International Journal of Intelligent Robotics and Applications (2021) 5:156–175
https://doi.org/10.1007/s41315-021-00181-4

REGULAR PAPER

Flexible multi-unmanned ground vehicles (MUGVs) in intersection coordination based on ε-constraint probability collectives algorithm

Zhengze Zhu[1,2,4] · **Lounis Adouane**[3] · **Alain Quilliot**[4]

Received: 31 October 2020 / Accepted: 26 April 2021 / Published online: 10 May 2021

Abstract
Cooperative navigation (CN) is a widespread technique to have efficient navigation of intelligent vehicles. Nonetheless, the CN strategies need to be more consistent in estimating and managing in-road risks. This paper outlines a flexible CN scheme for multiple unmanned ground vehicles (MUGVs) system to deal with such critical cooperative system. With its relative low execution time, the probability collectives (PC) algorithm has succeeded at generating fast and feasible solutions to cross intersections and roundabouts (Philippe et al. 1928–1934, 2019). However, the PC is still sensitive to uncertainty in the navigation process, which highlights the need to adopt several safety margins. This work focuses on balancing between the high-quality cooperative optimization and acceptable computational speed. Thus, a reliable risk management strategy is proposed by introducing a novel ε-constraint PC method. A real-time communication mechanism is suggested for a distributed system to avoid invalid behavior due to inconsistency. The novel ε-PC based navigation strategy allows the vehicles to adapt their dynamics and react to unexpected events while respecting real-time constraints. One finding appears to be well substantiated by the typical common-yet-difficult scenarios in intensive simulations. The ε-PC method can ensure collision-free behaviors and reserve at least 1.5s of reaction time for vehicles' safety insurance.

Keywords Multi-vehicle coordination · Probability collectives · Risk assessment and management · Real-time constraints · ε-constraint PC

1 Introduction

Similar to urban transportation systems, specific territories like large hospitals, university campus, commercial and industrial sites have taken steps to improve their navigation services in their shipment/transit areas (Hyland and Mahmassani 2018). Multi unmanned grounded vehicles (MUGVs) system in such restricted areas may help to provide more efficient transport services for passengers (Adouane 2016; Cordeau and Laporte 2007; Hyland and Mahmassani 2018). In the meantime, numerous simultaneous requests from multiple delivery locations may invoke cross-linked planning routes for MUGVs system. 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 cooperative navigation at intersection points.

1.1 Background

Before proceeding further, a typical graph of two connected UGVs cooperative navigation at an intersection is illustrated in Fig. 1. Similar to multi-robot systems (MRS), multiple levels of coordination between the different agents take place depending on the overall navigation system feed-backs.

In this paper, MUGVs are provided with an enhanced autonomy. They may manage the assignment of the navigation tasks by themselves through embedded decisional devices and inter-vehicle communication tools. Details about other important autonomous vehicle navigation technical issues, such as cooperative perception and localization,

✉ Zhengze Zhu
Zhengze.ZHU@etu.uca.fr

1 Institut Pascal, UMR CNRS 6602, Université Clermont Auvergne, 63178 Aubière Cedex, France

2 ASID, Hubei University of Automotive Technology, 442002 Shiyan, China

3 Heudiasyc, UMR CNRS 7253, Université de Technologie de Compiègne, 60203 Compiègne Cedex, France

4 LIMOS, UMR CNRS 6158, Université Clermont Auvergne, 63178 Aubière Cedex, France

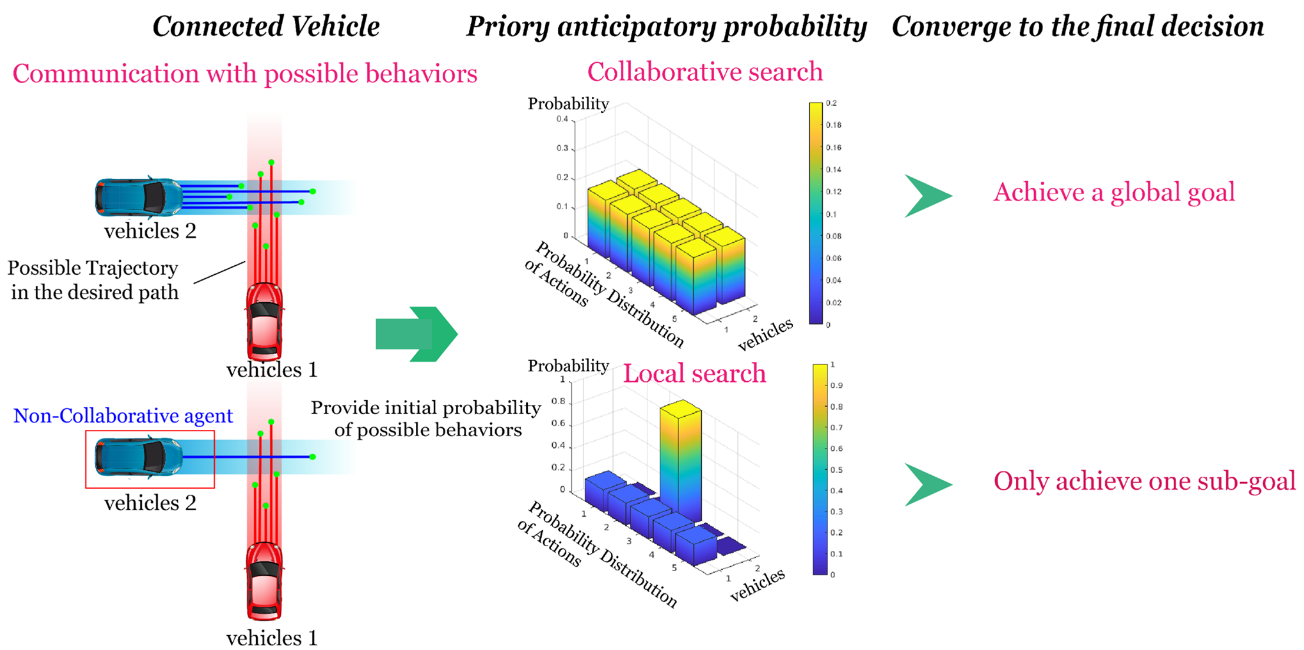


Fig. 1 MUGVs system with action probability distribution to predict their behaviors

planning and re-planning, control architectures may be found in Adouane (2016). Here in Fig. 1, two connected UGVs are which exchanging their predicted future motion trajectories. Because UGVs can better understand the behavior of each other, we consider UGVs more likely to have prosocial (or altruistic) behaviors rather than too conservative (or egoistic) behaviors (Schwarting et al. 2019). Thus, the two UGVs may perform a collaborative search of coordinated actions based on a utility-maximizing decision model. In presence of a non-collaborative agent (but this agent broadcast its estimated behaviors at first), then the UGV can only achieve a sub-goal of the navigation system by optimizing its own behavior. Let suppose that a priory anticipatory probability set is already specified to predict a potential action of the other vehicle. Nonetheless, the probability distribution of actions need to be updated while the UGVs preform their collaborative searching or local optimization process. In that respect, the data processed is stored and shared as files in a distributed system. Accordingly, a distributed approach is applied in a natural way to find out coordinated actions of the MUGVs system. Therefore, this paper aims to validate a decentralized approach to handle this distributed multi-agent optimization problem.

1.2 Related work

In the field of intersection coordination, the direct vehicle control approach has been applied to change the traffic lights pattern (Manzinger and Althoff 2018; Suzuki and Marumo 2018). Slots assignment to vehicles (Hult et al. 2015) is also a popular technique which is used in the same context. However, traditional traffic signal control methods in urban city usually cannot be applied directly in above mentioned areas, because traffic light is subjected to redundant cost in such a inappropriate formed crossing-road and in certain situations increase the level of traffic jam (Guo et al. 2019). Automation and communication have turned the cooperative intersection management into a more active research field (Chen and Englund 2015). 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) provide more details about such cases. Additionally, 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) (Kulkarni and Tai 2010), is applied to solve vehicle cooperation problem.

The PC algorithm is an efficient optimization searching framework for distributed systems, which was first proposed by Bieniawski (2005) and Wolpert (2006). It is a COllective INtelligence (COIN) framework that emerged from game theory, statistical physics, and optimization theory (Kulkarni and Tai 2010; Yang and Wu 2016). A comparative study has shown that the PC-based approach is superior to traditional genetic algorithm (GA) in both rate of decent and avoiding local minima (Huang et al. 2005). Kulkarni and Tai (2010) designed a shrink-sampling interval method to improve the algorithm performances via benchmark functions. After that, a PC-based approach successfully solved various discrete optimization problem like multiple traveling salesmen problem (MTSP) and vehicle routing problems (VRP) (Kulkarni and Tai 2010a, b). 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 (Philippe et al. 2019; Zhu et al. 2021). 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 $0.2s \sim 0.8s$ in the full optimization cycle). Thus, it is an interesting and promising method to process the aforementioned MUGVs navigation problem in restricted areas. The formulation of PC has been proposed in the air traffic control (Sislak et al. 2010). Although the goal was similar (conflict solving), the conflicts significantly increase in the cluttered land traffic environment. In the field of intersection management, PC algorithm was first proposed in Philippe et al. (2019) with the purpose of application to urban road transportation.

For a group of homogeneous MUGVs systems, researchers have tended to focus on efficient and effective controls to cut off with customer waiting time or energy consumption (Berbeglia et al. 2010). The existing MUGV dispatch study rarely discussed how to simultaneously maintain the optimal performance and also avoid risks at intersections. As a matter of fact, risk minimization has been shown to be considered as a priority in such a case (Chen and Englund 2015). Since sudden changes in the dynamics of ground vehicles in a short time are not realistic (Chen et al. 2018; Iberraken et al. 2018). There is still considerable ambiguity with regard to a risk assessment approach for safe and flexible navigation of MUGVs system.

Safe and smooth autonomous navigation technology have been widely considered in the intelligent advanced driver assistance systems (ADASs) (Nasri et al. 2019). Lane keeping assistance (LKA) and adaptive cruise control (ACC), for instance, are effective tools for obstacle avoidance driving in single vehicle control (Nasri et al. 2019). However, in the view of multi-vehicle cooperative navigation, the road accidents are more likely to be regarded as the failures of the multi-agents system rather than failures of any single vehicle (Milanés et al. 2012). The historical data approach is used to identify particular traffic accidents and apply safety countermeasures (Lord and Persaud 2004). Due to the sparse nature of traffic accidents, the use of such an approach is limited to perform safety analyses based on proper accident database records (Archer 2005). A more qualified form of risk management method is proximal safety indicator, which occurs more frequently for safety assessment and requires a short time for data collection (Archer 2005). Furthermore, the generally used proximal safety indicators are time measured metrics with a form of Time-To-X (e.g., Time-To-Accident, Time-To-Collision, Time-To-Break, etc.) (Hillenbrand et al. 2006; Horst 1991; Ward et al. 2014). Indeed, safety indicators provide an active approach assessing traffic conflicts to road users with reliable results. But these safe concerns lack of consistent definition or a robust theoretical foundation (Chin and Quek 1997). Among those methods, Time-To-Collision (TTC) is usually viewed as a more objective tool for predicting traffic accidents (Archer 2005; Ben Lakhal et al. 2019). A TTC based traffic event can be always recorded during the entire interactive process. Controllers can decide whether to adopt evasive maneuver in advance (with regard to the intention and purpose) rather than emergency braking at the last resort (Hydén 1987). The threshold for TTC is generally a definition that implies the risk-margin for drivers to react in a possible accident (Chin and Quek 1997). Arguably, TTC-concept is widely used as an important part of traffic conflict technique. However, it is more complex to detect the crucial traffic event by TTC in a two-dimensional spatial structure. Therefore, a 2D TTC is further developed in this paper as a risk-sensitive road proximal safety indicator.

