Collective Decision-Theoretic Planning for Robot Platton Formation

Arnaud Canu, Abdel-Illah Mouaddib, Matthieu Boussard

Collective Decision-Theoretic Planning for Robot Platoon Formation*

Arnaud Canu[1], Abdel-Illah Mouaddib[1] and Matthieu Boussard[1]

GREYC (UMR 6072), Université de Caen Basse-Normandie
Campus Côte de Nacre, boulevard du Maréchal Juin
BP 5186 - 14032 Caen CEDEX, FRANCE
{arnaud.canu, abdel-illah.mouaddib, matthieu.boussard}@info.unicaen.fr

Abstract : The robot platooning problem has been studied extensively by the robotics community under some assumptions such as communication existance and global full observability. In this paper, we consider the platooning problem where the previous assumptions are not valid. In such a context, platooning can be considered as a specific flocking which is a collective decision model. This model can, thus, be seen as a decentralized multi-criteria decision making process. Vector-Valued Decentralized Markov Decision Process (2V-DEC-MDP) is an interesting framework for multi-criteria collective decision. It has been shown that 2V-DEC-MDP does not consider communication, local interactions and use local full observability which is a sub-class of partial observability. In this paper, we adapt this framework to consider the notion of leader and the relationship with the stochastic games. The theoretic concept of optimality used in such contexts is the Stackelberg Equilibrium (SE). We give the assumptions under which the leader follows the SE when using 2V-DEC-MDP. We present, then, the adaptation of the initial value functions of the 2V-DEC-MDP, in order to reach an SE. Experiments shown us that using the initial 2V-DEC-MDP leads to a near SE with a weak complexity while the adapted 2V-DEC-MDP leads to a SE with a very high complexity and thus a limited scalability which limits its applicability in real-life robotic applications.

Mots-clés : Multiagent Planning, Markov Decisions Processes, Game Theory, Multi-Robot Systems

1 Introduction

Robot platooning (Michaud *et al.*, 2006) is a concrete problem where the goal is to build and to maintain a formation for a group of mobile robots. Robotics community addressed this problem with different techniques where the most popular are the bio-inspired, evolutionist and explicit cooperative approches. Most of these techniques are based on some strong assumptions such as communication, full observability and deterministic actions, or assuming the global desired behavior using top-down technique as ant algorithms (Berman *et al.*, 2007). Our approach relax most of the assumptions and we address these problems where there is no communication, no full and complete observability, actions are stochastic and it is a bottom-up technique where the global behavior is emerged from local behaviors. We will look at the particular problem of agents trying to organize themselves according to a line shape but other shapes could be considered. Those kinds of problems have been studied with flocking approaches, where agents have to maintain a global shape thanks to few local basic rules. In this context, the platooning is a specific flocking and so, a collective decision making process where each agent has to respect its individual interest and the collective interest. Satisfying these criteria leads us to collective multi-criteria decision making problem. Moreover, actions are stochastic and thus the decision making problem is under uncertainty.

For those problems, and for a single agent, Markov Decision process (MDP) (Puterman, 1994) framework allows the agent to compute an optimal policy. DEC-POMDP framework (Bernstein *et al.*, 2000) has been designed for the same purpose in multiagent settings. Although DEC-POMDPs can find optimal policies, their complexity generally is so high that it is hard to apply on real applications. Moreover, the objective function is considered to be mono-criterion.

*This work is supported by the DGA (Direction Générale de l'Armement).

2V-DEC-MDPs have been introduced to coordinate a large number of agents in multi-criteria problems, extending the MDP framework, by considering local interactions with local full observability (which is a specific Partial Observability) (Mouaddib *et al.*, 2007). We consider 2V-DEC-MDPs as stochastic games to provide a mathematical foundation to improve results presented in (Boussard *et al.*, 2008). Game theory (Shapley, 1953) is a formalism for situations where agent's reward does not only depends on its actions but on the actions of all the agents. In stochastics games, Stackelberg Equilibrium (SE) is the theoretic concept of optimality to use with a leader.

In this paper, we adapt flocking rules to platooning, we express them as criteria in a 2V-DEC-MDP and prove the conditions under which the leader follows the SE. Finally, we demonstrate that our 2V-DEC-MDP leads to a near SE with weak complexity while reaching SE requires a very high complexity.

2 Background

2.1 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.

