Towards a Formal Framework for Multi-Objective Multi-Agent Planning

Abdel-Illah Mouaddib, Mathieu Boussard, Maroua Bouzid
Maréchal Juin, Campus II
BP 5186
Computer Science Department
14032 Caen Cedex, France
(mouaddib,mboussar,bouzid)@info.unicaen.fr

ABSTRACT

Multi-Objective Multi-Agent Planning (MOMAP) addresses the problem of resolving conflicts between individual agent interests and the group interests. In this paper, we address this problem by presenting a formal framework to represent objective relationships, a decision model using a Vector-Valued Decentralized Markov Decision Process (2V-DEC-MDP) and an algorithm to solve the resulting 2V-DEC-MDP. The formal framework of a Vector-Valued MDP considered uses the value function which returns a vector representing the individual and the group interests. An optimal policy in such contexts is not clear but in this approach we develop a regret-based technique to find a good tradeoff between the group and individual interests. To do that, the approach we present uses Egalitarian Social Welfare orderings that allow an agent to consider during its local optimization the satisfaction of all criteria and reducing their differences. The obtained result is a good balance between individual and group satisfactions where the local policies can lead to more global satisfying behaviors in some settings. This result is illustrated in many examples and compared to alternate local policies.

1. INTRODUCTION

Planning rationally with an individual agent involves optimizing the achievement of its objectives. Planning rationally with a group of agents by optimizing the achievement of the overall objectives would not necessarily be optimal, or even acceptable, for an individual agent within the group. When optimizing the overall behavior of a group of agents, one difficulty is to address conflicts between individual agents' interests and the group's interests [3, 8]. To deal with some aspects of optimization in such a context requires considering a social law of the group that leads to some satisfying solution. A well-known theoretical tool studying such multi-agent systems is Von-Neumann-Morgenstern (VN-M) game theory [9]. In this concept, the solution of each agent is to compute the optimal decision for itself under the assumption that all others are doing likewise. This multi-agent decision process leads to a Nash equilibrium: if any agent were to change its decision, its payoff would be reduced. This approach considers that there are no sociological constraints and that agents can egoistically maximize their own benefit. Group preferences introduce some social relationships among agents from rationality of the group [4]. There has been a lot of work on deriving group preferences from individual preferences [8]. The fundamental issue is, given arbitrary preference orderings for each individual in a group, there always exists a way of combining these individual preference orderings to generate a consistent preference ordering group.

The system we consider is composed of agents, each of which has a set of objectives to attain. Each agent uses an MDP to define a local policy to solve its multi-objective problem. The accomplishment of objectives of an agent can have an effect on the accomplishment of the objectives of other agents. This problem is directly related to COIN (Collective Intelligence) which is about the effect of an individual's actions on the general welfare where only local utilities are exhibited without defining a desirable global behavior [11]. The solution we propose of this problem is a formal framework to represent the objective relationships and a decision model using a 2V-DEC-MDP where the vector-value functions of agents represent the individual and the group interests. The resulting vector-valued MDP allows each agent to derive a local policy where individual and group interests are respected. Concerning the group interest, we distinguish between two criteria which, sometimes, could be non-additive. These criteria are: a positive effect on the group where an agent improves the satisfaction of some agents and a negative effect where an agent degrades the satisfaction of some agents (nuisance). The vector of values considered allows an agent to represent its individual satisfaction, its positive effect on the group and its negative effect on the group. Thus, to prefer a decision over another one, an agent needs an operator to compare between vectors of values [10]. To deal with this issue, we consider *social welfare orderings* over multi-attributes value functions borrowed from welfare economics: (1) maximizing the sum of all utilities of the member of society (utilitarian concept), and (2) minimizing differences between the utilities of the member of society (egalitarian concept). The first

AAMAS'07, May 14–18, 2007, Honolulu, Hawai'i, USA.
Copyright 2007 IFAAMAS.

978-81-904262-7-5 (RPS) ©2007 IFAAMAS

makes sense when all members contribute to an overall goal of the society. This is not the case in many multi-objective applications where objectives could potentially be conflicting. The second, egalitarian functions consider differences in individual welfare unjust and removes these differences. We use this concept to respect both individual and group interests.

2. MULTI-OBJECTIVE MULTI-AGENT PLANNING

We describe our formal framework considering the objectives of agents and their relationships and how to best act to solve this multi-objectives problem. MOMAP can be seen as a tuple $(A, \{O_i, \mathcal{R}_i, \mathcal{F}_i\}_{i \in A})$ where:

