Stackelberg equilibrium in robot platooning*

Arnaud Canu and **Matthieu Boussard** and **Abdel-Illah Mouaddib**
GREYC (UMR 6072), Université de Caen Basse-Normandie,
Campus Côte de Nacre, boulevard du Marchal Juin
BP 5186 - 14032 Caen CEDEX, FRANCE

Abstract

We describe a formalism for robot platooning, based on the adaptation of flocking rules in the Vector-Valued Decentralized Markov Decision Process (shortly 2V-DEC-MDP) framework. After a reminder on stochastic games, we will prove the conditions under which the agents follow a Stackelberg equilibrium. This leads to the adaptation of the initial value functions of the 2V-DEC-MDP, in order to reach this property. Experimental results compare the initial framework with the Stackelberg equilibrium according to the quality of the solution and the complexity criteria.

Introduction

Multiagent planning has been widely studied. More particularly, planing under uncertainty allows to represent problems where actions' outcomes are uncertain. For those problems, and for a single agent, Markov Decision process (MDP) (Puterman 1994) framework allows the agent to maximize its expected reward of any state and derive an optimal policy. DEC-POMDP framework (Bernstein, Zilberstein, and Immerman 2000) has been designed for the same purpose in multiagent settings. Although DEC-POMDPs can find optimal policies, their complexity for the general case is so high that it is hard to deal with some real applications. When dealing with 50 agents, even the algorithms for finding approximate policies of DEC-MDP can't compute satisfying policies. Some sub-classes can deal with problems of this scale (Becker et al. 2004), but their expressiveness is reduced for the problem of coordinating a fleet of robots. The 2V-DEC-MDP have been introduced to coordinate a large number of agents, extending the MDP framework, by considering local interactions with local full observability.

We are starting here with a concrete problem, namely robot platooning (Michaud et al. 2006), where the goal is to build and to maintain a formation for a group of mobile robots from a starting point to a goal. We will look at the particular problem of agents who aim to organize themselves according to a line shape, but our formalism could easily be adapted to other shapes. Thus, we propose here a solution to make those kind of platoons. We are considering that environment has unpredictable properties so actions have nondeterministic effects (for example, an agent can skid on a wet ground). We aim at finding a fully decentralized approach, with no communication (hostile area). Those kind of problems have been studied with flocking approach, where the agents have to maintain a global shape thanks to few simple local basic rules. So we want to merge 2V-DEC-MDP properties with the flocking ones.

Stochastic games theory (Shapley 1953) is a formalism for situations where an agent's reward do not only depends on the action it does but on the actions of all the agents. Our problem could easily be formalized by a Stachastic game, especially by using a Stackelberg equilibrium (a situation where the leader of the group acts knowing that its actions influence all the group). Moreover, we could easily formalize such a Stochastic game by using a 2V-DEC-MDP and then compare a 2V-DEC-MDP using a Stackelberg equilibrium to a 2V-DEC-MDP not using any equilibrium.

This paper is organized as follows: first, we remind the basic of the flocking and 2V-DEC-MDP framework. Then, we reformulate the flocking rules with a 2V-DEC-MDP. Then we use stochastic games theory in order to compare the 2V-DEC-MDP quality to the one of agents following a Stackelberg equilibrium. We show under which assumption the 2V-DEC-MDP leads to a Stackelberg equilibrium, so that we propose new 2V-DEC-MDP value function according to those assumptions. We finish by showing experimental results comparing the quality and the complexity of the initial 2V-DEC-MDP to the one of Stackelberg equilibrium.

Flocking

Flocking rules (Reynolds 1987) are a set of three very simple rules describing the behaviour of the agents. Those rules are :

1. Cohesion : steer to move toward the average position of local flockmates,
2. Separation : steer to avoid crowding local flockmates,
3. Alignment : steer towards the average heading of local flockmates.

Despite the simplicity of those rules, agents manage to maintain the shape of the group. The main advantage of this ap-

*The authors would like to thank the DGA (Direction Générale de l'Armement) for supporting this work.

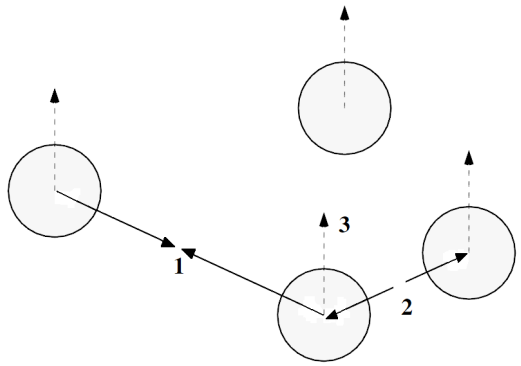

Figure 1: Flocking rules: (1) cohesion, (2) separation, (3) alignment

proach is that it is fully decentralized, with no communication at all.

In our formalism, we describe platooning as a particular form of flocking, where agents try to maintain a line shape and to move toward the platoon's objective (in this line, each agent has the same orientation as the previous agent if it is possible, and the leader heads to the objective. The global shape will then be a straight line or, if agents do not have enough space, a broken straight line). This can be done by giving particular flocking rules to each agent:

1. Cohesion : steer to wait for agents behind it,
2. Separation : steer to avoid agents in front of it,
3. Alignment : steer to move toward the near agent in front of it, or toward the objective if no one is in front of it.

