Onaindia De La Rivaherrera, E.; Sapena Vercher, O.; Torreño Lerma, A. (2010). Cooperative Distributed Planning through Argumentation. International Journal of Artificial Intelligence. 4(10):118-136. http://hdl.handle.net/10251/99676

Cooperative Distributed Planning through Argumentation

Eva Onaindia and Oscar Sapena and Alejandro Torreño

Departamento Sistemas Informaticos y Computacion
Universidad Politecnica Valencia
Camino de Vera s/n
46022 Valencia, SPAIN
{onaindia,osapena,atorreno}@dsic.upv.es

ABSTRACT

This paper addresses the problem of solving a cooperative distributed planning (CDP) task through an argumentation-based model. A CDP task involves building a central plan amongst a set of agents who will contribute differently to the joint task based on their abilities and knowledge. In our approach, planning agents accomplish the CDP task resolution through an argumentation-based model that allows them to exchange partial solutions, express opinions on the adequacy of the agents' solutions and adapt their own proposals for the benefit of the overall task. Hence, the construction of the joint plan is coordinated via a deliberation dialogue to decide what course of action should be adopted at each stage of the planning process. In this paper, we highlight the role of argumentation for planning tasks that require a coordinated behaviour for their resolution.

Keywords: Practical Reasoning, Distributed Planning, Conflict Management and Negotiation, Argumentation-based Decision Support Systems.

2000 Mathematics Subject Classification: 68T20.
1998 Computing Classification System: I.2.8, I.2.11.

1 Introduction

Planning is the art of building control algorithms that synthesize a course of action to achieve a desired set of goals. The mainstream in practical planning is that of using utility functions, which are usually called heuristics, to evaluate goals, and choices of action or states on the basis of their expected utility to the planning agent (Ghallab, Nau and Traverso, 2004). The application of a fixed heuristic function allows the planning agent to filter out the alternatives that are not regarded as acceptable by the agent, or, more precisely, to select the most preferable alternatives.

Distributed planning generalizes the problem of planning in domains where several agents plan and act together, and have to share resources, activities, and goals. In a Cooperative Distributed Planning (CDP) approach, where the agents are assumed to be cooperative, the emphasis is placed on how planning can be extended to a distributed environment. The planning agents of a CDP task exchange information about their plans, which they iteratively refine and revise until they fit together (desJardins, Durfee, Ortiz and Wolverton, 1999). Typically, research in Distributed Planning has been more concerned with the design of distributed planning architectures, mechanisms for plan coordination, or solutions for merging the resulting local plans of agents into a global plan (Durfee, 1999; Durfee, 2001; Cox, Durfee and Bartold, 2005; de Weerdt, ter Mors and Witteveen, 2005). Unlike these approaches, which emphasize the problem of controlling and coordinating a posteriori local plans of independent agents, our proposal focuses on argumentation mechanisms that allow agents to jointly devise a global shared plan, i.e., to decide what course of action should be adopted at each stage of the planning process.

Argumentation-based frameworks have been used for reasoning about what actions are the best to be executed by an agent in a given situation. Dung's abstract system for argumentation (Dung, 1995) has been applied on reasoning about conflicting plans and for generating consistent sets of intentions from a contradictory set of desires (Amgoud, 2003; Hulstijn and van der Torre, 2004). The work in (Simari, García and Capobianco, 2004) presents a defeasible argumentation framework for the definition of actions and the combination of these actions into plans. Other approaches like (Atkinson and Bench-Capon, 2007) propose undertaking practical reasoning, i.e., reasoning about what to do, through the instantiation of an argument scheme and associated critical questions (Walton, 1996; Atkinson, Bench-Capon and McBurney, 2006). In all of the aforementioned works, arguments are not about whether some belief is true, but about whether some action should or should not be performed.

We propose an argumentation model in which agents seek to jointly agree on a course of action via a deliberation dialogue (McBurney, Hitchcock and Parsons, 2007), and whose final objective is to form a competent global plan through the composition of the individual plans proposed by the participants. Following the work in (Atkinson et al., 2006; Atkinson and Bench-Capon, 2007), we interpret the instantiation of an argument scheme and associated challenges through the semantics of a partial-order planning paradigm (Barrett and Weld, 1994; Ghallab et al., 2004). Agents will use an argument scheme as a presumptive justification of the next best step towards a solution plan. Challenges associated to the argument scheme will allow the rest of the agents to attack or to defend this justification. The contribution of this paper is thus to present an argumentation model for cooperatively solving a distributed planning task through the instantiation of argumentation schemes to partial plans in order to identify how, when, and with whom agents can contribute to the construction of the final plan. The ultimate objective is to enhance the role of argumentation in a deliberative process that seeks common agreement amongst all of the participating planning agents.

The paper is organized as follows. Section 2 presents the formalization of a planning task. Then, on the basis of this formalization, we define a CDP organization, our working scenario. Section 3 presents the argumentation model for a CDP task. Following Walton's account of practical reasoning, we define the instantiation of an argument scheme in a planning context, and we provide formal semantics to the critical questions associated to this scheme. Section 4 outlines the interaction protocol followed by the agents to solve a CDP task, and section 5 presents a complete application example. Section 6 outlines some ideas on how our distributed planning framework can be developed to account for non-cooperative agents. Finally, in section

7 we offer some some concluding remarks.

2 Formalization of a planning task

Recently, a number of attempts have been made to use argumentation to capture practical reasoning, that is, reasoning about which actions are the best for a particular agent to do in a given situation. These logic-based argumentation frameworks require capturing arguments for generating consistent desires and plans in order to achieve the agent's desires (Hulstijn and van der Torre, 2004; Bench-Capon and Prakken, 2006). Desires are adopted if the world state justifies the appropriate conditions for the desire and the agent has an executable plan to achieve the desire (Rahwan and Amgoud, 2006). In contrast, reasoning in planning is conducted by a goal-driven logic of action where goals are desires combined with commitments, and the agent's task is to find a world state and a course of action that is executable in this state that achieves the goals. Thus, our work focuses exclusively on the generation of plans of action to achieve the goals allocated to the agents.

Our system takes as input an action domain description in which actions are described by their precondition axioms and direct effect axioms. In particular, we consider the STRIPS language (Fikes and Nilsson, 1971), where an action is described by a first-order formula that describes the conditions under which the action is executable (the action's preconditions), and also describes an add list and a delete list that enumerate the propositions that the action will make true and false, respectively, when the action is successfully executed in a situation. Although the STRIPS assumption imposes some limitations for describing actions in the real world, it is still a valid formalism for expressing the consequences of actions and the conditions under which a goal is achievable.

The base entities in STRIPS planning are facts, i.e., predicates whose arguments are constants or variables. A ground fact, or proposition, is a fact where all its arguments are constants. We will denote the set of all propositions by $\mathcal{P}$. A planning state s is defined as a finite set of true propositions $s \subseteq \mathcal{P}$. An action a is a tuple $a = (pre(a), add(a), del(a), cost(a))$, where $pre(a) \subseteq \mathcal{P}$ is the set of propositions that represents the action's preconditions, $add(a) \subseteq \mathcal{P}$ and $del(a) \subseteq \mathcal{P}$ are the sets of propositions that represent the positive and negative effects, respectively, and $cost(a)$ is the action cost.

A planning agent is assigned a (grounded) planning task as a triple $\mathcal{T} = \langle \Omega, \mathcal{I}, \mathcal{G} \rangle$, where Ω is the set of deterministic actions of the agent's model that describes the state changes, and $\mathcal{I} \subseteq \mathcal{P}$ (the initial state) and $\mathcal{G} \subseteq \mathcal{P}$ (the goals) are sets of propositions. A solution plan for a planning task $\mathcal{T}$ is a plan or set of actions $\Pi = \{a_1, \ldots, a_n\} \in \Omega^*$ such that when applied to $\mathcal{I}$, it leads to a final state in which $\mathcal{G}$ holds. A planning task $\mathcal{T}$ is solvable if there exists at least one plan for it.