- A is a set of agents i.
- Each agent has a set of objectives O_i.
- $\mathcal{R}_i = \{FB_i, LB_i, F_i, L_i\}$ are four functions that assign, from the state s_i^t of agent i at time t, to each objective $o_i^j \in O_i$ respectively the set of objectives developed by other agents that facilitate the achievement of objective o_i^j, the set of objectives that limit objective o_i^j, the set of objectives that are facilitated by objective o_i^j and the set of objective that are limited by objective o_i^j. Functions FB_i, LB_i, F_i and L_i allows us to express relationships between objectives similar to the ones found in TAEMS [3] and they can be defined more formally as follows :
  - $FB_i(o_i^j, s_i^t) = \{o_a^b \in O_a | s_i^t \wedge o_i^j \models \perp, \exists s_i^{t+1} \in out(o_a^b) : s_i^{t+1} \models o_i^j\}$ where s_i^t is the current state and s_i^{t+1} is one of the possible states could be reached after the achievment of o_a^b represented by out(o_a^b). This set contains the objectives of the other agents which, when achieved, allow the achievment of o_i^j.
  - $LB_i(o_i^j, s_i^t) = \{o_a^b \in O_a | s_i^t \models o_i^j, \exists s_i^{t+1} \in out(o_a^b) : s_i^{t+1} \wedge o_i^j \models \perp\}$. This set contains the objectives of the other agents which, when achieved, make the achievment of o_i^j not possible (soft relation could be "making difficult") the achievment of o_i^j.
  - $F_i(o_i^j, s_i^t) = \{o_a^b \in O_a | s_i^t \wedge o_a^b \models \perp, \exists s_i^{t+1} \in out(o_i^j) : s_i^{t+1} \models o_a^b\}$. This set contains objectives enabled by the achievment of objective o_i^j.
  - $L_i(o_i^j, s_i^t) = \{o_a^b \in O_a | s_i^t \models o_a^b, \exists s_i^{t+1} \in out(o_i^j) : s_i^{t+1} \wedge o_a^b \models \perp\}$. This set contains objectives disabled by the achievment of objective o_i^j.
- $\mathcal{F}_i = \{gain_i, Degrade\_Cost_i, \{R_{ij}, Penalty_{ij}\}_{i,j \in A,\ i \neq j}\}$ are functions assigned to each agent i to assess its local and social satisfaction when achieving an objective o_i^j:
  - $Gain_i(o_i^j | FB_i(o_i^j, s_i^t))$ is the reward gained by agent i when achieving objective o_i^j knowing the state of the set of objectives $FB_i(o_i^j, s_i^t)$;
  - $Degrade\_cost_i((o_i^j | LB_i(o_i^j, s_i^t))$ is the cost of degradation on achieving objective o_i^j knowing the state of the set of objectives $LB_i(o_i^j, s_i^t)$;
  - $R_{ic}(o_c^l | o_i^j)$ is the gain rewarded by an agent c when achieving objective $o_c^l \in F_i(o_i^j, s_i^t)$ given the achievement of objective o_i^j by agent i. This function measures the contribution of agent i in the society ;
  - $Penalty_{ic}(o_c^k | o_i^j)$ is the opportunity cost of achieving objective o_i^j on objective $o_c^k \in L_i(o_i^j, s_i^t)$ achievement of agent c. This function measures the nuisance of agent i in the society. It's also used as a cost of discoordination between agents.

Functions $Gain_i$ and $Degrade\_cost_i$ concern the satisfaction of th same agent i that's why its possible to aggregate them into the same function named in general a conditional utility function where its elicitation [2] is a problem orthogonal to the one studied in this paper. However, functions R_{ij} and $Penalty_{ij}$ often concern different agents and their aggregation is sometimes not possible because they involve on different types of dependency and different preferences of agents.

Each agent develops an MDP M_i to optimally reach the set of objectives. At each step of decision t, an agent achieves an objective o_i^j by considering the relationships FB_i, LB_i, F_i and L_i which are assumed to be deduced from its local observation at this time.

3. ILLUSTRATIVE EXAMPLES

For the significance of MOMAP settings, we present some examples considered for illustration inspired from "robocup rescue" or robots sweeping an area.

3.1 Example 1: Coordinated robot motion

We consider the scenario of heavy traffic of robots as depicted in Figure 1(a). Robots should make a decision of their locations that make the traffic easy. The actions of each robots are north (n), south (s) , east (e) , west (o), north-east (ne), north-west (no), south-east (se), south-west (so) and the wait action (w) allowing the robot not to move. Except action (w), all the other actions are stochastic with (80%) to reach the target cell and 10% to reach one of each neighbours (the left and the right cells). When one of the neighbour cell doesn't exist, there is 10% chance to stay at the same cell. This decision making problem can be described by our approach. Indeed, for each robot, the decision to move to a location has a social effect since it can make easy or difficult the movements of the others robots. The example presented consists of a fleet of robots organized into four coalitions moving into a grid as presented in Figure 1. The goal is that each coalition (s, c, t and p) has an initial location (a corner) and should move to a final location (diagonally opposed corner) where robots should try to trade-off between the self-interested and the cooperative-directed behaviors. This example can be formalized as a MOMAP problem as follows. First, the functions we used in our approach are measured as follows R_{ij} is the gain in Manhattan distance gained by the coalition, $penalty_{ij}$ concerns the cost of potential collisions with robots while R_i is just the gain in the distance (distance$(current_cell, destination) - shortest_distance$) by robot i. The cost of collisions between robots of coalitions s, c, t, p are given in the following payoff matrix. This matrix represents at which degree an agent i is harmful for another. For example, a robot of coalition c is harmfull for a robot of coalition t with cost -3. Differently speaking, when a robot of coalition c collides with a robot of coalition t, it pays -3.

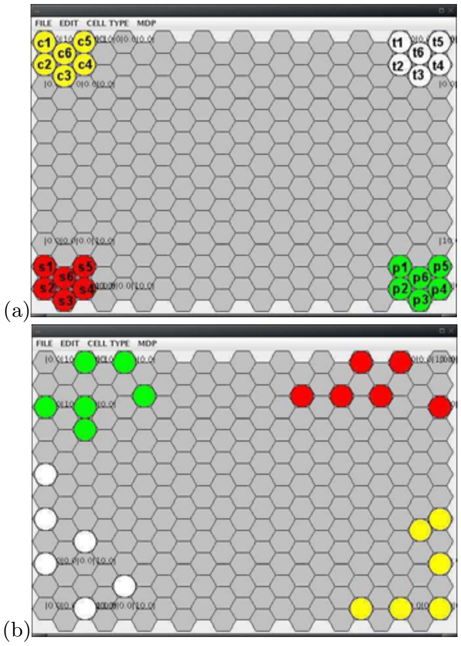

Figure 1: An example of robot traffic scenario: (a) initial state, (b) final state reached with MOMAP

	c	t	s	p
c	-1	-3	-1	-2
t	-1	-1	-2	-3
s	-3	-2	-1	-1
p	-2	-1	-3	-1

Second, the functions of $\mathcal{R}_i$ return at initial state s_{ini}, for example, for robot $c3$ acting with south-east (se) to reach the target cell (3,3) (the cells are numbered column by column): $F_{c3}((3,3),s_{ini}) = \{$ cells reached by action south (s) of $c6$, cells reached by action west of $c4$, cells reached by action east of robot $c2$ $\}$, $L_{c3}((3,3),s_{ini}) = \{$cells reached by action south of $c4$ $\}$ while $FB_{c6}((2,2),s_{ini}) = \{$ cells reached by actions south, south-west and south-east of $c3$ $\}$ and $LB_{c2}((2,2),s_{ini}) = \{$ cells reached by action wait of $c3$ $\}$. We can also generalize these sets to some compositions of objectives like $c3$ exits cell (2,2) and it will be occupied by $c4$ or $c2$ limit the achievement of objective $(2,2)$ by $c6$. How agents make decision is the concern of the next sections.

