A Minimax Framework for Two-Agent Scheduling With Inertial Constraints

Feihong Yang, Member, IEEE, and Yuan Shen, Senior Member, IEEE

Abstract—Multi-agent scheduling problems are common in applications such as intelligent transportation and smart manufacturing. When the agents are non-cooperative and inertially constrained, finding a safe and efficient policy under the trajectory uncertainty of other agents is a non-trivial problem. In this article, we establish a minimax framework to optimize the worst-case scheduling performance under the two-agent model. Specifically, a unified representation is proposed to characterize the trajectory uncertainty of the other agent, and a function is derived to evaluate different target states. Based on this evaluation, we further develop a control policy by adopting the minimax method, where a trajectory leading to the most robust target state is generated at each step. Algorithms are also provided to ensure the computational tractability of the policy. Furthermore, the safety of the policy is proved, and the global robustness is verified by numerical simulations, which show that the proposed policy reduces the worst-case scheduling cost by 13.1% compared with heuristic policies.

Index Terms—Minimax scheduling, inertial constraints, distributed control.

I. INTRODUCTION

MODERN communication and control technologies enable the proliferation of autonomous agents such as robots, intelligent vehicles and unmanned aerial vehicles (UAVs). Compared with static devices, their dynamics greatly extend the scope of industrial and civil applications to scenarios including environment sensing, intelligent transportation and smart manufacturing [1], [2], [3], [4]. In multi-agent systems, abundant coordination and competition schemes can be developed for different tasks, which attract massive attention in various researches [5], [6], [7].

When multiple agents in a system require a limited resource, scheduling schemes are necessary to avoid conflicts, as shown in Fig. 1. To model the characteristics of multi-agent scheduling problems, the following two system settings are considered in this article: (i) the agents are inertially constrained, i.e., their maximum accelerations and decelerations are limited; (ii) non-cooperative agents exist, and any agent does not have access to the policies or the future trajectories of the non-cooperative agents.

In existing studies of multi-agent scheduling problems, the inertial constraints are often considered, especially in the area of intelligent transportation. Based on the methods of handling the inertial constraints, these works can be divided into two main categories: hybrid solutions and direct solutions [8], [9], [10]. In hybrid solutions, upper-level scheduling problems without considering the inertial constraints are first formulated and solved to determine the occupation order or the occupation time period of the agents, and then lower-level trajectory planning policies are developed taking inertial constraints into consideration. Typical protocols for the upper-level problem include first-come-first-serve (FCFS) [11], adaptive traffic lights [12], [13], reservation-based or auction-based mechanisms [14], [15], [16], [17], [18], optimization methods [19], [20], [21], [22], [23], [24] and multi-agent learning methods [25]. In contrast, direct solutions determine the control variable of all agents by formulating optimization problems [26], [27], [28] or multi-agent learning frameworks [29] without introducing intermediate variables, which are generally more efficient but computationally more expensive. However, the works above in both categories require all agents to be cooperative, and thus safety cannot be ensured if there exists an agent not following the trajectory returned by the scheduling protocol.

Some other studies allow the existence of non-cooperative agents, while implicit assumptions on these agents are often supplemented. For example, traffic lights are adopted in [30], [31], [32], and [33] to indirectly control the non-cooperative human-driven vehicles, and it is assumed that all vehicles will obey the traffic lights. In [34] and [35], the trajectories of the human-driven vehicles are predicted by behavior models such as CMetric [36] and level-k reasoning [37], and thus the scheduling efficiency relies on the validity of these behavior

1558-0016 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
models. Single-agent learning methods such as [38], [39], [40], and [41] implicitly assume that the motion of other agents is close to a certain training sample; otherwise, these methods cannot ensure safe scheduling. Therefore, the modelling of the non-cooperative agents in these researches is different from our system setting, since we assume that no information can be obtained on the future trajectories of non-cooperative agents. Specifically, we allow the existence of randomly-evolved agents which do not obey any rule and have unpredictable trajectories.

Nevertheless, there exist a few works [42], [43] taking both system settings into consideration. In [42], authors present an explicit solution for the two-agent safety control problem based on a partial order with imperfect state information. However, the scheduling performance is not considered, and no control is specified as long as the safety is ensured. A game-theoretic strategy for autonomous vehicles in presence of malicious vehicles is proposed in [43], where four different levels of malicousness are considered. However, the method cannot ensure safety when there exist “irrational agents”, i.e., agents that randomly choose their accelerations. Unlike these works, we aim to improve the scheduling efficiency while always ensuring the safety of the system.

Note that when both system settings are considered in a scheduling system, we can find an intrinsic challenge in the trajectory planning of an agent, which, to the best of our knowledge, is not analyzed in existing studies. On the one hand, the inertial constraint means that the agent cannot immediately reach another state with a large velocity difference. Therefore, the target state needs to be set in advance to ensure its reachability. On the other hand, the existence of non-cooperative agents means that the agent cannot determine which time period is safe for its resource occupation, and thus the target state cannot be set in advance. An example is provided in Fig. 2 to interpret the challenge.

On account of the information uncertainty, instead of optimizing the performance in a specific scenario, we consider all possible scenarios and adopt the worst-case performance as the criterion to compare different decisions. Specifically, a minimax scheduling framework is proposed to generate a robust policy. For simplicity, we adopt the two-agent model in this article, which is sufficient to illustrate the main idea and obtain non-trivial insights. The main contributions are as follows.

- We propose a unified representation to characterize the trajectory uncertainty of the other agent, which allows a natural information fusion of various observations.
- We derive a function reflecting the worst-case efficiency of different decisions, and thus the challenge caused by the inertial constraints and the information uncertainty can be quantitatively characterized.
- We establish a minimax control policy and develop algorithms to ensure its computational tractability. The proposed policy can always guarantee safe scheduling and achieve robust performance against the uncertainty of the other agent.

The rest of the article is organized as follows. Section II formulates the two-agent scheduling problem by introducing the scheduling cost. Then in Section III, we characterize the unified representation of situation information and propose the state value function under a fixed situation. The minimax control policy is established in Section IV along with the safety guarantee and the calculation methods. Section V provides intuitions and several generalizations of the framework, and Section VI presents numerical results on the state value function and the scheduling performance of the proposed policy. Finally, the article is concluded in Section VII.

**Notations:** Throughout the article, when we want to refer to a variable, such as $p_i(t)$ (the position of agent $i$ at time $t$), with emphasizing the corresponding trajectory which, for example, is denoted by $\sigma$, we use the notations with superscript $\sigma$, such as $p_i^\sigma(t)$. We use $(a, b)$ and $[a, b]$ to represent an open interval and a closed interval, respectively, and $(a, b)$ is used to represent an ordered pair to avoid ambiguity. For a set $\mathcal{A}$, $\text{card}(\mathcal{A})$ represents its cardinality and $\mu(\mathcal{A})$ represents its Lebesgue measure. For equations, “$\triangleq$” is used when one side is undefined and it represents the definition of this side.

## II. Problem Formulation

In this section, we introduce the system model and formulate the non-cooperative two-agent scheduling problem with inertial constraints.

### A. System Model

Consider a system with two agents $s_0$ and $s_1$, both of which move along a given track and require a unique resource in some future stages, as illustrated in Fig. 1.\(^1\) For each agent $s_i$ ($i \in \{0, 1\}$), an axis is attached to characterize its movement, where the origin is always set to be the position of the resource. Let $L_i$ be the resource demand of the agent $s_i$

\(^1\)The intersection shown in Fig. 1 is a topological intersection instead of a geometric one. In other words, issues such as the intersection angle of the tracks or the widths of the agents are beyond the scope of the article.
under its own axis, and let \( p_i(t) \), \( v_i(t) \) and \( a_i(t) \) be the position, velocity and acceleration of \( s_i \) at time \( t \), respectively. Therefore, the resource is occupied by \( s_i \) at time \( t \) if and only if \( p_i(t) \in (0, L_i) \).

