Multiagent Argumentation for Cooperative Planning in DeLP-POP

Pere Pardo
IIIA - CSIC
Campus UAB, s/n
08193 Bellaterra, Spain
pardo@iiia.csic.es

Sergio Pajares
Univ. Politècnica de València
Camino de Vera, s/n
46022 Valencia, Spain
spajares@upv.dsic.es

Eva Onaindia
Univ. Politècnica de València
Camino de Vera, s/n
46022 Valencia, Spain
onaindia@upv.dsic.es

Lluís Godo
IIIA - CSIC
Campus UAB, s/n
08193 Bellaterra, Spain
godo@iiia.csic.es

Pilar Dellunde
IIIA - CSIC and Univ.
Autònoma de Barcelona
08193 Bellaterra, Spain
pilar@iiia.csic.es

ABSTRACT

This contribution proposes a model for argumentation-based multi-agent planning, with a focus on cooperative scenarios. It consists in a multi-agent extension of DeLP-POP, partial order planning on top of argumentation-based defeasible logic programming. In DeLP-POP, actions and arguments (combinations of rules and facts) may be used to enforce some goal, if their conditions (are known to) apply and arguments are not defeated by other arguments applying. In a cooperative planning problem a team of agents share a set of goals but have diverse abilities and beliefs. In order to plan for these goals, agents start a stepwise dialogue consisting of exchanges of plan proposals, plus arguments against them. Since these dialogues instantiate an A^* search algorithm, these agents will find a solution if some solution exists, and moreover, it will be provably optimal (according to their knowledge).

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Algorithms, Theory

Keywords

Argumentation, Multiagent Planning, Cooperation

1. INTRODUCTION

The present contribution proposes a formal model of argumentative dialogues for multi-agent planning, with a focus on cooperative planning. It consists in a multi-agent extension of the DeLP-POP framework in [5], where it is shown how to adapt partial order planning (POP) to model a planner agent able to reason defeasibly (DeLP). This framework combines POP's minimal constraints on execution ordering (see [9]), with DeLP inference based on interactions between arguments (see [4]). A DeLP-POP planner can enforce goals with a combination of actions and undefeated arguments, if their conditions (are known to) apply[1]. Arguments, though, are not like actions in that they apply even if unintended. Thus, arguments will not only occur to intentionally support some step of a plan, but also they will happen to defeat or defend some such supporting argument and the plan containing it.

The main challenge presented by cooperative multi-agent DeLP-POP is plan evaluation and search. We present some results[2] about dialogues for argumentative plan search that apply to cooperative scenarios. In these scenarios, we have a team of agents aware of a common set of goals (hence trustable), but ignorant of others' abilities and beliefs, who must find a plan. An obvious solution, centralized planning carried by some planner with knowledge of these agents' beliefs and actions, would arise questions of efficiency and privacy loss (beyond necessity). Instead we will use centralized DeLP-POP just for comparison with dialogues proposed. A dialogue consists in a series of exchanges[3] of (1) plan proposals addressing the current goal, plus (2) potential arguments against (1). Atomic information (facts, rules, actions) contained in others' messages (1) and (2) will be extracted and adopted to devise new ideas for both (1) and (2).

The main result of this contribution is that such a dialoguing team of planner agents actually implements an A^*

[1]The advantages of DeLP-POP towards reasoning about actions are clear: if planning techniques prevent the well-known frame problem, by getting rid of the need to explicitly represent what does not change after an action, DeLP-POP succeeds against the qualification problem as well, since DeLP-rules can be used to encode defeasible effects of actions, as shown in Section 2.2.

[2]The proofs of these formal results can be found in http : //www.iiia.csic.es/files/pdfs/AAMAS11ppogd.pdf.

[3]Dialogues are turn-based, since this choice models typically cooperative scenarios where all agents are treated in a uniform way, but also can (by adding some restrictions) model agents with power to veto information or decisions.

Cite as: Multiagent Argumentation for Cooperative Planning in DeLP-POP, Pere Pardo, Sergio Pajares, Eva Onaindia, Lluís Godo and Pilar Dellunde, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 971-978.

search procedure. Thus, the team of agents need not search the full space of plans: the dialogue terminates at a solution (if some solution exists) which is provably optimal.

2. PRELIMINARIES

Notation: Throughout the paper we make use of these conventions: the projection functions are $\pi_k(\langle a_0, \ldots, a_n\rangle) = = a_k$ (for $k \leq n$), and $\pi_{\widehat{k}}(\langle a_0, \ldots, a_{k-1}, a_k, a_{k+1}, \ldots, a_n\rangle) = = \langle a_0, \ldots, a_{k-1}, a_{k+1}, \ldots a_n\rangle$. Given propositional variables $p, \ldots \in \mathsf{Var}$, and a negation $\sim$, we define the set of literals $\ell \in \mathsf{Lit} = \mathsf{Var} \cup \{\sim p \mid p \in \mathsf{Var}\}$. Also, define $\overline{\ell}$ as $\overline{p} = \sim p$, and $\overline{\sim p} = p$, for any $p \in \mathsf{Var}$; and for $X \subseteq \mathsf{Lit}$, $\overline{X} = \{\overline{\ell} \mid \ell \in X\}$. In general, if $F : X \to Y$ is a function and $X' \subseteq X$, we denote $F[X'] = \{f(x) \mid x \in X'\}$. The transitive closure of a relation R is denoted $\mathsf{tc}(R)$. The size of a set X is denoted $|X|$. If X is a set, $\mathcal{P}(X)$ denotes its power set, and $X\binom{\sigma\tau\ldots}{\sigma'\tau'\ldots}$ denotes the set obtained by replacing σ by σ', τ by τ', ... in set X.

2.1 Defeasible Logic DeLP

In [4], the authors propose a non-monotonic consequence relation, called *warrant*, built upon the relation of defeat between constructible arguments for or against a literal. A defeasible logic program (or *de.l.p.*, henceforth) is a pair $T = (\Psi, \Delta)$ consisting of a strict and a defeasible part:

- a consistent set $\Psi \subseteq \mathsf{Lit}$ of *facts*, and
- a set Δ of defeasible *rules* $\delta = \ell \prec\!\!- \ell_0, \ldots, \ell_k$

where $\ell, \ell_0, \ldots, \ell_k \subseteq \mathsf{Lit}$. Rule $\ell \prec\!\!- \ell_0, \ldots, \ell_k$ expresses: warrant for $\ell_0, \ldots, \ell_n$ provide a (defeasible) reason for ℓ to be warranted[4]. We denote $\mathsf{body}(\delta) = \{\ell_0, \ldots, \ell_n\}$ and $\mathsf{head}(\delta) = \ell$ as, respectively, the *body* and *head* of δ.

Derivability in $T = (\Psi, \Delta)$ is closure under *modus ponens*: literals in Ψ are derivable and, given a rule δ, if each $\ell \in \mathsf{body}(\delta)$ is derivable, then $\mathsf{head}(\delta)$ is derivable.
An *argument* $\mathcal{A}$ for ℓ in a de.l.p. (Ψ, Δ), denoted $\langle \mathcal{A}, \ell\rangle$ or simply $\mathcal{A}$, is a set of rules $\mathcal{A} \subseteq \Delta$ such that (i) ℓ is derivable from $(\Psi, \mathcal{A})$, (ii) the set $\Psi \cup \mathcal{A}$ is non-contradictory, and (iii) $\mathcal{A}$ is a minimal subset of Δ satisfying (i) and (ii).

We also define, for an argument $\mathcal{A}$ for ℓ

$$\begin{array}{rcl} \mathsf{concl}(\mathcal{A}) & = & \ell, \\ \mathsf{base}(\mathcal{A}) & = & (\bigcup \mathsf{body}[\mathcal{A}]) \setminus \mathsf{head}[\mathcal{A}], \text{ and} \\ \mathsf{literals}(\mathcal{A}) & = & (\bigcup \mathsf{body}[\mathcal{A}]) \cup \mathsf{head}[\mathcal{A}] \end{array}$$

A derivation of -or argument for- a literal ℓ from (Ψ, Δ), still, does not suffice for its being warranted in (Ψ, Δ). The latter depends on the interaction among arguments, which will grant consistency.

Given two arguments $\mathcal{A}, \mathcal{B}$, we say $\mathcal{A}$ *attacks* $\mathcal{B}$ if the conclusion of $\mathcal{A}$ contradicts some fact used in $\mathcal{B}$, that is, if $\overline{\mathsf{concl}(\mathcal{A})} \in \mathsf{literals}(\mathcal{B})$. This attack relation may roughly be seen as symmetric, in the sense that each attacked argument $\mathcal{B}$ contains a sub-argument $\mathcal{B}'$ attacking $\mathcal{A}$. (A *sub-argument* of $\mathcal{B}$ is a subset $\mathcal{B}' \subseteq \mathcal{B}$ supporting some inner conclusion ℓ' of $\mathcal{B}$, i.e. with $\ell' \in \mathsf{literals}(\mathcal{B})$.) To decide which contending argument prevails, a notion for preference among pairs of conflicting arguments is needed. The formal criterion for preference here adopted lies in a comparison of information used in each argument: an attacking argument which makes use of more precise rules (or more premises) is a *proper defeater* for -is preferred to- the contending argument. If two contending arguments are not comparable in these terms, they are a *blocking defeater* for each other[5].

