# An Alternative Approach for Reasoning about the Goal-Plan Tree Problem

**Patricia Shaw**[1] and **Rafael H. Bordini**[2]

## 1 INTRODUCTION

BDI agents have goals to achieve and a library of plans that can be used to achieve them, typically requiring adopting further goals. A Goal-Plan Tree (GPT) structure can be used to naturally represent the goals of BDI agents with the required plans and subgoals to achieve them. These can be used to significantly improve an agent's *deliberation* and ability to make reasoned decisions on plan selection and whether to commit to achieving a new goal. In previous work, a Petri net based approach for reasoning about goal-plan trees was defined. This paper outlines an alternative approach for performing the reasoning using constraint logic programming.

In work by Thangarajah *et al.* [8, 9, 10], a GPT is defined to represent the structure of various plans and subgoals related to each goal for an individual agent. At each node in the tree, *summary information* is used to represent the various constraints under consideration. However, the amount of summary information could potentially grow exponentially with the size of the GPT [2], which could have a significant impact on an agent's performance for larger problems. To this end, a different approach was introduced by Shaw and Bordini [5], mapping a GPT into a Petri net in such a way as to avoid the need for summary information for reasoning about positive and negative interactions between goals.

In [6], the focus is on reasoning about resources using Petri nets, which are then *combined* into a coherent reasoning process encompassing the reasoning about positive and negative interactions from [5]. In this paper, we outline an alternative approach to reasoning about positive [9, 3, 5], negative [8, 1, 5] and resource [10, 4, 6] interactions using a constraint logic programming approach developed in GNU Prolog to define a set of constraints that are solved to generate a successful execution ordering of the plans to achieve the goals.

## 2 CONSTRAINT-BASED APPROACH

A GPT consists of a top-level goal at the root, with one or more alternative plans available to achieve that goal. Each of these plans may themselves include further subgoals forming the next level in the tree, followed by additional plans to achieve these subgoals. An example of a goal-plan tree is shown in Figure 1, showing a simplistic soil sample collection goal for a Mars Rover. An agent generally has multiple top-level goals each with its own GPT. While these could be achieved sequentially, there are often benefits to be gained from achieving them in parallel. This can of course lead to problems where goals interfere or where resources are limited, so reasoning is required for the agent to be successful or to be more efficient. Where

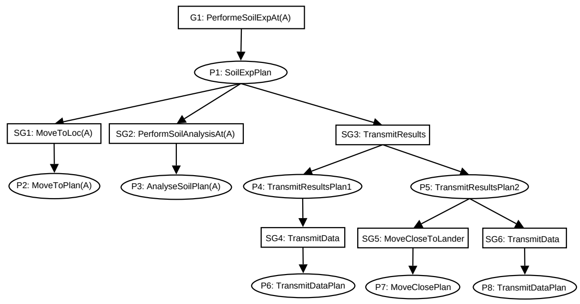



**Figure 1.** Goal-plan tree for a Mars rover as used by Thangarajah *et al.* The goals and subgoals are represented by rectangles while the plans are represented by ovals.

a goal or subgoal has multiple plan options, only one of these need to be executed for the goal to be achieved, however all the subgoals of a plan need to be achieved for the plan to be successful. Appropriate selection of plan options where available can help improve an agents performance, particularly where resources are highly constrained. An overview of the constraint-based approach for performing the reasoning using the GPT is presented here. Further details can be found in [7].

Within the constraint-based definition of the GPT, the goals, plans and their subgoals are listed as node predicates as shown below, where node `g1` is a top level goal with a plan node `p1` to achieve it. `p1` has two subgoals that have to be achieved, `sg2` and `sg3`. Subgoals are represented in the same style as the goal node, listing the plans that can be used to achieve the subgoal. The last three lists in the plan node represent the preconditions, effects and resource requirements of the plan. In this example, `p1` has no preconditions for execution, but it will require 1 unit of resource `r1` and cause the effect of assigning the environment variable `e3` to 7.

```
% Goal node
   node(g1,[p1]).
% Plan node
   node(p1,[sg2,sg3],[],[e3/7],[r1/1]).
```

An evaluation of the constraint-based model gives a partial ordering to the plans indicating a sequence in which they could be executed to avoid interference and to make best use of opportunities from combining plans for goals, as well as making best use of available resources.

[1] University of Durham, UK, email: p.h.shaw@durham.ac.uk
[2] Federal University of Rio Grande do Sul, Brazil, email: r.bordini@inf.ufrgs.br

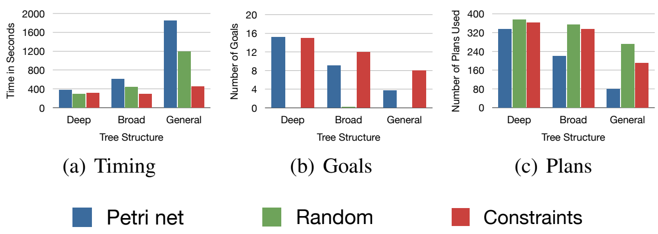
**Resource Reasoning** There are two main classes of resources: reusable resources and consumable resources. An instance of a reusable resource can only be used by one plan at a given time, but when that plan has finished executing, the resource is available again for another plan to use it, for example a communication channel. On the other hand, consumable resources can only be used once, and then no longer exist, for example (units of) energy. This reasoning only considers *consumable* resources. The amount of each resource available at the start is given as a predicate, "`resource(r1,50).`". The amount of each resource for each goal is calculated based on taking plans with minimum resource requirements where there is a choice and from this summary (e.g. `S = [r1/7, r2/5, r3/6, r4/0, r5/0]`), the goals are sorted into either increasing or decreasing order of resource requirements. Working through the list, achievable goals are selected until there are no longer sufficient resources available. Remaining goals and associated plans are then removed from consideration.

