Integrating individual preferences in multi-agent planning

Alejandro Torreño, Eva Onaindia, Óscar Sapena
Universitat Politècnica de València
Camino de Vera s/n, 46022 Valencia, SPAIN
{atorreno,onaindia,osapena}@dsic.upv.es

Abstract

In this paper we address the problem of incorporating self-interest agents which have individual preferences in a cooperative multi-agent planning (MAP) framework. Thus, agents are aimed at solving together the hard problem goals and, in addition, to satisfy as many as possible of their soft preferences.

We propose an extension of FMAP, an efficient general-purpose forward MAP algorithm that solves cooperative tasks over a diverse set of planning problems, to accommodate the individual preferences of agents. The new heuristic function of FMAP estimates now the cost to reach the common problem goals as well as the utility of a node regarding the individual preferences of the agents. We show some preliminary experimental results of the preference-based FMAP when agents use a Borda voting mechanism to select the best node according to their preference profiles.

Introduction

Multi-Agent Planning (MAP) extends the classical planning paradigm by introducing several independent entities that plan and act together. Over the last years, MAP has primarily focused on studying different types of architectures (distributed or centralized), coordination mechanisms of agents' solutions (coordination before planning, coordination after planning or interleaved planning and coordination) or issues like privay preserving.

Most MAP approaches stem from extensions of the classical single-agent planning paradigm, assuming agents to be fully cooperative. Agents are typically endowed with a set of global goals that must be collectively attained by the group in order to solve the task at hand (Nissim, Brafman, and Domshlak 2010; Torreño, Onaindia, and Sapena 2012; Borrajo 2013).

The inclusion of self-interested agents with individual preferences in MAP is a matter of study rarely addressed by the planning community. Preference-based planning (PBP) is a branch of classical single-agent planning that addresses the problem of determining when a plan is preferred over another (Baier and McIlraith 2009). Users specify the desirable properties of a solution plan in the form of preferences and PBP algorithms search for a solution that satisfies the largest number of preferences.

The combination of PBP and MAP emerges as an interesting and novel field of research. Most of the existing MAP approaches focus only on fully cooperative agents or apply game-theoretic concepts to combine local solutions into a multi-agent solution plan that ensures certain theoretical properties, such as Nash equilibrium (Brafman et al. 2009).

In (Torreño, Onaindia, and Sapena 2013; 2014b), we introduced FMAP (Forward Multi-Agent Planning), a general-purpose MAP framework that efficiently solves cooperative multi-agent planning tasks from different planning domains. FMAP shows to be very effective at solving hard problem instances where the level of interaction between subgoals is strong. FMAP features a refinement planning scheme (Kambhampati 1997), by which agents cooperatively explore a joint search tree. The nodes of the search tree are partial-order plans built through the contributions of one or more planning agents. At each iteration of the procedure, agents pose refinement plans by introducing actions over a base plan chosen among the leaf nodes of the tree through a novel MAP heuristic function. Each agent is provided with an embedded forward-chaining partial-order planner to build refinement plans.

The main limitation of FMAP is that it only attains cooperative tasks in which the goals are known to all the participating agents. This work generalizes the definition of MAP task introduced in (Torreño, Onaindia, and Sapena 2014b), explicitly allowing agents to have individual preferences over the goal state.

In its original form, FMAP agents apply a straightforward A* procedure, selecting the open node of the search tree that minimizes an evaluation function $f = g + h$ as the next base plan to refine. Since FMAP is based on a cooperative scheme, all the agents share the same f value for any given plan.

In this paper, we propose an extension of FMAP to accommodate individual preferences. Our aim is to allow agents to explore the search tree considering not only the global goals, but also their individual preferences. For this purpose, we defined a utility function that allows each agent to estimate the quality of the refinement plans with respect to the global goals as well as its individual preferences.

In single-agent PBP, it is necessary to establish an ordering relation that specifies which plans are preferred to others. A possible way to define this relation is to associate a numerical value to each preference, representing the penalty value for the plans that leave such preference unsatisfied. This approach is followed by PDDL3 (Gerevini and Long 2005), in which a preference is violated by a plan if it logically evaluates to false in such plan. As the aforementioned PBP approach, we associate a penalty to each of the preferences and evaluate the quality of the solution plans according to a metric function that takes into consideration the unfulfilled preferences of the agents.

We define a new plan selection scheme that aggregates the individual preferences of the agents when selecting the next base plan to explore. To do so, we rely on concepts from the social choice theory; particularly, we apply voting mechanisms to select the best base plan according to the preference profiles of the agents.

Most of the PBP algorithms follow an incremental approach in which the planner returns a sequence of plans with increasing quality (Baier and McIlraith 2009). For the experimental evaluation, we modified the stop criterion of FMAP, so that agents keep building solution plans of increasing quality after the first one is found.