3.2 Example 2: Local spatial coordination

The aggregate motion of a fleet of robots in a limited space can be seen as a "Flocking" which is a model for group movement. MOMAP can be used as a framework to formalize flocking since it consists of three simple steering behaviors (rules): *separation* steer to avoid crowding local flockmates, *alignment* steer towards the average heading of local flockmates and *cohesion* steer to move towards the average position of local flockmates. MOMAP can formalize these three behaviors using a multi-criteria decision making where *separation* can be seen as a first criterion represented by a social reward measured by an average distance between an agent and the others, *alignment* can be seen as a second criterion represented by a local reward by counting the number of agents heading the same direction and *cohesion* can be seen as the third criterion representing the distance between the agent and the middle of the group. An illustration of this example is given in Section 6.

3.3 Example 3: Emergent coalition formation

Another example can be an emergent coalition formation from local behavior of agents where a group of agents should be splited into coalitions to achieve tasks. Each agent can sequentially make a decision on which tasks it can participate. Initially an agent can decide to participate to all tasks and progressively can decide to eliminate some tasks according to the participation of other agents. This distributed decision process can be formalized by MOMAP where a decision can be assessed according to the individual reward to participate to the achievement of a task and the social reward taking into account the difficulty or the facility to achieve the task with the participation of the other agents. An illustration of this example is given in Section 6.

4. DECISION MODEL: VECTOR-VALUED DEC-MDP

The decentralized decision process can be described by $< \times_{i\in A} S_i, \times_{i\in A} A_i, \{P_i, AR_i\}_{i\in A}, T >$, where each agent i develops a decision process $M_i =< S_i, A_i, P_i, AR_i, T >$ characterized by: (1) The set of states S_i where a state s_i^t contains the set of objectives achieved by an agent i up to time t. Other features could be considered in the state. An agent i has only a partial observability about the global state of the whole system represented by the relationships F_i, L_i, FB_i, LB_i which we assume derived from its local observation. (2) Each agent i has a set of actions $A_i = \{a_i^1, a_i^2, \ldots, a_i^{k_\alpha}, \mathsf{w}\}$, where a_i^j is an action to satisfy objective j, w is the action null that dictates to the agent to do nothing. When acting with an action a_i at state s_i^t, the system moves from s_i^t to $s_i^{t+1} = s_i^t \bigcup \{o_i^j\}$. (3) The dynamic of the process is given by a distribution probability $P_i(s_i^{t+1}, a, s_i^t)$ that is the probability to reach the state s_i^{t+1} when acting with action a at state s_i^t. (4) Functions $\mathcal{F}_i$ defined in MOMAP allow us to augment the reward function AR_i by introducing the effect on the other agents. The definition of AR_i, in our context, and the intuitions it conveys are given in the following. (5) T is the number of decision steps (Horizon). And, (6) the Bellman equation of M_i using the augmented reward function AR_i and the dynamic process P_i which we explain in detail in MOMAP context:

$$V_i(s_i^t) = AR_i(s_i^t) + \max_{a_i^k} \sum_{s_i^{t+1}} P_i(s_i^{t+1}, a_i^k, s_i^t) \cdot V_i(s_i^{t+1}) \quad (1)$$

$$V_i(s_i^T) = AR_i(s_i^T)$$

4.1 Probability of transition

To compute P_i, we introduce the probability PM_i to achieve an objective o_i^j at time t. The probability PM_i of an objective o_i^j at time t being in state s_i^t is the probability that all objectives of $FB_i(o_i^j, s_i^t)$ are achieved before t and no objective of $LB(o_i^j, s_i^t)$ is achieved before t:

$$PM_i(o_i^j, t) = \sum_d Pr(\delta_{o_i^j} = d) \cdot \sum_{(k_1,\ldots,k_n)\in[0,t-d]^n} \Pi_\alpha PM_\alpha(o_\alpha^l, k_\alpha) \cdot$$
$$\sum_{(x_1,\ldots,x_n)\in[0,t-d]^m} (1 - \Pi_\beta PM_\beta(o_\beta^l, x_\beta))$$

where α is an agent and $o_\alpha^l \in O_\alpha$ $s.t.$ $o_\alpha^l \in FB_i(o_i^j, s_i^t)$ and $n = |FB_i(o_i^j, s_i^t)|$, β is an agent and $o_\beta^l \in O_\beta$ $s.t.$ $o_\beta^l \in LB_i(o_i^j, s_i^t)$ and $m = |LB_i(o_i^j, s_i^t)|$ and Pr is a distribution probability of $\delta_{o_i^j}$ which is the duration of achieving objective o_i^j. Now, to define P_i we can see that the transition from one state to another depends on the objective achievement. Then, the probability to move to a new state is the probability to achieve the objective. More formally speaking: $P_i(s_i^{t+1}, a_i^j, s_i^t) = PM_i(o_i^j, t+1)$. This probability is similar to the ones developed in [3, 1].