Furthermore, we impose the following two kinematic constraints on the two agents that for \( i \in \{0, 1\} \) and \( t \in \mathbb{R}_+ \),

\[
0 \leq v_i(t) \leq v_M \quad (1) \\
-a_{i,m} \leq a_i(t) \leq a_{i,M} \quad (2)
\]

where \( v_M, a_{i,m}, a_{i,M} \) are positive numbers. Note that the maximum velocity \( v_M \) is set to be equal for the two agents, which can be achieved without loss of generality by rescaling the axis of either agent. The acceleration bounds (2) correspond to the *inertial constraint* which is common among physical agents. Under the two constraints (1) and (2), we can regard the pair

\[
x_i(t) \triangleq (p_i(t), v_i(t))
\]

as the *state* and regard \( a_i(t) \) as the *control* of \( s_i \) at time \( t \).

To ensure the safety of the system, we do not allow the two agents to simultaneously occupy the resource, i.e.,

\[
\text{card}\{t \mid p_i(t) \in (0, L_i)\} \leq 1, \quad \forall t \geq 0. \quad (4)
\]

Furthermore, by defining

\[
t_{i,\text{in}} \triangleq \inf\{t \mid p_i(t) > 0\} \\
t_{i,\text{out}} \triangleq \sup\{t \mid p_i(t) < L_i\}
\]

for \( i \in \{0, 1\} \), the safety condition is equivalent to

\[
(t_{0,\text{in}}, t_{0,\text{out}}) \cap (t_{1,\text{in}}, t_{1,\text{out}}) = \emptyset. \quad (7)
\]

Therefore, scheduling is required in order to meet the safety condition.

**B. Formulation of the Scheduling Problem**

In this article, we consider the scheduling problem from the perspective of the agent \( s_0 \). In other words, we can only control \( s_0 \) and know nothing about the future control of \( s_1 \). Nevertheless, we assume that the constraints (1) and (2) are known by \( s_0 \). Furthermore, \( s_0 \) is allowed to obtain information on history or current states of \( s_1 \) by observation or by communication with other possible cooperative devices. However, due to the non-ideal environment, observation or communication may suffer from failure and delay, and thus \( s_0 \) cannot predict when and which information can be successfully obtained in the future.

To ensure the existence of a safe scheduling, we assume that \( s_0 \) is able to stop before the resource. In other words, the initial state of \( s_0 \) satisfies

\[
p_0(0) + \frac{v_0(0)^2}{2a_{0,m}} \leq 0. \quad (8)
\]

We aim at developing a causal control policy for \( s_0 \) which generates a trajectory with a low scheduling cost. Specifically, we consider a discrete set of decision times \( \{t_k \mid k \geq 0\} \) with \( t_0 = 0 \) and \( t_k < t_{k+1} \) for any \( k \geq 0 \). Then at each decision time \( t_k \), the policy can make use of all collected causal information and return the trajectory of \( s_0 \) on the next time period \([t_k, t_{k+1}]\). Now we define the scheduling cost of \( s_0 \) under a fixed trajectory of \( s_1 \).

**Definition 1 (Scheduling Cost):** Fix the trajectory of \( s_1 \). Then for a given trajectory \( s_0 \) of \( s_0 \), the scheduling cost is defined by

\[
C(s_0; t_{1,\text{in}}, t_{1,\text{out}}) \triangleq \begin{cases}
-v_M^2 + \frac{1}{2a_{0,m}}(v_M - v_0(s_0))^2, & \text{if } (t_{0,\text{in}}, t_{0,\text{out}}) \cap (t_{1,\text{in}}, t_{1,\text{out}}) = \emptyset; \\
+\infty, & \text{otherwise}.
\end{cases} \quad (9)
\]

According to (9), only the control before \( t_{0,\text{in}} \) makes a difference to the cost as long as the safety condition is satisfied. For completeness, we always set the control of \( s_0 \) to be

\[
a_0(t) = a_{0,A}(t) \triangleq \begin{cases}
a_{0,M}, & \text{if } v_0(t) < v_M; \\
0, & \text{if } v_0(t) = v_M
\end{cases}
\]

for \( t \geq t_{0,\text{in}} \), since it is always the best choice to reduce \( t_{0,\text{out}} \) and decrease the possibility of conflict.

Under this constraint, when the safety condition holds, the scheduling cost can be interpreted as the final distance between \( s_0 \) and a virtual agent which satisfies \( p(0) = 0 \) and \( v(t) = v_M \) for all \( t \), as shown in Fig. 3. Furthermore, note that the two terms of the cost can be regarded as the “time cost” related to \( t_{0,\text{in}} \) and the “velocity cost” related to \( v_0(t_{0,\text{in}}) \), respectively.

Therefore, the scheduling cost is intuitive since an earlier occupation time and a quicker occupation velocity implies a higher scheduling efficiency.

In summary, the scheduling problem we consider in this article can be formulated as follows.

**Problem:** Designing a causal control policy for the agent \( s_0 \) to reduce the scheduling cost \( C(s_0; t_{1,\text{in}}, t_{1,\text{out}}) \), where \( s_0 \) is the output trajectory of the policy.

**Remark 1:** The proposed framework also works for other cost functions as long as the “time cost” is linear and increasing with \( t_{0,\text{in}} \) and the “velocity cost” is non-increasing with \( v_0(t_{0,\text{in}}) \). Different cost functions correspond to different tradeoffs during the scheduling.

Due to the causality constraint, \( t_{1,\text{in}} \) and \( t_{1,\text{out}} \) are unknown in advance, and thus the scheduling cost \( C(s_0; t_{1,\text{in}}, t_{1,\text{out}}) \) cannot be directly accessed by \( s_0 \). To tackle the difficulty,
we propose a minimax control policy which aims to optimize the worst-case performance against the uncertainty of $t_{1,in}$ and $t_{1,out}$. The sketch of the following two sections is shown in Fig. 4.

### III. THEORETICAL PREPARATIONS

In this section, we characterize the obtained situation information from various sources with a unified form denoted by $T$, and then establish the state value function for a given situation.

Recall that all trajectories considered in this article need to satisfy the kinematic constraints (1) and (2). For convenience, we first provide some definitions on the agent trajectories.

**Definition 2 (Trajectory Set):** Let $t_0 \geq 0$ and $x = (p, v)$ with $0 \leq v \leq v_M$. Then for an agent $s_1$, define

$$
\Sigma_t(x; t_0) \triangleq \begin{cases}
\sigma & \left. \begin{array}{l}
x_i^t(t_0) = x \\
0 \leq v_i^t(t) \leq v_M, \forall t \geq 0 \\
a_{i,in} \leq a_i^t(t) \leq a_{i,out}, \forall t \geq 0 
\end{array} \right\}
\end{cases}
$$

be the set of all trajectories of $s_1$ with the state $x$ at time $t_0$.

**Definition 3 (Typical Trajectories):** Let $T \subseteq \mathbb{R}_+$ be an interval and let

$$
a_{i,A}(t) \triangleq \begin{cases}
a_{i,M}, & \text{if } v_i(t) < v_M; \\
0, & \text{if } v_i(t) = v_M.
\end{cases}
$$

$$
a_{i,D}(t) \triangleq \begin{cases}
a_{i,M}, & \text{if } v_i(t) > 0; \\
0, & \text{if } v_i(t) = 0.
\end{cases}
$$

Then a trajectory $\sigma$ of $s_1$ is called

- a DEC trajectory on $T$, if $a_i^t(t) = a_{i,D}(t)$ for all $t \in T$;
- an ACC trajectory on $T$, if $a_i^t(t) = a_{i,A}(t)$ for all $t \in T$;
- a DEC-ACC trajectory on $T$, if there exists $t_0^F \in T$, such that $a_i^t(t) = a_{i,D}(t)$ for $t \in T \cap (0, t_0^F)$ and $a_i^t(t) = a_{i,A}(t)$ for $t \in T \cap (t_0^F, +\infty)$;
- an ACC-DEC trajectory on $T$, if there exists $t_0^A \in T$, such that $a_i^t(t) = a_{i,A}(t)$ for $t \in T \cap (0, t_0^A)$ and $a_i^t(t) = a_{i,D}(t)$ for $t \in T \cap (t_0^A, +\infty)$.

### A. Information Extraction

In this subsection, we fix the current time $t_{cur}$ and aim at extracting useful information for the scheduling task of $s_0$. We assume that $p_0(t_{cur}) \leq 0$; otherwise, the scheduling is unnecessary. Furthermore, the following technical assumption is made to avoid the infinite occupation time of $s_1$.

