

Detecting Commitment Abandonment by Monitoring Sub-Optimal Steps during Plan Execution

(Extended Abstract)

Ramon Fraga Pereira
Pontifical Catholic University of
Rio Grande do Sul (PUCRS)
Porto Alegre, Brazil
ramon.pereira@acad.pucrs.br

Nir Oren
University of Aberdeen
Aberdeen, United Kingdom
n.oren@abdn.ac.uk

Felipe Meneguzzi
Pontifical Catholic University of
Rio Grande do Sul (PUCRS)
Porto Alegre, Brazil
felipe.meneguzzi@pucrs.br

CCS Concepts

•Computing Methodologies → Intelligent Agents;

Keywords

Commitments; Goals; Plan Execution; Abandonment

1. INTRODUCTION

Assessing whether an agent is actually executing steps towards a goal, or has abandoned it, is important when multiple agents are trying to achieve joint goals, or when agents commit to achieving goals for each other. Making such inference for a single goal by observing only plan traces is not a trivial task because agents often deviate from the optimal plans for various reasons, including the pursuit of multiple goals or the inability to act optimally. In this paper, we develop a domain-independent approach based on planning techniques that addresses the problem of detecting sub-optimal steps and determining whether an agent will honour a commitment. Specifically, we use domain-independent heuristics, landmarks (properties or actions that cannot be avoided to achieve a goal from an initial state), and fact partitions to make inferences about goal achievability. Unlike [3, 4], Geib and Goldman, who use plan libraries for goal and plan abandonment detection, we require minimal domain knowledge to detect abandonment. While Kafali *et al.* [8] approach to reason about commitment and goal achievement in an environment use a depth-first search algorithm in GOSU, we use the more flexible domain independent planning algorithms. Finally, Kafali and Yolum's [9] PISAGOR use operational rules to monitor commitments with deadlines and decide whether an observed agent is progressing well towards a commitment, complementing our work.

2. MONITORING SUB-OPTIMAL STEPS

To detect sub-optimal steps, we use a plan optimality monitoring approach [14] that combines planning techniques, *i.e.*, domain-independent heuristics [11, 2, 6, 12] to analyse

plan execution deviation, and landmarks [7] to predict upcoming actions in the plan execution. To analyse deviation, our approach uses domain-independent heuristics, that estimate the distance to a goal G for every state resulting from the execution of an observed action. Given a state s , a heuristic h returns an estimated distance $h(s)$ to G . If an observation o_i results in a state s_i , we consider a deviation from a plan to occur if $h(s_{i-1}) < h(s_i)$. Our approach uses landmarks to predict upcoming actions during execution and obtain information about way-points to achieve a monitored goal G . To predict which actions could be executed in the next observation, it estimates the distance to the extracted landmarks \mathcal{L} closest to the current state. Namely, for every fact landmark $l \in \mathcal{L}$ in which the estimated distance is 0, we select those actions $a \in \mathcal{A}$ such that $l \in pre(a)$; and for every fact landmark $l \in \mathcal{L}$ in which the estimated distance is 1, we select those actions $a \in \mathcal{A}$ such that $pre(a) \in current\ state$ and $l \in eff(a)^+$. These predicted actions may reduce the distance to the monitored goal and next landmarks. By combining these two approaches, Pereira *et al.* [14] developed an approach that detects sub-optimal steps (actions) in observation sequence O . Here, an observed action o is considered sub-optimal if $o \notin$ set of predicted actions **and** $(h(s_{i-1}) < h(s_i))$.

In some planning domains, predicates may provide additional information that can be extracted by analysing preconditions and effects in operator definition. Thus, Pattison and Long [13] classify facts into mutually exclusive partitions to reason about whether certain observations are likely to be goals for goal recognition. We use this classification to infer whether certain observations are consistent with a particular goal. A *Strictly Activating* fact appears as an operator's precondition, and does not appear as an add or delete effect in any operator definition, *i.e.*, unless defined in the initial state, this fact can never be added or deleted by an operator. An *Unstable Activating* fact appears as both a precondition and a delete effect in two operator definitions, so once deleted, this fact cannot be re-achieved, preventing goals that depend on it to be achieved. A *Strictly Terminal* fact does not appear as a precondition of any operator definition, and once added, cannot be deleted. For some planning domains, this kind of fact is the most likely to be in the set of goal facts, because once added in the current state, it remains true until the final state. Evidence of such predicates can help determine goal achievability.

3. COMMITMENT ABANDONMENT

Commitments have been used in multi-agent systems to enable autonomous agents to communicate and coordinate successfully to achieve a particular goal [10, 17, 1]. A commitment $C(\text{DEBTOR}, \text{CREDITOR}, \text{antecedent}, \text{consequent})$ formalizes that the agent **DEBTOR** commits to agent **CREDITOR** to bring about the **consequent** if the **antecedent** holds. The antecedent and consequent conditions are conjunctions or disjunctions of events and possibly other commitments. We aim to monitor the **DEBTOR**'s behaviour (*i.e.*, sequence of actions) and detect **DEBTOR** is individually committed to carrying out a plan to achieve the consequent for the **CREDITOR**.