4.2 Vector-Valued DEC-MDP for MOMAP

The value rewarded by an agent when achieving an objective depends on the objectives that facilitate and limit it and on the value rewarded to the other agents of the society represented by the agents for which the objective facilitates or limits their objectives. Consequently, we distinguish between different reward functions of an objective o_i^j: $R_i(o_i^j|FB_i(o_i^j, s_i^t), LB_i(o_i^j, s_i^t))$ that represents the immediate reward an agent gains when achieving its objective o_i^j knowing $FB_i(o_i^j), LB_i(o_i^j, s_i^t)$, the reward gained by the agent representing its contribution in the society of agents $JR_i(F_i(o_i^j), L_i(o_i^j, s_i^t)|o_i^j)$ and the opportunity cost measuring the cost of conflict or the degradation of the achievement of the other objectives $JPenalty_i$. This measure allows us to evaluate the cost an agent is willing to pay when its action can be in conflict with another action of another agent (cost of discoordination). It represents, also, the cost of nuisance of the agent. In the following, we will describe how those functions can be computed in an egalitarian society and used to calculate :

$$AR_i(s_i^t) = \begin{pmatrix} (R_i(o_i^j|FB_i(o_i^j, s_i^t), LB_i(o_i^j, s_i^t)) \\ JR_i(F_i(o_i^j)|o_i^j) \\ JPenalty_i(L_i(o_i^j, s_i^t)|o_i^j)) \end{pmatrix}$$

where o_i^j is the objective satisfied at time t. AR_i (a vector of rewards and costs) takes into account the self-interested criterion and the the cooperative criterion. We describe later how this augmented reward could be restricted to egoist or cooperative agents. The Bellman equation (equation 1) deals with the vector value function V_i which is a triplet (v_i^1, v_i^2, v_i^3) where v_i^1 is the expected value of the local satisfaction of agent i, v_i^2 is the expected value of the satisfaction of the group and v_i^3 is the expected value of nuisance of agent i. The value function V_i is of the same dimensionality as AR_i.

$$\begin{pmatrix} v_i^1(s^t) \\ v_i^2(s^t) \\ v_i^3(s^t) \end{pmatrix} = \begin{pmatrix} R_i(s^t) \\ JR_i(s^t) \\ JPenalty_i((s^t) \end{pmatrix} +$$
$$\vec{\mathbf{max}}_{a_i^j \in A_i} \sum_{s^{t+1}} P_i(s_i^{t+1}, a_i^j, s_i^t) \cdot \begin{pmatrix} v_i^1(s^{t+1}) \\ v_i^2(s^{t+1}) \\ v_i^3(s^{t+1}) \end{pmatrix} \qquad (2)$$

$\vec{\mathbf{max}}$ operator means that we have to maximize the first and second criteria and to minimize the third one. This leads to derive a policy π subject to: $V_i(s_i^t) =$

$$\begin{cases} R_i(o_i^j|FB_i(o_i^j, s_i^t), LB_i(o_i^j, s_i^t)) + \max_{a_i^k} \sum_{s_i^{t+1}} P_i(s_i^{t+1}, a_i^k, s_i^t).v_i^1(s_i^{t+1}) \\ JR_i(F_i(o_i^j, s_i^t)|o_i^j) + \max_{a_i^k} \sum_{s_i^{t+1}} P_i(s_i^{t+1}, a_i^k, s_i^t).v_i^2(s_i^{t+1}) \\ JPenalty_i(L_i(o_i^j, s_i^t)|o_i^j) + \min_{a_i^k} \sum_{s_i^{t+1}} P_i(s_i^{t+1}, a_i^k, s_i^t).v_i^3(s_i^{t+1}) \end{cases}$$

This equation means that an agent makes a decision that maximizes its local gain, and the potential gain of the agents y of objective $o_y^k \in F_i(o_i^j, s_i^t)$ and that minimizes the degradation of the gain of agents z of objectives $o_z^l \in L_i(o_i^j, s_i^t)$.

5. SOLVING 2V-DEC-MDP

The goal is to maximize the satisfaction of all individual criteria of agents by using the egalitarian concept. We introduce *social welfare orderings* over vector-value functions.

5.1 Egalitarian approach for MOMAP

We assume that all agents evolve with egalitarian laws. The egalitarian social welfare is then translated in our approach by: each agent acts to maximize its reward, the satisfaction of agents of objectives in $F_i(o_i^j)$ (maximizing the minimum profits in $F_i(o_i^j)$) and to minimize the nuisance on agents of objectives in $L_i(o_i^j)$ (minimizing the maximum opportunity cost in $L_i(o_i^j)$). Then, the satisfaction of an agent i when achieving an objective o_i^j is then assessed by:

$$JR_i(F_i(o_i^j)|o_i^j) = \max_{a_i^j \in A_i} \min_{b \neq i} \sum_{o_b^k \in F_i(o_i^j, s_i^t)} R_{ib}(o_b^k|o_i^j) \qquad (3)$$

$$JPenalty_i(L_i(o_i^j)|o_i^j) = \min_{a_i^j \in A_i} \max_{c \neq i} \sum_{o_c^k \in L_i(o_i^j, s_i^t)} Penalty_{ic}(o_c^k|o_i^j) \qquad (4)$$

Example 1. To illustrate equations 3 and 4, let consider agent 1 which has two actions a and b to achieve an objective o. This objective is related to objectives o_1 and o_2 of agent 2 and objectives o_3 and o_4 of agent 3 by the relationship F_1. When acting with action a or b, agent 1 rewards from agent 2 $R_{12}(o_1|o)+R_{12}(o_2|o)$ and $R_{13}(o_3|o)+R_{13}(o_4|o)$ from agent 3. Let these rewards be $10 = 3+7$ and $11 = 5+6$ for action a and $12 = 6+6$ and $5 = 3+2$ for action b. Equation 3 in the example is max(min(10,11),min(12,5)) which is 10.

In general, $R_i(o_i^j|FB_i(o_i^j, s_i^t), LB_i(o_i^j, s_i^t))$, is a conditional utility that sometimes is given by the equation:
$R_i(o_i^j|FB_i(o_i^j, s_i^t), LB_i(o_i^j, s_i^t)) =$
$\omega_1 \cdot gain_i(o_i^j|FB_i(o_i^j, s_i^t)) - \omega_2 \cdot Degrade\_Cost_i(o_i^j|LB_i(o_i^j, s_i^t)$
where functions $gain_i$ and $Degrade\_Cost_i$ are specific conditional utilities. In this paper, we assume that R_i is an aggregate linear function of $gain_i$ and $Degrade\_Cost_i$ because they concern the preference of the same agent. The rest of the paper describes how an agent constructs a policy about the sequence of achieving its objectives taking its effect on the society.