This paper is organized as follows: next section formalizes a generalized MAP task and presents the main concepts upon which our approach is based, as well as the changes introduced into our specification language to support individual preferences; next, we introduce the modified FMAP algorithm and thoroughly analyze the main changes included; following, we provide some initial experimental results; and finally, we conclude and summarize our upcoming lines of work.

MAP task formalization

Agents in our MAP model work under a limited knowledge of the planning task by assuming that the information not represented in an agent's model is unknown to the agent. The states of the world are modeled through a finite set of *state variables*, $\mathcal{V}$, each of them associated to a finite domain, $\mathcal{D}_v$, of mutually exclusive values that refer to the objects in the world. Assigning a value d to a variable $v \in \mathcal{V}$ generates a *fluent*. A *positive fluent* is a tuple $\langle v, d\rangle$, which indicates that the variable v takes the value d. A *negative fluent* $\langle v, \neg d\rangle$ indicates that v does not take the value d. A *state* is a set of positive and negative fluents.

An *action* is a tuple $\alpha = \langle PRE(\alpha),\ EFF(\alpha)\rangle$, where $PRE(\alpha)$ is a finite set of fluents modeling the preconditions of α, and $EFF(\alpha)$ is a set of *variable assignments* that model the effects of α. Executing an action α in a world state S leads to a new state S' as a result of applying $EFF(\alpha)$ over S.

Definition 1 *A* ***MAP task*** *is tuple* $\mathcal{T}_{MAP} = \langle \mathcal{AG}, \mathcal{V}, \mathcal{I}, \mathcal{G}, \mathcal{A}, \mathcal{P}\rangle$. $\mathcal{AG} = \{1, \ldots, n\}$ *is a finite non-empty set of agents.* $\mathcal{V} = \bigcup_{i\in\mathcal{AG}} \mathcal{V}^i$, *where* $\mathcal{V}^i$ *is the set of state variables known to an agent* i. $\mathcal{I} = \bigcup_{i\in\mathcal{AG}} \mathcal{I}^i$ *is a set of fluents that defines the initial state of* $\mathcal{T}_{MAP}$. $\mathcal{G}$ *is the set of global goals of* $\mathcal{T}_{MAP}$ *that are common to all the participating agents.* $\mathcal{A} = \bigcup_{i\in\mathcal{AG}} \mathcal{A}^i$ *is the set of planning actions of the agents. Finally ,* $\mathcal{P} = \bigcup_{i\in\mathcal{AG}} \mathcal{P}^i$ *is the set of preferences of the agents in* $\mathcal{T}_{MAP}$.

Some characteristics of the elements of $\mathcal{T}_{MAP}$ are:

- Since specialized agents are allowed, they may only know a subset of the initial state $\mathcal{I}$. However, the initial states of the agents never contradict each other.
- Tyipically, the sets of actions of two specialized agents are disjoint but they may also contain some common actions.
- $\mathcal{A}$ includes two fictitious actions α_i and α_f: α_i represents the initial state of $\mathcal{T}_{MAP}$, i.e., $PRE(\alpha_i) = \emptyset$ and $EFF(\alpha_i) = \mathcal{I}$, while α_f models the global goals of $\mathcal{T}_{MAP}$, i.e., $PRE(\alpha_f) = \mathcal{G}$, and $EFF(\alpha_f) = \emptyset$.
- The sets of individual preferences of the agents, $\mathcal{P}^i$, are disjoint sets.

The previous definition includes a private set of *preferences* $\mathcal{P}^i$ for each agent i. Preferences in our model are defined as soft goals since they are not required to be accomplished in order to solve the MAP task. Formally, a *preference* of an agent i, $p \in \mathcal{P}^i$, is a tuple $p = \langle f, penalty\rangle$, where f is a fluent that the agent i wants to achieve in $\mathcal{G}$, and $penalty$ is a numerical penalty applied to the agent in case that the preference is not satisfied in a solution plan.

As indicated in Definition 1, our model considers specialized agents such that each agent has a local and limited *view* on the MAP task. The view of an agent includes both the information it knows and the preferences it has on the MAP task at hand.

Definition 2 *The* ***view*** *of an agent* i *on a MAP task is defined as* $\mathcal{T}^i_{MAP} = \langle \mathcal{V}^i, \mathcal{A}^i, \mathcal{I}^i, \mathcal{G}, \mathcal{P}^i, \mathcal{T}h^i\rangle$. $\mathcal{V}^i$ *is the set of state variables known to agent* i; $\mathcal{A}^i \subseteq \mathcal{A}$ *is the set of its planning actions;* $\mathcal{I}^i$ *is the subset of fluents of the initial state* $\mathcal{I}$ *that are visible to agent* i; *and* $\mathcal{G}$ *is the set of global goals of* $\mathcal{T}_{MAP}$. *All the agents in* $\mathcal{T}_{MAP}$ *are aware of the global goals of the task.* $\mathcal{P}^i$ *is the set of individual preferences of agent* i, *and* $\mathcal{T}h^i$ *is the acceptable threshold of penalty for the agent to validate a solution plan.*

