), \mathbf{u}_d^{k-1})$ is non-empty and $\mathbf{u}_{safe}^{k,\infty}$ is well-defined. At time $k\tau$, for any $\mathbf{u}_d^k \in \mathbf{U}$, SUPERVISOR$(\mathbf{x}(k\tau), \dot{\mathbf{x}}(k\tau), \mathbf{u}_d^k)$ is not empty since it returns either $\mathbf{u}_d^k(\cdot) : t \mapsto \mathbf{u}_d^k$ or $\mathbf{u}_{safe}^k(\cdot) : t \mapsto \mathbf{u}_{safe}^{k,\infty}(t)$. We need to show that $\mathbf{u}_{safe}^{k+1,\infty}(\cdot)$ is well-defined in lines 6 or 13.

In line 6, since $\mathbf{T}_1^{\mathcal{F}*} = \{T_{ij,1}^{\mathcal{F}*} : (i,j) \in \mathcal{F}\}$ yields $s_U^* = 0$, we have $\bar{R}_{ij} \leq T_{ij,1}^{\mathcal{F}*} \leq \bar{D}_{ij}$ for all $(i,j) \in \mathcal{F}$ and thus there exists $u_j(\cdot) \in \mathcal{U}_j$ that satisfies $x_j(T_{ij,1}^{\mathcal{F}*}, u_j(\cdot), x_{j,d}^k, \dot{x}_{j,d}^k) = \alpha_{ij}$. Thus, $\sigma(\mathbf{x}_d^k, \dot{\mathbf{x}}_d^k, \mathbf{T}_1^{\mathcal{F}*})$ is well-defined.

In line 13, we will show that $\mathbf{T}_2^{\mathcal{F}*}$ satisfies $s_U^* = 0$ and thus a safe input signal is well-defined. We have that $(\mathbf{x}(k\tau), \dot{\mathbf{x}}(k\tau))$ is either $(\mathbf{x}_d^{k-1}, \dot{\mathbf{x}}_d^{k-1})$ or $(\mathbf{x}_{safe}^{k-1}, \dot{\mathbf{x}}_{safe}^{k-1})$ depending on the output of the supervisor at the previous time step. Thus, at time $(k-1)\tau$, APPROXVERIFICATION$(\mathbf{x}(k\tau), \dot{\mathbf{x}}(k\tau))$ yielded $s_U^* = 0$ with an optimal solution $\mathbf{T}^{\mathcal{F},k-1} = \{T_{ij}^{\mathcal{F},k-1} : (i,j) \in \mathcal{F}\}$. Let $\bar{R}_{ij}^{k-1}, \bar{D}_{ij}^{k-1}, \bar{T}_{ij}^{k-1}$, and $\bar{P}_{ij}^{k-1}$ be the parameters used in the problem. Now, consider APPROXVERIFICATION$(\mathbf{x}_{safe}^k, \dot{\mathbf{x}}_{safe}^k)$ with parameters $\bar{R}_{ij}^k, \bar{D}_{ij}^k, \bar{T}_{ij}^k$, and $\bar{P}_{ij}^k$. If we define $\tilde{T}_{ij}^{\mathcal{F}} = T_{ij}^{\mathcal{F},k-1} - \tau$ for all $(i,j) \in \mathcal{F}$, we have $\tilde{T}_{ij}^{\mathcal{F}} \in [\bar{R}_{ij}^k, \bar{D}_{ij}^k]$ since $T_{ij}^{\mathcal{F},k-1} \in [\bar{R}_{ij}^{k-1}, \bar{D}_{ij}^{k-1}]$ and $\mathbf{u}_{safe}^k(\cdot)$ is in $\mathcal{U}$. Also, since $(\bar{T}_{ij}^k, \bar{P}_{ij}^k) \subseteq (\bar{T}_{ij}^{k-1}, \bar{P}_{ij}^{k-1}) - \tau$ and $(\bar{T}_{ij}^{k-1}, \bar{P}_{ij}^{k-1}) \cap (\bar{T}_{ij'}^{k-1}, \bar{P}_{ij'}^{k-1}) = \emptyset$ for all $(i,j) \leftrightarrow (i,j') \in \mathcal{D}$, we have $(\bar{T}_{ij}^k, \bar{P}_{ij}^k) \cap (\bar{T}_{ij'}^k, \bar{P}_{ij'}^k) = \emptyset$. Therefore, $\tilde{\mathbf{T}}^{\mathcal{F}} = \{\tilde{T}_{ij}^{\mathcal{F}} :$ for all $(i,j) \in \mathcal{F}\}$ is a feasible solution that yields $s_U^* = 0$, and thus there exists the optimal solution $\mathbf{T}_2^{\mathcal{F}*}$ satisfying $s_U^* = 0$. Therefore, $\sigma(\mathbf{x}_{safe}, \dot{\mathbf{x}}_{safe}, \mathbf{T}_2^{\mathcal{F}*})$ is well-defined. ■