5.2 Mono-objective policies

Egoistic policy. An agent is egoist when it does not take into account how much utilities the other agents can potentially gain if it commits to the target objective. In this case, the Egoistic policy π_{eg} is then given by (i.e. p_1 in Figure 2):

$$\pi_{eg} = arg \max_{a_i^k} R_i(s_i^t) + \sum_{s_i^{t+1}} P_i(s_i^{t+1}, a_i^k, s_i^t) \cdot v_i^1(s_i^{t+1})$$

Optimistic completely cooperative policy. An agent is completely cooperative when its behavior is directed by the potential gain of the other agents. In this case, by using egalitarian definition of JR_i (equation 3), the optimistic completely cooperative policy is given by: $\pi_{oco} = arg$

$$\max_{a_i^j \in A_i} \min_{b \in A-\{i\}} \sum_{o_b^k \in F_i(o_i^j, s_i^t)} R_{ib}(o_b^k | o_i^j) + \sum_{s_i^{t+1}} P_i(s_i^{t+1}, a_i^j, s_i^t) \cdot v_i^2(s_i^{t+1})$$

Pessimistic completely cooperative policy. An agent is pessimistic completely cooperative when its behavior is directed by the potential opportunity penalty. In this case, by using egalitarian definition of $JPenalty_i$ (equation 4), the pessimistic completely cooperative policy is given by: $\pi_{pco} = arg \min_{a_i^j \in A_i} \max_{c \in A-\{i\}}$

$$\sum_{o_c^k \in L_i(o_i^j, s_i^t)} Penalty_{ic}(o_c^k | o_i^j) + \sum_{s_i^{t+1}} P_i(s_i^{t+1}, a_i^j, s_i^t) \cdot v_i^3(s_i^{t+1})$$

5.3 Regret-Based policy for MOMAP

The policy of an agent has to balance between its egoist and optimistic/pessimistic cooperative behaviors. To do that, we show how Equation 1 could be solved by deriving a policy reaching this balance. The resulting MDP is a Vector-Valued MDP in an egalitarian context where the overall goal of the agent is to maximize its local utility and the gain of the other agents and to minimize the opportunity penalty. The socially satisfying policy π_{ss} is, then, subject to: $V_i(s_i^t) =$

$$\begin{cases} \max_{a_i^j} R_i(s_i^t) + v_i'^1(s_i^{t+1}) \\ \max_{a_i^j} \min_{b \in A-\{i\}} \sum_{o_b^k \in F_i(o_{ij})} R_{ib}(o_b^k | o_i^j) + v_i'^2(s_i^{t+1}) \\ \min_{a_i^j} \max_{c \in A-\{i\}} \sum_{o_c^k \in L_i(o_i^j, s_i^t)} Penalty_{ic}(o_c^k | o_i^j) + v_i'^3(s_i^{t+1}) \\ \text{s.t. } v_i'^k(s_i^{t+1}) = \sum_{s_i^{t+1}} P_i(s_i^{t+1}, a_i^j, s_i^t) \cdot v_i^k(s_i^{t+1}), k \in \{1, 2, 3\} \end{cases}$$

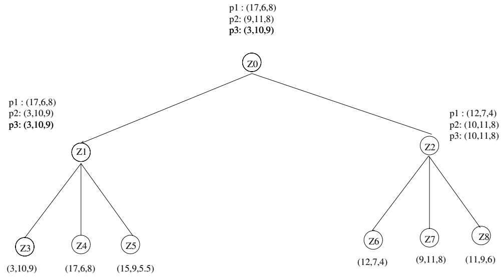

Figure 2: Mono-criterion policy computation

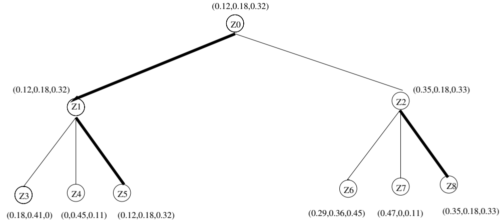

Figure 3: Regret-based algorithm computation

To derive the policy π_{ss} we take advantage of information obtained from the policies $\{\pi_{eg}, \pi_{oco}, \pi_{pco}\}$. To do that, let V_{eg}^*, V_{oco}^* and V_{pco}^* be the values of the initial state that an agent can expect to gain when it follows the policies $\pi_{eg}, \pi_{oco}, \pi_{pco}$ respectively (Example 2). These values are computed by using a standard dynamic programming technique. The following discussion concerns all systems where $(V_{eg}^*, V_{oco}^*, V_{pco}^*)$ is not a solution (multi-objective problems). In such cases, we use vector $(V_{eg}^*, V_{oco}^*, V_{pco}^*)$ as the best solution and the satisfying policy π_{ss} leads to values that are as close as possible to this vector. Then, the quality of the policy is measured by the distance between its values and the values of vector $(V_{eg}^*, V_{oco}^*, V_{pco}^*)$. The difficulty is to define an admissible measure to assess the distance between the vector value of a policy and vector $(V_{eg}^*, V_{oco}^*, V_{pco}^*)$. A policy using an Euclidean distance, as all the weighed sum approaches, does not take into account the fluctuation on the criteria. We consider, then, another measure similar to Tchebychev norm.

Definition 1. A vector of regrets of vector value (v^1, v^2, v^3) is given by a new vector: $(v^{r,1}, v^{r,2}, v^{r,3}) = (\frac{|v^{r,1} - V_{eg}^*|}{V_{eg}^*}, \frac{|v^{r,2} - V_{oco}^*|}{V_{oco}^*}, \frac{|v^{r,3} - V_{pco}^*|}{V_{pco}^*})$, $V_i^* > 0$, $i \in \{eg, oco, pco\}$ because at least one terminal state has a non-zero reward.