By using those flocking rules into a 2V-DEC-MDP, we want the agents to maintain the group's shape, and also to act considering the uncertainty on the outcome of the actions.

Vector-Valued DEC-MDP

In (Boussard, Bouzid, and Mouaddib 2007; Mouaddib, Boussard, and Bouzid 2007), the Vector-Valued Decentralized Markov Decision Process (2V-DEC-MDP) framework has been proposed to coordinate locally the actions of a group of agents. It is based on MDP (Puterman 1994) with an online coordination part. Assuming without loss of generality that all agents are identical, a 2V-DEC-MDP is a set of 2V-MDP, one per agent. A 2V-MDP is composed by an off-line part, an MDP, and an on-line part to adapt its actions with the other agents.

The MDP is a tuple $\langle S, A, P, R\rangle$, with:

- S a set of states,
- A a set of action,
- $P : S \times A \times S \rightarrow [0; 1]$, the transition function,
- $R : S \times A \times S \rightarrow \mathbb{R}$, the reward function which expresses both positive reward for goal states and negative reward for hazardous states.

For the optimality criteria, we use an expected reward on a finite horizon T. The optimal value function V^* of a state is defined by:

$$V^*(s) = \max_{a \in A}(R(s,a) + \sum_{s' \in S} P(s,a,s') \cdot V^*(s')), \forall s \in S$$

A policy is a function $\pi : S \rightarrow A$, the optimal policy is a policy π^*, such that:

$$\pi^*(s) = argmax_a(R(s,a) + \sum_{s' \in S} P(s,a,s') \cdot V^*(s')), \forall s \in S$$

We define the neighborhood for an agent i as the set of states of (detected) agents who can interact with i. Until now, we assumed that all the agents near enough (according to a fixed maximum distance d) could be detected and their states could be known. Taking into account partial observability will be the subject of some future works. If the neighborhood is too big, it can be restricted to a subset (the more the neighborhood will be big and the more the policy will be good but the more the computation of this policy will take time).

The on-line part of a 2V-MDP is built with the computation of local social impact, according to local observations. The functions for computing the value of the impact on the group are:

- ER for the individual reward (the value of the optimal policy of the MDP),
- JER for the group interest,
- JEP for the negative impact on the group.

Using those functions, the agents will use a $LexDiff$ operator to choose the policy (i.e. the best action) to apply.

$LexDiff$ build a vector $(ER(\pi_i), JER(\pi_i), JEP(\pi_i))$ for each policy π_i and normalize each values vector $v_i = (v_i^1, v_i^2, v_i^3)$ to a utilities vector $v_u = (v_u^1, v_u^2, v_u^3)$. $LexDiff$ then permute those utilities vectors so that each vector (v^1, v^2, v^3) be such as: $v^1 \geq v^2 \geq v^3$. The best vector is then founded by a lexicographic order: for two vectors $v_a = (v_a^1, v_a^2, v_a^3)$ and $v_b = (v_b^1, v_b^2, v_b^3)$, we choose v_a if $v_a^1 > v_b^1$ and v_b if $v_a^1 < v_b^1$. If $v_a^1 = v_b^1$, we compare v_a^2 and v_b^2, and so on.

Thanks to this design, the DEC-MDP is expressed as a set of 2V-MDP, allowing the coordination problem to be tractable. In (Boussard, Bouzid, and Mouaddib 2008), ER JER and JEP have been defined for platoon emergence, but this work does not try to keep the shape of the platoon. So we are interested here to use flocking rules in this framework to built the platoon and also to keep platoon's shape.

Flocking as a 2V-DEC-MDP

We use 2V-DEC-MDP to formalize our problem, by translating the three criteria into three formulae (each formula having one or more equations) which will parameterize each 2V-MDP. We consider ER as the alignment criterion, JER as the cohesion criterion and JEP as the separation criterion.

Notations

- s_i^j is the state j of agent i (the environment being reduced to a discrete set of possible positions, a state is one position of this set and one orientation),

- $\overrightarrow{s} = (s_1, \ldots, s_N)$ is the joint state vector,
- $face(s)$ gives all the agents the are closer to the objective than s,
- $distance(s^1, s^2)$ gives the number of actions needed to go from s^1 to s^2,
- $angle(s^1, s^2)$ gives the angle between the orientation of s^1 and the one of s^2. We have:

$$angle(s^1, s^2) = \frac{\|orientation_{s^1} - orientation_{s^2}\|}{angle_{max}}$$

- $back(s)$ gives the next place available behind s (if s^1, the location just behind s according to the orientation of s, is available, we return s^1. If it is not available, we return $back(s^1)$).