## VII. SIMULATION RESULTS

We implemented Algorithm 1 on the cases illustrated in Figures 1 and 2 to validate its collision avoidance performance and its non-blocking property. We implemented the algorithm on MATLAB and performed simulations on a personal computer consisting of an Intel Core i7 processor at 3.10 GHz and 8 GB RAM.

In the simulations, we consider the vehicle dynamics with a quadratic drag term [44] as follows: for all $j \in \{1, \ldots, n\}$

$$\ddot{x}_j = a u_j + b \dot{x}_j^2.$$

Also, the following parameters are used: $\tau = 0.1, a = 1, b = 0.005$. For all $j \in \{1, \ldots, n\}$, $u_{j,max} = 2, u_{j,min} = -2, \alpha_{j,min} = 20$. For all $(i,j) \in \mathcal{N}$, $\beta_{ij} - \alpha_{ij} = 5$ and for all $(i,j) \to (i',j) \in \mathcal{C}$, $\alpha_{i'j} - \alpha_{ij} = 6$.

Let us consider the scenario illustrated in Figure 2 with the following initial condition and parameters: $\mathbf{x}(0) = (0,0,0), \dot{\mathbf{x}}(0) = (10,8,8)$, and $\dot{x}_{j,min} = 8, \dot{x}_{j,max} = 10$ for all $j \in \{1 \ldots, n\}$. Without implementing the supervisor (Algorithm 1), we let the vehicles travel with the desired input $\mathbf{u}_d^k = (-2,-2,2)$ for all $k$ and plot the optimal values of

