A Decentralized Multi-Criteria Optimization Algorithm for Multi-Unmanned Ground Vehicles (MUGVs) Navigation at Signal-Free Intersection

Zhengze Zhu [*,**,****] **Lounis Adouane** [***] **Alain Quilliot** [**]

[*] *Institue Pascal, CNRS 6602, Université Clermont Auvergne, 63178 Aubière Cedex France (e-mail: Zhengze.ZHU@etu.uca.fr)*
[**] *LIMOS, CNRS 6158, Université Clermont Auvergne, 63178 Aubière Cedex France (e-mail: alain.quilliot@isima.fr)*
[***] *Heudiasyc, CNRS 7253, Université de Technologie de Compiègne, 60203 Compiègne Cedex France (e-mail: lounis.adouane@hds.utc.fr)*
[****] *ASID, Lab, Hubei University of Automotive Technology, 442000 China*

Abstract: In intelligent traffic management, one of its core challenges lies in Multiple Unmanned Ground Vehicle (MUGVs) navigation in cluttered and dynamic environments. To be viable, a multi-criteria navigation scheme is required to deal with several critical situations. With its relative low execution time , the Probability Collectives (PC) algorithm has succeeded in generating fast and feasible solutions when applied to manage challenging scenarios, such as in signal-free intersections and round-about (see Philippe et al. (2019)). Indeed, PC is an interesting decentralized approach for general cases that enables to manage complex systems under probabilistic hypotheses. However, the PC is sensitive to uncertainty in the navigation process, which highlights the need to adopt several safety margins. These margins permit vehicles to adapt their dynamics and to react properly to unexpected events. Accordingly, the present work aims to integrate a reliable risk management strategy into the PC algorithm by introducing a novel ε-constraint PC method. With the enhancements integrated into an already existing approach, the ε-PC based navigation strategy is able to obtain a better fusion strategy with considering both efficiency and safety. Accordingly, this work aims to develop an appropriate balance develop a proper balance between the high-quality solution and acceptable computational speed. Further, typical scenarios of unsignalized intersection have been used intensively in simulation setups to demonstrate the efficiency of the proposed approach.

Keywords: Intersection navigation, Probability collectives, Risk assessment and management, ε-constraint PC.

1. INTRODUCTION

Similar to urban transportation systems, specific territories like large hospitals, university campus, industrial sites have taken steps to improve their navigation services in their shipment/transit areas. As indicated in Vilca et al. (2013, 2018), reliable online obstacle avoidance and control are highly required for such Multiple Unmanned Ground Vehicles (MUGVs). However, the inherent trade-off between the control scheme quality and its computational demands is, therefore, a crucial issue that should be explored for this kind of navigation.

Obviously, some roads in few restricted areas consist of private/non-standard and narrow alleyways. This kind of roads may complicate the access to principle buildings. As shown in Hyland and Mahmassani (2018), Unmanned Grounded Vehicle (UGV) in such circumstances can help to provide more efficient transport services for passengers. In the mean time, numerous simultaneous requests from multiple delivery locations may invoke cross-linked planning routes for MUGVs systems. In such a manner, MUGVs navigation may become difficult to cope with at intersection points.

In the field of intersection coordination, traffic signal control system played an important role in solving successfully traffic congestion problems (Ma et al. (2018)). For UGV based intersection management, there are mainly three types of signal control methodologies. For instance, Manzinger and Althoff (2018) proposed a direct vehicle control method that enables vehicles to achieve certain targets. Suzuki and Marumo (2018) introduced an approach for optimizing traffic lights phases to improve the intersection traffic performances. Further, Xu et al. (2017) developed a signal-vehicle coupled control system for a better intersection coordination. These three types of intersection traffic control methods depend on the availability of Vehicle-to-everything (V2X) technology, see Guo et al. (2019). However, traditional traffic signal control methods in urban cities usually can not be applied directly in above mentioned areas, because traffic light is subject to redundant cost in such an inappropriate formed crossing-road and in certain situations increases the level of traffic jam as illustrated in Guo et al. (2019). Chen and Englund (2015) indicated that automation and communication have turned the cooperative intersection management into a more active research field. Roughly, distributed and decentralized control are becoming a promising way to deal efficiently with this multi-scale navigation problem in complex traffic scenarios. Studies reported in Chen and Englund (2015) and Gregoire et al. (2014) provided more details about such cases. Therefore, a non-signal management of vehicles from a shared space is studied in Philippe et al. (2019). A distributed and decentralized optimization algorithm, based on Probability Collectives (PC), is first applied to solve ground vehicle coordination problem.

The PC algorithm is an efficient optimization searching framework for distributed systems, which was first proposed by Wolpert (2006). It is a COllective INtelligence (COIN) framework that emerged from game theory, statistical physics, and optimization theory as indicated in Kulkarni and Tai (2010a). A comparative study in Huang et al. (2005) has showed that the PC-based approach is superior to traditional Genetic Algorithm (GA) in both rate of decent and avoiding local minima. Kulkarni and Tai (2010a) designed a perturbation approach to improve the algorithm performances, which had been tested and verified by benchmark functions. After that, a PC-based approach solved successfully various discrete optimization problem like Multiple Traveling Salesmen Problem (MTSP) and Vehicle Routing Problems (VRP), see Kulkarni and Tai (2010a). In an effort to solve dynamic vehicle coordination problem with low computational time, the authors' previous work addressed a PC-based approach to handle intersection coordination as previously mentioned in Philippe et al. (2019). The PC algorithm in Philippe et al. (2019) has two important qualities, namely probabilistic nature and decentralized nature. Its probabilistic nature allows a probability distribution over a vehicle behavior set, guaranteeing a risk averse decision strategy. It permits also to deal with uncertainty without inducing the deadlock of the shared decision process. Its decentralized nature allows it to be used without a specific infrastructure. Besides, vehicles can significantly benefit from an acceptable computing time (around $0.2s \sim 0.8s$ for the full optimization cycle). Thus, it is an interesting and promising method to process the aforementioned MUGVs navigation problem in restricted areas. To the best of the author's knowledge, this is the first study that handles PC in ground traffic.

