On Practical Reasoning and Automated Planning

Federico Cerutti
Department of Information Engineering
University of Brescia, Italy
federico.cerutti@ing.unibs.it

Mauro Vallati
School of Computing and Engineering
University of Huddersfield, UK
m.vallati@hud.ac.uk

Abstract

Practical reasoning and automated planning are strictly related as they strive to answer to the same question: "which is the best course of action for an agent?" While the first research field addressed this topic mainly from an *epistemological* point of view, automated planning dealt with this question for the most partfrom a *heuristic* or *reasoning* perspective. In this position paper we want to discuss the improvements in terms of computational complexity of algorithms, and of knowledge representation and reasoning power on complex planning problems, which can be derived by applying practical reasoning techniques to planning problems. In particular, we sketch how argumentation-based structures for practical reasoning may help for improving the computational complexity of a state-of-the-art approach in optimal planning.

Introduction

The (McCarthy & Hayes 1987) seminal paper distinguishes between two parts of intelligence: the *epistemological* and the *heuristic* (or *reasoning*). The epistemological part is the representation of the world in such a form that the solution of problems follows from the facts expressed in the representation. The heuristic part is the mechanism that on the basis of the information solves the problems and decides what to do.

Searching for the "best" (w.r.t. some criteria) course of action for an agent is one of the most challenging topic in AI at it has been addressed for the most part in two research areas. Practical reasoning, one of the two areas, is mostly focused on the epistemological part of this topic, and thus, as noticed in (Girle *et al.* 2003), it is often seen as domain-dependent reasoning. Indeed, a decision support system for any given domain would have to take account of salient features of the domain in which the reasoning takes place. A completely general system would, therefore, have to "model the world".

On the other hand, the most prominent approach for dealing with the heuristic part of this challenging topic has been named as *automated planning*. While *planning* is defined as the deliberation process that chooses and organises actions by anticipating their expected effects, *automated planning* is (Ghallab, Nau, & Traverso 2004) the area of artificial intelligence that studies this deliberation process computationally. Its aim is to support the planning activity by reasoning on conceptual models, i.e. abstract and formal representation of the domain, of the effects and the combinations of actions, and of the requirements to be satisfied and the objectives to be achieved. Generally, automated planning is domain independent, since only a formal representation of the state of the world and of actions (with preconditions and effects) is considered.

This means that from the epistemological part, the domain of automated planning is a very simple and constrained representation of reality, therefore it seems a good candidate for being analysed through practical reasoning approaches, which is the main idea underlying this position paper. In particular, we will consider those approaches in practical reasoning lying on argumentation theory. An argumentation system is mainly a way for modelling both commonsense and formalised defeasible knowledge, and to determine the "coherent" or "consistent" pieces of knowledge. As described in (Prakken & Vreeswijk 2002), an argumentation system requires five elements: an underlying logical language, definitions of an argument, of conflicts between arguments and of defeat among arguments and, finally, a definition of the assessment of arguments, which can be used to define a notion of defeasible logical consequence. In the context of practical reasoning, arguments and conflicts are usually defined using *argument schemes* and *critical questions* (Walton, Chris, & Macagno 2008). Usually, e.g. (Atkinson & Bench-Capon 2007), these approaches rely to the *abstract argumentation frameworks* (Dung 1995) (AF), or on its extensions, as a way for determining the assessment of arguments. Indeed an AF is composed by a set of arguments seen as atomic elements (the inner structure is left unspecified, this is why it is called *abstract*, and why an argumentation framework can be built on the top of a set of instances of argument schemes), and an attack relation among them. Given this very simple formalisation, the choice of a semantics leads to a set of extensions each of which is a set of arguments that are collectively acceptable according to the given semantics. Different semantics select different sets of arguments. For instance, stable semantics selects the sets of arguments attacking any argument not in the extension, while complete semantics chooses the sets of arguments that are collectively acceptable (no argument attacks another argument in the set, and if an argument not in the set attacks an argument in the set, then the attacker is in turn attacked by an argument in the set).

Exploiting argumentation based approaches for practical reasoning in a planning context can lead to two main directions: from one point of view widely studied problems can be addressed from a different prospective and this may lead to an improvement on the algorithms used for solving them. From another point of view, we could deal with issues related to planning with uncertainty by exploit the large corpus of studies in the context of nonmonotonic reasoning and argumentation theory. For instance, (Dung 1995) shows how an argumentation framework can encompass approaches like logic programming with negation as failure or Reiter's default logic (Reiter 1980), and more recently probabilistic reasoning and reasoning with preferences emerged as hot topics in argumentation community.