The state variables of an agent i are determined by the view it has on the initial state, I^i, the planning actions it can perform, A^i, and set of goals of $\mathcal{T}_{MAP}$. This also affects the domain D_v of a variable v. We define $\mathcal{D}^i_v \subseteq \mathcal{D}_v$ as the set of values of the variable v that are known to agent i. Agents in our model interact with each other by sharing information on their state variables. Given a pair of agents i and j, the set of variables they share is defined as $\mathcal{V}^{ij} = \mathcal{V}^{ji} = \mathcal{V}^i \cap \mathcal{V}^j$. Moreover, some of the values in the domain of a variable can also be public to both agents. The set of values of a variable v that are public to a pair of agents i and j is defined as $\mathcal{D}^{ij}_v = \mathcal{D}^{ji}_v = \mathcal{D}^i_v \cap \mathcal{D}^j_v$.

As introduced in (Torreño, Onaindia, and Sapena 2014b), our MAP model is based on a multi-agent refinement planning framework, in which agents apply a Partial-Order Planning (POP) search procedure in order to generate refinement plans. The next definitions briefly introduce standard concepts from the POP paradigm (Ghallab, Nau, and Traverso 2004) adapted to a MAP context with state variables.

Definition 3 *A* ***partial-order plan*** *or partial plan is a tuple* $\Pi = \langle \Delta, \mathcal{OR}, \mathcal{CL} \rangle$. $\Delta = \{\alpha | \alpha \in \mathcal{A}\}$ *is the set of actions in* Π. $\mathcal{OR}$ *is a finite set of ordering constraints* ($\prec$) *on* Δ. $\mathcal{CL}$ *is a finite set of causal links of the form* $\alpha \overset{\langle v,d \rangle}{\rightarrow} \beta$ *or* $\alpha \overset{\langle v,\neg d \rangle}{\rightarrow} \beta$, *where* α *and* β *are actions in* Δ. *A causal link* $\alpha \overset{\langle v,d \rangle}{\rightarrow} \beta$ *enforces a precondition* $\langle v, d \rangle \in PRE(\beta)$ *through an effect* $(v = d) \in EFF(\alpha)$. *Similarly, a causal link* $\alpha \overset{\langle v,\neg d \rangle}{\rightarrow} \beta$ *enforces* $\langle v, \neg d \rangle \in PRE(\beta)$ *through an effect* $(v \neq d) \in EFF(\alpha)$ *or* $(v = d') \in EFF(\alpha)$, $d' \neq d$.

An *empty* partial plan is defined as $\Pi_0 = \langle \Delta_0, \mathcal{OR}_0, \mathcal{CL}_0 \rangle$, where $\mathcal{OR}_0$ and $\mathcal{CL}_0$ are empty sets, and Δ_0 contains only the fictitious initial action α_i. A partial plan Π for a task $\mathcal{T}_{MAP}$ will always contain α_i.

The introduction of new actions in a partial plan may trigger the appearance of *flaws*, that is, preconditions that are not yet solved through a causal link, and threats. A *threat* over a causal link $\alpha \overset{\langle v,d \rangle}{\rightarrow} \beta$ is caused by an action γ that is not ordered w.r.t. α or β and might potentially modify the value of v ($(v \neq d) \in EFF(\gamma)$ or $(v = d') \in EFF(\gamma)$, $d' \neq d$), making the causal link unsafe.

A *flaw-free* plan is a threat-free partial plan in which the preconditions of all the actions are supported through causal links.

Planning agents in our model cooperate to solve MAP tasks by progressively refining an initially empty plan Π until a solution is found. We define a refinement plan as follows:

Definition 4 *A* ***refinement plan*** $\Pi_r = \langle \Delta_r, \mathcal{OR}_r, \mathcal{CL}_r \rangle$ *over a partial plan* $\Pi = \langle \Delta, \mathcal{OR}, \mathcal{CL} \rangle$, *is a flaw-free partial plan which extends* Π, *i.e.,* $\Delta \subset \Delta_r$, $\mathcal{OR} \subset \mathcal{OR}_r$ *and* $\mathcal{CL} \subset \mathcal{CL}_r$. Π_r *introduces a new action* $\alpha \in \Delta_r$ *in* Π, *resulting in* $\Delta_r = \Delta \cup \alpha$. *All the preconditions in* $PRE(\alpha)$ *are linked to existing actions in* Π *through causal links; i.e., all preconditions are supported and so it holds* $\forall p \in PRE(\alpha)$, $\exists\, \beta \overset{p}{\rightarrow} \alpha \in \mathcal{CL}_r$, *where* $\beta \in \Delta$.