In this paper, MUGVs are provided with embedded decisional devices and inter-vehicle communication tools to manage navigation tasks. For a group of homogeneous MUGVs systems, most of the studies consider generally several navigation issues without addressing enough risk-sensitive countermeasures. The MUGVs operators are put forward in an effort to seek efficient and effective controls to cut off with customer waiting time or energy consumption, see Berbeglia et al. (2010). Unfortunately, the evaluation of uncertainties is not considered sufficiently in these processes. Since sudden changes in the dynamics of ground vehicles in a short time are not realistic, Iberraken et al. (2018) and Lakhal et al. (2019) have applied a risk assessment approach to succeed hazard prediction for the navigation traffic. As a matter of fact, risk minimization has been shown to be considered as a priority in intersection coordination as indicated in Chen and Englund (2015). Vehicles collaboration with risk management capabilities is a promising way to solve this problem. Therefore, the proposed methodology in this paper aims to provide a flexible constraint decision-making approach that depends on the safety requirements. The adopted risk management strategy considers both the service quality (e.g., a fast crossing strategy) and safety at an intersection. For this motive, this paper integrates a ε-constraint method into the PC algorithm to add safety indicator constraints. The formulation of ε-constraint searching scheme in the PC algorithm is first addressed in this paper. Its key contribution lies in incorporating a more flexible multi-criteria decision-aids techniques into the collision prediction. Further, it balances between the high quality strategy and acceptable computational speed with the proposed method.

The rest of the paper is organized as follows: a conceptual review of the PC application to intersection coordination is introduced in Section 2. Section 3 presents the ε-constraint method in PC algorithm. The integration of MUGVs system and proposed ε-PC approach is shown in Section 4. A detailed use case is given in Section 5 to validate the addressed method for MUGVs systems. At last, conclusions and some prospects are given in Section 6.

2. A CONCEPTUAL REVIEW OF THE PC APPLICATION TO INTERSECTION COORDINATION

In order that the proposed paper can be simply read, let us sum-up in what follows the already proposed PC formulation to deal with the coordination of MUGVs in signal-free intersections and round-abouts as shown in Philippe et al. (2019).

2.1 Formulation of searching space

Several vehicles are considered crossing through the intersection with fixed known path. Then, the only control degree of freedom left is the speed of navigation. Vehicles should select their actions (velocities in our problem) over a particular predefined interval time to coordinate their navigation motions. An illustration of the possible actions in fixed time windows ($T = 10s$), which is long enough to permit vehicles to leave the intersection, is depicted in Fig. 1.

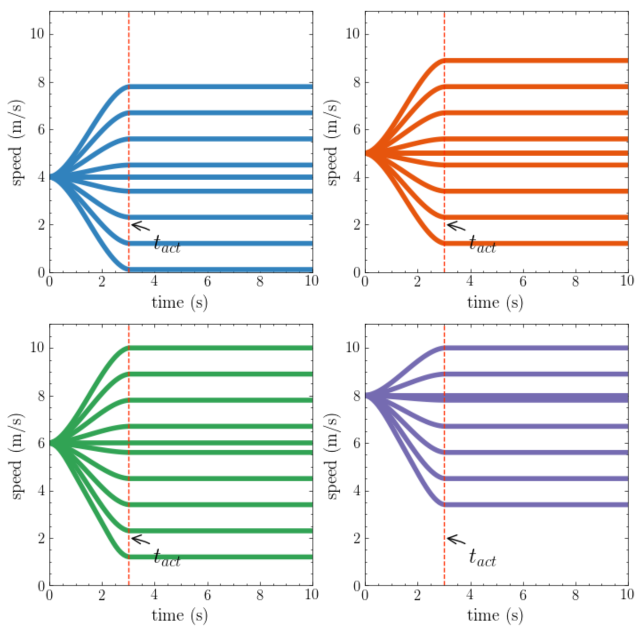

Fig. 1. Example of strategies hypotheses for vehicle actions according to their initial velocities

Apparently, in Fig. 1, there are considerable options N_i for each vehicle i depending on the initial speed $v_i(0)$. A further taken hypothesis is that all the vehicles will get a fixed speed $v_i(T)$ after a predefined action time t_{act} (such as $t_{act} = 3s$ in Fig. 1). At last, the searching space for vehicle i can be summed up as a 3-tuple $\mathbf{\Pi}^i$ i.e., $\mathbf{\Pi}^i \sim \{v_i(T), t_{act}, N_i\}, t \in [0, T]$. Then, the admissible member of actions set for the ego-vehicle can be presented as $\Pi^i \in \boldsymbol{\Pi}^i = \{\Pi^i_1, \ldots, \Pi^i_{N_i}\}$. Here, Π^i can be visualized as the velocity profile in Fig. 1. The generation of smooth speed profiles was inspired by the algorithm in Hult et al. (2015). A solver-based optimization problem was modeled by a quadratic cost function that considers cost on jerk (as input $u \in [u_{min}, u_{max}]$) and cost on the specified reference speeds $(v_i(T), N_i)$. Equality constraints are included to achieve targeted speed after the action time t_{act}.

Vehicles in PC are regarded as individual self-interested players in an iterative cooperation problem. Thus, the expected utility of a given action can be calculated by each vehicle. But to do so, the possible actions of the other vehicles must be got (or estimated). Therefore, the Probability Distribution (PD) have been used to model relative N_i actions for vehicle i like $q(\Pi^i_k) \in q(\boldsymbol{\Pi}^i) = \{q(\Pi^i_1), \ldots, q(\Pi^i_{N_i})\}$. Obviously, the preferred actions (or strategy) should have a high probability of being cost-effective. The driver model used to improve the precision of the predictive control with PD is a very active topic, but not the main research issue in this paper. Readers are recommended to read the review of Di Cairano et al. (2013) and Schwarting et al. (2019) to get clearer idea about the estimation of the PD of other ground vehicles. The hypotheses of prosocial (or altruistic) UGVs in this paper make us formulating an uniform distribution of all the agents' behaviors when $q(\boldsymbol{\Pi}^i)$ is initially loaded for computation.

2.2 Two steps for re-acceleration

For various collaborative navigation behaviors, vehicles choose a speed profile that allows them to safely cross an intersection based on an utility function (cf. Sect. 2.3). However, for vehicles that have to choose the arbitrary low speed (or a complete stop), the proposed algorithm allows them to re-accelerate. The re-acceleration permits the vehicles to clear the intersection as fast as possible while ensuring free collisions. Detailed re-acceleration algorithms have been designed in the previous work, see Philippe et al. (2019).

