Distributed Scheduling at Non-Signalized Intersections With Mixed Cooperative and Non-Cooperative Vehicles

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

Abstract—Intersection management with mixed cooperative and non-cooperative vehicles is crucial in next-generation transportation systems. For fully non-cooperative systems, a minimax scheduling framework was established, while it is inefficient in mixed systems as the benefit of cooperation is not exploited. This paper focuses on the efficient scheduling in mixed systems and proposes a two-stage decision framework that makes full use of the cooperation. First, a long-horizon self-organization policy is developed to optimize the passing order of cooperative vehicles in a distributed manner, which is proved convergent when in-bound roads are sufficiently long. Then a short-horizon trajectory planning policy is proposed to improve the efficiency when an ego-vehicle faces both cooperative and non-cooperative vehicles, and its safety and efficiency are theoretically validated. Moreover, simulation results verify that the proposed policies can effectively reduce the scheduling cost and improve the throughput for cooperative vehicles.

Index Terms—Non-signalized intersections, distributed control, benefit of cooperation.

I. INTRODUCTION

Traffic congestion has become an increasing concern with the development of modern cities. As reported in [1], each auto commuter in the US suffers, on average, from a delay of 54 hours and an extra cost of $1170 due to congestions in 2019. Traffic intersections are recognized as a critical bottleneck in transportation systems, and safe and efficient intersection management protocols are crucial to improve the traffic fluency [2], [3], [4].

Among existing schemes, the traffic-signal-based scheme is the most widely-used and time-honored one [5], [6], [7], which require no autonomy or cooperation of the vehicles. Over the recent decades, many researchers focus on designing adaptive traffic signals to improve the performance metrics such as vehicle delay, intersection throughput and fuel consumption. Specifically, the lengths of the vehicle queues on the inbound roads are typically utilized to optimize the phase change [8], [9], [10]; based on the evolving sensor technologies, the real-time traffic information are further exploited in some recent works [11], [12], [13], where the future arrival table of the vehicles is predicted to improve the signal design. However, the efficiency of the signal-based scheme is limited since the vehicles cannot be provided with the future scheduling in advance. Furthermore, it is assumed by default that all vehicles will obey the signals, and the safety cannot be guaranteed when irrational vehicles exist.

Advancing technologies in autonomous and connected vehicles [14], [15] yielded various types of non-signalized scheduling schemes such as time-space resource reservation [16], [17], market-inspired auction [18], [19], centralized trajectory optimization [20], [21], [22], [23], [24], distributed protocols [25], [26] and multi-agent reinforcement learning [27], which can further improve the performance by utilizing the intelligence of vehicles. However, these schemes only work for fully cooperative systems. Over a long period of time, connected vehicles will have to face a mixed environment with both cooperative and non-cooperative vehicles. Thus, developing scheduling policies while addressing the trajectory uncertainty of non-cooperative vehicles becomes an important challenge.
To effectively make decisions with the existence of non-cooperative vehicles, some existing works try to reduce the uncertainty by adopting mathematical models and assumptions or utilizing data samples. For example, non-cooperative vehicles are assumed to obey a priority rule in [28], and to come to a full stop in front of the intersection in [29]. In [30], authors divide the non-cooperative vehicles into four groups according to their levels of maliciousness and develop a control policy based on game theory. It is shown that the ego-vehicle can safely pass the intersection when facing the three groups of non-cooperative vehicles with lower maliciousness; however, collisions may occur when the last group of irrational vehicles exist. A game-theoretic trajectory planning method called GAMEPLAN is proposed in [31] by assuming that all the non-cooperative vehicles follow the CMetric behavior model. In [32], an adaptive control policy and a rule-based control policy are established for the cooperative vehicles with the assumption that the non-cooperative vehicles are low-level reasoners. Learning-based policies are introduced in [33] and [34], in which a sample set is required to characterize the uncertainty of the non-cooperative vehicles. Generally, the policies in these works are effective under their respective assumptions. However, if the non-cooperative vehicles violate the given behavior model or the sampled patterns, the safety and the efficiency cannot be guaranteed.

Some other works take the full uncertainty of the non-cooperative vehicles into consideration. For example, an empirical decision framework is proposed in [35], where a fuzzy controller is introduced to determine the throttle and brake of the ego-vehicle based on the trajectory of the non-cooperative vehicles. For the two-vehicle system with a given order, a game-theoretic control algorithm is proposed in [36] which is proved safe. A least restrictive supervisor is designed in [37], which only provides a non-zero output when there is a risk of collision. Although the uncertainty of the non-cooperative vehicles is completely addressed in these works, they only focus on the safety issue without considering the scheduling efficiency.

Different from the works above, the minimax scheduling framework proposed in [38] can both ensure the safety and achieve a robust performance with the presence of non-cooperative vehicles with arbitrary trajectories, which resolves the main challenges caused by the non-cooperative vehicles. However, since the minimax framework is developed on a fully non-cooperative system model, it is inefficient when both cooperative and non-cooperative vehicles exist as it fails to exploit the benefit of cooperation. To overcome this drawback and improve the scheduling efficiency in mixed cooperative and non-cooperative systems, this paper focuses on utilizing the cooperation relation based on the minimax framework. Specifically, the cooperation between vehicles can help improve the efficiency from the following two aspects.

First, the passing order of cooperative vehicles can be optimized in advance. However, in most existing methods of optimizing the passing order, a centralized controller with strong communication and computation abilities is often required [22], [23], and thus these methods are costly to implement extensively. A decentralized order optimization method is proposed in [39], while it only works for two-road intersections and cannot be extended to more general scenarios.

Second, the trajectory planning policy can be further modified. Note that the policies against non-cooperative vehicles are generally more conservative to ensure the robustness. When facing cooperative vehicles, more efficient trajectories can be generated without increasing the risk of collision. In [40], a trajectory planning framework for cooperative vehicles is proposed; however, it is not applicable for mixed cooperative and non-cooperative systems.

In this paper, we establish a two-stage decision framework for the systems with mixed cooperative and non-cooperative vehicles to exploit the benefit of cooperation by the two intuitions above. First, a long-horizon self-organization policy is developed to determine the passing order of cooperative vehicles through distributed optimizations and trajectory adjustments. Then, a short-horizon trajectory planning policy is proposed to improve the overall efficiency of cooperative vehicles while ensuring the robustness against non-cooperative vehicles. Moreover, theoretical and numerical results validating the safety and efficiency of proposed framework are also provided. The main contributions are as follows.

- A two-stage decision framework is established which can exploit the benefit of cooperation and improve the overall efficiency in mixed cooperative and non-cooperative systems.
- A self-organization policy is proposed in the long-horizon stage which optimizes the passing order of the cooperative vehicles in a distributed manner.
- A trajectory planning policy is developed in the short-horizon stage which can achieve a balance between safety and efficiency in passing the intersection.

The rest of the paper is organized as follows. Section II presents the system model and the problem formulation. The general decision framework of the proposed solution is established in Section III, and the long-horizon self-organization policy and the short-horizon trajectory planning policy are proposed and analyzed in detail in Section IV and Section V, respectively. In Section VI, we provide numerical simulations to verify the effectiveness of the proposed decision framework. Finally, Section VII concludes the paper.

Notations: We use angle brackets \( \langle a_1, \ldots, a_n \rangle \) to represent vectors; parentheses \((a_1, a_2)\) and square brackets \([a_1, a_2]\) are used to represent open and closed intervals, respectively. We use \( I \) with a statement on the subscript to represent the indicator function, which takes 1 when the statement is true and takes 0 otherwise. For equations, \( \triangleq \) is used to represent the definitions. Subscripts are in Italic font if they represent variables (e.g., \( v_i \)), and are in Roman font otherwise (e.g., \( v_M \)).

II. PROBLEM FORMULATION

Consider the system with \( R \) roads indexed by 1, \ldots, \( R \) crossing over an intersection area, as shown in Fig. 1. On each road \( r \), let \((0, d_r)\) be the spatial range of the intersection area. Let \( S_r \) be the set of vehicles on the road \( r \), and let \( S \triangleq \bigcup_{r=1}^R S_r \) be the set of all vehicles. We assume that all vehicles always
move along the corresponding road,\textsuperscript{1} and as in [20], [21], [22], [23], [24], we only focus on the longitudinal movement of vehicles. Both cooperative and non-cooperative vehicles exist in the system, and let $S^{\text{coop}}$ and $S^{\text{non-coop}}$ be the corresponding vehicle sets. All cooperative vehicles follow the same policy, which we aim to design in this paper. In contrast, the policy of any non-cooperative vehicle is unknown and uncontrollable.

Remark 1: In our model, vehicles on different roads cannot pass the intersection area concurrently. By adopting this model, we can focus on the behavior of cooperative vehicles when facing possible conflict without being caught in dealing with the concurrency relationship. Similar system models were also adopted in existing literature such as [37] and [21].

For each vehicle $s_i \in S$, the length, front position, velocity and acceleration are denoted by $l_i$, $p_i(t)$, $v_i(t)$ and $a_i(t)$, respectively, where $t$ represents the time. The vector $(p_i(t), v_i(t))$ is also denoted as the state of $s_i$ at time $t$. The velocity and acceleration of the vehicles are constrained by

$$v_i(t) \in [0, v_{i,M}], \quad a_i(t) \in [-a_{i,m}, a_{i,m}], \quad \forall i, t. \quad (1)$$

Similar to some existing works [20], [21], we assume that $l_i$, $v_{i,M}$, $a_{i,m}$ and $a_{i,M}$ are equal for all cooperative vehicles, and we drop the subscript $i$ in this case. For each vehicle $s_i$, the initial position $p_i(0) < 0$,\textsuperscript{2} and the initial velocity $v_i(0) = v_{i,M}$. In order to ensure the safety of a vehicle $s_i \in S$, the following conditions need to be satisfied:

- For any other vehicle $s_j \in S$, on the same road, any position $x \in (0, d_r)$ within the intersection area should not be simultaneously occupied by $s_i$ and $s_j$, i.e.,
  
  $$\{ t \mid p_i(t) \in (x, x + l_i), p_j(t) \in (x, x + l_j) \} = \emptyset. \quad (2)$$

- For any vehicle $s_j \in S_r$ with $r' \neq r$, the intersection area should not contain $s_i$ and $s_j$ simultaneously, i.e.,
  
  $$\{ t \mid p_i(t) \in (0, l_i + d_r), p_j(t) \in (0, l_j + d_{r'}) \} = \emptyset. \quad (3)$$

Remark 2: Note that the two conditions above are objective criteria to judge the safety of $s_i$ and are NOT part of policy design. Thus, an extra safety distance is not required. Furthermore, as implied by (2), we do not allow overtaking in the intersection area, while overtaking can occur outside the intersection area. Furthermore, for each cooperative vehicle $s_i \in S_r$, we make the following assumptions.