Given an argument $\mathcal{A}_0$ for ℓ, an *argumentation line* $\Lambda = [\mathcal{A}_0, \ldots, \mathcal{A}_n]$ in (Ψ, Δ) is a sequence of arguments constructible in (Ψ, Δ), where each argument $\mathcal{A}_{k+1}$ is a defeater for its predecessor $\mathcal{A}_k$. Some further conditions are needed to rule out circular or inconsistent argumentation lines; briefly, arguments supporting (resp. interfering with) $\mathcal{A}_0$, i.e. of the form $\mathcal{A}_{2n}$ (resp. $\mathcal{A}_{2n+1}$) must form a consistent set, and no sub-argument $\mathcal{A}'$ of an argument $\mathcal{A}_m \in \Lambda$ may appear later in Λ (i.e. it cannot be that $\mathcal{A}' = \mathcal{A}_{m'}$ with $m' > m$); see [4] and [5].

Since in a de.l.p. (Ψ, Δ) an argument can have several defeaters, different argumentation lines rooted in $\mathcal{A}_0$ can exist. Their union gives rise to a tree-like structure, the *dialectical tree* for $\mathcal{A}_0$, denoted $\mathcal{T}_{\mathcal{A}_0}(\Psi, \Delta)$. To check whether $\mathcal{A}_0$ is *defeated* or *undefeated*, the following procedure on $\mathcal{T}_{\mathcal{A}_0}(\Psi, \Delta)$ is applied: label with a U (for *undefeated*) each terminal node in the tree (i.e. each argument with no defeaters at all). Then, in a bottom-up fashion, we label a node with:

$$\begin{cases} U & \text{if each of its successors is labeled with a } D \\ D & \text{(for } \textit{defeated}\text{) otherwise} \end{cases}$$

Finally, we say a literal ℓ is *warranted* in (Ψ, Δ), denoted $\ell \in \mathsf{warr}(\Psi, \Delta)$, iff there exists an argument $\mathcal{A}$ in (Ψ, Δ) with $\mathsf{concl}(\mathcal{A}) = \ell$ and $\mathcal{A}$ labeled U in $\mathcal{T}_{\mathcal{A}}(\Psi, \Delta)$. Henceforth, $\mathcal{B}$ *defeats* $\mathcal{A}$ will stand for: $\Lambda = [\ldots, \mathcal{A}, \mathcal{B}, \ldots]$ is acceptable.

2.2 A DeLP extension for POP planning

We briefly recall here state-based and POP planning methods, before introducing DeLP-POP. A planning domain is a tuple $\mathbb{M} = (\Psi, A, G)$ where $\Psi \subseteq \mathsf{Lit}$ represents initial atomic facts, A is a set of actions and $G \subseteq \mathsf{Lit}$ is the set of goals of an agent. Here, an action $\alpha = \langle \mathsf{P}(\alpha), \mathsf{X}(\alpha)\rangle$ is a set of preconditions (for α to be applicable) and effects. A solution is a plan Π leading a Ψ-world into a G-world by means of actions $A_\Pi \subseteq A$.

In state-based planning, a plan Π is a linear sequence of actions, and thus before each action α_k in A_Π, we know which consistent state $\sigma_k \subseteq \mathsf{Lit}$ will hold, with σ_k consistent.

In contrast, a partial order plan (henceforth: plan) Π is a set of actions whose execution ordering $\prec_\Pi$ (i.e. links on action pairs) is only *partially specified* (thus encoding multiple linear plans). In POP, Ψ and G are encoded as dummy actions $\alpha_\Psi \prec_\Pi \alpha_G$ with $\mathsf{X}(\alpha_\Psi) = \Psi$, $\mathsf{P}(\alpha_G) = G$ and $\mathsf{P}(\alpha_\Psi) = \mathsf{X}(\alpha_G) = \emptyset$. Partial orderings give rise to the notion of *threat* in Π: an action step *potentially* interfering with (applicability of) some other action step. The set of all threats to a plan Π will be denoted $\mathsf{AllThreats}(\Pi)$. When detected, threats are to be solved by some threat resolution step. Thus in POP, the set of *flaws* to be solved in a plan Π includes threats and pending goals(initially being $\mathsf{AllThreats}(\Pi) = \emptyset$ and $\mathsf{goals}(\Pi) = \mathsf{P}(\alpha_G)$). The partial order of Π determines, for each $\alpha \in A_\Pi$, a (possibly inconsistent) set of facts *potentially* planned to occur before α (i.e. the threats to this α). This set, called here the proto-state of α (in Π), will be denoted S^Π_α.

An extension of POP with DeLP-style argumentation, denoted DeLP-POP, was introduced in [5]. A DeLP-POP plan-

[4]Strict rules, introduced in [11], [4], have not been considered in planning, see [5].

[5]Or, less abstractly, one could instead specify some particular preference between rules and then induce a defeat relation for arguments out of it. See [11] for details.

ner can appeal both to arguments and actions as a way to resolve goals or threats. The original DeLP or POP notions of argument, planning domain, plan, link and threat must be modified accordingly. An argument $\mathcal{A} \subseteq \Delta$ is consistent if $\mathsf{base}(\mathcal{A}) \cup \mathcal{A}$ *is non-contradictory* (instead of condition (ii) above for $\Psi \cup \mathcal{A}$, since now arguments may apply everywhere, not just at Ψ). DeLP-POP planning domains $\mathbb{M} = (T, A, G)$ contain now a de.l.p. $T = (\Psi, \Delta)$, where the set of initial facts $\Psi \subseteq \mathsf{Lit}$ induces α_Ψ as before and the new element Δ contains defeasible rules that may apply anywhere in the plan. An *action* is a 3-tuple of the form $\alpha = \langle \mathsf{P}(\alpha), \mathsf{C}(\alpha), \mathsf{X}(\alpha) \rangle$, described by, resp., sets of preconditions, constraints and effects. If literals in $\mathsf{P}(\alpha)$ are enforced (or warranted) and those in $\mathsf{C}(\alpha)$ fail to be enforced (or warranted), then action α is *applicable* and its execution will enforce each $\ell \in \mathsf{X}(\alpha)$ (thus deleting $\overline{\ell}$ if holding previously). An argument $\mathcal{A}$ is *applicable* at S^Π_α if $\mathsf{base}(\mathcal{A})$ is enforced in S^Π_α; in this case $\mathsf{concl}(\mathcal{A})$ is derivable[6].

Let ℓ be an open goal, motivated by some step $\beta \in A_\Pi$ or $\mathcal{A} \subseteq \Delta$; i.e. $\ell \in \mathsf{P}(\beta)$ or $\ell \in \mathsf{base}(\mathcal{A})$. If goal ℓ is planned to be enforced by an action α, this is encoded as a *causal link* of Π, in a set denoted by $\mathcal{CL}(\Pi)$: $(\alpha, \ell, \kappa) \in \mathcal{CL}(\Pi) \subseteq A_\Pi \times \mathsf{goals}(\Pi) \times (A_\Pi \cup \mathcal{P}(\Delta))$, with $\kappa = \beta$ or $\kappa = \mathcal{A}$. If goal $\ell \in \mathsf{P}(\beta)$ is to be enforced by an argument, this is encoded as a *support link* of Π, in a set denoted $\mathcal{SL}(\Pi)$: $(\mathcal{B}, \ell, \beta) \in \mathcal{SL}(\Pi) \subseteq \mathcal{P}(\Delta) \times \mathsf{goals}(\Pi) \times A_\Pi$. (Note an argument $\mathcal{B}$ cannot support some other argument $\mathcal{A}$ *as a link* in $\mathcal{SL}(\Pi)$. To get $\mathcal{B}$ to support step $\mathcal{A}$, just replace step $\mathcal{A}$ by $\mathcal{A} \cup \mathcal{B}$.) Additional *ordering constraints* between action steps are encoded simply as $(\alpha, \beta) \in \mathcal{OC}(\Pi) \subseteq A_\Pi \times A_\Pi$. The union of causal links, support links (ignoring their $\mathsf{goals}(\Pi)$ component) and ordering constraints $\mathcal{OC}(\Pi)$ induce, by taking the transitive closure, the *partial order of* Π, i.e. the order between its steps, denoted $\prec_\Pi$:

$$\prec_\Pi = \mathsf{tc}(\mathcal{OC}(\Pi) \cup \pi_{\widehat{1}}(\mathcal{CL}(\Pi)) \cup \pi_{\widehat{1}}(\mathcal{SL}(\Pi)))$$

Now we define a DeLP-POP plan Π for $\mathbb{M} = ((\Psi, \Delta), A, G)$ as a tuple $\Pi = (A_\Pi, \mathsf{goals}(\Pi), \mathcal{OC}(\Pi), \mathcal{CL}(\Pi), \mathcal{SL}(\Pi))$ containing actions to be used $A_\Pi \subseteq A$, current open goals of Π, and links or constraints on the execution ordering.

