



Optimal cooperative motion planning for vehicles at intersections

Jean Gregoire, Silvère Bonnabel, Arnaud de la Fortelle

► **To cite this version:**

Jean Gregoire, Silvère Bonnabel, Arnaud de la Fortelle. Optimal cooperative motion planning for vehicles at intersections. 2012. hal-00877765

HAL Id: hal-00877765

<https://hal.archives-ouvertes.fr/hal-00877765>

Preprint submitted on 29 Oct 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Optimal cooperative motion planning for vehicles at intersections

Jean Grégoire*

Silvère Bonnabel*

Arnaud de La Fortelle*†

Abstract— We consider the problem of cooperative intersection management. It arises in automated transportation systems for people or goods but also in multi-robots environment. Therefore many solutions have been proposed to avoid collisions. The main problem is to determine collision-free but also deadlock-free and optimal algorithms. Even with a simple definition of optimality, finding a global optimum is a problem of high complexity, especially for open systems involving a large and varying number of vehicles. This paper advocates the use of a mathematical framework based on a decomposition of the problem into a continuous optimization part and a scheduling problem. The paper emphasizes connections between the usual notion of vehicle priority and an abstract formulation of the scheduling problem in the coordination space. A constructive locally optimal algorithm is proposed. More generally, this work opens up for new computationally efficient cooperative motion planning algorithms.

I. INTRODUCTION

In this paper, we consider the problem of designing an intelligent transportation system meant to improve the safety at intersections, and to decrease the average time spent by vehicles going through the intersection. Due to the achievements and promises in autonomous cars design, this subject has attracted much interest, and a variety of systems have been proposed to address this issue [4], [8], [16], [6], [3]. The clear improvements in the intersection flow compared to the use of traffic lights [4], [11], and in terms of safety [5], have become strong drivers for research. Moreover, conflict resolution is also an intensive field of research in railway and air transportation systems for instance [9], [18].

As the task has a potentially very high complexity (essentially one tries to compactify the vehicles trajectories in space-time), the problem is often decomposed into two parts. The first part consists of determining fixed paths along which vehicles cross the intersection. The second part consists of adapting the velocity of each vehicle along the path in order to avoid collisions — generally resulting from human mistakes — and to optimize as well traffic by increasing flows, i.e. by decreasing the average time spent in the intersection. This approach was initiated in [10] for multi-robot applications, and has become standard for vehicle management at intersections, e.g., [6], [8], [14], [1].

The problem of planning velocities of robots along fixed paths has attracted much attention over the past decades, and was essentially motivated by applications for robot manipulators and automated guided vehicles in factories. A usual approach in motion planning consists of transferring a

planning problem in the physical space to a planning problem in an abstract space: the configuration space. For the problem of planning velocities along fixed paths, the configuration space is called coordination space, first introduced in [17] for robot manipulators, and possesses a specific cylindrical structure, as first noticed by [13]. In this paper, we propose to use the coordination space approach as a rigorous mathematical framework particularly well-suited to the problem of motion planning at intersections. We suppose the paths of the intersection fixed and we only focus on the coordination problem.

The process is then inherently cooperative: some vehicles need to brake to let some others go ahead until the conflict configuration is exited. The purpose of the present paper is to cast the problem in a well-defined mathematical framework, and to provide a provably collision-free solution that minimizes the average exit time over the vehicles. To do so we revisit the notion of priority introduced in [2] and [6]. It appears to be especially meaningful for the problem considered as we recover the usual and intuitive notion of vehicle priority at intersections. We show it is a powerful tool to classify the locally optimal trajectories that have been studied as special representatives of homotopy classes in [7]. We introduce a priority graph and prove several properties of this graph with respect to our problem. We use it to build an algorithm for the design of collision-free and deadlock-free optimal trajectories.

The paper has two main contributions. From an applicative viewpoint, it shows the coordination space is a powerful mathematical tool that is especially suited to the problem of coordinating vehicles at intersections. In particular, it opens a new research avenue for the design of efficient algorithms. From a theoretical viewpoint, our contribution consists of drawing links between a discrete scheduling problem (priority graph) and a continuous problem formulated in the abstract coordination space. It illustrates those links in an intuitive way and relates those notions to the existence of collisions or deadlocks and to the optimality problem. New results are derived (see in particular Propositions 7 and 9) and a constructive optimal algorithm is built upon the proposed methodology.

Section II formulates the problem and introduces the fixed-path approach. Section III defines a priority relation between vehicles underlying the scheduling problem and presents some new results. In section IV we present algorithms for building an optimal motion planning based on schedule decoupling.