We define commitment abandonment as a situation in which an agent switches from executing the actions of one plan that achieves the consequent it is committed to, by executing actions from another plan. This plan may achieve other goals, including the consequent of other commitments. Actions in a plan that do not contribute to achieve the consequent of a commitment may indicate that the debtor agent is likely to abandon the commitment that it is committed to its creditor. In this paper, we take inspiration in earlier work [1, 10] that connects commitments to planning, so the domain definition Ξ represents the environment where agents can interact and act, *i.e.*, Σ is set of environment properties and \mathcal{A} is a set of available actions. Now, consider a commitment $C(\text{DEBTOR}, \text{CREDITOR}, \text{At}, \text{Ct})$: in order for a **DEBTOR** to achieve the consequent **Ct** from the antecedent **At** we define that: the antecedent **At** must be in the initial state \mathcal{I} , *i.e.*, $\text{At} \subseteq \mathcal{I}$; and the consequent **Ct** is the goal G . Thus, a plan π for $C(\text{DEBTOR}, \text{CREDITOR}, \text{At}, \text{Ct})$ is a sequence of actions $[a_1, a_2, \dots, a_n]$ (where $a_i \in \mathcal{A}$) that modifies the state $\text{At} \subseteq \mathcal{I}$ into one where **Ct** holds by the successive (ordered) execution of actions in a plan π .

To decide if a debtor agent will abandon a commitment, we observe the sequence of actions it executes, which should achieve a consequent from an antecedent. When a **DEBTOR** settles to commit to an agent **CREDITOR** to bring about the **consequent** of a commitment, the **DEBTOR** should individually commit to achieving such a **consequent** state, and to achieve such state, the **DEBTOR** has to execute a plan. Since an observer does not know **DEBTOR**'s internal state, and consequently to what plan it has committed to, we need to determine which plan, from multiple optimal plans, **DEBTOR** is pursuing. Definition 1 formalises an individual commitment from an observer's point of view.

DEFINITION 1 (Individual Commitment). *Given a set of plans, a DEBTOR agent is individually committed to a plan π if, given a sequence of observations o_1, \dots, o_m : i) $o_k \in \pi$ where $(1 \leq k \leq m)$; and ii) if $o_k = a_j$, then $\forall i = 1 \dots j - 1$, $a_i \in O$ and a_i occurs before a_{i+1} in O . An observation o_p does not contribute to achieve a consequent **Ct** if the **DEBTOR** agent is committed to plan π and: $o_p \notin \pi$; or if $o_p = a_j$, a_k has not yet been observed where $k < j$.*

Using the notion of an individual commitment, we formally define a commitment abandonment problem over a planning theory in terms of a large enough deviation from such an individual commitment in Definition 2. Note that with Definition 1, we can now think of deviations from observations that constitute a strict sub-sequence of any optimal plan that start with the initial state, allowing an agent to infer abandonment at any point in a partial plan execution.

DEFINITION 2 (Commitment Abandonment). *A commitment abandonment problem is a tuple $CA = \langle \Xi, C, \mathcal{I},$*

Algorithm 1 Detecting Commitment Abandonment.

Input: $\Xi = \langle \Sigma, \mathcal{A} \rangle$ planning domain, At antecedent condition ($\text{At} \subseteq \mathcal{I}$), Ct consequent condition, \mathcal{I} initial state, O observation sequence, and θ threshold.

```

1: function HASABANDONED( $\Xi, \text{At}, \text{Ct}, \mathcal{I}, O, \theta$ )
2:    $\langle F_{sa}, F_{ua}, F_{st} \rangle \leftarrow \text{PARTITIONFACTS}(\Sigma, \mathcal{A})$ 
3:   if  $F_{sa} \cap (\text{At} \subseteq \mathcal{I}) = \emptyset$  then
4:     return true  $\triangleright$  Ct is no longer possible.
5:   for each observed action  $o$  in  $O$  do
6:      $\delta \leftarrow \delta.\text{APPLY}(o)$ 
7:     if  $(F_{ua} \cup F_{st}) \subseteq (\delta)$  then
8:       return true  $\triangleright$  Ct is no longer possible.
9:    $A_{\text{SubOptimal}} \leftarrow \text{DETECTSUBOPTIMALSTEPS}(\Xi, \mathcal{I}, \text{Ct}, O)$ 
10:  if  $A_{\text{SubOptimal}} > (\theta * |O|)$  then
11:    return true  $\triangleright$  Debtor abandoned commitment.
12:  return false  $\triangleright$  Debtor may still be committed.
```

O, θ), in which Ξ is a planning domain definition; C is the commitment, in which $C(\text{DEBTOR}, \text{CREDITOR}, \text{At}, \text{Ct})$, **DEBTOR** is the debtor, **CREDITOR** is the creditor, **At** is the antecedent condition, and **Ct** is the consequent; \mathcal{I} as the initial state (*s.t.*, $\text{At} \subseteq \mathcal{I}$); O is an observation sequence of the plan execution with each observation $o_i \in O$ being an action from domain definition Ξ ; and θ is a threshold that represents the percentage of actions in an observation sequence that do not contribute to achieving **Ct** from $\text{At} \subseteq \mathcal{I}$ in which the **DEBTOR** can execute in O .