**Assumption 1:** $t_{1,out} \leq B$, where $B$ is a large enough positive number.

Note that $p_0(t_{cur}) \leq 0$ implies $t_{cur} \leq t_{0,in} \leq t_{0,out}$. According to the scheduling cost defined in (9), it is easy to check that

$$C(\sigma_0; t_{1,in}, t_{1,out}) = C(\sigma_0; t_{1,in},\hat{t}_{1,out})$$

where

$$\hat{t}_{1,in} \triangleq \max\{t_{1,in}, t_{cur}\}, \quad \hat{t}_{1,out} \triangleq \max\{t_{1,out}, t_{cur}\}.$$ 

In other words, all information that $s_0$ needs for its scheduling at time $t_{cur}$ is contained in the pair $(\hat{t}_{1,in}, \hat{t}_{1,out})$.

Although the exact value of $(\hat{t}_{1,in}, \hat{t}_{1,out})$ is generally unknown at time $t_{cur}$, a subset $I \subseteq \mathbb{R}^2$ in which $(\hat{t}_{1,in}, \hat{t}_{1,out})$ must lie can always be obtained. Therefore, we can use this uncertainty set to characterize the situation information obtained by $s_0$ at $t_{cur}$. Specifically, if nothing is known about $s_1$ at $t_{cur}$, we can let

$$I = \left\{ (\hat{t}_{1,in}, \hat{t}_{1,out}) \left| \begin{array}{c}
\hat{t}_{1,in} = \max\{t_{1,in}, t_{cur}\} \\
\hat{t}_{1,out} = \max\{t_{1,out}, t_{cur}\} \end{array} \right. \right\}.$$ 

When some information on $s_1$ is available, $I$ can be narrowed to a subset of (16), and different types of information can be fused by intersecting their respective subsets.

In the next two propositions, we specifically calculate and characterize the uncertainty set $I$ for the scenario where the **accurate state** of $s_1$ at time $t_{obs}$ is $t_{cur}$. Note that this scenario allows a positive age of information and is common in various practical scenarios.

**Proposition 1:** Suppose $x_{s_1}(t_{obs}) = x_s$ is observed at time $t_{cur}$, where $t_{obs} \leq t_{cur}$. Then for any trajectory of $s_1$ according to the observation, $(\hat{t}_{1,in}, \hat{t}_{1,out})$ must lie in $I$, where $I$ is defined as follows.

(i) If $p_1(t_{obs}) \leq 0$, then

$$I = \left\{ (\hat{t}_{1,in}, \hat{t}_{1,out}) \left| \begin{array}{c}
\hat{t}_{1,in} \leq t_{1,in} \leq t_{1,in}^{out} \\
\hat{t}_{1,out} \leq t_{1,out} \leq t_{1,out}^{out}
\end{array} \right. \right\}.$$ 

where $\sigma_A, \sigma_D \in \Sigma_{I}(x_s; t_{obs})$ are the ACC trajectory and the DEC trajectory on $[t_{obs}, +\infty)$, respectively; for any $t_{s}$, $\sigma_{DA}(t_s), \sigma_{AD}(t_s) \in \Sigma_{I}(x_s; t_{obs})$ are the DEC-ACC trajectory and the ACC-DEC trajectory on $[t_{obs}, +\infty)$ satisfying $t_{1,in}^{out} = t_{1,in}^{out} = t_{s}$, respectively.

(ii) If $0 < p_1(t_{obs}) < L_1$, then

$$I = \left\{ (\hat{t}_{1,in}, \hat{t}_{1,out}) \left| \begin{array}{c}
\hat{t}_{1,in} \leq t_{cur} \\
\hat{t}_{1,out} \leq t_{1,out}^{out} \leq t_{1,out}^{out}
\end{array} \right. \right\}.$$ 

where $\sigma_A, \sigma_D \in \Sigma_{I}(x_s; t_{obs})$ are the ACC trajectory and the DEC trajectory on $[t_{obs}, +\infty)$, respectively.

\footnote{For the last inequality in (17), if the trajectory $\sigma_{DA}(t_{1,in})$ or $\sigma_{AD}(t_{1,in})$ does not exist for some $t_{1,in}$, we omit the corresponding side of the inequality.}
Proposition 2 implies that Proposition 2 can be intuitively checked. The last result in based on the same observation.

Fig. 5. The uncertainty set calculated by Proposition 1 at different $t_{\text{cur}}$

(iii) If $p_1(t_{\text{obs}}) \geq L_1$, then $I = \{t_{\text{cur}}, t_{\text{cur}}\}$.

Proof: See Appendix B in Supplementary Material. □

Proposition 2: Let $I$ be the set defined in Proposition 1. Then $I$ is closed and simply connected, and $I \cap \{t_{\text{in}} = t_s\}$ is either empty or a closed interval for any $t_s$. Furthermore, if $I \cap \{t_{\text{in}} = t_s\}$ is non-empty, let

$$
\hat{t}_{\text{out}}(t_s) = \min (I \cap \{t_{\text{in}} = t_s\}) \quad (19)
$$

$$
\hat{t}_{\text{in}}(t_s) = \max (I \cap \{t_{\text{in}} = t_s\}) \quad (20)
$$

Then both $\hat{t}_{\text{out}}(t_s) - t_s$ and $\hat{t}_{\text{in}}(t_s) - t_s$ are positive and non-decreasing functions with respect to $t_s$.

Proof: See Appendix C in Supplementary Material. □

Fig. 5 provides two examples of the uncertainty set $I$ which makes it satisfy Assumption 2.

More generally, states with smaller $V$ are more appealing based on the situation information $I$, since they have a potential to result in a smaller manageable cost $C^*$.

\[ V(x; t_{\text{cur}}) \triangleq \min_{\sigma_0 \in \Sigma_{\text{in}}(x; t_{\text{cur}})} C^*(\sigma_0; t_{\text{cur}}, I). \] (26)

Intuitively, states with smaller $V$ are more appealing based on the situation information $I$, since they have a potential to result in a smaller manageable cost $C^*$. However, (26) is
defined by a minimization problem and is indirect to calculate. Thus, we provide the next proposition characterizing a solution of the minimization in (26). Specifically, the problem of calculating the state value function is simplified to calculating the manageable cost for a specific DEC-ACC trajectory.

**Proposition 3:** Let \( \Lambda \) be the set of DEC-ACC trajectories on \([t_{\text{cur}}, +\infty)\) in \( \Sigma_0(x; t_{\text{cur}}) \) satisfying the robust safety condition (23). If \( \Lambda \neq \emptyset \), let \( \sigma^* \equiv \arg \min_{\sigma \in \Lambda} \tau^D_1 \). Then

\[
V(x; t_{\text{cur}}, I) = \begin{cases} 
C^*(\sigma^*; t_{\text{cur}}, I), & \text{if } \Lambda \neq \emptyset; \\
+\infty, & \text{otherwise.} 
\end{cases}
\]  

(27)

Furthermore, if \( \Lambda \neq \emptyset \), then \( \sigma^* \) has the smallest \( t_{0,\text{in}} \) and the largest \( v_0(t_{0,\text{in}}) \) within all trajectories in \( \Sigma_0(x; t_{\text{cur}}) \) satisfying (23).

**Proof:** See Appendix D in Supplementary Material. 

---

**IV. MINIMAX CONTROL POLICY**

In this section, we establish the framework of the minimax control policy for the agent \( s_0 \), present the safety guarantee and provide algorithms for the implementation of the policy.

**A. Policy Statement and Safety Guarantee**

Recall that decisions are made at times \( \{t_k | k \geq 0\} \) where \( t_0 = 0 \) and \( t_k < t_{k+1} \) for any \( k \geq 0 \). Specifically, we only need to focus on the one-step decision at \( t_k \), when the state \( x_0(t_k) = (p_0(t_k), v_0(t_k)) \) has been fixed with \( p_0(t_k) \leq 0 \), and \( s_0 \) needs to determine the trajectory during the time interval \([t_k, t_{k+1}]\) based on the situation information \( I_k \).