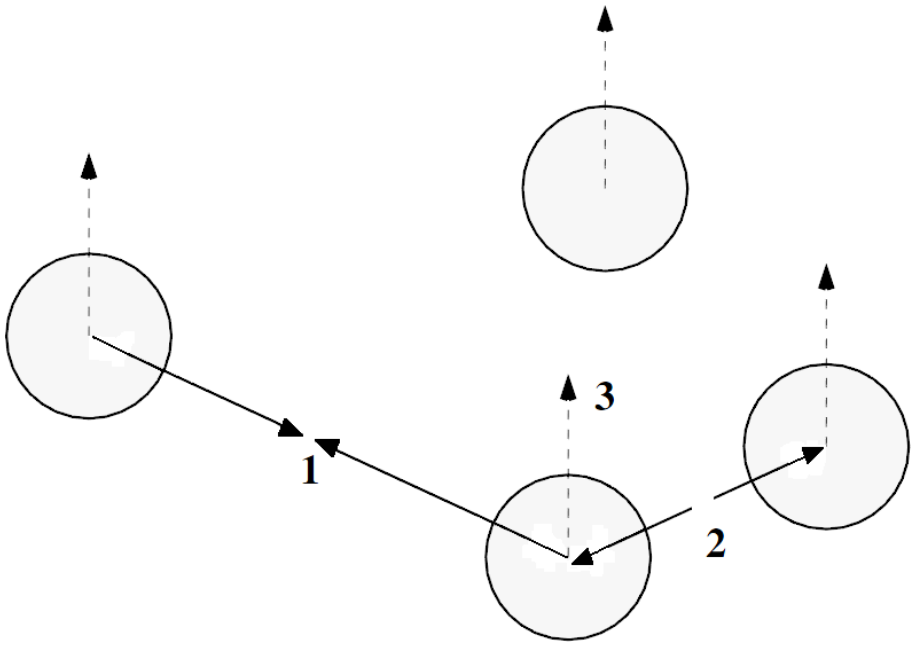

Figure 1: Flocking rules: (1) cohesion, (2) separation, (3) alignment

Despite the simplicity of those rules, agents manage to maintain the shape of the group. The main advantage of this approach is that it is fully decentralized, with no communication at all.

In this approach, 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, based on agents "perceived" so as we keep the locality property.

Those rules are the following:

1. Cohesion : steer to wait for agents behind it,
2. Separation : steer to avoid collisions with 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.

In the following, we show how we formalize these rules with an extended 2V-MDP (White, 1982) to multiagent settings using 2V-DEC-MDP (Mouaddib *et al.*, 2007).

2.2 Stochastic games

2V-DEC-MDPs are a sub-class of DEC-MDPs, which are equivalent to stochastic games (Shapley, 1953; Chaib-draa, 2008), so we can use stochastic games to estimate the quality of the 2V-DEC-MDP's behaviour. A stochastic game is defined by a tuple $\langle N, S, A, T, R\rangle$. We describe this tuple as follows:

- N is the number of agents (or players) taking part in the game,
- S is the set of states in which the game can be (a state describes the world and every player/agent),
- $A = \{A_1, A_2, \ldots, A_N\}$ is the set of possible actions for every agent i with $A_i = \left\{a_i^1, \ldots, a_i^{|A_i|}\right\}$,
- $R = \{R_1, R_2, \ldots, R_N\}$ is the set of reward functions of every agent with $R_i : S \times A_1 \times \ldots \times A_N \rightarrow \mathbb{R}$,
- $P : S \times A_1 \times \ldots \times A_N \times S \rightarrow [0, 1]$ is the transitions model between states, according to the joint actions.

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 is noted π_i and the joint policy for every agent in the game is $\pi = (\pi_1, \ldots, \pi_N)$. To estimate a strategy's value, it is necessary to know the utility for a given player to follow a given strategy. Let $\pi_1(s)$ be the chosen action by applying the π_1 policy on the state s. We can then write $\pi(s) = (\pi_1(s), \ldots, \pi_N(s))$ as the joint policy for this state. In this game, every agent i has (by definition) an immediate reward $R_i(s, \pi(s))$. We will be able to calculate $U_i^{\pi(s)}(s)$ the expected utility for an agent i if, in a state s, every agents apply π (with $\beta \leq 1$):

$$U_i^{\pi}(s) = R_i(s, \pi(s)) + \beta \sum_{s' \in S} P(s, \pi(s), s') \cdot U_i^{\pi}(s')$$

In a game, a Stackelberg Equilibrium (Stackelberg, 1952) (SE) is a situation where the leader of a group knows that it is the leader. It makes decisions, and its followers apply BR the best response (the best decision) according to this decision. The leader can then estimate the reactions of the other agents, and makes the decision which will bring it the best reward (or the best group reward), according to those reactions. SE is the theoretic concept of optimality to be used in such a situation.

Let us define the SE for a stochastic game. First, we define:

$$BR(\pi_i^*) = \{(\pi_1, ..., \pi_n) | \forall j \neq i, \pi_j \in BR(\pi_1, ..., \pi_i^*, ..., \pi_n)\}$$

Let $BR(\pi_i^*)$ be the set of best joint policies $(\pi_1, ..., \pi_n)$ for players 1 to n, knowing that the leader i apply the optimal policy π_i^*. If, for every state $s \in S$, the leader's policy π_i^* respects Eq.1, then we have SE.

$$\pi_i^* = \underset{\pi_i}{\text{Argmax}}(\min_{\pi \in BR(\pi_i)} U_i^{\pi}(s)) \tag{1}$$

2.3 2V-DEC-MDP

In (Mouaddib *et al.*, 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. Assuming without loss of generality that all agents are identical, a 2V-DEC-MDP is a set of 2V-MDPs, one per agent. A 2V-MDP is composed by an off-line part (an MDP (Puterman, 1994)) and an on-line part to adapt its actions with the other agents (in the neighborhood).