Due to space limit, in this position paper we will discuss briefly how seeing automated planning as a practical reasoning problem could lead to an improvement of current approaches in the case of study we considered, namely optimal planner SatPlan (Kautz & Selman 1992; 1999). A very preliminary discussion on planning with uncertainty is provided in the conclusions.

A Case of Study: Optimal Planning

First of all, we consider a planning problems in a STRIPS-like language composed by function-free first-order literals under the closed-world assumption. Formally, a *planning problem* is a tuple $\mathbb{M} = \langle \Psi, A, G \rangle$ where Ψ is the goal of the planning problem, viz. a set of ground literals representing the initial facts, A is a set of actions, and G is a set of ground literals. An action $\alpha = \langle \mathsf{P}(\alpha), \mathsf{X}(\alpha) \rangle$ is composed by a set of preconditions and a set of effects. A *solution plan* to a planning problem is a linearly ordered finite sequence. Moreover, we assume an *incompatibility* relation as a symmetric non transitive relation on the set of literals. A set of literals is *contradictory* if at least two literals in this set are incompatible; otherwise it is *non-contradictory*.

In this paper, given a planning problem, we will focus on the problem of finding a makespan optimal solution plan. This is known to be an NP-complete problem. Among the outstanding approaches aimed at solving this problem, at the current stage of this research we considered a single case of study, namely SatPlan (Kautz & Selman 1992; 1999), which is based on the Graphplan's planning graph (Blum & Furst 1997). A planning graph is a directed acyclic leveled graph that alternates between a proposition level, i.e. a set of problem propositions, and an action level, i.e. a set of ground actions, and a set of special dummy actions, called *no-ops*, which propagate propositions of the previous level to the next one. If an action is in the graph, then its preconditions and effects appear in the corresponding proposition levels of the graph. SatPlan (Kautz & Selman 1992; 1999) uses a preprocessing algorithm to compute a lower (possibly exact) bound k of the optimal planning horizon. It converts the planning graph, constructed up to the length k, into a SAT problem, i.e., a propositional formula encoding the planning problem. If the SAT problem is solvable (there exists a variable assignment that satisfies the formula), a plan with at most k time steps can be derived. If the SAT problem is unsolvable (the formula is unsatisfiable), SatPlan generates a larger SAT problem using an increased bound ($k+1$), and so on, until the first satisfiable formula is reached.

While encoding a planning problem as a SAT problem gives the advantage of reusing the large corpus of efficient SAT algorithms, usually the starting value of k, computed by SatPlan, is significantly lower than the length of the optimal actual plan, therefore several useless unsolvable SAT instances are generated, in order to find the optimal solution. Moreover existing SAT-solvers are "blind" w.r.t. the structure of the planning problem since they consider only the SAT encoded problem, thus they do not reuse previously obtained results.

From a practical reasoning point of view, the problem of finding a makespan optimal plan is analogous to answer to the question: what action should I execute at time t? From an epistemological point of view, this requires to build an argument $\mathbf{A}$, whose scheme encompasses the following four elements:

1. the name of the action $a_{\mathbf{A}}$;
2. the time when an action should be executed $t_{\mathbf{A}}$;
3. the non-contradictory set of literals that are the preconditions required by the action $\mathcal{P}_{\mathbf{A}}$;
4. the non-contradictory set of literals that are the effects of the action $\mathcal{C}_{\mathbf{A}}$.

Therefore, a makespan optimal plan is represented by the minimal sets of arguments suggesting timed actions that once linearly ordered are the makespan optimal plan.

For a clearer comparison with SatPlan, let us recall two inconveniences of it:

Q1: it requires to iteratively increment the bound of the planning graph thus requiring to build several graphs and to evaluate them;

Q2: the evaluation of each planning graph requires to solve the whole associated satisfiability problem, and intermediate results cannot be reused at subsequent steps.

Let us see how argumentation can address these two issues and let us call this approach ARGOPTPLAN. Let us suppose that ARGOPTPLAN builds the planning graph resulting from a planning problem up to a length l equal or higher than the optimal one[1]. Then, it determines the relevant part of the planning graph, namely the subgraph containing all the paths, also including no-ops, from the initial facts, leading to the goal literals at the higher propositional level (Brafman 2001). The relevant part of the planning graph can thus be encoded as a set of arguments, which are instances of the argument scheme shown before.