2.3 Objective function

In its initial formulation, the original PC approach considers only an unconstrained minimization problem. Such a research case generally involves n vehicles, where each vehicle $i \in n$ possesses a strategies/actions set of $\boldsymbol{\Pi}^i = \{\Pi^i_1, \ldots, \Pi^i_N\}(i = 1, \ldots, n)$ including an equal amount of N options (cf. Sect. 2.1). After performing a local motion planning through its on-board embedded devices, each vehicle applies a strategy $\Pi^i_k \in \boldsymbol{\Pi}^i(k = 1, \ldots, N)$ during time interval $[0, T]$. Here, T refers to the prediction time horizon. During the period $[0, T]$, a particular set of combined strategies $\boldsymbol{Y} = [\Pi^1_k, \Pi^2_k, \ldots, \Pi^n_k]$ is selected (randomly fixed to initialise the process) to reach at least a minimum system utility level $J([\Pi^1_k, \Pi^2_k, \ldots, \Pi^n_k])$. The proposed objective function (by Philippe et al. (2019)) can be formulated as given by equation (1):

$$J(\boldsymbol{Y}) = W_{sep} \sum_{i_v \neq i} \sum_{t_k=1}^{max} \frac{1}{d_k(i_v, i_{self})^2} + W_{cross}(v_{max} - v_{avg})^2 \quad (1)$$

where $d_k(i_v, i_{self})$ is the distance between the ego vehicle i_{self} and the vehicle i_v at time step t_k (a discretization of $t \in [0, T]$). v_{max} refers to the maximum speed legally allowed on the road. In addition, v_{avg} is the average recorded speed of all the vehicles during $t \in [0, T]$. W_{sep} and W_{cross} are the weights to balance between the different criterion characterizing (1): low separation and slow average intersection crossing speed. It should be noted that the proposed $J(\boldsymbol{Y})$ value is updated iteratively during the PC algorithm execution by the agents taking part in the coordination process. The number of iteration rounds is about $20 \sim 50$. Thus, the searching space should be carefully designed via an approach that ensures a sampling of "good" quality during the first action time. Readers are encouraged to read Philippe et al. (2019) for further information.

2.4 The draw-backs of the weighted method in original PC

As mentioned before, equation (1) is utilized without explicit safety constraints. For several cases, a very high weight W_{sep} may be admitted to penalize low separation distance to ensure more safe navigation. This can lead vehicles to preferably choose arbitrary low speeds (or a complete stop). Such behaviors may be regarded as very conservative. In real-time traffic navigation, UGV must have appropriate control architecture with reliable and real-time Risk Assessment and Management Strategies (RAMS). These targeted RAMS must reduce drastically the navigation risk in order to face sudden road hazards and dangerous situations. Unfortunately, the work shown in Philippe et al. (2019) does not provide a fully nil risk of collision and explicit risk-sensitive strategy. Thus, this paper aims to fill this gap and provide an ε-constraint PC algorithm (cf. Sect. 3) to compute the corresponding multi-criteria risk management strategy to guarantee 100% collision-free navigation in an appropriate prediction horizon.

3. RISK ASSESSMENT AND ε-CONSTRAINT METHOD IN PC ALGORITHM

Due to the probabilistic nature of the decision making problem between vehicles, it is hard and not straightforward to directly convert the constrains to probability space. Therefore, several heuristic repair approaches are applied to narrow the optimal solution, see Kulkarni and Tai (2010b). The elevated computational load limits thus the use of the PC approach in real-time environment. Kulkarni and Tai (2011) then handled these constrains by a penalty function method while knowing that the appropriate weights parameters (between sub-criteria) are not easy to be precisely obtained. In the proposed paper, the existing ε-constraint method inspired by Haimes (1971) is used in addition to the PC algorithm to solve the multi-criteria safety assignment MUGVs coordination problem. In this paper, a 2D Time-To-Collision (TTC) (cf. Sect. 3.1) is introduced as a constraint indicator in subsection 3.1. Accordingly, the assumed ε-PC will be detailed in sub-section 3.2.

3.1 2D TTC as a safety management indicator

As mentioned before, the previous work needs a risk assessment approach to succeed in the road hazard prediction. Thus, the TTC is used as a predictive safety measure of vehicle's trajectory. TTC is a risk indicator that describes the remaining time for a probable collision (i.e., traffic crash) between two vehicles as shown in Lakhal et al. (2019); Ben Lakhal. et al. (2020). It was originally defined by Hayward (1972) in car following scenarios. Generally, TTC can measure a road-user's time to react. To calculate TTC in two dimensions, we simply consider a collision of two circles as shown in Fig. 2.

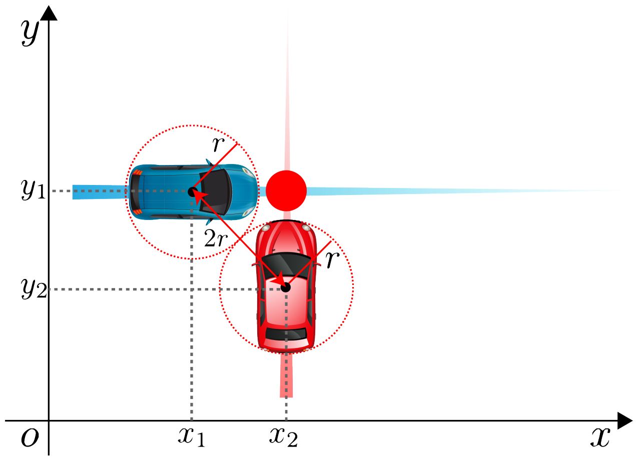

Fig. 2. Example of 2D TTC in two dimensions navigation

As one may notice that it is a "collision" of two circles (not a real crash of two vehicles). We use these circles to anticipate real accident. Here, $2r$ can be seen as vehicle length. In spite of sacrificing some accuracy, the 2D TTC between vehicles i, j can be more easily formulated in two dimensions as:

$$\begin{aligned} &[(x_i(t)+\dot{x}_i(t)\cdot TTC_{ij})-(x_j(t)+\dot{x}_j(t)\cdot TTC_{ij})]^2 \\ &+[(y_i(t)+\dot{y}_i(t)\cdot TTC_{ij})-(y_j(t)+\dot{y}_j(t)\cdot TTC_{ij})]^2 \\ &=(2r)^2 \end{aligned} \tag{2}$$

In equation (2), (x_i, y_i) and (x_j, y_j) are the position of vehicles i and j at time instant $t \in [0, T]$. $\dot{x}_i(t), \dot{x}_j(t), \dot{y}_i(t)$ and $\dot{y}_j(t)$ denote the relative speeds measured in x, y directions. Accordingly, we can get a polynomial function of the 2D TTC_{ij}, which can be solved by a quadratic discriminant. If there are real roots in (2), we can take the positive lower value as the nearest 2D TTC. For cases where roots are negative or equal to zero, it represents the collision that happened or will never happen with this dynamic.