In distributed planning, agents will contribute to the global plan being developed in the form of partially ordered networks of actions and subgoals (desJardins and Wolverton, 1999). In the Partial-Order Planning (POP) approach (Penberthy and Weld, 1992; Barrett and Weld, 1994), search is done in the space of incomplete partially ordered plans rather than in a state-based space. In the following, we define a partial-order plan and all the elements that are involved, as this will be the core of our argumentation model.

Definition 1. A **partial order plan** is a triple $\Pi = \langle \mathcal{A}, \mathcal{OR}, \mathcal{CL} \rangle$, where $\mathcal{A} \subseteq \Omega$ is the set of ground actions[1], $\mathcal{OR}$ is a set of ordering constraints ($\prec$) over $\mathcal{A}$, and $\mathcal{CL}$ is a set of causal links over $\mathcal{A}$. A causal link is of the form (a_i, p, a_j), and denotes that the precondition p of action a_j will be supported by an add effect of action a_i.

This structural definition of a partial order plan actually represents a mapping of a plan into a directed acyclic graph, where $\mathcal{A}$ represents the nodes of the graph and $\mathcal{OR} \subseteq \mathcal{A} \times \mathcal{A}$ and $\mathcal{CL} \subseteq \mathcal{A} \times \mathcal{A}$ are sets of directed edges representing the required precedences of these actions and the causal links among them, respectively.

Definition 2. A partial order plan $\Pi = \langle \mathcal{A}, \mathcal{OR}, \mathcal{CL} \rangle$ is **conflict-free** if it has no unsafe causal links and there exists a total ordering of the actions in $\mathcal{A}$ that is consistent with $\mathcal{OR}$ (i.e., topological sort). Let $(a_i, p, a_j) \subseteq \mathcal{CL}$; (a_i, p, a_j) is unsafe if there exists an action $a_k \in \mathcal{A}$ such that $p \in del(a_k)$ and $\mathcal{OR} \cup \{a_i \prec a_k \prec a_j\}$ is consistent.

An unsafe causal link indicates that an action might possibly interfere with the precondition being supported by the causal link. In such a case, it is said that a_k *threatens* the causal link and hence the proposition supported by the causal link. A causal link flaw can be repaired by either *promotion*, i.e., adding the ordering constraint $a_k \prec a_i$ to $\mathcal{OR}$, or *demotion*, i.e., adding $a_j \prec a_k$ to $\mathcal{OR}$. In what follows, the function $\texttt{threats}(\Pi)$ returns the unsafe link flaws of a given plan Π.

Definition 3. A partial order plan $\Pi = \langle \mathcal{A}, \mathcal{OR}, \mathcal{CL} \rangle$ is **incomplete** if it has open conditions. Let $a_i \in \mathcal{A}$; if $\exists p \in pre(a_i) \wedge \nexists a_j \in \mathcal{A} / (a_j, p, a_i) \subseteq CL$, then p is said to be an open precondition. On the contrary, a partial order plan Π is **complete** if it has no open conditions.

A precondition of an action that has not yet been satisfied is called an open condition. There are two general mechanisms for solving an open condition p: either an action that is already in $\mathcal{A}$ has a positive effect that unifies with p, or a new action that supports p must necessarily be inserted in Π. In what follows, the function $\texttt{open\_cond}(\Pi)$ returns the open condition flaws of a given plan Π.

Definition 4. A partial order plan $\Pi = \langle \mathcal{A}, \mathcal{OR}, \mathcal{CL} \rangle$ is a **solution plan** for a planning task $\mathcal{T}$ if Π is conflict-free and complete.

Given a planning task $\mathcal{T} = \langle \Omega, \mathcal{I}, \mathcal{G} \rangle$, a POP algorithm starts with an empty partial plan and keeps refining it until a solution plan is found. The initial empty plan $\Pi_0 = \langle \mathcal{A}, \mathcal{OR}, \mathcal{CL} \rangle$ contains only two dummy actions $\mathcal{A} = \{a_0, a_f\}$, where $pre(a_f) = \mathcal{G}$, $add(a_0) = \mathcal{I}$, $\mathcal{CL} = \emptyset$, $\{a_0 \prec a_f\} \subseteq \mathcal{OR}$, $\texttt{threats}(\Pi_0) = \emptyset$ and $\texttt{open\_cond}(\Pi_0) = \mathcal{G}$. The empty plan has no causal links or threat flaws, but, has open condition flaws corresponding to the preconditions of a_f (the top-level goals $\mathcal{G}$).

A POP algorithm works through the application of successive refinement steps at each iteration. A refinement step involves selecting a flaw (threat or open condition) in a partial order

---

[1] Partial-order planners are capable of handling partially instantiated action instances and hence, the definition of a partial order plan typically includes a set of equality constraints on free variables in $\mathcal{A}$ (Penberthy and Weld, 1992). We will, however, restrict our attention to ground action instances without any loss of generality for our purposes.

plan $\Pi_i = \langle \mathcal{A}_i, \mathcal{CL}_i, \mathcal{OR}_i \rangle$ and resolving it by adding a new ordering constraint in $\mathcal{OR}_i$, a new causal link in $\mathcal{CL}_i$ or a new action in $\mathcal{A}_i$, thus resulting in a new partial plan Π_j. A refinement step can therefore be regarded as a plan structure, and Π_j as the result of the composition of two plans.

Definition 5. A plan $\Pi_j = \langle \mathcal{A}_j, \mathcal{CL}_j, \mathcal{OR}_j \rangle$ is a **refinement** of a plan $\Pi_i = \langle \mathcal{A}_i, \mathcal{CL}_i, \mathcal{OR}_i \rangle$ for a planning task $\mathcal{T}$ if and only if $\mathcal{A}_i \subseteq \mathcal{A}_j$, $\mathcal{OR}_i \subseteq \mathcal{OR}_j$ and $\mathcal{CL}_i \subseteq \mathcal{CL}_j$.

A refinement plan Π_j is actually a composite plan resulting from the composition of Π_i, the *base plan*, and a *refinement step* Π', where $\Pi' = \langle \mathcal{A}', \mathcal{CL}', \mathcal{OR}' \rangle$ and $\mathcal{A}_j = \mathcal{A}_i \cup \mathcal{A}'$, $\mathcal{CL}_j = \mathcal{CL}_i \cup \mathcal{CL}'$ and $\mathcal{OR}_j = \mathcal{OR}_i \cup \mathcal{OR}'$. We will denote the plan composition as $\Pi_j = \Pi_i \circ \Pi'$.

Since POP algorithms monotonically increase the set of causal links and actions, each iteration returns a refinement of the initial plan Π_0 (Penberthy and Weld, 1992). It is important to note that a larger number of actions, causal links or orderings of a composite plan Π_j may embody a step ahead towards a solution, but it does not necessarily imply a net gain over Π_i in terms of open conditions or unsafe causal links. Heuristics are specifically used at this point to estimate the difficulty of reaching a solution from a given plan, and POP approaches usually apply heuristics that consider the number of pending open conditions and threats (Gerevini and Schubert, 1996).