- $s_i$ knows the existence of all vehicles.
- $s_i$ can actively request any desired information from other cooperative vehicles, and the communication delay is smaller than $\tau$.
- $s_i$ can passively receive the state information on non-cooperative vehicles, and the delay is not guaranteed.
- $s_i$ knows the indexes of the cooperative vehicles right in front of it and right behind it on the same road, denoted by

\textsuperscript{1}This assumption does not sacrifice the generality of the model. Specifically, if there exist vehicles turning from Road $r_1$ to Road $r_2$, we can define a new road to accommodate these vehicles.

\textsuperscript{2}Note that the two zeros in $p_i(0) < 0$ stand for different meanings. The former 0 represents the initial time, while the latter 0 represents the entrance position of the intersection.

Fig. 2. The illustration of the relative scheduling cost.

\[ \mathfrak{S}(s_i) \text{ and } \mathfrak{B}(s_i), \text{ respectively,} \] and also knows the index of the vehicle

$$C_r(s_i) \triangleq \arg \min_{s_j \in S, r \neq r'} |p_j(0) - p_i(0)| \quad (4)$$

for any $r' \neq r$.

Our goal is to ensure the safety and improve the overall efficiency for all cooperative vehicles through distributed decision and control. Specifically, we use the relative scheduling cost to evaluate the efficiency of a cooperative vehicle $s_i$, which is defined as follows.

Definition 1: The relative scheduling cost of the cooperative vehicle $s_i$ during the time interval $[0, t_0]$ is defined by

$$C_i(t_0) \triangleq p_i(0) + v_{M} t_0 - \frac{(v_{M} - v_i(0))^2}{2a_{M}}. \quad (5)$$

Remark 3: Intuitively, if there is no intersection area, the ideal trajectory of $s_i$ is to first accelerate to $v_{M}$ with the acceleration $a_{M}$ and then remain the constant velocity. Let $\sigma_0$ be the ideal trajectory of $s_i$ at time 0, and $\sigma_{t_0}$ be the ideal trajectory of $s_i$ at time $t_0$. Then $C_i(t_0)$ is the relative distance between $\sigma_0$ and $\sigma_{t_0}$ at a sufficiently large time (as shown in Fig. 2). Therefore, the relative scheduling cost quantifies the trajectory adjustment during the time interval $[0, t_0]$ caused by scheduling.

Based on the discussions above, the performance metric is defined by the average relative scheduling cost of cooperative vehicles, i.e.,

$$\overline{C} \triangleq \left\{ \begin{array}{ll}
+\infty, & \text{if any safety condition is not satisfied;} \\
\frac{1}{N^{\text{coop}}} \sum_{s_i \in S^{\text{coop}}} C_i(t_{i,\text{in}}), & \text{otherwise}
\end{array} \right. \quad (6)$$

where $N^{\text{coop}} \triangleq |S^{\text{coop}}|$ is the number of cooperative vehicles, and $t_{i,\text{in}}$ represents the time when $s_i$ enters the intersection area.

In our previous work [38], we considered the special case with only one cooperative vehicle (i.e., $S^{\text{coop}} = \{s_0\}$), and formulated a minimax policy for $s_0$ to optimize the worst-case performance against the trajectory uncertainty of the non-cooperative vehicles. Although the minimax policy is applicable in the general case with multiple cooperative vehicles, it is inefficient since it fails to make use of the cooperation among the vehicles in $S^{\text{coop}}$. Based on [38], this paper focuses on exploiting the benefit of cooperation while maintaining the robustness against non-cooperative vehicles.

\textsuperscript{3}For the first (or last) cooperative vehicle on a road, $\mathfrak{S}(s_i)$ (or $\mathfrak{B}(s_i)$) can be defined as a virtual vehicle with position $+\infty$ (or $-\infty$).
III. FRAMEWORK DESIGN

In this section, we establish the framework of the proposed distributed scheduling policy. We start from discussing the source of the benefit of cooperation. Intuitively, the following two types of benefit can be obtained through the cooperation of vehicles.

(i) When facing non-cooperative vehicles, the passing order can only be determined by analyzing their possible trajectory evolutions after they are close enough to the intersection area. However, among cooperative vehicles, the passing order can be determined by communication in advance, and thus cooperative vehicles can have more time to adjust their trajectories. Furthermore, the passing order of cooperative vehicles can also be optimized for a better performance.

(ii) In non-cooperative systems, the robustness against the trajectory uncertainty of the non-cooperative vehicles needs to be addressed, and specifically, we aimed to optimize the worst-case performance in our previous work [38]. However, cooperative vehicles will not behave maliciously when approaching the intersection area, and thus they do not need to be overcautious when facing each other near the intersection area. Therefore, more efficient policies can be established.

Note that the two types of benefit naturally correspond to different spatial regions. Specifically, to exploit the benefit (i), we should focus on the region where the vehicles are far from the intersection area; in contrast, the benefit (ii) should be exploited near the intersection area. Therefore, we divide the system into two stages based on the position range and separately develop policies to exploit the two benefits, as shown in Fig. 3.

The tasks of the two stages are listed in Table I. Note that the problems of high-level order determination and low-level trajectory planning are considered in both stages. Specifically, in the “long-horizon” stage, we address the order determination problem of cooperative vehicles, and also adjust the relative positions of cooperative vehicles based on their order consensus. In the “short-horizon” stage, we need to provide safe and efficient trajectories for cooperative vehicles to pass the intersection, during which the order between cooperative and non-cooperative vehicles is implicitly determined.

In the next two sections, we separately propose the policies in the long-horizon and the short-horizon stages. Specifically, we define the position ranges of the two stages to be \([-A - B, -B]\) and \([-B, 0]\), respectively.

IV. LONG-HORIZON SELF-ORGANIZATION

In this section, we develop and analyze the policy in the long-horizon position range \([-A - B, -B]\), in which the main step is the distributed self-organization.

A. Policy Statement

Recall Table I that the aim of the long-horizon stage includes forming a consensus on the passing order of the cooperative vehicles and correspondingly adjusting their relative positions. Specifically, we expect that the following two conditions are satisfied after this stage.

- Each cooperative vehicle \(s_i\) has specified the last cooperative vehicle that will pass the intersection area before itself, denoted by \(\Sigma(s_i)\). Furthermore, a total order of cooperative vehicles can be generated by defining \(s_i \leq \Sigma(s_i)\) for any cooperative vehicle \(s_i\).
- For any cooperative vehicle \(s_i\), let \(\tau_i\) and \(\iota_i\) be the times when \(s_i\) and \(\Sigma(s_i)\) pass the position \(-B\), respectively. Then \(v_i(\tau_i) = v_M\), and

\[
v_M \cdot (\tau_i - \iota_i) \geq l + \delta + d_{r_1} \cdot \mathbb{1}_{r_1 \neq r_2}
\]  

where \(r_1\) and \(r_2\) are the roads of \(s_i\) and \(\Sigma(s_i)\), respectively; \(\delta\) is a positive parameter which accounts for the extra safety distance between vehicles.

**Remark 4:** Under the conditions above, if there are no non-cooperative vehicles in the system, and all cooperative vehicles maintain the maximum velocity \(v_M\) after the long-horizon stage, then the safety conditions (2) and (3) will be satisfied. In other words, the conditions intuitively mean that the ideal trajectories of all cooperative vehicles are compatible after the long-horizon stage.

Note that a trivial solution can be obtained by preserving the initial order of cooperative vehicles, i.e., the order based on \(p_i(0)\) on \(\mathcal{S}_{\text{coop}}\). Specifically, each cooperative vehicle \(s_i\) only needs to search on the \(R\) roads to find

\[
\Sigma(s_i) = \arg \min_{s_j \in \mathcal{S}_{\text{coop}}, p_j(0) > p_i(0)} p_j(0)
\]  

obtain the value of \(\iota_i\) by communication, and then decelerate to meet the condition (7) if necessary. However, due to the asymmetry of the safety conditions (2) and (3), the initial order is often inefficient. Intuitively, the overall efficiency will be improved if vehicles from the same road pass the intersection area continuously.

**Remark 5:** Note that changing the order of vehicles from the same road will not benefit the efficiency. Therefore, for
the cooperative vehicles on the same road, the initial order is preserved in our solution without loss of generality.

The proposed solution divides the long-horizon position range \([-A-B,-B]\) into two sub-ranges with lengths \(A_1\) and \(A_2\), respectively, where \(A_1 + A_2 = A\). Specifically, we adopt the idea of self-organization to optimize the order of cooperative vehicles in the first sub-range \([-A-B,-A_2-B]\), and the passing order is determined according to the time when they reach the position \(-A_2-B\). Then in the second sub-range \([-A_2-B,-B]\), each cooperative vehicle \(s_i\) confirms the passing order to find \(\Sigma(s_i)\), and correspondingly adjusts its trajectory until (7) is satisfied. Clearly, the self-organization policy is the main distinction of our proposed solution.

Now we present the procedure of the self-organization policy. To start, let \(\{t_k' \mid k \geq 0\}\) be a set of times shared by all cooperative vehicles. Specifically, we set the gap between any adjacent \(t_{k-1}'\) and \(t_k'\) to be sufficiently large, so that (i) by fixing \(v_{i_1}(t_{k-1}') - v_{i_2}(t_k') = v_{m},\) the average velocity of any cooperative vehicle \(s_i\) during \([t_{k-1}', t_k']\) can arbitrarily choose from the interval \([v_{m} - v_{\text{var}}, v_{m}]\) under the dynamic constraints (1), where \(0 < v_{\text{var}} < v_{m}\) is a predetermined parameter denoting the variation of the velocity; and (ii) any cooperative vehicle can communicate with at least \(W + 2R\) cooperative vehicles during \([t_{k-1}', t_k']\), where \(W \geq 1\) is a parameter representing the scale of the vehicle set that will be used in order optimization and \(R\) is the number of roads in the system. Then, during the interval \([t_{k-1}', t_k']\), each vehicle \(s_i\) determines its trajectory in the next interval \([t_k', t_{k+1}']\) by the following three steps.