Firstly, \( s_0 \) needs to determine whether there exists a trajectory \( \sigma \in \Sigma_0(x_0(t_k); t_k) \) allowing it to safely access the resource during the upcoming interval \([t_k, t_{k+1}]\). In other words, \( \sigma \) should simultaneously satisfy

\[
(t_{0,\text{in}}^\sigma, t_{0,\text{out}}^\sigma) \cap \mathcal{W}(I_k) = \emptyset, \quad t_{0,\text{in}}^\sigma \leq t_{k+1}. 
\]  

(28)

The next proposition characterizes an equivalent condition for the existence of such a trajectory which is easier to verify.

**Proposition 4:** Let \( \Lambda \) be the set of DEC-ACC trajectories on \([t_k, +\infty)\) in \( \Sigma_0(x_0(t_k); t_k) \) satisfying \( (t_{0,\text{in}}^\sigma, t_{0,\text{out}}^\sigma) \cap \mathcal{W}(I_k) = \emptyset \). If \( \Lambda \neq \emptyset \), let \( \sigma^* \equiv \arg \min_{\sigma \in \Lambda} \mathcal{W}(I_k) \). Then a trajectory \( \sigma \in \Sigma_0(x_0(t_k); t_k) \) satisfying (28) exists if and only if

\[
\Lambda \neq \emptyset, \quad t_{0,\text{in}}^\sigma \leq t_{k+1}. 
\]  

(29)

**Proof:** If \( \sigma^* \) satisfying (29) exists, then it can serve as the \( \sigma \) we are looking for. Conversely, if the trajectory \( \sigma \) satisfying (28) exists, then \( V(x_0(t_k); t_k, I_k) \) is finite. By Proposition 3, \( \Lambda \neq \emptyset \) and we have \( t_{0,\text{in}}^\sigma \leq t_{0,\text{in}}^\sigma \leq t_{k+1} \), which completes the proof. \( \square \)

According to Proposition 4, if (29) holds, then \( s_0 \) is able to obtain the resource before \( t_{k+1} \). In this case, \( \sigma^* \) is exactly the trajectory which can result in the smallest manageable cost by Proposition 3, and thus \( s_0 \) should follow \( \sigma^* \) during \([t_k, t_{k+1}]\). On the contrary, if (29) does not hold, we divide the decision of \( s_0 \) at \( t_k \) into two steps. First, a target state \( x_{\text{tar}} = (p_{\text{tar}}, v_{\text{tar}}) \) is determined which satisfies \( p_{\text{tar}} \leq 0 \), and second, a trajectory

\( \sigma = \arg \min_{\sigma \in \Lambda} V(x_0(t_k); t_k, I_k) \) is the most preferred for the time \( t_k \).

\[
\frac{x_{\text{tar}}(t_{k+1})}{x_{\text{tar}}(t_{k+1}) = x_{\text{tar}}}. 
\]

(30)

**Algorithm 1** The Decision of \( s_0 \) at Time \( t_k \)

**Input:** \( L_0, v_M, a_{0,in}, a_{0,M}, x_0(t_k), I_k \). 

**Output:** Trajectory on \([t_k, t_{k+1}]\).

1. Find \( \sigma^* \) by Proposition 4.
2. if (29) holds then
3. Return \( \sigma^* \).
4. else
5. Solve (34) to obtain \( x_{\text{tar}} \in \mathcal{F}_k \).
6. Return an arbitrary trajectory \( \sigma \in \Sigma_0(x_0(t_k); t_k) \) on \([t_k, t_{k+1}]\) satisfying (30).
7. end if

\[
\sigma \in \Sigma_0(x_0(t_k); t_k) \text{ on the interval } [t_k, t_{k+1}] \text{ is generated which satisfies}
\]

\[
x_{\text{tar}}(t_{k+1}) = x_{\text{tar}}. 
\]

(31)

Then we can find that as long as \( x_{\text{tar}} \in \mathcal{F}_k \) is fixed, the choice of the specific trajectory \( \sigma \) in the second step does not make any difference for subsequent decisions. Therefore, we focus on the first step of the decision, which aims to determine which target state \( x_{\text{tar}} \in \mathcal{F}_k \) is the most preferred for the time \( t_k \).

Recall that the state value function \( V(x_{\text{tar}}; t_k, I_{k+1}) \) proposed in Definition 5 is a natural criterion to evaluate the preference of different target states. However, note that \( s_0 \) has no access to \( I_{k+1} \) at the decision time \( t_k \). Therefore, a minimax framework is adopted for the decision, where we take into account all possible \( I_{k+1} \) and optimize the worst-case state value function. Specifically, note that since \( I_{k+1} \) must be consistent with the current information \( I_k \), it must lie in

\[
\mathcal{F}_k = \left\{ I_{k+1} \mid I_{k+1} \subseteq \text{max}(t_{k+1}, I_k) \right\} 
\]

(32)

where

\[
\text{max}(t_{k+1}, I_k) \triangleq \left\{ (x_{f_{1,\text{in}}^0}, x_{f_{1,\text{out}}^0}) \mid (f_{1,\text{in}}^0, f_{1,\text{out}}^0) \in I_k \right\}. 
\]

(33)

Then the decision is made by

\[
\arg \min_{x_{\text{tar}} \in \mathcal{F}_k} \max_{I_{k+1} \in \mathcal{F}_k} V(x_{\text{tar}}; t_k, I_{k+1}). 
\]

(34)

The full procedure of the proposed control policy at the decision time \( t_k \) is concluded in Algorithm 1.

Now we analyze the safety of the policy; in other words, we need to show the output trajectory of the proposed policy satisfies the safety condition (7). Recall (24) and (26) that for a state \( x \) at time \( t_k \), \( V(x; t_k, I_k) < +\infty \) implies that there exists a trajectory in \( \Sigma_0(x; t_k) \) satisfying the robust safety condition, which intuitively means that the state \( x \) has the potential of maintaining safety in the future. Therefore, the safety of the policy can be verified by showing that \( V(x_0(t_k); t_k, I_k) < +\infty \)
always holds at each time $t_k$, which can be proved by induction. Specifically, we have the following theorem.

**Theorem 1:** Let $\{t_k \mid k \geq 0\}$ be a set of decision times of $s_0$ with $t_0 = 0$ and $t_k < t_{k+1}$ for any $k \geq 0$. At each decision time, $s_0$ makes decision according to Algorithm 1. Then the trajectory of $s_0$ always satisfies the safety condition (7).

**Proof:** See Appendix E in Supplementary Material. $\Box$

Finally, we provide a remark on the proposed framework.

**Remark 2:** Since the proposed framework focuses on determining the most robust action for the agent $s_0$ instead of designing a complete scheduling protocol, we did not consider the deadlocks between agents. In fact, deadlocks may occur as long as two non-cooperative agents exist where both agents know nothing on the other’s policy, regardless of what specific policies they adopt. To eliminate deadlocks, extra mechanisms need to be introduced. For example, we can directly specify the resource occupation order by a central node when both agents have stopped near the resource.

The next two subsections provide further characterizations and algorithms on (34).

### B. Range of Feasible Target States

In this subsection, we focus on calculating the objective function of the minimization in (34), i.e.,

$$V_{\max}(x_{\text{tar}}; t_{k+1}, I_k) \triangleq \max_{I_{k+1} \in \mathcal{J}} V(x_{\text{tar}}; t_{k+1}, I_{k+1})$$

which itself is represented by another optimization problem. For simplicity of notations, we omit $x_{\text{tar}}$ and $t_{k+1}$ which are always fixed in this subsection, and write

$$V_{\max}(I_k) \triangleq \max_{I_{k+1} \in \mathcal{J}} V(I_{k+1}).$$

According to Definitions 4 and 5, the state value function $V(I_{k+1})$ only depends on $V(I_{k+1})$, where the definition of $V(V)$ is provided in (22). By utilizing Assumption 2, for any $I_{k+1} \in \mathcal{J}$, we have

$$V(I_{k+1}) = (\hat{t}_{\text{in}}, \hat{t}_{\text{out}})$$

where

$$\hat{t}_{\text{in}} \triangleq \min \{\hat{t}_{\text{in}} \mid \exists \hat{t}_{\text{out}}, (\hat{t}_{\text{in}}, \hat{t}_{\text{out}}) \in I_k \}; \quad \hat{t}_{\text{out}} \triangleq \max \{\hat{t}_{\text{out}} \mid \exists \hat{t}_{\text{in}}, (\hat{t}_{\text{in}}, \hat{t}_{\text{out}}) \in I_k \}.$$

The calculation of $\hat{t}_{\text{in}}$ and $\hat{t}_{\text{out}}$ is illustrated in the left subfigure in Fig. 8. Therefore, we can use the alternative notation $V(\hat{t}_{\text{in}}, \hat{t}_{\text{out}})$ for $V(I_{k+1})$.