So now, using those notations, we can write the formulae for ER, JER and JEP int the platooning context:

ER: Alignment

$$ER(s, a) = \sum_{s' \in S} p(s, a, s') ER_i, \quad i = 1, 2, 3$$

Depending on the situation, ER_i are defined by:

$$ER_1 = V^*(s')$$
$$ER_2 = - \min_{s_j \in face(s')} (distance(s', s_{b1}) + \frac{angle(s', s_{b1})}{angle_{max}})$$
$$ER_3 = -(distance(s', s_{b2}) + \frac{angle(s', s_{b2})}{angle_{max}})$$

where $s_{b1} = back(s_j)$, $s_{b2} = back(leader)$ and $V^*(s)$ a function of the distance between s and the objective of the platoon. $distance(s^1, s^2)$ gives the cost of going from s^1 to s^2 and $angle(s^1, s^2)$ gives the cost of rotating from the orientation of s^1 to the one of s^2. Thus, we add in those equations two cost: we look for the cost of going from a state s^1 to a state s^2, wich means the cost of reaching the position of s^2 AND rotating to the good orientation. We divide the angle by the maximum angle, because we want to be sure that the cost of the distance will always be bigger than the cost of the angle, so the agent will not choose to stay on a distant place for saving the cost of a rotation. In ER_2 and ER_3, we use $back(target)$ instead of $target$, because the agent wants to go behind its target.

An agent does not have the same objectives whether it is on a leader position or inside a platoon. Indeed, a leader will move in the direction of its objective, while a non-leader agent will follow the one in front of it. Hence, an agent have to choose which equation to follow before resolving its 2V-MDP.

So, if the agent is a leader, or if it is out of range of any platoon, it chooses ER_1. If it is inside a platoon but it knows that the leader is behind it, it chooses ER_3. Finally, if it is inside a platoon and have no leader behind it, it chooses ER_2.

JEP: Separation

$$JEP(s, a) = \sum_{s' \in S} [p(s, a, s') \cdot \sum_{s_j \in D} (\sum_{a_j^k, k=1}^{|A_j|} p(s_j, a_j^k, s') \cdot C)]$$

Where D is the set of states of detected agents in neighborhood and C a constant equal to the cost of a collision between two agents.

JER: Cohesion

$$JER(s, a) = \sum_{s' \in S} (p(s, a, s') \cdot K(s'))$$

Where $K(s)$ is the function which estimate the gain of a given situation for the group. $K(s)$ gives a reward if at least one agent is behind s.

After choosing an equation for the ER criteria, the agent has to fix the weight of ER, JER and JEP. For a leader, we set w_{JEP} to 0 since the criterion is with no sense for it and, typically, w_{ER} to 0.49 and w_{JER} to 0.51. For a non-leader, $w_{JEP} = 0.35$, $w_{ER} = 0.32$ and $w_{JER} = 0.33$ (except if a leader is detected behind the agent, in which case $w_{JER} = w_{JEP} = 0$, and $w_{ER} = 1$). Finally, for any agent, $w_{JER} = 0$ as soon as it is near to the objective of the platoon. Experimentations proved that values of those weights do not change anything on the behavior of the agents. The only important thing is the order of those weights: the most important criteria has to have the biggest weight, the second criteria has to have the second weight, etc., so we choose arbitrary values for those weights.

Stochastic games

We introduce in this section stochastic games approach which is able to solve similar problems than our approach and it will be used as a basis to compare the performance of our approach.

A stochastic game (Shapley 1953) is defined in (Chaibdraa 2005) by a tuple $\langle N, S, A, R, T \rangle$, where:

- N is the number of agents taking part in the game,
- S is the set of states in which the game can be (a state of the game describing the state of the world and of every player/agent),
- $A = \{A_1, A_2, \ldots, A_N\}$ the set of possible actions for every agent, where A_i is the set of actions for agent i ($A_i = \left\{a_i^1, \ldots, a_i^{|A_i|}\right\}$),
- $R = \{R_1, R_2, \ldots, R_N\}$ the set of reward functions of every agent such as, for a given agent i, we have $R_i : S \times A_1 \times \ldots \times A_N \rightarrow \mathbb{R}$,
- T represent the transition's model between states, according to the joint actions of the agents. We have $T : S \times A_1 \times \ldots \times A_N \times S \rightarrow [0, 1]$.

At each step, each player chooses an action based on its actual state and its policy. The game then moves to a new state s'. The i-agent's policy, noted π_i, might be of 2 types: if the player follows a pure strategy, we will have $\pi_i : S \rightarrow A$ but

if the player follows a mixed strategy, we will have a probabilities distribution on states ($\pi_i : S \rightarrow [0,1]^{|A_i|}$). The joint policy for every agent of the game is $\overrightarrow{\pi} = (\pi_1, \ldots, \pi_N)$.

To estimate a strategy's value, it is necessary to know the utility for a given player to follow a given strategy. Let write $\pi_1(s)$ the chosen action by applying the π_1 policy on the state s. We can then write $\overrightarrow{\pi(s)} = (\pi_1(s), \ldots, \pi_N(s))$ the joint policy for this state. In this game, every agent i has (by definition) an immediate utility $u_i^s(\overrightarrow{\pi(s)})$. We can write $u_i^s(\overrightarrow{\pi(s)}) = R_i(s, \overrightarrow{\pi(s)})$ and we will be able to calculate $U_i^{\overrightarrow{\pi(s)}}(s)$ the expected utility for an agent i if, in a state s, every agents apply the joint policy $\overrightarrow{\pi}$:

$$U_i^{\overrightarrow{\pi}}(s) = u_i^s(\overrightarrow{\pi(s)}) + \beta \sum_{s' \in S} T(s, \overrightarrow{\pi(s)}, s') \cdot U_i^{\overrightarrow{\pi}}(s')$$

In a game, a Stackelberg equilibrium (Stackelberg 1952; Könönen 2003) is a situation where the leader of a group knows that it is the leader. It makes decisions, and its followers estimate BR the best response (e.q. the best decision) to apply, according to this decision. The leader can then estimate what reactions will the other agents have, and makes the decision which will bring him the best reward, relative to those reactions. If the objective of the leader is to maximize the group's reward, it will make the decision witch will bring the best reward to this group.