2.1 Cooperative Distributed Planning

A Cooperative Distributed Planning (CDP) organization, as defined in this paper, is formed by a group of non-self-interested planning agents who join together to achieve the goals $\mathcal{G}$ of a planning task $\mathcal{T} = \langle \Omega, \mathcal{I}, \mathcal{G} \rangle$. A planning agent is an entity with planning capabilities and as such can be assigned a planning task and solve it. However, in our CDP organization, there is no specific assignation of (sub)planning tasks to agents because there is no specific allocation of a subset of goals from $\mathcal{G}$ to each agent. In our approach to CDP, $\mathcal{G}$ is a joint goal set that exists in all agents by design. Agents cooperate either because they do not have a complete view of the world or simply because solving a problem that requires collective effort, like the construction of a plan, is better accomplished when working with other agents (Durfee, 2001).

Definition 6. A **CDP task** is defined as a tuple $\langle \mathcal{AG}, \Omega, \mathcal{I}, \mathcal{G} \rangle$ where $\mathcal{AG} = \{ag_1, ag_2, \ldots, ag_n\}$ is a finite, non-empty set of agents such that $\mathcal{I} = \bigcup_{\forall ag_i \in \mathcal{AG}} \mathcal{I}_i$, and $\Omega = \bigcup_{\forall ag_i \in \mathcal{AG}} \Omega_i$.

This definition states that the initial state of the problem is the union of the initial agent's beliefs, and the actions of the CDP task is the union of the individual actions of all agents. Distributed information, expertise, or resources are common characteristics in inherently distributed domains although they are not mandatory requirements for a CDP organization. We want to highlight that in our CDP conception, distribution is related to a dynamic allocation of goals among the agents in $\mathcal{AG}$, i.e., there is not an a priori static allocation of individual goals to each agent involved in the problem.

Goal or task allocation is one of the activities that has received much attention in distributed planning, typically being undertaken through a resource allocation process. There are, however, other methods to establish such a task assignment in a more distributed way, e.g., via agent coalition formation (Shehory and Kraus, 1998), market mechanisms to analyze economic efficiency in decentralized task allocation (Walsh and Wellman, 1999), or via contracting mechanisms (Lesser, Decker, Wagner, Carver, Garvey, Horling, Neiman, Podorozhny, NagendraPrasad, Raja, Vincent, Xuan and Zhang, 2004). In our approach, the problem of determining which agent is more competent to solve a particular goal is dynamically solved during the argumentation process. At this stage, agents analyze how a particular choice of action, and the order in which the actions are executed will affect the overall performance and ensure the long-term effectiveness of the plan under construction. The objective of the argumentation process is thus to optimize the agents' combined efforts.

The core elements in our CDP organization are the partial order plans proposed and exchanged by the agents, which they iteratively refine and revise until they fit together. Given a CDP task $\mathcal{T} = \langle \mathcal{AG}, \Omega, \mathcal{I}, \mathcal{G} \rangle$, the starting base plan for solving $\mathcal{T}$ is the initial empty plan Π_0. Next, an agent in $\mathcal{AG}$ makes a refinement step proposal to solve any open conditions in Π_0. An argumentation process is then initiated by the agents to attack or support this agent proposal. Once a refinement step proposal Π_i, that is argued by agent ag_i is approved, a composite plan $\Pi_0 \circ \Pi_i$ is created as the new base plan. The process is subsequently repeated until $\mathcal{T}$ is solved. Therefore, a solution plan for $\mathcal{T}$ is a joint plan that is formed by a collection of partial plans argued by the agents in the society such that the composition of these partial plans solves $\mathcal{T}$.

Definition 7. Given a CDP task $\mathcal{T} = \langle \mathcal{AG}, \Omega, \mathcal{I}, \mathcal{G} \rangle$, a joint solution plan $\Pi_\mathcal{T}$ for $\mathcal{T}$ is a sequence of pairs $\Pi_\mathcal{T} = \langle (\Pi_0, ag_i), \ldots, (\Pi_l, ag_k) \rangle$ where:

1. A pair $(\Pi_i, ag_j) \subseteq \Pi_\mathcal{T}$ indicates that the plan Π_i is a refinement step that is argued by some ag_j, and finally accepted by all agents in $\mathcal{AG}$
2. Given two pairs (Π_i, ag_j) and (Π_k, ag_l) in $\Pi_\mathcal{T}$, $ag_j = ag_l$ or $ag_j \neq ag_l$
3. The result of $\Pi_0 \circ \Pi_1 \circ \ldots \circ \Pi_l$ is a conflict-free and complete plan that achieves $\mathcal{G}$

A coordinated behaviour in a CDP organization is achieved through cooperation. In our case, cooperation is implicitly encoded in the composition function; the contribution of an agent must fit the previously approved plan as well as conduct the composite plan towards a joint solution plan. We identify three different modes of cooperation:

- Assistance. Overcome the limitations of an agent to achieve a particular goal by delegating the task to other agents. This is the case of having an agent achieve the open conditions of the refinement step of another agent.
- Synchronization. Agents agree on some coordination to synchronize their actions so as to avoid negative interactions (threats) in their respective contribution plans.
- Improvement. Agents may offer an alternative choice of action to achieve the goal attained by another agent when the proposed alternative turns out to be a more efficient solution than the original plan proposal (in terms of optimization criteria).

Assistance is likely to be the cooperation mode that best defines coordinated behaviour. In our CDP organization, assistance is implicitly encoded in the agent's behaviour without the need of an explicit request for help to others. *Synchronization* is a common coordination mechanism in planning systems to avoid conflicting goals or conflicting situations, particularly, in plan merging methods aimed at the construction of a joint plan, given the individual plans of each of the participants. Finally, *improvement* is subject to internal agent modeling, which typically follows a specific optimization metric such as time cost, resource cost, or net-benefit.

The refinement steps contributed by the agents to the joint plan must be compliant with the cooperation modes besides ensuring that the construction of the joint plan progresses successfully towards a solution plan. Let (Π', ag_k) be the refinement step contributed by agent ag_k to a base plan Π_i:

- Π' may be an incomplete plan, i.e., it may contain open conditions that, hopefully, will be solved by other agents through assistance during the argumentation process.
- Π' is a conflict-free plan such that $|\texttt{threats}(\Pi_i \circ \Pi')| = 0$, that is, the composite plan $\Pi_i \circ \Pi'$ must be a conflict-free plan as well. The possible synchronization interactions with other agents' proposals will also be revealed during the argumentation process.
- $\Pi_i \circ \Pi'$ must improve the current base plan, i.e., promote progress towards a joint solution plan. Therefore, Π' must solve at least one open condition in Π_i.

Definition 8. A plan $\Pi_j = \Pi_i \circ \Pi'$ is a **valid refinement** of a plan Π_i for a planning task $\mathcal{T}$ iff Π_j is a refinement of Π_i, Π_j is conflict-free, and there exists a set of propositions P such that $P \subseteq \texttt{open\_cond}(\Pi_i) \wedge P \not\subseteq \texttt{open\_cond}(\Pi_j)$. Note that $\texttt{open\_cond}(\Pi_j) = (\texttt{open\_cond}(\Pi_i) \setminus P) \cup \texttt{open\_cond}(\Pi')$.

3 Argumentation-based model for CDP

In this section, we propose an adaptation of the computational representation of practical argumentation presented in (Atkinson et al., 2006; Atkinson and Bench-Capon, 2007) for solving a CDP task. These works, which present an extension to the explanation of presumptive reasoning in terms of an argument scheme and associated critical questions popularized by Walton (Walton, 1996), are very suitable for representing the central piece of our argumentation-based model, a partial-order plan. Nevertheless, we will point out some different interpretations and the necessary adaptations to fit a CDP task.

This section is divided into two subsections. The first one presents the instantiation of argument schemes to partial-order plans. The second subsection details the challenges that agents will pose to a given argument.