1) Information Collection: In this step, \(s_i\) collects the position information \(\{p_j(t_k') \mid s_j \in S_k^i\}\) of a subset \(S_k^i\) of cooperative vehicles, and performs a “rationalization” on this position information to obtain a function \(P_k^i : S_k^i \rightarrow \mathbb{R}\). By “rationalization,” we decrease the values of some obtained positions until

\[
P_k^i(\mathcal{G}(s_j)) - P_k^i(s_j) \geq l + \delta
\]

holds for all \(s_j \in S_k^i\) with \(\mathcal{G}(s_j) \in S_k^i\), where \(\mathcal{G}(s_j)\) represents the cooperative vehicle right in front of \(s_j\) on the same road based on the initial order.

The set \(S_k^i\) is determined by successively adding vehicles to its initialization \(\{s_i\}\) based on the previously obtained information, and the detailed method is shown in Algorithm 1. Intuitively, \(S_k^i\) should contain at least one vehicle on each road (Lines 3–4), and the last vehicle on each road in \(S_k^i\) should not be in front of \(s_j\) based on the order generated by \(P_k^i(\cdot) - d]/2\) (Lines 5–6). Furthermore, note that \(S_k^i\) contains \(W + 2R\) vehicles according to Line 2, which is larger than the scale \(W\) of the vehicle set to be used in the order optimization. This is because some vehicles will be dropped in the follow-up set truncation step, and this margin can empirically reduce the situation that less than \(W\) vehicles are preserved after the truncation.

Algorithm 1: The Information Collection Step of the Self-Organization Policy for \(s_i\) During \([t_{k-1}', t_k']\).

\begin{itemize}
  \item [Output:] \(S_k^i, P_k^i\)
  \item [1] Initialize \(S_k^i = \{s_i\}, \ P_k^i(s_i) = p_i(t_k')\).
  \item [2] while \(|S_k^i| < W + 2R\) do
  \item [3] if there exists some road \(r'\) with no vehicles in \(S_k^i\) then
  \item [4] Add the vehicle \(\mathcal{C}_r(s_i)\) to \(S_k^i\).
  \item [else]
  \item [5] if there exists some road \(r'\) on which the last vehicle (based on the initial order) in \(S_k^i\), denoted by \(s_j\), satisfies \(P_k^i(s_j) - d_r/2 > P_k^i(s_i) - d_r/2\) then
  \item [6] Add the vehicle \(\mathcal{F}(s_j)\) to \(S_k^i\).
  \item [else]
  \item [7] Find the road \(r'\) on which the first vehicle (based on the initial order) in \(S_k^i\), denoted by \(s_j\), has the smallest \(P_k^i(s_j) - d_r/2\) among all roads.
  \item [8] Add the vehicle \(\mathcal{F}(s_j)\) to \(S_k^i\).
  \item [9] Reset \(P_k^i(s_j) = p_i(t_k')\) for all \(s_j \in S_k^i\), and on each road \(r'\), successively update
  \[
P_k^i(s_m) \leftarrow \min \{P_k^i(s_m), P_k^i(\mathcal{G}(s_m)) - (l + \delta)\}
\]
  \item [10] where \(s_m\) varies from the second vehicle to the last vehicle (based on the initial order) in \(S_k^i \cap S_{r'}\).
\end{itemize}

2) Set Truncation: In this step, \(s_i\) truncates the set \(S_k^i\) to obtain a subset \(\tilde{S}_k^i\). Specifically, \(\tilde{S}_k^i\) is initialized as \(S_k^i\) and truncated by the following two steps:

(i) On each road \(r'\), remove all the vehicles \(s_j\) satisfying \(P_k^i(s_j) - d_r/2 < P_k^i(s_i) - d_r/2\).

(ii) If \(|\tilde{S}_k^i| > W\), then remove \(\tilde{S}_k^i\) from all vehicles \(s_j\) with the largest \(P_k^i(s_j) - d_r/2\) where \(r'\) is the road of \(s_j\).

In other words, \(\tilde{S}_k^i\) contains at most \(W\) vehicles \(s_j\) with the smallest \(P_k^i(s_j) - d_r/2\) among those satisfying \(P_k^i(s_j) - d_r/2 \geq P_k^i(s_i) - d_r/2\).

Remark 6: The step (i) of the truncation intuitively ensures the unidirectionality of the information flow during the distributed self-organization, which helps the convergence of the policy (see Theorem 1 in Section IV-B and its proof). The step (ii) aims to limit the complexity of the optimization problem.

3) Order Optimization: In this step, \(s_i\) solves an optimization problem and determines the target position \(p_i(t_{k+1}')\). Specifically, the optimization problem is formulated as follows.

\[
\min_{p_j} \sum_{s_j \in \tilde{S}_k^i} (P_k^i(s_j) - p_j)
\]

s.t. \(p_j^* \leq p_j(t_k'),\) \(\forall s_j \in \tilde{S}_k^i\)

\[
p_j^* - p_j^* \geq l + \delta,
\]

\[
\forall s_j, s_{j'} \in \tilde{S}_k^i \cap S_{r'}\text{ with }p_j^*(0) > p_{j'}^*(0), \forall r'
\]

\[
p_j^* - p_{j'}^* \geq l + \delta + d_{j'}\text{ or }p_{j'}^* - p_j^* \geq l + \delta + d_{j'}.
\]

\[
\forall s_j \in \tilde{S}_k^i \cap S_{r'}, s_{j'} \in \tilde{S}_k^i \cap S_{r'}, r' \neq r_{j'}^*.
\]

\[
\text{(11)}
\]

---

4 Considering the non-synchronization, an error in the scale of millisecond is allowed, which has little effect on the states of vehicles.

5 Note that \(p_j(t_k')\) is determined by \(s_j\) during the interval \([t_{k-2}', t_{k-1}']\). Therefore, \(p_j(t_k')\) is known during the current interval \([t_{k-1}', t_k']\).


Intuitively, the objective function in (11) represents the total position adjustment of all vehicles in $S_i^k$; the first constraint corresponds to the feasible condition; the last two constraints correspond to the safety conditions (2) and (3). After solving the optimization (11), we set the target state of $s_i$ at $t_{k+1}^i$ to be

$$p_i(t_{k+1}^i) = \max \{ p_i(t_k^i) + (vM - v_{\text{var}})(t_{k+1}^i - t_k^i),$$

$$p_i + vM(t_{k+1}^i - t_k^i) \} \quad (12)$$

$$v_i(t_{k+1}^i) = vM. \quad (13)$$

By (12), we ensure that the average velocity of $s_i$ during the time interval $[t_k^i, t_{k+1}^i]$ lies in $[vM - v_{\text{var}}, vM]$. Therefore, a corresponding trajectory satisfying the dynamic constraints (1) can be generated.

**Remark 7:** Note that the objective function in (11) can be regarded as a “local and online version” of our performance metric (6). On the one hand, they share the same form of total trajectory adjustment. On the other hand, the set of vehicles $S_i^k$ is a subset of $S_{\text{coop}}$ (local), and the baseline $P_i^k$ is the position information obtained by the proposed policy instead of the initial position at time 0 (online).

**Remark 8:** Compared with centralized order optimization schemes, the proposed self-organization policy is fully distributed. On the one hand, the communication and computation abilities of all cooperative agents are exploited, which avoids the high load on a small fraction of vehicles. On the other hand, cooperative vehicles can fully utilize the lengths of the roads to gradually adjust their trajectories, which improves the trajectory smoothness.

**B. Convergence Analyses**

In this subsection, we provide further analyses on the proposed self-organization policy. Note that at the same decision time, the optimization problems (11) formulated by different cooperative vehicles vary from each other. Nevertheless, in the next theorem, we show that this mismatch will not influence the effectiveness of the self-organization policy.

**Theorem 1:** Let $S_{\text{coop}}$ be a finite set. If all cooperative vehicles always follow the proposed self-organization policy, and $p_i(0) (\forall s_i \in S_{\text{coop}})$, $d_\epsilon(\forall r)$, $vM(t_{k+1}^i - t_k^i)(\forall k)$, $v_{\text{var}}(t_{k+1}^i - t_k^i)(\forall k)$ and $l + \delta$ are linearly independent under integer coefficients, then there exists $t_c < +\infty$ such that (i) all cooperative vehicles maintain the maximum velocity $vM$ after time $t_c$, and (ii) any two cooperative vehicles $s_{j_1}, s_{j_2} \in S_{t_c}$ satisfy

$$p_{j_1}(t_c) - p_{j_2}(t_c) \geq l + \delta + d'_j \cdot \mathbb{1}_{r_j \neq r'_j}$$

or

$$p_{j_2}(t_c) - p_{j_1}(t_c) \geq l + \delta + d'_j \cdot \mathbb{1}_{r_j \neq r'_j}. \quad (14)$$

**Proof:** See Appendix A. \hfill $\square$

Intuitively, (14) is directly related with the condition (7), which means that the positions of $s_{j_1}$ and $s_{j_2}$ have been sufficiently separated such that these two vehicles will not collide if they maintain the maximum velocity in the future. Therefore,

$$\text{Algorithm 2: The Short-Horizon Trajectory Planning Policy for } s_i \text{ at Time } t_{i,k}. \quad (15)$$

**Output:** Trajectory of $s_i$ over $[t_{i,k}, t_{i,k+1}]$

1. if $p_i(t_{i,k}) + \frac{v_i(t_{i,k})^2}{2\sigma_m} > 0$ then
   2. Return the trajectory with maximum acceleration.
3. else
4. Find $\sigma^*$, the trajectory with the smallest cost among all safe trajectories against non-cooperative vehicles based on the current knowledge. Let $\langle p^*, v^* \rangle$ be the state of $s_i$ at time $t_{i,k+1}$ corresponding to $\sigma^*$.
5. Find $F^i_{0,k}$, the set of states $\langle p_{\text{tar}}, v_{\text{tar}} \rangle$ satisfying:
   (i) $s_i$ can reach the state $\langle p_{\text{tar}}, v_{\text{tar}} \rangle$ at time $t_{i,k+1}$ by a feasible trajectory satisfying (1);
   (ii) $p_{\text{tar}} + \frac{v_{\text{tar}}^2}{2\sigma_m} \leq 0$;
   (iii) for any $t_D \geq t_{i,k+1}$, by fixing the state $\langle p_{\text{tar}}, v_{\text{tar}} \rangle$ of $s_i$ at $t_{i,k+1}$ and the state $\langle p_j(t_{\text{obs}}, v_j(t_{\text{obs}}) \rangle$ of $s_j$ at $t_{\text{obs}}$, and assuming that both $s_i$ and $s_j$ adopt the minimum acceleration before $t_D$ and adopt the maximum acceleration after $t_D$, it holds that

6. if $p_j(t_{\text{obs}}) + \frac{v_j(t_{\text{obs}})^2}{2\sigma_m} > 0$ and $p^* + \frac{v^*}{2\sigma_m} > 0$ then
7. Return $\sigma^*$.
8. else if $F^i_{0,k} = \emptyset$ then
9. Return the trajectory with minimum acceleration.
10. else
11. Obtain $p_{\text{tar}}$ and $v_{\text{tar}}$ by solving the minimax optimization problem (16).
12. Return an arbitrary trajectory over $[t_{i,k}, t_{i,k+1}]$ satisfying $p_i(t_{i,k+1}) = p_{\text{tar}}, v_i(t_{i,k+1}) = v_{\text{tar}}$.