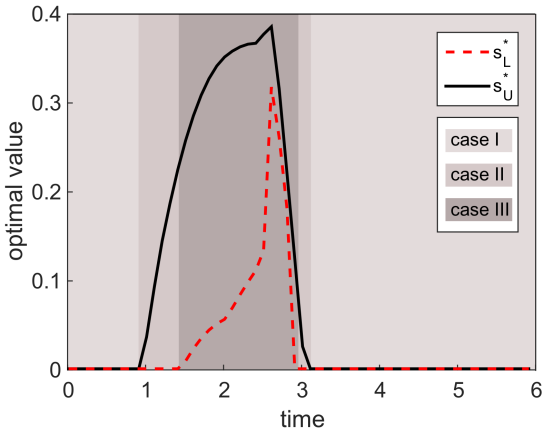


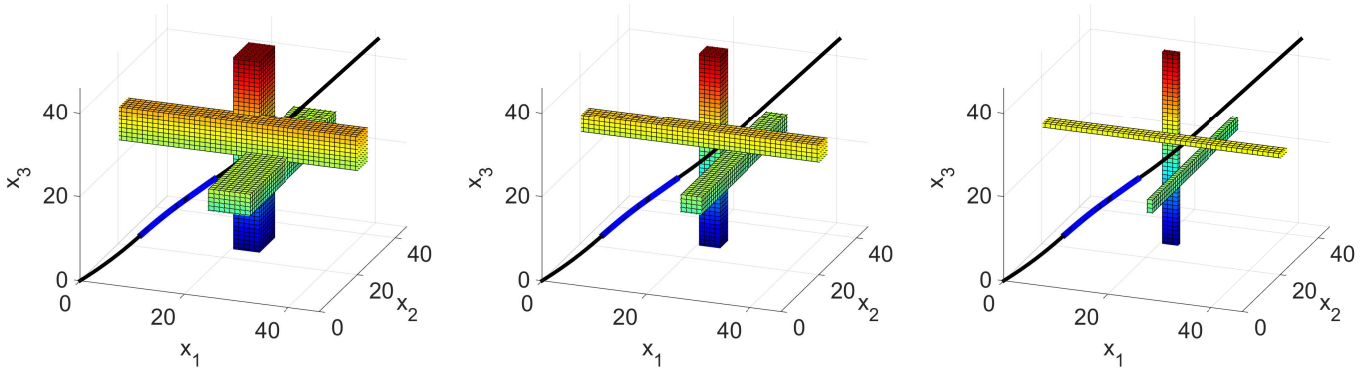

Fig. 6. Simulation results without the supervisor (Algorithm 1) for the scenario in Figure 2. Cases I, II, and III denote the same cases in Table I.

Problems 3 and 4. These are shown in Figure 6. As proved in Theorems 2 and 3, $s_U^* = 0$ implies $s_L^* = 0$. The trajectory of the system with implementing the supervisor is shown in Figure 7(a)-(c). The trajectory (black line) is controlled by the supervisor when $s_U^* > 0$ (the line is thicker in blue) so that it avoids the bad set (solid in (b)). Notice that the trajectory penetrates the inflated bad set (solid in (a)) but not the shrunk bad set (solid in (c)) as proved in Theorems 4 and 5.

Now, let us consider the scenario illustrated in Figure 1 with the following initial condition and parameters: $\mathbf{x}(0) = (0,-2,5,-5,0,5,0,1,5,4,0,-2,5,5,0,5,-2,0,-2,0)$ and for all $j \in \{1, \ldots, n\}, \dot{x}_j(0) = 5, \dot{x}_{j,min} = 1, \dot{x}_{j,max} = 10$. With the desired input $\mathbf{u}_d^k = \mathbf{u}_{max}$ for all $k$, the result is shown in Figure 8. The trajectory of vehicle 1 (black line) and the trajectories of other vehicles that share the same conflict area (red dotted lines) never stay inside the conflict area simultaneously. This implies that the supervisor overrides vehicles when necessary (when blue boxes appear) to make them cross the intersection without collisions. Because unnecessary to prove optimality, solving feasibility problems requires less computational effort than solving optimization problems [43]. That is, solving the following problem takes less computation time than solving Problem 4: given an initial condition, determine if there exists a feasible solution $(\mathbf{T}^{\mathcal{F}}, \mathbf{k})$ that satisfies (P4.1), (P4.2), and $T_{ij}^{\mathcal{F}} \leq \bar{D}_{ij}$ for all $(i,j) \in \mathcal{F}$. Notice that this problem is equivalent to Problem 4. Based on the solution of this feasibility problem, Algorithm 1 takes no more than 0.05 s per iteration, even in this realistic size scenario involving 20 vehicles, 48 conflict areas, and 120 operations on a representative geometry of dangerous intersections. Given that the allocated time step for intelligent transportation systems is 0.1 s [2], this algorithm can be implemented in real time.

## VIII. CONCLUSIONS

In this paper, we presented the design of a supervisory algorithm that determines the existence of a future collision among vehicles at an intersection (safety verification) and overrides the drivers with a safe input if a future collision is detected




(a) Inflated bad set $\check{\mathcal{B}}$ (b) Bad set $\mathcal{B}$ (c) Shrunk bad set $\hat{\mathcal{B}}$

Fig. 7. Simulation results with the supervisor for the scenario in Figure 2. The black line represents the system trajectory and is the same on each figure. The line turns to blue when the supervisor intervenes to prevent a predicted collision. The solid in each figure is (a) the inflated bad set, (b) the bad set, and (c) the shrunk bad set. The supervisor manages the system to avoid entering the bad set.

