

# Extending Classical Planning to the Multi-agent Case: A Game-Theoretic Approach

Ramzi Ben Larbi, Sébastien Konieczny, and Pierre Marquis

CRIL - CNRS, Université d'Artois - Lens - France

**Abstract.** When several agents operate in a common environment, their plans may interfere so that the predicted outcome of each plan may be altered, even if it is composed of deterministic actions, only. Most of the multi-agent planning frameworks either view the actions of the other agents as exogeneous events or consider goal sharing cooperative agents. In this paper, we depart from such frameworks and extend the well-known single agent framework for classical planning to a multi-agent one. Focusing on the two agents case, we show how valuable plans can be characterized using game-theoretic notions, especially Nash equilibrium.

## 1 Introduction

In classical planning, one is interested in computing plans enabling an agent to reach her goals when performed. Among the standard assumptions in classical planning are that the initial state of the world is known by the agent, that each possible action is deterministic and its outcome can be perfectly predicted, that the goals are binary ones (i.e., each state of the world is either a fully satisfactory one or is fully unsatisfactory), and that the world is static in the sense that the only way to change it is to perform one of the agent's actions (thus, not only exogeneous events cannot take place but the world has no intrinsic dynamics).

More sophisticated planning frameworks are obtained by relaxing some of the assumptions above. In particular, in conformant planning, goals are still binary ones and the considered plans are unconditional ones, but it is not assumed that the initial state of the world is fully known or that the available actions are deterministic ones.

In this paper, we extend the classical planning setting to a multi-agent planning one. We consider a group of agents where each agent has its own actions and goals. Agents operate in a common environment. In this new setting, the standard assumptions of classical planning are made. Nevertheless, such assumptions (especially, the static world one and the deterministic actions one) are not enough to allow an agent to predict how the world will evolve after her plan is executed. Indeed, agents' plans interaction introduces some uncertainty. Each agent generally ignores which plans the other agents will point out and how her plan will be interleaved with theirs. We suggest to handle this issue thanks to concepts from game theory; in the new setting we put forward, we show how any agent can achieve a strategic diagnosis of the scenario under consideration, from its game representation.

Our approach to classical, yet multi-agent planning is meaningful in scenarios where it is not possible to sense/observe during plan execution (a usual assumption in classical planning), or when constraints (on time or resources) prevent from online re-planning. As a matter of example, consider autonomous high-speed flying robots or infobots working on highly volatile markets. For the sake of simplicity, we focus on the case where each agent knows the goals of each agent from the group, as well as the set of all plans each agent can point out. Those are standard assumptions in game theory. We also assume that the agents have the possibility to coordinate, which means that they can decide to build a common plan. In this case the uncertainty of the execution is removed. Let us consider a (toy) example as an illustration:

*Example 1.* Two agents, a robot-painter and an robot-electrician, operate in a single room. The bulb has to be changed (which is the goal of the electrician) and the ceiling has to be painted (which is the goal of the painter). The electrician has a new bulb and the painter the materials needed to paint the ceiling. Now, there is a single ladder in the room (the ladder is thus a critical resource). Furthermore, the painter needs some light in the room in order to make her job. The electrician can achieve three actions: TLe (“take the ladder”), CB (“change the bulb”), RLe (“release the ladder”); and the painter three actions: TLp (“take the ladder”), P (“paint”), RLp (“release the ladder”) ; P succeeds only if CB has been performed before. TLe and TLp succeed only if the ladder is available (i.e., it has been released before).

The following interactions can be easily envisioned:

- If the painter takes the ladder first, she will not be able to achieve her goal (since the bulb has to be changed first); if she does not release the ladder, then the electrician will not be able to achieve her goal.
- If the electrician takes the ladder first, she will be able to achieve her goal; then, the painter will be able to achieve her goal if and only if the electrician releases the ladder. Accordingly, if both agents coordinate so as to execute the joint plan TLe.CB.RLe.TLp.P, then both agents will be satisfied.

The key questions we address in this paper are the two following ones: for each agent of the group, what are her “best” plans? And does a given plan require coordination to be achieved in a satisfying way? Focusing mainly on the two agents case, we show how a game can be associated to any multi-agent planning problem; accordingly, the “best” plans for a rational agent can be characterized using game-theoretic notions, especially Nash equilibrium. We also identify the scenarios for which a cooperation between agents is likely to occur and show how many strategic information can be derived from the corresponding game. We finally show that several important settings where interacting agents are considered can be cast in our framework, including conformant planning [13,6,7,12] and Boolean games [10,9,8,3].