Refinement plans are individually evaluated by each agent to assess not only their quality, but also how they accomplish the agent's preferences. More precisely, an agent i evaluates a refinement plan Π through an evaluation function $f^i(\Pi) = g(\Pi) + selfInterest^i \cdot h_{pub}(\Pi) + (1 - selfInterest^i) \cdot \sum_{\forall p \in \mathcal{P}^i}(\beta_p \cdot h_{pri}(\Pi, p))$, where:

- $g(\Pi)$ is the cost of Π, measured as the number of actions of Π.
- $h_{pub}(\Pi)$ is an estimate of the number of actions required to reach the global goals of the task, $\mathcal{G}$.
- $h_{pri}(\Pi, p)$ estimates the number of actions to satisfy the agent's preference p in the refinement plan Π.
- $\beta_p \in [0, 1]$ is a parameter used to assess the relevance of achieving a particular individual preference p over the others. β_p is defined in terms of the penalty associated to each preference: $\beta_p = penalty(p) / \sum_{\forall p' \in \mathcal{P}^i} penalty(p')$.
- $selfInterest^i \in [0, 1]$ indicates the weight that the agent i gives to the accomplishment of the global goals and its private preferences. The higher the value, the more self-interested the agent, meaning that achieving the individual preferences is more relevant to the agent. In the case of a more cooperative agent that puts the emphasis on achieving the global goals of the task, the value of this parameter will be lower.

For every open node Π (refinement plan) of the search tree, an agent i applies $f^i(\Pi)$, thus creating an individual *preference profile* over the open nodes of the tree. As we will explain in the next sections, agents aggregate the individual preference profiles to select the next base plan to be refined.

Definition 5 *A* ***solution plan*** *for* $\mathcal{T}_{MAP}$ *is a refinement plan* $\Pi = \langle \Delta, \mathcal{OR}, \mathcal{CL} \rangle$ *that addresses all the global goals* $\mathcal{G}$ *of* $\mathcal{T}_{MAP}$ *and meets the penalty threshold for a majority of the agents. Hence, a refinement plan* Π *is a solution iff* $\alpha_f \in \Pi$ *(and thus, all the goals in* $\mathcal{G}$ *are satisfied; that is,* $\forall g \in PRE(\alpha_f)$, $\exists\, \beta \overset{g}{\rightarrow} \alpha_f \in \mathcal{CL}$, $\beta \in \Delta$*) and* $|PT| > |\mathcal{AG}|/2$, *where* $PT = \{i \in \mathcal{AG} | PlanPenalty^i(\Pi) \leq \mathcal{T}h^i\}$.

Refinement plans in our model include parallel actions introduced by different agents. As described in (Torreño, Onaindia, and Sapena 2014b), we ensure that the effects and preconditions of such actions are mutually consistent through the resolution of threats over the causal links of the plan. Consistency between any two non-sequential actions introduced by different agents is always guaranteed since refinement plans are flaw-free plans.

Every time an agent i refines a plan by introducing an action $\alpha \in \mathcal{A}^i$, it communicates the resulting refinement plan Π to the rest of the agents in $\mathcal{T}_{MAP}$. As in (Torreño, Onaindia, and Sapena 2014b), privacy is preserved by communicating only the fluents of the new action α that are relevant to the sender and receiver agents. The information of a refinement plan Π that an agent j receives from agent i configures its *view* of such plan, $view^j(\Pi)$.

Extensions to the MAP language

In (Torreño, Onaindia, and Sapena 2014a), we firstly introduced our MAP language based on *PDDL3.1* (Kovacs 2011). We have extended the language with some additional constructs in order to support the extensions introduced in this section. The domain files of the specialized agents do not suffer any modification but the problem file of each agent includes now new constructs to define the behaviour and individual preferences of the agent. Throughout this section, we will illustrate the changes using a simple example from the well-known `driverlog` domain.

Currently, we only allow for the definition of individual preferences regarding the goal state. Therefore, now the `:goal` construct includes the definition of the preferences:

```
(:goal (and
 (= (in package1) s1)
 (= (in package2) s2)
 (preference p0 (= (at driver1) s1))
 (preference p1 (= (pos truck1) s1))
))
```

As shown in the previous example, each preference is associated to an identifier that is then used in the specification of the problem metric and the penalties that are applied to the agent when its preferences are not accomplished. The `:metric` section is defined as in *PDDL2.1* (Fox and Long 2003).

```
(:metric minimize
  (+ (total-time)
     (is-violated p0)
     (* (is-violated p1) 2)
))
```

The above example shows that the metric minimizes the sum of the plan duration and the penalties associated to each preference. Since FMAP does not explicitly manage time, the `total-time` parameter is interpreted as the *makespan* or duration of the plan. In the example, $penalty(\texttt{p0}) = 1$ and $penalty(\texttt{p1}) = 2$.