To summarize, vehicles collaboration with risk management capabilities is a promising way to solve above mentioned problem. Additionally, the consideration of the real-time concerns and the management of several simultaneous actions are of utmost importance for such a distributed navigation system. Thereby, distributed real-time cooperative systems with a safety constraint (collision avoidance) can be generated in our case. Based on the PC theory, the previous research in Kulkarni et al. (2016) has investigated several off-line PC optimizations with soft constraints (e.g., tension/compression spring design problem). This paper focuses on the analysis of real-time MUGVs intersection coordination, by integrating a ε-constraint method into the PC algorithm to add safety indicator constraints (Haimes 1971; Mavrotas 2009). A more flexible multi-criteria decision-aids techniques and time consistency in distributed system will be furthered discussed in this paper.

1.3 Contributions and organization

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. The present paper also outlines different mode to handle newly approaching vehicle in context of real-time navigation. Hence, the main contributions of this paper can be summarized as follows:

- The formulation of ε-constraint searching scheme in the PC algorithm is first addressed in this paper. The proposed ε-PC can offer a risk-sensitive strategy for MUGVs system, which has been proven to be Pareto optimal. The risk margin management depends on different safety demands, which enables flexible and safe coordination to improve performances of MUGVs systems.
- A time-slot-based (TSB) vehicle communication mechanism is proposed to manage the cooperation of distributed MUGVs system. This method can cope in a better way with the iterative optimization process in the PC algorithm. We consider both the time interval for optimization and a more accurate prediction horizon for assisting vehicle navigation to avoid invalid behaviors related to inconsistency.
- The computational/communication analysis and navigation performance are analyzed accordingly for the dedicated application in both full collaborated mode and single optimal mode. The ε-PC can guarantee 100% collision free navigation. At the same time, its may play as a fast crossing strategy compared to the original PC via including some flexible safety requirements. A real-time application of the PC method for continuous traffic flow is tested. Our experiments prove that vehicles can adopt different modes to satisfy the real time constraints.

The rest of the paper is organized as follows: Sect. 2 introduces the development of the already proposed approach based on PC algorithm. The MUGVs system with optimal control is shown in Sect. 3. Section 4 presents the ε-constraint method in PC algorithm. A detailed use case is given in Sect. 5 to validate the addressed method for MUGVs systems. At last, conclusions and some prospects are given in Sect. 6.

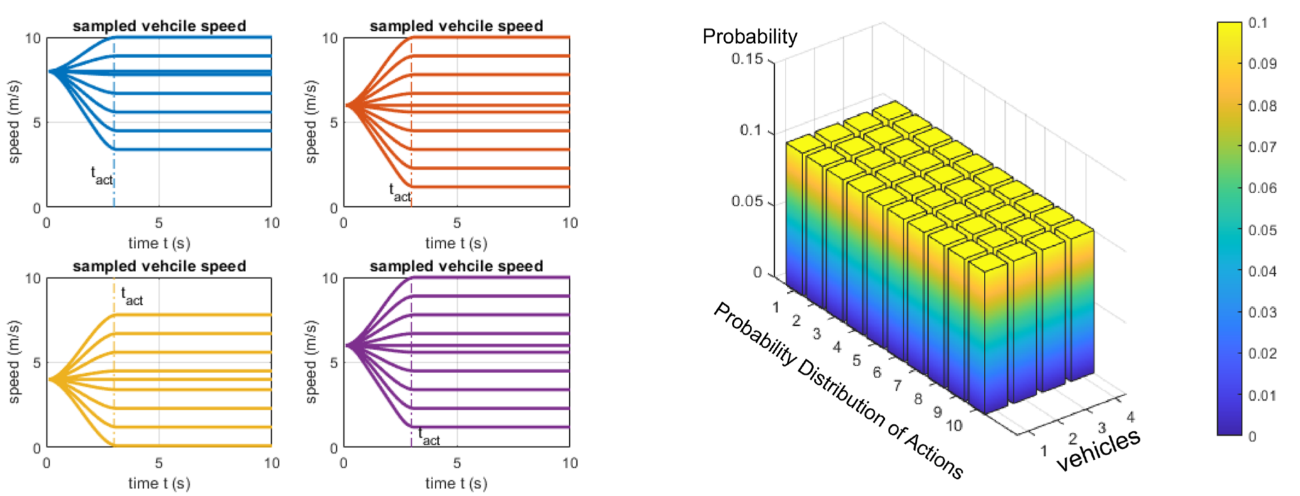
2 A conceptual review of the PC application to intersection coordination

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 intersections and roundabouts (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 of the MUGVs are the speed of navigation. PC treats the vehicles in a coordination problem as individual self-interested players iteratively (Wolpert 2006). Thus, these agents, in our case of study several vehicles (as shown in Fig. 2), 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 leaving an intersection as depicted in Fig. 2a.

Apparently, in Fig. 2a, there are considerable options N_i for each vehicle i depending on the initial speed $v_i(0)$. By both considering the safety and comfortable requests, intersection has a speed limit below $10m/s$ and vehicles tend to restrict acceleration in $[-2m/s^2, 2m/s^2]$ according to a restrained speed profile as illustrated in Fig. 2a. A further



(a) Possible speeds (b) Uniform distribution of all the agents' behaviors

Fig. 2 Example of strategies hypotheses for vehicle actions. **a** Possible speeds. **b** Uniform distribution of all the agents' behaviors 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. 2a). At last, the searching space for vehicle i can be summed up as a tuple $\boldsymbol{\Pi}^i$ i.e., $\boldsymbol{\Pi}^i \sim \{v_i(T), t_{act}, N_i\}, t \in [0, T]$. Then, the admissible member of actions set for the ego-vehicle can be presented as $\Pi^i \in \boldsymbol{\Pi}^i = \{\Pi_1^i, \dots, \Pi_{N_i}^i\}$. Here, Π^i can be visualized as the velocity profile in Fig. 2a.

As mentioned before, in the PC theory the expected utility of a given action can be calculated by each vehicle. But to do so, it must get (or estimate) the possible actions of the other vehicles. It has been used probability distribution to model relative actions like $q(\Pi_k^i) \in q(\boldsymbol{\Pi}^i) = \{q(\Pi_1^i), \dots, q(\Pi_{N_i}^i)\}$. Obviously, the preferred actions (or strategy) have a high probability of being cost-effective. The driver model used to improve the precision of the predictive control with probability distribution is a hot topic, but not the main research topic in this paper. Indeed, it is considered in the proposed work that this probability distribution is given by dedicated algorithm and according to that it is proposed an appropriate strategy to take the most appropriate decision making under this initial probability distribution. Readers are recommended to read Di Cairano et al. (2013) and Schwarting et al. (2019) to get clearer idea about the estimation of the probability distribution of other ground vehicles. The hypotheses of prosocial (or altruistic) UGVs in this paper make us formulating an uniform distribution (as shown in Fig. 2b) 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 (see Sect. 2.3). However, for the vehicles which 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. An important point that needs to be addressed is that re-acceleration should ensure continuity constraints of the speed. An algorithm that enables a continue speed profile after the action time have been designed in the previous work (Philippe et al. 2019).

2.3 Objective function

In its initial shape, the original PC approach considers only an unconstrained minimization problem. Such a research case generally involves n vehicles, and each vehicle $i \in n$ possesses a strategies/actions set of $\boldsymbol{\Pi}^i = \{\Pi_1^i, \dots, \Pi_N^i\} (i = 1, \dots, n)$ including an equal amount of N options (cf. Sect. 2.1). After performing a local motion planning through their on-board embedded devices, each vehicle applies a strategy $\Pi_k^i \in \boldsymbol{\Pi}^i (k = 1, \dots, N)$ during time interval $[0, T]$. Here, T refers to the prediction time horizon. During the period $[0, T]$, a particular set of combined strategies $\boldsymbol{Y} = [\Pi_k^1, \Pi_k^2, \dots, \Pi_k^n]$ is selected (randomly fixed to initialize the process) to reach at least a minimum system utility level $J([\Pi_k^1, \Pi_k^2, \dots, \Pi_k^n])$. The proposed objective function (Philippe et al. 2019) can be formulated as given by Eq. (1):

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

where $d_k(i_v, i_{self})$ is the distance between the ego vehicle i_{self} and the vehicle i_v (i.e., all collaborative vehicles) 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 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. Thus, the delicate designed searching space approach must ensure a sampling of "good" quality during the first action time. Readers are encouraged to read Kulkarni and Tai (2010) and Philippe et al. (2019) for further information.

2.4 The drawbacks of the weighted method in MUGVs safes navigation

As mentioned before, Eq. (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 risky situations. Unfortunately, the proposed previous work does not provide a fully nil risk of collision (Philippe et al. 2019) and explicit risk-sensitive strategy. Further, PC running time is inconsistent depending on the number of collaborative involved entities. Theoretically, MUGVs systems should have a certain time interval to start executing self-satisfied strategy targeting lower navigation risk. Thus, this paper aims to fill this gap and provide cycle-accurate description of these mechanisms in a systemic way. A method of limiting the time spent for optimizing is suggested at a common-yet-difficult scenario (cf. Sect. 3), which can calculate the consistent action execution time before entering a conflict zone. Furthermore, in the latter case, a constraint PC algorithm is proposed (cf. Sect. 4) to compute the corresponding multi-criteria risk management strategy to guarantee 100% collision-free navigation in an appropriate prediction horizon.

3 Application of MUGVs system: main assumptions and modeling aspects

Our specific objective is to explore PC theory enabling flexible and safe coordination to improve service performance of MUGVs system in restrained and complex areas. Therefore, we can cast our case in a customer pickup-and-delivery scenario. After routes are scheduled, MUGVs have to decide the actions at an intersection with an on-board autonomous control system as shown in Fig. 3. While considering the real life application, the PC method consists in planning motions. These motions (referenced as speed profiles) can be used to control a vehicle or to warn/assist the driver to avoid dangerous situations. Accordingly, a study of motion planning that satisfies real application of MUGVs system will be formulated within optimal control in Sect. 3.1. However, these planning is restricted by system intrinsic dynamic limitations and surrounding communication environments. As it will be clarified later, the Time-Slot-Based (TSB) communication approach is considered for better predicting how the vehicles will interact and collaborate with each others (cf. Sect. 3.2). More precisely, based only on every vehicle own-observations, the "single mode" addresses each vehicle individual motion planning without any further cooperation with other road participants. Contrarily, the "full mode" manages the reactions between all existing vehicles while ensuring the motion planning task as seen in Sect. 3.3.