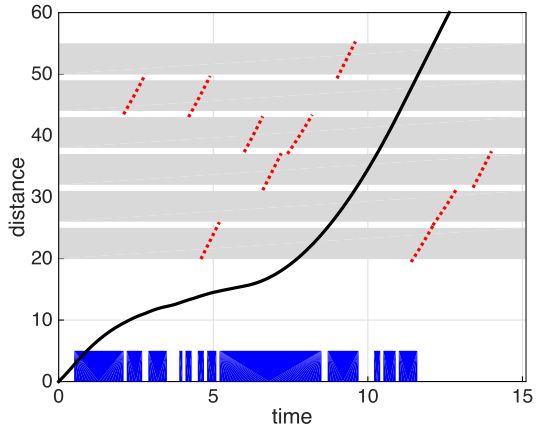



Fig. 8. Trajectory of vehicle 1 in the scenario of Figure 1 crossing six conflict areas. The blue boxes represent the times at which the supervisor overrides the vehicles. The red dotted lines are the trajectories of the other vehicles that share the same conflict area. This graph shows that each conflict area is used by only one vehicle at a time.

(control design). We translated the safety verification problem into a scheduling problem by exploiting monotonicity of the system. This scheduling problem minimizes the maximum lateness and determines if the optimal cost is zero where the zero optimal cost corresponds to the case in which all vehicles can cross the intersection without collisions. Because of the nonlinear second-order dynamics of vehicles, the scheduling problem is a Mixed Integer Nonlinear Programming (MINLP) problem, which is computationally difficult to solve. We thus approximately solved this scheduling problem by solving two Mixed Integer Linear Programming (MILP) problems that yield lower and upper bounds of the optimal cost of the scheduling problem. We quantified the approximation error between the exact and approximate solutions to the scheduling problem. We presented the design of the supervisor based on the MILP problem that computes the upper bound and proved that it is non-blocking. Computer simulations validated that the supervisor can be implemented in real time applications.

While we assumed in this paper that there is only one vehicle per lane, our approach can be easily modified to deal with the case in which multiple vehicles are present on each lane. One possible modification can be solving the scheduling problem only for the first vehicles on lanes while letting the following vehicles maintain a safe distance from their front vehicles. Instead of this naive approach, we are currently investigating a less conservative approach. Also, Problem 4 can be extended to include the presence of uncertainty sources, such as measurement noises, process errors, and not communicating vehicles, as done in our previous works [18], [19] for the single conflict area intersection model. Also, in future work, the assumption that the routes of vehicles are known in advance will be relaxed.

## REFERENCES


[1] E. Ackerman, "Fatal Tesla self-driving car crash reminds us that robots aren't perfect," Jul. 2016. [Online]. Available: http://spectrum.ieee.org/cars-that-think/transportation/self-driving/fatal-tesla-autopilot-c

[2] U.S. Department of Transportation, "ITS Strategic research plan 2015-2019," http://www.its.dot.gov/strategicplan.pdf, 2014.

[3] C. Tomlin, G. J. Pappas, and S. Sastry, "Conflict resolution for air traffic management: a study in multiagent hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 509–521, Apr. 1998.

[4] C. Tomlin, J. Lygeros, and S. Sastry, "A game theoretic approach to controller design for hybrid systems," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 949–970, Jul. 2000.

[5] J. Lygeros, C. Tomlin, and S. Sastry, "Controllers for reachability specifications for hybrid systems," *Automatica*, vol. 35, no. 3, pp. 349–370, Mar. 1999.

[6] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi, "Algorithmic analysis of nonlinear hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 540–554, Apr. 1998.

[7] R. Alur, T. Dang, and F. Ivani, "Predicate abstraction for reachability analysis of hybrid systems," *ACM Transactions on Embedded Computing Systems*, vol. 5, no. 1, pp. 152–199, Feb. 2006.