Now we characterize the range of the pair $(\hat{t}_{\text{in}}, \hat{t}_{\text{out}})$ in the next proposition.

**Proposition 6:** For a given situation information $I_k$, the corresponding range of $(\hat{t}_{\text{in}}, \hat{t}_{\text{out}})$ is

$$B(I_k) \triangleq \left\{ \left(\min\{\hat{t}_{\text{in}}(1), \hat{t}_{\text{in}}(2)\}, \max\{\hat{t}_{\text{in}}(1), \hat{t}_{\text{in}}(2)\} \right), \left(\min\{\hat{t}_{\text{out}}(1), \hat{t}_{\text{out}}(2)\}, \max\{\hat{t}_{\text{out}}(1), \hat{t}_{\text{out}}(2)\} \right) \in I_k \right\}.$$
Note that here we regard \( t \) and \( \tilde{t} \) as two free variables. Before solving (43), we first show another property of the optimization range \( B(I_k) \).

**Proposition 7.** Let \( (\underline{t}, \tilde{t}) \), \( (\underline{t}, \tilde{t}) \) \( \not\in B(I_k) \). Then the following four pairs are also in \( B(I_k) \):

\[
\begin{align*}
&\{\underline{t}, \max(\tilde{t}, \tilde{t})\}, \{\underline{t}, \max(\tilde{t}, \tilde{t})\}, \\
&\{\min(\underline{t}, \tilde{t}), \tilde{t}\}, \{\min(\underline{t}, \tilde{t}), \tilde{t}\}.
\end{align*}
\]

**Proof:** See Appendix G in Supplementary Material. \( \square \)

Now we analyze the monotonicity of the function \( V(\underline{t}, \tilde{t}) \), which is defined on the domain

\[
D \triangleq \{\underline{t}, \tilde{t} \mid t_k + 1 \leq \underline{t} \leq \tilde{t} \leq B\}.
\]

To start, we divide this domain into the following three parts\(^5\)

\[
\begin{align*}
D_1 & \triangleq \{\underline{t}, \tilde{t} \mid t_{\sigma_A}^{\text{out}} \leq \underline{t} \leq \tilde{t} \leq B\} \\
D_2 & \triangleq \{\underline{t}, \tilde{t} \mid t_{k+1} \leq \underline{t} \leq \tilde{t} \leq t_{\sigma_A}^{\in}\} \\
D_3 & \triangleq \{\underline{t}, \tilde{t} \mid t_{k+1} \leq \underline{t} \leq \tilde{t} \leq t_{\sigma_A}^{\text{out}}\}
\end{align*}
\]

in which \( \sigma_A \in \Sigma_0(\text{tar}_i \to t_{k+1}) \) is the ACC trajectory on \([t_{k+1}, +\infty)\). The next theorem analyzes the monotonicity of \( V(\underline{t}, \tilde{t}) \) on the three sub-regions defined above.

**Theorem 2:** For a given state \( \text{tar}_i \) of the\( t_k+1 \) time, the function \( V(\underline{t}, \tilde{t}) \) satisfies the following properties.

(i) \( V \) is constant on \( D_1 \).

(ii) \( V \) only depends on \( \tilde{t} - \underline{t} \) on \( D_2 \), and decreases with \( \tilde{t} - \underline{t} \).

Furthermore, the value of \( V \) on \( D_2 \) is always no larger than its value on \( D_1 \).

(iii) \( V \) is non-decreasing with respect to both \( \underline{t} \) and \( \tilde{t} \) on \( D_3 \).

**Proof:** See Appendix H in Supplementary Material. \( \square \)

Fig. 9 shows the graph of the function \( V(\underline{t}, \tilde{t}) \) from two angles under certain parameters, which supports our results in Theorem 2. Based on the theorem, solving the optimization problem (43) becomes straightforward, and the solution is presented in the next proposition.

\[^5\] Although we use the word “divide”, we set all three sub-regions closed for convenience, and thus their mutual intersections are one-dimensional line segments instead of the empty set.
A. Intuitions

In this subsection, we provide intuitive interpretations on the proposed minimax policy and discuss its advantages.

Intuitively, when facing another agent \( s_1 \), \( s_0 \) can choose from two high-level decisions:

(i) The aggressive decision: Trying to occupy the resource before \( s_1 \) under the premise of safety.

(ii) The conservative decision: Occupying the resource after \( s_1 \), but as early and as fast as possible.

Clearly, due to the inertial constraint of \( s_0 \), these two decisions will generate different trajectories. Under the first decision, \( s_0 \) tends to maintain the maximum velocity as long as the safety is ensured. In contrast, under the second decision, deceleration may be required in early stages.

However, due to the trajectory uncertainty of \( s_1 \), \( s_0 \) cannot tell in advance which high-level decision is better. Intuitively, the first choice is clearly more efficient if the trial succeeds; however, if \( s_1 \) moves too fast, then the failed trial will result in a large “velocity cost”. In contrast, the second choice can reduce the “velocity cost”, while it may face a large “time cost” if \( s_1 \) moves too slow. Therefore, no matter which high-level decision is made, it may turn out to be inefficient.

This article aims to find a balance between aggressive and conservative policies based on the criterion of “optimizing the worst-case performance”. Specifically, our framework does not make the high-level decision in advance. Instead, the low-level trajectory planning is directly performed, by which the possibilities of both high-level choices are still preserved. In each step, the tradeoff between acceleration and deceleration is obtained by optimizing the potential scheduling cost in the worst case, which includes both the “time cost” and the “velocity cost” part. Thus, the output trajectory is neither too aggressive nor too conservative, and it is reasonable for the proposed policy to achieve a robust scheduling performance.

The proposed framework has the following advantages. First, our framework considers the safety and the efficiency under all possible trajectories of \( s_1 \). In fact, if a policy is only designed based on a specific prediction of the trajectory of \( s_1 \), it will suffer from a large cost or even collision if the prediction is inaccurate. Second, the proposed minimax framework is based on theoretical analyses. Specifically, each step of the policy returns the most robust target state based on the obtained information, which guarantees the global robustness.

B. Generalizations

In this subsection, we introduce some generalizations on the applicable scenarios of the proposed minimax scheduling framework.

1) Fixed Resource Occupation Order: The proposed framework is applicable if the resource occupation order of the two agents is predetermined. Specifically, if \( s_0 \) has the priority, then it can ignore \( s_1 \) and follow the ACC trajectory. Otherwise, if \( s_1 \) has the priority, then \( s_0 \) should change the length of \( s_1 \) to \( L_1 + \alpha \) and change any position observation \( p_1(t_{obs}) \) to \( p_1(t_{obs}) + \alpha \) for a large enough \( \alpha \) in its calculation.

2) Multi-Agent Systems: In principle, the proposed framework can be generalized to multi-agent systems. Specifically, the safety condition should be changed to \( (t_{0, in}, t_{0, out}) \cap (t_{j, in}, t_{j, out}) = \emptyset \) for all other agents \( s_j \). The situation information \( I \) is then expressed by a \( 2(N-1) \)-dimensional uncertainty set, where \( N \) is the number of agents. Then the state value function and the minimax control policy can also be generalized accordingly. However, the algorithm in Section IV-C cannot be directly generalized, and new algorithms are required to efficiently solve the optimization problem.

Nevertheless, if the resource occupation order of all agents is predetermined, the two-agent framework can directly apply to multi-agent systems. Specifically, although multiple agents exist, \( s_0 \) only needs to consider two agents in its modelling, i.e., \( s_0 \) itself and the agent right before it.