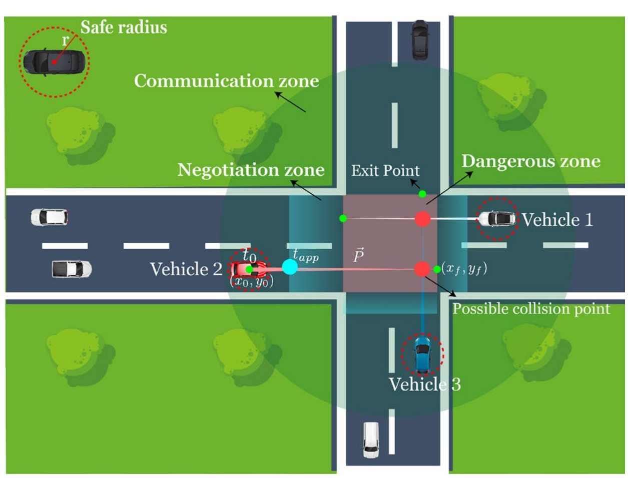


Fig. 3 Application scenario and main zones characterizing the addressed MUGVs systems

3.1 Optimal control framework of MUGVs system

Let suppose that our experimental scenario is as the following: vehicles track a desired path **P**, while searching the most appropriate velocity. If the chassis of an actual car is defined in an $x - y$ reference frame, we can denote the vehicle's position as (x, y), which represent the vehicle. The driving routes are identified by a series of way-points $(x_i, y_i) \in \mathbf{P}_i$ as illustrated in Fig. 3. Here, $(x_0, y_0)_i$ correspond to the the initial position at the time when the computation time is lunched for vehicle i, where $(x_f, y_f)_i$ is its final position. It comes that what we want to compute is the correspondence between time $t \in [0(\text{initial registered time}), T(\text{vehicle reaches its final position})]$. In Fig. 3, three positions of vehicles are indicated (at time $t = 0$) and the control is related to the speed $v_i(t)$ (that remains in an interval $[0, v_{max}]$). At any time t, the distance between any vehicle $i, j \in \{1, 2, 3\}$ cannot be less than a $2r$ threshold (in order to avoid the collision of the vehicles), where r is a safe radius for vehicle i, j as shown in Fig. 3.

Furthermore in Fig. 3, the communication zone is specified for inter-vehicle communication (IVC). After loading the computing at t_0, vehicles in communication zone can exchange the state information and priory anticipatory probability of possible actions before entering the intersection. Due to real-time computing environment, it has been given a deadline for MUGVs system return the strategies/actions for critical applications at the intersection. Here, a negotiation zone is defined w.r.t. action time t_{app} (data processing deadline) for synchronous cooperative navigation. Because vehicle's initial speed is different, the position of the vehicle begins to collaborate in negotiation zone will be different w.r.t. t_{app}. The dangerous zone (red block in Fig. 3) is more critical area which contains the possible collision points.

It is important to notice that all the vehicles are provided in this setup with the same kind of control devices (or control protocol). Henceforth, vehicles will follow the same algorithm logic and share current states when loading computation at t_0. As indicated before, we recommend setting a time limit for the solver, which ensures that the PC program will terminate in a reasonable period of time Δt_{sol} (cf. Sect. 3.2). This motion planning can be applied at time $t_{app} = t_0 + \Delta t_{sol}$. Because vehicles in our system is rolling without slipping (i.e., Pfaffian constraints), we can accurately predict vehicle's states (i.e., position and speed) at t_{app} when loading the algorithm at t_0. So, it is better to plan motions having the predicted horizon time that starts at $t = t_{app}$ for MUGVs system ($[t_{app}, t_{app} + T]$), where T is the prediction horizon. To do this, we can always pursue an optimal solution that guarantee well-coordinated motions in time. It is worth noting that the vehicle will precisely execute its final desirable actions/strategies (as $\Pi_k^i \rightarrow v_i(t)$) during time interval $[t_{app}, t_{app} + T]$. Some additional constraints are highlighted below for applying motion planning Π_k^i at t_{app}:

– The vehicle that has already entered the intersection is not concerned by the optimization process.
– Vehicle i keeps constant speed $v_i(0)$ (and less than v_{max}) before executing Π_k^i.
– If vehicle i will enter the intersection immediately after t_{app} with current speed $v_i(0)$, then treat it as a "non-collaborative agent" with maximizing self-utility strategy.

Dynamic constraints (e.g., inertial delay in powertrain) and trajectory deviation are not considered in this model. As mentioned above, we designed the PC to run in an iterative way. When the on-board algorithm is launched, it produces a possible action plan based on states of the current agents in communication zone with a prior knowledge of each other (cf. Fig. 3).

The best combined strategies should be saved and updated if new feasible optimal solution is obtained by the MUGVs system. To address the dynamic nature of the transportation system, users are permitted to change the action after adding any other agent in the MUGVs system. This means that the previous plan will be executed with the possibility of few changes occurrence when new cooperative actions become available.

3.2 Time-slot-based (TSB) inter-vehicle communication mechanism

In the above real life application of MUGVs system, the critical issue is the computing of each plan "fast enough" during Δt_{sol} by PC, in order so that the system can react to the changing environment without exceeding its motion capability [(defined as maneuverability (Bertolazzi et al. 2007)]. In our proposed PC implementation, the data exchange is aimed to be minimal and a solving time of $\Delta t_{sol} = 0.2s$ is targeted [($\Delta t_{sol} = 0.8s$ have been achieved so far for 4 vehicles (Philippe et al. 2019)]. Moreover, a reasonable Δt_{sol} leaves the system a maneuverability margin that can be used to move it into a safe configuration or state. To address this concern, we use time-slot-based (TSB) approach to further explain the vehicle's sequential optimization and communication mechanism in PC (see Fig. 4).

In TSB vehicle communication system, the basic time interval Δt_{sol} is divided into multiple duration. Here, "class regions" are highlighted for different message classes. We can use different wireless bandwidth for these regions. In class A, vehicles transmit the status information to other connected agents and exchange possible actions for entering an intersection. After vehicles get priory anticipatory probability of other vehicles' behaviors, the navigation problem can be formulated as an optimization model and the PC will be run in its default iterations. All the vehicles successively participate in the optimization with on-board PC algorithm at each iteration. They broadcast the updated probability distribution over the set of possible strategies for repeated computing reference. The successive iterations continues until all the updated probability distributions converge to stable distributions. However, long-running can be time-consuming and difficult to optimize. The timeout mechanism in class C helps to limit that time while supplying a satisfactory motion planning. Our optimizer triggers timeout when:

– All vehicles converge to stable probability distributions.
– The algorithm's running time exceeds timeout limits.

The conduct of timeout setup may lead to complete or to incomplete search. We are aware of the fact that vehicles do not guarantee convergence to the global optimum in our standard solving time Δt_{sol}. But the PC algorithm always retains the current best results at each iteration. These strategies tend to produce high quality solutions in short time.

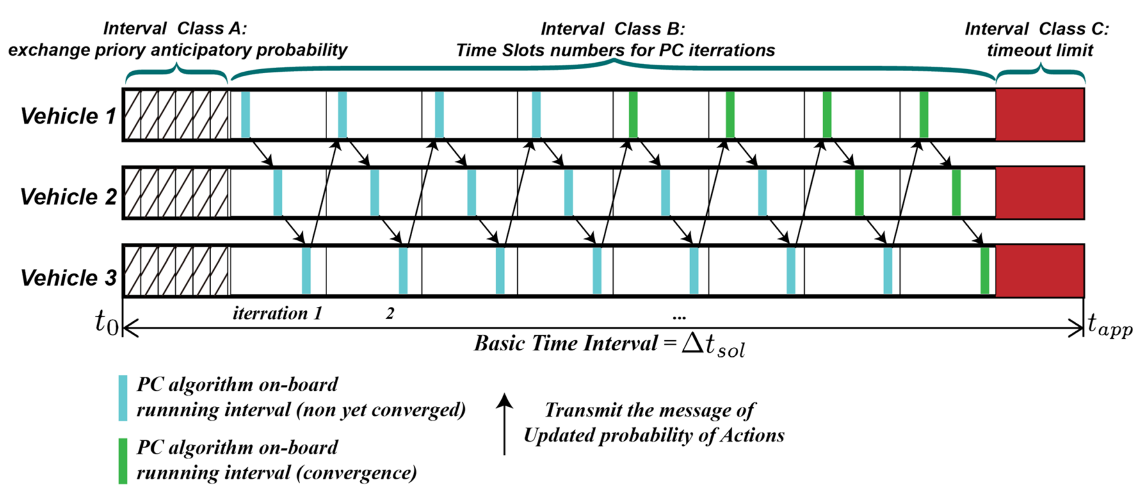


Fig. 4 Time-slot-based (TSB) communication mechanism for applying PC

The precise timeout limit value is changed depending on the running machines and experiment configuration but very close to the motion planning applied time t_{app}. It is interesting to note that the majority of our proposed PC models are either completed very quickly or they converge very slowly. Therefore, changing the timeout value (not too much) will not dramatically influence the computing results.

3.3 Real-time motion planning for MUGVs

To the best knowledge of the authors, the PC algorithm has not been used yet to perform repeated optimizations to deal with a continuous flow of vehicles. The closest application (Sislak et al. 2010) for air traffic management, but the algorithm was demonstrated on fixed initial situations. Since a repeated full optimization of MUGVs system is time consuming, we already defined two real-time motion planning modes of PC for searching feasible solutions in practical application:

– Single vehicle optimization (denoted “single” mode).
– Full optimization (“full” mode).

As explained, in the single mode, a vehicle runs the PC optimization as soon as it enters the communication zone as shown in Fig. 3. All the other vehicles are considered to be connected but non-collaborative agents. In this mode, the coordination is sequential more than the collaborative case, since the vehicles decide what to do one by one. This is an important option to reduce calculation time, because ego vehicle only needs to pick up the best actions with respect to fixed strategies of others. Also, the intersection crossing performance with the single mode may be sub-optimal as not all the vehicles coordinate with each other.

In the full optimization mode, the PC algorithm will run in its default iterative mode where all the vehicles participate in an optimization process as shown in Fig. 4. We recommend to perform the “full” mode long enough (10s for instance) to ensure vehicle exiting the intersection, and it should be triggered by a predefined event (such as a threshold number of vehicles at intersection). New vehicles entering in the restricted communication zone are not allowed to rerun the full optimization mode until the previous optimization is completed.

Later in this paper, authors focus in Sect. 5.3 to prove that the real-time solution of two modes can handle continue vehicles waves in practical real-time applications.

4 Risk assessment and ε-PC constraint method for PC algorithm

Due to the probabilistic nature of the decision-making problem between vehicles, it is hard and not straightforward to directly convert the constraints to probability space. Therefore, several heuristic repair approaches are applied to narrow the optimal solution (Kulkarni and Tai 2010a, b). The elevated computational load limits thus the use of the PC approach in hard real-time vehicle. Kulkarni and Tai et al., then handle the constraints by a penalty function method (Kulkarni and Tai 2011) while knowing that the appropriate weights parameters (between sub-criteria) are not easy to be obtained precisely. In the proposed paper, the existing ε-constraint method (Mavrotas 2009) is used in addition to the PC algorithm to solve the real time multi-criteria safety assignment MUGVs coordination problem. The navigation characteristics of MUGVs system and main constraints are highlighted in Sect. 3.1. Then, we introduce Time-To-Collision (TTC) as a constraint indicator in Sub-Sect. 4.1. Accordingly, the assumed ε-PC will be detailed in Sub-Sect. 4.2.