3.1 Instantiation of argument schemes

Our argumentation framework is aimed at deciding whether the refinement step contributed by an agent to a base plan is a good alternative at each choice point. Given a current base plan, which is initially the empty plan Π_0, an agent can suggest a refinement over the base plan that, according to its beliefs and capabilities, represents a good step toward the goals. In most cases, this is a presumptive argument since agents do not usually have complete knowledge of the world and the capabilities of the other agents. Formally, we define a partial plan argument as follows:

Definition 9. Given a CDP planning task $\mathcal{T} = \langle \mathcal{AG}, \Omega, \mathcal{I}, \mathcal{G} \rangle$, a partial plan argument is a pair $PPA = \langle \Pi_i, \Pi' \rangle$, where plan Π_i is a valid refinement of the initial plan Π_0 for $\mathcal{T}$ and $\Pi_i \circ \Pi'$ is a valid refinement of Π_i.

Basically, this definition states an argument as the presumptive partial plan put forward by an agent, Π', as a contribution to the joint plan under construction, Π_i. Similarly to the argument scheme developed in (Atkinson et al., 2006), we define an argumentation scheme (AS) that responds to our definition of argument. Thus, we have:

AS In the current circumstances and considering the current base plan Π_i
Agent ag_i should perform the refinement step Π'
Which will result in a new partial plan Π_j
Which will realise some subgoals G
Which will promote some values V

The argument scheme is interpreted as follows:

- The current circumstances are given by, $\mathcal{I}$, the initial state of the problem. Even though agents may have only a partial knowledge of the world, their beliefs are always true and the only mechanism for changing their view of the world is through the execution of planning actions. On the other hand, Π_i is the last base plan approved by all the agents over which they will articulate their arguments to continue the construction of the joint plan. The current circumstances along with the base plan make up the support of the partial plan argument.

- Π' represents the refinement step argued by agent ag_i, i.e., the presumptive contribution of ag_i.

- $\Pi_j = \Pi_i \circ \Pi'$ is a valid refinement, the plan that results from the composition of the plan base and the refinement suggested by ag_i. Contrary to the action-based alternating system (van der Hoek, Roberts and Wooldridge, 2007) that is used in (Atkinson and Bench-Capon, 2007), states are not explicitly represented in plan-based approaches. However, Π_j represents itself the new current planning situation, and, hence, it maintains valuable information such as the open conditions or side-effects of the new planning snapshot.

- G is the set of goals achieved by Π', i.e., $G = \texttt{open\_cond}(\Pi_i) \setminus \texttt{open\_cond}(\Pi_j)$; it contains the solved open preconditions of Π_i, some of which can be top-level goals from $\mathcal{G}$. Note we admit a set of goals rather than a single goal because a planning action may achieve more than one goal, which in terms of planning optimization is always preferable.

- V is a set of two function values, `cost` and `progress`:
  - $\texttt{cost} : \Pi_j \to \mathbb{R}_0^+$ represents the cost of the plan Π_j. The utility function defined in the agent planning model is to minimize the cost of the final plan, where `cost` is defined as a generic numeric function whose semantics depends on the planning problem. For instance, if `cost` represents the fuel consumption in a transportation problem, the cost of a plan $\Pi_j = \langle \mathcal{A}_j, \mathcal{OR}_j, \mathcal{CL}_j \rangle$ is calculated as $\texttt{cost}(\Pi_j) = \sum_{\forall a \in \mathcal{A}_j} \texttt{cost}(a)$. However, if `cost` is the plan duration, the plan cost is computed as the duration of the longest sequence of actions in Π_j. It is important to note that, in case Π_j is an incomplete plan, this function does not account for the cost of the potential actions which will be further inserted to solve the pending open conditions in Π_j. Therefore, accurate heuristic functions to estimate the cost of the further refinements of Π_j would be desirable, but this is out of the scope of the paper.
  - $\texttt{progress} : \Pi_i \times \Pi' \to \mathbb{R}_0^+$ represents the progress brought by the refinement step Π' to the base plan Π_i. Assuming $\Pi_j = \Pi_i \circ \Pi'$, $\texttt{progress}(\Pi_i, \Pi')$ represents a progression step of Π_i towards a solution, i.e. the improvement of Π_j with respect to Π_i. More particularly, $\texttt{progress}(\Pi_i, \Pi')$ measures the number of pending open conditions in Π_i (top-level goals or subgoals) which are solved with the composite plan Π_j. Given two refinement steps, Π' and Π'', over Π_i, such that $\Pi_j = \Pi_i \circ \Pi'$ and $\Pi_k = \Pi_i \circ \Pi''$, $\texttt{progress}(\Pi_i, \Pi') > \texttt{progress}(\Pi_i, \Pi'')$ if the composite plan Π_j solves more open conditions of Π_i than Π_k. In case that both Π_j and Π_k solve the same number of goals in Π_i, we will consider that the composite plan that leaves fewer open conditions progresses more than the other one towards a solution. In other words, the fewer pending open conditions of a composite plan, the closer will be the plan to a solution and, therefore, the greater the progress of the plan.

Now, we have all necessary ingredients to initiate the debates among agents in order to select the most appropriate partial plan at each particular choice point.

3.2 Argument evaluation

The specific situation that we consider is the following: given a CDP task $\mathcal{T}$ and a partial plan argument $PPA = \langle \Pi_i, \Pi' \rangle$ proposed by agent ag_i, agents will try to persuade or convince each other of the convenience of proceeding through $\Pi_j = \Pi_i \circ \Pi'$ to solve $\mathcal{T}$. We will see how to adapt the critical questions or attacks defined in (Atkinson et al., 2006) to our argumentation framework as well as the necessity of defining a defence relationship.

Let us first analyze the nature of the possible attacks. As mentioned above, the lack of uncertainty and the use of deterministic planning actions in our model allow us to rule out the consideration of critical questions concerned with the problem of formulating or denying of premises. This leads us to affirm there are neither discrepancies about the current circumstances or base plan nor disputes about the truth of the goals, side-effects, open conditions, or values achieved by Π'. Therefore, we only have to consider the critical questions concerned with the choice of action, as is the case in a classical planning formalism.

The first point to note is that our argument scheme accounts for all the goals achieved by Π' as well as for all the consequences raised by such a plan. Hence, since all possible effects are covered and encoded in Π', there is no need to attack a proposal for some unconsidered consequence. In short, our framework specifically deals with disputes on the alternatives used by Π' to achieve the elements of the argument and with the conflicting situations that prevent Π_j from progressing.

When an argument receives a **negative evaluation**, this comes together with a *justification*. It is important to note that, in a CDP organization, we must promote progress towards a minimal cost solution plan. Hence, a negative evaluation is articulated in case the attacking agent provides a reasonable justification of its attack. In other words, an attack is only put forward by an agent if it can justify that the attack presumptively represents an advantage over the original proposal. More specifically, given the argument scheme AS described above, we can define the following negative evaluations:

Negative Evaluation 1: alternative way of achieving G (NE1)

- **Attack**: There is another refinement step Π'' such that $\Pi_i \circ \Pi''$ is a valid refinement of Π_i and $\texttt{open\_cond}(\Pi_i) \setminus \texttt{open\_cond}(\Pi_i \circ \Pi'')$ is the same set as G.
- **Justification**: Π'' promotes *cost*, i.e., $\texttt{cost}(\Pi_i \circ \Pi'') < \texttt{cost}(\Pi_i \circ \Pi')$

Negative Evaluation 2: alternative way of achieving G (NE2)

- **Attack**: There is another refinement step Π'' such that $\Pi_i \circ \Pi''$ is a valid refinement of Π_i and $\texttt{open\_cond}(\Pi_i) \setminus \texttt{open\_cond}(\Pi_i \circ \Pi'')$ is the same set as G.
- **Justification**: Π'' promotes *progress*, i.e., $\texttt{progress}(\Pi_i, \Pi'') > \texttt{progress}(\Pi_i, \Pi')$

NE1 and NE2 denote the presumptive existence of an alternative plan that achieves the same G but promoting any of the two values. Note that the two evaluations can be regarded as a direct combination of CQ5 and CQ6 in (Atkinson et al., 2006), and they are indirectly related to CQ8 and CQ9 because, if the presumptive justification is right then it means Π' demotes the `cost` or the `progress`.