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

- S a set of states,
- A a set of action,
- $T : S \times A \times S \rightarrow [0; 1]$, 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 an infinite horizon. With $\gamma \leq 1$, the optimal value function V^* of a state s is defined by:

$$V^*(s) = \max_{a \in A}(R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') \cdot V^*(s')), \forall s \in S$$

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

$$\pi^*(s) = \operatorname*{Argmax}_{a}(R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') \cdot V^*(s')), \forall s \in S$$

This MDP represents the behaviour of an agent ignoring its neighbors (we will consider the other agents during the on-line part).

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 social impact are:

- ER for the individual reward (using the previous MDP),
- JER for the group interest,
- JEP for the negative impact on the group.

The on-line part is a sequence of small DEC-MDPs with a special transition model. During this part, an agent builds a new DEC-MDP after each decision it takes (only considering agents in its neighborhood and planning on a short horizon, less than 3). In this DEC-MDP, the probability for an agent i to go from a joint state s to a joint state s' only depends on a_i.

Deriving a policy for this DEC-MDP consists of solving a multi-criteria Bellman equation based on an Augmented Reward $AR = (ER, JER, JEP)$. To solve this equation, a regret based value iteration using $LexDiff$ operator (Mouaddib *et al.*, 2007) has been designed.

For each possible policy π_i, $LexDiff$ builds a vector $v = (ER(\pi_i), JER(\pi_i), JEP(\pi_i))$ 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 uses a leximin operator to find the best vector (it permutes those utilities vectors so that each vector (v^1, v^2, v^3) be such that $v^1 \geq v^2 \geq v^3$. The best vector (and so the best policy) 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).

To use flocking rules in a 2V-DEC-MDP, we translate the three rules into three formulae to parameterize each 2V-MDP. We consider ER as the alignment criterion, JER as the cohesion criterion and JEP as the separation criterion.

3 Platooning as a 2V-DEC-MDP

We use notations from sec.2.2, with $s = (s_1, ..., s_n)$, s_i being the part of s relative to the i^{th} agent and $s_1, ..., s_{i-1}, s_{i+1}, ..., s_n$ the states of the neighbors of i (we define the neighborhood for an agent i as the set of (detected) agents who can interact with i). In the platooning problem, s_i is the location of i (so we can say a state is "near" an other state, for exemple).

The ER function will represent the "alignment" into the group, JER will represent the "cohesion" and JEP will represent the "separation". We assume in those functions that an agent i is solving the 2V-MDP. Moreover, we only consider a_i the action of i in the transition function (and $P(s, a_i, s')$ only changes s_i to s'_i and leaves the s_j unchanged).

3.1 Alignment

An agent does not have the same objectives whether it is a leader or a follower. Indeed, a leader will move in the direction of its objective, while a follower will follow the agent in front of it, so an agent have to choose which equation to follow before resolving its 2V-MDP. 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. Otherwise, it chooses ER_2.

$$ER(s, a_i) = \sum_{s' \in S} P(s, a_i, s') ER_k(s') \quad k = 1, 2, 3 \tag{2}$$

Depending on the situation, ER_k are defined by:

$$\begin{aligned} ER_1(s) &= V^*(s_i) \\ ER_2(s) &= -\min_{s_j \in face_i(s)} distance(s_i, back_i(s_j)) \\ ER_3(s) &= -distance(s_i, back_i(s_{leader})) \end{aligned}$$

with $face_i(s) = \{s_j \in s | distance(s_j, objective) \leq distance(s_i, objective)\}$, $back_i(s_j)$ the next state s_i behind s_j reachable for agent i and $V^*(s)$ the function computed during the off-line part. In ER_2 and ER_3, we use $back(target)$ instead of $target$, because the agent wants to go behind its target.

3.2 Separation

$$JEP(s, a_i) = \sum_{s' \in S} \left[P(s, a_i, s') \cdot \sum_{j=1}^{N} \left(\frac{\sum_{a_j} P(s, a_j, s'') \cdot c(s'_i, s''_j)}{|A_j|} \right) \right] \tag{3}$$

With $c(s_i, s_j) = \{C$ if $s_i = s_j, 0$ otherwise$\}$ and C a constant equal to the cost of a collision between two agents.

3.3 Cohesion

$$JER(s, a_i) = \sum_{s' \in S} (P(s, a_i, s') \cdot K_i(s')) \tag{4}$$

Where $K_i(s')$ is the function which estimates the group gain at a state s' and gives a reward if at least one agent is behind s_i:

$$K_i(s') = \begin{cases} r > 0 \text{ if } \exists s_j \in s', j \neq i \text{ such as } isBack(s_j, s'_i) = true \\ 0 \text{ otherwise} \end{cases}$$

and where $isBack(s_1, s_2)$ is a function which returns true if s_1 is behind s_2.

3.4 Computation of a policy

Solving the platooning problem consists in deriving a policy from a 2V-DEC-MDP. Our approach is mainly based on local interactions allowing us to compute ER, JER, JEP. Thus we use the near-optimisation operator $LexDiff$ to solve the modified Bellman equation and thus to derive a policy.

4 Stackelberg equilibrium in 2V-DEC-MDPs

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

4.1 Detecting Stackelberg equilibrium

Theorem 1
An agent using a 2V-MDP is following an SE if and only if each criterion leads to an SE.