Finally, we include the `:behaviour` construct to define both the agent's self-interest level and its metric threshold. This section is defined as in the following example:

```
(:behaviour
  (self-interest 0.5)
  (metric-threshold 0)
)
```

The agent in this example gives the same level of relevance to the global goals and its preferences when it evaluates refinement plans. The value of the metric threshold indicates that the agent will only accept a plan as a solution when all its individual preferences are satisfied.

Preference-based FMAP

This section describes the preference-based FMAP algorithm, which was originally designed for fully cooperative agents (Torreño, Onaindia, and Sapena 2014b) and has been revised to accomodate individual preferences. FMAP agents build a joint search tree in which nodes are refinement plans (partial-order plans) whose actions are contributed by one or more planning agents. Each agent independently devises refinement plans over a centralized base plan through an embedded forward-chaining POP (FPOP) procedure.

Algorithm 1 shows the preference-based FMAP algorithm as executed by an agent i. The main stages of the procedure can be summarized as follows:

- **Individual refinement plan generation**: each agent individually applies its embedded FPOP procedure to generate a set of refinement plans over the current base plan, Π_b. In Algorithm 1, the RP^i set stores the refinement plans devised by agent i. A refinement plan introduces a new fully-supported action in Π_b.
- **Communication of refinement plans**: agents communicate each other the refinement plans they generated. Agent i in Algorithm 1 sends each other agent j $view^j(\Pi_i)$, for each $\Pi_i \in RP^i$, thus occluding the information that is private to j. In turn, agent i receives, from each other agent j, $view^i(\Pi_j)$, for all $\Pi_j \in RP^j$. The $Refinements^i$ set stores the view an agent i has on all the refinement plans created by the participating agents in a particular iteration of FMAP.
- **Evaluation of the refinement plans**: each agent i individually applies the utility function f^i, described in the previous section, to evaluate the refinement plans, and it stores them into the $openNodes^i$ set. This set keeps the plans ordered according to the agent's utility function f^i. Agents make use of a DTG-based heuristic function, h_{DTG} (Torreño, Onaindia, and Sapena 2014b), to calculate the heuristic estimates of f^i.
- **Base plan selection**: agents select, among all the open nodes of the multi-agent plan-space search tree, the refinement plan that is preferred by the group of agents. To do so, agents aggregate their individual preferences on the open nodes by means of a social choice mechanism.

Algorithm 1: Preference-based FMAP algorithm as applied by an agent i

```
$SolutionPlans \leftarrow \emptyset$
$openNodes^i \leftarrow \emptyset$
$\Pi_b \leftarrow \Pi_0$
repeat
    $RP^i \leftarrow FPOP(\Pi_b)$
    $Refinements^i \leftarrow RP^i$
    for $j \in \mathcal{AG}$, $j \neq i$ do
        $\forall \Pi^i \in RP^i$, send $view^j(\Pi^i)$ to $j$
        $\forall \Pi^j \in RP^j$, receive $view^i(\Pi^j)$ from $j$
        $Refinements^i \leftarrow Refinements^i \cup RP^j$
    $\forall\ \Pi_r \in Refinements^i$, compute $f^i(\Pi_r)$
    $openNodes^i \leftarrow openNodes^i \cup Refinements^i$
    $\Pi_b \leftarrow SocialChoice(openNodes^i)$
    $openNodes^i \leftarrow openNodes^i \setminus \Pi_b$
    if $\alpha_f \in \Pi_b$ then
        if $MajorityApproval(\Pi_b)$ then
            $SolutionPlans \leftarrow \Pi_b$
until $Timeout \vee openNodes^i = \emptyset$
return $SolutionPlans$
```

If a base plan Π_b supports the task goals $\mathcal{G}$, i.e., $\alpha_f \in \Pi_b$, agents vote to decide if such a plan is accepted as a solution ($MajorityApproval(\Pi_b)$ function in Algorithm 1). In order for the plan Π_b to be accepted, the sum of the agent's penalties caused by Π_b, $PlanPenalty^i(\Pi_b)$, has to be equal or lower than the agent's threshold $\mathcal{T}h^i$ for a majority of agents. Otherwise, the agents keep searching for plans that are compliant with the threshold of more than half of the agents.

The original, cooperative FMAP algorithm ended its execution after finding a solution plan. For the preference-based version, we modified the stop criterion, allowing agents to proceed searching for better solution plans until a timeout is reached. This feature will be used in the experimental results to better assess the quality of the different solution plans obtained.

In the following, we provide more insight on how an agent i performs the individual evaluation of a refinement plan, and how social choice is applied to select a base plan that is preferred by the group of agent.