The solution for this problem is whether an observation sequence O has deviated more than θ from the optimal plan to achieve the consequent **Ct** of commitment c . Algorithm 1 formalizes our approach to solve a commitment abandonment problem, bring together the techniques described in Section 2 (function **DETECTSUBOPTIMALSTEPS**). The algorithm takes as input a CA tuple and returns whether a commitment has been abandoned, based on whether one of the following occurs during plan execution: (1) if *Strictly Activating* facts that we extracted are not in the initial state (Line 3); (2) if we observe the evidence of any *Unstable Activating* and *Strictly Terminal* facts during the execution of actions in the observations (Line 7); or (3) if the number of sub-optimal steps are greater than the threshold θ (*i.e.*, the percentage of actions away from optimal execution that the creditor allows the debtor to deviate in achieving the consequent state) defined by the creditor (Line 9). If none of these conditions hold, the debtor is considered to remain committed to achieving the consequent state of the commitment. Note that in condition (2), the presence of predicates from two of the fact partitions can determine that the monitored goal (or consequent) is unreachable.

4. CONCLUSIONS

In this paper, we developed an approach for detecting commitment abandonment that uses a plan optimality monitoring approach and planning techniques. As future work, we intend to explore partial observability (*i.e.*, missing observations), interleaving plans, more modern heuristics [5, 15], and explore other planning techniques, such as symmetries in planning [16].

Acknowledgments: This research was carried out in cooperation with HP Brazil using incentives of the Brazilian Informatics Law (# 8.248 of 1991).

REFERENCES

- [1] M. Baldoni, C. Baroglio, F. Capuzzimati, and R. Micalizio. Programming with commitments and goals in JaCaMo+. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1705–1706, 2015.
- [2] B. Bonet and H. Geffner. Planning as heuristic search. *Journal of Artificial Intelligence Research (JAIR)*, 129(1-2):5–33, 2001.
- [3] C. W. Geib. Problems with Intent Recognition for Elder Care. In *Proceedings of the Eighteenth AAAI Conference on Artificial Intelligence*, pages 13–17, 2002.
- [4] C. W. Geib and R. P. Goldman. Recognizing Plan/Goal Abandonment. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1515–1517, 2003.
- [5] M. Helmert and C. Domshlak. Landmarks, Critical Paths and Abstractions: What’s the Difference Anyway? In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, 2009.
- [6] J. Hoffmann and B. Nebel. The FF Planning System: Fast Plan Generation Through Heuristic Search. *Journal of Artificial Intelligence Research (JAIR)*, 14(1):253–302, May 2001.
- [7] J. Hoffmann, J. Porteous, and L. Sebastia. Ordered Landmarks in Planning. *Journal of Artificial Intelligence Research (JAIR)*, 22(1):215–278, Nov. 2004.
- [8] Ö. Kafali, A. Günay, and P. Yolum. GOSU: omputing GOal SUPport with Commitments in Multiagent Systems. In *Proceedings of the Twenty-First European Conference on Artificial Intelligence (ECAI)*, pages 477–482, 2014.
- [9] Ö. Kafali and P. Yolum. PISAGOR: a proactive software agent for monitoring interactions. *Knowledge and Information Systems*, 47(1):215–239, 2016.
- [10] F. Meneguzzi, P. R. Telang, and M. P. Singh. A First-Order Formalization of Commitments and Goals for Planning. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, pages 697–703, 2013.
- [11] X. Nguyen and S. Kambhampati. Extracting Effective and Admissible State Space Heuristics from the Planning Graph. In *Proceedings of the Fourteenth AAAI Conference on Artificial Intelligence*, pages 798–805, 2000.
- [12] X. Nguyen, S. Kambhampati, and R. S. Nigenda. Planning graph as the basis for deriving heuristics for plan synthesis by state space and CSP search. *Artificial Intelligence*, 135(1-2):73–123, 2002.
- [13] D. Pattison and D. Long. Domain Independent Goal Recognition. In *STAIRS*, volume 222 of *Frontiers in Artificial Intelligence and Applications*, pages 238–250. IOS Press, 2010.
- [14] R. F. Pereira, N. Oren, and F. Meneguzzi. Monitoring Plan Optimality using Landmarks and Domain-Independent Heuristics. In *The AAAI 2017 Workshop on Plan, Activity, and Intent Recognition*, 2017.
- [15] S. Scherrer, F. Pommerening, and M. Wehrle. Improved Pattern Selection for PDB Heuristics in Classical Planning (Extended Abstract). In *Proceedings of the Eighth Annual Symposium on Combinatorial Search, SOCS 2015*, 2015.
- [16] A. Shleyfman, M. Katz, M. Helmert, S. Sievers, and M. Wehrle. Heuristics and symmetries in classical planning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 3371–3377, 2015.
- [17] P. Telang, F. Meneguzzi, and M. Singh. Hierarchical Planning about Goals and Commitments. In *Proceedings of the Twelfth International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 877–884, 2013.