Proof. We aim to show that each criterion leads to an SE if and only if applying $LexDiff$ on those criteria leads to an SE.

Here, the agent's policy is built on the fly by $LexDiff$: for each state, it chooses an action to apply. Because the SE 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 $Q_c^{\pi}(s, \pi_i(s))$ be the utility for an agent i for applying its policy π_i when the system is in the state s and the joint policy is π, according to c. If we assume that each criterion c leads to an SE, we have:

$$\min_{\pi^* \in BR(\pi_L^*)} Q_c^{\pi^*}(s, \pi_L^*(s)) \tag{5}$$

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

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

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

And, we know that $LexDiff$ seeks the action which minimizes the biggest regret. Minimizing regret relative to a criterion is equivalent to maximizing utility relative to this criterion. So $LexDiff$ seeks the lowest utility maximizing action. In other words, because it is a local SE, we have:

$$\begin{aligned} U_L^{\pi}(s) &= LexDiff_{\pi_L}(\overrightarrow{Q}_{criteria}) \\ &= \max_{\pi_L} \left[\min_{Q_c \in \overrightarrow{Q}_{criteria}} Q_c^{\pi}(s, \pi_L(s)) \right] \end{aligned}$$

We can also deduce the following equality:

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

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

$$\min_{\pi^* \in BR(\pi_L^*)} U_L^{\pi^*}(s) \tag{9}$$

$$= \max_{\pi_L}(\min_{\pi \in BR(\pi_L)} U_L^{\pi}(s)) \tag{10}$$

So the $LexDiff$ operator leads to an SE. □

5 Adapting the 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 necessarily lead to a Stackelberg equilibrium.

Proof. According to theorem.1 We aim to show that ER, JER or JEP are not an SE. Using Eq.1, showing that ER, JER of JEP are not SE means to show one of the following:

$$\min_{\pi^* \in BR(\pi_L^*)} ER^{\pi^*}(s, \pi_L^*(s)) \neq \max_{\pi_L}(\min_{\pi \in BR(\pi_L)} ER^{\pi}(s, \pi_L(s))) \tag{11}$$

$$\min_{\pi^* \in BR(\pi_L^*)} JER^{\pi^*}(s, \pi_L^*(s)) \neq \max_{\pi_L}(\min_{\pi \in BR(\pi_L)} JER^{\pi}(s, \pi_L(s))) \tag{12}$$

$$\min_{\pi^* \in BR(\pi_L^*)} JEP^{\pi^*}(s, \pi_L^*(s)) \neq \max_{\pi_L}(\min_{\pi \in BR(\pi_L)} JEP^{\pi}(s, \pi_L(s))) \tag{13}$$

We will show 11. An agent only following ER will try to maximize the ER value so we have, with π_L^* the leader's policy resulting from the ER criterion:

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

However, the equation:

$$ER(s, \pi_L^*(s)) = \sum_{s' \in S} (P(s, \pi_L^*(s), s') \cdot V^*(s'))$$

supposes that no joint policy π is assumed for agents in the neighborhood when computing the value of ER. Indeed, $V^*(s')$ 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:

$$ER_{2VMDP} = ER(s, \pi_L^*(s)) = \max_{\pi_L} ER(s, \pi_L(s))$$

while the equation for an SE will be:

$$ER_{SE} = \max_{\pi_L}(\min_{\pi} ER^{\pi}(s, \pi_L(s)))$$

We then have $ER_{2VMDP} \geq ER_{SE}$ and thus :

$$\min_{\pi^* \in BR(\pi_L^*)} ER^{\pi^*}(s, \pi_L^*(s)) \geq \max_{\pi_L}(\min_{\pi \in BR(\pi_L)} ER^{\pi}(s, \pi_L(s)))$$

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

5.1 Rewriting criteria

According to theorem 1 and 2, the leader does not necessarily follow an SE. We show how to adapt the criteria and its effects on the complexity in order to reach an SE.

5.1.1 ER criterion (Eq.11):

With $d(s, g)$ the distance between s and the goal g if we don't know what the other agents do, and $d_\pi(s, g)$ the distance between s and g knowing the 1 to N agents' policies, we have:

$$ER(s, \pi_L(s)) = \sum_{s' \in S} P(s, \pi_L(s), s') \cdot V^*(s')$$

In our problem, their is only one reward, placed on the platoon's objective, so $V^*(s')$ only depends on the distance to the objective, so we can write (with $\alpha > 0$):

$$ER(s, \pi_L(s)) = \sum_{s' \in S} P(s, \pi_L(s), s') \cdot (-\alpha d(s'_L, g))$$

Because a_i only changes s_i to s'_i but leaves the s_j unchanged, we can use $P(s_i, a_i, s'_i)$. So, taking the other agents' decisions into account, we can write:

$$ER^{\pi}(s, \pi_L(s)) = \sum_{s' \in S} (P(s, \pi_L(s), s') \cdot (-\alpha d_{\pi}(s'_L, g)))$$

$$\text{With } d_{\pi}(s'_L, g) = \sum_{s'} \left[\left(\prod_{j=1}^{N} P(s_j, \pi_j(s_j), s'_j) \right) \cdot d_{s'}(s'_L, g) \right]$$