* Mines ParisTech, Centre de Robotique, Mathématiques et Systèmes, 60 Bd St Michel 75272 Paris Cedex 06, France

† Inria Paris - Rocquencourt, IMARA team, Domaine de Voluceau - Rocquencourt, B.P. 105 - 78153 Le Chesnay, France

II. PROBLEM FORMULATION

A. Modeling assumptions

The main assumption we make is to constrain the vehicles to follow predefined paths to go through the intersection. This assumption fits well intersections in a road network which are a highly constrained environment. Thus, every vehicle i follows a particular path γ_i and we denote $s_i \in \mathbb{R}$ its curvilinear coordinate on the path. The configuration of the system of vehicles is $s = (s_i)_{i \in \{1 \dots n\}}$ and we denote $\phi(t)$ the evolution of s through time. The curvilinear coordinates are normalized and we let $\chi = [0, 1]^n$ be the coordination space, where n is the number of vehicles going through the intersection (possibly changing through time) and $\{e_i\}_{1 \leq i \leq n}$ the canonical basis of χ . The boundedness condition on χ is rather technical but ensures the whole intersection lies in a bounded region of \mathbb{R}^n (somehow interactions are limited to a bounded area).

To perform motion planning optimization, we need to make hypothesis on the technical constraints of the vehicles. These include kinematic constraints (maximum velocity, maximum curve radius, etc.) and dynamic constraints (limited acceleration, adherence, jerk, etc.). In this paper, as a first step, we assume that the velocity of every vehicle on its path is limited to $v_{max} = 1$ without accounting for dynamic constraints. Moreover we assume the velocities are always positive (the cars do not go in reverse in the intersection). These assumptions are standard [6].

As every vehicle occupies a non-empty geometric region, some states must be excluded to avoid collisions between vehicles. We define the obstacle region χ_{obs} as the open set of all collision states. A collision occurs when two vehicles occupy a same region of space, so that the obstacle region can be described as the union of $n(n-1)/2$ open cylinders χ_{obs}^{ij} corresponding to as many collision pairs: $\chi_{obs} = \bigcup_{i>j} \chi_{obs}^{ij}$ [12]. Indeed obviously $\chi_{obs}^{ij} = \chi_{obs}^{ji}$ and $\chi_{obs}^{ii} = \emptyset$. Figure 2 displays the obstacle region and a collision-free path for a two-path intersection.

Moreover, we assume that each cylinder χ_{obs}^{ij} has an open bounded convex cross-section (in the plane generated by e_i and e_j). This assumption implies cross-sections are simply connected and excludes some cases of real intersections such as a pair of opposite turn-left, but the results we present in this paper could be easily extended with a more relaxed assumption. Cross-sections are open so that the complementary set is closed and hence complete. It also ensures all cross-sections are included in the interior of $[0, 1]^2$: no collision can occur for a vehicle at coordinates 0 or 1. Such assumptions are very realistic, except for the case of vehicles following each other on a same path, but our framework can be extended to this case, as we plan to prove in future work.

B. Formulation in the coordination space

Let $\chi_{free} = \chi \setminus \chi_{obs}$ denote the obstacle-free space and $\mathbf{1} = (1 \dots 1)$. The initial condition x_{init} belongs to χ_{free} , and the goal region is $\chi_{goal} = \{\mathbf{1}\} \subset \chi_{free}$. In this paper we will consider continuous paths $\phi : [0, T] \rightarrow \chi$ which are

piecewise right differentiable. A collision-free path will be such that $\phi(t) \in \chi_{free}$ for all t .

Definition 1 (Feasible paths). *A feasible path for the considered problem is a right-differentiable collision-free path such that*

- $\phi(0) = x_{init}$ and $\phi(T) \in \chi_{goal}$
- $\phi'(t) \geq 0$ and $\|\phi'(t)\|_\infty \leq 1 \quad \forall t \in [0, T]$.

In order to define an optimality criterion for the motion planning problem, we need to define a cost function $c(\phi)$ enabling to compare feasible paths. As explained in the introduction, the purpose is to minimize the average exit time over the vehicles. As a consequence, we define the cost function c as:

$$c(\phi) = \frac{1}{n} \sum_{i=1}^n T_i = \frac{1}{n} \sum_{i=1}^n \phi_i^{-1}(1)$$

with $\phi_i^{-1}(s_i)$ denoting the first date at which ϕ_i reaches s_i and $T_i = \phi_i^{-1}(1)$ being the exit time for the vehicle i .

The optimality problem consists of finding a feasible path ϕ^* that minimizes the cost, i.e. for any feasible path ϕ , we have $c(\phi^*) \leq c(\phi)$. Figure 1 depicts three feasible trajectories with different costs in a two-dimensional coordination space for a given initial condition.