## 2 A Framework for Multi-agent Classical Planning

We consider a group of agents  $N = \{1, 2, \dots, k\}$ , where each agent is identified by an integer. Let  $S$  be a finite (and non-empty) set of (abstract) states. Let us denote by  $s_0$  the initial state, assumed to be the actual state of the world.  $s_0$  is known by each agent from  $N$ .

Each agent of  $N$  is associated to a finite set of available actions:

**Definition 1. (action)** An action  $\alpha$  is a mapping from  $S$  to  $S$ . The set of actions of agent  $i$  is denoted by  $A^i$ .

In the following, an action will be denoted by a small Greek letter. Note that the previous definition means that actions are deterministic and fully executable. This last assumption is not very demanding, since if one wants to model that an action is not executable in a state  $s$ , then this can be typically represented by an action that does not change the world when performed in  $s$ , i.e.  $\alpha(s) = s$ , or that leads to a “sink” state, i.e.,  $\alpha(s) = s_\perp$ , with  $s_\perp$  a non-goal state such that  $\beta(s_\perp) = s_\perp$  for every action  $\beta$ .

From her set of available actions, each agent can build some plans:

**Definition 2. (plan)** Let  $A$  be a set of actions. A plan  $p$  on  $A$  is denoted by a (possibly empty) sequence of actions of  $A$ , i.e.,  $p = \alpha_1.\alpha_2.\dots.\alpha_n$ , where each  $\alpha_i \in A$ . Semantically, it is a mapping from  $S$  to  $S$ , defined from sequential composition of its actions, i.e., for any  $s \in S$ ,  $p(s) = s$  if  $p = \epsilon$  (the empty sequence), and  $p(s) = \alpha_n(\dots(\alpha_1(s))\dots)$  otherwise. The set of all plans on  $A$  is denoted by  $A^*$ .

Let  $p = \alpha_1.\dots.\alpha_n$  be a plan. A subplan of  $p$  is a subsequence of it, i.e.,  $p' = \alpha'_1.\dots.\alpha'_m$  is a subplan of  $p$  if and only if there exists a strictly increasing mapping  $t$  from  $\{1, \dots, m\}$  to  $\{1, \dots, n\}$  s.t.  $\forall q \in \{1, \dots, m\}, \alpha'_q = \alpha_{t(q)}$ .

Let  $p' = \beta_1.\dots.\beta_r$  be another plan.  $p.p'$  denotes the concatenation of  $p$  and  $p'$ , i.e.,  $p.p' = \alpha_1.\dots.\alpha_n.\beta_1.\dots.\beta_r$ .

**Definition 3. (solution plan)** Let  $G^i \subseteq S$  be the set of goal states for agent  $i^1$ . Let  $s_0 \in S$  be the initial state. A plan  $p$  is a solution plan for  $i$  iff  $p(s_0) \in G^i$ .

In many cases, it is reasonable to assume that only a non-empty subset  $\Pi^i$  of  $A^{i*}$  is envisioned by agent  $i$ ; in particular, due to computational limitations, plans whose length exceeds a given preset bound can be discarded. Nevertheless, it makes sense to assume that  $\Pi^i$  is closed under subplan, i.e., when a plan  $p$  belongs to  $\Pi^i$ , then every subplan of it belongs to  $\Pi^i$  as well; in particular, the empty plan  $\epsilon$  always belongs to  $\Pi^i$ .

We are now ready to define the notions of agent representation and of multi-agent planning problem:

**Definition 4. (agent representation)** Each agent  $i \in N$  is characterized by a triple  $\mathcal{A}^i = \langle A^i, \Pi^i, G^i \rangle$  consisting of a set of actions  $A^i$ , a set of plans  $\Pi^i \subseteq A^{i*}$  and a set of goal states  $G^i$ .

<sup>1</sup> We also write  $G^i(s) = 1$  when  $s \in G^i$  and  $G^i(s) = 0$  otherwise.

**Definition 5. (*multi-agent planning problem*)** A multi-agent planning problem (MAPP) for a set  $N$  of agents is a triple  $\langle S, s_0, \{\mathcal{A}^i \mid i \in N\} \rangle$  consisting of a set of states  $S$ , an initial state  $s_0 \in S$  and a set of agent's representations  $\mathcal{A}^i$  (one per agent).