A Nash equilibrium is a situation in which no agent can change its action without reducing its reward. In (Chaibdraa 2005), they described how a Nash equilibrium can be adapted to a stochastic game. By the same way, let define the Stackelberg equilibrium for a stochastic game. First, let $BR_j(\pi_i)$ be the set of best policies π_j for j, knowing that i apply the policy π_i, and $\overrightarrow{BR} = (\pi_1 \in BR_1(\pi_i^*), \ldots, \pi_N \in BR_N(\pi_i^*))$. If, for every state $s \in S$, the leader's policy π_i^* (i being the leader agent) respects Eq.1, then we have a Stackelberg equilibrium.

$$\min_{\overrightarrow{BR}} U_i^{\pi_1,\ldots,\pi_i^*,\ldots,\pi_N}(s) = \max_{\pi_i}(\min_{\overrightarrow{BR}} U_i^{\pi_1,\ldots,\pi_N}(s)) \quad (1)$$

Stackelberg equilibrium in 2V-DEC-MDP

In order to compare our formalism to the Stackelberg equilibrium, we aim to find when a 2V-DEC-MDP leads an agent to follow such an equilibrium.

Remainder on the leader's behavior

We call a leader any agent that is in the front of the platoon, whether if it had been indicated or not. It's behavior is depicted by those equations:

$$ER(s,a) = \sum_{s' \in S} (p(s,a,s') \cdot V^*(s')) \quad (2)$$

$$JEP(s,a) = \sum_{s' \in S} [p(s,a,s') \cdot \sum_{s_j \in D} (p(s_j, a_j, s') \cdot C)] \quad (3)$$

$$JER(s,a) = \sum_{s' \in S} (p(s,a,s') \cdot K(s')) \quad (4)$$

Let L be the leader, then the utility function of this leader is for a policy π and a state s:

$$U_L(s) =$$
$$LexDiff_{\pi_L}(ER(s,\pi_L(s)), JER(s,\pi_L(s)), JEP(s,\pi_L(s)))$$

The agent will then use the π_L chosen by $LexDiff$.

Detecting Stackelberg equilibrium

Theorem 1. *An agent using a 2V-MDP is following a Stackelberg equilibrium, if and only if each criterion leads to a Stackelberg equilibrium.*

Proof. We aim to show that each criterion leads to a Stackelberg equilibrium if and only if applying $LexDiff$ on those criteria leads to a Stackelberg equilibrium.

Here, the agent's policy is build on the fly by $LexDiff$: for each state, this operator choose the action the agent will apply. Because the Stackelberg equilibrium definition says that a leader is in equilibrium if and only if it follows Eq.1 for each state s, we just have to show that the $LexDiff$ equation is equivalent to Eq.1 for every state.

Let c be a criterion and $Q_c(s, \pi_i(s))$ be the utility for agent i to apply its policy π when it is in the state s, according to c. If we assume that each criterion c leads to a Stackelberg equilibrium, then we can write:

$$\min_{\overrightarrow{BR(\pi_L^*)}} Q_c(s, \pi_L^*(s)) \quad (5)$$

$$= \max_{\pi_L}(\min_{\overrightarrow{BR(\pi_L)}} Q_c(s, \pi_L(s))) \forall Q_c \in \overrightarrow{Q}_{criteria} \quad (6)$$

$$\leftrightarrow \min_{\overrightarrow{BR(\pi_L^*)}} \left[\min_{Q_c \in \overrightarrow{Q}_{criteria}} Q_c(s, \pi_L^*(s)) \right] \quad (7)$$

$$= \max_{\pi_L} \left[\min_{\overrightarrow{BR(\pi_L)}} (\min_{Q_c \in \overrightarrow{Q}_{criteria}} Q_c(s, \pi_L(s))) \right] \quad (8)$$

And, we know that $LexDiff$ seeks the action which minimizes the biggest regret. Moreover, minimizing regret relative to a criterion is equivalent to maximizing utility relative to this criterion. We can then reformulate $LexDiff$ like an operator which seeks the lowest utility maximizing action. In other words, because we work at horizon 1, we have:

$$U_L^{\pi_1,\ldots,\pi_N}(s) = LexDiff_{\pi_L}(\overrightarrow{Q}_{criteria})$$
$$= \max_{\pi_L} \left[\min_{Q_c \in \overrightarrow{Q}_{criteria}} Q_c(s, \pi_L(s)) \right]$$

We can also deduce the following equality:

$$U_L^{\pi_1,\ldots,\pi_L^*,\ldots,\pi_N}(s) = \min_{Q_c \in \overrightarrow{Q}_{criteria}} Q_c(s, \pi_L^*(s))$$