Definition 2. A solution (v^1, v^2, v^3) is preferred over a solution(v'^1, v'^2, v'^3) when $(v^{r,1}, v^{r,2}, v^{r,3})$ is preferred over $(v'^{r,1}, v'^{r,2}, v'^{r,3})$ using a leximin order (Leximin Regret).

The motivation of these definitions is to make leximin independent from the scale factors and to perform the leximin order on values with more global sense among criteria rather than values with a local sense as in Tchebychev norm. That's why the use of the ratio competitive regret measure improves standard leximin techniques [10].

Example 2. Let consider the example given in Figure 2 where states Z_i have a vector reward function where the scores of dimension 2 and 3 are respectively computed from equation 3 and 4 as depicted in Example 1. p_1, p_2, p_3 are mono-criterion policies which prefer a vector over another by comparing respectively the value of criterion 1, 2 and 3. From p_1, p_2, p_3 we derive an ideal vector value (17,11,9) which is used to assign to each state $Z_3, Z_4, Z_5, Z_6, Z_7, Z_8$ its regret vector value. Regret vector values of states Z_2, Z_1, Z_0 are computed using a lexicographic order on the regret vector values of $Z_3, Z_4, Z_5, Z_6, Z_7, Z_8$ as depicted in Figure 3.

Regret-Based Algorithm to solve 2V-DEC-MDP

```
1. Let $\mathbf{r}(s^t)$ be the vector of regrets of $(R(s^t), JR(s^t), JPenalty(s^t))$
```

```
2. Compute $V^*_{eg}, V^*_{oco}, V^*_{pco}$ of mono-objective policies.
3. **For all** terminal state $s^T$ **do**
4.     $(v^{r,1}(s^T), v^{r,2}(s^T), v^{r,3}(s^T)) = \mathbf{r}(s^T)$ using equation 1
5. **For** t = T-1 **downto** 1 **do**
6.     **For** all states $s^t$ **do**
7.         $(v^{r,1}(s^t), v^{r,2}(s^t), v^{r,3}(s^t)) = \mathbf{r}(s^t) + \mathsf{LexminRegret}_{a \in A_i} \sum_{s^{t+1}} P_i(s^{t+1}, a, s^t) \cdot (v^{r,1}(s^{t+1}), v^{r,2}(s^{t+1}), v^{r,3}(s^{t+1}))$
8.         $\pi_{lex,r}(s^t) = argLexminRegret_{a \in A_i} \sum_{s^{t+1}} P_i(s^{t+1}, a, s^t) \cdot (v^{r,1}(s^{t+1}), v^{r,2}(s^{t+1}), v^{r,3}(s^{t+1}))$
9. **return** $\pi_{lex,r}$
```

A policy $\pi_{lex,r}$ allows an agent to reduce the regret ratio for each criterion by considering the leximin Regret.

Theorem 1. *The algorithm to solve 2V-DEC-MDP using lex, r measure is polynomial*

Proof. The complexity of the algorithm is polynomial because of n polynomial mono-optimisations to compute V^*_i of criterion i and a further optimization using the values $v^{lex,r}$.□

Theorem 2. *The policy $\pi_{lex,r}$ is Pareto optimal.*

Proof. This policy uses the leximin order which leads to a Pareto optimal solution [10]. □

Theorem 3. *If the vector value $(V^*_{eg}, V^*_{oco}, V^*_{pco})$ is a solution then it's the vector value of the initial state using $\pi_{lex,r}$.*

Proof. The assigned vector $(V^{r,eg}, V^{r,oco}, V^{r,pco})$ to $(V^*_{eg}, V^*_{oco}, V^*_{pco}))$ using the ratio competitive regret is (0,0,0). This vector is lexicographically preferred over the others. □

6. ILLUSTRATION AND SOME EMPIRICAL RESULTS

In this section, we present some illustrative examples showing how MOMAP framework can help in formalising and solving such problems. Some of those examples have been implemented using MOMAP where the obtained results are presented in the experimental section.

6.1 Example 1 : Coordinated motion

In this section we show how robots evolve in the grid of the example, depicted in Figure 1, following the policy derived by our approach. Given the initial state of the grid we can develop the vector of values of actions of each robot. We consider robots $s5$, $c4$, $p5$, $t4$ for illustration of how robots make their decisions according to the vector-value function.

Regarding the vector of values of Table 1, robot $s5$ performs action e (by symmetry robot $t2$ performs o), $c4$ performs action s (by symmetry $p2$ performs action n) and $t4$ performs action w ($s2$ performs action w).

	$s5$	$c4$	$p5$	$t4$
n	$(-1, 2, -3)$	$-$	$(\mathbf{0}, \mathbf{2}, \mathbf{-2})$	$-$
no	$(-2, 2, -2)$	$-$	$-$	$-$
ne	$(0, 2, -6)$	$(-2, 2, -6)$	$(-1, 1, -2)$	$-$
s	$-$	$(\mathbf{-1}, \mathbf{2}, \mathbf{-1})$	$-$	$(-2, 2, -6)$
so	$-$	$(0, 2, -6)$	$-$	$(-1, 0, -6)$
se	$(-3, 2, -2)$	$(-2, 2, -1)$	$-$	$-$
w	$(-2, 0, 0)$	$(-2, 0, 0)$	$(-4, 0, 0)$	$(\mathbf{-4}, \mathbf{0}, \mathbf{0})$
e	$(\mathbf{-1}, \mathbf{2}, \mathbf{-1})$	$(-1, 2, -3)$	$-$	$-$

Table 1: Vector values for actions of robots s5,c4,p5 and t4 (preferred vectors are in bold)

The decisions made at the next steps are similar. The global behavior is close to the optimal one where robots in the first steps move towards the middle of the grid (high value of the first criterion, individual reward, and a weak value of the third criterion, penalty) and then they rotate around the cells in the middle of the grid (high value of the penalty criterion and a weak value of individual reward, as shown by Table 1) leading to the final state (Figure 1(b)) where a good balance between reaching the destination and avoiding collisions is achieved. If we use an egoistic policy, the robots tend to go toward cells in the centre that provoke many collisions because this policy uses only the first criterion without being concerned with the other criteria. The optimistic approach leads to behaviors of agents that make a long time to reach destinations because the policy uses only the second criterion without being concerned with the other criteria. This leads to an agent dedicated to the satisfaction of the other agents. It considers its own satisfaction only when the satisfaction of the others is respected. The pessimistic policy leads to a very careful behavior. In many situations, agents prefer waiting rather than executing any other actions because they consider only the penalty of potential collisions. Optimistic and pessimistic behaviors are not convergent.