[8] A. Girard, A. A. Julius, and G. J. Pappas, "Approximate simulation relations for hybrid systems," *Discrete Event Dynamic Systems*, vol. 18, no. 2, pp. 163–179, Oct. 2007.

[9] E. Asarin, O. Bournez, T. Dang, and O. Maler, "Approximate reachability analysis of piecewise-linear dynamical systems," in *Hybrid Systems: Computation and Control*. Springer, Mar. 2000, no. 1790, pp. 20–31.

[10] P. Gagarinov and A. A. Kurzhanskiy, "Ellipsoidal Toolbox," 2014. [Online]. Available: http://systemanalysisdpt-cmc-msu.github.io/ellipsoids/

[11] T. Dreossi, T. Dang, and C. Piazza, "Parallelotope bundles for polynomial reachability," in *Proc. 19th International Conference on Hybrid Systems: Computation and Control (HSCC)*. ACM, 2016, pp. 297–306.

[12] T. Moor and J. Raisch, "Abstraction based supervisory controller synthesis for high order monotone continuous systems," in *Modelling, Analysis, and Design of Hybrid Systems*. Springer, 2002, no. 279, pp. 247–265.

[13] D. Del Vecchio, M. Malisoff, and R. Verma, "A separation principle for a class of hybrid automata on a partial order," in *Proc. American Control Conference (ACC)*, 2009, pp. 3638–3643.

[14] M. R. Hafner and D. Del Vecchio, "Computational tools for the safety control of a class of piecewise continuous systems with imperfect information on a partial order," *SIAM Journal on Control and Optimization*, vol. 49, pp. 2463–2493, 2011.

[15] Massachusetts Department of Transportation, "2012 Top Crash Locations Report," 2014. [Online]. Available: https://www.massdot.state.ma.us/Portals/8/docs/traffic/CrashData/12TopCrashLocations-Rpt.pdf

[16] A. Colombo and D. Del Vecchio, "Efficient algorithms for collision avoidance at intersections," in *Proc. 15th international conference on Hybrid Systems: Computation and Control (HSCC)*. ACM, 2012, pp. 145–154.

[17] ——, "Least restrictive supervisors for intersection collision avoidance: A scheduling approach," *IEEE Transactions on Automatic Control*, 2014.

[18] L. Bruni, A. Colombo, and D. Del Vecchio, "Robust multi-agent collision avoidance through scheduling," in *Proc. IEEE Conference on Decision and Control (CDC)*, Dec. 2013, pp. 3944–3950.

[19] H. Ahn, A. Colombo, and D. Del Vecchio, "Supervisory control for intersection collision avoidance in the presence of uncontrolled vehicles," in *Proc. American Control Conference (ACC)*, Jun. 2014, pp. 867–873.

[20] H. Ahn, A. Rizzi, A. Colombo, and D. Del Vecchio, "Experimental testing of a semi-autonomous multi-vehicle collision avoidance algorithm at an intersection testbed," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2015, pp. 4834–4839.

[21] H. Ahn and D. Del Vecchio, "Semi-autonomous intersection collision avoidance through job-shop scheduling," in *Proc. 19th International Conference on Hybrid Systems: Computation and Control (HSCC)*. ACM, Apr. 2016, pp. 185–194.

[22] J. Peng and S. Akella, "Coordinating Multiple Double Integrator Robots on a Roadmap: Convexity and Global Optimality," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Apr. 2005, pp. 2751–2758.

[23] ——, "Coordinating multiple robots with kinodynamic constraints along specified paths," *The International Journal of Robotics Research*, vol. 24, no. 4, pp. 295–310, Apr. 2005.

[24] K.-D. Kim and P. Kumar, "An MPC-based approach to provable system-wide safety and liveness of autonomous ground traffic," *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3341–3356, Dec. 2014.

[25] M. Kamal, J. Imura, T. Hayakawa, A. Ohata, and K. Aihara, "A Vehicle-Intersection Coordination Scheme for Smooth Flows of Traffic Without Using Traffic Lights," *IEEE Transactions on Intelligent Transportation Systems*, 2014.