4.1 TTC as a safety management indicator

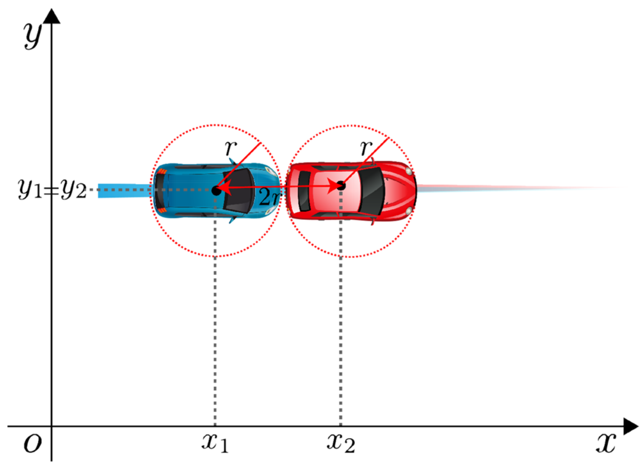
The purpose of MUGVs system is to compute a feasible solution, which serves all the customer in a flexible and risk-sensitive manner. The system objective function in Eq. (1) can offer a combined solution that penalizes low separation and slow average speed. However, as mentioned before, the previous 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. It was originally defined by Hayward (1972) in car following scenarios. Generally, TTC can measure a road-user’s time to react (for a critical collision event). The TTC at time instant $t \in [0, T]$ can be calculated according to the first order case (in co-linear navigation case between vehicles) (Ben Lakhal et al. 2020):

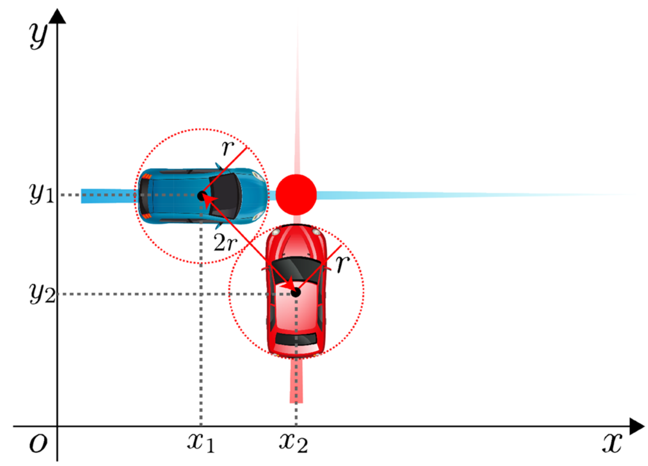
$$TTC = \frac{x_{lead}(t) - x_i(t) - 2r}{\dot{x}_i(t) - \dot{x}_{lead}(t)} \, , \tag{2}$$

where $(x_{lead}, y_{lead}), (x_i, y_i) \in \mathbf{P}$, $y_{lead} = y_i$, and $2r$ is the vehicle real length as Fig. 5a. In Eq. (2), x_{lead}, if exist, can be measured as the position of leading vehicle for vehicle i at x_i with speed $\dot{x}_i(t) > \dot{x}_{lead}(t)$. To calculate TTC in two dimensions, we simply consider a collision of two circles as shown in Fig. 5b.

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 l as in Eq. (2). In spite of sacrificing some accuracy, the TTC between vehicles i, j can be more easily formulated in two dimensions as:



(a) 2D TTC in co-linear navigation case



(b) 2D TTC in two dimensions navigation case

Fig. 5 A collision of two-vehicle based on circle area. **a** 2D TTC in co-linear navigation case. **b** 2D TTC in two dimensions navigation case

$$[(x_i(t) + \dot{x}_i(t) \cdot TTC_{ij}) - (x_j(t) + \dot{x}_j(t) \cdot TTC_{ij})]^2 + [(y_i(t) + \dot{y}_i(t) \cdot TTC_{ij}) - (y_j(t) + \dot{y}_j(t) \cdot TTC_{ij})]^2 = (2r)^2. \tag{3}$$

In Eq. (3), setting $(x_i, y_i) \in \mathbf{P}_i, (x_j, y_j) \in \mathbf{P}_j$ is the position of vehicle i, j at time instant $t \in [0, T]$. $\dot{x}_i(t), \dot{x}_j(t), \dot{y}_i(t), \dot{y}_j(t)$ denote the relative speeds measured in x, y directions. Accordingly, we can get a quadratic function of TTC_{ij} like:

$$\begin{aligned}&[(\dot{x}_i(t) - \dot{x}_j(t))^2 + (\dot{y}_i(t) - \dot{y}_j(t))^2] \cdot TTC_{ij}^2 \\ &\quad + 2[(x_i(t) - x_j(t))(\dot{x}_i(t) - \dot{x}_j(t)) + (y_i(t) - y_j(t))(\dot{y}_i(t) - \dot{y}_j(t))] \cdot TTC_{ij} \\ &\quad + [(x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2 - (2r)^2] = 0.\end{aligned} \tag{4}$$

Equation (4) can be solved by quadratic discriminant. If there are real roots in (4), we can take the positive lower value as the nearest TTC in the prediction horizon. For cases where roots are negative or equal to zero, it represents the collision that have happened. To avoid any confusion, we defined the solutions in Eq. (4) as "2D TTC" in the following of this paper. The objective of MUGVs system is to maximize the final agents' 2D TTC to improve the navigation safety. Thus, the corresponding objective function is defined as:

$$\begin{aligned}&\mathit{max} && J_{TTC}(\boldsymbol{Y}) = \min_{i,j \in \{1,2,\dots,n\}(i \neq j)} \{TTC_{ij}(\boldsymbol{Y})\} \\ &\mathit{subject\ to} && \boldsymbol{Y} = [\Pi_k^1, \Pi_k^2, \dots, \Pi_k^n](k = 1, \dots, N) \\ &&& \Pi_k^i \in \boldsymbol{\Pi}^i = \{\Pi_1^i, \dots, \Pi_N^i\}(i = 1, \dots, n),\end{aligned} \tag{5}$$

where $\min\{TTC_{ij}(\boldsymbol{Y})\}$ represents the minimum 2D TTC value of the most critical situation between n agents in the prediction horizon $t \in [0, T]$ within vehicles' combined actions/strategies $\boldsymbol{Y}$. J_{TTC} aims to maximize the critical 2D TTC value for more safety response to the concerned situation. Above all, an optimization problem can be formulated by considering Eqs. (1) and (5). To handle the TTC constraint, ε-PC algorithm is addressed in next section (cf. Sect. 4.2).

4.2 ε-PC algorithm

The original PC algorithm focuses on a straightforward task with only one objective function as shown in Eq. (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 in Haimes (1971), can be introduced to handle this trade-off problem. Only one objective function is optimized in the 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 Eq. (5) can be adopted as a constraint during optimizing the main objective function J in Eq. (1). Hence, the transformed optimization problem is formulated as below:

$$\begin{aligned}&\mathit{min} && J(\boldsymbol{Y}) \\ &\mathit{subject\ to} && \boldsymbol{Y} = [\Pi_k^1, \Pi_k^2, \dots, \Pi_k^n](k = 1, \dots, N) \\ &&& \Pi_k^i \in \boldsymbol{\Pi}^i = \{\Pi_1^i, \dots, \Pi_N^i\}(i = 1, \dots, n) \\ &&& J_{TTC}(\boldsymbol{Y}) \geq \varepsilon.\end{aligned} \tag{6}$$

According to the model, the optimal results could be given by the following theorems. The interested readers may consult Miettinen (2012) for more details:

Theorem 1 *If objective J and vector $\varepsilon = (\varepsilon_1, \dots, \varepsilon_m)$ exist, such that $\boldsymbol{Y}^*$ is an optimal solution to the problem (6), 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 $\varepsilon = (\varepsilon_1, \dots, \varepsilon_m)$, such that $\boldsymbol{Y}^*$ is the unique objective vector corresponding to the optimal solution of the problem given in Eq. (6).*

4.2.1 ε selection

An advantage of the ε-constraint method, presented in Eq. (6), is that we do not need to scale different objective functions by adding weights. The obtained solution, if it exists in Eq. (6) with a given parameter $\varepsilon = (\varepsilon_1, \dots, \varepsilon_m)$, is proved to a weakly Pareto optimal solution as Theorem 1 and 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 (6), 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 Eq. (7):

$$J_{TTC}(\boldsymbol{Y}^*_{inf}) \leq \varepsilon_m \leq J_{TTC}(\boldsymbol{Y}^*_{sup}), \tag{7}$$

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 maximize J_{TTC} in Eq. (5) in a predefined searching space. After that, for the bounded value in Eq. (7), we define the range of normal J_{TTC} values as $J_{TTC}(\boldsymbol{Y}^*_{sup}) - J_{TTC}(\boldsymbol{Y}^*_{inf})$ in problem (6). Note that, with the ε-constraint, we can get different efficient solutions close to a strict Pareto optimal solution. Therefore, a more rich and flexible solutions are favorable in the applied traffic scenario. Thus, we can divide the ε range into p equal intervals by $p + 1$ "grids points" (Mavrotas 2009) like the following:

$$\varepsilon_m = J_{TTC}(\boldsymbol{Y}^*_{inf}) + (J_{TTC}(\boldsymbol{Y}^*_{sup}) - J_{TTC}(\boldsymbol{Y}^*_{inf})) \cdot \left(\frac{m}{p}\right), (m = 0, 1, \dots, p). \tag{8}$$

Let consider Eq. (8), we can also get efficient solutions by properly adjusting the the number of "grid points" gradually increasing ε_m by 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:

$$\sigma(\varepsilon_m) = \begin{cases} 1 & \text{if } \varepsilon_m = \varepsilon_p \\ \frac{\varepsilon_m - J_{TTC}(\boldsymbol{Y}^*_{inf})}{J_{TTC}(\boldsymbol{Y}^*_{sup}) - J_{TTC}(\boldsymbol{Y}^*_{inf})} & \textit{others} \\ 0 & \text{if } \varepsilon_m = \varepsilon_0, \end{cases} \tag{9}$$

For the bounded value in Eq. (7), we define the range of normal ε_m by two bound values J_{TTC} with respect to the individual optimal problems. 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 a good trajectory schedule. To guarantee the feasible solution in the bounds, we divide ε_m into several equal intervals as a constraint in original PC algorithm. Only the feasible solutions afford the constraints will be reserved in the PC searching procedure as depicted in Fig. 6. It is also essential to note that too small bound intervals will lead to ineffective 2D TTC constraint for a safety-sensitive solution. A simple remedy in order to bypass the difficulty of estimating lower bound is to define reservation values as shown in Mavrotas (2009). We capture minimum 2D TTC threshold as $1.5s$ for a reference in this paper (Coffey and Park 2020). Because the strategy hypotheses include full stop actions to avoid the extreme situation (i.e., conflict immediately), thus ε-PC can filter the decision states while remains optimal feasible solutions.

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

- ε-constraint method in PC algorithm avoids scaling multi-objective function in a complex target function by adding too much weights.
- we can control the number of efficient vehicle's actions by properly adjusting ε_m with predefined grid points p. A membership function σ can indicates the degree of optimization in different objective functions.
- the feasible solutions obtained after the optimization are indeed Pareto optimal solutions.

A simply 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 (Mavrotas 2009). Thus, we only need to calculate the maximum $J_{TTC}(\boldsymbol{Y}^*_{sup})$ in conventional PC algorithm. In the context of the proposed MUGVs system, several approximate block solvers are recommended as initialization fast algorithms. For example, adopting max-min resource method in Jansen (2004) to calculate $J_{TTC}(\boldsymbol{Y}^*_{sup})$. It is also essential to note that too small equal intervals will lead ineffective 2D TTC constraint for safety sensitive solution. Therefore, setting 2D TTC constraint indicators are expected to regard real-life situations. Furthermore, the ε-PC algorithm can be explained with detailed flow diagram as shown in Fig. 6.