When several plans operating on a common environment are furnished (one plan per agent), the final course of events corresponding to their joint execution is one of their shuffles, unless a coordination is achieved. We denote by  $\oplus$  the mapping from  $A^* \times A^*$  to  $2^{A^*}$  that associates to any pair of plans  $p_i$  and  $p_j$ , the set containing all their shuffles:

**Definition 6. (*shuffle, shuffle set*)** Let  $p_i = \alpha_1^i \cdots \alpha_n^i \in A^{i*}$ ,  $p_j = \alpha_1^j \cdots \alpha_p^j \in A^{j*}$ . Then  $p_i \oplus p_j$  is the set of plans  $p$  which are permutations of  $p_i.p_j$  for which both  $p_i$  and  $p_j$  are subplans. Each  $p$  is said to be a shuffle of  $p_i$  and  $p_j$ , and  $p_i \oplus p_j$  is called the shuffle set of  $p_i$  and  $p_j$ .

Observe that  $\oplus$  is a permutative function (i.e., it is associative and commutative), so the previous definitions of shuffle and shuffle set can readily be extended to the case of more than 2 agents. Observe also that  $\epsilon$  (the empty sequence) is a neutral element for  $\oplus$ . Note that such an execution model based on plans shuffling is at work in concrete MAS, like Open Real-Time Strategy Games, see [5].

*Example 2.* Let us consider again the scenario given in Example 1. Let us call  $p_1$  the robot-electrician plan: TLe.CB and  $p_2$  the robot-painter plan: TLp.P. Then  $p_1 \oplus p_2 = \{\text{TLe.CB.TLp.P, TLe.TLp.CB.P, TLe.TLp.P.CB, TLp.TLe.P.CB, TLp.P.TLe.CB, TLp.TLe.CB.P}\}$ .

In the deterministic single agent case, evaluating a plan is quite easy. It is enough to look at the predicted state resulting from the (virtual) execution of the plan: what the agent foresees is what she gets. Characterizing the best plans is an easy task for the agent under consideration: the better the reached state, the better the plan. In the non-deterministic single agent case, the agent has to consider all possible reached states, and to aggregate their scores in order to evaluate a plan (many aggregation functions can be used, e.g. *min* (Wald criterion) for reflecting the behaviour of a pessimistic agent, or using expected utility when the scores are quantitative ones and non-deterministic actions are given by sets of probability distributions).

In the multi-agent (deterministic) case, which is the case we consider in this paper, the situation is similar to the non-deterministic single agent case in the sense that each agent has to consider all possible reached states in order to evaluate her plans. The main difference comes from the nature of uncertainty: in our setting, the uncertainty results from the interaction with the plans furnished by the other agents. Accordingly, each agent has to exploit the fact that she knows the other agents' goals and feasible plans in order to figure out what are her "best" plans. Contrastingly, in the non-deterministic single agent case, the need for handling non-determinism mainly comes from the impossibility to predict in a precise way the result of some actions, like "tossing a coin".

*Example 3.* If the robot-painter from Example 1 puts forward the plan  $p = \text{TLp.P.Rlp}$ , she is only ensured that the actions of  $p$  will be executed in the desired order. While she knows the electrician representation, she does not know which plan the electrician will choose (indeed, the set of feasible plans is not a singleton in general). Even if this set is a singleton, the painter still ignores the execution ordering, i.e., how her plan will interact with the electrician's one. Suppose that the electrician puts forward the plan  $p' = \text{TLe.CB.RLe}$ . The joint plan that will be finally executed can be any plan from  $p \oplus p'$ . The resulting uncertainty disappears whenever the two agents coordinate to put forward a common plan  $p'' = \text{TLp.P.Rlp.TLe.CB.RLe}$ .

In our setting, a key issue for each agent is to evaluate the interaction of her plans with the plans of the other agents. Formally, this calls for an evaluation of each shuffle set. To this purpose, we define the notion of satisfaction profile (SP), which is an abstract, summarized, view of shuffle sets evaluation for all the agents of the group. Let us explain how we construct a SP in the two agents situation. Given a pair of plans  $p_i \in \Pi^i$  and  $p_j \in \Pi^j$ , each shuffle from the shuffle set  $p_i \oplus p_j$  is a plan built from the actions of both agents; the execution of such a plan leads to a specific final state which is more or less satisfactory for each agent. The evaluation of a plan depends on the state resulting from its execution. We can depict the evaluation of this shuffle set by agent  $i$  using a 2-axis representation associating a dot on coordinate  $(x,y)$  to a shuffle  $p$  if  $G^i(p(s_0)) = x$  and  $G^j(p(s_0)) = y$ . Note that such a representation can be easily generalised to a  $n$ -player situation.