From the set of arguments obtained at the step before, ARGOPTPLAN should apply the following rules for determining if an attack is in force between two arguments $\mathbf{A_x}$ and $\mathbf{A_y}$:

1. if either the preconditions, or the effects of two actions are conflicting, or if the effects of one action are incompatible with the preconditions of another action, and they should be executed at the same time, then the arguments supporting these two actions are mutually conflicting ($\mathbf{A_x}$ attacks $\mathbf{A_y}$ if $\mathbf{A_x} \neq \mathbf{A_y}$ and $t_{\mathbf{A_x}} = t_{\mathbf{A_y}}$ and ($\mathcal{P}_{\mathbf{A_x}} \cup \mathcal{P}_{\mathbf{A_y}}$ is contradictory or $\mathcal{C}_{\mathbf{A_x}} \cup \mathcal{C}_{\mathbf{A_y}}$ is contradictory or $\mathcal{C}_{\mathbf{A_x}} \cup \mathcal{P}_{\mathbf{A_y}}$ is contradictory)); or

[1] The value of l can be derived through efficient existing techniques like (Gerevini, Saetti, & Vallati 2011).

2. if the effects of an action are incompatible with the preconditions of another action that should be executed at the subsequent level, then the argument considering the first action has to attack the argument in favour of the second action ($\mathbf{A_x}$ attacks $\mathbf{A_y}$ if $\mathbf{A_x} \neq \mathbf{A_y}$ and $t_{\mathbf{A_x}} = t_{\mathbf{A_y}} - 1$ and $\mathcal{C}_{\mathbf{A_x}} \cup \mathcal{P}_{\mathbf{A_y}}$ is contradictory).

If an attack between two arguments holds according to conditions of the point 1 above, then the actions suggested by these arguments are mutually exclusive according to (Blum & Furst 1997)'s terminology (viceversa not necessarily holds). Moreover, conditions of point 2 deal with the case where an action at a given level prevent the execution of an action at a subsequent level, and this is not considered in (Blum & Furst 1997) as its mutual exclusion relationships are between propositional nodes, whileARGOPTPLAN considers conflicts among arguments each of which supports the execution of a specific action at a specific time.

Given the set of arguments and attacks, ARGOPTPLAN can then derive a (Dung 1995)'s argumentation framework (AF) $\langle \mathcal{A}, \rightarrow \rangle$, where $\mathcal{A}$ is a set of arguments, and $\rightarrow \subseteq \mathcal{A} \times \mathcal{A}$ is an attack relation. An AF is representable as a direct graph where the nodes are the arguments, and the edges are the attacks. In order to address both the issues highlighted before, let us consider a recently new semantics (Baroni, Giacomin, & Guida 2005) called $CF2$.

The idea is that (i) the AF is partitioned into its SCCs thus forming a partial order. Then (ii) the initial SCCs are considered and the maximal conflict-free[2] sets on them are computed. For each possible choice determined at (ii), the nodes attacked within subsequent SCCs are suppressed (iii). Steps (i) to (iii) are then applied recursively on the restricted AFs obtained at (iii).

More formally, given an argumentation framework $AF = \langle \mathcal{A}, \rightarrow \rangle$, let $\mathcal{E}_{CF2}(AF)$ the set of $CF2$ extensions of AF; $\mathcal{MCF}(AF)$ be the set of maximal conflict-free sets of AF; SCCS_{AF} be the set of strongly connected components of AF; for any $E, S \subseteq \mathcal{A}$, $UP_{AF}(S, E) = \{\mathbf{A_x} \in S \mid \nexists \mathbf{A_y} \in E : \mathbf{A_y} \notin S, \mathbf{A_y} \rightarrow \mathbf{A_x}\}$, and $AF{\downarrow}_{UP_{AF}(S,E)}$ be the restriction[3] of AF to $UP_{AF}(S, E)$. Then, a set $E \subseteq \mathcal{A}$ is an extension of $CF2$ semantics, i.e. $E \in \mathcal{E}_{CF2}(AF)$, if and only if:

- $E \in \mathcal{MCF}(AF)$ if $|\text{SCCS}_{AF}| = 1$
- $\forall S \in \text{SCCS}_{AF}$ $(E \cap S) \in \mathcal{E}_{CF2}(AF{\downarrow}_{UP_{AF}(S,E)})$ otherwise.