**Remark 9:** The linear independence condition in Theorem 1 is introduced in order to avoid the case that the positions of two vehicles are equal, and it can be satisfied by randomly adding a small quantity to each parameter.

**V. SHORT-HORIZON TRAJECTORY PLANNING**

In this section, we focus on the policy development and analysis in the short-horizon stage. During this stage, the passing order of cooperative vehicles has been fixed.

**A. Policy Statement**

In this subsection, we propose the trajectory planning policy of the cooperative vehicles within the short-horizon position range $[-B, 0]$. Note that both cooperative and non-cooperative vehicles exist in the system, and we aim to improve the efficiency of cooperative vehicles while ensuring their safety.
We focus on the policy of a general cooperative vehicle $s_i \in S$, and let $\{t_{i,j} \mid k \geq 0\}$ be the set of its decision times satisfying $p_{i}(t_{i,0}) = -B$\textsuperscript{7}. Let $s_j = \mathcal{L}(s_i) \in S^\mathcal{L}$ be the last cooperative vehicle in front of $s_i$ based on the passing order. According to our assumption, $s_i$ can obtain the position and velocity information of $s_j$ through communication. Let $(p_j(t_{obs}), v_j(t_{obs}))$ be the latest information that $s_i$ has obtained on $s_j$ at time $t_{i,k}$, where $t_{obs} \leq t_{i,k}$.

The proposed trajectory planning policy for $s_i$ at time $t_{i,k}$ is shown in Algorithm 2. In the following, we explain how the proposed policy achieves a safe and efficient performance in the mixed cooperative and non-cooperative system from three perspectives.

First, the safety and robustness against the non-cooperative vehicles are achieved by solving the minimax optimization problem

$$\arg\min_{(p_{tar},v_{tar}) \in F_{i,k}^{fol}} \max_{k+1 \in C_{i,k}} V(p_{tar}, v_{tar}; t_{i,k+1}, K_{i,k+1})$$

(16)

in Line 11. This minimax problem was first introduced in [38]. Specifically, the objective function $V(p, v; t, K)$ provides an evaluation of the state $(p, v)$ under the situation information $K$; the inner optimization aims to find the worst situation $K_{i,k+1}$ based on the current knowledge $K_{i,k+1} \in F_{i,k}$; and the outer minimization aims to find the target state $(p_{tar}, v_{tar})$ resulting in the best worst-case performance. The detailed definition of $V(p, v; t, K)$ can be found in [38], and we provide a brief version in Appendix B for completeness. Note that the function $V(p, v; t, K)$ is directly related to the relative scheduling cost of the ego-vehicle $s_i$ based on its definition, and thus the optimization (16) is expected to effectively reduce our performance metric (6).

Second, the safety with the last cooperative vehicle $s_j$ is achieved by restricting the feasible space of $(p_{tar}, v_{tar})$ in the optimization problem (16) to the set $F_{i,k}^{fol}$ defined in Line 5. Specifically, the condition (iii) in Line 5 implies that $s_i$ will maintain a sufficient safety distance with $s_j$ if its state lies in $F_{i,k}^{fol}$.

Third, the benefit of cooperation is exploited by imposing extra restrictions on the output trajectory. Specifically, Lines 1–2 show that $s_i$ will always adopt the maximum acceleration after its state enters the region

$$C \triangleq \left\{(p, v) \mid p + \frac{v^2}{2a_m} > 0\right\}.
$$

(17)

Two intuitive benefits can be obtained by this design. First, the choice of trajectory (i.e., maximum acceleration) reduces the relative scheduling cost of the ego-vehicle $s_i$. Second, restricting the set of possible output trajectories can provide other cooperative vehicles with clearer information and resolve their concern on the risk of collision, which can benefit the overall efficiency.

\textsuperscript{7}Note that $t_{i,k}$ is different from $t_{i,k}^{\star}$ used in the long-horizon policy.

\textsuperscript{8}Recall that $\tau$ is the maximum communication delay among cooperative vehicles defined in Section II.

B. Performance Analyses

In this subsection, we analyze the performance of the proposed short-horizon trajectory planning policy from the perspective of safety and efficiency, respectively.

First, we show in the next theorem that the safety of cooperative vehicles can be ensured under the proposed policy.

\textbf{Theorem 2:} If all cooperative vehicles follow Algorithm 2 in the short-horizon position range $[-B, 0]$ and adopt the maximum acceleration after passing the position 0, and

$$B \geq \frac{v_{M}^2}{a_m} + \frac{v_{M}^2}{2a_M} + l + \max \{d_{\mathcal{L}} \mid 1 \leq r' \leq R\}$$

(18)

holds, then all cooperative vehicles can safely pass the intersection area.

\textit{Proof:} See Appendix C. \hfill $\Box$

Then, we characterize the efficiency of the proposed trajectory planning policy in the next theorem.

\textbf{Theorem 3:} Let $\mu$ be an upper bound of $t_{i,k+1} - t_{i,k}$ for all $s_i \in S^\mathcal{L}$ and $k \geq 0$, and define $\delta^* \triangleq \max \{v_{M}\} (1 + \tau) (1 + \frac{M}{\alpha M})$.

If (i) there are no non-cooperative vehicles in the system, (ii) $\nu_i(t_{i,0}) = v_{M}$ holds for any $s_i \in S^\mathcal{L}$, and (iii) $\nu_M(t_{j,0} - t_{j,0}) \geq l + \delta^* + d_{r'}^v \cdot \mathbb{1}_{r' \neq r'_2}$

or $\nu_M(t_{j,0} - t_{j,0}) \geq l + \delta^* + d_{r'}^v \cdot \mathbb{1}_{r' \neq r'_2}$

(19)

holds for any $s_j \in S^\mathcal{L} \cap S_{r'}$ and $s_j \in S^\mathcal{L} \cap S_{r'_2}$, then by following Algorithm 2, all cooperative vehicles will maintain the maximum velocity $v_{M}$ in the short-horizon position range $[-B, 0]$.

\textit{Proof:} See Appendix D. \hfill $\Box$

By noting the similarity between (19) and (7), the conditions in Theorem 3 mean that all vehicles are cooperative, and they have completed the long-horizon position adjustments where the safety distance parameter $\delta$ is not smaller than $\delta^*$. Under these conditions, Theorem 3 claims that vehicles will not further decelerate in the short-horizon stage. Therefore, the proposed short-horizon policy reduces unnecessary loss of efficiency by making full use of the cooperation relation.

C. Discussions

In this subsection, we discuss the insights of the proposed short-horizon trajectory planning policy.

The main challenge in the short-horizon stage is that each vehicle needs to consider the existence of both cooperative and non-cooperative vehicles. On the one hand, if a vehicle can adjust its trajectory more freely, it can better deal with the uncertainty of the non-cooperative vehicles. For example, in the minimax policy [38] designed for fully non-cooperative systems, the ego-vehicle is allowed to arbitrarily adjust its trajectory within its physical ability. On the other hand, the benefit of cooperation results from restricting the set of possible output trajectories. For example, we can request all vehicles to follow predetermined trajectories in a fully cooperative system. Therefore, in a mixed

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
cooperative and non-cooperative system, a balance needs to be found on how much the trajectory set should be restricted.

In order to achieve a balance, we define a set \( C \) in (17). For convenience, we use “stage I” and “stage II” to represent the stages before and after a vehicle enters the region \( C \), respectively. By the definition, the cooperative vehicles in stage II will always enter the intersection area in finite time regardless of their subsequent trajectories. Then we have the following intuitions.

- When a cooperative vehicle evaluates the risk of collision, stage I is more important. This is because for a vehicle in stage II, further adjustments can only make a slight difference on the time period of its intersection occupation. Therefore, the proposed policy allows free adjustments in stage I, which preserves the ability of cooperative vehicles to achieve a robust performance against the uncertainty of non-cooperative vehicles.

- When a cooperative vehicle evaluates the risk of collision with other cooperative vehicles, the vehicles in stage II should be paid more attention on since they are closer to the intersection area. In the proposed policy, we prohibit any trajectory adjustment for the cooperative vehicles in stage II, which brings crucial public information to help eliminate concerns and build trust among cooperative vehicles.

In conclusion, the proposed policy addresses both issues and thus can simultaneously ensure the safety and improve the efficiency.

VI. NUMERICAL RESULTS

In this section, we provide numerical results on the proposed long-horizon and short-horizon policies. In simulations, we set \( R = 3, d_c = 5 \) (yr), \( l_i = 5 \), \( v_{i,M} = 20 \), \( \alpha_{i,m} = 4 \), and \( \alpha_{i,M} = 3 \). Furthermore, let \( v_{\text{min}} = 1 \), \( \tau = 0.1 \), \( \mu = 0.1 \), \( \delta = 6^\circ \approx 9.33 \). The gaps between adjacent \( t_k \) are set to 2, and for each cooperative vehicle \( s_i \), the gaps between adjacent \( t_{i,k} \) are set to 0.1.

We set the time interval between two cooperative vehicles passing the position \(-A-B\) (i.e., entering the long-horizon stage) to be a random variable that follows independent exponential distribution with expectation \( 1/\lambda_{\text{coop}} \), and set the time interval between two non-cooperative vehicles passing the position \(-A-B\) to be a constant \( 1/\lambda_{\text{non-coop}} \). Thus, \( \lambda_{\text{coop}} \) and \( \lambda_{\text{non-coop}} \) are the traffic rates of cooperative and non-cooperative vehicles, respectively. Furthermore, each vehicle randomly lies on the three roads with equal probability, and the initial velocity is \( v_{i,M} \).