3) Regularity of Information Collection: In our framework introduced above, we did not assume any regularity on the process of information collection. In other words, at a decision time \( t_k \), the uncertainty set for the next decision time \( t_{k+1} \) can take any element in \( F_k \) as defined in (32). However, a certain regularity can help us further narrow down the set \( F_k \). For example, if \( s_0 \) already knows at \( t_k \) that it will not obtain any new information at \( t_{k+1} \), then the maximum in (34) can be removed by taking \( I_{k+1} = \max(I_{k+1}, I_k) \); in contrast, if \( s_0 \) is confident to make undelayed observations at all decision times, then \( I_{k+1} \) must lie in a 2-dimensional subset of \( F_k \) parameterized by the observations. Note that although the framework is always adaptable, the algorithm in Section IV-C needs to be re-developed case by case.

4) Bounded Control Error: Assume there exist bounded control errors for the agent \( s_0 \). In other words, when \( s_0 \) tries to control itself to the state \( x_{tar} \) at time \( t_k \), it will actually arrive at the state \( x_{tar} + n \) at time \( t_{k+1} \), where the unknown error \( n \) is bounded. In this case, we can add a maximization over the control errors and change (34) to

\[
\min_{x_{tar} \in F_k} \max_n \max_{I_{k+1} \in F_k} V(x_{tar} + n; t_{k+1}, I_{k+1}).
\]

VI. Numerical Results

In this section, we provide numerical experiments to show the characteristics of the objective function, verify the performance of the proposed scheduling policy, and evaluate the effects of several parameters.

A. Objective Function \( V_{\max} \)

In this subsection, we numerically characterize the objective function \( V_{\max}(x_{tar}; t_{k+1}, I_k) \) of the one-step decision (34) at time \( t_k \), which is defined in (37). We assume that \( s_0 \) is able to observe \( x_1(t_k) \), based on which the situation information \( I_k \) can be calculated by Proposition 1. Then the graph of \( V_{\max}(x_{tar}; t_{k+1}, I_k) \) is shown in Fig. 10 with respect to \( x_{tar} = (p_{tar}, v_{tar}) \) for four different values of \( p_1(t_k) \).

According to Fig. 10, the general trend of \( V_{\max} \) decreases with both \( p_{tar} \) and \( v_{tar} \), which is intuitive since a state closer to the resource and with larger velocity is generally more appealing. However, the existence of \( s_1 \) makes the function...
discontinuous. Specifically, the right side of the transition has a lower value of $V_{\text{max}}$, since this region corresponds to the case that $s_0$ is always able to occupy the resource before $s_1$ by the ACC trajectory. With $p_1(t_k)$ approaching zero, the transition moves right accordingly and the gap becomes larger, since the room for $s_0$ to make adjustment decreases.

### B. Performance Comparison

In this subsection, we compare the output trajectory and the scheduling cost of the proposed policy with two other families of policies Queueing($d$) and Following($d$) for $d \geq 0$. Specifically, 5

- Queueing($d$) is the policy in which $s_0$ moves with the maximum possible acceleration which can lead to the state $(p_0, v_0) = (−d, 0)$ before it finds an ACC trajectory ensuring robust safety. Mathematically, $a_0(t) = a_{0,A}(t)$ if $a_{0,A}$ satisfies (23) or if $p_0(t) + \frac{v_0^2(t)}{2a_{0,m}} < −d$ holds; otherwise, $a_0(t) = a_{0,D}(t)$.

- Following($d$) is the policy in which $s_0$ moves with the maximum possible acceleration which can ensure $p_0(t) < p_1(t) − L_1 − d$ for all $t$ in the future before it finds an ACC trajectory ensuring robust safety. Mathematically, $a_0(t) = a_{0,A}(t)$ if $a_{0,A}$ satisfies (23) or if $p_0(t) + \frac{v_0^2(t)}{2a_{0,m}} < p_{1}^{p_1,i}(t) − L_1 − d$ holds for all $t' \geq t$; otherwise, $a_0(t) = a_{0,D}(t)$.

Intuitively, both Queueing($d$) and Following($d$) policies correspond to commonly-used ideas in trajectory planning, and they are also typical policies under the aggressive and the conservative high-level decisions in Section V-A, respectively.

Now we set the motion of $s_1$ in simulations. Assume that $s_1$ starts from the position $p_1(0)$ and maintains the constant velocity 15 until it reaches the position $−(v_F^2 − 15^2)/(2a_F)$; then, it changes the acceleration to $a_F$ until it reaches the position 0 with velocity $v_F$; finally, it follows the constant velocity $v_F$ until it releases the resource. 7 Here, $v_F$ is a free parameter; $a_F \triangleq a_{1,M}$ if $v_F \geq 15$ and $a_F \triangleq −a_{1,m}$ if $v_F < 15$. Furthermore, we assume that $s_0$ can always observe the real-time state of $s_1$.

Fig. 11 provides the output trajectories of different policies for $p_1(0) = −160$ and $v_F = 15$. First, we can find that within the Queueing($d$) policies and the Following($d$) policies, the change of $d$ provides different tradeoffs between $t_{0,\text{in}}$ and $v_F(t_{0,\text{in}})$, while the trend of the trajectories in each family is consistent. Second, the output trajectory of the proposed policy intuitively lies “between” the Queueing($d$) policies and the Following($d$) policies: the start time of deceleration is later than that of Following($d$), which can preserve the possibility of occupying the resource before $s_1$ when the situation is still unclear; in contrast, it is earlier than that of Queueing($d$), which can avoid a too small occupation velocity. This observation coincides with our intuition in Section V-A. Furthermore, we can also find that the proposed policy turns from deceleration to acceleration earliest among all policies.

Now we perform $10^5$ random experiments by choosing $p_1(0)$ uniformly in $[−200, −150]$ and choosing $v_F$ uniformly

---

5 Here, $a_{0,A}$, $a_{0,D}$ and $a_{1,D}$ represent the ACC or the DEC trajectory of $s_0$ or $s_1$ on $[r, +\infty)$, respectively.

7 Note that the motion of $s_1$ presented here is only the simulation setting, and $s_0$ is not informed of this knowledge when making decisions. Furthermore, we can check that this mobility pattern of $s_1$ is diverse enough. In other words, basically all policies of $s_0$ can encounter unfavorable situation evolutions under this simulation setting.
Specifically, the motion of the decision periods and observation periods on the output trajectory.

C. Effect of Decision and Observation Periods

Therefore, the simulation result verifies the robustness of the proposed policy against the trajectory uncertainty of $s_1$. Specifically, if we allow the decision times $t_k$ and $t_{\text{obs}}$ to be set to $k \delta_{\text{dec}}$ and $k \delta_{\text{obs}}$, respectively. According to the two figures, the safety of the system can always be guaranteed regardless of the choices of $\delta_{\text{dec}}$ and $\delta_{\text{obs}}$. Both a large $\delta_{\text{dec}}$ and a large $\delta_{\text{obs}}$ can degrade the scheduling performance. However, as long as the two parameters have been small enough (smaller than 0.3 under the given parameters), more frequent decisions or observations will no longer make a difference. Intuitively, “small enough” means that the velocity $v_1$ mainly occurs when $s_1$ occupies the resource after $s_0$ can adjust the state more freely, and thus it can delay the determination on whether to occupy the resource before or after $s_1$ until the situation becomes more clear, which can generally result in a smaller cost.

D. Effect of Different Inertial Constraints

In this subsection, we discuss the effect of different inertial constraints of $s_0$ and $s_1$, i.e., the bounds on their accelerations ($a_0, a_0, a_1, a_1$) and ($a_1, a_1, a_1, a_1$). For the motion of $s_1$, we still use the one proposed in Section VI-B with $\eta_{\text{FP}} = 15$. Then the relations between the scheduling cost $C$ and $p_1(0)$ for different ($a_0, a_0, a_1$) and ($a_1, a_1, a_1$) are shown in Fig. 15 and Fig. 16, respectively. Note that all curves are discontinuous: the left (or right) side of the discontinuity means that $s_0$ occupies the resource before (or after) $s_1$.