**Goal Interaction** Conflicts can arise within a single agent when it has taken on two or more goals that are not entirely compatible. This can often be avoided by scheduling the plan execution so as to protect the causal links until they are no longer required. Causal links exist where one plan has caused an effect that is the precondition of another plan. If a plan changing this effect was allowed to executed between the two linked plans this would cause conflict, potentially leading to a goal failing if the effect could not be reproduced. In the constraint model, these linked plans are identified, along with any plans that could interfere with them. Restrictions are then placed on the ordering of these plans to ensure the interfering plan does not execute between the linked plans.

Conversely, two or more goals may have plans that achieve the same effect(s). As the effect only needs to be achieved once, only one of the plans needs to be executed. This also applies to the sub-tree of the plan as the subgoals all form part of the process for achieving the goal. Selecting the plan with the lower resource requirements also aids with the number of goals that can potentially be achieved. In the constraint model, the plans achieving common effects between goals are identified from the top down within the GPTs. A selection is then made from the choice of plans based on resource requirements of their respective sub-trees, with the lower cost plan being kept while the other plans and sub-trees are removed from consideration, thereby reducing the overall number of plans, resources and time that are required to achieve the feasible goals. It should be noted that by making this reduction, it is sometimes feasible to achieve more goals with the same quantity of resources, therefore this part of the reasoning is evaluated before the resource reasoning calculates the resource requirements of top-level goal.

## 3 EXPERIMENTS AND CONCLUSIONS

Detailed evaluation of this approach, along with a comparison of the approach to the Petri net model are presented in [7]. Here a brief summary of the constraint-based results are presented with comparison to the Petri net results. When evaluating performance, the three types of reasoning were evaluated separately as well as combined together under varying conditioned including tree structure (deep vs. broad and a general mid-point), tree size, levels of interaction and availability of resources amongst others. The aim of the experiments was to stress test the approaches to identify subclasses of the reasoning problem where the model was most suited. The results below show one set of experiments with all three types of reasoning in three different tree structures. A 'Random' model was introduced, which randomly started goals and plans as a base case for comparison. As can be seen from figure 2(b), the constraints model performs well when there is greater branching in the tree structure. The constraint model was also more scalable, in its ability to continue to reason about a larger number of goals than the Petri net model.



(a) Timing (b) Goals (c) Plans

**Figure 2.** Comparison results for combined reasoning across the three tree structures under highly constrained conditions.

The reasoning about resources that has been considered here has focused on that of consumable resources that are limited in their availability. Another type of resource that is often used are reusable resources, such as communication channels. A model was shown in [6] of how this could be incorporated into the Petri net model, with further work needed to extend the constraint-based approach. However, further work to extend both approaches to allow for more generic maintenance goals where consumable resources could be regenerated, as well as achievement goals is required.

## REFERENCES

[1] B.J. Clement and E.H. Durfee, 'Theory for coordinating concurrent hierarchical planning agents using summary information', in *AAAI '99: Proc. of the 16th conf. on AI*, pp. 495–502, (1999).
[2] B.J. Clement and E.H. Durfee, 'Performance of coordinating concurrent hierarchical planning agents using summary information', in *proc. of 4th ICMAS*, pp. 373–374, (2000)
[3] J.F. Horty and M.E. Pollack, 'Evaluating new options in the context of existing plans', *AI*, **127**(2), 199–220, (2004).
[4] A. Raja and V. Lesser, 'Reasoning about coordination costs in resource-bounded multi-agent systems', *proc. of AAAI 2004 Spring Symposium on Bridging the multiagent and multi robotic research gap*, 25–40, (2004).
[5] P. Shaw and R.H. Bordini, 'Towards alternative approaches to reasoning about goals', in *Proc. of the 5th Int. Workshop on Declarative Agent Languages and Technologies*, vol. 4897/2008 of *LNCS*, pp. 104–121, (2008).
[6] P. Shaw, B. Farwer, and R.H. Bordini, 'Theoretical and experimental results on the goal-plan tree problem', *Proc. of the 7th AAMAS*, **3**, 1379–1382, (2008).
[7] P. Shaw, *Reasoning about Goal-Plan Trees in Autonomous Agents*, Ph.D. dissertation, Department of Computer Science, Durham University, (2010).
[8] J. Thangarajah, L. Padgham, and M. Winikoff, 'Detecting and avoiding interference between goals in intelligent agents', in *proc. of 18th IJCAI*, pp. 721–726, (2003).
[9] J. Thangarajah, L. Padgham, and M. Winikoff, 'Detecting and exploiting positive goal interaction in intelligent agents', in *Proc. of the 2nd AAMAS*, pp. 401–408, (2003).
[10] J. Thangarajah, M. Winikoff, and L. Padgham, 'Avoiding resource conflicts in intelligent agents', in *proc. of 15th ECAI*, pp. 18–22, (2002).