Negative Evaluation 3: alternative way of promoting V (NE3)

- **Attack**: There is another refinement step Π'' such that $\Pi_i \circ \Pi''$ is a valid refinement of Π_i, where G' is the set of goals solved by $\Pi_i \circ \Pi''$, i.e., $G' = \texttt{open\_cond}(\Pi_i) \setminus \texttt{open\_cond}(\Pi_i \circ \Pi'')$, and G and G' are two different sets of goals.
- **Justification**: Π'' promotes *cost*, i.e., $\texttt{cost}(\Pi_i \circ \Pi'') < \texttt{cost}(\Pi_i \circ \Pi')$

Negative Evaluation 4: alternative way of promoting V (NE4)

- **Attack**: There is another refinement step Π'' such that $\Pi_i \circ \Pi''$ is a valid refinement of Π_i, where G' is the set of goals solved by $\Pi_i \circ \Pi''$, i.e., $G' = \texttt{open\_cond}(\Pi_i) \setminus \texttt{open\_cond}(\Pi_i \circ \Pi'')$, and G and G' are two different sets of goals.
- **Justification**: Π'' promotes *progress*, i.e., $\texttt{progress}(\Pi_i, \Pi'') > \texttt{progress}(\Pi_i, \Pi')$

NE3 and NE4 can be indirectly regarded as a combination of CQ7-CQ8-CQ9. However, none of the critical questions in (Atkinson et al., 2006) fit our alternative ways of promoting V. Observe that, the concrete critical question here should be "*Is there any alternative way of proceeding towards a solution?*" and, we might add, "*rather than solving G*". This critical question deserves a special consideration, when solving a CDP task, for two reasons:

1. In planning, it is necessary not only to handle the choice of action but also the choice of goal because the order in which goals are selected for solving is decisive in the completeness and soundness of the resolution process.
2. Solving a CDP task involves an incremental construction of a solution. The agents' opportunism to recognize and respond to potential partial plans will depend on the composite plan, and, more particularly, on when this plan appears, which is dependent on the choice of goal.

Finally, we define the negative evaluation concerned with the conflicts between plans.

Negative Evaluation 5: conflicts in the promotion of `progress` (NE5)

- **Attack**: There is no refinement step Π' such that $\Pi_i \circ \Pi'$ is a valid refinement of Π_i
- **Justification**: Given S, a non-empty set of open preconditions of Π_i ($S \subseteq \texttt{open\_cond}(\Pi_i)$, there is no refinement step that achieves S, thus not promoting the progress of Π_i with respect to the set S.

NE5 states the impossibility of an agent to solve a particular combination of open conditions, S, in Π_i. In contrast with the impossibility for the agent to achieve a particular goal due to its own capabilities, the articulation of NE5 is exclusively referred to the plan Π_i. In consequence, we need to distinguish between these two incompatibilities of preventing Π_i from progressing towards a solution plan. If an agent is not endowed with the appropriate capabilities for achieving a goal, it will never be able to achieve such a goal, no matter what the planning argument under discussion is. However, an agent might dispose of the necessary resources and capabilities to achieve the goal and not be capable to successfully attain it because of the information comprised in Π_i. If we assume that NE5 is always concerned with the structural composition of Π_i, then the absence of NE5 could mean a permanent incapacity of the agents to achieve a goal or viceversa.

One way to make such a distinction is by including ***positive evaluations*** to an argument in our framework. A positive evaluation is made by an agent when it can solve a set, S, of open conditions of Π_i, thus promoting its progress:

Positive Evaluation 1: no conflicts in the promotion of *progress* (PE1)

- **Attack**: There is a refinement step Π' such that $\Pi_i \circ \Pi'$ is a valid refinement of Π_i
- **Justification**: Given S, a non-empty set S of open preconditions of Π_i ($S \subseteq \texttt{open\_cond}(\Pi_i)$), there is a refinement step Π' such that $S \subseteq \texttt{open\_cond}(\Pi_i) \setminus \texttt{open\_cond}(\Pi_i \circ \Pi') \wedge |\texttt{threats}(\Pi_i \circ \Pi')| = 0$, thus promoting the progress of Π_i, i.e., $\exists \Pi' / \texttt{progress}(\Pi_i, \Pi') > 0$ with respect to the set S.

4 CDP protocol

In this section, we present a basic interaction protocol for solving a CDP task. The goal of this protocol is to show how the proposed argumentation model can be used to cooperatively solve the problem. We assume that agents are ordered according to their indexes: 1, 2, ..., n. This protocol follows a rotating shift approach in which an agent can only participate during its turn. Initially, the turn is on the first agent ag_1, and the current base plan, Π_i, is the empty plan Π_0, which only contains the two fictitious actions a_0 and a_f. The protocol is divided into the following stages:

1. **Proposal of a partial plan argument** $PPA = \langle \Pi_i, \Pi' \rangle$. On the agent's turn, it can make a proposal on the current base plan if it knows a valid refinement $\Pi_j = \Pi_i \circ \Pi'$ to Π_i. If the agent cannot contribute to the joint plan under construction, then it passes the turn to the following agent. After a complete round without any proposal, a backtracking step is carried out.

2. **Backtracking step**. This step occurs when the current base plan cannot make progress towards a plan solution. Then, Π_i is rejected, and the current base plan is replaced by the corresponding base plan of Π_i. If the initial base plan Π_0 is rejected, then there is no solution for the CDP task. Otherwise, the protocol continues with the following stage.

3. **Argument evaluation**. The current proposal $\langle \Pi_i, \Pi' \rangle$ is evaluated at this stage. On the agents' turn, they can present alternative ways to refine Π_i (NE1-NE4). They can also attack (NE5) and/or defend (PE1) the current proposal, Π', or any of the alternative refinement steps presented. This stage ends after a complete round without any evaluation. After this stage, we have a set of refinement steps for the current base plan: Π' and the set of proposed alternatives. Then, agents analyze the negative and positive evaluations for each refinement step to figure out which refinement steps are unfeasible. A refinement step Π'' is considered unfeasible when $\Pi_i \circ \Pi''$ has open preconditions that no agent can achieve, which can be deduced from the NE5 and PE1 evaluations. If there is no feasible refinement step for Π_i, then a backtracking step is carried out.

4. **Voting**. At this stage, agents vote for the feasible refinement step they consider to be more advisable according to their own preferences and beliefs. The next current base plan will be Π_i concatenated with the refinement step with the highest number of votes and, in case of tie-breaking, the agent who has the turn will make the final decision. If the new current base plan Π is a solution plan for the CDP task, then the process ends with successfully. Otherwise, the process continues with a new PPA proposal (first stage).

The following section illustrates how this protocol is applied to an example scenario to obtain a solution to a CDP task.

5 Example of application

Figure 1 shows the planning scenario where we will put our argumentation-based model to work. This example is inspired by the one proposed in (Parsons, Sierra and Jennings, 1998).

The planning task is to hang the two pictures, p_1 and p_2, within 50 time units, hence, the problem has two top-level goals: g_1 = hung(p_1, l_1) and g_2 = hung(p_2, l_2). The utility function to minimize is the plan duration, so the cost of an action corresponds to its duration in this example.