Where $d_{s'_1,\ldots,s'_N}(s, g)$ 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 assume the worst answer from the other agents and write:

$$\pi^*_L = \underset{\pi_L}{\text{Argmax}}(\min_{\pi \in BR(\pi_L)} ER^{\pi}(s, \pi_L(s)))$$

5.1.2 JER criterion (Eq.12):

We said that JER was the following, with $K_i(s)$ as defined in section 3.3:

$$JER(s, \pi_L(s)) = \sum_{s' \in S} [P(s, \pi_L(s), s') \cdot K_L(s')]$$

Taking the other agents' policies into account, we write:

$$JER^{\pi}(s, \pi_L(s)) = \sum_{s' \in S} (P(s, \pi_L(s), s') \cdot K^{\pi}_L(s'))$$

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

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

in

$$K^{\pi}_i(s) = \sum_{s' \in s_b(s)} \left[\prod_{j=1}^{N} P(s_j, \pi_j(s_j), s'_j) \right]$$

To maximize the criterion, we will have:

$$\pi^*_L = \underset{\pi_L}{\text{Argmax}}(\min_{\pi \in BR(\pi_L)} JER^{\pi}(s, \pi_L(s)))$$

5.1.3 JEP criterion (Eq.13):

The JEP criterion is the following, with $c(s_i, s_j)$ the function which represents the cost of a collision between two agents:

$$JEP(s, \pi_L(s)) = \sum_{s' \in S} \left[P(s, \pi_L(s), s') \cdot \sum_{j=1}^{N} \left(\frac{\sum_{a_j} P(s, a_j, s'') \cdot c(s'_L, s''_j)}{|A_j|} \right) \right]$$

We can rewrite JEP to consider the other agents' policies:

$$JEP^{\pi}(s, \pi_L(s)) = \sum_{s' \in S} \left[P(s_L, \pi^*_L(s_L), s'_L) \cdot \sum_{j=1}^{N} P(s_j, \pi_j(s_j), s'_j) \cdot c(s'_L, s'_j) \right]$$

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

$$\pi^*_L = \underset{\pi_L}{\text{Argmax}}(\min_{\pi \in BR(\pi_L)} JEP^{\pi}(s, \pi_L(s)))$$

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

6 Complexity

We will now compare the complexity of our initial formalism to the one with an SE. We will make this comparison on the JEP criterion (those results are in the same level of complexity for ER and JER).

6.1 Initial criteria

Complexity for computing $JEP(s, a)$ is $O(S \cdot N \cdot A)$, with N the number of agents, S the number of states and A the number of actions an agent can do. Moreover, to find the best action, we will have to compute A times JEP. The complexity for this criterion will then be $O(S \cdot N \cdot A^2)$.

So, complexity for finding the best action according to JEP is polynomial in S, N and A which is easy to compute.

6.2 Stackelberg adapted criteria

For a fixed joint policy, a leader will have to compute one value. The complexity for computing this value is $O(S \cdot N)$. The joint policy being not known, the leader will have to compute X values (each one in $O(S \cdot N)$) to estimate the value of its own policy, X being the size of $BR(\pi_L)$.

In the worst case, the value of X is PI^N, PI being the number of possible policies for one agent. Because a policy is a mapping from S to A, we have $PI = A^S$ but once reduced to the actual state we have $PI = A$, so $X = A^N$. The complexity for computing the value of one π_L is then $O(A^N \cdot S \cdot N)$. In order to find its optimal policy, the leader will have to compute this cost Y times, Y being the number of its possible policies.

Finally, in a given state, the global complexity for finding π_L^* is $O(A^{N+1} \cdot S \cdot N)$, which is an exponential complexity. Thus, according to this criterion, complexity is much higher in the Stackelberg case.

Similary, complexity for ER and JER are polynomial with the initial criteria and exponential with the Stackelberg adapted criteria.

7 Experimental results and real robots tests

7.1 Experimental results: with and without SE

We made a simulator in order to test our formalism, where agents' behavior is directed by a 2V-DEC-MDP parametrized as follow:

- we use no communication at all,
- agents know the initial "map" of their environnement,
- agents can perceive neighbors in a short range,
- agents have full observation inside their neighborhood, and null observation outside it (but they do not know their neighbors' policies),
- a state contains position and orientation of the agent and its neighbors,
- actions are "go forward", "go backward", "turn left", "turn right" and "do nothing",
- actions "go forward" and "go backward" can lead with a small probability to slip on a side, while actions "turn" and "do nothing" have probability 1 to be a success,
- DEC-MDPs generated during the on-line part are solved at horizon 1,
- moving cost a little, being on the objective bring a big reward.

Because nearly all the planning task is done on-line, we made some tests in dynamic environnements, by replanning the off-line part AND the on-line part every time a change is perceived in the environnement (for exemple: a new obstacle appear). Results were good but sometimes agents fell into a deadlock: some agents perceived the change while some other didn't perceived it, so their objectives became conflicting. Those results won't be dicussed here but will probably be subject of futur works.