Evaluating refinement plans

In the cooperative FMAP version, refinement plans were evaluated by the agent that generated them. At the plan communication stage, all the agents received the result of the evaluation along with the refinement plan. In the preference-based version of FMAP, it is not possible to follow such a scheme, since the utility function introduced requires the refinement plans to be evaluated independently by each of the agents.

As shown in the formalization, given a plan Π, an agent i applies a utility function $f^i(\Pi) = g(\Pi) + selfInterest^i \cdot h_{pub}(\Pi) + (1 - selfInterest^i) \cdot \sum_{\forall p \in \mathcal{P}^i} (\beta_p \cdot h_{pri}(\Pi, p))$ to evaluate how a plan Π adjusts to the global goals and agent i's interests. The $g(\Pi)$ parameter stands for the number of actions of the plan; it is thus common to all the participating agents. The $selfInterest^i$ and β values are straightforwardly infered from the agent's problem file.

The heuristic values, $h_{pub}(\Pi)$ and $h_{pri}(\Pi, p)$, $\forall p \in \mathcal{P}^i$, are jointly estimated by each agent individually. In (Torreño, Onaindia, and Sapena 2014b), we introduced our novel MAP heuristic function, which is based on the concept of Domain Transition Graphs (DTGs) (Helmert 2004). The idea behind this function is to take account of the number of actions of a relaxed plan built in a backwards fashion between the frontier state of the refinement plan, $FS(\Pi)$, and the set of goals of $\mathcal{T}_{MAP}$, $\mathcal{G}$. The frontier state of a plan Π, $FS(\Pi)$, is the set of fluents $\langle v, d \rangle$ achieved by applying the actions in Π over the initial state of $\mathcal{T}_{MAP}$, $\mathcal{I}$.

We have modified the DTG heuristic function to jointly estimate both the number of actions required to reach the global goals, $\mathcal{G}$, and each of the agent's preferences. The procedure first builds the relaxed plan for $\mathcal{G}$ as described in (Torreño, Onaindia, and Sapena 2014b), and stores in $h_{pub}(\Pi)$ the number of actions of the relaxed plan. Next, the preferences of the agents are arranged according to their associated penalties, from the highest to the lowest penalty value.

The heuristic function processes the preferences in order. For each preference $p \in \mathcal{P}^i$, the procedure reuses the existing relaxed plan and adds the necessary actions for p to be reached. Once p is processed, the number of extra actions added to the relaxed plan to support this preference are summed up and returned as $h_{pri}(\Pi, p)$. The procedure uses all the information in the relaxed plan to analyze the next preferences, both the actions required to reach the goals in $\mathcal{G}$ and the ones introduced to support the previously processed preferences.

Selecting a base plan

A key aspect of the preference-based FMAP algorithm is related to the base plan selection stage. At each iteration, the agents select the refinement plan that best accommodates to the global goals and their preferences as the next base plan. Selecting such a plan implies aggregating the individual preference profiles of the agents into a global one.

Social choice theory attains the problem of a set of agents selecting a single outcome among a set of candidates according to their individual preferences (Shoham and Leyton-Brown 2009). More precisely, a *social choice function* selects a single outcome from a set of preference profiles, while a *social welfare function* aggregates a set of preference profiles into a single one. Therefore, the tools and mechanisms provided by the social choice theory fit into the problem of a set of self-interested agents selecting a base plan.

Social choice mechanisms are democratic voting systems by which the alternative preferred by the voters is chosen. One key result in social choice theory is the Arrow impossibility theorem, which states that, given three or more candidates, there is not a social welfare function that meets a specific set of criteria, namely Pareto efficiency and independence of irrelevant alternatives, without being dictatorial (Arrow 1951).

In practice, the previous result shows that there is not an ideal voting method, as it is not possible to simultaneously meet all the desirable social choice theoretical properties. Voting methods can be classified as follows:

- **Simple or sequential election**: in these methods, agents select a single outcome among their preference profiles. These mechanism perform one or more election rounds, so that each agent selects a single candidate in each round and the result of the voting is always a single winner.
- **Ranking methods**: these voting systems allow each agent to provide a complete or partial preference profile. A ranking method aggregates the agents' individual preference profiles into a joint preference profile. The first-classified outcome in the global preference profile is selected as the winner.
- **Condorcet methods**: a subset of the ranking methods accomplish the Condorcet criterion, which can be enunciated as follows: given two outcomes o_1 and o_2, if o_1 is preferred to o_2 by a majority of agents, a Condorcet-compliant method will keep the preference $o_1 \prec o_2$ (Shoham and Leyton-Brown 2009). This property also ensures that the Condorcet winner is also Pareto efficient. On the other hand, finding a Condorcet winner is not always possible, and thus, it is necessary to define a tie-breaking mechanism. Moreover, Arrow theorem shows that Condorcet methods are not independent of irrelevant alternatives (Arrow 1951).
- **Rating methods**: these mechanisms allow agents to provide a rating for each plan in their preference profiles, which makes them more flexible than ranking methods.