The trajectories of non-cooperative vehicles are set as follows. We randomly take a parameter \( v_{i,F} \in [5, 20] \) for each non-cooperative vehicle \( s_i \), and let \( s_i \) maintain the maximum velocity \( v_{i,M} = 20 \) before the position \(-A-B\). Decelerate from \( v_{i,M} = 20 \) to \( v_{i,F} \) with the fixed acceleration \(-3\) within the position range \([-20, 0]\), and maintain the velocity \( v_{i,F} \) after the position 0.\(^{10}\) Note that this trajectory information is unknown to the cooperative vehicles.

Remark 10: Note that our simulation setting is stochastic and includes diverse traffic scenarios. Specifically, the random arrival of cooperative vehicles can test the adaptation capability of the long-horizon policy, and the random behavior of non-cooperative vehicles near the intersection area can verify the performance of the short-horizon policy under different scenarios.

A. Output Trajectories of the Proposed Policies

In this subsection, we present the output trajectories of some cooperative vehicles in a specific experiment to intuitively show the effectiveness of the proposed policies. We set \( A_1 = 600, A_2 = 200 \) and \( B = 200 \). Thus, the self-organization policy is followed in the position range \([-1000, -200]\); the cooperative vehicles confirm their order in \([-400, -200]\); and the short-horizon trajectory planning policy is adopted in \([-200, 0]\). After passing the position 0, all cooperative vehicles adopt the maximum acceleration. Furthermore, we set \( W = 6 \) in the self-organization policy.

First, we track the movement of several cooperative vehicles when they are in the long-horizon stage, i.e., the position range \([-1000, -200]\), and show their positions at four times \( t = 0, 10, 20, 30 \) in Fig. 4. Note that if all vehicles maintain the maximum velocity 20 and make no adjustments, then the four subfigures would seem identical, since the time difference and the position offset between adjacent subfigures are 10 and 200, respectively. Therefore, the horizontal shift of a vehicle between subfigures in Fig. 4 characterizes the amount of its trajectory adjustments, which is always leftward as the vehicle can only decelerate to make adjustments. Now we focus on the movement of the three vehicles which lie in \([-870, -860]\) at time 0, i.e., the first yellow vehicle and the third and fourth blue vehicle.\(^{11}\)

\(^{10}\) This simulation setting coincides with the intuition that human-driven vehicles tend to decelerate when approaching the intersection area, and \( v_{i,F} \) characterizes the aggressiveness of the non-cooperative vehicle \( s_i \).

\(^{11}\) When referring to the “first/third/fourth vehicle,” we count from front (right) to back (left).
At time 0, the yellow vehicle has the largest position among the three vehicles, while it becomes the one with the smallest position at time 30. The change of order allows four blue vehicles on the same road to pass the intersection area continuously, and thus the final order is intuitively more efficient than the initial order. This shows that the proposed self-organization policy can effectively optimize the order of cooperative vehicles.

Then, we focus on the movement of vehicles in the short-horizon stage, and Fig. 5 presents the trajectories of several vehicles. From the figure, we have the following two observations. First, note that the position-time curves of the first two cooperative vehicles are straight lines, which means that these two vehicles always maintain the maximum velocity 20. This observation coincides with the result in Theorem 3. Second, due to the existence of the three non-cooperative vehicles, the last five cooperative vehicles adaptively decelerate to ensure the safety in the intersection area. This observation validates the safety guarantee in Theorem 2.

B. Effectiveness of Long-Horizon Self-Organization

In this subsection, we focus on the effectiveness of the proposed long-horizon self-organization policy. Since the performance metric (6) can only be calculated after all cooperative vehicles have entered the intersection area, we still need to simulate the short-horizon policy. To eliminate the influence of the short-horizon policy, we set $\lambda^{\text{non-coop}} = 0$ in simulations, in which case the short-horizon policy will not make extra trajectory adjustments and the scheduling cost totally comes from the long-horizon stage according to Theorem 3. Furthermore, we fix $A_2 = 200$ and $B = 200$. As we cannot find other distributed long-horizon policies for comparison in existing works, the performance of the initial order is adopted as a baseline.

The average relative scheduling cost (6) for different $A_1$, $W$ and $\lambda^{\text{coop}}$ is shown in Fig. 6. The following observations can be made from Fig. 3.

![Fig. 5. The position curve of several vehicles in the short-horizon stage in an experiment. Cooperative vehicles are represented by solid curves and non-cooperative vehicles by dotted curves. Vehicles on different roads are shown in different colors. The position range of the intersection area [0,5] is colored red.](image)

- First, compared with the initial order, the proposed long-horizon policy makes little difference when $\lambda^{\text{coop}}$ is low (e.g., $\lambda^{\text{coop}} = 0.6$), while it substantially improves the efficiency when $\lambda^{\text{coop}}$ is high (e.g., $\lambda^{\text{coop}} = 1$). Intuitively, the benefit results from gathering the cooperative vehicles from the same road that are initially close to each other and letting them continuously pass the intersection area. In dense traffic, vehicles are closer and thus optimizing the passing order has a larger potential.

- Second, for a large $\lambda^{\text{coop}}$, a better performance can be achieved by enlarging $W$ and $A_1$. Specifically, $W$ influences the achievable efficiency after convergence, while $A_1$ determines the progress of the convergence process. Note that $A_1$ is often a predetermined parameter in practice, and $W$ can be freely chosen. Therefore, a small $W$ is often sufficient when $A_1$ is small, while for a larger $A_1$, $W$ should also be chosen larger.

Furthermore, to evaluate the calculation speed of the proposed long-horizon policy, we list the average running time of a single decision in Table II. According to the table, the computation complexity increases with the scale of the order optimization problem $W$ as expected, while it remains at an acceptable level even when we take $W = 9$, which verifies the computational feasibility of the proposed policy.

![Fig. 6. The average relative scheduling cost of cooperative vehicles under different $A_1$, $W$ and $\lambda^{\text{coop}}$. For each data point, 1000 simulations are performed, and the time span of input vehicles is 500 in each simulation.](image)

| Running Time (ms) | $W = 3$ | $W = 6$ | $W = 9$ |
|-------------------|---------|---------|---------|
| $\lambda^{\text{coop}}$ = 0.6 | 0.67 | 0.76 | 1.41 |
| $\lambda^{\text{coop}}$ = 0.8 | 0.68 | 0.79 | 1.46 |
| $\lambda^{\text{coop}}$ = 1 | 0.70 | 0.80 | 1.52 |

The simulations are performed on Intel Xeon E5-2697 v4, 2.30GHz. We fix $A_1 = 600$ for the table.

12 Recall Section IV-A that $W$ is the scale of the vehicle set used in order optimization.

13 Note that the time consumed by the information collection step (i.e., Algorithm 1), which is dominated by the communication delay, is not included in the running time; only the set truncation and the order optimization steps are included.
Consider the optimization problem \( \Lambda^{\text{coop}} \), all cooperative vehicles will maintain the maximum acceleration as long as there is no risk of collision.

To ensure the fairness of comparison, the proposed long-horizon policy is adopted in all experiments, and thus different short-horizon policies have the same input. Furthermore, we set \( W = 6, A_1 = 600, A_2 = 200 \) and \( B = 200 \) in simulations.

Fig. 7 provides the average relative scheduling cost (6) for different \( \Lambda^{\text{coop}} \) and \( \Lambda^{\text{non\text{-}coop}} \) under the two policies. Note that for each curve, the average relative scheduling cost grows slowly when \( \Lambda^{\text{coop}} \) is low, and increases quickly when \( \Lambda^{\text{coop}} \) is close to the maximum throughput of the system. Then, we have the following observations.

- When there do not exist non-cooperative vehicles (i.e., \( \Lambda^{\text{non\text{-}coop}} = 0 \)), all cooperative vehicles will maintain the maximum velocity under both the proposed policy (by Theorem 3) and the greedy policy, and thus the corresponding two curves coincide. However, the minimax policy performs worse as it is overcautious about the worst-case situation that never occurs.

- When there exist some non-cooperative vehicles (i.e., \( \Lambda^{\text{non\text{-}coop}} = 0.2 \)), the performance of the greedy policy degrades significantly since the cooperative vehicles often have to stop near the intersection, and finding an opportunity to pass the intersection becomes even harder in this case due to the slow start. The minimax policy successfully avoids some worst cases and achieves a better performance than the greedy policy, while the proposed policy, by further utilizing the cooperation relationship, performs best.

In summary, the proposed policy has the best performance among the three policies in both scenarios.

### C. Effectiveness of Short-Horizon Trajectory Planning

In this subsection, we validate the effectiveness of the proposed short-horizon trajectory planning policy, i.e., Algorithm 2. Two other short-horizon policies are adopted for comparison.

- The minimax policy in [38], where the cooperative vehicles regard all other vehicles as non-cooperative ones and optimize the worst-case performance.

- A greedy policy with a safety supervisor [37], where the cooperative vehicles always maintain the maximum acceleration as long as there is no risk of collision.

To ensure the fairness of comparison, the proposed long-horizon policy is adopted in all experiments, and thus different short-horizon policies have the same input. Furthermore, we set \( W = 6, A_1 = 600, A_2 = 200 \) and \( B = 200 \) in simulations.

Fig. 7 provides the average relative scheduling cost (6) for different \( \Lambda^{\text{coop}} \) and \( \Lambda^{\text{non\text{-}coop}} \) under the two policies. Note that for each curve, the average relative scheduling cost grows slowly when \( \Lambda^{\text{coop}} \) is low, and increases quickly when \( \Lambda^{\text{coop}} \) is close to the maximum throughput of the system. Then, we have the following observations.

- When there do not exist non-cooperative vehicles (i.e., \( \Lambda^{\text{non\text{-}coop}} = 0 \)), all cooperative vehicles will maintain the maximum velocity under both the proposed policy (by Theorem 3) and the greedy policy, and thus the corresponding two curves coincide. However, the minimax policy performs worse as it is overcautious about the worst-case situation that never occurs.

- When there exist some non-cooperative vehicles (i.e., \( \Lambda^{\text{non\text{-}coop}} = 0.2 \)), the performance of the greedy policy degrades significantly since the cooperative vehicles often have to stop near the intersection, and finding an opportunity to pass the intersection becomes even harder in this case due to the slow start. The minimax policy successfully avoids some worst cases and achieves a better performance than the greedy policy, while the proposed policy, by further utilizing the cooperation relationship, performs best.

In summary, the proposed policy has the best performance among the three policies in both scenarios.