In this paper, the objective of MUGVs system is to maximize the final agents' critical 2D TTC to improve the navigation safety. Thus, the corresponding objective function is defined as:

$$\begin{aligned} &J_{TTC}(\boldsymbol{Y}) = \min_{i,i\in\{1,2,\ldots,n\}(i\neq j)} \{TTC_{ij}(\boldsymbol{Y})\} \\ &\boldsymbol{Y} = [\Pi_k^1, \Pi_k^2, \ldots, \Pi_k^n](k=1,\ldots,N) \\ &\Pi_k^i \in \boldsymbol{\Pi}^i = \{\Pi_1^i, \ldots, \Pi_N^i\}(i=1,\ldots,n) \end{aligned} \tag{3}$$

Where $min\{TTC_{ij}(\boldsymbol{Y})\}$ represents the minimum TTC value of the most critical situation between n agents in the prediction horizon $t \in [0, T]$ within vehicles' combined actions/strategies $\boldsymbol{Y}$. It aims to maximize the J_{TTC} for more safety response to the concerned situation. Above all, an optimization problem can be formulated by considering equations (1) and (3). To handle the 2D TTC constraint, ε-PC algorithm is addressed in next section.

3.2 ε-constraint method in PC algorithm

The original PC algorithm focuses on a straightforward task with only one objective function as shown in (1). Nevertheless, the MUGVs system needs to deal with RAMS as suggested by the discussion given in Sect. 2.4. The ε-constraint method, which was firstly proposed by Haimes (1971), can be introduced to handle this trade-off problem. Only one objective function is optimized in that method, while others are converted into constraints with a permitted value ε by a limited range. In our case, the objective function J_{TTC} in (3) can be adopted as a constraint during optimizing the main objective function J in (1). Hence, the transformed objective function in PC algorithm is formulated as below:

$$\begin{aligned} &min && J(\boldsymbol{Y}) \\ &subject\ to && \boldsymbol{Y} = [\Pi_k^1, \Pi_k^2, \ldots, \Pi_k^n](k=1,\ldots,N) \\ &&& \Pi_k^i \in \boldsymbol{\Pi}^i = \{\Pi_1^i, \ldots, \Pi_N^i\}(i=1,\ldots,n) \\ &&& J_{TTC}(\boldsymbol{Y}) \geq \varepsilon \end{aligned} \tag{4}$$

We state the following theorems from Miettinen (2012):

Theorem 1. If objective J and vector $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_m)$ exist, such that $\boldsymbol{Y}^*$ is an optimal solution to the problem (4), then $\boldsymbol{Y}^*$ is a weakly Pareto optimal solution.

Theorem 2. $\boldsymbol{Y}^*$ is a strict Pareto optimal solution if and only if, for objective J, there exists a vector $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_m)$, such that $\boldsymbol{Y}^*$ is the unique objective vector corresponding to the optimal solution of the problem in equation (4).

An advantage of the ε-constraint method, as formulated in equation (4), is that we do not need to scale different objective functions by adding weights. The obtained solution, if it exists in (4) with a given parameter $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_m)$, is proved to be a weakly Pareto optimal solution as indicated by *Theorem 1* and *Theorem 2*. Actually, the Pareto front can be obtained by varying the vector ε. To find an efficient solution (that means close to a strict Pareto optimal solution) in problem (4), selecting an appropriate ε is the key. Accordingly, for calculating a more efficient solution, we must have at least the range of constraint objective function J_{TTC}. Unfortunately, the calculation of the J_{TTC} range in searching space is not a trivial task. The worst value is hard to compute, while we can get the best value in an individual optimization. Hence, a general selection of ε_m can be provided by (5):

$$J_{TTC}(\boldsymbol{Y}_{inf}^*) \leq \varepsilon_m \leq J_{TTC}(\boldsymbol{Y}_{sup}^*) \tag{5}$$

Where $\boldsymbol{Y}_{inf}^*$ is the optimal solution of single optimal problem (1) for minimum objective function J without any constraint, and $\boldsymbol{Y}_{sup}^*$ is the optimal solution for single optimal problem that maximizes J_{TTC} in equation (3) in a predefined searching space. After that, for the bounded value in equation (5), we define the range of normal J_{TTC} values as $J_{TTC}(\boldsymbol{Y}_{sup}^*) - J_{TTC}(\boldsymbol{Y}_{inf}^*)$ in problem (4). Note that, with the ε-constraint, we can get different efficient solutions close to a strict Pareto optimal solution. Therefore, more rich and flexible solutions are preferred in the applied traffic scenario. Mavrotas (2009) proposed to divide the ε range into p equal intervals by $p+1$ "grids points" like the following:

$$\begin{aligned} &\varepsilon_m = J_{TTC}(\boldsymbol{Y}_{inf}^*) + (J_{TTC}(\boldsymbol{Y}_{sup}^*) - J_{TTC}(\boldsymbol{Y}_{inf}^*)) \cdot (\frac{m}{p}) \\ &(m = 0, 1, \ldots, p) \end{aligned} \tag{6}$$

It is also essential to note that too small equal intervals will lead to ineffective 2D TTC constraint for safety sensitive solution. Let us consider equation (6), we can also get efficient solutions by properly adjusting the number of "grid points" gradually increasing ε_m by real-life referential signs and linear logic. An indicator $\sigma(\varepsilon_m)$ to interpret the linear relationship between J and J_{TTC} with different ε_m is calculated as in equation (7):

$$\sigma(\varepsilon_m) = \begin{cases} 1 & \text{if } \varepsilon_m = \varepsilon_p \\ \dfrac{\varepsilon_m - J_{TTC}(\boldsymbol{Y}^*_{inf})}{J_{TTC}(\boldsymbol{Y}^*_{sup}) - J_{TTC}(\boldsymbol{Y}^*_{inf})} & others \\ 0 & \text{if } \varepsilon_m = \varepsilon_0 \end{cases} \tag{7}$$

To sum up, the advantages of ε-constraint method are:

- ε-constraint method in PC algorithm avoids scaling multi-objective targets by changing too many weights.
- the number of efficient vehicle's actions can be chosen by properly adjusting the predefined grid points p. The membership function σ is designed to be adaptable.
- the feasible solutions obtained after the optimization are indeed Pareto optimal solutions.