In DeLP-POP an agent with planning domain $\mathbb{M}$ builds a plan incrementally: she keeps refining it with a new step at a time until a solution (a plan with no unsolved flaws) is found. The algorithm used in [5] is the following: For a given $((\Psi, \Delta), A, G)$, plan search starts with the empty plan $\Pi_\emptyset$, only containing dummy actions $\alpha_\Psi \prec_\Pi \alpha_G$. At each iteration, with current plan $\Pi_\emptyset(\xi_0, \ldots, \xi_k)$, the algorithm nondeterministically selects an unsolved flaw (a threat, preferably) and a refinement step ξ_{k+1} for it (action-, argument- or threat resolution step); after this refinement we obtain plan $\Pi_\emptyset(\xi_0, \ldots, \xi_k, \xi_{k+1})$, and the algorithm updates the set of detected unsolved flaws, so goals and threats are added (if new) or deleted (if solved). If a failure occurs (no refinement is available), the algorithms backtracks to the parent node.

We will denote by $\mathsf{Plans}(\mathbb{M})$ the graph whose nodes are plans for $\mathbb{M}$, related by *is 1-step refinable into*; the set of solution plans will be denoted by $\mathsf{Sol}(\mathsf{Plans}(\mathbb{M}))$.

Threat detection is based on *proto-states*, defined next. For a fixed $\mathbb{M} = ((\Psi, \Delta), A, G)$, a plan Π and $\alpha \in A_\Pi$, S^Π_α denotes the set of literals obtaining before α when we extend $\prec_\Pi$ with some new constraint[7]:

$$S^\Pi_\alpha = \{\ell \in \mathsf{Lit} \mid \exists \alpha' \in A_\Pi \text{ s.t. } \ell \in \mathsf{X}(\alpha') \text{ and } \prec_\Pi \cup \{\langle \alpha', \alpha \rangle\} \text{ is consistent, and } \forall \beta \in A_\Pi, \text{ if } \overline{\ell} \in \mathsf{X}(\beta) \text{ then } \{\langle \alpha', \beta \rangle, \langle \beta, \alpha \rangle\} \not\subseteq \mathsf{tc}(\prec_\Pi \cup \{\langle \alpha', \alpha \rangle\})\}$$

We use proto-state S^Π_α to compute which actions or unintended arguments might be triggered by Π in a way interfering with other steps of Π.

Three kinds of threats must be checked during plan construction in DeLP-POP, see also Figure 1:

(a) action-action: $(\beta, (\alpha_0, \ell, \alpha_1)) \in A_\Pi \times \mathcal{CL}(\Pi)$, s.t. $\overline{\ell} \in \mathsf{X}(\beta)$ and $\prec \Pi \cup \{\langle \alpha_0, \beta \rangle, \langle \beta, \alpha_1 \rangle\}$ is consistent; here β threatens the link between α_0 and α_1,

(b) action-argument: $((\beta, \overline{n}), (\mathcal{B}, b, \alpha_1)) \in (A_\Pi \times \mathsf{Lit}) \times \mathcal{SL}(\Pi)$, with $\overline{\mathsf{X}(\beta)} \cap \mathsf{literals}(\mathcal{B}) \supseteq \{n\}$, where $\prec_\Pi$ makes β to supply $\overline{n} \in S^\Pi_{\alpha_1}$; here β threatens some literal used in $\mathcal{B}$, and

(c) argument-argument: $(\mathcal{C}, (\mathcal{B}, b, \alpha_1)) \in \mathcal{P}(\Delta) \times \mathcal{SL}(\Pi)$, with $\mathcal{C}$ defeating $\mathcal{B}$ and $\mathsf{base}(\mathcal{C}) \subseteq S^\Pi_{\alpha_1}$, $\mathcal{C}$ undefeated in $S^\Pi_{\alpha_1}$.

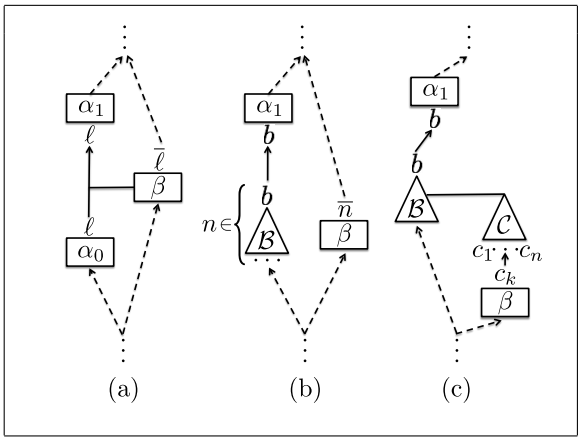

Figure 1: Threat types: (a) action-action, (b) action-argument and (c) argument-argument.

For each kind of threat, different maneuvers, inspired by those in POP, may be tried: moving the cause of the threat to a harmless position (with new ordering constraints; see Figures 2 and 3(c')); or eliminating the threat itself (with a counter-argument[8] or a new action step; see Figures 3(c")-(c"') [9]. We refer the reader to the algorithm in [5] for details.

Finally we describe how to model an action with defeasible effects. Suppose action α has indisputable effects $p_0, p_1, \ldots$ as well as n defeasible effects $d_0, d_1, \ldots$, which are defeated

[6]See [5]'s backward planning algorithm for a full description of an *instance* κ of an action- or argument-steps, or an open goal *in a plan* Π. Each such instance κ is labeled by its full path of links up to some $g \in G$, i.e. $\langle \kappa, \ldots, g \rangle$.

[7]Note that S^Π_α is computed as if α was already applicable. In particular, arguments occurring before α play no role in S^Π_α.

[8]Informally, we might see this threat detection-resolution process as generating a dialectical tree $\mathcal{T}_{(S^\Pi_\alpha, \Delta)}(\mathcal{A}_0)$ for each $(\mathcal{A}_0, \cdot, \alpha) \in \mathcal{SL}(\Pi)$. But now the tree is built w.r.t. varying Π, due to new threat resolution refinements.

[9]Note a new precondition $\overline{p}$ and new link of type $\mathcal{SL}(\Pi)$ or $\mathcal{CL}(\Pi)$ are needed to preserve these maneuvers' effect in future refinements.

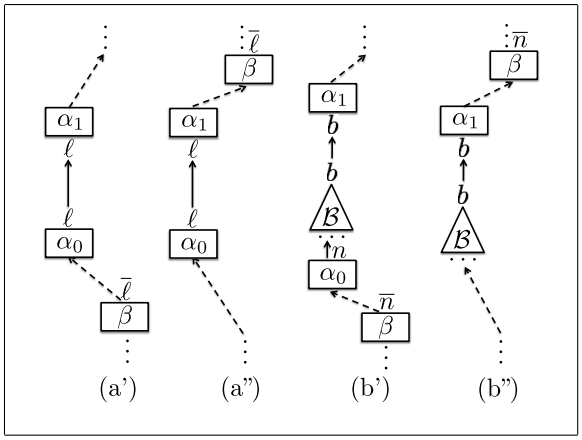

Figure 2: Solutions to (a), (b). Demote: (a'), (b'); and Promote: (a"), (b").

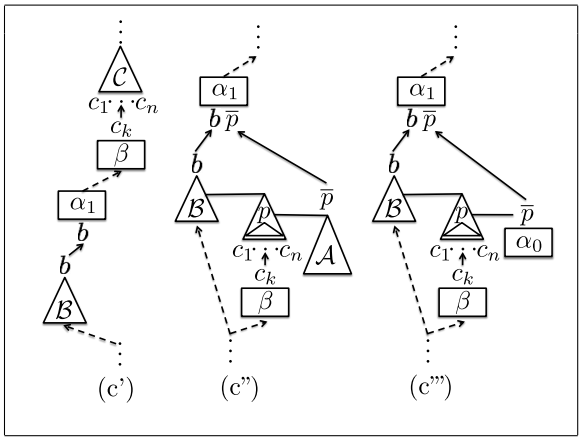

Figure 3: Solutions to (c): Delay (c'), Defeat (c") and Disable (c"').

by conditions $d'_0, d'_1, \ldots$ respectively. At its turn, the latter $d'_0, \ldots$ can be defeated, resp., by $d''_0, \ldots$, and so on. To represent this α, introduce an instrumental irrevocable effect μ' (meaning α *was just executed*); then define $\mathsf{X}(\alpha) = \{p_0, p_1, \ldots, \mu'\}$ and expand the set of rules Δ with $\{d_k \prec \mu'\}_{k<n} \cup \{\overline{d_k} \prec \mu', d'_k\}_{k<n} \cup \ldots\ldots \cup \{d_k \prec \mu', d'_k, d''_k\}_{k<n}$ etc. This way DeLP-POP deals with the qualification problem.