We made some tests on non-dynamic environnement. Fig.2 represents the situation on which we made our tests: circles are agents (with a black point indicating the agent's direction), polygons are locations where an agent can go and dark patches are obstacles. We made several tests:

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

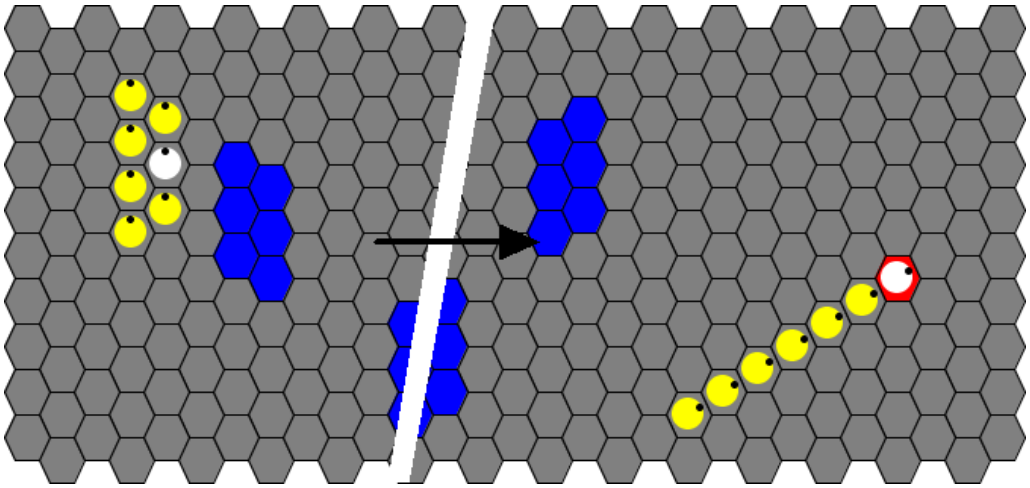

Figure 2: Test environment (start → end)

We summarize results of our tests in the following figures. They show results from the environment presented before, as an example, but we did some tests with other initial configurations and other environments.

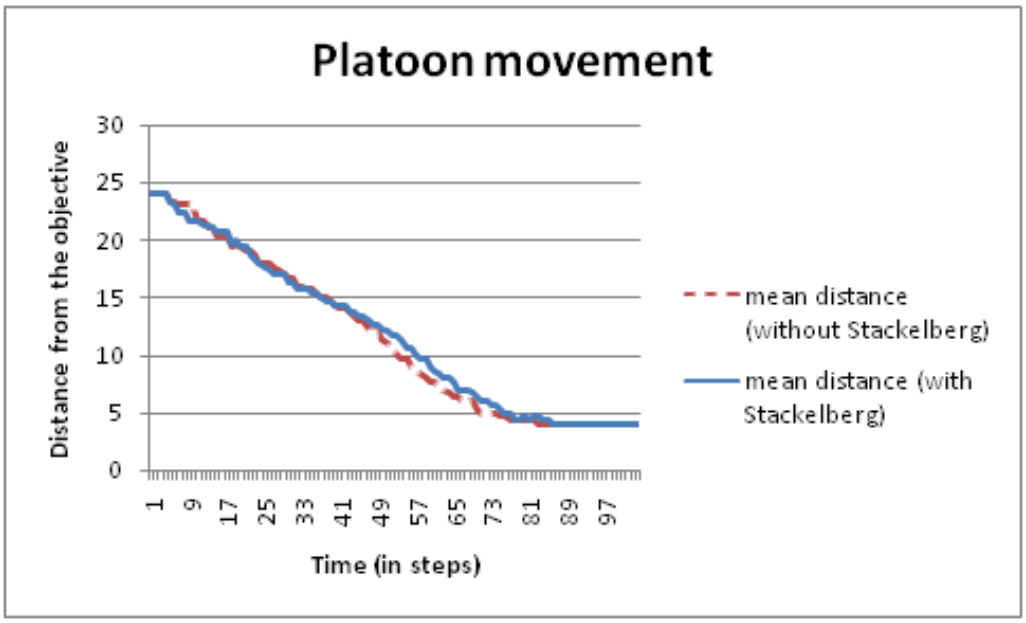

Figure 3: Distance to the objective

Fig.3 shows platoon's distance (according to its objective) evolution over time. Distance is a good mean to estimate the platoon's behavior quality, because it shows how fast the group is able to move. We can see that a platoon which considers a SE moves exactly at the same speed as a platoon which does not consider it. Thus, quality of the behavior following SE or not are almost the same.

There is an interesting point here: at the end of its evolution, the platoon moves a little faster without SE than with it. Why this difference ? With SE, 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 SE. Now, what about the complexity ?