[26] F. Zhu and S. V. Ukkusuri, "A linear programming formulation for autonomous intersection control within a dynamic traffic assignment and connected vehicle environment," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 363–378, Jun. 2015.

[27] N. Murgovski, G. R. de Campos, and J. Sjöberg, "Convex modeling of conflict resolution at traffic intersections," in *Proc. IEEE Conference on Decision and Control (CDC)*, Dec. 2015, pp. 4708–4713.

[28] L. Chen and C. Englund, "Cooperative intersection management: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 570–586, Feb 2016.

[29] R. Tachet, P. Santi, S. Sobolevsky, L. I. Reyes-Castro, E. Frazzoli, D. Helbing, and C. Ratti, "Revisiting street intersections using slot-based systems," *PLoS ONE*, vol. 11, no. 3, p. e0149607, Mar. 2016.

[30] F. Altché, X. Qian, and A. de La Fortelle, "Time-optimal coordination of mobile robots along specified paths," 2016. [Online]. Available: http://arxiv.org/abs/1603.04610

[31] A. Richards, T. Schouwenaars, J. P. How, and E. Feron, "Spacecraft Trajectory Planning with Avoidance Constraints Using Mixed-Integer Linear Programming," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 4, pp. 755–764, 2002.

[32] L. Pallottino, E. Feron, and A. Bicchi, "Conflict resolution problems for air traffic management systems solved with mixed integer programming," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 1, pp. 3–11, Mar. 2002.

[33] A. E. Vela, S. Solak, J. P. B. Clarke, W. E. Singhose, E. R. Barnes, and E. L. Johnson, "Near real-time fuel-optimal en route conflict resolution," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 4, pp. 826–837, Dec. 2010.

[34] A. Alonso-Ayuso, L. Escudero, and F. Martn-Campo, "Collision Avoidance in Air Traffic Management: A Mixed-Integer Linear Optimization Approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 1, pp. 47–57, Mar. 2011.

[35] A. Alonso-Ayuso, L. F. Escudero, and F. J. Martn-Campo, "On modeling the air traffic control coordination in the collision avoidance problem by mixed integer linear optimization," *Annals of Operations Research*, vol. 222, no. 1, pp. 89–105, Mar. 2013.

[36] J. Omer, "A space-discretized mixed-integer linear model for air-conflict resolution with speed and heading maneuvers," *Computers & Operations Research*, vol. 58, pp. 75–86, Jun. 2015.

[37] M. Soler, M. Kamgarpour, J. Lloret, and J. Lygeros, "A hybrid optimal control approach to fuel-efficient aircraft conflict avoidance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 1826–1838, Jul. 2016.

[38] W. Chen, J. Chen, Z. Shao, and L. T. Biegler, "Three-dimensional aircraft conflict resolution based on smoothing methods," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 7, pp. 1481–1490, 2016.

[39] L. Pallottino, V. Scordio, A. Bicchi, and E. Frazzoli, "Decentralized Cooperative Policy for Conflict Resolution in Multivehicle Systems," *IEEE Transactions on Robotics*, vol. 23, no. 6, pp. 1170–1183, Dec. 2007.

[40] T. Keviczky, F. Borrelli, K. Fregene, D. Godbole, and G. J. Balas, "Decentralized receding horizon control and coordination of autonomous vehicle formations," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 1, pp. 19–33, 2008.

[41] W. Zhang, M. Kamgarpour, D. Sun, and C. J. Tomlin, "A hierarchical flight planning framework for air traffic management," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 179–194, Jan. 2012.

[42] M. Pinedo, *Scheduling*, 4th ed. Springer, 2012.

[43] IBM Corporation, "CPLEX User's Manual," 2015. [Online]. Available: http://www.ibm.com/support/knowledgecenter/SSSA5P_12.6.3/ilog.odms.studio.help/pd

[44] M. Hafner, D. Cunningham, L. Caminiti, and D. Del Vecchio, "Cooperative collision avoidance at intersections: algorithms and experiments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1162–1175, 2013.