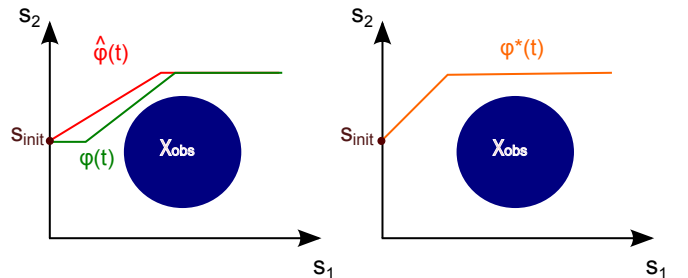


Fig. 1. The left drawing depicts two feasible trajectories in a two-dimensional coordination space. With the trajectory $\hat{\phi}$, the vehicle 2 leaves the intersection earlier than with ϕ , while exit time is not changed for the vehicle 1. Hence, $c(\hat{\phi}) < c(\phi)$. ϕ^* is an optimal trajectory since every vehicle is always at maximum speed: $\phi^* = \arg \min c(\phi)$.

Note that the cost can be easily bounded from above and from below. Suppose for simplicity's sake that $x_{init} = 0$ and $\chi_{goal} = \{\mathbf{1}\}$. If $\chi_{obs} = \emptyset$ the optimal path is given by $\phi' \equiv \mathbf{1}$. The corresponding cost $c(\phi) = \frac{1}{n} \sum_{i=1}^n 1 = 1$ is thus a lower bound on the optimal time in the general case. On the other hand, a suboptimal trajectory is easily found moving the vehicles one after the other, i.e. moving along the edges of the hypercube. This trajectory is collision-free, because by assumption a vehicle cannot collide another one if its coordinate is 0 or 1. It yields $\bar{c}(\phi) = \frac{1}{n} \sum_{i=1}^n i = \frac{n+1}{2}$.

Note that vehicle i is free after T_i when its coordinate is 1: there is no further interaction with any other vehicle. We hereafter take $\phi_i(t) = 1$ for $t \geq T_i$ in order to stay in $\chi = [0, 1]^n$ but any further trajectory $\phi_i(t) \geq 1$ would be suitable had we taken $\chi = \mathbb{R}^n$.

C. Geometry of the coordination space

In this section, we propose several geometric concepts in the coordination diagram.

1) *Completing the obstacle region*: For each collision pair (i, j) we define the south of χ_{obs} along the plane $\mathbf{e}_i \times \mathbf{e}_j$ as:

$$\begin{aligned} S_{ij}(\chi_{\text{obs}}) &= \chi_{\text{obs}}^{ij} - \mathbb{R}_+ \mathbf{e}_j \\ &= \left\{ s - \lambda \mathbf{e}_j : \forall s \in \chi_{\text{obs}}^{ij}, \forall \lambda \in \mathbb{R}_+ \right\} \end{aligned} \quad (1)$$

We say the south of χ_{obs}^{ij} along the plane $\mathbf{e}_i \times \mathbf{e}_j$ is the west of χ_{obs}^{ij} along the plane $\mathbf{e}_j \times \mathbf{e}_i$ and denote $W_{ij}(\chi_{\text{obs}}) = S_{ji}(\chi_{\text{obs}})$. These sets are clearly open. They play the role of gates: a feasible trajectory must go through one or the other and they are exclusive. As a result, the intersection of the south and the west is forbidden for feasible trajectories, even if there is no collision. As the velocities must be positive, being in the intersection at some time would necessarily lead to a future collision unless the cars stop moving (see Figure 3). This situation is referred to as a deadlock. To avoid such situations, an extended obstacle region is defined, so that no basic one-to-one dead-lock situation can occur [17].

Definition 2 (SW-Completion). *The South-West completion of χ_{obs} along the plane $\mathbf{e}_i \times \mathbf{e}_j$ is given by*

$$SW_{ij}(\chi_{\text{obs}}) = S_{ij}(\chi_{\text{obs}}) \cap W_{ij}(\chi_{\text{obs}})$$

where $S_{ij}(\chi_{\text{obs}})$ is defined in (1). The completed obstacle region $\overline{\chi_{\text{obs}}}$ is defined by

$$\overline{\chi_{\text{obs}}} = \bigcup_{i \neq j} SW_{ij}(\chi_{\text{obs}})$$

We call $\overline{\chi_{\text{free}}} = \chi \setminus \overline{\chi_{\text{obs}}}$ the obstacle-free completed region.

The South-West closure is applied on each cylinder χ_{obs}^{ij} as depicted on Figure 3. An important fact — not explicitly mentioned in the literature yet — is that this method cannot be readily extended to avoid dead-locks in multi-dimensional planning problems while ensuring optimality. For this reason, the next sections rather present tools based on gates' positions to avoid dead-locks caused by multiple vehicles' interaction.