**Definition 7. (satisfaction profile)** *Given a MAPP for a set  $N = \{1, \dots, m\}$  of agents, with an initial state  $s_0$ , a satisfaction profile (SP) for the shuffle set  $p_1 \oplus p_2 \oplus \dots \oplus p_m$  where each  $p_i \in \Pi^i$  (with  $i \in \{1, \dots, m\}$ ) is a set  $SP(p_1 \oplus p_2 \oplus \dots \oplus p_m)$  of vectors  $(x_1, \dots, x_m)$  such that  $(x_1, \dots, x_m) \in SP(p_1 \oplus p_2 \oplus \dots \oplus p_m)$  if and only if  $\exists p \in p_1 \oplus p_2 \oplus \dots \oplus p_m$  such that for all  $i \in \{1, \dots, m\}$ ,  $G^i(p(s_0)) = x_i$ .*

When we consider only two agents  $i$  and  $j$ , the set of all possible SPs is given on Figure 1.

Numerous conclusions can be drawn from such SPs. Thus, some SPs are clearly better for an agent than other ones. Clearly, SP 2, where all shuffles lead to states that agent  $i$  evaluates to 1, is more interesting for her than SP 10, where all shuffles lead to non-goal states (i.e., states that agent  $i$  evaluates to 0). Let us also consider SP 3: for each of the two agents, at least one shuffle leads to a bad state (i.e., a non-goal state), and at least one shuffle leads to a goal state. This SP also shows the existence of at least one win-win shuffle (leading to the  $(1, 1)$  vector). In such a case, if both agents are rational ones (i.e., they act so as to make the world change to a goal state), then they have to coordinate. Indeed, coordination is a way to get rid of uncertainty. If the two agents  $i$  and  $j$  put forward two plans  $p_i \in \Pi^i$  and  $p_j \in \Pi^j$  in an independent way, they risk that the joint execution from  $p_i \oplus p_j$  leads to a state evaluated as  $(1, 0)$  or as  $(0, 1)$ , in which case one of the two agents will be unsatisfied. Contrastingly, if they coordinate and jointly put forward a plan corresponding to a win-win shuffle,

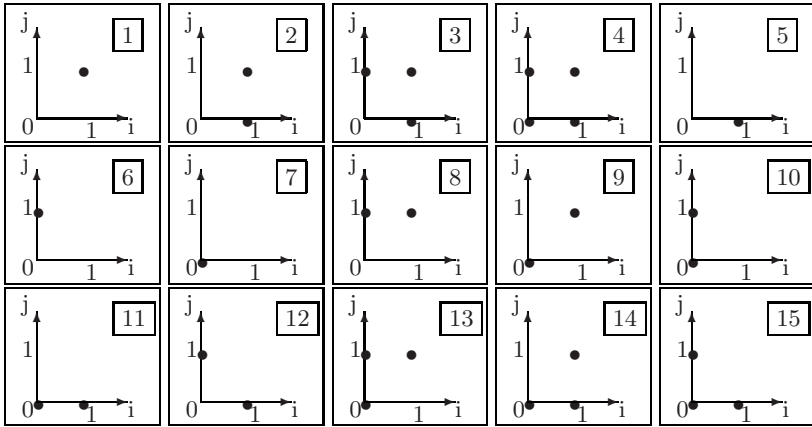


Fig. 1. All possible SPs

they are guaranteed to be both satisfied. So, when a shuffle leads to SP 3, both agents have interest in offering (and accepting) a coordination.

In absence of further information (especially, a probability distribution on the shuffle set), it makes sense to classify all the SPs w.r.t. an ordered scale reflecting the agent's opportunities. Let us take the point of view of agent  $i$  and show how the SPs can be gathered and ordered:

**Always Satisfied.** SPs 1, 2, 5. For all those SPs, agent  $i$  is ensured to reach her goals even if agent  $j$  does not accept any coordination. This is the most favourable case for agent  $i$ .

**Mutual Interest.** SPs 3, 4, 9, 13, 14. For any of those SPs, some joint executions do well and others do bad (for both agents), but they all share the  $(1, 1)$  vector, meaning that if the two agents coordinate, they can both reach their goals.

**Dependence.** SPs 8, 11. For those SPs, the evaluation of the shuffle set for the other agent does not depend on the joint execution. This means that, *a priori*, there is no "objective" reason for the other agent to accept/decline a coordination in order to help agent  $i$  to reach her goal.

**Antagonism.** SPs 12, 15. Those SPs reflect more problematic scenarios than the previous ones since the interests of the two agents are clearly distinct. This means that if one is satisfied, then the other one is not (in particular the coordination  $(1, 1)$  is never an option). In such cases, agent  $i$  can just hope that the joint execution will be good for her.

**Always Dissatisfied.** SPs 6, 7, 10. In every course of event, agent  $i$  will be dissatisfied (no joint execution allows the agent's goals to be reached). Such SPs are clearly the worst ones for agent  $i$ .