A simple remedy in order to bypass the difficulty of estimating the worst values of the searching results (e.g., $J_{TTC}(\boldsymbol{Y}^*_{inf})$ with optimal $\boldsymbol{Y}^*_{inf}$ in (1) for minimum J) is to define reservation values for the objective functions as shown in Mavrotas (2009). Thus, we only need to calculate the maximum $J_{TTC}(\boldsymbol{Y}^*_{sup})$ in conventional PC algorithm. To calculate $J_{TTC}(\boldsymbol{Y}^*_{sup})$, several approximate methods, as the greedy heuristic search in Talbi (2009) for example, are recommended as fast initialization algorithms. To have a reference set of normal TTC assumption, the observed minimum TTC threshold depends on traffic scenario approximating $1.5s \sim 4.5s$, see Coffey and Park (2020). We capture minimum TTC threshold as $1.5s$ in our case.

4. MUGVS SYSTEM WITH ε-PC FRAMWORK

For proper analysis of the intersection cooperation, it is assumed that the proposed ε-PC algorithm is iteratively running on embedded navigation system. Vehicles in MUGVs system can communicate their actions/strategies (cf. Sect. 2.1) between each others. Thus, the on-board ε-PC can find sequentially an approximated optimal solution with respect to problem (4). A current best joint strategy $\boldsymbol{Y}^*_{cur,i}$ will be stored after a comparison of all solutions in vehicles i. The basic outline of the ε-PC algorithm to find $\boldsymbol{Y}^*_{cur,i}$ is presented as Algorithm 1 in the view of the ego-vehicle with the index i.

The proposed ε-PC is supposed to converge to the optimal results when the algorithm reaches the maximum permitted iterations or if the optimized results do not change in the predetermined number of iteration rounds. In addition, to discover more PC procedures of updating strategy probability distribution in Algorithm 1, readers are recommended to refer to Kulkarni and Tai (2010a). Further, the all time best solution $\boldsymbol{Y}^*_{opt}$ will be directly shared and applied in the predefined time horizon $[0,T]$ (cf. Sect. 2.1). Additionally, to integrate the risk constrained (i.e., 2D TTC) motion in MUGVs system, the ε-PC will identify the best solution $\boldsymbol{Y}^*_{opt}$ with a predefined "grid point" ε_m as mentioned before. The algorithm is executed outside the main PC loop. Considering the decentralized quality in the supposed case, it is likely to set the same control protocol for each vehicle. Hence, the accompanying pseudo code is also shown in Algorithm 2.

Algorithm 1: Vehicle i on-board ε-PC

Input: Possible actions $\boldsymbol{\Pi}^i, (i = 1, \ldots, n)$
Input: According probability $q(\boldsymbol{\Pi}^i)$
Output: Current best joint strategy $\boldsymbol{Y}^*_{cur,i}$
Output: Updated self-probability $q(\boldsymbol{\Pi}^i_{self})$

initial unified probabilities $q(\boldsymbol{\Pi}^i_{self})$
while *no convergence of* ε*-PC optimization* **do**
    for *each* $\Pi^i_{self} \in \boldsymbol{\Pi}^i_{self}$ **do**
        $\boldsymbol{Y}_{cur,i} \leftarrow \Pi^i_{self}$
        $\boldsymbol{Y}_{cur,i} \leftarrow$ Randomly sample $\Pi^{i_v}, i_v \neq i$ based on the probabilities $q(\boldsymbol{\Pi}^{i_V})$
    Compute expected utility $E(\boldsymbol{Y}_{cur,i})$ w.r.t. equation (4)
    $\boldsymbol{E}(\boldsymbol{Y}_{cur}) \leftarrow E(\boldsymbol{Y}_{cur,i})$ Store in a vector
$q(\boldsymbol{\Pi}^i_{self}) \leftarrow$ PC Optimization w.r.t. $\boldsymbol{E}(\boldsymbol{Y}_{cur})$
if $J(\boldsymbol{Y}_{cur,i}) < J(\boldsymbol{Y}^*_{cur,i})$ **then**
    $\boldsymbol{Y}^*_{cur,i} \leftarrow \boldsymbol{Y}_{cur,i}$;
return $\boldsymbol{Y}^*_{cur,i}, q(\boldsymbol{\Pi}^i_{self})$;

Algorithm 2: MUGVs system ε-PC framework

Input: Each vehicle i current best joint strategy $\boldsymbol{Y}^*_{cur,i}$
Input: Pre-selected grid point ε_m
Output: All time best joint strategy $\boldsymbol{Y}^*_{opt}, \Pi^i_{opt}$

initial Boolean value $BETTER = 0, SAFE = 0$
while *no convergence of* ε*-PC optimization* **do**
    for *each* $\boldsymbol{Y}^*_{cur,i}$ **do**
        if $J(\boldsymbol{Y}^*_{cur,i}) < J(\boldsymbol{Y}^*_{opt})$ **then**
            $BETTER = 1$
        if $J_{TTC}(\boldsymbol{Y}^*_{cur,i}) > \varepsilon_m$ **then**
            $SAFE = 1$
        if $BETTER$ & $SAFE$ **then**
            $\boldsymbol{Y}^*_{opt} \leftarrow \boldsymbol{Y}^*_{cur,i}$
        reset Boolean value $BETTER = 0, SAFE = 0$
Store $\boldsymbol{Y}^*_{opt}$ in all the vehicles
if *convergence of* ε*-PC* **then**
    Obtain $\boldsymbol{Y}^*_{opt}$ as all time best solution;
    Concatenating $\boldsymbol{Y}^*_{opt}$ with optimal strategy Π^i_{opt};
return $\boldsymbol{Y}^*_{opt}, \Pi^i_{opt}$;