2) *Gates in the coordination space*: Let define the region $H_{i>j}$ as follows:

$$H_{i>j} = S_{ij}(\chi_{\text{obs}}) \setminus SW_{ij}(\chi_{\text{obs}})$$

The top drawings of Figure 2 display $H_{i>j}$ and $H_{j>i}$. Over the convexity hypothesis of the cylinders of χ_{obs} , we can assert that any feasible trajectory will necessarily cross $H_{i>j}$ or $H_{j>i}$ exclusively. This is why these two boxes can be viewed as gates in the coordination space. In Figure 2, the path $\phi_{i>j}$ intersects the gate $H_{i>j}$: the vehicle j slows down to let the vehicle i go ahead. Each gate favours a vehicle over another i.e. a priority is assigned, and passing through a gate possibly prevents from passing through many other gates because of the positive speed assumption.

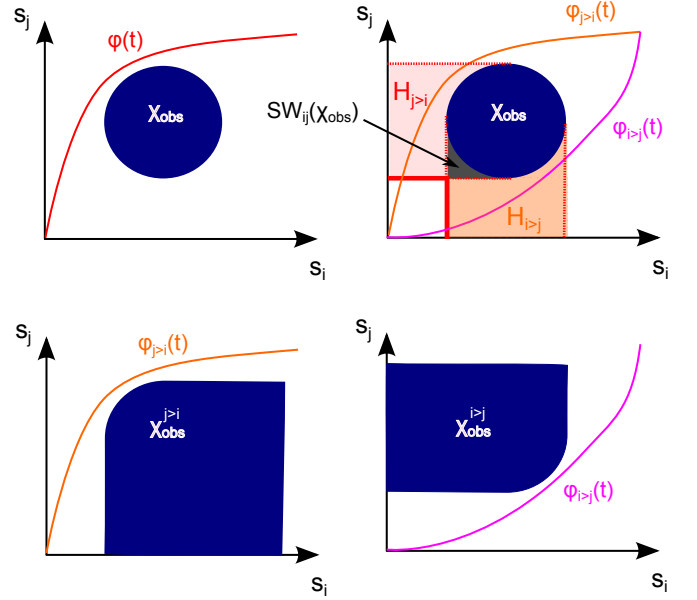


Fig. 2. The top-left drawing represents in the plane (s_i, s_j) the obstacle region χ_{obs} and a feasible trajectory ϕ . The top-right drawing visualizes the two gates $H_{i>j}$ and $H_{j>i}$. For each gate, a feasible trajectory passing through the gate is given. The bottom drawings depict $\chi_{\text{obs}}^{i>j}$ and $\chi_{\text{obs}}^{j>i}$.

We propose to define the obstacle region $\chi_{\text{obs}}^{i>j}$ as a set that is incompatible with the crossing of gate $H_{i>j}$, as illustrated on Figure 2:

$$\chi_{\text{obs}}^{i>j} = \chi_{\text{obs}}^{ij} - \mathbb{R}_+ \mathbf{e}_i + \mathbb{R}_+ \mathbf{e}_j$$

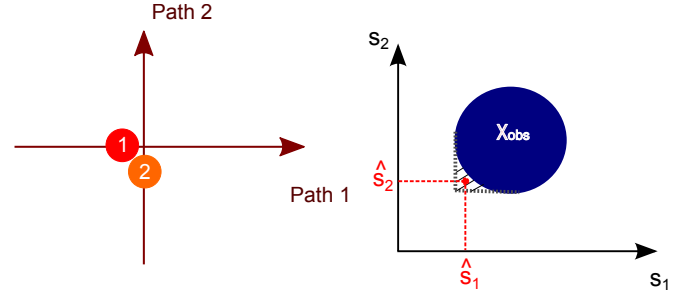


Fig. 3. Two representations of a dead-lock situation involving two circle-shaped vehicles. The left drawing depicts the real intersection with its two orthogonal paths and the two circle-shaped vehicles. Both of them have moved forward too much, so that now they cannot exit the intersection without turning around. The right drawing represents what happens in the two-dimensional coordination space and highlights the South-West region that must be added to χ_{obs} to avoid the one-to-one dead-lock.

III. PRIORITY RELATIONS AND ORDERING

The geometry of the coordination space leads us to define a natural binary relation corresponding to priority relations between vehicles: a very familiar and intuitive notion in real life. Indeed, we say the vehicle i has priority over the vehicle j in the intersection if the associated path goes through the gate $H_{i>j}$ in the coordination space.

Definition 3 (Priority relation). *A feasible path ϕ induces a binary relation \succ on the set $\{1..n\}$ as follows. For $i \neq j$,*

$i \succ j$ if ϕ intersects $H_{i \succ j}$.

The following proposition proves a result that is not totally obvious from an intuitive viewpoint.