According to Fig. 15, the changes of $a_0$ and $a_0$ have similar effects on the scheduling cost, since both of them affect the control ability of $s_0$. Clearly, larger $a_0$ and $a_0$ mean that $s_0$ can adjust the state more free, and thus it can delay the determination on whether to occupy the resource before or after $s_1$. Therefore, the simulation result verifies the robustness of the proposed policy against the trajectory uncertainty of $s_1$. In this subsection, we evaluate the effect of different decision periods and observation periods on the output trajectory. Specifically, the motion of $s_1$ is the same as the last subsection with $p_1(0) = -160$ and $\eta_{\text{FP}} = 15$, and we assume that $s_0$ can observe the real-time state of $s_1$ at times $m \delta_{\text{obs}}$ ($m \geq 0$). The decision times $t_k$ are set to $k \delta_{\text{dec}}$.

Fig. 13 and Fig. 14 illustrate the output trajectories for different $\delta_{\text{dec}}$ and $\delta_{\text{obs}}$, respectively. According to the two figures, the safety of the system can always be guaranteed regardless of the choices of $\delta_{\text{dec}}$ and $\delta_{\text{obs}}$. Both a large $\delta_{\text{dec}}$ and a large $\delta_{\text{obs}}$ can degrade the scheduling performance. However, as long as the two parameters have been small enough (smaller than 0.3 under the given parameters), more frequent decisions or observations will no longer make a difference. Intuitively, “small enough” means that the velocity $v_1$ mainly occurs when $s_1$ occupies the resource after $s_0$ can adjust the state more freely, and thus it can delay the determination on whether to occupy the resource before or after $s_1$ until the situation becomes more clear, which can generally result in a smaller cost.

In contrast, $a_1$ and $a_1$ are related to the information collection ability of $s_0$. As shown in Fig. 16, the effect of $a_1$ mainly occurs when $s_0$ occupies the resource after $s_1$. Intuitively, a smaller $a_1$ means that $s_0$ can better estimate $t_{\text{out}}$ in advance, which earns more time for it to make preparations. On the other hand, the change of $a_1$ influences the position of the discontinuity point, since a smaller $a_1$ can

![Fig. 13. The output trajectories of the proposed policy with different decision periods $\delta_{\text{dec}}$. Parameters $v_0 = 20$, $a_0, a_1 = 4$, $a_0, a_1 = 3$, $L_0 = L_1 = 5$. $p_0(0) = -200$, $v_0(0) = 15$, $p_1(0) = -160$, $v_1(0) = 15$, and $\delta_{\text{obs}} = 0.1$. On each curve, the two markers represent $t = t_{0, \text{in}}$ and $t = t_{0, \text{out}}$.](image)

![Fig. 14. The output trajectories of the proposed policy with different observation periods $\delta_{\text{obs}}$. Parameters $v_0 = 20$, $a_0, a_1 = 4$, $a_0, a_1 = 3$, $L_0 = L_1 = 5$. $p_0(0) = -200$, $v_0(0) = 15$, $p_1(0) = -160$, $v_1(0) = 15$, and $\delta_{\text{dec}} = 0.1$. On each curve, the two markers represent $t = t_{0, \text{in}}$ and $t = t_{0, \text{out}}$. Note that the first three curves coincide.](image)

![Fig. 15. The scheduling cost $C$ of the proposed policy. Parameters $v_0 = 20$, $a_0, a_1 = 4$, $a_1, a_1 = 3$, $L_0 = L_1 = 5$. $t_k = 0.1k$. $p_0(0) = -200$, $v_0(0) = 15$, and $v_1(0) = 15$.](image)

\[ P_{\text{emp}}(C \geq c) \doteq 10^{-5} \cdot \text{card}(n \mid C^{(n)} \geq c) \]
eliminate possible conflicts caused by the acceleration of $s_1$, which enables $s_0$ to safely occupy the resource before $s_1$ in some critical cases.

VII. CONCLUSION

This article established a minimax framework for the two-agent scheduling problem, where the agents are non-cooperative and inertially constrained. To deal with the uncertainty of the other agent, we take all possibilities into account in order to ensure safety in all cases and improve the worst-case efficiency. At each step of the proposed policy, the agent first optimizes the worst-case state value function in order to find the most robust target state, and then generates a trajectory leading to this target state. We proved the safety of the policy, and verified its robustness through numerical simulations.

Intuitively, the proposed framework achieves a tradeoff between aggressive and conservative policies, and resolves the challenge in trajectory planning caused by the inertial constraints and the uncertainty of the other agent. Due to its low computational complexity, strong interpretability, high generalizability, and most importantly, the ability to safely and efficiently deal with complex environmental uncertainties, the minimax framework is also a promising solution in widespread practical applications.

REFERENCES

[1] J. Wan et al., “Software-defined industrial Internet of Things in the context of industry 4.0,” IEEE Sensors J., vol. 16, no. 20, pp. 7373–7380, Oct. 2016.
[2] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, “Industrial Internet of Things: Challenges, opportunities, and directions,” IEEE Trans. Ind. Informat., vol. 14, no. 11, pp. 4724–4734, Nov. 2018.
[3] G. Karagiannis et al., “Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions,” IEEE Commun. Surveys Tuts., vol. 13, no. 4, pp. 584–616, 4th Quart., 2011.

[4] L. Gupta, R. Jain, and G. Vaszkun, “Survey of important issues in UAV communication networks,” IEEE Commun. Surveys Tuts., vol. 18, no. 2, pp. 1123–1152, 2nd Quart., 2016.
[5] J. R. Marden, G. Arslan, and J. S. Shamma, “Cooperative control and potential games,” IEEE Trans. Syst., Man, Cybern., B (Cybern.), vol. 39, no. 6, pp. 1393–1407, Dec. 2009.
[6] Y. Cao, W. Yu, W. Ren, and G. Chen, “An overview of recent progress in the study of distributed multi-agent coordination,” IEEE Trans. Ind. Informat., vol. 9, no. 1, pp. 427–438, Feb. 2013.
[7] Y. Cai and Y. Shen, “An integrated localization and control framework for multi-agent formation,” IEEE Trans. Signal Process., vol. 67, no. 7, pp. 1941–1956, Apr. 2019.
[8] L. Chen and C. Englund, “Cooperative intersection management: A survey,” IEEE Trans. Intell. Transp. Syst., vol. 17, no. 2, pp. 570–586, Feb. 2016.
[9] J. Rios-Torres and A. A. Malikopoulos, “A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps,” IEEE Trans. Intell. Transp. Syst., vol. 18, no. 5, pp. 1066–1077, May 2017.
[10] A. Gholamhosseini and J. Seitz, “A comprehensive survey on cooperative intersection management for heterogeneous connected vehicles,” IEEE Access, vol. 10, pp. 7937–7972, 2022.
[11] A. I. Morales Medina, N. van de Wouw, and H. Nijmeijer, “Cooperative intersection control based on virtual platooning,” IEEE Trans. Intell. Transp. Syst., vol. 19, no. 6, pp. 1727–1740, Jun. 2018.
[12] B. Xu et al., “Cooperative method of traffic signal optimization and speed control of connected vehicles at isolated intersections,” IEEE Trans. Intell. Transp. Syst., vol. 20, no. 4, pp. 1390–1403, Apr. 2019.
[13] T. T. Phan, D. Ngoduy, and L. B. Le, “Space distribution method for autonomous vehicles at a signalized multi-lane intersection,” IEEE Trans. Intell. Transp. Syst., vol. 21, no. 12, pp. 5283–5294, Dec. 2020.
[14] K. L. Zhang, D. F. Zhang, A. de La Fortelle, X. Wu, and J. Grégoire, “State-driven priority scheduling mechanisms for driverless vehicles approaching intersections,” IEEE Trans. Intell. Transp. Syst., vol. 16, no. 5, pp. 2487–2500, Oct. 2015.
[15] M. O. Sayin, C.-W. Lin, S. Shiraisi, J. Shen, and T. Basar, “Information-driven autonomous intersection control via incentive compatible mechanisms,” IEEE Trans. Intell. Transp. Syst., vol. 20, no. 3, pp. 912–924, Mar. 2019.
[16] F. Yang and Y. Shen, “An adaptive approach for intersection scheduling with limited state information,” in Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC), Rhodes, Greece, vol. 1, Sep. 2020, pp. 1–6.
[17] Y. Zhang, L. Liu, Z. Lu, L. Wang, and X. Wen, “Robust autonomous intersection control approach for connected autonomous vehicles,” IEEE Access, vol. 8, pp. 124486–124502, 2020.
[18] D. Rey, M. W. Levin, and V. V. Dixit, “Online incentive-compatible mechanisms for traffic intersection auctions,” Eur. J. Oper. Res., vol. 293, no. 1, pp. 229–247, Aug. 2021.
[19] R. Hult, M. Zanon, S. Gros, H. Wymeersch, and P. Falcone, “Optimisation-based coordination of connected, automated vehicles at intersections,” Vehicle Syst. Dyn., vol. 58, no. 5, pp. 726–747, May 2020.
[20] P. Tallapragada and J. Cortes, “Hierarchical-distributed optimized coordination of intersection traffic,” IEEE Trans. Intell. Transp. Syst., vol. 21, no. 5, pp. 2100–2113, May 2020.
[21] A. I. Morales Medina, F. Cremeres, E. Lefeber, and N. van de Wouw, “Optimal access management for cooperative intersection control,” IEEE Trans. Intell. Transp. Syst., vol. 21, no. 5, pp. 2114–2127, May 2020.
[22] M. Cai et al., “Multi-lane unsignalized intersection cooperation with flexible lane direction based on multi-vehicle formation control,” IEEE Trans. Veh. Technol., vol. 71, no. 6, pp. 5778–5798, Jun. 2022.
[23] B. Chalaki and A. A. Malikopoulos, “A priority-aware replanning and resequencing framework for coordination of connected and automated vehicles,” IEEE Control Syst. Lett., vol. 6, pp. 1772–1777, 2022.
[24] S. D. Kumaravel, A. A. Malikopoulos, and R. Ayyagari, “Optimal coordination of platoons of connected and automated vehicles at signal-free intersections,” IEEE Trans. Intell. Vehicles, vol. 7, no. 2, pp. 186–197, Jun. 2022.
[25] Y. Xu, H. Zhou, T. Ma, J. Zhao, B. Qian, and X. Shen, “Leveraging multiagent learning for automated vehicles scheduling at nonsignalized intersections,” IEEE Internet Things J., vol. 8, no. 14, pp. 11427–11439, Jul. 2021.
[26] J. Lee and B. Park, “Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment,” IEEE Trans. Intell. Transp. Syst., vol. 13, no. 3, pp. 81–90, Mar. 2012.
[27] M. A. S. Kamal, J.-I. Imura, T. Hayakawa, A. Ohata, and K. Aihara, “A vehicle-intersection coordination scheme for smooth flows of traffic without using traffic lights,” IEEE Trans. Intel! Transp. Syst., vol. 16, no. 3, pp. 1136–1147, Jun. 2015.