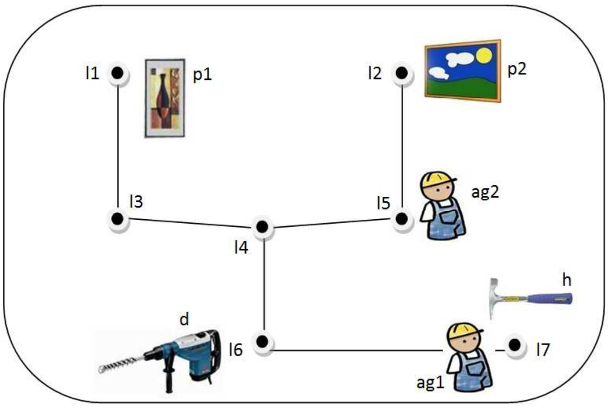

Figure 1: Scenario of the application example

There are two agents to undertake this mission: $AG = \{ag_1, ag_2\}$. Agent ag_1 can hang the pictures by making use of one of two resources, the hand drill d or the hammer h. Agent ag_2 is only able to use the hammer to hang a picture. There are seven different locations in this scenario $l_1, \ldots, l_7$. Agents can walk from one location to an adjacent one. Adjacent locations are linked with a line in Figure 1. Walk actions take 10 time units whereas the duration of the rest of the actions is one time unit.
The actions the agents can perform are the following ones:

- walk(ag_i, l_j, l_k): agent ag_i walks from location l_j to l_k, which must be adjacent locations.
- pickup(ag_i, r_j, l_k): agent ag_i picks up a resource, which can be the hammer or the hand drill, in l_k. Both the agent and the resource must be in l_k.
- pass(ag_i, ag_j, l_n, r_k): agent ag_i passes the resource r_k, which can be the hammer or the hand drill, to ag_j in location l_n. Both agents must be in l_n, and ag_i must have the resource.
- hang(ag_i, p_j, l_k, r_n): agent ag_i hangs the picture p_j with the resource r_n in location l_k.

The argumentation process starts with an empty base plan Π_0. Agent ag_1 proposes the first partial plan argument:

Arg$_1$ [ag_1]: ag_1 proposes $\langle \Pi_0, \Pi_{0.1} \rangle$, where $\Pi_{0.1}$ is a partial plan to hang $p1$ with the hand drill.

$\Pi_{0.1}$ = $\langle$walk(ag_1, l_7, l_6), pickup(ag_1, d, l_6), walk(ag_1, l_6, l_4), walk(ag_1, l_4, l_3), walk(ag_1, l_3, l_1), hang(ag_1, p_1, l_1, d)$\rangle$,

which promotes the value of $\texttt{progress}(\Pi_0, \Pi_{0.1})$ because $\Pi_0 \circ \Pi_{0.1}$ realizes the open goal g_1 of Π_0.

Then, agent ag_2 articulates its evaluations to Arg$_1$:

Ev1. [ag_2] PE1: ag_2 has a valid plan refinement $\Pi_{0.1.1}$ to solve g_2, thus promoting the value of $\texttt{progress}(\Pi_0 \circ \Pi_{0.1}, \Pi_{0.1.1})$

$\Pi_{0.1.1} = \langle$pass(ag_1,ag_2,h,l_5),walk(ag_2, l_5, l_2), hang(ag_2, p_2, l_2, h)$\rangle$

Ev2. [ag_2] NE3: ag_2 proposes $\Pi_{0.2}$ as an alternative plan to $\Pi_{0.1}$ because it solves g_2 and $\texttt{cost}(\Pi_0 \circ \Pi_{0.2}) < \texttt{cost}(\Pi_0 \circ \Pi_{0.1})$

$\Pi_{0.2} = \langle$pass(ag_1,ag_2,h,l_5),walk(ag_2, l_5, l_2), hang(ag_2, p_2, l_2, h)$\rangle$

With evaluation Ev1, ag_2 states that it can hang picture p_2, i.e., the top-level goal g_2. However, as can be observed in plan $\Pi_{0.1.1}$, ag_2 needs to receive the hammer from ag_1 in location l_5 because going to l_7 to pick up the hammer, and then hang p_2, takes 72 time units, which exceeds the imposed time limit. This means that the effect $has(ag_2, h, l_5)$ of the action pass(ag_1,ag_2,h,l_5) in plan $\Pi_{0.1.1}$ depends on the previous achievement of the open condition $has(ag_1, h, l_5)$.

The second evaluation, Ev2, argues that it is preferable to hang p_2 first because ag_2 believes it is preferable to start with a less costly plan. Note that plan $\Pi_{0.2}$ also has the same open condition as $\Pi_{0.1.1}$ ($has(ag_1, h, l_5)$) because, in this proposal, ag_2 expects likewise that ag_1 passes it the hammer at location l_5.

After these two evaluations, ag_2 passes the turn to ag_1:

Ev3. [ag_1] PE1: ag_1 has a valid plan refinement $\Pi_{0.2.1}$, which promotes the progress of $\Pi_0 \circ \Pi_{0.2}$ because it solves the open condition $has(ag_1, h, l_5)$, and, consequently, it achieves the fact $has(ag_2, h, l_5)$

$\Pi_{0.2.1} = \langle$pickup(ag_1, h, l_7), walk(ag_1, l_7, l_6), walk(ag_1, l_6, l_4), walk(ag_1, l_4, l_5), pass(ag_1, ag_2, l_5, h)$\rangle$.

Ev4. [ag_1] NE5: ag_1 cannot refine $\Pi_0 \circ \Pi_{0.2}$ to solve both $has(ag_1, h, l_5)$ and g_1; that is, there is no refinement step Π' that solves $S = \{has(ag_1, h, l_5), g_1\}$; so every attempt to solve S does not promote the progress of $\Pi_0 \circ \Pi_{0.2}$ with respect to the set S.

Agent ag_1 evaluates the alternative plan $\Pi_{0.2}$ proposed by ag_2 in Ev2. As can be observed, the plan $\Pi_0 \circ \Pi_{0.2}$ has two open conditions: $has(ag_1, h, l_5)$ and g_1, which is the pending top-level goal in this composite plan. The positive evaluation Ev3 shows that ag_1 can pass the hammer to ag_2, thus solving $has(ag_1, h, l_5)$. However, in Ev4, ag_1 argues that it cannot solve both open conditions within the time limit; i.e., going to l_5 to pass the hammer to ag_2, and hanging the picture p_1 takes 64 time units. The overall argumentative process about Arg_1 is depicted on the left side of Figure 2.

Next, agent ag_1 passes the turn to ag_2:

Ev5. [ag_2] NE5: ag_2 cannot refine $\Pi_0 \circ \Pi_{0.2}$ because it cannot solve either the open condition $has(ag_1, h, l_5)$ or the top-level goal g_1, so it cannot promote the progress of $\Pi_0 \circ \Pi_{0.2}$ towards a solution plan.