Different social choice methods can be tested in FMAP for the agents to jointly select the next base plan. Next section shows the preliminary results obtained after testing the preference-based FMAP along with a ranking strategy.

Experimental results

This section provides the preliminary experimental results we collected. The tests assess the performance of the

Domain	Pfile	First solution plan					Best solution plan				
		Actions	Makespan	Metric	Time	Iter	Actions	Makespan	Metric	Time	Iter
Driverlog	1	13	9	10	3,66	177	20	7	7,5	56,54	3300
	2	13	7	8	4,91	232	19	7	7	173,58	8784
	3	10	7	8	1,20	42	13	7	7	6,38	291
	4	12	5	6,3	1,82	51	13	5	5,7	81,89	2778
	5	13	7	8,3	3,30	81	16	7	7,7	27,94	928
Elevators	1	11	7	8,3	3,26	63	13	7	7,3	19,12	436
	2	6	6	7	7,41	119	10	3	3,3	71	1513
	3	4	4	5,3	0,68	5	8	4	4,7	7,26	118
	4	11	6	7	20,00	355	-	-	-	-	-
	5	17	8	9,3	57,75	990	17	7	8	75,78	1289
Zenotravel	3	6	5	5,5	1,22	27	8	4	4	15,07	433
	4	5	4	5	1,63	41	10	4	4	24,64	845
	5	9	5	6	3,75	103	11	5	5,5	9,67	295
	6	6	3	4	1,26	15	10	3	3,5	12,03	300
	7	14	7	8	14,29	398	16	7	7	101,29	2929

Table 1: Experimental results

preference-based FMAP system by using three different planning domains adapted from the International Planning Competitions[1] (IPC) benchmarks.

FMAP is entirely encoded in Java, and it makes use of *Magentix2*[2] (Such et al. 2012), a middleware multi-agent platform that provides the communication services required by the agents. Magentix2 agents communicate by means of the FIPA Agent Communication Language (O'Brien and Nicol 1998). Messaging is carried out through the Apache QPid broker[3].

All the experimental tests were performed on a single machine with a quad-core Intel Core i7 processor and 8 GB RAM. This machine runs both the complete FMAP system and the QPid broker. We set up the planner by establishing a 5-minute time limit for each planning task, so that agents are allowed to find as many solutions as possible within the time limit. For each task, we take account of the first and best solution (measured by computing the agents' average metric for each solution plan) obtained by the agents within the time limit.

We selected three different domains from the IPC benchmarks and adapted five of the STRIPS tasks to a preference-based MAP context. We modeled some of the goals in the original problems as agent-specific preferences, and added extra preferences so that each agent has two different preferences. All the preferences have 1 unit of associated penalty, and the metric threshold is set to 1, meaning that an agent will approve solution plans that meet at least one of its preferences. The self-interest value is set to 0.5 for all the agents, and thus, they assign the same weight to their preferences and the global goals.

The specific design guidelines we applied to adapt each domain are described as follows:

- **Driverlog** (pfiles 1-5): the `driver` objects of the original domain are defined as agents. The preferences are defined through the predicates `(at-driver driver)` and `(pos truck)`.
- **Elevators** (pfiles 1-5): each `slow-elevator` and `fast-elevator` is defined as an agent. The agents' individual preferences were established using the predicates `(lift-at elevator)` and `(at passenger)`.
- **Zenotravel** (pfiles 3-7): The `aircraft` objects are defined as agents in the *zenotravel* MAP version. Preferences are modeled by means of the predicates `(at aircraft)` and `(in person)`.

Table 1 summarizes the early results we collected. For each MAP task, we provide information about the first and best solution plan found by FMAP according to the average of the agents' metric values. *Actions* and *Makespan* columns refer to the number of actions and duration of the solution plans. *Metric* stands for the average metric value of the solution plan. Finally, *Time* and *Iter* indicate the execution time and number of iterations required by FMAP to find each solution plan.

The social choice method selected for these tests is a Borda voting (Shoham and Leyton-Brown 2009), one of the most common ranking methods. For these experiments, we modified the Borda method in order to use incomplete preference profiles. Using complete preference profiles entails processing all the leaf nodes of the search tree at each iteration of the FMAP procedure, which is too complex even in middle-range tasks. For this reason, we limited the preference profile used by each agent in the voting to a fixed number of 10 candidates.

As shown in Table 1, agents find the first solution that meets the metric thresholds in a matter of seconds (less than one minute in all the tasks). This first solution found by the agents tends to adjust to the metric threshold defined for the majority of agents; that is, a majority of the agents solve one

---

[1] http://ipc.icaps-conference.org/

[2] http://www.gti-ia.upv.es/sma/tools/magentix2

[3] http://qpid.apache.org/

of their preferences, while the rest of agents do not attain any preference.