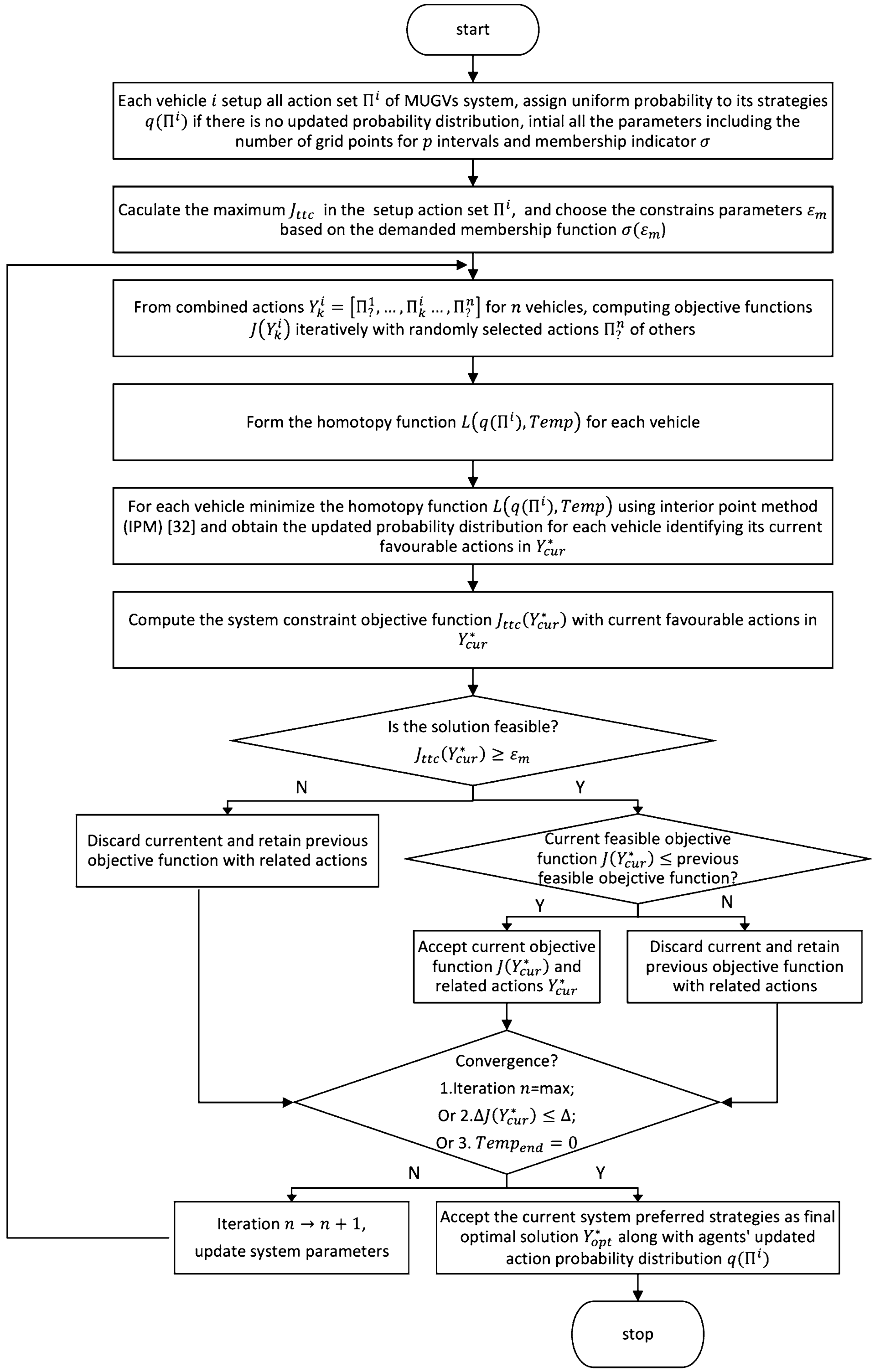


Fig. 6 Flowchart of the proposed ε-PC algorithm

4.2.2 ε-PC framework

In Fig. 6, as the original PC method, vehicle i assigns uniform probabilities $q(\Pi_k^i)$ to its strategies/actions set $\boldsymbol{\Pi}^i$ (for example, $q(\Pi_k^i) = 1/N$ is a distribution over $\boldsymbol{\Pi}^i$) at $t = 0$. From a vantage point of associate a probability for the strategy Π_k^i, vehicle i can further compute the N corresponding expected system objective function values w.r.t. its strategies set $\boldsymbol{\Pi}^i$. Thus, when vehicle i in turn to run its PC algorithm, it can help to optimize the distribution $q(\boldsymbol{\Pi}^i)$ (for ego-vehicle) in a expectation function like Eq. (10).

$$\begin{aligned} & min && E(q(\boldsymbol{\Pi}^i)) = \sum_{k=1}^{N} J(\boldsymbol{Y}_k^i) q(\Pi_k^i) \prod_{(i)} q(\Pi_?^{(i)}) \\ & subject\ to && (\boldsymbol{Y}_k^i = [\Pi_?^1, \Pi_?^2, \dots, \Pi_k^i \dots, \Pi_?^n]) \\ & && q(\Pi_k^i) \in q(\boldsymbol{\Pi}^i) = \{q(\Pi_1^i), \dots, q(\Pi_N^i)\} \\ & && \sum_{k=1}^{N} q(\Pi_k^i) = 1, \quad q(\Pi_k^i) \geq 0, \end{aligned} \tag{10}$$

where (i) represents every vehicle other than i, and $\Pi_?^{(i)}$ is the other vehicle's strategies selected randomly (with question marks "?") depending on its probability $q(\Pi_?^{(i)})$. It is important to underline that $q(\Pi_?^{(i)})$ is a priory anticipatory probability of the actions/strategies of all the other agents. For vehicle i in turn to minimize expectation function $E(q(\boldsymbol{\Pi}^i))$, a combined strategy $\boldsymbol{Y}_k^i$ include its own strategy Π_k^i and other randomly selected strategy $\Pi_?^{(i)}$ (i.e., $\boldsymbol{Y}_k^i = [\Pi_?^1, \Pi_?^2, \dots, \Pi_k^i \dots, \Pi_?^n]$). Thus, we can underline these the combined probabilities distribution of in $\boldsymbol{Y}_k^i$ w.r.t. each Π_k^i as the following:

$$\begin{aligned} q(\boldsymbol{Y}_1^i) &= [q(\Pi_?^1), q(\Pi_?^2), \dots, q(\Pi_1^i), \dots, q(\Pi_?^n)] \\ &\vdots \\ q(\boldsymbol{Y}_k^i) &= [q(\Pi_?^1), q(\Pi_?^2), \dots, q(\Pi_k^i), \dots, q(\Pi_?^n)] \\ &\vdots \\ q(\boldsymbol{Y}_N^i) &= [q(\Pi_?^1), q(\Pi_?^2), \dots, q(\Pi_N^i), \dots, q(\Pi_?^n)]. \end{aligned} \tag{11}$$

Thanks to the cost function in Eq. (10), it is easier to optimize probability than the original problem in (6). Such a method is referred as the homotopy approach that converts the primal problem into the probability space. Next, a key attraction, and most important maximum-entropy (MaxEnt) principle in PC algorithm is applied, so that we formulate $E(q(\boldsymbol{\Pi}^i))$ into Eq. (12):

$$\begin{aligned} L(q(\boldsymbol{\Pi}^i), Temp) &= E(q(\boldsymbol{\Pi}^i)) - Temp \times E_{free} \\ &= \sum_{k=1}^{N} J(\boldsymbol{Y}_k^i) q(\Pi_k^i) \prod_{(i)} q(\Pi_?^{(i)}) - Temp \\ &\times \left(- \sum_{k=1}^{N} q(\Pi_k^i) \log_2 q(\Pi_k^i) \right). \end{aligned} \tag{12}$$

The objective function (12) is called MaxEnt Lagrangian and is widely used in statistical physics considering free energy as E_{free} (Yang and Wu 2016): a parameter $Temp$ called temperature is specific for the simulated annealing (SA) process. At the beginning of the ε-PC algorithm, the parameter $Temp \in [0, \inf)$ is huge, which weights more the entropy term. Then we can always get a uniform probabilities. Since function E_{free} can stand for the largest uncertainty (highest entropy) at the beginning (this means each vehicle's actions has probability $1/N$ of being most favorable). Shannon entropy is a general choice for function $E_{free} = -\sum_{k=1}^{N} q(\Pi_k^i) \log_2 q(\Pi_k^i)$, where it can be proved mathematically that: $argmax(E_{free}) \rightarrow q(\Pi_k^i) = 1/N$.

The formulation of the Maxent Lagrangian $L_i(q(\boldsymbol{\Pi}^i), Temp)$ is very appropriate in the original PC theory, since the probability nature may handle the rest of work to solve in a convex space of probability distribution. To obtain the updated probability, the Broyden–Fletcher–Goldfarb–Shannon (BFGS) method is used in PC algorithm for the reformulated optimization problem in Eq. (13):

$$\begin{aligned} & min && L(q(\boldsymbol{\Pi}^i), Temp) \\ & subject\ to && \\ & && \sum_{k=1}^{N} q(\Pi_k^i) = 1 \\ & && q(\Pi_k^i) \geq 0. \end{aligned} \tag{13}$$

In Eq. (13), the expected global utility $L(q(\boldsymbol{\Pi}^i), Temp)$ based on the combined strategy is calculated under a specific temperature $Temp$. Vehicle i updates the probabilities $q(\boldsymbol{\Pi}^i)$ of all the actions after each iteration. However, the adoption of BFGS method cannot keep the probability value within [0, 1]. Even though standardization may be used to handle such case, the interior point method (Yang and Wu 2016) is recommended in the proposed ε-PC algorithm. Because it can be guaranteed that the probability keep within [0, 1] during each iteration. Interior point method have been proven to efficiently solve linear (or-nonlinear) constraints with less iterations.

After that every vehicle runs the ε-PC algorithm, the probability distribution of its actions will be updated. Combined strategies $J(\boldsymbol{Y}_{cur}^*)$ with the minimum value $J(\boldsymbol{Y}^*)$ will be saved as current preferred solution. It must be mentioned that, the accepted combined strategies $\boldsymbol{Y}_{cur}^*$ as current preferred solution at an iteration afford $J_{TTC}(\boldsymbol{Y}_{cur}^*) \geq \varepsilon_m$. Otherwise, discard $J(\boldsymbol{Y}_{cur}^*)$ and retain previous objective function value with related actions. As last, if any of the three criteria listed below is valid, then, accept the current system best solution $\boldsymbol{Y}_{cur}^*$ as the final optimal strategy $\boldsymbol{Y}_{opt}^*$ of all the vehicle.

- if temperature $Temp = Temp_{end} \rightarrow 0$
- if maximum number of iterations exceeded
- if the difference of objective function $J(Y_{cur}^*)$ between two iterations reaches the prescribed threshold of Δ

Above all, the main difference between ε-PC algorithm and the original PC framework can be highlighted as following:

- The process which confirms an available constraints to the feasible solutions is considered in the PC framework in a randomly improved way with additional calculation steps. The proposed method uses accessible individual optimization process to define the ranges of constraints in advance. The grid points is inserted in the algorithm, therefore, a feasible solution can always be calculated in due time.
- The interior point method is used in the improved PC algorithm to guarantee the probability value within [0, 1]. It is essential to apply the Monte-Carlo sampling principle based these probability distribution rather than standardization.
- The process of narrowing/updating sampling interval is excluded in the proposed ε-PC algorithm. Because the searching space is well-designed before (see Sect. 2.1). This fact leads us to the obvious advantage of reducing computational time.

The main interest of the proposed ε-PC algorithm is a proper balance between the high-quality solution and acceptable computational speed. The method is flexible and produces approximation algorithm solution rather than global optimal results. It is supposed to be a good decision support system for transportation service and risk assessment. The typical MUGVs coordination will be tested in Sect. 5.