Proposition 4. *The binary relation \succ induced by a feasible trajectory ϕ does not necessarily defines an order.*

Figure 4 depicts a 3-paths intersection with 3 intersection points and as many vehicles going through the intersection. In the right drawing, the red vehicle goes ahead the blue vehicle and the green vehicle goes ahead the red vehicle, while the blue vehicle goes ahead the green vehicle. It yields in a non transitive binary relation: the set $\{1\dots n\}$ is not partially ordered.

A. Priority graph

In this subsection, we propose to define the priority relations in terms of a graph. As for any collision pair i, j we have either $i \succ j$ or $j \succ i$, the priority relation can be defined by an oriented graph G whose vertices are $\{1\dots n\}$.

Definition 5. *Given a feasible trajectory ϕ , we call the priority graph the oriented graph G whose vertices are $\{1\dots n\}$ and such that $i \xrightarrow{G} j$ if and only if $i \succ j$.*

Note the graph is complete in the sense that for all $i \neq j$ there is one, and only one, arc linking i and j . There is no arc linking i to itself. Generally speaking, there are potentially $2^{\frac{n(n-1)}{2}}$ possible priority graphs. Actually, one can easily understand that some priorities are not acceptable. The next section deals with this issue and characterizes what we call feasible priorities.

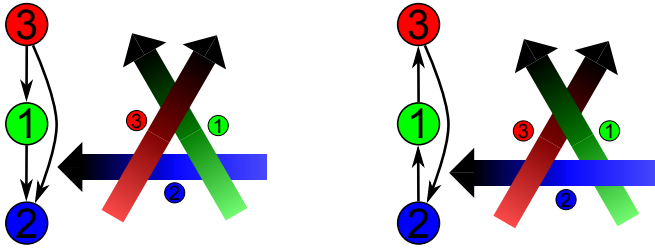


Fig. 4. Two representations of priority relations. In each drawing relation is represented in two ways: as a complete oriented graph, where orientation yields the priority; and as trajectories over time, foreground being first, background later. The left drawing represents a relation that is an order (even a total order). The right drawing shows a relation that is *not* an order.

B. Feasible priorities

Definition 6 (Feasible Priority). *A priority graph G is feasible if there exists a feasible trajectory ϕ whose priority graph is G .*

In the intersection depicted on the left drawing of Figure 5, we can assert that some priorities are not feasible. Indeed, if the the vehicle 2 goes ahead the vehicle 1 and the vehicle 2 goes ahead the vehicle 3, the vehicle 1 must go ahead the vehicle 3, because the three paths have a common intersection point. Hence, for this particular situation, the

priority graph cannot be cyclic and the priority relation is an order relation.

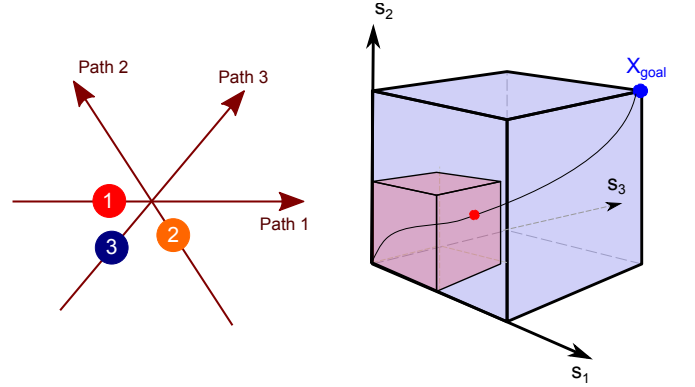


Fig. 5. The left drawing represents a 3-path intersection with a common intersection point and 3 vehicles going through the intersection. The right drawing depicts what happens in the coordination space when cyclic priorities are set. Any feasible trajectory respecting cyclic priorities should stay in the red box $[0, \frac{1}{2}]^3$, but reaching the goal requires to cross this box.

The proposition below provides a necessary condition for priorities feasibility based on the relative position of the fixed-priority collision cylinders $\chi_{obs}^{i \succ j}$. The proof is omitted due to space limitations.

Proposition 7 (Characterization of feasible priority graphs). *A priority graph G is feasible if and only if for any cycle C in G ,*

$$\bigcap_{i \xrightarrow{C} j} \chi_{obs}^{i \succ j} = \emptyset$$

As $\chi_{obs}^{ij} \subset \chi_{obs}^{i \succ j}$ it readily implies:

Proposition 8 (Necessary condition for priorities feasibility). *Suppose G is a feasible graph. Then, for any cycle C in G ,*

$$\bigcap_{i \xrightarrow{C} j} \chi_{obs}^{ij} = \emptyset$$

In the intersection of Figure 5, the state $(s_i = \frac{1}{2})_{i=1,2,3}$ corresponding to a common intersection point is in each of the three collision cylinders. Hence, the necessary condition applies and it can be asserted that no cycle between the 3-vehicle priority graph can exist. The underlying reason in the coordination space is that if there is a cycle in the priorities, then the trajectory must stay in the box $[0, \frac{1}{2}]^3$ at any time to respect these priorities, i.e. cannot reach X_{goal} .