6.1.1 Performance Analysis

The criteria used to compare the policies are: the number of collisions (discoordination) and time needed by a coalition to reach destination. Table 1 shows that the egoistic policy leads to many collisions and conflicting decisions, while full optimistic cooperative policy that can be seen as an egoistic group policy for a coalition leads to less conflicts but it takes a long time to reach the destination. The full pessimistic policy allows a coalition to avoid collisions and conflicting decisions but it takes a very long time to reach destination. Our approach is an equilibrium among all the criteria since it takes a reasonable time to reach destination (close to egoistic policy) and it reduces significantly collisions and conflicting decisions (close to full pessimistic cooperative policy). As we can see the policies derived from local MDPs are not completely coordinated (4 conflicts). $JPenalty$ measures the cost the agents are willing to pay when they accept that their local policies are discoordinated in some states.

6.1.2 Comparaison with Classical Approaches

We have compared (using the example of Figure 1) our approach with approaches using cooperation structures and communication mechanisms similar to GPGP or PGPP [5]. This approach is based on an incremental coordination of

policy	collisions	steps to reach destination
π_{eg}	40	41
π_{ocop}	9	137
π_{pcop}	0	289
$\pi_{lex,r}$	4	73

Table 2: Results of different policies

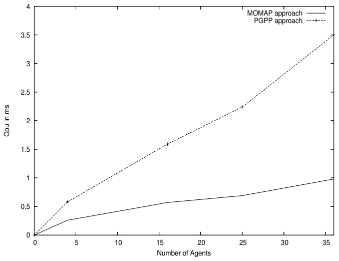

Figure 4: A comparison of time to reach destinations

partial plans of agents to solve conflicts between all agents. Each agent builds a partial plan that it broadcasts to collect and solve conflicts. When the coordination of the partial plan successes, another planning step followed by a coordination step cycle is performed. This incremental processing is repeated until a full plan is constructed and coordinated or a coordination step fails. Our approach converges quickly and it reduces the number of conflicts (Figures 4, 5). The PGPP and similar approaches are slower because they spend more time in collecting and solving conflicts.

6.2 Example 2 : Local spatial coordination

In this experiment, we consider a fleet of robots to coordinate their movement taking into account obstacles. The coordinated movement is the result of local decisions as depicted in Figure 6. The objective is that robots can move towards their destinations (individual reward), reduce the disturb of the movement of the other robots (penalty) and help them to easily reach their destination (social reward).

In Figure 6, we can see that agent are, initially, grouped together at the bottom of (Figure 6(a)). The first step they will be self-organized into groups to move, in a coordinated way, towards their destination (Figure 6(b,c)). The second step, agents have to cross towards two narrow corridors where conflicts can occur between agents of the same group (Figure 6(c)). The third step, agents in a distributed way have avoided conflicts by making coordinated decisions (Figure 6(d)) where an agent crosses the corridor while the other makes a decision to be far from the corridor. The fourth step where we see in Figure 6(e,f) agents moving towards their destination. At destination agents surround the destination which is a good equilibrium between reaching destination and avoiding collisions.

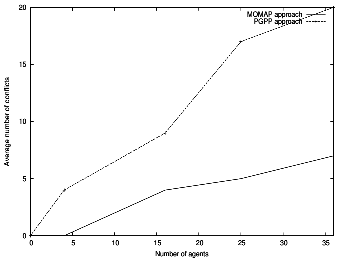

Figure 5: A comparison of conflict number

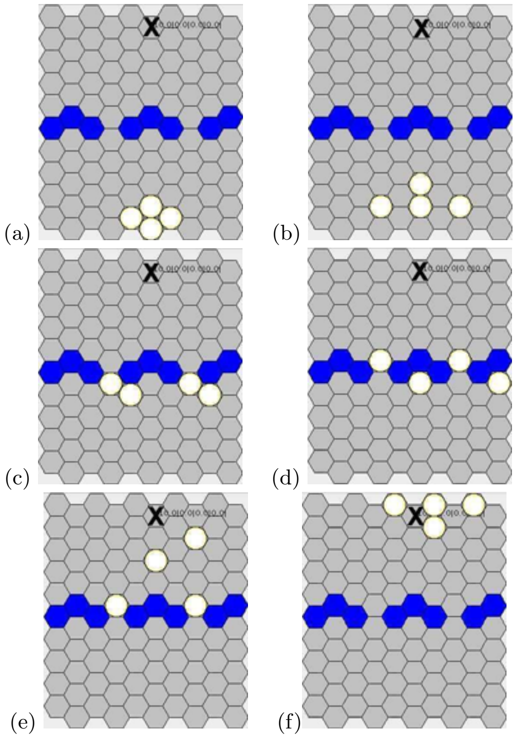

Figure 6: Example of Local Spatial Coordination

6.3 Example 3 : Emergent coalition formation

In this experiment, we consider a fleet of robots (fire fighters) at the middle of the grid (station of fire fighters) as depicted in Figure 7, a set of tasks (fires) defined at the four corners of the grid and we present how agents with local decisions create four coalitions to achieve the four defined tasks. As explained in section 2, agents consider initially the four tasks and then by an incremental decision making with multiple criteria, individual reward and social reward, agents form coalitions to deal with the four tasks. This example is inspired from the "robocup rescue" where a fleet of robots could be the fire fighters and tasks could be fires to extinguish. With the MOMAP framework we can see in Figure 7 that agents behave in flocks. Initially, agents are grouped together in the middle of the grid (station of fire fighters) and there are four destinations (fires). Agents at the beginning make distributed decisions that organise themselves into groups (Figure 7(b,c)), then the agents, in a distributed and coordinated way, they move towards their destinations (Figure 7(d,e,f)). Steps (d,e,f) show how agents coordinate their locations and they surround the target fire.