So, Eq.7=Eq.8 is equivalent to:

$$\min_{\overrightarrow{BR(\pi_L^*)}} U_L^{\pi_1,\ldots,\pi_L^*,\ldots,\pi_N}(s) \quad (9)$$

$$= \max_{\pi_L}(\min_{\overrightarrow{BR(\pi_L)}} U_L^{\pi_1,\ldots,\pi_L,\ldots,\pi_N}(s)) \quad (10)$$

Thus, the $LexDiff$ operator leads to a Stackelberg equilibrium. We shown that $LexDiff$ leads to a Stackelberg equilibrium if and only if every criterion leads to such an equilibrium. □

Adapting 2V-DEC-MDP to reach a Stackelberg equilibrium

Theorem 2. *The criteria of the 2V-DEC-MDP described in (Eq.2,Eq.3,Eq.4) don't lead to a Stackelberg equilibrium.*

Proof. According to theorem.1 We aim to show that ER, JER or JEP is not a Stackelberg equilibrium. Because of Eq.1, we can write:

$$\min_{\overrightarrow{BR(\pi_L^*)}} ER(s,\pi_L^*(s)) \neq \max_{\pi_L}(\min_{\overrightarrow{BR(\pi_L)}} ER(s,\pi_L(s))) \quad (11)$$

$$\min_{\overrightarrow{BR(\pi_L^*)}} JER(s,\pi_L^*(s)) \neq \max_{\pi_L}(\min_{\overrightarrow{BR(\pi_L)}} JER(s,\pi_L(s))) \quad (12)$$

$$\min_{\overrightarrow{BR(\pi_L^*)}} JEP(s,\pi_L^*(s)) \neq \max_{\pi_L}(\min_{\overrightarrow{BR(\pi_L)}} JEP(s,\pi_L(s))) \quad (13)$$

So, if we follow ER only, the objective will be to maximize the ER value. We then have, with π_L^* the leader's policy resulting from the ER criterion:

$$ER(s_L,\pi_L^*(s_L)) = \max_{\pi_L} ER(s_L,\pi_L(s_L))$$

However, the equation:

$$ER(s_L,\pi_L^*(s_L)) = \sum_{s'_L \in S_L} (p(s_L,\pi_L^*(s_L),s'_L)\cdot V^*(s'_L))$$

suppose that only one policy π_i is assumed for each agent in the neighborhood when computing the value of ER. Indeed, $V^*(s'_L)$ is the reward of the leader when it gets closer to its objective (ie. when s'_L is closer to the objective than s_L). However, even if an agent gets closer to a point according to a pure geographic point of view, it can in reality move away from this point because of the other agents: if they go between the leader and its objective, it will have to avoid them, what will imply additional movements. Another decision could make the leader to get closer to its objective without being constrained by the other agents. So we have:

$$ER_{2VMDP} = \max_{\pi_L} ER(s,\pi_L(s)|\overrightarrow{\pi}_{assumed})$$

while the equation for a Stackelberg equilibrium following leader will be:

$$ER_{st} = \max_{\pi_L}(\min_{\overrightarrow{\pi}} ER(s,\pi_L(s)|\overrightarrow{\pi}))$$

But we are not sure that the agents will follow the assumed policies. We then have:

$$ER_{2VMDP} \leq ER_{st}$$

So we have:

$$\min_{\overrightarrow{BR(\pi_L^*)}} ER(s,\pi_L^*(s)) \leq \max_{\pi_L}(\min_{\overrightarrow{BR(\pi_L)}} ER(s,\pi_L(s)))$$

So, ER does not lead to a Stackelberg equilibrium, nor a 2V-DEC-MDP parametrized by ER, JER and JEP. □

criteria rewriting

According to the preceding theorem, the leader does not follow a Stackelberg equilibrium, because its criteria don't lead to such an equilibrium. Is it possible to change those criteria, to make the leader following a Stackelberg equilibrium? If yes, what will be the complexity of computing such an equilibrium? Indeed, the leader will not only have to consider the other agent's state anymore but also their policies.

ER criterion (Eq.11): We have $distance(s,g)$ the distance between s and the goal g if we don't know what the other agents do, and $distance_{\overrightarrow{\pi}}(s,g)$ the distance between s and g knowing the 1 to N agents' policies. Actually, we have:

$$ER(s_L,\pi_L^*(s_L)) = \sum_{s'_L \in S_L} (p(s_L,\pi_L^*(s_L),s'_L)\cdot(-distance(s'_L,g)))$$

If we want to take the other agents' decisions into account, we can write:

$$ER(s_L,\pi_L^*(s_L)) = \min_{\overrightarrow{BR(\pi_L^*)}} \sum_{s'_L \in S_L} (p(s_L,\pi_L^*(s_L),s'_L)\cdot(-distance_{\overrightarrow{\pi}}(s'_L,g)))$$

And:

$$distance_{\overrightarrow{\pi}}(s'_L,g) = \sum_{\overrightarrow{s}} p(\overrightarrow{s'};\overrightarrow{\pi(s)};\overrightarrow{s'})\cdot distance_{\overrightarrow{s}'}(s'_L,g)$$

$$= \sum_{\overrightarrow{s}'} \left[\left(\prod_{i=1}^{N} p(s_i,\pi_i(s_i),s'_i)\right)\cdot distance_{\overrightarrow{s}'}(s'_L,g)\right]$$

Where $distance_{s'_1,\ldots,s'_N}(s,g)$ is the distance from s to g without crossing s'_1, nor s'_2, nor …, nor s'_N.