5 Experimental verification of ε-PC

5.1 Communication characterization for the expected data exchange between vehicles

In ε-PC algorithm, we designed a distributed approach in the hypothesis that there are n processors with the ith processor (vehicle i) assigned the responsibility of updating the ith probabilities $q(\Pi_k^i)$ of actions/strategies set $\boldsymbol{\Pi}^i$ according to Eq. (13). Processor i inform processor j (e.g., vehicle j) to calculate its preferred actions/strategies relying information of preceded probabilities $q(\Pi_k^i)$. All the processors repeat the process until the convergence the aforementioned iterative sequence (cf. Fig. 4).

Due to the limitations in the measurement and control units, it is often impossible to acquire measurements at an arbitrarily fast speed and to execute the control inputs instantaneously. Thus, the MUGVs system in this paper is described in a continuous-time setting while measurements and control inputs are described in a precise constant sampled data $T_{sample} = 0.2s$. It is assumed also that the messages are received correctly within a finite time (but still leave an open way to consider synchronous or asynchronous implementations). In our predefined scenario, for example three connected vehicles in Fig. 3, each action/strategy is a float vector of size 50 ($10s$ horizon and $0.2s$ sampling time) and the set of possible velocity profile has 10 strategies as shown in Fig. 2a. That is a total of 1500 floats—5.86 kB (kilo Bytes)—for three vehicles. It is done again at the beginning of the re-acceleration phase (cf. Sect. 2.2). Consequently total $11.72kB$ is prior data volume need to be broad cast in the considered case. Then, for each iteration the vehicle broadcasts its updated probability vector $q(\Pi_k^i)$ of 500 floats ($10s$ horizon and $0.2s$ sampling time with 10 strategies). The number of iteration is various in different mode according to experimental statistics. So the total broadcast per vehicle is depend on the iterations. Accordingly, the communication demand in “single” (only one vehicle) and “full” modes (for three vehicles) are compared in Table 1 as an instance.

We suppose that the required network throughput for all modes will not exceed a magnitude of 0.4 MB/s as shown in Table 1. Then, the optimization can be achieved in about $0.2 \sim 0.8s$. A faster network throughput (e.g., 4 MB/s) is physically possible. Therefore, our ε-PC method may be executed with on-board processors that have a better C++ implemented code for a faster running. But it is important to note that the experiments in this paper were all run by a program developed in MATLAB with a computer of Core i5-6300HQ, 2.30 GHz and 8 GB RAM.

Table 1 Data exchange in “single” mode and “full” mode for three vehicles

Single mode	Value	Full mode	Value
Prior data	11.72 kB	Prior data	11.72 kB
Searching strategy	10	Searching strategy	10^3
Iterations estimation	10	Iterations estimation	$20 \sim 50$
Probability broadcast	500 floats (1.95 kB)	Probability broadcast	1500 floats (5.86 kB)
Data volume	19.53 kB	Data volume	$117.19 \sim 292.97$ kB
Total volume	31.25 kB	Total volume	$128.91 \sim 304.69$ kB
Solver time estimation	0.2 s	Solver time estimation	0.8 s
Network request	0.4 MB/s	Network request	0.4 MB/s
Physically possible	4 MB/s	Physically possible	4 MB/s

5.2 Parameter setting and results evaluation for full optimization

Indeed, the single and full modes are both processed by ε-PC. The main difference between the two strategies is that the single mode only considers its best options w.r.t. the fixed strategies of others. It is a special case of full optimization that vehicles have several actions that may be chosen for self-interested behavior. Therefore, it is discussed in what follows only full optimization of MUGVs system. To explain better the performance of the proposed intersection navigation scheme for the MUGVs system based on ε-PC, let us decompose the experiments into several parts to evaluate the characteristics of the algorithm.

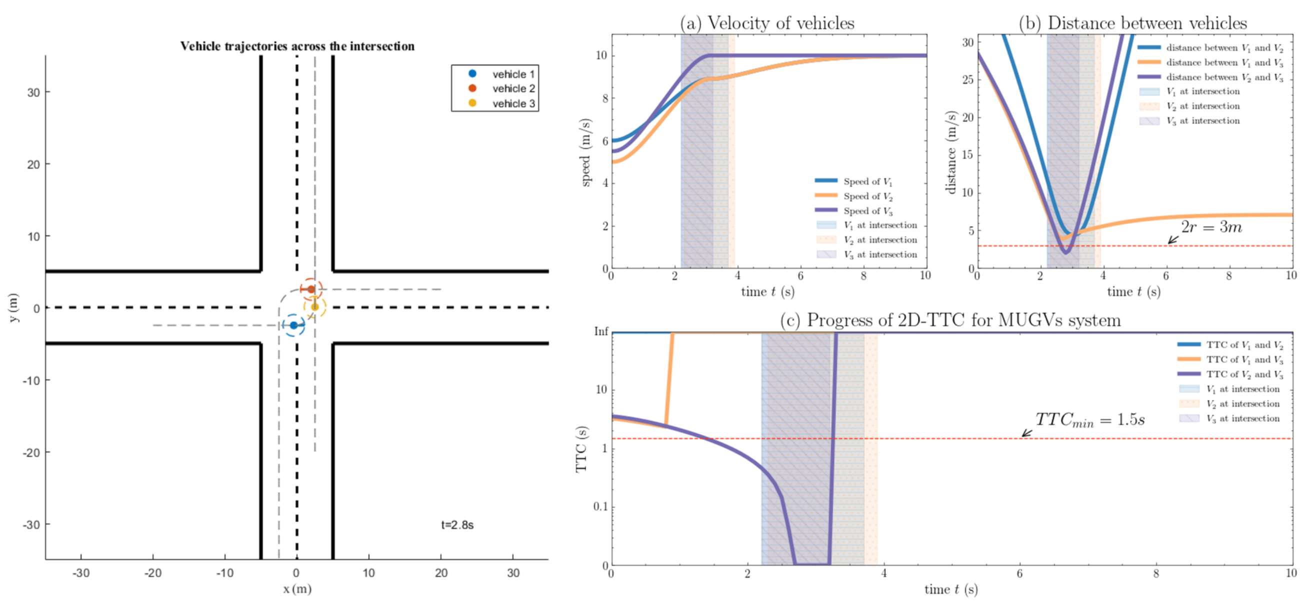
5.2.1 Effect of sampled ε for cooperative navigation

The main parameters in our proposed algorithm are presented in Table 2:

To evaluate the proposed method, contract experiments between original PC and ε-PC are given in the simulation. Three vehicles cooperative navigation at an intersection with predefined trajectories which include left-turn maneuvers for vehicle 1 and vehicle 2 (as shown in Figs. 7a, 8a). One of the important property highlighted in the simulation is the safety of the proposed navigation strategy and the ability to avoid collisions. In original PC, the cost function considered by MUGVs system includes the average crossing time (altruistic objective) and the separation distance as shown in Eq. (1). The simulation results are illustrated in Fig. 7b. Because the control effort has been focused on the crossing time ($W_{cross} > W_{sep}$), the original PC method attempted to maintain a fast crossing speed. Thus, there is a

Table 2 Parameters and initial states

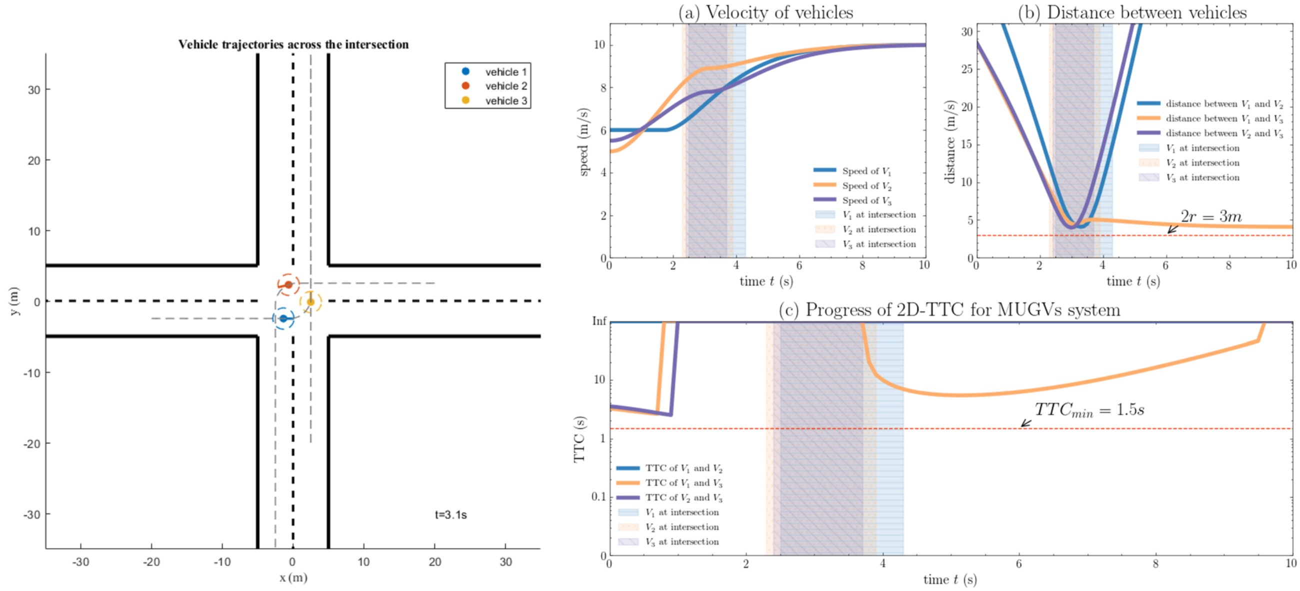
Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
(x_1, y_1)	[−20, −2.5] [m]	v_{min}	0 [m/s]	W_{sep}	1	$N_{samples}$	10
(x_2, y_2)	[20, 2.5] [m]	v_{max}	10 [m/s]	W_{cross}	10	$N_{vehicles}$	3
(x_3, y_3)	[2.5, −20] [m]	r	1.5 [m]	Δ	0.01	TTC_{min}	1.5 [s]
$v_1(0)$	6.0 [m/s]	T_{sample}	0.2 [s]	ε_1	1.5	$\sigma(\varepsilon_0)$	0%
$v_2(0)$	5.0 [m/s]	T	10	ε_2	1.97	$\sigma(\varepsilon_1)$	33.33%
$v_3(0)$	5.5 [m/s]	t_{act}	3	ε_3	2.43	$\sigma(\varepsilon_2)$	66.66%



(a) Critical situation with the original PC

(b) Performance indicators: distance, velocity and 2D TTC in MUGVs system by original PC

Fig. 7 Three vehicles navigation by original PC. **a** Critical situation with the original PC. **b** Performance indicators: distance, velocity and 2D TTC in MUGVs system by original PC



(a) Critical situation with the ε-PC

(b) Performance indicators: distance, velocity and 2D TTC in MUGVs system by ε-PC

Fig. 8 Three vehicles navigation by ε-PC. **a** Critical situation with the ε-PC. **b** Performance indicators: distance, velocity and 2D TTC in MUGVs system by ε-PC

low probability but high impact to choose extreme strategy which allows all vehicles to accelerate as shown in Fig. 7b (a). Such speed growth had led to a collision as the distance indicator exhibited in Fig. 7b (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. 7b (c).

In 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. 8b (a). Due to the threshold constraint of $\epsilon \geq TTC = 1.5s$ with respect to Eq. (6), 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. 8b (b), (c).