3. ARGUING ON MULTI-AGENT PLANS

The purpose of multi-agent argumentative dialogues is to let agents reach an agreement on (i) the evaluation of plans (Section 4.1); and (ii) adoption of a plan in decentralized plan search (Section 4.2), by allowing agents to refine or revise other agents' plans and defend one's proposals. Before addressing (i) and (ii), though, several modifications of single-agent DeLP-POP are in order.

First, each agent $x \in \mathsf{Ag}$ is initially endowed with a planning domain $\mathbb{M}_x = ((\Psi_x, \Delta_x), A_x, G_x)$. Communication (of facts, rules, actions) from agent x to an agent y will be rendered as an expansion (resp., in Ψ_y, Δ_y, A_y) of $\mathbb{M}_y$.

Second, towards collaborative discovery of potential argument steps or threats and their applicability, agents must send each other known initial facts and pre-arguments; these are like arguments but with partial knowledge of its base, and can be expanded with others' known rules and facts. Given an agent x's plan Π and some $\alpha \in A_\Pi$, we define a *pre-argument* $\mathcal{A}$ as a pair of literals and rules $(X, \mathcal{A})$, where $X \subseteq \mathsf{base}(\mathcal{A})$ are literals known to hold before α, and $\mathsf{base}(\mathcal{A}) \setminus X$ contains literals that may not be known that hold, or how to derive them. We define the set of pre-arguments in a proto-state S^Π_α as $\mathsf{PArgs}(S^\Pi_\alpha, \Delta_x) := \{(X, \mathcal{A}) \mid X \subseteq S^\Pi_\alpha, \mathcal{A} \subseteq \Delta_x\}$. Third, we introduce the *cost* of an action, e.g. define action α as $\langle \mathsf{P}(\alpha), \mathsf{X}(\alpha), \mathsf{cost}(\alpha)\rangle$ where $\mathsf{cost}(\alpha) \in \mathbb{R}^+$. This induces an additive plan cost function $\mathsf{cost}(\Pi_\emptyset(\xi_0, \ldots, \xi_k) = \Sigma_{k' \le k} \mathsf{cost}(\xi_{k'})$ that will guide plan search. Another modification needed is the following.

Relativizing plans to domains:

Even if any plan Π originates from a fixed planning domain $\mathbb{M}$, we can think of so-originated Π also as a plan for some other planning domain $\mathbb{M}'$, and (re-)evaluate Π w.r.t. $\mathbb{M}'$. This is useful when an agent revises her beliefs or is communicated a plan. We denote by $\mathbb{M} \sqsubseteq \mathbb{M}'$ that $\mathbb{M}'$ is an expansion of $\mathbb{M}$, i.e. $\mathbb{M}'$ is such that for all $X \in \mathbb{M}$, its counterpart $X' \in \mathbb{M}'$ satisfies $X \subseteq X'$. And similarly for $T \sqsubseteq T'$. All these expansions may actually translate Π into $\Pi' = \Pi\binom{\alpha_\Psi \alpha_G}{\alpha_{\Psi'} \alpha_{G'}}$.

LEMMA 1. *Proto-states S^Π_α are $\subseteq$-monotonic under expansions of T: $T \sqsubseteq T'$ implies $S^\Pi_\alpha \subseteq S^{\Pi'}_\alpha$, where $\Pi' := \Pi\binom{\alpha_\Psi}{\alpha_{\Psi'}}$.*

Also, note that $\mathsf{PArgs}(S^\Pi_\alpha, \cdot)$ is $\subseteq$-monotonic under expansions of Δ: $\Delta \subseteq \Delta'$ makes $\mathsf{PArgs}(S^\Pi_\alpha, \Delta) \subseteq \mathsf{PArgs}(S^\Pi_\alpha, \Delta')$.

LEMMA 2. *Action-action and action-argument threats (with action $\neq \alpha_\Psi$) do not increase after expansions of T.*

In contrast, new (α_Ψ) action- and argument-argument threats may appear after expansions of Ψ and, resp., Ψ-or-Δ.

For expansions $\mathbb{M}' \sqsupseteq \mathbb{M}$ a sufficient condition for $\mathbb{M}'$ to accept Π' is that $\mathbb{M}'$ at least contains the elements of Π (and, for Ψ', no more than Ψ).

LEMMA 3. *Let $\mathbb{M} = ((\Psi, \Delta), A, G)$ be a planning domain and Π a plan for $\mathbb{M}$. Define $\mathbb{M}^\Pi = ((\Psi^\star, \Delta^\star), A^\star, G^\star)$ as: $\Psi^\star = \{\ell \in \mathsf{Lit} \mid (\alpha_\Psi, \ell, \cdot) \in \mathcal{CL}(\Pi)\}$, $\Delta^\star = \bigcup \pi_0[\mathcal{SL}(\Pi)]$, $G^\star = G \setminus \mathsf{goals}(\Pi)$ and $A^\star = (A_\Pi \setminus \{\alpha_\Psi, \alpha_G\}) \cup \{\alpha_{\Psi^\star}, \alpha_{G^\star}\}$. Then, for any $\mathbb{M}' = ((\Psi', \Delta'), A', G')$ with $\Psi' \subseteq \Psi$,*

$$\Pi\tbinom{\alpha_\Psi \alpha_G}{\alpha_{\Psi'} \alpha_{G'}} \text{ is a plan for } \mathbb{M}' \text{ iff } \mathbb{M}^\Pi \sqsubseteq \mathbb{M}'$$

Only these types of threats that may increase after expansions will be open to argumentation when evaluating the plan's flaws (or its planhood). These results justify the sufficiency of the next relativizations[10]:

DEFINITION 1. *Let Π be a POP for a given $((\Psi, \Delta), A, G)$, and let $T' = (\Psi', \Delta')$ be another de.l.p.. We define the* relativization *of S^Π_α to Ψ', as $S^{\Psi'}_\alpha = S^{\Pi'}_\alpha$, with $\Pi' = \Pi\binom{\alpha_\psi}{\alpha_{\psi'}}$. We denote by $\mathsf{Threats}^{T'}(\Pi)$ the set of threats to argument steps in Π according to T', as the set of tuples $(\kappa, (\mathcal{A}, g, \alpha)) \in (\mathcal{P}(\Delta) \cup \mathsf{Lit}) \times \mathcal{SL}(\Pi)$ such that either:*

[10]Initial dummy action α_Ψ is also initially different to each agent. We will assume each agent x, when speaking, uses the convention of referring to *her* initial action, i.e. α_{ψ_x}, by using the neutral symbol α_Ψ.

$\kappa \subseteq \Delta$, $\mathsf{base}(\kappa) \subseteq S_\alpha^{\Psi'}$, *$\kappa$ defeats $\mathcal{A}$, and undefeated in $S_\alpha^{\Psi'}$; or $\kappa = \ell$, with $\ell \in \mathsf{X}(\alpha_{\Psi'}) \cap \overline{\mathsf{literals}(\mathcal{A})}$, and $\alpha_{\Psi'}$ makes $\ell \in S_\alpha^{\Psi'}$ true.*

4. COOPERATIVE PLANNING

In the following, we assume we have a set of agents $\mathsf{Ag} = \{1, \ldots, k\}$, each one with a planing domain $\mathbb{M}_x = ((\Psi_x, \Delta_x), A_x, G_x)$. In purely cooperative scenarios, agents have no individual interests (i.e. $G_i = G_j$ for any $i, j \in \mathsf{Ag}$) and hence no incentives to retain relevant information. Moreover, we assume $\bigcup_{i \in \mathsf{Ag}} \Psi_i$ is a consistent set. Also, a unique team dialogue to find a solution would suffice. Before presenting dialogues for cooperative plan search, we introduce first a simpler dialogue to evaluate a fixed plan.

4.1 Argumentative Plan Evaluation.

We present now a turn-based dialogue (an agent talking only during her turns) permitting agents i, j to collaborate to discover threats to any argument step $\mathcal{A}$, i.e. with $(\mathcal{A}, \cdot, \alpha) \in \mathcal{SL}(\Pi)$. Here Π is a plan for some $\mathbb{M}_i$ made public. (That is, we assume $\mathbb{M}_\Pi \sqsubseteq \mathbb{M}_x$, $x \in \mathsf{Ag}$.). All agents may contribute to argue against $\mathcal{A}$.