5. EXPERIMENTAL VERIFICATION OF ε-PC

The experiments were run by a program developed in MATLAB with a computer of Core i5, 2.30GHz and 8GB RAM. Main parameters considered in the scenario are summarized in Table 1:

Table 1. Parameters and initial states

Parameters	Value	Parameters	Value
(x_1, y_1)	$[-20, -2.5]$ [m]	$[v_{min}, v_{max}]$	$[0, 10]$ [m/s]
(x_2, y_2)	$[20, 2.5]$ [m]	N_i	10
(x_3, y_3)	$[2.5, -20]$[m]	W_{sep}	1
$v_1(0)$	6.0 [m/s]	W_{cross}	10
$v_2(0)$	5.0 [m/s]	T_{sample}	0.1 [s]
$v_3(0)$	5.5 [m/s]	r	1.5 [m]

Three vehicles including left-turn maneuvers at an intersection were set in the simulation scenario (see Fig. 3 and Fig. 5 for example). To evaluate the proposed method, contrast experiments between original PC and ε-PC are given in the simulation (see respectively videos given in https://bit.ly/2PoYfjQ).

We highlight at first that the proposed approach can guarantee plan collision-free path in experiments. In original PC, the cost function considered by MUGVs system includes the average crossing time (altruistic objective) and the separation distance as shown in equation (1). The simulation results are illustrated in Fig. 4.

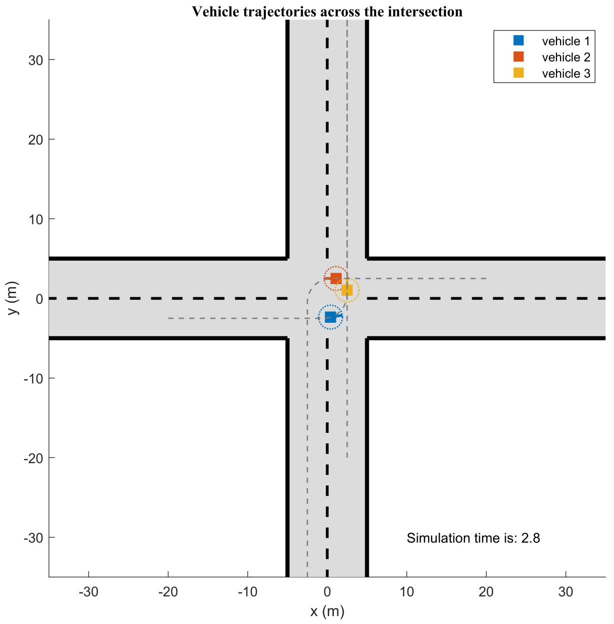

Fig. 3. Critical situation with the original PC

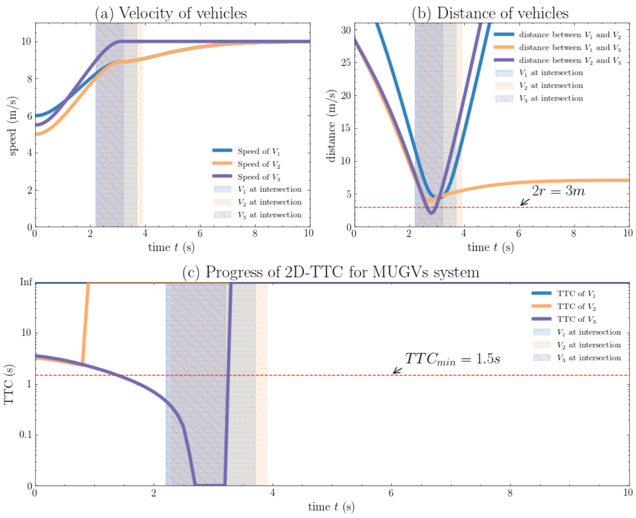

Fig. 4. Performance indicators: distance, velocity and 2D TTC in MUGVs system with original PC

Because the control effort has been focused on the crossing time ($W_{cross} > W_{sep}$, see Table 1), the original PC method attempted to maintain a fast crossing speed when it was applied in the simple-yet-dangerous scene like Fig. 3. Thus, there is a low probability but high impact to choose extreme strategy which allows all vehicles to accelerate as shown in Fig. 4 (a). Such speed growth had led to a collision as the distance indicator exhibited in Fig. 4 (b): the distance between vehicle 2 and 3 (purple line) violated the safety limit of $2r = 3m$. The 2D TTC profile of this two-vehicle also collapsed to zero during the collision as Fig. 4 (c).

In a comparison, ε-PC made vehicle 1 to maintain current speed at the beginning two seconds in order to increase the distance between adjacent vehicles as indicated in Fig. 6 (a). Due to the threshold constraint of $\varepsilon \geq TTC = 1.5s$ with respect to equation (4), only the feasible solutions respecting the constraints will be reserved in the ε-PC searching procedure. As a matter of fact, the proposed "ε-Constraint" in the bounds is to correctly estimate the trade off between crossing time and risk which we aimed to achieve safe and optimal trajectory schedule. Because the strategy hypotheses include full stop actions to avoid the extreme situation. Thus, a 100% free collision navigation can be guaranteed in the whole time horizon $[0s, 10s]$ as the indicator of distance and 2D TTC underline in Fig. 6 (b) and Fig. 6 (c).