Several ε-constraint values are carried out to highlight the performance of the proposed method in the previous scenario in Fig. 8a. The approximated maximum $J_{TTC}(\boldsymbol{Y}^*_{sup}) = 2.8975$ is fixed in predefined MUGVs system by initially heuristic searching (cf. Sect. 4.2.1). Here, the reservation value $J_{TTC}(\boldsymbol{Y}^*_{inf}) = 1.5$ for a minimum 2D TTC in whole time horizon $[0s, 10s]$. After that, it is used several grids point [cf. Eq. (8)] in the range of $1.5 \leq \varepsilon_m \leq 2.8975$ to get a constraint set ε_i and membership function $\sigma(\varepsilon_i)$ [cf. Eq. (9)] as presented in Table 3.

A comparison of average intersection crossing time (in presence of three vehicles) during four trails is shown in Table 3. 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 as expected a lowest performances of 2D TTC. 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 trial 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.

Table 3 Performance comparison of different ε_i

	ε_i	$\sigma(\varepsilon_i)$	Average crossing time	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	1.97	33.33%	4.70 [s]	21	2.07 [s]
Trial 4	2.43	66.66%	4.93 [s]	26	2.51 [s]

5.2.2 Effect of strategy sample size for executing time

To enable the qualified strategy to fulfill certain specific situations in MUGVs navigation (the long tail challenge), it is important to reserve enough strategies in searching space. In order to have a clear picture of the computational demanding under such applications, the execution time of varied number of strategies for MUGVs system is compared in the Fig. 9.

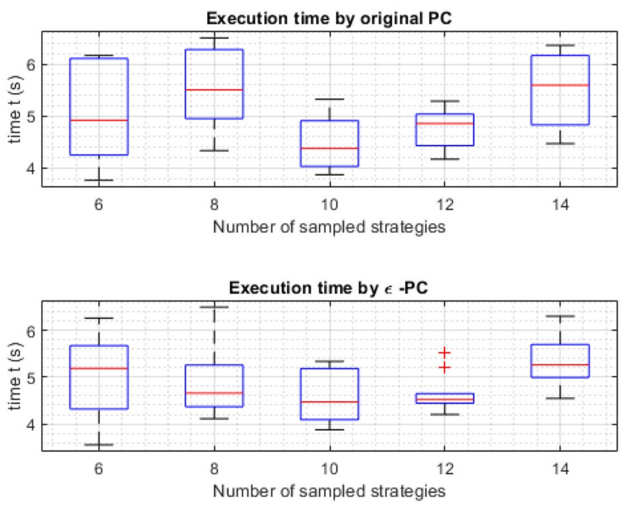
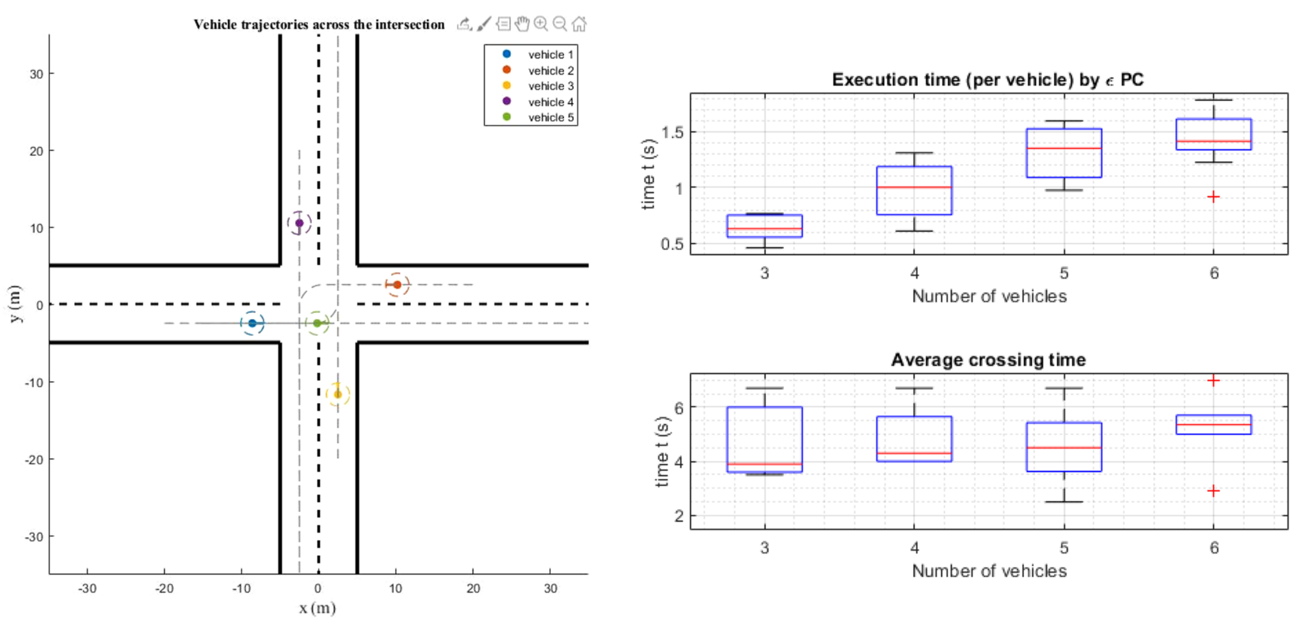


Fig. 9 Effect of strategy number on the execution time

The results in Fig. 9 show a slight difference in terms of the execution time between the original PC and ε-PC. More importantly and within this small decrease in the execution time, the ε-PC outperforms the original method by its capacity in avoiding the potential collision. Based on the results in Fig. 9, it can be admitted that the ε-PC method guarantees the no collision between vehicles without any additional computational burden to satisfy the minimum safety navigation requests. As a reminder, minimum 2D $TTC = 1.5s$ is specifically considered in all these simulations of ε-PC. Furthermore, it appears that setting $N_{samples} = 10$ improves the satisfaction of the computationally demanding. To conclude, over-or under-estimated static sampling strategy may lead to more execution time to find optimal results that can meet the requirements.

5.2.3 Scalability properties

The complexity of the ε-PC algorithm is strongly correlated to the scalability properties of the proposed method. So that we attempt to run ε-PC algorithm with an increasing number of vehicles. The randomized initial parameters (position and speed) are within the same range. Vehicles are generated with different distance to the intersection. The result has further explore the limiting number of vehicles for the application of ε-PC in more exhaustive navigation states (see Fig. 10).



(a) Critical situation with ε-PC (simulation video https://bit.ly/2PoYfjQ)

(b) Effect of vehicle number on the performance of ε-PC

Fig. 10 Effect of vehicle number on the performance of ϵ-PC. **a** Critical situation with ϵ-PC (simulation video https://bit.ly/2PoYfjQ). **b** Effect of vehicle number on the performance of ϵ-PC

Figure 10 shows results for $N_{vehicles} = 3$ to $N_{vehicles} = 6$. The execution time of per vehicle show a steady increase for both the number of vehicle and the average crossing time. The $\varepsilon = 1.5s$ is fixed in each simulations. This indicates that ε-PC can guarantee a safe navigation with increasing number in MUGVs system. A further analysis of complexity will be carried out in further works, and more realistic execution time will be measured after an optimization of the produced code and its parallelization.

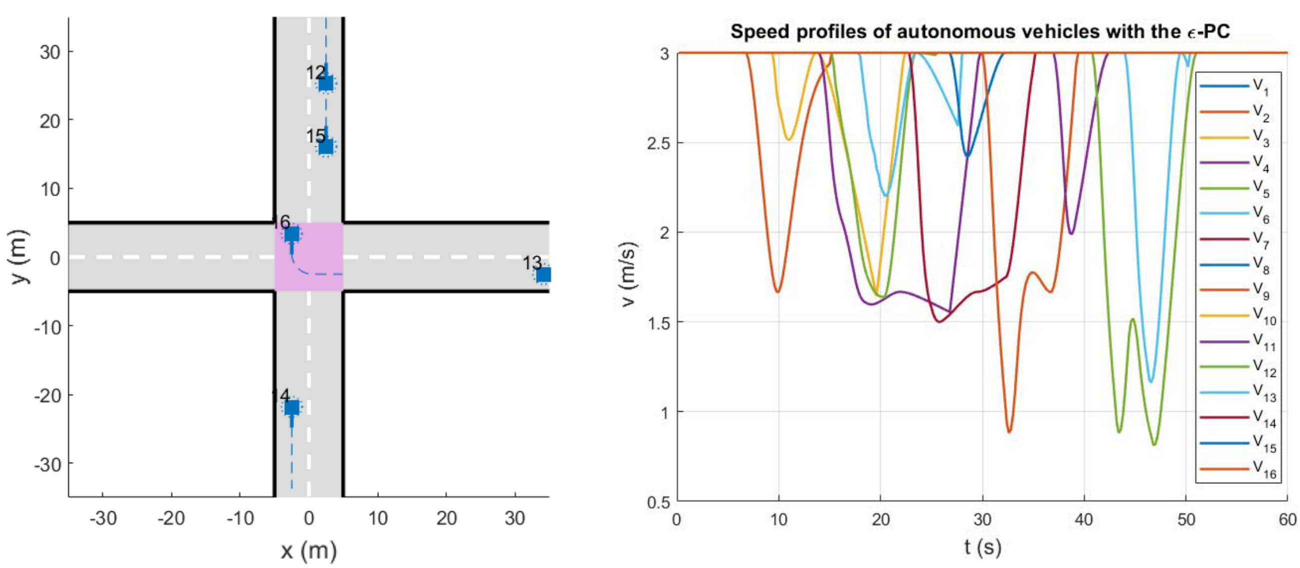
5.3 Real-time ε-PC

In the case of the "single mode", all the vehicles are non-collaborative. This mode is useful to demonstrate the capabilities of the algorithm to deal with a high number of non-collaborative connected vehicles. It is demonstrated also that the use of the algorithm in the special case of single vehicle optimizes the navigation performances. On the other hand, the "full" mode shows the exact opposite as it forces all the vehicles to run the optimization when triggered by a predefined event. This mode has been used in the previous section when an optimization is done on a fixed initial situation with all the vehicles. It should be noted that "single mode" has been preferred for a fast decision making. While "full mode" is more preferred for collaboration in complex environment.

In the tackled simulations, no conflicts did happened between any of the present vehicles, even though they did the optimization one by one. Fig. 11 illustrates the continuous traffic situation during the simulation.

6 Conclusion

This paper proposed a distributed optimal approach for dynamic MUGVs navigation system with risk-sensitive management strategy. A distributed architecture of MUGVs system has been formulated. Then, all the vehicles run iteratively a real-time on-board ε-PC algorithm to find a feasible solution for cooperative navigation in a shared area. The proposed formulation uses the PC theory, which is a promising method, to optimize the distributed problem. Further, a combinatorial optimization problem is applied under the context of convex probability searching space. To this end, vehicles in the system only need to update the probability distribution of actions set instead of searching best combined strategy in a non-linear objective function. Our method allows to ensure a competitive solving time ($0.2s \sim 0.8s$) by approximating optimal solution in real-life application. Furthermore, in real-time traffic navigation, the proposed approach must have reliable behaviors to afford the demand of RAMS system in order to deal with sudden road hazards and risky situations. Therefore, a key safety indicator 2D TTC was adopted as a risk assessment measure to impact vehicle's decision. We model the 2D 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 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



(a) Real time scenario for MUGVs

(b) Real time speed with the ε-PC

Fig. 11 MUGVs navigation in real-time (simulation video https://bit.ly/2PoYfjQ). **a** Real time scenario for MUGVs. **b** Real time speed with the ε-PC tracking, maneuvers warning/assisting or directly as feedback control law in MPC. Thus, this paper designed two modes (“single” vehicle optimization or “full” optimization) for real-time ε-PC operation.