The best solution for each task offers a more polished plan that in most cases increases the number of actions, but, in turn, offers an equal or lower duration and attains a higher amount of the agents' preferences. Agents find the best solution in less than a half of the time limit in most cases. In some tasks, the average metric equals the plan duration (makespan), which means that all the preferences of the agents are fulfilled (and thus, the metric only takes account of the plan duration).

Conclusions and future work

In this work, we presented an extension of FMAP, a cooperative MAP system, to support individual preferences. We extended our MAP definition language, based on *PDDL3.1*, to include new constructs to define the agents' preferences and behaviour.

The refinement planning procedure of FMAP has been adapted to a preference-based context. Each agent makes use of a utility function to measure how a plan adjusts to the global goals and its private preferences. The application of this utility function gives rise to an individual preference profile for each of the agents. We apply social theory-based mechanisms in order to aggregate the agents' preference profiles and to select the most appropriate base plan according to the interests of the group of agents at each iteration of the procedure. More precisely, agents apply voting methods to democratically elect the candidate plan that is preferred by the group.

Finally, we provide some early experimental results that focus on solving simple tasks from the IPC benchmarks adapted to a MAP context with preferences. In these experiments, agents apply a Borda voting, a well-known ranking method, to select plans.

This article presents a very early stage of our preference-based MAP work. We have already modified and tested the FMAP framework and the definition language to accommodate individual preferences. However, at this point we have just a single functional coordination method.

We intend to add other more complex social choice schemes that ensure different theoretical properties, such as Condorcet-compliant ranking methods and rating mechanisms. Moreover, we will develop an in-depth experimental comparison to study how the search procedure and the quality of the solution plans are affected by the behavioral parameters defined for each agent, such as the social choice mechanism, the level of self-interest or the metric threshold.

Acknowledgments

This work has been partly supported by the Spanish MICINN under projects Consolider Ingenio 2010 CSD2007-00022 and TIN2011-27652-C03-01, the Valencian Prometeo project II/2013/019, and the FPI-UPV scholarship granted to the first author by the Universitat Politècnica de València.

References

Arrow, K. 1951. Individual values and social choice. *Wiley, New York.*

Baier, J., and McIlraith, S. 2009. Planning with preferences. *AI Magazine* 29(4):25.

Borrajo, D. 2013. Multi-agent planning by plan reuse. In *Proceedings of the 12th International Conference on Autonomous Agents and Multi-agent Systems (AAMAS)*, 1141–1142.

Brafman, R.; Domshlak, C.; Engel, Y.; and Tennenholtz, M. 2009. Planning games. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, 73–78.

Fox, M., and Long, D. 2003. PDDL2.1: an extension to PDDL for expressing temporal planning domains. *Journal of Artificial Intelligence Research* 20:61–124.

Gerevini, A., and Long, D. 2005. Plan constraints and preferences in PDDL3. *Technical Report, Department of Electronics for Automation, University of Brescia, Italy.*

Ghallab, M.; Nau, D.; and Traverso, P. 2004. *Automated Planning. Theory and Practice*. Morgan Kaufmann.

Helmert, M. 2004. A planning heuristic based on causal graph analysis. In *Proceedings of the 14th International Conference on Automated Planning and Scheduling (ICAPS)*, 161–170.

Kambhampati, S. 1997. Refinement planning as a unifying framework for plan synthesis. *AI Magazine* 18(2):67–97.

Kovacs, D. L. 2011. Complete BNF description of PDDL3.1. Technical report.

Nissim, R.; Brafman, R.; and Domshlak, C. 2010. A general, fully distributed multi-agent planning algorithm. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 1323–1330.

O'Brien, P., and Nicol, R. 1998. FIPA - towards a standard for software agents. *BT Technology Journal* 16(3):51–59.

Shoham, Y., and Leyton-Brown, K. 2009. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press.

Such, J.; García-Fornes, A.; Espinosa, A.; and Bellver, J. 2012. Magentix2: A privacy-enhancing agent platform. *Engineering Applications of Artificial Intelligence* 96–109.

Torreño, A.; Onaindia, E.; and Sapena, O. 2012. An approach to multi-agent planning with incomplete information. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI)*, volume 242, 762–767. IOS Press.

Torreño, A.; Onaindia, E.; and Sapena, O. 2013. FMAP: a heuristic approach to cooperative multi-agent planning. In *Proceedings of the 1st Workshop on Distributed and Multi-Agent Planning (DMAP 2013)*, 84–92.

Torreño, A.; Onaindia, E.; and Sapena, O. 2014a. A flexible coupling approach to multi-agent planning under incomplete information. *Knowledge and Information Systems* 38(1):141–178.

Torreño, A.; Onaindia, E.; and Sapena, O. 2014b. FMAP: distributed cooperative multi-agent planning. *Applied Intelligence*.