Our claim is that, in absence of further information, such a classification is the most rational one. Hence, we consider that each agent  $i$  has the following preferences on the evaluations of shuffle sets:

**Always Satisfied > Mutual Interest >**

**Dependence > Antagonism > Always Dissatisfied**

where  $X > Y$  meaning that SPs of class  $X$  are strictly preferred to SPs of class  $Y$ , and that all SPs in a given class are indifferent. We can easily encode such a total pre-order in a concise way, using numbers. Thus, we write  $e_i(p_i \oplus p_j) = 4$  if and only if  $SP(p_i \oplus p_j) \in \mathbf{Always\ Satisfied}(i)$ ,  $\dots$ ,  $e_i(p_i \oplus p_j) = 0$  if and only if  $SP(p_i \oplus p_j) \in \mathbf{Always\ Dissatisfied}(i)$  (see Table 1).

**Table 1.** SPs evaluation

Class	Evaluation
Always Satisfied	4
Mutual Interest	3
Dependence	2
Antagonism	1
Always Dissatisfied	0

Such evaluations  $e_i(p_i \oplus p_j)$  can be roughly seen as utilities, but they do not depend solely on the goals of agent  $i$ . Note also that the exact numbers that are used are not really important, just the order matters (our setting is not quantitative at all).

Note finally that, while the definitions to come will use those evaluations  $e_i(p_i \oplus p_j)$  and  $e_j(p_i \oplus p_j)$ , such definitions are still meaningful when other evaluations are used. Thus, if one disagrees with the proposed scale, the following definitions still apply (as soon as all the possible pairs of plans can be evaluated and totally ordered by the agents).

### 3 Solving the Game and Generating Strategic Diagnoses

From the previous construction we are now able to associate to each shuffle set an evaluation for each agent. This allows us to model the interaction between agents' plans as a game in strategic form. Note that extensive form game cannot work here since it cannot handle the shuffle situation (more exactly, it would lead to awful games in extensive form since there are too many possibilities).

Indeed, to each MAPP for a set of two agents  $N = \{1, 2\}$ , one can associate a game in strategic form, defined by the set  $N$  of players, the set of strategies for each player (the sets  $\Pi^1$  and  $\Pi^2$  of plans in our case), and by an evaluation function for each player that associates an evaluation to each profile of strategies (the evaluations  $e_1(p_1 \oplus p_2)$  and  $e_2(p_1 \oplus p_2)$  of each shuffle set  $p_1 \oplus p_2$  in our case).

*Example 4.* Let us consider the following MAPP:  $\langle S, s_0, \{\mathcal{A}^i \mid i \in \{1, 2\}\} \rangle$ .  $\mathcal{A}^1 = \langle A^1, \Pi^1 = \{p_1, p'_1\}, G^1 \rangle$ .  $\mathcal{A}^2 = \langle A^2, \Pi^2 = \{p_2, p'_2\}, G^2 \rangle$ . Suppose that the obtained SPs are the ones given in Figure 2.

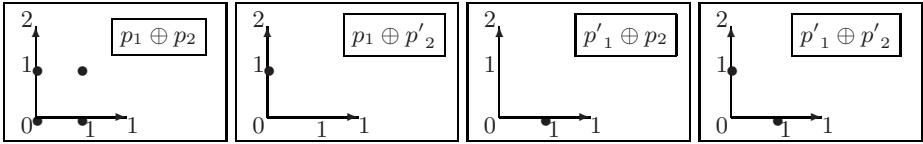


Fig. 2. Example of SPs

We can now associate to this MAPP the game in strategic form given in Table 2.

Table 2. Associated game

	$p_2$	$p'_2$
$p_1$	(3,3)	(0,4)
$p'_1$	(4,0)	(1,1)

In such a setting there are several candidates for the notion of “best plan”. One of them is based on the security level of the plans.<sup>2</sup>

**Definition 8. (security level of a plan)** Given a MAPP for  $N = \{1, 2\}$ , the security level of a plan  $p_i$  of an agent  $i$  ( $i \in N$ ) facing the set  $\Pi^j$  of plans of agent  $j$  ( $j \neq i$ ) is defined as the minimum evaluation of the shuffle set between plan  $p_i$  and a plan of player  $j$ , i.e.,

$$S_{\Pi^j}(p_i) = \min_{p_j \in \Pi^j} e_i(p_i \oplus p_j).$$

From the security levels of plans of an agent one can define the security level of the agent:

**Definition 9. (security level of an agent)** Given a MAPP for  $N = \{1, 2\}$ , the security level of agent  $i$  facing the set  $\Pi^j$  of plans of agent  $j$ , is the greatest security level of agent  $i$ ’s plans, i.e.,

$$S_{\Pi^j}(i) = \max_{p_i \in \Pi^i} S_{\Pi^j}(p_i).$$

A solution of the game associated to a given MAPP can be defined as a pair of plans  $\langle p_1 \in \Pi^1, p_2 \in \Pi^2 \rangle$  such that  $p_1$  (resp.  $p_2$ ) maximizes the security level of agent 1 (resp. 2) facing  $\Pi^2$  (resp.  $\Pi^1$ ).