The proposed ε-PC rebalanced between the high-quality strategy and acceptable computational speed. In general, this method is flexible and may obtain sub-optimal solutions rather than global optimal results. 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. Robustness of the overall ε-PC framework for real-time traffic management was demonstrated via Matlab simulations. For future work, vehicle strategies will be calculated in highly uncertain environments (for instance in presence of human drivers). The method of optimization for faster execution needs to be further considered w.r.t. the vehicles’ communication/perception capabilities.

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

Adouane, L.: Autonomous Vehicle Navigation: From Behavioral to Hybrid Multi-Controller Architectures. CRC Press, USA (2016)

Archer, J.: Indicators for traffic safety assessment and prediction and their application in micro-simulation modelling: a study of urban and suburban intersections. Ph.D. thesis, Kungliga Tekniska högskolan (2005)

Ben Lakhal, N.M., Adouane, L., Nasri, O., Hadj Slama, J.B.: Risk management for intelligent vehicles based on interval analysis of ttc. In: 10th IFAC Symposium on Intelligent Autonomous Vehicles IAV 2019: Gdansk, Poland. IFAC-PapersOnLine **52**(8), 338–343 (2019). 3–5 July 2019

Ben Lakhal, N.M., Nasri, O., Adouane, L., Slama, J.: 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, pp. 545–552. INSTICC, SciTePress, Paris-France (2020)

Berbeglia, G., Cordeau, J.F., Laporte, G.: Dynamic pickup and delivery problems. Eur. J. Oper. Res. **202**(1), 8–15 (2010)

Bertolazzi, E., Biral, F., Da Lio, M.: Real-time motion planning for multibody systems. Multibody Syst. Dyn. **17**(2–3), 119–139 (2007)

Bieniawski, S.R.: Distributed optimization and flight control using collectives. Ph.D. thesis, Stanford University (2005)

Chen, L., Englund, C.: Cooperative intersection management: a survey. IEEE Trans. Intell. Transport. Syst. **17**(2), 570–586 (2015)

Chen, C., Xiang, H., Qiu, T., Wang, C., Zhou, Y., Chang, V.: A rear-end collision prediction scheme based on deep learning in the internet of vehicles. J. Parallel Distrib. Comput. **117**, 192–204 (2018)

Chin, H.C., Quek, S.T.: Measurement of traffic conflicts. Saf. Sci. **26**(3), 169–185 (1997)

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

Cordeau, J.F., Laporte, G.: The dial-a-ride problem: models and algorithms. Ann. Opera. Res. **153**(1), 29–46 (2007)

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

Gregoire, J., Qian, X., Frazzoli, E., De La Fortelle, A., Wongpiromsarn, T.: Capacity-aware backpressure traffic signal control. IEEE Trans. Control Netw. Syst. **2**(2), 164–173 (2014)

Guo, Q., Li, L., Ban, X.J.: Urban traffic signal control with connected and automated vehicles: a survey. Transport. Res. Part C Emerg. Technol. **101**, 313–334 (2019)

Haimes, Y.: On a bicriterion formulation of the problems of integrated system identification and system optimization. IEEE Trans. Syst. Man Cybernet. **1**(3), 296–297 (1971)

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

Hillenbrand, J., Spieker, A.M., Kroschel, K.: A multilevel collision mitigation approach-its situation assessment, decision making, and performance tradeoffs. IEEE Trans. Intell. Transport. Syst. **7**(4), 528–540 (2006)

Horst, R.: Time-to-collision as a cue for decision-making in braking. Vision in Vehicles–III (1991)

Huang, C.F., Bieniawski, S., Wolpert, D.H., Strauss, C.E.: 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, pp. 751–752 (2005)

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

Hydén, C.: The development of a method for traffic safety evaluation: the swedish traffic conflicts technique. Bull. Lund Inst. Technol. Depart. **70** (1987)

Hyland, M., Mahmassani, H.S.: Dynamic autonomous vehicle fleet operations: optimization-based strategies to assign avs to immediate traveler demand requests. Transport. Res. Part C Emerg. Technol. **92**, 278–297 (2018)

Iberraken, D., Adouane, L., Denis, D.: 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), pp. 3984–3990. IEEE (2018)

Jansen, K.: Approximation algorithms for the general max-min resource sharing problem: faster and simpler. In: Scandinavian Workshop on Algorithm Theory, pp. 311–322. Springer (2004)

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

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

Kulkarni, A.J., Tai, K.: Solving constrained optimization problems using probability collectives and a penalty function approach. Int. J. Comput. Intell. Appl. **10**(04), 445–470 (2011)

Kulkarni, A.J., Patankar, N.S., Tai, K.: Constraint handling in probability collectives using a modified feasibility-based rule. Int. J. Comput. Sci. Eng. **13**(4), 303–321 (2016)

Lord, D., Persaud, B.N.: Estimating the safety performance of urban road transportation networks. Accident Anal. Prev. **36**(4), 609–620 (2004)

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

Mavrotas, G.: Effective implementation of the ε-constraint method in multi-objective mathematical programming problems. Appl. Math. Comput. **213**(2), 455–465 (2009)

Miettinen, K.: Nonlinear Multiobjective Optimization, vol. 12. Springer Science & Business Media, Germany (2012)

Milanés, V., Villagrá, J., Godoy, J., Simó, J., Pérez, J., Onieva, E.: An intelligent v2i-based traffic management system. IEEE Trans. Intell. Transport. Syst. **13**(1), 49–58 (2012)

Nasri, O., Ben Lakhal, N.M., Adouane, L., Ben Hadj Slama, J.: Automotive decentralized diagnosis based on can real-time analysis. J. Syst. Arch. **98**, 249–258 (2019). https://doi.org/10.1016/j.sysarc.2019.01.009

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

Schwarting, W., Pierson, A., Alonso-Mora, J., Karaman, S., Rus, D.: Social behavior for autonomous vehicles. Proc. Nat. Acad. Sci. **116**(50), 24972–24978 (2019)

Sislak, D., Volf, P., Pechoucek, M., Suri, N.: Automated conflict resolution utilizing probability collectives optimizer. IEEE Trans. Syst. Man Cybernet. Part C (Appl. Rev.) **41**(3), 365–375 (2010)

Suzuki, H., Marumo, Y.: 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), pp. 362–367. IEEE (2018)

Ward, J., Agamennoni, G., Worrall, S., Nebot, E.: Vehicle collision probability calculation for general traffic scenarios under uncertainty. In: 2014 IEEE Intelligent Vehicles Symposium Proceedings, pp. 986–992. IEEE (2014)

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

Yang, B., Wu, R.: A modified probability collectives optimization algorithm based on trust region method and a new temperature annealing schedule. Soft Comput. **20**(4), 1581–1600 (2016)

Zhu., Z., Adouane., L., Quilliot., A.: A decentralized multi-criteria optimization algorithm for multi-unmanned ground vehicles (mugvs) navigation at signal-free intersection. In: The 16th IFAC Symposium on Control in Transportation Systems (CTS 2021). Lille-France (2021)

Zhengze Zhu is carrying his research work on intelligent vehicle control in a joint Ph.D. program between the Hubei University of Automotive Technology (China) and Clermont Auvergne University (France). He received his B.S. degree in 2011 from Southwest Jiaotong University (SWJTU), China, and his M.S. degree from Hubei University of Automotive Technology (HUAT) in 2018, China. He focalizes his research on vehicle safe and autonomous navigation, optimal control, and the application of vehicle dynamic risk management in Intelligent Transportation Systems.

Lounis Adouane is a full Professor at Université de Technologie de Compiègne (UTC, France). He received the Ph.D. degree in automatic control from the FEMTO-ST Laboratory, ENSMM, in 2005. During his Ph.D. studies, he deeply investigated the field of mobile multi-robot systems; especially those related to bottom-up and decentralized control architectures. In 2005, he joined the Ampere Laboratory, INSA Lyon, and he studied hybrid (continuous/discrete) control architectures applied to cooperative mobile robots arms. In 2015, he obtained an HDR (habilitation to steer research in robotics) from Blaise Pascal University. From 2006 to 2019, he was an Associate Professor at the Institut Pascal, Polytech Clermont-Ferrand. Currently, Lounis Adouane hold a full Professor position at Heudiasyc, UMR 7253 CNRS / UTC. Between 2018 and 2019, he was the head of ISPR Group (Images, Perception Systems and Robotics, 80 members) at Institut Pascal. Since 2018, Prof. Lounis Adouane is a member of the Technical Committee of IFAC–Intelligent Autonomous Vehicles and serves in the Editorial Board of the Journal of Intelligent and Robotic Systems. Prof. Lounis Adouane is the author/coauthor of more than 120 refereed international papers and two books. His main research interest include planning and control of mobile robots / intelligent vehicles (autonomous and clean), decision-making under uncertainty, hybrid (continuous/discrete) and hybrid (reactive/cognitive) multi-controller architectures, Lyapunov based synthesis and stability, cooperative multi-robot systems, artificial intelligence (e.g., MDP or MAS approaches), energy management (e.g., optimal control or neuro-fuzzy approaches).

Alain Quilliot (born on the 25/10/1953), former student of ENS Saint-Cloud (1973-77), « Agrégé de Mathématiques », and Ph.D. graduate from PARIS VI University (1978, Dir. C.BERGE). Professor Alain QUILLIOT started his academic career at UNAM University of MEXICO (179-81) as Cooperating Professor, and next (1981-82), at South Carolina University in COLUMBIA (USA) as Invited Professor. He next got a permanent position at ENSIMAG GRENOBLE and, at the same time, defended his Thèse d'Etat in June 1983 in PARIS VI. After he got in 1987 a Full Professor position at Blaise Pascal University in CLERMONT-FERRAND, he created a Master/Engineering department in Mathematics and Modelling inside the CUST/POLYTECH Institute, which he managed from 1989 until 1994. In 1993, he simultaneously created the Engineering Institute ISIMA, member of the Concours Communs Polytechniques (ex ENSI, CPGE) Network, and the inter-university research laboratory LIMOS (Informatics, Modelling and System Optimization: currently 85 permanent searchers, 94 Ph.D., 12 associated members, and 7 technical staff), which is a mixed CNRS/University Laboratory and which got A+ AERES evaluation in 2006 and A evaluation in 2011. He led ISIMA from 1993 until 2007, and LIMOS from 1994 until 2014. He also was, in 2003, a founding member of the TIMS CNRS Federation (Technologies of Information, Mobility, and Safety), which was made of 6 research groups in Computing, Mechanics, Robotics, which he led from 2007 until the end of 2011, and which gave rise to the newly labeled LABEX structure IMOBS3 (Intelligent and Innovative Mobility). He is currently the head of the CNRS network GDR 3002: Operations Research, which gathers 500 permanent searchers. Deeply involved in international and industrial partnerships, Professor QUILLIOT was also currently Chairman of the International Affairs Office (M.R.I) of CLERMONT-FERRAND town between 2006 and 2015, and vice-chairman of AUVERGNE-TIC cluster, which gathered from 2007 until 2015 regional I.T companies and academic players. He is still Director of the CNRS French-Chinese Joint Laboratory (LIA) Smart Computing for Sustainable Growth. Its personal contributions as a scientist include the supervision of 30 Ph.D. and 9 HDR (Research Supervision Agreement) thesis, the redaction of more than 200 publications (80 in international or national journals and 6 book chapters), and several direct or indirect partnership agreements with companies like EDF, THALES, MICHELIN, France Telecom, SNCF.