Now, it is clear that $\Pi_0 \circ \Pi_{0.2}$ is unfeasible because there is no way for agent ag_2 to have the hammer at l_5 to hang the picture p_2 at l_2, and have the picture p_1 hung at l_1 within the deadline. In other words, it is not possible to satisfy both g_1 and g_2 via plan $\Pi_0 \circ \Pi_{0.2}$ because the open

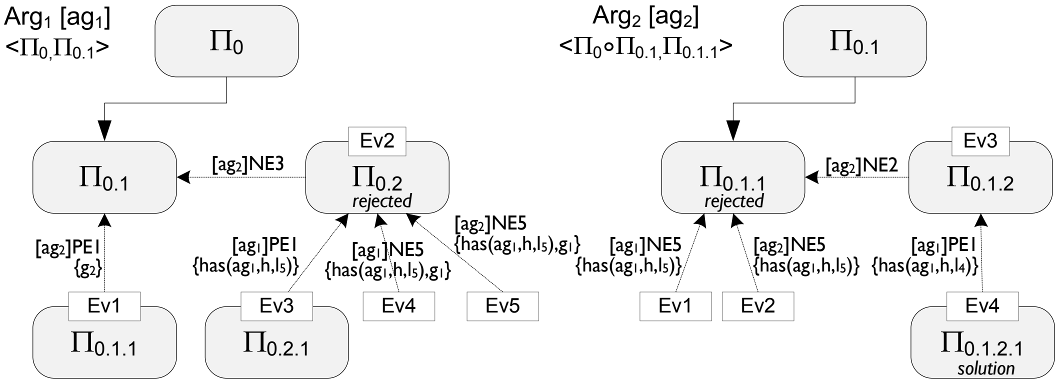

Figure 2: Evaluations of the argumentation process carried out by the agents

conditions in $\Pi_0 \circ \Pi_{0.2}$ cannot be solved within the deadline by either of the two agents, or any cooperation between them. This conclusion is reported through evaluations Ev4 and Ev5 so both agents are aware of the impracticability of this refinement.

Since the agents do not make more evaluations, they vote for the following base plan. The only feasible refinement is $\Pi_0 \circ \Pi_{0.1}$, so it becomes the new base plan. Agent ag_2 then proposes a partial plan argument for this new base plan:

Arg$_2$ $[ag_2]$: ag_2 proposes $\langle \Pi_0 \circ \Pi_{0.1}, \Pi_{0.1.1} \rangle$, where $\Pi_{0.1.1}$ is a partial plan to hang the picture $p2$ with the hammer.

$\Pi_{0.1.1} = \langle$pass(ag_1,ag_2,h,l_5),walk(ag_2, l_5, l_2), hang(ag_2, p_2, l_2, h)$\rangle$,

which promotes the value of `progress`$(\Pi_0 \circ \Pi_{0.1}, \Pi_{0.1.1})$ because it achieves the open condition g_2 of $\Pi_0 \circ \Pi_{0.1}$.

Then, agent ag_1 starts the evaluation of Arg$_2$, which can be seen on the right side of Figure 2:

Ev1. $[ag_1]$ NE5: ag_1 cannot refine $\Pi_0 \circ \Pi_{0.1} \circ \Pi_{0.1.1}$ to solve $has(ag_1, h, l_5)$, so there is no valid refinement step Π' that increases the value of `progress`$(\Pi_0 \circ \Pi_{0.1} \circ \Pi_{0.1.1}, \Pi')$

Agent ag_1 argues that it cannot pass the hammer to ag_2 in l_5 because approaching to l_5 will prevent ag_1 from hanging p_1 within the specified deadline. Following, agent ag_2 confirms that $\Pi_0 \circ \Pi_{0.1} \circ \Pi_{0.1.1}$ is unfeasible and proposes an alternative refinement step for $\Pi_0 \circ \Pi_{0.1}$:

Ev2. $[ag_2]$ NE5: ag_2 cannot refine $\Pi_0 \circ \Pi_{0.1} \circ \Pi_{0.1.1}$ to solve the open condition $has(ag_1, h, l_5)$, thus not promoting the progress of this plan
Ev3. $[ag_2]$ NE2: ag_2 proposes a new plan, $\Pi_{0.1.2}$, that also solves g_2 and `progress`$(\Pi_0 \circ \Pi_{0.1}, \Pi_{0.1.2}) >$ `progress`$(\Pi_0 \circ \Pi_{0.1}, \Pi_{0.1.1})$

$\Pi_{0.1.2} = \langle$walk(ag_2, l_5, l_4), pass(ag_1,ag_2,h,l_4), walk(ag_2, l_4, l_5), walk(ag_2, l_5, l_2), hang(ag_2, p_2, l_2, h)$\rangle$

Agent ag_2 argues that plan $\Pi_{0.1.2}$ is a less costly way for ag_1 to pass it the hammer. Through an examination of plan $\Pi_{0.1}$, ag_2 knows that ag_1 will go through location l_4 to hang the picture p_1,

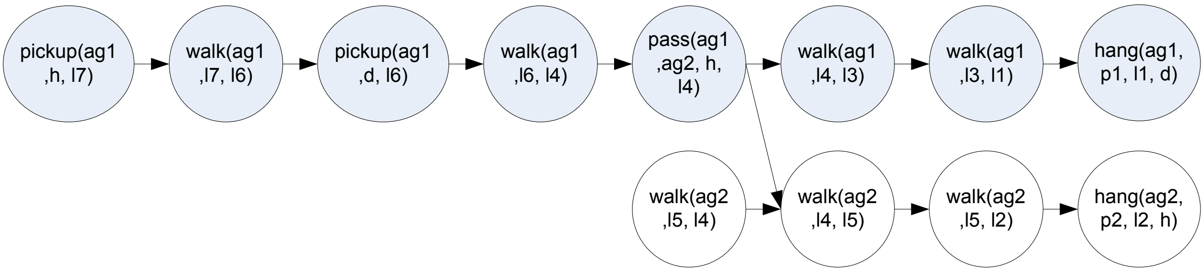

Figure 3: The obtained solution plan, Π_{sol}, for the proposed problem example. Dark ellipses and white ellipses are actions executed by ag_1 and ag_2, respectively.

so it proposes a plan in which ag_2 moves to l_4 so that ag_1 can pass it the hammer. In particular, the plan $\Pi_{0.1.2}$ attains the fact $at(ag_2, l_4)$ so the two agents coincide at the same location, thus solving two of the preconditions of the action pass(ag_1,ag_2,h,l_4), namely $at(ag_1, l_4)$ and $at(ag_2, l_4)$. In contrast, the plan $\Pi_{0.1.1}$ *expects* ag_1 to move to l_5 to pass the hammer to ag_2, so the precondition $at(ag_1, l_5)$ is still an open condition that has to be satisfied for the action pass(ag_1,ag_2,h,l_5). Therefore, $\Pi_{0.1.2}$ leaves fewer open conditions than $\Pi_{0.1.1}$ and, therefore, the progress of $\Pi_{0.1.2}$ to the base plan is greater than the progress of $\Pi_{0.1.1}$.
Over this proposal, ag_1 proposes a positive evaluation to solve the pending open condition $has(ag_1, h, l_4)$, for that ag_1 can pass the hammer to ag_2 at l_4. The last refinement step proposed by ag_1 only comprises the action pickup(ag_1, h, l_7).

Ev4. [ag_1] PE1: ag_1 has a valid plan refinement $\Pi_{0.1.2.1}$ to solve $has(ag_1, h, l_4)$, which promotes the value of `progress`($\Pi_0 \circ \Pi_{0.1} \circ \Pi_{0.1.2}, \Pi_{0.1.2.1}$)

The resulting plan $\Pi_{sol} = \Pi_0 \circ \Pi_{0.1} \circ \Pi_{0.1.2} \circ \Pi_{0.1.2.1}$ is shown in Figure 3. This plan achieves all top-level goals and does not have open conditions, so it is a solution plan for the planning task. The cost of this plan is the duration of the longest sequence of actions, that is, 44 time units. At this point, agents can finish the argumentation process or they can continue sending new partial plan arguments to improve the cost of the current solution plan.