Such a notion of solution makes sense in our framework since it can be roughly seen as a worst case analysis of the strategic interaction. Indeed, SPs are a (summarized view of the) set of possible results, and as the SP classification we

<sup>2</sup> While we focus on the 2-agent case, the following notions can be straightforwardly extended to the  $n$ -agent case.



pointed out mainly relies on worst case analysis, it is sensible to use security levels as well to compare shuffles.

Nevertheless, security levels do not take into account all agents' opportunities. A much more widely accepted concept of solution is Nash equilibrium [11].

**Definition 10. (Nash equilibrium)** *Given a MAPP for  $N = \{1, 2\}$ , a pair of plans  $\langle p_1 \in \Pi^1, p_2 \in \Pi^2 \rangle$  is a Nash equilibrium if none of the agents can get a better evaluation by choosing another plan, i.e.,  $\langle p_1, p_2 \rangle$  is a Nash equilibrium if and only if  $\nexists p \in \Pi^1$  s.t.  $e_1(p \oplus p_2) > e_1(p_1 \oplus p_2)$  and  $\nexists p \in \Pi^2$  s.t.  $e_2(p_1 \oplus p) > e_2(p_1 \oplus p_2)$ .*

*Example 5.* Let us step back to the game given in Table 2. And let us consider the pair  $\langle p'_1, p'_2 \rangle$ . Agent 1 has no incentive to deviate alone from this pair. Indeed,  $\langle p_1, p'_2 \rangle$  leads her to a less favorable situation ( $e_1(p_1 \oplus p'_2) < e_1(p'_1 \oplus p'_2)$ ). Similarly,  $\langle p'_1, p_2 \rangle$  is clearly less profitable to agent 2 than  $\langle p'_1, p'_2 \rangle$ . Thus, we can conclude that  $\langle p'_1, p'_2 \rangle$  is a Nash equilibrium. It is easy to check that it is the only Nash equilibrium of this game.

In our setting, as in the general case in game theory, it may happen that no Nash equilibrium (in pure strategies) exists, or that several Nash equilibria exist. When there are several Nash equilibria, other criteria, such as Pareto optimality,<sup>3</sup> can be used so as to discriminate them further. The following propositions give two sufficient conditions for the existence of such equilibria.

**Proposition 1.** *Let us consider a MAPP for two agents 1 and 2 such that  $G^1 = G^2$ . Then the associated game exhibits a Nash equilibrium.*

In particular, if the agents share the same goals and if there exists a joint plan that can achieve one of these goals, then our model will point it out as a solution.

**Proposition 2.** *Let us consider a MAPP for two agents 1 and 2. Let us denote by  $G^{1,+}$  (resp.  $G^{2,+}$ ) the subset of  $G^1$  (resp.  $G^2$ ) of states reachable using plans on  $A^1$  (resp.  $A^2$ ) and by  $G^{1,2,+}$  (resp.  $G^{2,1,+}$ ) the subset of  $G^1$  (resp.  $G^2$ ) of states reachable using plans on  $A^1 \cup A^2$ . If  $G^{1,+} = G^{2,+} = \emptyset$  and  $G^{1,2,+} = G^{2,1,+} \neq \emptyset$ , then the game associated to MAPP exhibits a Nash equilibrium.*

Note that, in our setting, the “prisoner’s dilemma” situation, a particular game situation widely studied (see e.g. [1,2]), can also be reached. Like in Example 4 (see Table 2):  $\langle p'_1, p'_2 \rangle$  is a Nash equilibrium, but the pair  $\langle p_1, p_2 \rangle$  which Pareto-dominates (i.e. is more profitable for both agents than)  $\langle p'_1, p'_2 \rangle$  is not a Nash equilibrium (so each agent is tempted to use the other plan).

Interestingly, each of the two agents  $i$  and  $j$  involved in the MAPP under consideration can derive a number of strategic information from the corresponding game. Due to space limitations, let us only focus on the notions of robust plan, synergetic effect and independence, successively

---

<sup>3</sup> A vector Pareto-dominates another one if each of the components of the first one is greater or equal to the corresponding component in the second one.