### VII. CONCLUSION

This paper established a distributed two-stage decision framework for the cooperative vehicles passing an intersection, where both cooperative and non-cooperative vehicles exist. Specifically, the long-horizon stage aims to optimize the passing order of the cooperative vehicles by self-organization, where each cooperative vehicle independently formulates optimization problems and adjusts the trajectory; the short-horizon stage aims to generate safe and efficient trajectories when cooperative vehicles approach the intersection area, where a tradeoff is obtained to balance the robustness against non-cooperative vehicles and the benefit of cooperation. Simulation results showed that both the long-horizon and the short-horizon policies significantly improve the efficiency of cooperative vehicles during the scheduling.

### APPENDIX

#### A. Proof of Theorem 1

We first introduce the following lemma.

**Lemma 1:** Consider the optimization problem (11). Assume that

\[
P_i^k(s_{j_1}) - P_i^k(s_{j_2}) \geq l + \delta,
\]

\[
\forall s_{j_1}, s_{j_2} \in S_i^k \cap S_{r'} \setminus \{s_i\} \text{ with } p_{j_1}(0) > p_{j_2}(0), \forall r' \quad (20)
\]

\[
P_i^k(s_{j_1}) - P_i^k(s_{j_2}) \geq l + \delta + d_{r_i} \]

or \( P_i^k(s_{j_1}) - P_i^k(s_{j_2}) \geq l + \delta + d_{r_i} \),

\[
\forall s_{j_1} \in S_i^k \cap S_{r'} \setminus \{s_i\}, s_{j_2} \in S_i^k \cap S_{r'} \setminus \{s_i\}, r_i' \neq r_1'.
\]

\[
(21)
\]

Let \( s_m \in S_{r'} \) be the vehicle in \( S_i^k \setminus \{s_i\} \) with the smallest \( P_i^k \), and assume that \( P_i^k(s_m) - d_{r_i}/2 = P_i^k(s_i) - d_{r_i}/2 \). Then the solution of (11) satisfies

\[
p_i^* = \{P_i^k(s_m) - (l + \delta + d_{r_i}, r_{i'}, r_i) \mid P_i^k(s_i)\}.
\]

\[
(22)
\]
Proof: Note that the solution \( \{ p_j^* \} \) of (11) also gives an order for all vehicles in \( S^k_i \). Based on this order, we divide \( S^k_i \) into the following three parts:
- the ego-vehicle \( s_i \);
- a subset \( S_{\text{later}} \) containing the vehicles that pass the intersection after \( s_i \);
- a subset \( S_{\text{earlier}} \) containing the vehicles that pass the intersection before \( s_i \).

We only need to prove that \( S_{\text{later}} = \emptyset \).

Now we prove by contradiction. If \( S_{\text{later}} \) is non-empty, we first note that vehicles in \( S_{\text{later}} \) must be on different roads from \( s_i \), since the initial order of the vehicles from the same road is preserved. In the following, we will define another \( \{ p'_j^* \} \) and show that it has a better performance than \( \{ p_j^* \} \).

First, we consider the case that \( s_m \in S_{\text{earlier}} \). We can define \( p'_j = P^k_i(s_j) \) for any \( s_j \in S_{\text{later}} \) and define \( p'_j = p^*_j \) for other \( s_j \). We can easily check that \( \{ p'_j \} \) satisfies all the constraints and results in a smaller objective function.

Then, we consider the case that \( s_m \in S_{\text{later}} \), and thus \( s_m \) is on a different road from \( s_i \). Now we define \( \{ p'_j^* \} \) as follows:
- For \( s_j \), \( p'_j^* \) is defined by the right-hand side of (22).
- For vehicles \( s_j \in S_{\text{later}} \), take \( p'_j = p^*_j \).

It is easy to check that \( \{ p'_j^* \} \) satisfies the constraints. Then we analyze the objective function.
- If \( P^k_i(s_m) - (l + \delta + d_r) \geq P^k_i(s_i) \), then we have \( p'_m = P^k_i(s_i) \), \( p'_m = P^k_i(s_j) \leq p^*_j \). Thus, \( p'_m \leq p^*_j \) holds for all \( s_j \), and the inequality strictly holds for the vehicles in \( S_{\text{later}} \). Therefore, the objective function corresponding to \( p'_m \) is smaller.
- If \( P^k_i(s_m) - (l + \delta + d_r) < P^k_i(s_i) \), then
  \[
  \hat{p}_i^* - p^*_i \geq \hat{p}_m^* - P^k_i(s_i) = \left[ P^k_i(s_m) - (l + \delta + d_r) - P^k_i(s_i) \right] \geq \left[ \frac{(d_r - d_r) + (d_r - d_r)}{2} \right] = 0 
  \] (23)

The last inequalities in (23) and (24) come from the truncation step when defining the set \( S^k_i \) and the assumption \( P^k_i(s_m) - d_r / 2 \neq P^k_i(s_i) - d_r / 2 \) in the lemma. Thus, the total cost of \( s_i \) and \( s_m \) decreases. Noting that the cost of other vehicles is non-increasing, we conclude that the objective function corresponding to \( \hat{p}_m^* \) is smaller.

The discussions above illustrate the contradiction. Therefore, we have \( S_{\text{later}} = \emptyset \).

Now we provide the proof of Theorem 1.

Proof: Assume that there are \( N^{\text{coop}} \) cooperative vehicles in the system. For any \( 1 \leq n \leq N^{\text{coop}} \) and any decision time \( t_k' \), define \( W_k,n \) to be the subset of \( S^{\text{coop}} \) which contains the \( n \) vehicles \( s_i \) with the largest \( \hat{p}_i(t_k') = p_i(t_k') - d_r / 2 \), where \( r \) is the road index corresponding to \( s_i \). According to the linear independence assumption in the theorem, the values of \( \hat{p}_i(t_k') \) for different \( s_i \in S^{\text{coop}} \) are unequal.

We will prove the theorem by induction. Specifically, we will show that for any \( 1 \leq n \leq N^{\text{coop}} \), there exists some \( k_n \geq 0 \) such that
- the set \( W_{k,n} \) is the same for any \( k \geq k_n \);
- at time \( t_k' \), we have (i) any two cooperative vehicles \( s_j \in W_{k,n} \) and \( s_j \in W_{k,n} \) are on different roads from \( n = p \), we can choose a subset \( \{ s_i \} \) containing the vehicles that pass the intersection \( k \) such that \( s_i \in S^{\text{coop}} \) and \( s_j \in S^{\text{coop}} \) satisfy
  \[
  p_j(t_k') - p_j(t_k') < l + \delta 
  \]
  or \( p_j(t_k') - p_j(t_k') < l + \delta 
  \]
  (25)

and (ii) all vehicles in \( W_{k,n} \) maintain the maximum velocity \( v_M \) after time \( t_k' \).

For the case \( n = 1 \), we can choose \( k_1 = 0 \). This is because for the vehicle \( s_i \) with the largest \( \hat{p}_i(t_k') \), the corresponding vehicle set \( S^0_i = \{ s_j \} \) holds. Therefore, it will maintain the maximum velocity \( v_M \) during \( t_k' \). We can inductively check that it will always be the vehicle with the largest \( \hat{p}_i(t_k') \), and hence \( W_{k,n} \) remains constant for \( k \geq 0 \).

Now we consider the general case with \( 2 \leq n \leq N^{\text{coop}} \) by assuming that the induction for \( n - 1 \) has completed. First we introduce some definitions. Let \( s_m \in S_{\text{earlier}} \) be the vehicle in \( W_{k,n-1} \) with the smallest \( \hat{p}_m(t_{k-1}) \). On each road, we take a vehicle which is outside the set \( W_{k,n-1} \) and has the largest initial position, and we collect these vehicles and form a set \( A \).

For any vehicle \( s_i \in A \), define
  \[
  G_i(t_k') = p_m(t_k') - p_i(t_k') - (l + \delta + d_r, 1_{r \neq r_i}) \] (26)

where \( r \) is the road index corresponding to \( s_i \).

For a given \( k \), define the vehicle
  \[
  s_{\text{in}(k)}(t_{k'}) = \arg \max_{s_i \in A} \hat{p}_i(t_{k'}) \] (27)

We aim to show that there exists a large enough \( k' \) such that
  \[
  G_{\text{in}(k')}(t_{k'}) \geq 0 \] (28)

Actually, if (28) holds, then we can complete the induction through the following steps.

(i) According to (27), we have \( S_{\text{in}(k')}(t_{k'}) \subseteq W_{k,n-1} \cup \{ s_{\text{in}(k')} \} \). Thus, we can check that the corresponding optimization problem (11) satisfies the condition of Lemma 1.

(ii) According to Lemma 1, the vehicle \( s_{\text{in}(k')} \) will maintain the maximum velocity \( v_M \) within the time period \( t_{k'}', t_{k'+1}' \). Thus, we have \( s_{\text{in}(k')} = s_{\text{in}(k')} \).

(iii) By repeating the steps (i) and (ii), we can conclude that \( s_{\text{in}(k')} \) will always maintain the velocity \( v_M \) after time \( t_{k'}' \), and \( s_{\text{in}(k')} \) holds for all \( k \geq k' \).

(iv) Since the initial order is preserved for the vehicles from the same road, there exists \( k_n \geq k' \) such that \( W_{k,n} = W_{k,n-1} \cup \{ s_{\text{in}(k')} \} \). Then we can easily check that the induction has been completed.

Finally, we show that (28) holds for some \( k' \). The following two results can firstly be obtained.
For any \( s_i \in \mathcal{A} \), \( G_i(t'_k) \) is non-decreasing with respect to \( t'_k \).

For each \( k \geq k_{n-1} \), if \( G_{h_i}(t'_k) < 0 \), then either \( G_{h_i}(t'_k + 1) = 0 \), or \( G_{h_i}(t'_k + 1) - G_{h_i}(t'_k) \) is not smaller than a positive constant.

Specifically, the first result can be directly checked by the definition (26) and the induction hypothesis; to obtain the second result, we should note that the optimization problem corresponding to \( s_{i}(t_{k_{n-1}}) \) in the time interval \([t'_{k_{n-1}}, t'_k)\) satisfies the conditions in Lemma 1, and then utilize the result of Lemma 1 and (12).

Note that for any \( k \), \( G_{h_i}(t'_k) \) must be in a finite set \( \{ G_i(t'_k) \mid s_i \in \mathcal{A} \} \). According to the two results above, there cannot exist an infinite number of \( k \) such that \( G_{h_i}(t'_k) < 0 \) holds. Therefore, \( k'_n \) must exist.

**B. Definition of the Function \( V(p, v; t, \mathcal{K}) \)**

In this appendix, we provide a brief definition of the objective function \( V(p, v; t, \mathcal{K}) \) in (16) for the completeness of the paper. For the detailed definition, please refer to [38].