Agents are enumerated by function $\varepsilon : \mathbf{N}^+ \to \mathsf{Ag}$ as: $\varepsilon(i + r \cdot |\mathsf{Ag}|) = i$ for any $r, i \in \mathbb{N}^+$ and $i \leq |\mathsf{Ag}|$; that is, ε assigns turns to agents this way: $1, 2, \ldots, k, 1, 2, \ldots$ At each turn $n+1$, agent $\varepsilon(n+1)$ sends a set $\mathbb{A}^{n+1}$ of pre-arguments[11] $(X, \mathcal{B})$ or initial facts $(\emptyset, \ell)$, against an argument $\mathcal{A}$ used in some support link $(\mathcal{A}, \cdot, \alpha)$. For each $(X, \mathcal{B}) \in \mathbb{A}^{n+1}$, any other agent $j \neq \varepsilon(n+1)$ learns as initial facts those literals stated in X that are not in her view of the proto-state, i.e. with $\ell \in X \setminus S_\alpha^{\psi_j^n}$. All rules from $\mathcal{B}$ which are novel to j are learned as well. Formally,

DEFINITION 2. *For $x \in \mathsf{Ag}$ let $\mathbb{M}_x = ((\Psi_x, \Delta_x), A_x, G)$ be given, and $\varepsilon : \mathbb{N}^+ \to \mathsf{Ag}$ as above. Let Π be a plan communicated by, say, agent 1 to Ag. We define for each $x \in \mathsf{Ag}$, $\mathbb{A}^0 = \emptyset$, $\psi_x^0 = \Psi_x$, $\Delta_x^0 = \Delta_x$ and*

$$\mathbb{A}^{n+1} = \{(\kappa, (\mathcal{A}, \cdot, \alpha) \in \mathcal{P}(\mathsf{Lit}) \times \mathsf{Threats}^{T^{n+1}_{\varepsilon(n+1)}}(\Pi) \mid \text{either } \kappa = (X, \mathcal{B}) \text{ and } X \subseteq \mathsf{base}(\mathcal{B}) \cap S_\alpha^{\psi^{n+1}_{\varepsilon(n+1)}}; \text{ or } \kappa = (\emptyset, \ell) \in \{\emptyset\} \times \mathsf{X}(\alpha_{\psi^{n+1}_{\varepsilon(n+1)}})\}$$

$$\psi_x^{n+1} = \psi_x^n \cup \bigcup\{X \setminus S_\alpha^{\psi_x^n} \mid ((X, \mathcal{B}), (\mathcal{A}, \cdot, \alpha)) \in \mathbb{A}^{n+1}\} \cup \{\ell \in \mathsf{Lit} \mid ((\emptyset, \ell), (\mathcal{A}, \cdot, \cdot)) \in \mathbb{A}^{n+1}\}$$

$$\Delta_x^{n+1} = \Delta_x^n \cup (\pi_1[\mathbb{A}^{n+1}] \setminus \mathsf{Lit})$$

Finally, let $n^\star$ be the smallest number such that $\mathbb{A}^{n^\star} = \ldots = \mathbb{A}^{n^\star + |\mathsf{Ag}|} = \emptyset$. We define $\psi_x^\omega = \psi_x^{n^*}$, and $\Delta_x^\omega = \Delta_x^{n^*}$.

First note that literals learned in ψ_x^{n+1} from some $((X, \mathcal{B}), \cdot) \in \mathbb{A}^{n+1}$ really come from the agent $n+1$'s ψ-set and propagated to this proto-state.

LEMMA 4. *If $\ell \in X \setminus S_\alpha^{\psi_x^n}$ for some $((X, \mathcal{B}), (\mathcal{A}, \cdot, \alpha)) \in \mathbb{A}^{n+1}$, then $\ell \in \psi^n_{\varepsilon(n+1)}$.*

Also note that, since the de.l.p. of each agent is finite, $n^\star$ is finite, i.e. these dialogues will always terminate in a finite number of steps. This dialogue is compared next with centralized plan evaluation, where (a) we consider the *fusion* of agents' initial de.l.p.'s $T_{\Sigma\mathsf{Ag}} = (\Psi_{\Sigma\mathsf{Ag}}, \Delta_{\Sigma\mathsf{Ag}}) = (\bigcup_{x \in \mathsf{Ag}} \Psi_x, \bigcup_{x \in \mathsf{Ag}} \Delta_x)$, and then (b) a central planner computes arguments and threats in this new de.l.p. $(\Psi_{\Sigma\mathsf{Ag}}, \Delta_{\Sigma\mathsf{Ag}})$. The next theorem, then, compares the result of any agent after the evaluation dialogue for $\mathsf{Threats}^{T_x^\omega}(\Pi)$ with that of centralized evaluation $\mathsf{Threats}^{T_{\Sigma\mathsf{Ag}}}(\Pi)$. Even if $T_x^\omega \sqsubset T_{\Sigma\mathsf{Ag}}$ may hold, both evaluations agree on threats detected in Π and whether Π is a plan.

THEOREM 1. *Given $\mathbb{M}_x = ((\Psi_x, \Delta_x), A, G)$ for each $x \in \mathsf{Ag}$, Π a plan for $\mathbb{M}_1$ communicated to $\mathsf{Ag} \setminus \{1\}$. Then, for each x, Π is a plan for $((\psi_x^\omega, \Delta_x^\omega), A, G)$ iff it is for $(T_{\Sigma\mathsf{Ag}}, A, G)$, and $\mathsf{Threats}^{(\psi_x^\omega, \Delta_x^\omega)}(\Pi) = \mathsf{Threats}^{(\Psi_{\Sigma\mathsf{Ag}}, \Delta_{\Sigma\mathsf{Ag}})}(\Pi)$*

4.2 Dialogue-based A^* plan search.

The next step is to use these dialogues as part of more dynamic dialogues wherein new plans are proposed. The main result of this paper is that we can decentralize multi-agent planning, at least in cooperative scenarios, by using a dialogue-based plan search procedure. This is done by comparing these dialogues with centralized planning in the fusion of agents' planning domains $\mathbb{M}_{\Sigma\mathsf{Ag}} = (T_{\Sigma\mathsf{Ag}}, A_{\Sigma\mathsf{Ag}}, G)$, where $A_{\Sigma\mathsf{Ag}} = \bigcup_{x \in \mathsf{Ag}} A_x$. But first, we recall A^* search and show it can be used in single-agent DeLP-POP.

4.2.1 A^* search in DeLP-POP.

Search algorithms, in the literature, are abstractly defined with non-deterministic choice. In DeLP-POP plan search we saw two such places for non-deterministic choice exist: the selection of the next flaw to be solved[12] (this is optional) and a selection function g for the next refinement, based on minimizing some evaluation function $f(\Pi)$ that estimates the cost of a solution refining Π.

We opt for an A^* search algorithm, based on delayed termination and an additive evaluation function $f(\Pi) = \mathsf{cost}(\Pi) + f'(\Pi)$, where $f'(\Pi)$ is some heuristic estimation of the cost of some best solution $\Pi^\star$ extending Π.

Recall that A^* procedure is as follows. Start with the initial node $\Pi_\emptyset$, and define sets $\mathsf{open} = \{\Pi_\emptyset\}$ and $\mathsf{closed} = \emptyset$. At each iteration, open is expanded with all generated refinements of current node Π, while Π is sent to closed. Then, we minimize $f[\mathsf{open}]$ to select a refinement $\Pi(\xi)$.

Notice that A^* does not terminate at the first solution, but keeps exploring for less costly possibilities, guided by $g(\mathsf{open}) = \mathrm{argmin}(f(\mathsf{open}))$. If, moreover, f' is optimistic, i.e. $f'(\Pi) \leq f'(\Pi^\star) = \mathsf{cost}(\Pi^\star$, then this $A^\star$ search finds an optimal solution (if a solution exists). Below we will consider the particular case $f'(\Pi) = 0$, so our next-refinement choice function will be just $g(\mathsf{open}) := \mathrm{argmin}(\mathsf{cost}[\mathsf{open}])$.

For a given planning domain $\mathbb{M}$, we define $\mathsf{Plans}_g(\mathbb{M})$ as the set of nodes in $\mathsf{Plans}(\mathbb{M})$ that are generated under A^* search with g.

PROPOSITION 1. *If f' be optimistic, g is admissible for DeLP-POP search: $\mathsf{Sol}(\mathsf{Plans}(\mathbb{M})) \neq \emptyset$ iff $\mathsf{Sol}(\mathsf{Plans}_g(\mathbb{M})) \neq \emptyset$, and a solution $\Pi^\star$ in the latter is optimal.*

The reason is as follows. Suppose $\mathbb{M} = (T, A, G)$ is a finite domain, so that the cost of any action $\alpha \in A$ has positive lower bound $\mathsf{cost}[A] \geq \delta > 0$. Then if (T, A, G) is solvable,

[11] By exchanging arguments only, an agent might fail to share information, if unaware of its relevance.