- A plan  $p_i$  for agent  $i$  is *robust* with respect to agent  $j$  if and only if its joint execution with any plan from agent  $j$  is ensured to reach the goals of  $i$ . In the game in strategic form, such a plan corresponds to a row (or a column) for which all the evaluations for this agent are 4:  $\forall p_j \in \Pi^j, e_i(p_i \oplus p_j) = 4$ . Clearly enough, such a plan maximizes the security level of agent  $i$ . If a robust plan exists for agent  $i$ , then no coordination is needed with agent  $j$ .
- The existence of a synergy between the two agents can also be easily derived from the game in strategic form. Indeed, a *synergetic effect* for agents  $i$  and  $j$  is possible if and only if there exist  $p_i \in \Pi^i$  and  $p_j \in \Pi^j$  such that  $e_i(p_i \oplus p_j) > \max_{p \in \Pi^i} e_i(\{p\})$  and  $e_j(p_i \oplus p_j) > \max_{p \in \Pi^j} e_j(\{p\})$ . Clearly enough, no synergetic effect is possible when at least one of the two agents has a robust plan.
- A notion of independence between agents, reflecting the fact that no interaction occurs, can also be easily derived from the game in strategic form. Indeed, the two agents are *independent* if and only if  $\forall p_i \in \Pi^i, \forall p_j \in \Pi^j, e_i(p_i \oplus p_j) = e_i(\{p_i\})$  and  $e_j(p_i \oplus p_j) = e_j(\{p_j\})$ .

## 4 Generality of the Framework

### 4.1 Conformant Planning

In conformant planning (see e.g. [13,6,7,12]), one is interested in determining whether a sequence of actions (i.e., a plan) is robust (or conformant), i.e., whether it will achieve the goal for all possible contingencies.

**Definition 11. (*conformant planning*)**

- A non-deterministic action  $\alpha$  over a finite and non-empty set  $S$  of states is a mapping from  $S$  to  $2^S \setminus \{\emptyset\}$ .
- A non-deterministic plan  $\pi$  on a set  $A$  of non-deterministic actions (over  $S$ ) is a finite sequence of elements of  $A$ .
- A trajectory for a non-deterministic plan  $\pi = \alpha_1 \dots \alpha_n$  given an initial state  $s_0 \in S$  is a sequence of states  $s_0, \dots, s_{n+1}$  s.t. for every  $i \in 0 \dots n$ ,  $s_{i+1} \in \alpha_i(s_i)$ .
- A non-deterministic plan  $\pi = \alpha_1 \dots \alpha_n$  on  $A$  is conformant for a goal  $G \subseteq S$  given an initial state  $s_0 \in S$  if and only if for every trajectory  $s_0, \dots, s_{n+1}$  for  $\pi$ ,  $s_{n+1} \in G$ .

This problem can be easily cast into our framework. The key idea is to consider every possible trajectory attached to a non-deterministic plan as the result of a possible shuffle with a plan supplied by a second agent who plays the role of Mother Nature; consider the first action  $\alpha$  of the plan and assume it has at most  $k$  possible outcomes. In this case, the second agent's plan will start with actions  $\alpha'_1, \dots, \alpha'_k$  where each  $\alpha'_j$  is the void action if  $\alpha$  has not been executed (which is encoded using a specific fluent) and achieves the  $j^{\text{th}}$  outcome of  $\alpha$  otherwise. It mainly remains to repeat it for every action of the first plan and to update the second agent's plan by concatenating it with the subplan obtained at each step.

## 4.2 Boolean Games

**Definition 12. (Boolean game)**[10,9,8,3] A Boolean game is a 4-tuple  $G = \langle N, V, \Pi, \Phi \rangle$  where  $N = \{1, \dots, n\}$  is a set of agents,  $V$  is a set of propositional variables (decision variables),  $\Pi : N \rightarrow 2^V$  a control assignment function that induces a partition  $\{\pi_1, \dots, \pi_n\}$  of  $V$  with  $\pi_i$  the set of variables controlled by agent  $i$ ,  $\Phi = \{\phi_1, \dots, \phi_n\}$  a set of formulas.

For a player  $i \in N$ , a strategy is a truth assignment of her controlled variables (i.e., a mapping from  $\Pi(i)$  to  $\{0, 1\}$ ). A strategy profile consists of the assignments of all the considered agents, and can be viewed as a truth assignment on  $V$  (i.e., a mapping from  $V$  to  $\{0, 1\}$ ). Agent  $i$  is satisfied by a strategy profile  $P$  if and only if  $P$  is a model of  $\phi_i$ .