The characterization of feasible priorities shows that the feasibility of a priority graph relies on the relative position of the collision cylinders in the coordination space.

IV. THE MOTION PLANNING PROBLEM

The last section shows that the set of feasible trajectories can be divided into as many disjoint subsets as there are feasible priority graphs. In this section, we shall present an algorithm that builds an optimal trajectory for a fixed priority graph. We will then discuss the computational issues for determining the globally optimal trajectory.

A. Optimal trajectories with fixed priority graph

The main idea is that when priorities are fixed, if at any time every vehicle runs as much distance as possible in a time step, then we can assert that the trajectory is optimal (i.e. minimizes the average time spent in the intersection) for the fixed priorities. It means the optimal solution can be defined reactively.

Let H_G denote the union of gates corresponding to the priorities fixed by G . After setting $\phi(0) = x_{\text{init}}$, the algorithm proceeds as follows. If G is not a feasible priority graph, the algorithm stops, else, until χ_{goal} is reached, at each date t , the trajectory is defined via $\phi'(t)$ which is calculated as a function of $\phi(t)$ by the following algorithm, that essentially consists of projecting the vector $\mathbf{1}$ consecutively:

- $\phi'(t) \leftarrow \mathbf{1}$
- *Stay in χ* : For i such that $\phi_i(t) = 1$, do $\phi'_i(t) \leftarrow 0$
- *Avoid Forbidden Gates*: While there exists $i \xrightarrow{G} j$ such that $\phi(t)$ belongs to the boundary of the gate $H_{j \succ i}$, do: $\phi'(t) \leftarrow$ orthogonal projection of $\phi'(t)$ on the boundary of $H_{j \succ i}$
- *Avoid Obstacle Region*: While there exists $i \xrightarrow{G} j$ such that $\phi(t)$ belongs to the boundary of $\overline{\chi_{\text{obs}}^{ij}}$, if $\mathbf{1}$ points towards $\overline{\chi_{\text{obs}}^{ij}}$, do: $\phi'(t) \leftarrow$ orthogonal projection of $\phi'(t)$ on any tangential hyperplane to $\overline{\chi_{\text{obs}}^{ij}}$ at $\phi(t)$

Proposition 9 (Optimal motion planning for fixed priorities). *Given a feasible priority graph G , the trajectory $\phi^*(t)$ generated by the algorithm described above is an optimal trajectory for the given priorities.*

It can be noted that the vehicles may brake for two reasons: avoiding the obstacle region or avoiding a gate that is not compatible with the priorities defined by G . Figure 6 displays a trajectory generated by the algorithm for a 3-paths intersection. For simplicity's sake, for each collision pair, the one-to-one obstacle region is approximated to a square.

Proof. Suppose there exists a trajectory ψ respecting the priorities G and the initial condition $\psi(0) = \phi^*(0)$ such that ψ_{i_0} overtakes $\phi_{i_0}^*$ for the first time at the date $t_0 > 0$. Consider the trajectory defined by $\phi_i(t) = \max(\psi_i, \phi_i^*)(t)$. The trajectory ϕ admits G as priority graph, it is collision-free (because ψ and ϕ^* respect the same priorities, see Figure 2) and it respects the maximum velocity constraint. Moreover, $\phi_i(t_0) = \psi_i(t_0)$ and for small enough $\epsilon > 0$, $\phi_{i_0}(t_0 + \epsilon) \geq \phi_{i_0}^*(t_0)$. By construction of ϕ^* , for small enough ϵ , $\phi_i(t_0 + \epsilon) \leq \phi_i^*(t_0)$, so that the last two conditions require $\phi_{i_0}(t_0 + \epsilon) = \phi_{i_0}^*(t_0 + \epsilon) = \psi_{i_0}(t_0 + \epsilon)$ to respect both inequalities. As a consequence, for all $i \in \{1 \dots n\}$, the trajectory ϕ_i^* is never strictly overtaken by a vehicle in another feasible trajectory ψ_i : ϕ^* is an optimal trajectory. \square

The optimal solution constructed by the algorithm is known as the "left-greedy" solution [7]. In this latter paper the authors address a more general problem of optimization in coordination space, and regroup the trajectories in equivalence classes based on homotopy (two trajectories in the