We can here sketch the prove that given a planning graph built up to a level l, if exists a makespan optimal plan P of length $n \leq l$, then the arguments associated to the actions in P are altogether in at least one of the $CF2$ extensions generated from the relevant part of the planning graph of length l. Indeed we know that at each level of the planning graph each action can be executed only if its preconditions are satisfied either from other actions at the previous level, or from no-ops. Therefore, only the sequences of actions not conflicting each other need to be identified. Since ARGOPTPLAN, as described before, considers all the incompatibilities among preconditions and postconditions, there is a bijective correspondence between incompatibilities among actions, and attacks among derived arguments. From this we can infer that the set of derived arguments of the sequence of actions composing a plan is always conflict-free, and obviously this set of derived arguments is a subset of a maximal conflict-free set. From (Baroni, Giacomin, & Guida 2005) we have that each $CF2$ extension is a maximal conflict-free set, therefore we need to prove that the set of derived arguments on the optimal plan is contained in a $CF2$ extension. By construction, the AF derived by ARGOPTPLAN consists of more than one SCCs: in particular ARGOPTPLAN starts computing the extensions from the set of arguments derived from the first level of the planning graph, identifying the maximal conflict-free sets in it. These maximal conflict-free sets clearly represent the actions that can be executed at the beginning of a plan. Then, incrementally, the attacks from the argument in these maximal conflict-free sets against arguments outside them sets are considered, and they are enlarged by the arguments that are unattacked. This recursive process ends only when no additional argument can be added to the conflict-free sets. Therefore, a optimal solution plan has to be included in a $CF2$ semantics by construction. This prove that a makespan optimal solution plan is derivable from the set of $CF2$ extensions. In particular, it is represented by those extensions where actions making true the goal's literals are suggested to be executed at the same earliest time. This addresses Q1: indeed, ARGOPTPLAN does not need to iteratively build several planning graph each time with higher bound limit, since it can find the optimal solution and demonstrate its optimality, even if it is not of the same length of the planning graph at hand.

Moreover, an argumentation framework built in such a way enjoys favourable computational characteristics. Indeed, it will have a strong directionality (the attacks will be between arguments supporting actions at the same time, or from arguments supporting actions at a specific time against arguments at immediately subsequent time, but never the viceversa) and it will never be composed by a single SCC. Therefore, we do not need to consider the whole argumentation framework at a glance (which is what SatPlan does when it transform the whole planning graph into satisfiability formulae), while we can build the extensions incrementally using memoization techniques. This provides an enhancement in the direction of overcoming Q2.

Conclusions

The aim of this position paper is far from proposing a complete planning system and comparing it with the state-of-the-art approaches. Rather it is to show that looking at a planning problem from an argumentation-based point of view can open a wide spread of research directions that may affect both the computational side of automated planning, and the representation and computation of complex problems.

As a very preliminary case of study, in this short paper we briefly discussed how argumentation could be used for ameliorating the underlying idea of SatPlan optimal planner,

[2] Given a generic AF $\langle \mathcal{A}, \rightarrow \rangle$: $S \subseteq \mathcal{A}$ is *conflict-free* if $\nexists \mathbf{A_x}, \mathbf{A_y} \in S$ s.t. $\mathbf{A_x} \rightarrow \mathbf{A_y}$.

[3] The *restriction* of AF to $S \subseteq \mathcal{A}$ is the argumentation framework $AF{\downarrow}_S = \langle S, \rightarrow \cap (S \times S) \rangle$.

namely the need of building incrementally several planning graphs, and the fact that it requires to find a satisfiability assignment of the whole set of formulae each time. From the previous discussion, it seems that an approach like ARGOPTPLAN may reuse already computed partial solutions, and that it could just require to build a single planning graph rather than severals. This is just a preliminary idea, far from being unquestionable and clearly we are working for implementing ARGOPTPLAN and testing whether or not it is more efficient than other optimal planner.

The idea of using argumentation theory for computing plans is not original, as it has been considered in (García, García, & Simari 2008; Pardo *et al.* 2011; Amgoud, Devred, & Lagasquie-Schiex 2011): the first considers partial order planning only, the second addresses the problem of building a plan in a cooperative way among a set of agents through the cooperation of a set of agents, while the third is focused on the computation of possible intentions of an agent. The proposed approach, instead, shows a direct correspondence between planning specific structures, the planning graph, and argumentation semantics, the $CF2$ semantics, and how this can overcome weakness in existing approaches.