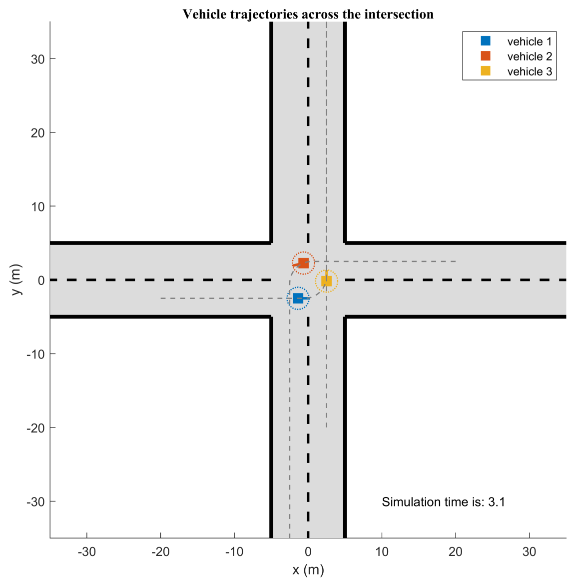

Fig. 5. Critical situation with the ε-PC

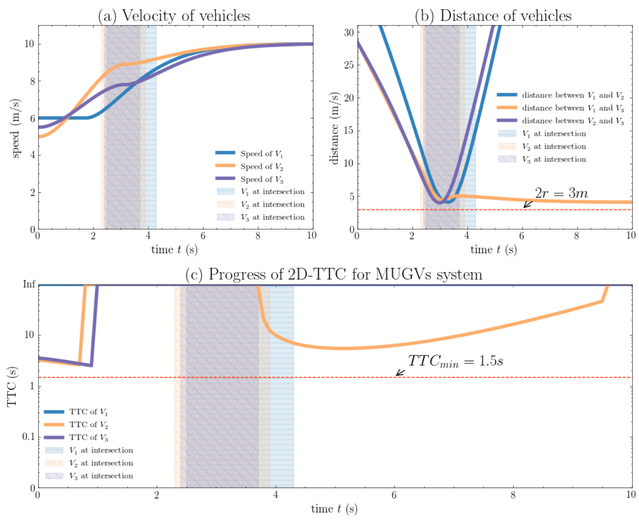

Fig. 6. Performance indicators: distance, velocity and 2D TTC in MUGVs system with ε-PC

Secondly, several ε-constraint values are carried out to highlight the performance of the proposed method in the previous scenario. The approximated maximum $J_{TTC}(\boldsymbol{Y}^*_{sup}) = 2.8975$ is fixed for the MUGVs system by initially using the heuristic searching (detailed in Sec 3.2). Here, the reservation value $J_{TTC}(\boldsymbol{Y}^*_{inf})$ is set equal to $1.5s$ for a minimum 2D TTC in whole time horizon $[0s, 10s]$. After that, several grids point are used in the range of $1.5 \leq \varepsilon_m \leq 2.8975$ to get a constraint set ε_i and membership function $\sigma(\varepsilon_i)$ as presented in Table 2.

Table 2. Parameters and initial states

	ε_i	$\sigma(\varepsilon_i)$	average T_{cross}	iterations	J_{TTC}
Trial 1	0	-	3.60 [s]	20	0.04 [s]
Trial 2	1.5	0%	4.70 [s]	21	2.07 [s]
Trial 3	2	36%	4.70 [s]	21	2.07 [s]
Trial 4	2.5	72%	3.93 [s]	26	2.51 [s]

It is instructive to note that the original PC algorithm, which does not use any 2D TTC constraint (i.e., $\varepsilon = 0$), shows a fastest crossing time with a lowest performances of 2D TTC as expected. Increasing ε_i with weighting more on the membership function $\sigma(\varepsilon_i)$ can generally provide more better temporal margins of 2D TTC while increasing vehicles average crossing time. However, ε-PC can still avoid conservative actions/strategies with low crossing time in dangerous situation as in trial 4 (about $1.33s$ late than trail 1). Moreover, the increasing iteration numbers implies that the convergence of the model needs more execution time. Therefore, MUGVs system has potential applications in different navigation environments when a proper selective ε-constraint model is designed.

Lastly, we investigate the influence of increasing vehicle number with ε-PC. MUGVs size ranged from three to six members. The average crossing time and iteration number regarding to the execution time are depicted in Fig. 7.

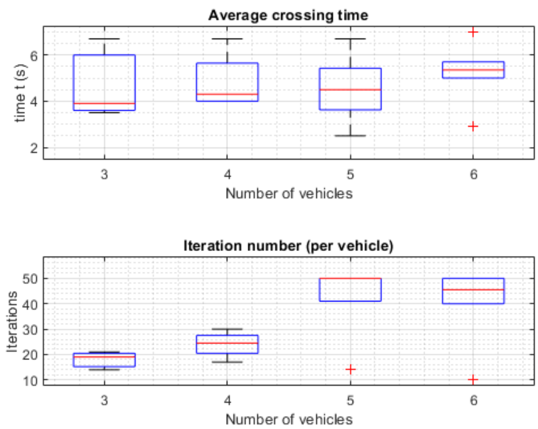

Fig. 7. Effect of vehicle number on the performance of ε-PC

The average crossing time shown in Fig. 7 did not decrease too much (from $t = 4.70s$ to $t = 5.22s$) after increasing the vehicle number. However, the outliers observed mean that an additional intermediate action (decelerate or accelerate) has been undertaken for individual vehicles. The iteration number plots show a non-linear increase in execution time with respect to the number of vehicles involved in the MUGVs system.

To summarize, the proposed method guarantees 100% of non collision between vehicles. Despite its sensitivity to the number of vehicles present in the considered intersection, the navigation still fluid i.e., operational and not too conservative. Notably, in a real situation at intersections, the maximum number of vehicles in cooperation will not exceed logically 10 vehicles for instance. Hence, the obtained results demonstrate that the proposed method is promising in solving navigation problems in intersections.

6. CONCLUSION

This paper proposed a distributed optimal approach for dynamic MUGVs navigation system with risk-sensitive management strategy. The proposed formulation uses the PC theory, which is a promising convex probability searching method, to optimize such a distributed problem. Vehicles in the MUGVs system only need to update the PD of actions set instead of searching best combined strategy in a non-linear objective function. Furthermore, in real-time traffic navigation, the proposed approach must have reliable behaviors to afford the demand of Risk Assessment and Management Strategies (RAMS) system in order to deal with sudden road hazards and risky situations. Therefore, a key safety indicator 2D TTC has been used as a risk assessment measure to impact vehicle's decision. We modeled the TTC in two dimensions and assume that the minimum 2D TTC value in MUGVs is the corresponding objective function. Such an application with the proposed ε-PC lead to promising results. Thanks to the proposed method, an approximate optimal and feasible solution can always be reached to provide risk margin in final actions/strategies. The proposed ε-PC can be applied in trajectories tracking, maneuvers warning/assisting or directly as feedback control law in MPC. The experiments shown in the paper prove the efficiency of the proposed ε-PC, and also the reliability of the 2D TTC as a risk indicator to guarantee a 100% no constraint violation even in extreme situations. Besides, the ε-PC will be implemented in real vehicles with better C coding for real-time experiments.