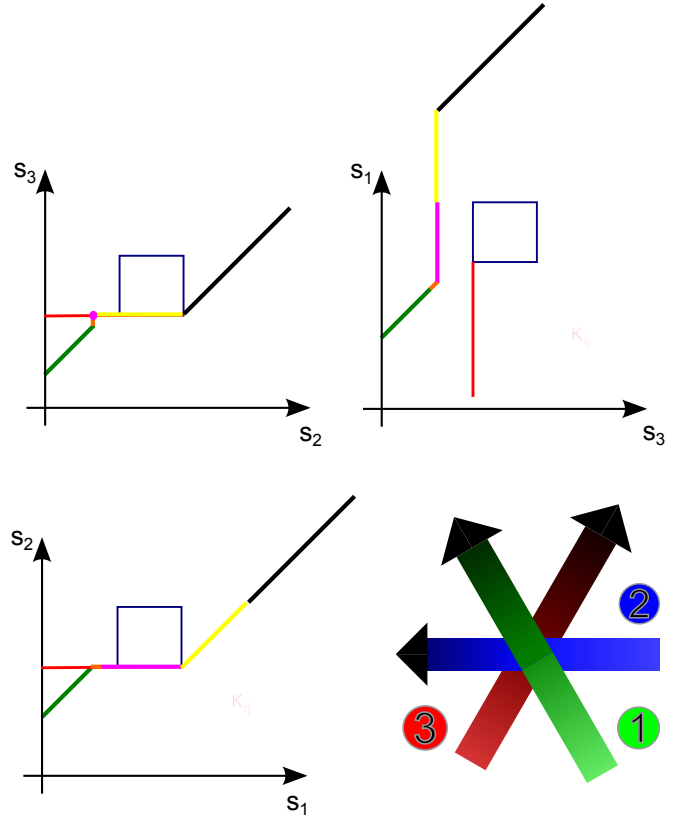


Fig. 6. The bottom-left drawing represents a 3-path intersection where priorities are fixed to $1 \succ 2$, $2 \succ 3$ and $1 \succ 3$. The three other drawings visualize in each plane (s_i, s_j) the obstacle region (in blue), the forbidden gates (red), and the optimal trajectory generated by the algorithm. The color of the trajectory changes when a collision pair leaves or reaches a gate or the boundary of the obstacle region.

same class can be deduced from each other by a continuous deformation). In our formalism, each admissible graph defines an homotopy class. In our particular application, we recover their optimality result with simple means (as opposed to their proof that is based on concepts of advanced differential geometry, namely Gromov's link condition for CAT(0) spaces).

B. Optimal trajectories

As the algorithm of section IV-A generates an optimal trajectory for a given priority graph, the motion planning optimization boils down to a combinatorial problem. By computing and comparing the costs of the optimal trajectories among the feasible priority graphs, we can conclude on the optimal planning solution by choosing the least costly trajectory. This method is suitable when there are few vehicles crossing the intersection, but its high algorithmic complexity (due to the combinatorics of priorities) can become a prohibitive numerical cost when the number of vehicle grows. This motivates the following heuristic algorithm.

C. A heuristic algorithm with unconstrained graph

The last section proves that once the priority graph is fixed, the optimum is reached following algorithms belonging to

the "bug" family, emanating from the work of [15]. Indeed, the optimal paths go at maximum speed until they reach the frontier of a forbidden region. Then, they follow the boundary as long as necessary. In the absence of fixed priority graph, we propose to have the same reactive approach, moving the vehicles as fast as possible. A graph is thus iteratively built, and each time a gate is reached, the retained priority must be compatible with the graph already built.

After setting $\phi(0) = x_{\text{init}}$ and initializing G to an empty graph, the algorithm proceeds as follows until χ_{goal} is reached:

- $\phi'(t) \leftarrow \mathbf{1}$
- *Stay in χ* : For i such that $\phi_i(t) = 1$, do $\phi'_i(t) \leftarrow 0$
- *Update G* : While there exists a collision pair i, j whose relative priorities are not defined in G and such that $\phi(t)$ belongs to the boundary of $H_{i>j}$, do $i \xrightarrow{G} j$ if it yields a feasible priority graph, else do $j \xrightarrow{G} i$
- *Avoid Forbidden Gates*: While there exists $i \xrightarrow{G} j$ such that $\phi(t)$ belongs to the boundary of the gate $H_{j>i}$, do: $\phi'(t) \leftarrow$ orthogonal projection of $\phi'(t)$ on the boundary of $H_{j>i}$
- *Avoid Obstacle Region*: While there exists $i \xrightarrow{G} j$ such that $\phi(t)$ belongs to the boundary of $\bar{\chi}_{\text{obs}}^{ij}$, If $\mathbf{1}$ points towards χ_{obs}^{ij} , do:
 $\phi'(t) \leftarrow$ orthogonal projection of $\phi'(t)$ on any tangential hyperplane to $\bar{\chi}_{\text{obs}}^{ij}$ at $\phi(t)$

The trajectory is defined recursively, the algorithmic complexity remains relatively low, and yields a generated priority graph G *a posteriori*. Moreover, we have guarantees the built trajectory is optimal for this priority graph.