First, we fix a current time \( t_{cur} \). Let \( \mathcal{K} \) be the knowledge of the non-cooperative vehicles obtained by the ego-vehicle \( s_i \) at time \( t_{cur} \), and let \( \mathcal{W}(\mathcal{K}) \subseteq [t_{cur}, +\infty) \) be the set of future time when the intersection area can be occupied by any non-cooperative vehicle based on the knowledge \( \mathcal{K} \). Then we define the *manageable cost* of the ego-vehicle \( s_i \) at time \( t_{cur} \) by

\[
C^*(\sigma; t_{cur}, \mathcal{K}) = \begin{cases} 
+\infty, & \text{if } (t_{i,\text{in}}, t_{i,\text{out}}) \cap \mathcal{W}(\mathcal{K}) \neq \emptyset; \\
v_{M} \cdot \delta \left( [t_{cur}, t_{i,\text{in}}] \setminus \mathcal{W}(\mathcal{K}) \right) + \frac{(v_{M} - v_{i})(t_{i,\text{out}})}{2a_{M}}, & \text{otherwise}
\end{cases}
\]  

(29)

where \( \delta \) represents the Lebesgue measure; \( \sigma \) represents the trajectory of \( s_i \); \( t_{i,\text{in}} \) and \( t_{i,\text{out}} \) are functions of \( \sigma \) representing the times when \( s_i \) enters and leaves the intersection, respectively.

Note that \( C^* \) is closely related to the relative scheduling cost. Specifically, when \( C^* \) takes finite value, we have

\[
C^*(\sigma; t_{cur}, \mathcal{K}) = C_t(t_{i,\text{in}}) - [p_i(t_0) + v_{M}t_{cur}] + v_{M} \cdot \delta \left( [t_{cur}, t_{i,\text{in}}] \cap \mathcal{W}(\mathcal{K}) \right).
\]  

(30)

In this equation, the two former difference terms \( p_i(t_0) \) and \( v_{M}t_{cur} \) are constant with respect to the trajectory. The last term \( v_{M} \cdot \delta \left( [t_{cur}, t_{i,\text{in}}] \cap \mathcal{W}(\mathcal{K}) \right) \) represents the length of a time interval when the intersection area can be occupied by non-cooperative vehicles; the ego-vehicle \( s_i \) should not occupy the intersection area in this time interval due to the safety condition, and thus the corresponding loss should not be considered in the decision.

Now, we define the state value function \( V(p, v; t_{cur}, \mathcal{K}) \) to be the best manageable cost that can be obtained by trajectories starting from the state \((p, v)\), i.e.,

\[
V(p, v; t_{cur}, \mathcal{K}) = \min_{\sigma \in \Sigma_t(p, v; t_{cur})} C^*(\sigma; t_{cur}, \mathcal{K})
\]  

(31)

where the set \( \Sigma_t(p, v; t_{cur}) \) includes all possible trajectories of \( s_i \) starting from the state \((p, v)\) at time \( t_{cur} \). Therefore, the function \( V \) is also directly related to the relative scheduling cost.

**C. Proof of Theorem 2**

Proof: According to Algorithm 2, if a cooperative vehicle \( s_1 \) has not entered the region \( \mathcal{C} \) at time \( t_{s_1} \), then there exist three possibilities which correspond to Lines 6–7 (Case (a)), Lines 8–9 (Case (b)) and Lines 10–12 (Case (c)) in the algorithm, respectively. Note that Case (a) will only occur at the last decision time before \( s_1 \) enters the region \( \mathcal{C} \), and we denote this decision time as \( t_{i,k_0} \).

According to the definition of \( \sigma^* \) in Line 4, the output trajectory of \( s_i \) after \( t_{i,k_0} \) must be conflict-free with all non-cooperative vehicles. In the following, we will show that \( s_i \) is conflict-free with \( s_j \), which is the last cooperative vehicle in front of \( s_i \), based on the passing order. Specifically, by defining

\[
L_j = l + d_r \cdot \mathds{1}_{r \neq r'}
\]  

(32)

where \( r \) and \( r' \) are the road index of \( s_i \) and \( s_j \), respectively, we will prove that

\[
\{ t \mid p_i(t) \in (x, +\infty), p_j(t) \in (x, x + L_j) \} = \emptyset
\]  

(33)

holds for any \( x > 0 \). We can easily check that (33) is a sufficient condition for the safety condition (2) or (3). Now we separately consider the following two situations.

(1) First, we assume that Case (c) never occurs in the decision of \( s_i \). In other words, before Case (a) occurs, all decision times correspond to Case (b). Then \( s_j \) always follows the minimum acceleration before time \( t_{i,k_0} \). Now we consider the decision of \( s_0 \) at \( t_{i,k_0} \). Based on the “if”-sentence in Line 6, \( p_j(t_{obs}) + v_{i}(t_{i,k_0}) \geq 0 \) must hold. Since \( t_{i,k_0} \geq t_{obs} \), we know that \( s_j \) has entered the region \( \mathcal{C} \) at \( t_{i,k_0} \), and thus it will always follow the maximum acceleration after \( t_{i,k_0} \). Therefore, for any \( t \geq t_{i,k_0} \), we have

\[
p_j(t) - p_i(t) \geq \left( p_j(t_{i,k_0}) - \frac{v_{M}^2}{2a_{M}} + v_{M}(t - t_{i,k_0}) \right) - \left( p_i(t_{i,k_0}) + v_{M}(t - t_{i,k_0}) \right)
\geq p_j(t_{obs}) - p_i(t_{i,k_0}) - \frac{v_{M}^2}{2a_{M}}
\geq -\frac{v_{M}^2}{2a_{M}} - \left( -B + \frac{v_{M}^2}{2a_{M}} \right) - \frac{v_{M}^2}{2a_{M}}
\geq l + \max \{ d_r \mid 1 \leq r' \leq R \} \geq L_j,
\]  

(34)

Hence, (33) holds.

(2) Second, we assume that Case (c) occurs in the decision of \( s_i \), and let \( t_{i,k_0} \) be the decision time when Case (c) first occurs. Now we prove by induction that any decision time \( t_{k,k} \) with \( k_1 < k < k_0 \) also corresponds to Case (c), i.e., the set \( \mathcal{F}_{k_{i,k},k}^{\text{oc}} \) is non-empty. If the set \( \mathcal{F}_{k_{i,k},k}^{\text{oc}} \) is non-empty, then \( p_i(t_{k,k+1}) \) and \( v_i(t_{k,k+1}) \) must belong to it by Lines 11–12. Then we consider the trajectory \( \sigma_D \) starting from the state \( (p_i(t_{k,k+1}), v_i(t_{k,k+1})) \) and following the minimum acceleration. We can easily check that under this trajectory, \( (p_i(t_{k,k+1}), v_i(t_{k,k+1})) \in \mathcal{F}_{k_{i,k},k}^{\text{oc}} \). Thus we complete the induction.

Therefore, the decision time \( t_{k,k+1} \) also corresponds to Case (c). Since \( t_{k,k+1} \) corresponds to Case (a), \( s_j \) must have entered the region \( \mathcal{C} \) before \( t_{k,k+1} \) and will always follow the maximum
acceleration after \( t_{i,k_0} \). Let \( \sigma_A \) be the trajectory of \( s_i \) starting from the state \( \langle p_i(t_{i,k_0}), v_i(t_{i,k_0}) \rangle \) and following the maximum acceleration. Then, noting that \( \langle p_i(t_{i,k_0}), v_i(t_{i,k_0}) \rangle \in F_{i,k_0}^{\text{fol}} \) and according to the definition of \( F_{i,k_0}^{\text{fol}} \) in Line 5, we must have \( p_i(t) \leq p_i(t) - L_j \) for any \( t \geq t_{i,k_0} \) if \( s_i \) follows the trajectory \( \sigma_A \). Thus, \( p_i(t) \leq p_i(t) - L_j \) (\( \forall t \geq t_{i,k_0} \)) also holds for any trajectory of \( s_i \) after \( t_{i,k_0} \). In other words, (33) holds. \( \square \)

\section{D. Proof of Theorem 3}

Proof: We perform induction on the time. Assume that the outputs of all decisions of the cooperative vehicles before time \( t \) are the trajectories with the maximum velocity. Then we consider the first short-horizon decision made by the vehicles after \( t \), for which the corresponding vehicle is denoted by \( s_i \) and the decision time is \( t_{i,k} \). According to the induction hypothesis, all vehicles will follow the maximum velocity \( v_M \) before \( t_{i,k} \).

If \( s_i \) is the first vehicle based on the passing order, then it will always follow the maximum velocity, and the induction is completed.

In the following, we assume that \( s_i \) is not the first vehicle, and denote \( s_j \) to be the last cooperative vehicle in front of \( s_i \) based on the passing order. Assume that for \( s_i \), the last state observation on \( s_j \) is \( \langle p_j(t_{obs}), v_j(t_{obs}) \rangle \), and we must have \( t_{obs} \geq t_{i,k} - \tau \). Now we focus on Algorithm 2. Since there are no non-cooperative vehicles in the system, the trajectory \( \sigma^* \) defined in Line 4 is actually the trajectory with the maximum acceleration.

Therefore, we have

\[
 p^* + \frac{v^*^2}{2a_m} = \left[ p_j(t_{obs}) + v_M(t_{i,k+1} - t_{i,k}) \right] + \frac{v_M^2}{2a_m} 
 \leq \left[ p_j(t_{obs}) - L_j - \delta^* \right] + v_M(t_{i,k+1} - t_{i,k}) + \frac{v_M^2}{2a_m} 
 = \left[ p_j(t_{obs}) + v_M(t_{i,k} - t_{obs}) \right] - L_j - \delta^* 
 + v_M(t_{i,k+1} - t_{i,k}) + \frac{v_j(t_{obs})^2}{2a_m} 
 \leq \left[ p_j(t_{obs}) + \frac{v_j(t_{obs})^2}{2a_m} \right] - (L_j + \delta^* - v_M\tau - v_M\mu) 
 \leq p_j(t_{obs}) + \frac{v_j(t_{obs})^2}{2a_m} \tag{35}
\]

where \( \langle p^*, v^* \rangle \) is defined in Line 4 and \( L_j \) is identically defined as in (32).

We first consider the case \( p^* + \frac{v^*^2}{2a_m} > 0 \). According to (35), we also have \( p_j(t_{obs}) \geq \frac{v_j(t_{obs})^2}{2a_m} > 0 \). By Line 6, \( s_i \) will return the trajectory \( \sigma^* \) and the induction is completed.