[28] N. Chohan, M. A. Nazari, H. Wymeersch, and T. Charalambous, “Robust trajectory planning of autonomous vehicles at intersections with communication impairments,” in Proc. 57th Annu. Allerton Conf. Commun., Control, Comput. (Allerton), Monticello, IL, USA, Sep. 2019, pp. 832–839.

[29] A. P. Capasso, P. Maramotti, A. Dell’Eva, and A. Broggi, “End-to-end intersection handling using multi-agent deep reinforcement learning,” in Proc. IEEE Intell. Vehicles Symp. (IV), Nagoya, Japan, Jul. 2021, pp. 443–450.

[30] P. Li and X. Zhou, “Recasting and optimizing intersection automation as a connected-and-automated-vehicle (CAV) scheduling problem: A sequential branch-and-bound search approach in phase-time-traffic hyrerpnetwork,” Transp. Res. B, Methodol., vol. 105, pp. 479–506, Nov. 2017.

[31] P. Lin, J. Liu, P. J. Jin, and B. Ran, “Autonomous vehicle-intersection coordination method in a connected vehicle environment,” IEEE Trans. Intell. Transp. Syst., vol. 9, no. 4, pp. 37–47, Oct. 2017.

[32] K. Yang, S. I. Guler, and M. Menendez, “Isolated intersection control for various levels of vehicle technology: Conventional, connected, and automated vehicles,” Transp. Res. C, Emerg. Techn., vol. 72, pp. 109–129, Nov. 2016.

[33] A. Ghosh and T. Parisini, “Traffic control in a mixed autonomy scenario at urban intersections: An optimal control approach,” IEEE Trans. Intell. Transp. Syst., early access, Apr. 27, 2022, doi: 10.1109/TITS.2022.3166452.

[34] R. Chandra and D. Manocha, “GamePlan: Game-theoretic multi-agent planning with human drivers at intersections, roundabouts, and merging,” IEEE Robot. Autom. Lett., vol. 7, no. 2, pp. 2676–2683, Apr. 2022.

[35] R. Tian, N. Li, I. Kolmanovsky, Y. Yildiz, and A. R. Girard, “Game-theoretic modeling of traffic in unsignalized intersection network for autonomous vehicle control verification and validation,” IEEE Trans. Intell. Transp. Syst., vol. 23, no. 3, pp. 2211–2226, Mar. 2022.

[36] R. Chandra, U. Bhattacharya, T. Mittal, A. Bera, and D. Manocha, “CMetric: A driving behavior measure using centrality functions,” in Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS), Las Vegas, NV, USA, Oct. 2020, pp. 2035–2042.

[37] M. A. Costa-Gomes and V. P. Crawford, “Cognition and behavior in two-person guessing games: An experimental study,” Amer. Econ. Rev., vol. 96, no. 3, pp. 1737–1768, 2006.

[38] D. Isele, R. Rahimi, A. Cosgun, K. Subramanian, and K. Fujimura, “Navigating occluded intersections with autonomous vehicles using deep reinforcement learning,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), Brisbane, QLD, Australia, May 2018, pp. 2034–2039.

[39] Z. Qiao et al., “POMDP and hierarchical options MDP with continuous actions for autonomous driving at intersections,” in Proc. Int. Conf. Intell. Transp. Syst. (ITSC), Maui, HI, USA, Nov. 2018, pp. 2377–2382.

[40] X. Lin, J. Zhang, J. Shang, Y. Wang, H. Yu, and X. Zhang, “Decision making through occluded intersections for autonomous driving,” in Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC), Auckland, New Zealand, Oct. 2019, pp. 2449–2455.

[41] S. Kai, B. Wang, D. Chen, J. Hao, H. Zhang, and W. Liu, “A multi-task reinforcement learning approach for navigating unsignalized intersections,” in Proc. IEEE Intell. Vehicles Symp. (IV), Las Vegas, NV USA, Oct. 2020, pp. 1583–1588.

[42] 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 J. Control Optim., vol. 49, no. 6, pp. 2463–2493, 2011.

[43] S. Pruekprasert, X. Zhang, J. Dubut, C. Huang, and M. Kishida, “Decision making for autonomous vehicles at unsignalized intersection in presence of malicious vehicles,” in Proc. IEEE Intell. Transp. Syst. Conf. (ITSC), Auckland, New Zealand, Oct. 2019, pp. 2299–2304.

[44] F. Yang and Y. Shen, “Performance limit of two-agent scheduling with kinematic constraints,” in Proc. IEEE Global Commun. Conf. (GLOBECOM), Madrid, Spain, Dec. 2021, pp. 1–6.

Feihong Yang (Member, IEEE) received the B.E. degree (Hons.) in electronic engineering from Tsinghua University, Beijing, China, in 2017, where he is currently pursuing the Ph.D. degree with the Department of Electronic Engineering. His research interests include network localization, multi-agent systems, and cooperative control.

Yuan Shen (Senior Member, IEEE) received the B.E. degree in electronic engineering from Tsinghua University in 2005 and the S.M. and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), in 2008 and 2014, respectively. He is currently a Professor and the Vice Chair with the Department of Electronic Engineering, Tsinghua University. His research interests include network localization and navigation, integrated sensing and control, and multi-agent systems. He was a recipient of the IEEE ComSoc Fred W. Ellersick Prize and three Best Paper Awards from IEEE conferences. He has served as the TPC Symposium Co-Chair for IEEE ICC and IEEE GLOBECOM for several times. He was the Elected Chair for the IEEE ComSoc Radio Communications Committee from 2019 to 2020. He is an Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE WIRELESS COMMUNICATIONS LETTERS, and China Communications.