V. CONCLUSION

The mathematical framework we propose to address the cooperative motion planning problem at intersections is based on path-velocity decomposition. Hence, the problem boils down to searching an optimal path in a high-dimensional coordination space. We define a priority relation between the vehicles that provides a tool to classify the feasible trajectories, and propose an algorithm to construct an optimal trajectory for given priorities.

As the search for a global optimum is inherently related to the determination of all optima given a graph, it is of high algorithmic complexity. As a result, a good heuristics for fixing priorities is at the core of a simple quasi-optimal solution, as suggested by the intuition.

In the future, we plan to find such heuristics, and to take into account more realistic assumptions: varying number of vehicles, more general coordination space, and above all dynamic constraints. The next step will be the integration of the selected algorithms in a real intersection crossed by numerous vehicles such as cooperative cybercars developed at INRIA.

REFERENCES

[1] S. Akella and S. Hutchinson. Coordinating the motions of multiple robots with specified trajectories. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 1, pages 624 – 631 vol.1, 2002.



Fig. 7. Cooperative cybercars in the French city of Nancy during the MobiVIP project: INRIA develops fully automated transportation systems that need cooperative motion planning (here is a platooning example).

[2] S. J. Buckley. Fast motion planning for multiple moving robots. In *Proceedings IEEE International Conference on Robotics & Automation*, pages 322–326, 1989.

[3] A. Colombo and D. Del Vecchio. Efficient algorithms for collision avoidance at intersections. *Hybrid Systems: Computation and Control*, 2012.

[4] K. Dresner and P. Stone. Multiagent traffic management: a reservation-based intersection control mechanism. In *Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004. Proceedings of the Third International Joint Conference on*, pages 530 –537, july 2004.

[5] Kurt Dresner and Peter Stone. Mitigating catastrophic failure at intersections of autonomous vehicles. In *AAMAS Workshop on Agents in Traffic and Transportation*, pages 78–85, Estoril, Portugal, May 2008.

[6] T. Fraichard and C. Laugier. Planning movements for several coordinated vehicles. In *Intelligent Robots and Systems '89. The Autonomous Mobile Robots and Its Applications. IROS '89. Proceedings., IEEE/RSJ International Workshop on*, pages 466 –472, sep 1989.

[7] Robert Ghrist and Steven M. LaValle. Nonpositive curvature and pareto optimal coordination of robots. *SIAM J. Control Optim.*, 45:1697–1713, November 2006.

[8] Caminiti Hafner, Cunningham and Del Vecchio. Automated vehicle-to-vehicle collision avoidance at intersections. 2011.

[9] Ismail and Sahin. Railway traffic control and train scheduling based on inter-train conflict management. *Transportation Research Part B: Methodological*, 33(7):511 – 534, 1999.

[10] K. Kant and S. W. Zucker. Toward efficient trajectory planning: The path-velocity decomposition. *International Journal of Robotics Research*, 5(3):72–89, 1986.

[11] H. Kowshik, D. Caveney, and P.R. Kumar. Provable systemwide safety in intelligent intersections. *Vehicular Technology, IEEE Transactions on*, 60(3):804 –818, march 2011.

[12] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006. Available at <http://planning.cs.uiuc.edu/>.

[13] S.M. LaValle and S.A. Hutchinson. Optimal motion planning for multiple robots having independent goals. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 3, pages 2847 –2852 vol.3, apr 1996.

[14] S. Leroy, J. P. Laumond, and T. Simeon. Multiple path coordination for mobile robots: A geometric algorithm. In *In Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1118–1123, 1999.

[15] V. J. Lumelsky and A. A. Stepanov. *Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape*, pages 363–390. Springer-Verlag New York, Inc., New York, NY, USA, 1990.

[16] Olivier Mehani and Arnaud De La Fortelle. Trajectory planning in a crossroads for a fleet of driverless vehicles. In *Proceedings of the 11th international conference on Computer aided systems theory, EUROCAST'07*, pages 1159–1166, Berlin, Heidelberg, 2007. Springer-Verlag.

[17] P.A. O'Donnell and T. Lozano-Periz. Deadlock-free and collision-free coordination of two robot manipulators. In *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pages 484 –489 vol.1, may 1989.

[18] C. Tomlin, G.J. Pappas, and S. Sastry. Conflict resolution for air traffic management: a study in multiagent hybrid systems. *Automatic Control, IEEE Transactions on*, 43(4):509 –521, apr 1998.