As mentioned in the introduction, argumentation-based practical reasoning may lead to improvements in the field of planning with uncertainty. Concerning the second type of improvement we may obtain, the most relevant are clearly in the context of planning with uncertainty. This relatively young research topic has considered three cases of uncertainty (Ghallab, Nau, & Traverso 2004): partial observability (i.e. different states may be indistinguishable), nondeterminism (i.e. actions with non deterministic effects), and extended goals (i.e. goals with priorities). The usage of argumentation-based approaches for dealing with the above cases may be fruitful. Indeed, as to the first kind of uncertainty, while adopting a probabilistic approach this would lead to an infinite search space, dealing with it by a non-monotonic logic approach like (Reiter 1980; Dung 1995) or an evolution of argument scheme for encompassing *defaults* and/or *assumptions* would allow to do certain reasoning from uncertain and defeasible premises that further knowledge may invalidate. Moreover, the second kind of uncertainty is generally analysed through a probabilistic approach assigning different values to the various effects of the action. In this context, several attempts in argumentation theory, e.g. (Hunter 2012), are aimed at representing probabilistic knowledge and at reasoning with it. Finally, preferences in argumentation for practical reasoning is currently one of the most active research topic in argumentation theory and practical reasoning community, e.g. (Modgil 2009).

References

Amgoud, L.; Devred, C.; and Lagasquie-Schiex, M. 2011. Generating possible intentions with constrained argumentation systems. *Int. J. of Approx. Reason.* 52(9):1363–1391.

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

Baroni, P.; Giacomin, M.; and Guida, G. 2005. SCC-recursiveness: a general schema for argumentation semantics. *Artif. Intell.* 168(1-2):165–210.

Blum, A., and Furst, M. L. 1997. Fast planning through planning graph analysis. *Artif. Intell.* 90:281–300.

Brafman, R. 2001. On reachability, relevance, and resolution in the planning as satisfiability approach. *JAIR* 14:1–82.

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

García, D. R.; García, A. J.; and Simari, G. R. 2008. Defeasible reasoning and partial order planning. In *FoLKS'08*, 311–328.

Gerevini, A.; Saetti, A.; and Vallati, M. 2011. Exploiting macro-actions and predicting plan length in planning as satisfiability. In *AI*IA-2011*, 189–200. Springer.

Ghallab, M.; Nau, D.; and Traverso, P. 2004. *Automated Planning: Theory & Practice*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Girle, R.; Hitchcock, D. L.; McBurney, P.; and Verheij, B. 2003. Decision support for practical reasoning: A theoretical and computational perspective. In Reed, C., and Norman, T. J., eds., *Argumentation Machines. New Frontiers in Argument and Computation*, 55–84.

Hunter, A. 2012. Some foundations for probabilistic abstract argumentation. In Verheij, B.; Szeider, S.; and Woltran, S., eds., *COMMA*, volume 245, 117–128. IOS Press.

Kautz, H., and Selman, B. 1992. Planning as satisfiability. In *ECAI*, 359–363. AAAI Press.

Kautz, A. H., and Selman, B. 1999. Unifying sat-based and graph-based planning. In *IJCAI*. AAAI Press.

McCarthy, J., and Hayes, P. J. 1987. In Ginsberg, M. L., ed., *Readings in nonmonotonic reasoning*. Morgan Kaufmann Publishers Inc. chapter Some philosophical problems from the standpoint of artificial intelligence, 26–45.

Modgil, S. 2009. Reasoning about preferences in argumentation frameworks. *Artif. Intell.* 173(9-10):901 – 934.

Pardo, P.; Pajares, S.; Onaindia, E.; Godo, L.; and Dellunde, P. 2011. Multiagent argumentation for cooperative planning in DeLP-POP. In *AAMAS-2011*, 971–978.

Prakken, H., and Vreeswijk, G. A. 2002. Logics for defeasible argumentation. In Gabbay, D. M., and Guenthner, F., eds., *Handbook of Philosophical Logic, second edition, vol. 4*. Dordrecht etc. 219–318.

Reiter, R. 1980. A logic for default reasoning. *Artif. Intell.* 13(1–2):81–132.

Walton, D. N.; Chris, R.; and Macagno, F. 2008. *Argumentation Schemes*. Cambridge University Press, NY.