[12] As examples of such heuristics: FAF, where flaws are according *fewer alternatives first*, as [6]'s Z-LIFO. Or the threat detect-&-solve order used in [5]'s algorithm.

a search algorithm guided by g is guaranteed to output an optimal solution in $\mathsf{Sol}(\mathsf{Plans}_g(\mathbb{M}))$ if every infinite path has unbounded cost (see [8]). To see this: if the path contains infinite action steps then it is unbounded, since A is finite implies that $0 < \delta \leq \mathsf{cost}[A]$ for some δ. Now, if $\mathbb{M}$ is finite, so is $\mathsf{flaws}(\Pi)$; hence null-cost threat resolution moves must be finite. The same reasoning, plus the no-argument-supports-argument policy, implies there can be no infinite sequence of null cost argument steps so we are done.

Hence, A^* can be applied to DeLP-POP plan search for a fixed domain, e.g. centralized $\mathbb{M}_{\Sigma\mathsf{Ag}}$. Below, we show that A^* is also applicable to *dialogue-based* multi-agent plan search.

4.2.2 A^* search in cooperative DeLP-POP.

Given agents $\mathsf{Ag} = \{1, \ldots, k\}$, decentralized plan search is also realized as a turn-based dialogue. Turns are now of the form $(n, m) \in \mathbb{N} \times \mathbb{N}$, ordered lexicographically: (n, m) occurs before (n', m') iff $n < n'$, or $n = n'$ and $m < m'$. The agent speaking at (n, m) is $\varepsilon(m)$, who sends a set $\mathbf{\Pi}^{(n,m)}$ of refinements of the plan selected at the n-th iteration of A^*, and a set $\mathcal{U}^{(n,m)}$ of potential threats to previous plans in $\mathbf{\Pi}^{(n,m')}$ for $m' \leq m$. Potential threats are now labeled with the link *and* the plan targeted, say Π' in $\mathbf{\Pi}^{(n,m')}$. In terms of evaluation dialogues, $\mathcal{U}^{(n,m)}$ contains, for each such Π', the corresponding $\mathbb{A}^{m-m'} \times \{\Pi'\}$ (under some permutation $\tau : \mathsf{Ag} \to \mathsf{Ag}$ and initial domains set at $\langle \mathbb{M}^{(n,m')}_{\tau(x)} \rangle_{x \in \mathsf{Ag}}$).

Other agents $x \neq \varepsilon(m)$ learn from $\mathcal{U}^{(n,m)}$ and $\mathbf{\Pi}^{(n,m)}$: (1) literals from pre-arguments and causal links of the form $(\alpha_\Psi, \ell, \cdot)$, (2) rules from pre-arguments and support links, and (3) other agents' actions from suggested plans. This grants that each $\Pi' \in \mathbf{\Pi}^{(n,m)}$ is understood: $\mathbb{M}^{\Pi'} \sqsubseteq \mathbb{M}^{(n,m)}_x$.

Only when, during $|\mathsf{Ag}|$ successive turns $(n, m), \ldots, (n, m + |\mathsf{Ag}|)$, agents do not submit more plans or possible threats, we set $\omega(n) = m$ and move to turn $(n + 1, 0)$. To do so, the set of open nodes is updated with refinements for the current plan: $\mathbf{\Pi}^{(n,\omega(n))} = \mathbf{\Pi}^{(n-1,\omega(n-1))} \cup \bigcup_m \mathbf{\Pi}^{(n,m)}$.

At $(n+1, 0)$ agents select the best of open nodes: $\mathbf{\Pi}^{(n+1,0)} = \{g(\mathbf{\Pi}^{(n,\omega(n))}\}$. If this contains no flaw, the dialogue terminates. Otherwise the procedure starts again for this plan.

Definition 3. *Given $\mathbb{M}_x = ((\psi_x, \Delta_x), A_x, G)$ as before, we set $\mathbb{M}^{(0,0)}_x := \mathbb{M}_x$ and define $\mathbf{\Pi}^{(0,0)} = \mathcal{U}^{(0,\cdot)} = \mathcal{U}^{(\cdot,0)} = \emptyset$, $\mathsf{flaw}(0) = h(G)$, and $\mathbf{\Pi}^{(0,1)} = \{\Pi_\emptyset\}$. And,*

$$\mathbf{\Pi}^{(n,m+1)} = \{\Pi(\xi) \in \mathsf{Plans}(\mathbb{M}^{(n,m)}_{\varepsilon(m+1)}) \mid \Pi \in \mathbf{\Pi}^{(n,0)}, \text{ and } \mathsf{flaws}(\Pi(\xi)) \smallsetminus \mathsf{flaws}(\Pi) \neq \emptyset\}$$

$$\mathbf{\Pi}^{(n+1,\omega(n+1))} = (\mathbf{\Pi}^{(n,\omega(n))} \smallsetminus g(\mathbf{\Pi}^{(n,\omega(n))})) \cup \mathbf{\Pi}^{(n,m_n)},$$

where $m_n = min\ m$ s.t. $\mathbf{\Pi}^{(n,m)} = \ldots = \mathbf{\Pi}^{(n,m+|\mathsf{Ag}|-1)}$ and $\mathcal{U}^{(n,m)} = \ldots = \mathcal{U}^{(n,m+|\mathsf{Ag}|-1)} = \emptyset$

$$\mathbf{\Pi}^{(n+1,0)} = \{g(\mathbf{\Pi}^{(n,\omega(n))})\}$$

$$\mathcal{U}^{(n,m+1)} = \{(\kappa_0, \kappa_1), (\kappa', \ell, \kappa''), \Pi') \mid \Pi' \in \mathbf{\Pi}^{(n,m+1)} \text{ and } (\kappa_1, (\kappa', \ell, \kappa'')) \in \mathsf{Threats}^{T^{(n,m)}_{\varepsilon(m+1)}}(\Pi') \text{ and } \kappa_0 \subseteq \mathsf{base}(\kappa_1) \text{ or } (\kappa_0, \kappa_1) \in \{\emptyset\} \times \mathsf{Lit}\}$$

At turns of the form $(n, m + 1)$ agents learn as follows:

Definition 4. *Each agent $x \neq \varepsilon(m+1)$ updates, at turn $(n, m + 1)$,*

$$\begin{aligned}
\psi^{(n,m+1)}_x &= \psi^{(n,m)}_x \cup (\pi_1(\mathcal{U}^{(n,m+1)}) \cap \mathsf{Lit}) \cup \\
&\quad \bigcup\{X \smallsetminus S^{\psi^{(n,m)}_x}_{\alpha_1} \mid ((X, \mathcal{B}), (\mathcal{A}, \cdot, \alpha_1)) \in \mathcal{U}^{(n,m+1)}\} \\
\Delta^{(n,m+1)}_x &= \Delta^{(n,m)}_x \cup \{\pi_0(\xi) \mid \xi \in \mathcal{SL}[\mathbf{\Pi}^{(n,m+1)}] \cup \ldots \\
&\quad \cup \pi_1(\{(\kappa, \ldots) \in \mathcal{U}^{(n,m+1)}) \mid \pi_1(\kappa, \ldots) \notin \mathsf{Lit}\}) \\
A^{(n,m+1)}_x &= A^{(n,m)}_x \cup \{\alpha \in A_{\Pi(\xi)} \mid \Pi(\xi) \in \mathbf{\Pi}^{(n,m+1)}\}
\end{aligned}$$

For sets $X^{(n,\cdot)}_x$ defined here plus $\mathbb{M}^{(n,\cdot)}_x$ we define $X^{(n+1,0)}_x = X^{(n,\omega(n))}_x = \bigcup_m X^{(n,m)}_x$, and $X^{\omega}_x = \bigcup_{n \in \omega} X^{(n,0)}_x$.

Theorem 2. *Let $\langle \mathbb{M}_x \rangle_{x \in \mathsf{Ag}}$ and g be as above. Then, $\mathsf{Sol}(\mathsf{Plans}_g(\mathbb{M}_{\Sigma\mathsf{Ag}})) \neq \emptyset$ iff $\mathsf{Sol}(\mathsf{Plans}_g(\mathbb{M}^{\omega}_x)) \neq \emptyset$, for any x; moreover, a solution $\Pi^\star$ in the latter is optimal.*

Thus, agents may safely use these dialogues to find an optimal, cooperative plan which makes use of their abilities.