Then, we consider the case \( p^* + \frac{v^*^2}{2a_m} \leq 0 \). To start, we prove that \( F_{i,k}^{\text{fol}} \neq \emptyset \) by showing \( \langle p^*, v^* \rangle \in F_{i,k}^{\text{fol}} \). We arbitrarily fix \( t_{D} \geq t_{i,k+1} \), fix the state \( \langle p^*, v^* \rangle \) of \( s_i \) at \( t_{i,k+1} \), and fix the state \( \langle p_j(t_{obs}), v_j(t_{obs}) \rangle \) of \( s_j \) at \( t_{obs} \). Then we assume that both \( s_i \) and \( s_j \) adopt the minimum acceleration before \( t_{D} \) and adopt the maximum acceleration after \( t_{D} \). Under this setting, we can check that the velocity difference between \( s_i \) and \( s_j \) is always not larger than \( a_m(t_{i,k+1} - t_{obs}) \leq a_m(\tau + \mu) \). Therefore, the difference between their displacements before time \( t_{D} \) is not larger than \( a_m(\tau + \mu) \cdot \frac{t_{D}}{a_m} \), and the difference between their displacements after time \( t_{D} \) is not larger than \( a_m(\tau + \mu) \cdot \frac{t_{D}}{a_m} \). According to (19), \( p_i(t) \leq p_i(t) - L_j \) always holds. Therefore, according to the definition of \( F_{i,k}^{\text{fol}} \) in Line 5, we have \( \langle p^*, v^* \rangle \in F_{i,k}^{\text{fol}} \).

Since \( F_{i,k}^{\text{fol}} \) is non-empty, the algorithm goes to Lines 11–12, and the minimax optimization problem will be solved. However, since there are no non-cooperative vehicles, by noting that \( \langle p^*, v^* \rangle \in F_{i,k}^{\text{fol}} \), we can easily check that \( \langle p^*, v^* \rangle \) will be the result of the optimization and the policy will return the trajectory with the maximum acceleration. So far, we have completed the induction in all scenarios. \( \square \)

\section*{REFERENCES}

[1] D. Schrank et al., \textit{Urban Mobility Report}. Texas, TX, USA: Texas A&M Transp. Inst., 2021.

[2] L. Chen and C. Englund, “Cooperative intersection management: A survey,” \textit{IEEE Trans. Intell. Transp. Syst.}, vol. 17, no. 2, pp. 570–586, Feb. 2016.

[3] 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,” \textit{IEEE Trans. Intell. Transp. Syst.}, vol. 18, no. 5, pp. 1066–1077, May 2017.

[4] A. Gholamhosseinian and J. Seitz, “A comprehensive survey on cooperative intersection management for heterogeneous connected vehicles,” \textit{IEEE Access}, vol. 10, pp. 7937–7972, 2022.

[5] A. G. Simos and K. W. Dobinson, “The Sydney coordinated adaptive traffic (SCAT) system philosophy and benefits,” \textit{IEEE Trans. Veh. Technol.}, vol. 29, no. 2, pp. 130–137, May 1980.

[6] D. I. Robertson and R. D. Bretherton, “Optimizing networks of traffic signals in real time—the SCOOT method,” \textit{IEEE Trans. Veh. Technol.}, vol. 40, no. 1, pp. 11–15, Feb. 1991.

[7] W. Brilon and T. Wietholt, “Experiences with adaptive signal control in Germany,” \textit{Transport. Res. Rec.}, vol. 2366, no. 1, pp. 9–16, Jan. 2013.

[8] C. P. Pappis and E. H. Mandanis, “A fuzzy logic controller for a traffic junction,” \textit{IEEE Trans. Syst. Man, Cybern.}, vol. 7, no. 10, pp. 707–717, Oct. 1977.

[9] E. Bingham, “Reinforcement learning in neurofuzzy traffic signal control,” \textit{Eur. J. Oper. Res.}, vol. 131, no. 2, pp. 232–241, Jun. 2001.

[10] J. Wu, D. Ghosal, M. Zhang, and C.-N. Chuah, “Delay-based traffic signal control for throughput optimality and fairness at an isolated intersection,” \textit{IEEE Trans. Veh. Technol.}, vol. 67, no. 2, pp. 896–909, Feb. 2018.

[11] Y. Feng et al., “A real-time adaptive signal control in a connected vehicle environment,” \textit{Transp. Res. Proc. C: Emerg. Technol.}, vol. 55, pp. 460–473, Jun. 2015.

[12] J. Zhao, W. Li, J. Wang, and X. Ban, “Dynamic traffic signal timing optimization strategy incorporating various vehicle fuel consumption characteristics,” \textit{IEEE Trans. Veh. Technol.}, vol. 65, no. 6, pp. 3874–3887, Jun. 2016.

[13] S. Li, C. Wei, X. Yan, L. Ma, D. Chen, and Y. Wang, “A deep adaptive traffic signal controller with long-term planning horizon and spatial-temporal state definition under dynamic traffic fluctuations,” \textit{IEEE Access}, vol. 8, pp. 37087–37104, 2020.

[14] G. Karagiannis et al., “Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions,” \textit{IEEE Commun. Surveys Tuts.}, vol. 13, no. 4, pp. 584–616, Oct.–Dec. 2011.

[15] P. Papadimitratos, A. D. La Fortelle, K. Enevssis, R. Brignolo, and S. Cosenza, “Vehicular communication systems: Enabling technologies, applications and future outlook on intelligent transportation,” \textit{IEEE Commun. Mag.}, vol. 47, no. 11, pp. 84–95, Nov. 2009.

[16] K. Dresner and P. Stone, “A multiagent approach to autonomous intersection management,” \textit{J. Artif. Intell. Res.}, vol. 31, pp. 591–656, Mar. 2008.

[17] Y. Zhang, L. Liu, Z. Lu, L. Wang, and X. Wen, “Robust autonomous intersection control approach for connected autonomous vehicles,” \textit{IEEE Access}, vol. 8, pp. 124486–124502, 2020.
[18] M. Vasirani and S. Ossowski, “A market-inspired approach for intersection management in urban road traffic networks,” J. Artif. Intell. Res., vol. 43, no. 1, pp. 621–659, Jan. 2012.

[19] 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.

[20] J. Wang, X. Zhao, and G. Yin, “Multi-objective optimal cooperative driving for connected and automated vehicles at non-signalised intersection,” IET Intell. Transp. Syst., vol. 13, no. 1, pp. 79–89, Oct. 2019.

[21] D. Miculescu and S. Karaman, “Polling-systems-based autonomous vehicle coordination in traffic intersections with no traffic signals,” IEEE Trans. Autom. Control, vol. 65, no. 2, pp. 680–694, Feb. 2020.

[22] A. I. Morales Medina, F. Creemers, 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.

[23] W. Zhao, R. Liu, and D. Ngodu, “A bilevel programming model for autonomous intersection control and trajectory planning,” Transportmetrica A: Transp. Sci., vol. 17, no. 1, pp. 34–58, Jan. 2021.

[24] S. D. Kumaravel, A. Malikopoulos, and R. Ayyagari, “Optimal coordination of platoons of connected and automated vehicles at signal-free intersections,” IEEE Trans. Intell. Veh., vol. 7, no. 2, pp. 186–197, Jun. 2022.

[25] G. R. de Campos, P. Falcone, and J. Sjöberg, “Autonomous cooperative driving: A velocity-based negotiation approach for intersection crossing,” in Proc. IEEE Int. Conf. Intell. Transp. Syst., 2013, vol. 1, pp. 1456–1461.

[26] W. Wu et al., “Distributed mutual exclusion algorithms for intersection traffic control,” IEEE Trans. Parallel Distrib. Syst., vol. 26, no. 1, pp. 65–74, Jan. 2015.

[27] A. P. Capasso Dell’ et al., “End-to-end intersection handling using multi-agent deep reinforcement learning,” in Proc. IEEE Intell. Veh. Symp., 2021, pp. 443–450.

[28] J. Alonso et al., “Autonomous vehicle control systems for safe crossroads,” Transp. Res. Part C: Emerg. Technol., vol. 19, no. 6, pp. 1095–1110, Dec. 2011.

[29] B. Yang and C. Monterola, “Efficient intersection control for minimally guided vehicles: A self-organised and decentralised approach,” Transp. Res. Part C: Emerg. Technol., vol. 72, pp. 283–305, Nov. 2016.

[30] 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 Int. Conf. Intell. Transp. Syst., 2019, pp. 2299–2304.

[31] 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.

[32] 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.

[33] 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., 2019, pp. 2449–2455.

[34] 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. Veh. Symp., 2020, pp. 1583–1588.

[35] V. Milanes, J. Perez, E. Onieva, and C. Gonzalez, “Controller for urban intersections based on wireless communications and fuzzy logic,” IEEE Trans. Intell. Transp. Syst., vol. 11, no. 1, pp. 243–248, Mar. 2010.

[36] 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, Dec. 2011.

[37] A. Colombo and D. Del Vecchio, “Least restrictive supervisors for intersection collision avoidance: A scheduling approach,” IEEE Trans. Autom. Control, vol. 60, no. 6, pp. 1515–1527, Jun. 2015.

[38] F. Yang and Y. Shen, “A minimax scheduling framework for inertially-constrained multi-agent systems,” IEEE Trans. Intell. Transp. Syst., vol. 23, no. 12, pp. 24414–24427, Dec. 2022.

[39] F. Yang and Y. Shen, “Distributed long-horizon vehicle scheduling for traffic intersection with delayed information,” in Proc. IEEE Glob. Commun. Conf., 2020, pp. 1–6.

[40] A. I. M. 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.

Feihong Yang received the B.E. degree (with Hon.) in electronic engineering and the Ph.D. degree in information and communication engineering from Tsinghua University, Beijing, China, in 2017 and 2022, respectively. 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, Beijing, China, in 2005, and the S.M. and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 2008 and 2014, respectively.

He is currently a Professor with the Department of Electronic Engineering, Tsinghua University. His research interests include network localization and navigation, integrated sensing and control, and multi-agent systems.

His papers have received the IEEE ComSoc Fred W. Ellersick Prize and several Best Paper awards from IEEE conferences. He was the TPC Symposium Co-Chair for IEEE ICC and IEEE Globecom for several times. He was also the Elected Chair of the IEEE ComSoc Radio Communications Committee from 2019 to 2020. He is the Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE WIRELESS COMMUNICATIONS LETTERS, and China Communications.