ACKNOWLEDGEMENTS

This work has been sponsored by the Chinese Ministry of Industry and Information Technology (MIIT) research program the 2020 Innovative Development of the Industrial Internet (TC200H033 and TC200H01F), by the Dongfeng Motor Corporation project 928 (DF928-2020-040). This work received also the support of IMobS3 Laboratory of Excellence (ANR-10-LABX-16-01).

REFERENCES

Ben Lakhal., N., Nasri., O., Adouane., L., and Slama., J. (2020). Reliable modeling for safe navigation of intelligent vehicles: Analysis of first and second order set-membership ttc. In *Proceedings of the 17th International Conference on Informatics in Control, Automation and Robotics - Volume 1: ICINCO,*, 545–552. INSTICC, SciTePress.

Berbeglia, G., Cordeau, J.F., and Laporte, G. (2010). Dynamic pickup and delivery problems. *European journal of operational research*, 202(1), 8–15.

Chen, L. and Englund, C. (2015). Cooperative intersection management: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 17(2), 570–586.

Coffey, S. and Park, S. (2020). Part-time shoulder use operational impact on the safety performance of interstate 476. *Traffic injury prevention*, 21(7), 470–475.

Di Cairano, S., Bernardini, D., Bemporad, A., and Kolmanovsky, I.V. (2013). Stochastic mpc with learning for driver-predictive vehicle control and its application to hev energy management. *IEEE Transactions on Control Systems Technology*, 22(3), 1018–1031.

Gregoire, J., Qian, X., Frazzoli, E., De La Fortelle, A., and Wongpiromsarn, T. (2014). Capacity-aware backpressure traffic signal control. *IEEE Transactions on Control of Network Systems*, 2(2), 164–173.

Guo, Q., Li, L., and Ban, X.J. (2019). Urban traffic signal control with connected and automated vehicles: A survey. *Transportation research part C: emerging technologies*, 101, 313–334.

Haimes, Y. (1971). On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE transactions on systems, man, and cybernetics*, 1(3), 296–297.

Hayward, J.C. (1972). Near miss determination through use of a scale of danger. *Report No. TTSC 7715*.

Huang, C.F., Bieniawski, S., Wolpert, D.H., and Strauss, C.E. (2005). A comparative study of probability collectives based multi-agent systems and genetic algorithms. In *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, 751–752.

Hult, R., Campos, G.R., Falcone, P., and Wymeersch, H. (2015). An approximate solution to the optimal coordination problem for autonomous vehicles at intersections. In *2015 American Control Conference (ACC)*, 763–768. IEEE.

Hyland, M. and Mahmassani, H.S. (2018). Dynamic autonomous vehicle fleet operations: Optimization-based strategies to assign avs to immediate traveler demand requests. *Transportation Research Part C: Emerging Technologies*, 92, 278–297.

Iberraken, D., Adouane, L., and Denis, D. (2018). Multi-level bayesian decision-making for safe and flexible autonomous navigation in highway environment. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3984–3990. IEEE.

Kulkarni, A.J. and Tai, K. (2010a). Probability collectives: a multi-agent approach for solving combinatorial optimization problems. *Applied Soft Computing*, 10(3), 759–771.

Kulkarni, A.J. and Tai, K. (2011). Solving constrained optimization problems using probability collectives and a penalty function approach. *International Journal of Computational Intelligence and Applications*, 10(04), 445–470.

Kulkarni, A.J. and Tai, K. (2010b). Probability collectives: a distributed optimization approach for constrained problems. In *IEEE Congress on Evolutionary Computation*, 1–8. IEEE.

Lakhal, N.M.B., Adouane, L., Nasri, O., and Slama, J.B.H. (2019). Interval-based/data-driven risk management for intelligent vehicles: Application to an adaptive cruise control system. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, 239–244.

Lakhal, N.M.B., Adouane, L., Nasri, O., and Slama, J.B.H. (2019). Risk management for intelligent vehicles based on interval analysis of ttc. *IFAC-PapersOnLine*, 52(8), 338–343. 10th IFAC Symposium on Intelligent Autonomous Vehicles IAV 2019.

Ma, C., Hao, W., Wang, A., and Zhao, H. (2018). Developing a coordinated signal control system for urban ring road under the vehicle-infrastructure connected environment. *Ieee Access*, 6, 52471–52478.

Manzinger, S. and Althoff, M. (2018). Tactical decision making for cooperative vehicles using reachable sets. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 444–451. IEEE.

Mavrotas, G. (2009). Effective implementation of the ε-constraint method in multi-objective mathematical programming problems. *Applied mathematics and computation*, 213(2), 455–465.

Miettinen, K. (2012). *Nonlinear multiobjective optimization*, volume 12. Springer Science & Business Media.

Philippe, C., Adouane, L., Tsourdos, A., Shin, H.S., and Thuilot, B. (2019). Probability collectives algorithm applied to decentralized intersection coordination for connected autonomous vehicles. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, 1928–1934.

Schwarting, W., Pierson, A., Alonso-Mora, J., Karaman, S., and Rus, D. (2019). Social behavior for autonomous vehicles. *Proceedings of the National Academy of Sciences*, 116(50), 24972–24978.

Suzuki, H. and Marumo, Y. (2018). A new approach to green light optimal speed advisory (glosa) systems for high-density traffic flowe. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 362–367. IEEE.

Talbi, E.G. (2009). *Metaheuristics: from design to implementation*, volume 74. John Wiley & Sons.

Vilca, J.M., Adouane, L., and Mezouar, Y. (2013). Reactive navigation of mobile robot using elliptic trajectories and effective on-line obstacle detection. *In Gyroscopy and Navigation Journal, Ed. Springer, Russia ISSN 2075 1087*, 4(1), 14 – 25.

Vilca, J.M., Adouane, L., and Mezouar, Y. (2018). Stable and flexible multi-vehicle navigation based on dynamic intertarget distance matrix. *IEEE Transactions on Intelligent Transportation Systems*, 20(4), 1416–1431.

Wolpert, D.H. (2006). Information theory—the bridge connecting bounded rational game theory and statistical physics. In *Complex Engineered Systems*, 262–290. Springer.

Xu, B., Ban, X.J., Bian, Y., Wang, J., and Li, K. (2017). V2i based cooperation between traffic signal and approaching automated vehicles. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, 1658–1664. IEEE.