6 Extending CDP to non-cooperative agents

Our CDP organization only considers cooperative agents and deliberatively ignores the existence of self-interested agents with private interests. In this paper, our main purpose was to present an initial approach for an argumentation model that is specifically devoted to solving planning tasks that are better solved through cooperation. However, our CDP organization can be easily extended to a scenario with non-cooperative agents. Typically, in distributed planning, agents do not have a complete view of the world but, rather a *partial visibility*, so not all information is accessible to all agents. Additionally, when dealing with self-interested agents, the existence of *private goals* (and perhaps conflicting goals) is commonly assumed. The partial visibility encourages cooperation provided that agents cannot achieve their own goals by themselves; *privacy* endangers cooperation as private goals may prevail over global ones. Following, we outline some ideas about how to extend the CDP organization to take into account these issues:

- Partial visibility or local knowledge of the world is easily represented by limiting the initial knowledge $\mathcal{I}$ and the set of abilities Ω of each agent. Under this new perspective, since agents do not have a complete knowledge of the world, solving a CDP task will require cooperation because, otherwise, it could not be solved. At this point, it is also important to determine the level of communication among agents and which knowledge/data agents will exchange with each other, while keeping privacy. This is a key aspect in CDP organizations because agents must somehow decide the information they are willing to share in order to promote cooperation. In this respect, it is necessary for agents to share a common ontology in order for them to know which information can be exported to/imported from others and thus how they can help each other.
- Besides the global goals of a CDP task, non-cooperative agents can also have their private goals which may conflict with others' private goals, or even with the global goals of the CDP task. Consequently, agents' proposals of refinement steps are not always meant to satisfy a CDP goal but rather their own private goals, or both. If privacy prevails over cooperation, then agents will be motivated to perform actions that threaten one or more causal links of the global plan. The acceptance or rejection of these *selfish* proposals will depend on the willingness of other agents to accept them, the level of trust and reputation of the proposer, and the level of complexity that the proposal creates in the global plan.

7 Conclusions and future work

In this paper, we have presented an approach to practical reasoning in a planning through an argumentation-based model. This proposal is particularly aimed at solving a cooperative distributed planning problem where the collective behaviour and cooperation among the agents play a crucial role. Under this planning view, agents are designed to work together to construct a joint plan that achieves the goals of a planning task. For this purpose, agents argue plan refinements and reach an agreement on the presumptively best plan composition.

The argumentation-based model is designed in terms of argument schemes and critical questions, whose interpretation is given through the semantic structure of a partial-order planning paradigm. The flexibility of the POP semantic model facilitates reasoning about the components of the individual plans, and encapsulate these reasonings in negative or positive evaluations to a plan argument. Evaluations are aimed at promoting a plan improvement, either in terms of cost or progress, a parameter that measures the agents' contributions to refine the plan.

We can enumerate several advantages of our argumentation-based model with respect to similar approaches: (1) the instantiation of the argumentation scheme to a set of elements instead of to a single action, goal or value; (2) the specific consideration of choice of goal as a means to forward the discussion to, presumptively, more promising argumentation lines; and

(3) a sophisticated evaluation of conflicting situations focused to facilitating the future agents' contributions. The ultimate objective of these contributions is to promote cooperation for a collective resolution of a planning problem.
As a whole, we can conclude that, to the best of our knowledge, this is the first attempt of solving a distributed planning problem through argumentative cooperation. The presented model is particularly suitable for problems that involve an incremental construction of a solution as, for example, a sequence of successive refinement steps over a given plan.

References

Amgoud, L. 2003. A formal framework for handling conflicting desires, *ECSQARU*, pp. 552–563.

Atkinson, K. and Bench-Capon, T. 2007. Practical reasoning as presumptive argumentation using action based alternating transition systems, *Artificial Intelligence* **171**: 855–874.

Atkinson, K., Bench-Capon, T. J. M. and McBurney, P. 2006. Computational representation of practical argument, *Synthese* **152**(2): 157–206.

Barrett, A. and Weld, D. S. 1994. Partial-order planning: Evaluating possible efficiency gains, *Artificial Intelligence* **67**(1): 71–112.

Bench-Capon, T. and Prakken, H. 2006. Choosing what to do by accruing arguments, *Computational Models of Argument. Proceedings of COMMA 2006*, Vol. 144, Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 247–258.

Cox, J., Durfee, E. and Bartold, T. 2005. A distributed framework for solving the multiagent plan coordination problem, *AAMAS'05*, pp. 821–827.

de Weerdt, M., ter Mors, A. and Witteveen, C. 2005. Multi-agent planning. an introduction to planning and coordination, *Handouts of the European Agent Systems Summer School (EASSS-05)*, pp. 1–32.

desJardins, M., Durfee, E., Ortiz, C. and Wolverton, M. 1999. A survey of research in distributed continual planning, *AI Magazine* **20**(4): 13–22.

desJardins, M. and Wolverton, M. 1999. Coordinating a distributed planning system, *AI Magazine* **20**(4): 45–53.

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, *Artificial Intelligence* **77**(2): 321–358.

Durfee, E. H. 1999. *Distributed problem solving and planning*, Vol. In Gerhard Weiss editor, The MIT PRess, San Francisco, CA, pp. 118–149.

Durfee, E. H. 2001. Distributed problem solving and planning, *Multi-agents Systems and Applications (EASSS 2001)*, Vol. LNAI 2086, Springer-Verlag, pp. 118–149.

Fikes, R. and Nilsson, N. 1971. STRIPS: A new approach to the application of theorem proving to problem solving, *Artificial Intelligence* **2**(3): 189–208.

Gerevini, A. and Schubert, L. K. 1996. Accelerating partial-order planners: Some techniques for effective search control and pruning, *J. Artif. Intell. Res. (JAIR)* **5**: 95–137.

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

Hulstijn, J. and van der Torre, L. 2004. Combinining goal generation and planning in an argumentation framework, *Proc. Workshop on Argument, Dialogue and Decision at the International Workshop on Non-monotonic Reasoning (NMR'04)*.

Lesser, V., Decker, K., Wagner, T., Carver, N., Garvey, A., Horling, B., Neiman, D., Podorozhny, R., NagendraPrasad, M., Raja, A., Vincent, R., Xuan, P. and Zhang, X. 2004. Evolution of the GPGP/TAEMS domain-independent coordination framework, *Autonomous Agents and Multi-Agent Systems* **9**(1): 87–143.

McBurney, P., Hitchcock, D. and Parsons, S. 2007. The eightfold way of deliberation dialogue, *Int. J. Intell. Syst.* **22**(1): 95–132.

Parsons, S., Sierra, C. and Jennings, N. R. 1998. Agents that reason and negotiate by arguing, *Journal of Logic Computation* **8**(3): 261–292.

Penberthy, J. and Weld, D. 1992. UCPOP: A sound, complete, partial order planner for ADL, *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning* pp. 103–114.

Rahwan, I. and Amgoud, L. 2006. An argumentation-based approach for practical reasoning, *ArgMAS*, pp. 74–90.

Shehory, O. and Kraus, S. 1998. Methods for task allocation via agent coalition formation, *Artificial Intelligence* **101**(1-2): 165–200.

Simari, G. R., García, A. J. and Capobianco, M. 2004. Actions, planning and defeasible reasoning, *10th International Workshop on Non-Monotonic Reasoning (NMR) 2004)*, pp. 377–384.

van der Hoek, W., Roberts, M. and Wooldridge, M. 2007. Social laws in alternating time: effectiveness, feasibility, and synthesis, *Synthese* **156**(1): 1–19.

Walsh, W. E. and Wellman, M. P. 1999. Efficiency and equilibrium in task allocation economies with hierarchical dependencies, *IJCAI*, pp. 520–526.

Walton, D. 1996. *Argumentation Schemes for Presumptive Reasoning*, Lawrence Erlbaum Associates, Mahwah, NJ.