We can cast this framework into our one by associating to each variable  $v \in V$  a deterministic action  $v^+$  which sets variable  $v$  to 1. To each boolean game  $G = \langle N, V, \Pi, \Phi \rangle$  we can associate a MAPP  $\langle S, s_0, \{\mathcal{A}^i \mid i \in N\} \rangle$  where  $S$  is the set of all truth assignments on  $V$ ,  $s_0$  is the truth assignment s.t.  $s_0(v) = 0$  for all  $v \in V$ . For every agent  $i$ ,  $A^i = \{v^+ \mid v \in \pi_i\}$ ,  $\Pi^i$  is the subset of plans from  $A^{i*}$  such that every action has at most one occurrence in each plan and  $G^i$  is the set of models of  $\phi_i$ .

## 5 Related Work and Conclusion

While much work has been devoted for the past few years to multi-agent planning, they typically assume that agents share some common goals. Relaxing this assumption has a major impact on the possible approaches to tackle the problem and calls for game-theoretic notions.

A closely related approach to our own one is described in [4]. In this paper, policies at the group level are evaluated w.r.t. each agent and the “best ones” are characterized as Nash equilibria, as it is the case in our work. The approach nevertheless departs from our own one by a number of aspects:

- The framework under consideration is planning under uncertainty with full observability and not classical planning. Non-deterministic actions are considered and a set of possible initial states (and not a single state) is known by each agent. Policies are mappings associating actions to states and not linear plans (sequences of actions), and the quality of a plan is not binary in essence (contrariwise to what happens in the classical framework).
- Policies at the group level are part of the input and policies at the agent level are not (while possible plans at the group level are characterized as shuffles from plans at the agent level in our setting).
- Finally, no notion of strategical diagnosis is considered (especially, the need for coordination cannot be derived from the input since policies at the agent level are not relevant).

In this work we have proposed a framework to model multi-agent planning problems. This framework allows to draw strategic conclusions about specific

interactions, and also allows to *solve* many situations. A main point is to show how each MAPP into consideration can be associated to a suitable representation (SP) which can be evaluated (as a number), and this allows for exploiting easily notions and results from game theory. As far as we know, there is no similar notions of SP and evaluations in the literature. Representation and algorithmic aspects are issues for further research.

## References

1. Axelrod, R.: The Evolution of Cooperation. Basic Books, New York (1984)
2. Beauflis, B., Delahaye, J.P., Mathieu, P.: Complete classes of strategies for the classical iterated prisoner's dilemma. In: Proc. of EP'98, pp. 33–41 (1998)
3. Bonzon, E., Lagasquie-Schiex, M., Lang, J., Zanuttini, B.: Boolean games revisited. In: Proc. of ECAI'06, pp. 265–269 (2006)
4. Bowling, M., Jensen, R., Veloso, M.: A formalization of equilibria for multiagent planning. In: Proc. of IJCAI'03, pp. 1460–1462 (2003)
5. Buro, M.: Real-time strategy games: A new AI research challenge. In: Proc. of IJCAI'03, pp. 1534–1535 (2003)
6. Cimatti, A., Roveri, M.: Conformant planning via model checking. In: Proc. of ECP'99, pp. 21–34 (1999)
7. Cimatti, A., Roveri, M., Bertoli, P.: Conformant planning via symbolic model checking and heuristic search. Artificial Intelligence 159(1-2), 127–206 (2004)
8. Dunne, P.E., van der Hoek, W.: Representation and complexity in boolean games. In: Alferes, J.J., Leite, J.A. (eds.) JELIA 2004. LNCS (LNAI), vol. 3229, pp. 347–359. Springer, Heidelberg (2004)
9. Harrenstein, P.: Logic in Conflict. PhD thesis, Utrecht University (2004)
10. Harrenstein, P., van der Hoek, W., Meyer, J., Witteveen, C.: Boolean games. In: Proc. of TARK'01, pp. 287–298 (2001)
11. Nash, J.: Equilibrium points in N-person games. Proceedings of the National Academy of Sciences of the United States of America 36(1), 48–49 (1950)
12. Palacios, H., Geffner, H.: Compiling uncertainty away: Solving conformant planning problems using a classical planner (sometimes). In: Proc. of AAAI'06 (2006)
13. Son, T.C., Tu, P.H., Gelfond, M., Morales, A.R.: Conformant planning for domains with constraints—a new approach. In: Proc. of AAAI'05, pp. 1211–1216 (2005)