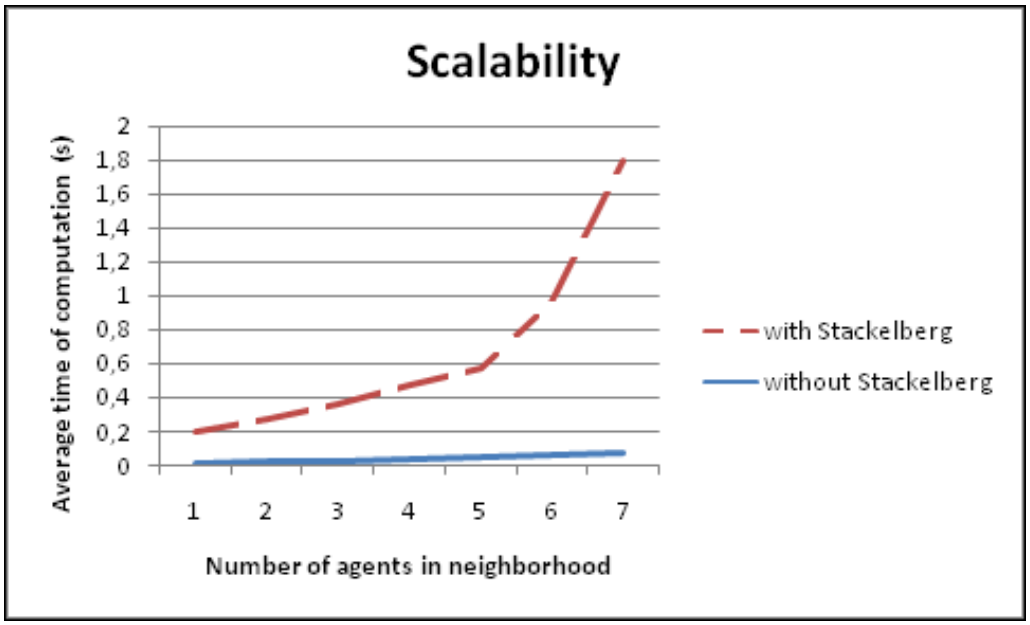

Figure 4: Computation time

Fig.4 shows the complexity according to the number of agents in the neighborhood. When we don't use an SE, the complexity seems to be proportional to the number of agents. This is trivial according to the equations of ER, JER and JEP: they depend on the agents in the neighborhood. Time needed grows 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 exponential with an SE. When more than 7 agents are in the neighborhood of the leader, time for computing an action becomes too high to be tractable by our simulation. Complexity is then much better with our formalism 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.3 and Fig.4. Thus, a platoon using our formalism acts as well as a platoon following SE, with a clearly better complexity.

7.2 Algorithms performance

The tests we made show that an agent which uses our formalism has a good behavior, but what about efficiency in time? In 2V-DEC-MDPs, policies are computed online, so computation times have to be good or the agents will have to stop between every action during computation of the next new policy. Several tests have been developped to show the average computation time of one joint action $(a_1, \ldots, a_N)$. Fig. 5 shows computation time according to how many agents are present (average results, on several simulations).

When agents begin scattered, complexity is nearly constant (from 0 to 0,01 seconds for 1 to 50 agents). With a lot of agents, environment is nearly full, so the probability for an agent to have interactions with other agents is high and complexity grows slowly. When agents start grouped, complexity grows slowly with the number of agents: with 50 agents, computation time of one action is less than 0,05 seconds.

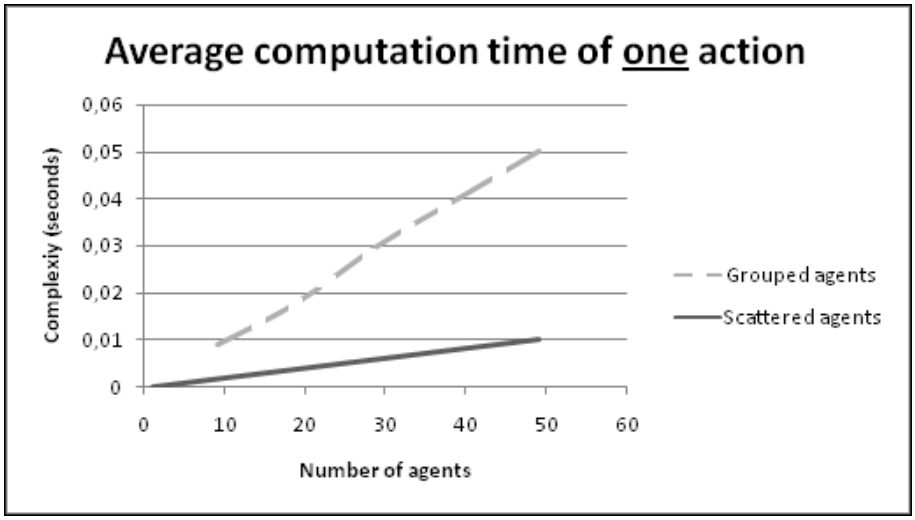

Figure 5: Complexity according to the number of agents

7.3 Tests on real robots (Koalas)

We made tests on 3 robots running a 2V-DEC-MDP. We parametrized them exactly like we parametrized our simulation. Local full observabilty has been artificialy made by using a central controler which observes the situation and give to every agent the informations it is supposed to know. We placed our robots on a line and put an objective in front of them. In Fig.6 are captions from a video of those tests.