5. EXAMPLE OF APPLICATION

The next example (see Figure 4[13]) shows a scenario to Cooperative Planning. There are three different locations in this scenario *Bejing*, *Fuzhou* and *Taipei*. Our multi-agent systems is composed of two agents, *Joe* and *Ann*, who wish to travel to *Taipei* to attend the AAMAS conference as invited speakers. As can be seen, there are several direct or indirect connections between *Bejing* and *Taipei*: via car and ship, train and ship, or plane. The agents, the car, the train and the plane are initially located at *Bejing*, and the goal ($G = \{(at\ \mathsf{Ag}\ l3)\}$) is to have the two agents at *Taipei* subject to the restriction that they must always travel together. Literals and actions are the following[14]:

- $l1$, $l2$, $l3$ - *Bejing*, *Fuzhou* and *Taipei*,
- *car*, *tra*, *pl*, *shi* - a car, a train, a plane, a ship,
- *r*, *rl*, *al*, *ml* - a road, a railway, an airline company, a maritime line,
- *bw*, *sn*, *wg*, *ss* - bad weather, snow, wind gusts, stormy sea,
- *br*, *ll*, *esf*, *aeo* - bad railroad, landslides, electrical supply failure, airplane engines work well (after test)
- *va*, *ds*, *ip*, *gw* - volcano ash cloud, dangerous situation, risk of increased pollution, contribution to global warming,
- *h*, *tj*, *kudTV*, *kudI* - holidays, traffic jam, kept up to date by TV news, kept up to date by Internet news,
- μ_C, μ_P, μ_T, μ_S - moved car, moved plane, moved train and moved ship

1. $mP(pl, j, k)$: moving plane '*pl*' from location '*j*' to '*k*'. It is necessary an airline company to travel from '*j*' to '*k*', the plane in '*j*' and both *Joe* and *Ann* in '*j*'. Moving a plane takes 2 time unit and 400 cost units.
2. $mT(tra, j, k)$: moving train '*tra*' from location '*j*' to '*k*'. This action takes 6 time units and 200 cost units.
3. $mS(shi, j, k)$: moving ship '*shi*' from location '*j*' to '*k*'. This action takes 3 time units and 100 unit cost.
4. $fMc(car, j, k)$: fast-moving car '*car*' from location '*j*' to '*k*'. This action takes 8 time units and 80 cost units.

[13]Get Directions on Google maps, http://maps.google.es

[14]We consider propositional STRIPS planning representation, and the default proposition (*have p*) to any literal p that does not have an associated proposition.

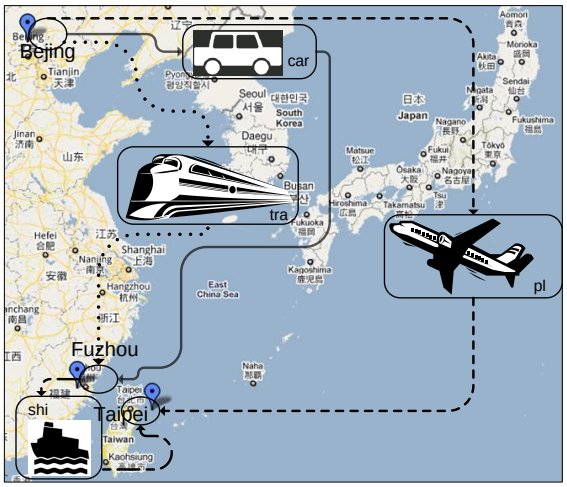

Figure 4: Scenario of the application example

$$A^{(0,0)}_{Joe} = \left\{ \begin{array}{l} 1.\ \{\mu_C, ip\} \xleftarrow{fMc} \{(link\ r\ l1\ l2),\ (at\ car\ l1), \\ (at\ \mathsf{Ag}\ l1)\} \\ 2.\ \mu_P \xleftarrow{mP} \{(link\ al\ l1\ l3),\ (at\ pl\ l1), \\ (at\ \mathsf{Ag}\ l1)\} \end{array} \right\}$$

$$A^{(0,0)}_{Ann} = \left\{ \begin{array}{l} 3.\ \mu_T \xleftarrow{mT} \{(link\ rl\ l1\ l2),\ (at\ tra\ l1), \\ (at\ \mathsf{Ag}\ l1)\} \\ 4.\ \mu_S \xleftarrow{mS} \{(link\ ml\ l2\ l3),\ (at\ shi\ l2), \\ (at\ \mathsf{Ag}\ l2)\} \end{array} \right\}$$

$$\Psi^{(0,0)}_{Joe} = \left\{ \begin{array}{c} wg;\ aeo;\ kudTV; (at\ \mathsf{Ag}\ l1); \\ (at\ pl\ l1);\ (link\ al\ l1\ l3);\ (link\ r\ l1\ l2); \end{array} \right\}$$

$$\Psi^{(0,0)}_{Ann} = \left\{ \begin{array}{c} kudI;\ (at\ \mathsf{Ag}\ l1);\ (at\ tra\ l1);\ (at\ shi\ l2) \\ (link\ rl\ l1\ l2);\ (link\ ml\ l2\ l3) \end{array} \right\}$$

Figure 5: Knowledge of actions and initial facts.

We describe next the initial planning domains: for $x \in \mathsf{Ag} = \{Ann, Joe\}$, let $\mathbb{M}^{(0,0)}_x = ((\Psi^{(0,0)}_x, \Delta^{(0,0)}_x), A^{(0,0)}_x, G)$ be defined as in Figures 5 and 6. Actions $\alpha = (\mathsf{P}(\alpha), \mathsf{X}(\alpha), \cdot)$ are represented under the form $\mathsf{X}(\alpha) \xleftarrow{\alpha} \mathsf{P}(\alpha)$. *Ann* and *Joe* have different knowledge so two pieces of derived information from each agent can appear to be contradictory. Let's assume that *Joe* uses TV as a source of information, but *Ann* prefers Internet to keep up to date, and both agree in finding a plan that minimizes the time units.

In what follows, we explain how to obtain an optimal plan $\Pi^\star$ that satisfies the goal $G = \{(at\ \mathsf{Ag}\ l3)\}$.

The planning process starts with *Ann*'s empty plan $\Pi_\emptyset$, essentially, $\{\alpha_\emptyset \prec \alpha_G\}$ and $\mathcal{U}^{(0,1)} = \emptyset$. Joe learns nothing from it; and both agents set $g(\mathbf{\Pi}^{(0,\omega(0))}) = \Pi_\emptyset$. Then $\mathsf{flaws}(\Pi)$ returns $(at\ \mathsf{Ag}\ l3)$. At turn (1,1) *Ann* suggests the ship argument, while at next turn $(1,2)$, *Joe* puts forward this argument step (Figure 7(a)):

$\Pi_\emptyset(\xi^{Joe}) \in \mathbf{\Pi}^{(1,2)}$ where $\xi^{Joe} = (\mathcal{A}^{\text{Joe}}, (at\ \mathsf{Ag}\ l3), \alpha_G))$ and $\mathcal{A}^{\text{Joe}} = (\{(at\ \mathsf{Ag}\ l3) \prec \mu_P\})$

Ann learns the rule in $\mathcal{A}^{\text{Joe}}$. This is the plan with less cost, so it selected at $\mathbf{\Pi}^{(2,0)}$ with $\mathsf{flaws}(\Pi_\emptyset(\xi^{Joe})) = \{\mu_P\}$.

At $(2,1)$ turn, *Ann* cannot refine this plan. This is done,

$$\Delta^{(0,0)}_{Joe} = \left\{ \begin{array}{c} \{(at\ pl\ l3), (at\ \mathsf{Ag}\ l3)\} \prec \mu_P; \\ \{(at\ car\ l2), (at\ \mathsf{Ag}\ l2)\} \prec \mu_C; \\ \{\sim(at\ tra\ l2), \sim(at\ \mathsf{Ag}\ l2)\} \prec \{\mu_T, br\}; \\ \{\sim(at\ shi\ l3), \sim(at\ \mathsf{Ag}\ l3)\} \prec \{\mu_S, ss\}; \\ br \prec ll;\ ll \prec wg;\ br \prec esf;\ esf \prec sn; \\ sn \prec kudTV;\ tj \prec h;\ h \prec kudTV; \\ ss \prec bw;\ bw \prec wg;\ \sim va \prec aeo; \end{array} \right\}$$

$$\Delta^{(0,0)}_{Ann} = \left\{ \begin{array}{c} \{\sim(at\ pl\ l3), \sim(at\ \mathsf{Ag}\ l3)\} \prec \{\mu_P, ds\} \\ \{\sim(at\ car\ l2), \sim(at\ \mathsf{Ag}\ l2)\} \prec \{\mu_C, tj\} \\ \{(at\ tra\ l2), (at\ \mathsf{Ag}\ l2)\} \prec \mu_T; \\ \{(at\ shi\ l3), (at\ \mathsf{Ag}\ l3)\} \prec \mu_S; \\ ds \prec va;\ va \prec kudI;\ \sim ss \prec \sim bw; \\ \sim bw \prec h;\ h \prec kudI;\ \sim ll \prec \sim bw;\ \sim br \prec \sim bw; \\ \sim bw \prec kudI;\ \sim sn \prec kudI;\ gw \prec ip; \end{array} \right\}$$

Figure 6: Defeasible rules known by each agent.

at turn $(2,2)$ by *Joe*: $\Pi_\emptyset(\xi^{Joe}, (mP, \mu_P, \mathcal{A}^{\text{Joe}})) \in \mathbf{\Pi}^{(2,2)}$, where he proposes the action $mP(pl,\ l1,\ l3)$ to enforce μ_P (Figure 7(b)). Let Π' denote this plan. Each agent x learns in $(2,2)$ that $\mu_P \in S^{\psi^{(2,2)}_x}_{\alpha_G}$. Ann learns action mP.