Because we want to maximize this criterion, we will have:

$$ER(s,\pi_L^*(s)) = \max_{\pi_L}(\min_{\overrightarrow{BR(\pi_L)}} ER(s,\pi_L(s)))$$

Thus, this new ER version leads to a Stackelberg equilibrium.

JER criterion (Eq.12): We said that JER was the following, with $K(s) = 1$ if there are agents behind s and 0 if not:

$$JER(s_L,\pi_L^*(s_L)) = \sum_{s'_L \in S_L} [p(s_L,\pi_L^*(s_L),s'_L)\cdot K(s'_L)]$$

For taking the other agents' policies into account, we can change this criterion as follows:

$$JER(s_L,\pi_L^*(s_L)) = \min_{\overrightarrow{BR(\pi_L^*)}} \left(\sum_{s'_L \in S_L} [p(s_L,\pi_L^*(s_L),s'_L)\cdot K_{\overrightarrow{\pi}}(s'_L)]\right)$$

Where $K_{\overrightarrow{\pi}}(s)$ is the function which estimates the probability that at least one agent stays behind s (the objective being not to break the platoon). K is defined by:
let

$$s_b(s) = \{\overrightarrow{s}' \in S^{|N|} | \exists s_i \in \overrightarrow{s}' \text{ with } isBack(s_i,s) = true\}$$

in

$$K_{\overrightarrow{\pi}}(s) = \sum_{\overrightarrow{s}' \in s_b(s)} p(\overrightarrow{s},\overrightarrow{\pi(s)},\overrightarrow{s'}) = \sum_{\overrightarrow{s}' \in s_b(s)} \left[\prod_{i=1}^{N} p(s_i,\pi_i(s_i),s'_i)\right]$$

Where $isBack(\overrightarrow{s^1,s^2})$ is a function which returns true if s^1 is behind s^2.

Because we aim to maximize the criterion, we will have:

$$\begin{array}{rl} & JER(s,\pi_L^*(s)) = \max_{\pi_L} JER(s,\pi_L(s)) \\ \iff & \min_{\overrightarrow{BR(\pi_L^*)}} JER(s,\pi_L^*(s)) \\ = & \max_{\pi_L}(\min_{\overrightarrow{BR(\pi_L)}} JER(s,\pi_L(s))) \end{array}$$

Thus, we have an equation for the JER criterion such as if the leader only follows this criterion, it will be in a Stackelberg equilibrium.

JEP criterion (Eq.13): the JEP criterion is the following, with C a negative constant which represents the cost of a collision between two agents:

$$JEP(s_L,\pi_L^*(s_L)) = \sum_{s'_L \in S_L} p(s_L,\pi_L^*(s_L),s'_L)\lambda$$

where:

$$\lambda = \sum_{s_j,j=1}^{N} \left(\sum_{\pi_j} p(s_j,\pi_j(s_j),s'_j)\cdot C \right)$$

If we want to take into account the other agents' policies, we can rewrite JEP, with $p_L = p(s_L,\pi_L^*(s_L),s'_L)$ and $p_j = p(s_j,\pi_j(s_j),s'_j)$:

$$\begin{array}{l} JEP(s_L,\pi_L^*(s_L)) \\ = \sum_{s'_L \in S_L} \left[p_L \cdot \sum_{s_j,j=1}^{N} \left(\min_{\pi_j \in BR(\pi_L^*)} p_j \cdot C \right) \right] \\ = \min_{\overrightarrow{BR(\pi_L^*)}} \left(\sum_{s'_L \in S_L} \left[p_L \cdot \sum_{s_j,j=1}^{N} p_j \cdot C \right] \right) \end{array}$$

Because we aim to maximize the criterion, we will have:

$$\begin{array}{rl} & JEP(s,\pi_L^*(s)) = \max_{\pi_L} JEP(s,\pi_L(s)) \\ \iff & \min_{\overrightarrow{BR(\pi_L^*)}} JEP(s,\pi_L^*(s)) \\ = & \max_{\pi_L}(\min_{\overrightarrow{BR(\pi_L)}} JEP(s,\pi_L(s))) \end{array}$$

Thus, this JEP criterion leads to a Stackelberg equilibrium.

So, we rewrote the 3 criteria so that they all lead, independently of each other, to a Stackelberg equilibrium. Thus, if the leader follows those criteria, it will be in a Stackelberg equilibrium.

Complexity

We will now compare the complexity of our initial formalism to the one of the Stackelberg formalism. We will make this comparison on the ER criterion, but those results are the same for JER and JEP.

Initial criteria

For the ER criterion, the hard part is the distance computation ($distance(s,g)$). We will have to compute once this distance to estimate the value of $ER(s,a)$. So, to find the best action, we will have to compute A times this distance, A being the number of different actions an agent can do. Thus, if we write d the time to compute a distance, the global complexity for this criterion will be in:

$$O(A \cdot d)$$

Moreover, if all the distances have been computed before, in V^*, the complexity become: $O(A)$.