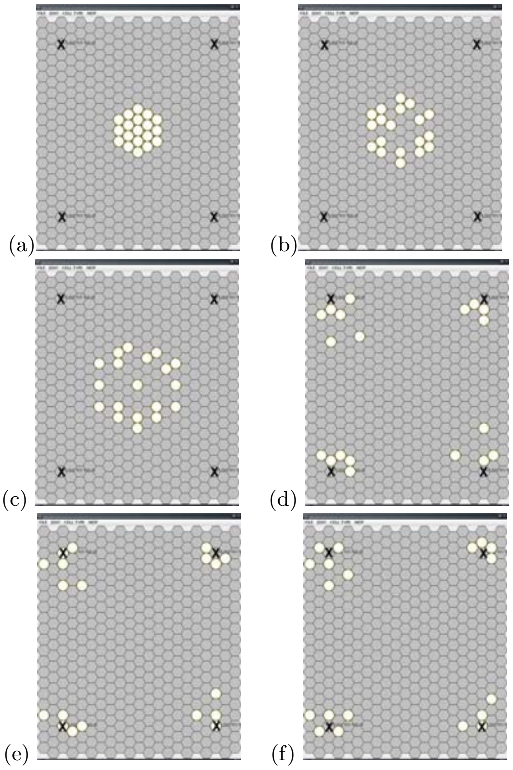

Figure 7: Example of Emergent Coalitions

7. RELATED WORK

This work could pave the way to new approaches to deal with Collective Intelligence which is about inducing a collection of agents with no exhibition of desired global behavior. Wolpert et al. define a *Wonderful life utility* (WLU) functions as a sum of R_i where local rewards are assumed cumulative while in our approach this assumption could be relaxed by the use of an augmented reward function AR_i. This augmented function can represented non-cumulative rewards as a vector considering individual and social rewards.

This approach is also a contribution in Multi-agent systems because it overcomes the main difficulty encountered in MAS design problem and mechanism designs which is the design of artificial cooperation structures to enforce coordination and to exhibit a desired behavior. Such cooperation structures make scaling up difficult and often non-robust because of a costly communication. Most of these approaches need a centralized common communication mechanisms while in MOMAP the communication is very limited or prohibited.

This approach is in the spirit of many existing models of MDPs with vector value functions [10, 6] and appropriate algorithms to solve them where most of them use backward induction, policy iteration and value iteration by substituting operations $(+, \times)$ by $(\max, \min)$ in computations. Other approaches have been interested in the use of a qualitative version of MDPs and algebraic MDPs [6, 7]. Besides these positive results, we propose an alternative to standard MDPs combining regret measure similar to Tchebychev norm with an appropriate lexicographic order and a backward induction algorithm to derive a satisfying policy. Further comparisons with these non-classical MDPs model will be developed in the future work. Another contribution of our model is the use of these non-classical models of MDP for multi-agent planning coordination problem.

8. CONCLUSION

We have presented a multi-criteria decision making technique for multi-objective multi-agent planning. We presented three contributions: (1) We have introduced a framework to represent objective relationships, (2) a related decision model using vector-valued decentralized Markov decision process and (3) a regret-based algorithm to solve the obtained DEC-MDP. We have shown that solving this DEC-MDP can lead to a more satisfying social behavior in certain settings. Furthermore, this approach reduces the number of conflicts and can converge quickly enough. Further experiments and analysis are needed to characterize more specifically the emerging global behavior and its convergence. Future work will also concern the use of multi-criteria Reinforcement Learning [12] of R_i, JR_i and $JPenalty_i$ and its effect on the emerging behavior and the coordination (discoordination) of local policies.

9. REFERENCES

[1] A. Beynier and A. Mouaddib. A polynomial algorithm to solve decentralized mdp with temporal constraints. In *AAMAS*, 2005.

[2] C. Boutilier, F. Bacchus, and R. Brafman. Ucp-networks: A directed graphical representation of conditional utilities. In *International Conference on Uncertainty in Artificial Intelligence (UAI)*, 2001.

[3] S. Jiaying, Z. Xiaoqin, and V. Lesser. Degree of local coordiantion and its implication on global utility. In *Proceedings of AAMAS*, pages 546–553, 2004.

[4] R. Keeney and H. Raiffa. *Decision with Multiple Objectives: Preferences and Value Tradeoffs.* John Wiley and Sons, Inc., 1976.

[5] A. Mouaddib. Anytime coordination for progressive planning agents. In *AAAI-99*, pages 564–569, 1999.

[6] P. Perny, O. Spanjaard, and P. Weng. Algebraic markov decision processes. In *IJCAI05*, 2005.

[7] R. Sabbadin. Possibilistic markov decision process. In *ECAI*, pages 586–590, 2000.

[8] W. Stirling, M. Goodrich, and P. Packard. Satisficing equilibria: A non-classical approach to games and decisions. *Journal of Autonomous Agents and Multi-Agent Systems*, 5:305–328, 2002.

[9] J. Von-Neumann and O. Morgenstern. The theory of games and economic behavior. In *Princeton Univ. Press*, 1947.

[10] K. Wakuta and K. Togawa. Solution procedures for multi-objective markov decison processes. *Optimization*, 43:29–46, 1998.

[11] D. Wolpert and K. Tumer. Introduction to collective intelligence. *Handbook of Agent Technology, AAAI Press/MIT Press*, 2000.

[12] G. Zoltan, Z. Kalmar, and C. Szepesvari. Multi-criteria reinforcement learning. In *International Conference on machine Learning (ICML)*, pages 197–205, 1998.