Now its *Ann*'s turn $(2,3)$. She finds an argument-argument threat to $\mathcal{A}^{\text{Joe}}$ based on her initial knowledge of $kudI$. She sends $\mathcal{U}^{(2,3)} = \{((\{kudI\}, \mathcal{B}^{\text{Ann}}), (\mathcal{A}^{\text{Joe}},\ at\ \mathsf{Ag}\ l3, \alpha_G), \Pi')\}$ where $\mathcal{B}^{\text{Ann}} = \{\sim(at\ \mathsf{Ag}\ l3) \prec \{\mu_P, ds\};\ ds \prec va;\ va \prec kudI\}$ (Figure 7(c)). The initial fact $kudI$ and these rules are learnt by *Joe*. Assume *Joe*'s plan is selected at $\mathbf{\Pi}^{(3,0)}$ with $\mathsf{flaws}(\Pi')$ containing *Ann*'s threat based on $\mathcal{B}^{Ann}$.

At Ann's turn $(3,1)$, she finds nothing else relevant to Joe's plan. Joe's turn (3,2). To solve *Ann*'s threat, *Joe* selects a *Defeat* move against ds, based on his knowledge. $\mathbf{\Pi}^{(3,2)} = \{\Pi'(\text{Defeat}(\mathcal{C}^{\text{Joe}}, \mathcal{B}^{\text{Ann}}))\}$ where $\mathcal{C}^{\text{Joe}} = (\{aeo\}, \{\sim va \prec aeo\})$. It is a Defeat resolution move since: $\sim\mathsf{concl}(\mathcal{C}^{\text{Joe}}) \in \mathsf{literals}(\mathcal{B}^{\text{Ann}}))$ (Figure 8(d)).

In summary, *Joe* suggested to take the plane to arrive to *Taipei*, but *Ann* attacked the proposal because the volcano ashes are expected according to the Internet information, and *Joe* replied that this situation will not affect the flight between *Beijing* and *Taipei* (according to the results on engine tests). For space reasons, we omit the rest of the dialogue showing this is plan can be refined to an optimal solution.

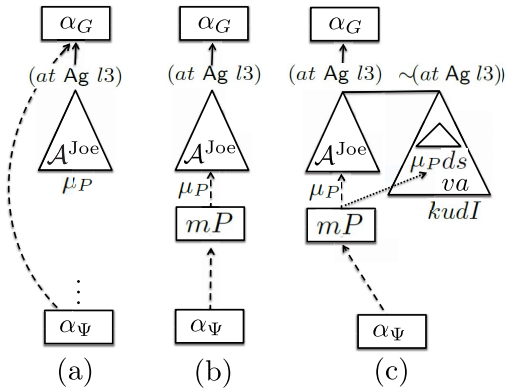

Figure 7: (a), (b): Joe's turns and (c): Ann's turn

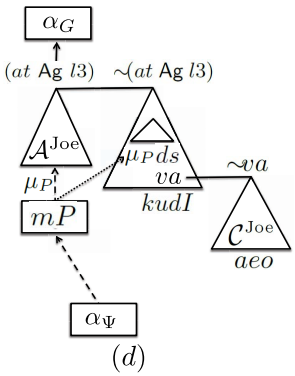

Figure 8: (d): Joe's turn

6. RELATED WORK

The work presented here is similar to several proposals found in the literature: multi-agent argumentation (in non-dynamic scenarios), cooperative planning (without defeasible argumentation) and centralized planning.

Some systems that build on argumentation apply theoretical reasoning for the generation and evaluation of arguments to build applications that deal with incomplete and contradictory information in dynamic domains. Some proposals in this line focus on planning tasks, or also called practical reasoning, i.e. reasoning about what actions are the best to be executed by an agent in a given situation. Dung's abstract system for argumentation [3] has been used for reasoning about conflicting plans and generate consistent sets of goals [1, 7]. Further extensions of these works present an explicit separation of the belief arguments and goals arguments and include methods for comparing arguments based on the worth of goals and the cost of resources [10]. In any case, none of these works apply to a multi-agent environment. A proposal for dialogue-based centralized planning is that of [12], but no argumentation is made use of. The work in [2] presents a dialogue based on an argumentation process to reach agreements on plan proposals. Unlike our focus on an argumentative and stepwise construction of a plan, this latter work is aimed at handling the interdependencies between agents' plans. On the other hand, we can also find some systems that realize argumentation in multi-agent systems using defeasible reasoning but are not particularly concerned with the task of planning [13]. All in all, the novelty of our approach is the combination of all these aspects: defeasible reasoning, decentralized planning and multi-agent systems.

7. CONCLUSIONS AND FUTURE WORK

We have presented a decentralized A^* plan search algorithm for multiagent argumentative planning in the framework of DeLP-POP. This search is implemented as a dialogue between agents, which cooperate to criticize or defend alternative plans by means of defeasible arguments. Only potentially relevant information is exchanged in the dialogue process, which terminates in a provably optimal solution upon which agents cannot disagree.

For future work, several directions seem promising: extending the present approach to other multiagent scenarios, like Argumentation-based Negotiation, or an extension into Temporal Planning.

8. ACKNOWLEDGMENTS

The authors acknowledge partial support of the Spanish MICINN projects CONSOLIDER-INGENIO 2010 Agreement Technologies CSD2007-00022; TIN2009-14704-C03-03; LoMoReVI FFI2008-03126-E/FILO (FP006); FPU grant AP2009-1896; TIN2008-06701-C03-01; TIN2008-04446 and PROMETEO/2008/051; and the Generalitat de Catalunya grant 2009-SGR-1434.

9. REFERENCES

[1] L. Amgoud. A formal framework for handling conflicting desires. In Proc. of *7th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU 2003*, LNAI 2711, Springer, pp. 552–563, 2003.

[2] A. Belesiotis, M. Rovatsos, and I. Rahwan. Agreeing on plans through iterated disputes. In Proc. of *9th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2010*, pp. 765–772, 2010.

[3] P. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence*, 77(2):321–357, 1995.

[4] A. García and G. Simari. Defeasible logic programming: An argumentative approach. *Theory and Practice of Logic Programming*, 4:95–138, 2004.

[5] D. García, A. García, and G. Simari. Defeasible reasoning and partial order planning. In Proc. of the *5th International Conference on Foundations of information and knowledge systems, FoIKS 2008*, LNCS 4932, pp. 311–328, 2008.

[6] A. Gerevini and L. Schubert. Accelerating partial-order planners: Some techniques for effective search control and pruning. *Journal of Artificial Intelligence Research*, 5:95–137, 1996.

[7] J. Hulstijn and L. van der Torre. Combining goal generation and planning in an argumentation framework. In Proc. of *NMR 2004 Workshop on Argument, Dialogue and Decision*, J. P. Delgrande and T. Schaub (Eds.), pp. 212-218, 2004.

[8] J. Pearl. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley, 1984.

[9] J. Penberthy and D. Weld. Ucpop: A sound, complete, partial order planner for adl. In Proc. of the *3rd International Conference on Knowledge Representation and Reasoning (KR'92)*, pp. 103–114, 1992.

[10] I. Rahwan and L. Amgoud. An argumentation-based approach for practical reasoning. In Proc. of *5th Conference on Autonomous Agents and Multi-Agent Systems, AAMAS 2006*, pp. 347–354, 2006.

[11] G. Simari and R. Loui. A mathematical treatment of defeasible reasoning and its implementation. *Artificial intelligence*, 53:125–157, 1992.

[12] Y. Tang, T. Norman, and S. Parsons. A model for integrating dialogue and the execution of joint plans. In Proc. of *ArgMAS 2009*, P. McBurney et al. (Eds.), LNAI 6057, pp. 60-78, 2010.

[13] M. Thimm. Realizing argumentation in multi-agent systems using defeasible logic programming. In Proc. of *ArgMAS 2009*, P. McBurney et al. (Eds.), LNAI 6057, pp. 175-194, 2010