Stackelberg adapted criteria

A leader which follow the Stackelberg version of the ER criterion first have to estimate the different policies for all the agents. The complexity to estimate the policies of an agent is in $O(A \cdot d)$, because those agents apply the "normal" criteria. The agent then have a global complexity for this step in $O(N \cdot A \cdot d)$, with N the number of agents which were detected around it.

We then compute $\overrightarrow{distance_{\vec{\pi}}}(s,g)$ X times, X being the number of different $\overrightarrow{BR(a)}$. If we write D the complexity for computing $\overrightarrow{distance_{\vec{\pi}}}(s,g)$, we then have a global complexity for this step in $O(X \cdot D)$.

In the worst case, the value of X is A^N, with N the number of agents. Indeed, the worst case is the one where any action of A is possible for any agent j. The number of different possible tuples $(actionAgent1,...,actionAgentN)$ is then A^N.

The D value depends on how many times we have to compute a distance, i.e. the number of different possible tuples $(s'_1,...,s'_N)$. Since, in the worst case, an agent can finish in 3 different states after applying an action (because of the uncertainty, 3 being a value fixed in our formalism), we will have 3^N possible tuples, and a complexity for D in $O(3^N \cdot d)$.

Global complexity for computing the value of an action according to ER is then in $O((N \cdot A \cdot d) + (X \cdot D)) = O((N \cdot A \cdot d) + (A^N \cdot 3^N \cdot d))$. We can then estimate the complexity for computing the best action according to ER: $O(A \cdot [(N \cdot A \cdot d) + (A^N \cdot 3^N \cdot d)]) =$

$$O((N \cdot A^2 \cdot d) + (A^{N+1} \cdot 3^N \cdot d))$$

Thus, according to this criterion, complexity is much higher in the Stackelberg case. Moreover, we only used the ER criterion to compare those two formalisms, but we can apply the same reasoning on the two other criteria. Indeed, the complexity growth comes essentially from the "$\min_{\overrightarrow{BR(\pi_L^*)}}$", which is responsible of the A^N factor.

Platoon maintaining

Until now, we only interested into the platoon formation, but what is about maintaining this platoon? Indeed, once the platoon formed, the leader does not have more than 1 agent in its neighborhood anymore: the one which follows it. Complexity for following a Stackelberg equilibrium is then in:

$$O((1 \cdot A^2 \cdot d) + (A^{1+1} \cdot 3^1 \cdot d)) = O(4 \cdot A^2 \cdot d)) = O(A^2 \cdot d))$$

Thus, once the platoon formed, complexity is nearly the same as the one for initial criteria.

Experimental results

We tested our formalism in a simulator which we made for testing those kind of Multi-Agent Systems. In this simulator, agents' behavior is directed by a 2V-DEC-MDP parametrized by the way we described in the previous sections. Actions are stochastic and we use no communication at all. Fig.2 and Fig.3 represent the situation on which we made our tests: circles are agents and polygons are locations where agents can go. Clear polygons are open places, while darker polygons are unavailable places. We made several tests:

- with 7 agents, running the simulator several times with Stackelberg and several times without (10 times each), to compare complexity and quality,
- with a chosen leader, running several simulations with 1, 2, ..., 7 agents in its neighborhood (running 5 times each situation), to analyze complexity.

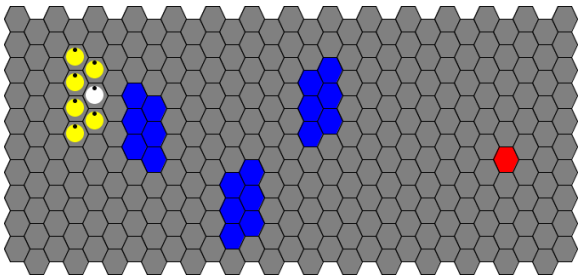

Figure 2: Test environment (start)

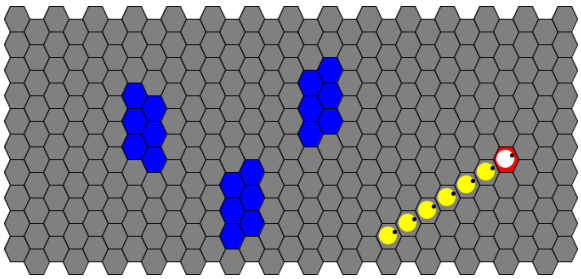

Figure 3: Test environment (end)

We summarize results of our tests in the following graphs. The first one represents the distance evolution from the platoon to its objective during time, while the second one represents the complexity evolution according to the number of agents. Distance is a good mean to estimate the platoon's behavior quality, because it shows how fast the group is able to move. Those graphs show results from the environment presented before, as an example, but we did some tests with other initial configurations and other environments.

The first graph (Fig.4) shows platoon's distance (according to its objective) evolution over time. We can see that a Stackleberg using platoon moves exactly at the same speed as a non Stackelberg using platoon. Thus, the behavior when

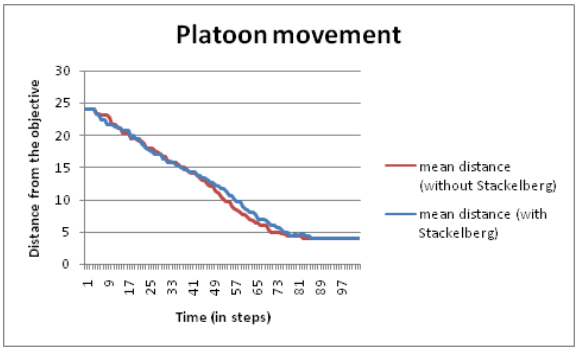

Figure 4: Distance to the objective

we don't use a Stackelberg equilibrium seems to be the same as the one when we use such an equilibrium.