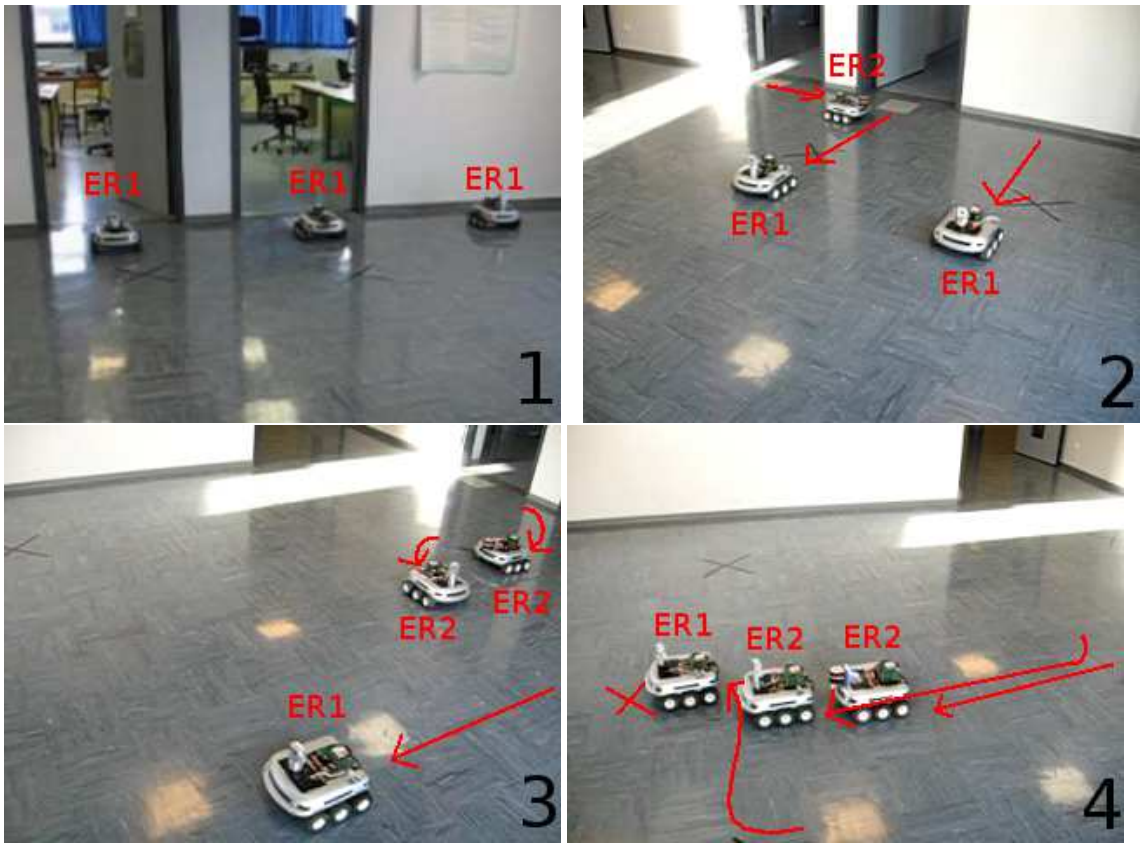

Figure 6: Platoon formation

When the test starts, every robot chooses the $ER1$ function and go toward its objective. After a few

seconds (caption 2), one robot chooses the $ER2$ function and start to follow a robot which is nearest to the objective than it. The two others continue tu use $ER1$ but, because of the JER function, they move slowly enough to let the third agent follow them. In caption 3, we can see only one agent is still using $ER1$ while the two others use $ER2$.

At this point, the platoon starts to emerge from those interactions, and the simulation only started 37s ago. The leader moves slowly because of JER and the two other agents follow it. In caption 4, the platoon is fully made and the leader reached the objective. Because of JEP, the two followers move very slowly to not take any risk and stop just near the leader.

Many other initial configurations were considered and we can see that, for each configuration, robots fully form a platoon after some moves.

8 Conclusion

In this paper we addressed the platoon formation problem with no cummunication, partial observability and stochastic actions. We shown how to transform this problem into a flocking problem and we formalized it using 2V-DEC-MDP which is the most suitable decision model in this context. To evaluate the performance of our approach, we related 2V-DEC-MDP to stochatic game and we used SE as a theoretic concept for optimality. We shown that the initial 2V-DEC-MDP leads to a near SE with a weak complexity while its adapted version can lead to SE with a very high compelxity.

In future work, we will study the impact of adding human controlled agents into the platoon or impredictible events. We will add a learning layer to avoid deadlocks.

References

BERMAN S., HALASZ A., KUMAR V. & PRATT S. (2007). Bio-inspired group behaviors for the deployment of a swarm of robots to multiple destinations. volume International Conf. on Robotics and Automation, Rome, Italy: IEEE.

BERNSTEIN D. S., ZILBERSTEIN S. & 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*, p. 32–37, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

BOUSSARD M., BOUZID M. & 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. (2008). *Processus Décisionnels de Markov en Intelligence Artificielle*, volume 1, chapter 4. Groupe PDMIA.

MICHAUD F., LEPAGE P., FRENETTE P., LETOURNEAU D. & GAUBERT N. (2006). Coordinated maneuvering of automated vehicles in platoons. *ITS*, **7**(4), 437–447.

MOUADDIB A., BOUSSARD M. & 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.

WHITE D. J. (1982). Multi-objective infinite-horizon discounted markov decision processes. *Journal of mathematical analysis and applications*, **89**, 639–647.