There is an interesting point here: at the end of its evolution, the platoon moves a little faster without Stackelberg than with it. Why this difference ? With Stackelberg, the platoon's leader is more prudent: it chooses to move slowly, to be sure not to break the platoon. Although, this difference is not representative of the global platoon behavior: during most of the time, there is no difference at all.

So, it seems that a platoon using our formalism acts as well as a platoon using a Stackelberg equilibrium. Now, what is about the complexity of computing a decision ?

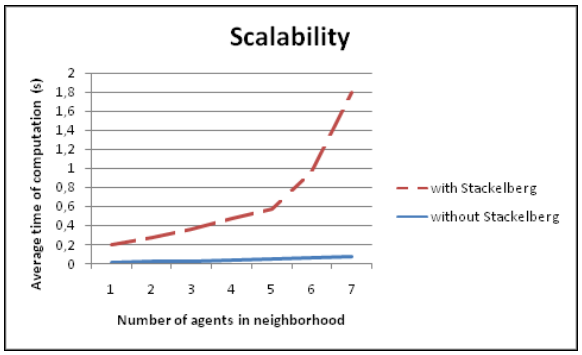

Figure 5: Computation time

the second graph (Fig.5) shows the complexity according to the number of agents in neighborhood. When we don't use a Stackelberg equilibrium, the complexity seems to be proportional to the number of agents. This is logic according to the equations of ER, JER and JEP: each of those equations depends on the agents in the neighborhood. Anyway, computation time grows very slowly with the number of agents. We made some tests with agents starting scattered in the environment: computation time then stay under 0.01 second even with 50 agents.

Complexity is much bigger with a Stackelberg equilibrium: its exponential. We can see that, when more than 7 agents are in the neighborhood of the leader, time for computing an action becomes too hight to be tractable by our simulation. Complexity is then much better with our for-

malism which can deal with problems up to 50 agents.

During tests on other situations, with other environments, results were the same than the ones depicted in Fig.4 and Fig.5: quality of the behavior was the same with and without Stackelberg, complexity was exponential with Stackelberg and proportional without. Moreover, without stackelberg, we are always able to deal with problems up to 50 agents. Thus, it appears that a platoon using our formalism acts as well as a platoon following a Stackelberg equilibrium, with a clearly better complexity.

Conclusion

In this paper we have presented a 2V-DEC-MDP for the platoon formation problem, with a straight line shape. We shown the relationship between the value functions of the 2V-DEC-MDP and the stochastic games. This allowed us to compare the initial formulation of the flocking rules in the 2V-DEC-MDP, with another heading to the Stackelberg equilibrium. The results are that without any loss in quality, the 2V-DEC-MDP complexity is lower than the computation of the Stackelberg equilibrium.

In future work, we will study the impact of adding human controlled agents into the platoon. We will also study the interactions between different platoons (crossing, merging, splitting...). This framework will be validated by implementing this formalism onto an heterogeneous group of robots (Wifibots, Koalas).

References

Becker, R.; Zilberstein, S.; Lesser, V.; and Goldman, C. V. 2004. Solving Transition Independent Decentralized Markov Decision Processes. *Journal of Artificial Intelligence Research* 22:423–455.

Bernstein, D. S.; Zilberstein, S.; and Immerman, N. 2000. The complexity of decentralized control of markov decision processes. In *UAI '00: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, 32–37. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Boussard, M.; Bouzid, M.; and Mouaddib, A.-I. 2007. Multi-criteria decision making for local coordination in multi-agent systems. In *19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007), October 29-31, 2007, Patras, Greece*, volume 2, 87–90. IEEE Computer Society.

Boussard, M.; Bouzid, M.; and Mouaddib, A.-I. 2008. Vector valued markov decision process for robot platooning. In *European Conference on Artificial Intelligence (ECAI 2008), July 21-25, 2008, Patras, Greece*.

Chaib-draa, B. 2005. *Processus Décisionnels de Markov et Intelligence Artificielle*. Groupe PDMIA, 1.1 edition. chapter Jeux, jeux répétés et jeux Markoviens.

Könönen, V. 2003. Asymmetric multiagent reinforcement learning. *iat* 0:336.

Michaud, F.; Lepage, P.; Frenette, P.; Letourneau, D.; and Gaubert, N. 2006. Coordinated maneuvering of automated vehicles in platoons. *ITS* 7(4):437–447.

Mouaddib, A.; Boussard, M.; and Bouzid, M. 2007. Towards a framework for multi-objective multiagent planning. In *AAMAS*.

Puterman, M. L. 1994. Markov decision processes: Discrete stochastic dynamic programming. In *John Wiley and Sons, New York, NY*.

Reynolds, C. W. 1987. Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics* 21(4):25–34.

Shapley, L. 1953. Stochastic games. In *National Academy of Sciences*.

Stackelberg, H. 1952. *The theory of the market economy*. New York, Oxford: Oxford University Press.