Optimally Relaxing Partial-Order Plans with MaxSAT

Christian Muise and **Sheila A. McIlraith**
Dept. of Computer Science
University of Toronto
Toronto, Canada. M5S 3G4
{cjmuise,sheila}@cs.toronto.edu

J. Christopher Beck
Dept. of Mechanical & Industrial Engineering
University of Toronto
Toronto, Canada. M5S 3G8
jcb@mie.utoronto.ca

Abstract

Partial-order plans (POPs) are attractive because of their least commitment nature, providing enhanced plan flexibility at execution time relative to sequential plans. Despite the appeal of POPs, most of the recent research on automated plan generation has focused on sequential plans. In this paper we examine the task of POP generation by relaxing or modifying the action orderings of a plan to optimize for plan criteria that promotes flexibility in the POP. Our approach relies on a novel partial weighted MaxSAT encoding of a plan that supports the minimization of deordering or reordering of actions. We further extend the classical least commitment criterion for a POP to consider the number of actions in a solution, and provide an encoding to achieve least commitment plans with respect to this criterion. We compare the effectiveness of our approach to a previous approach for POP generation via sequential-plan relaxation. Our results show that while the previous approach is proficient at heuristically finding the optimal *deordering* of a plan, our approach gains greater flexibility with the optimal *reordering*.

1 Introduction

Partial-order planning reflects a least commitment strategy (Weld 1994). Unlike a sequential plan that specifies a set of actions and a total order over those actions, a partial-order plan (POP) only specifies those action orderings necessary and sufficient to achieve the goal. In doing so, a POP embodies a family of sequential plans – a set of linearizations all sharing the same actions, but differing with respect to the order of the actions. The flexibility afforded by POPs makes them attractive for real-time execution, multi-agent taskability, and a range of other applications (Veloso, Pollack, and Cox 1998; Weld 1994). Nevertheless, in recent years research on plan generation has shifted away from partial-order planning towards sequential planning, primarily due to the success of heuristic-based forward-search planners. To regain the least commitment nature of POPs while leveraging fast sequential plan generation, it is compelling to examine the computation of POPs via sequential planning technology. Indeed this approach has been realized in the planner POPF (Coles et al. 2010), which generates a POP by searching in a heuristic-based forward-chaining manner.

Another possibility for leveraging the strengths of sequential planning is to generate a sequential plan with a state-of-the-art planner, and subsequently relax the plan. Removing ordering constraints from the sequential plan, referred to as a *deordering*, or allowing changes in the ordering constraints, referred to as a *reordering*, are approaches that have been theoretically investigated (Bäckström 1998). Unfortunately, finding the optimal deordering or reordering is NP-hard, and difficult to approximate within a constant factor.

We focus on the problem of computing the minimum deordering and minimum reordering of a sequential plan treated as a POP. These notions cover a natural aspect of least commitment planning – minimizing the ordering constraints placed on a plan. We extend this characterization to consider the number of actions in a plan. Our approach is to use a family of novel encodings for partial weighted MaxSAT whose solution corresponds to an optimal least commitment plan. Unlike typical SAT-based planning techniques, we represent an action occurrence once, giving us a succinct representation. We empirically compare our approach to an existing algorithm for relaxing a sequential plan due to Kambhampati and Kedar (1994). We find that the existing algorithm is extremely proficient at computing the minimum deordering, matching the optimal solution in every problem tested. However, we find that the minimum reordering is usually far more flexible than the minimum deordering (having fewer ordering constraints and far more linearizations). We further see a benefit in the flexibility of a POP when we consider the number of actions. Our approach gives us the first technique, to the best of our knowledge, for computing the optimal reordering of a POP.

2 Least Commitment Criteria

Partial-Order Plans We restrict our attention to STRIPS planning problems, and adopt the following notation: $PRE(a)$, $ADD(a)$, $DEL(a)$ to signify the preconditions, add effects, and delete effects of the action a; **adders**(f) (resp. **deleters**(f)) to signify the set of actions that add (resp. delete) the fluent f; a_I (resp. a_G) is a special action that has the effect of the initial state (resp. the preconditions of the goal state). A POP, P, is the tuple $\langle \mathcal{A}, \mathcal{O} \rangle$ where $\mathcal{A}$ is the set of actions in the plan (including a_I and a_G), and $\mathcal{O}$ is the set of ordering constraints over the actions $\mathcal{A}$ (e.g., $(a_1 \prec a_2) \in \mathcal{O}$), assumed to be transitively closed. We refer to a total ordering of the actions in $\mathcal{A}$ that respects $\mathcal{O}$ as a *linearization* of P. A POP provides a compact representation for multiple linearizations and it is *valid* iff every linearization is an executable sequential plan (note that we handle the goal through a_G). An action a_i *supports* action a_j if there is an ordering between them, and a_i adds a fluent that is in a_j's precondition list. A sufficient condition for POP validity is that every action has support for all its preconditions and no support is threatened by an intervening action that could delete the precondition. For further details, we refer the reader to (Russell and Norvig 2009).

Our aim of least commitment planning is to find flexible solutions that allow us to defer ordering decisions until the execution of the plan. Considering only the ordering constraints of a POP, two appealing notions for least commitment planning include *deordering* and *reordering* (Bäckström 1998). A reordering of a POP $P = \langle \mathcal{A}, \mathcal{O} \rangle$ is any POP $P' = \langle \mathcal{A}^{'}, \mathcal{O}^{'} \rangle$ where $\mathcal{A} = \mathcal{A}'$. A deordering of a POP P is any reordering $P' = \langle \mathcal{A}^{'}, \mathcal{O}^{'} \rangle$ such that $\mathcal{O}' \subseteq \mathcal{O}$. *Minimal* deorderings (resp. reorderings) are those where no proper subset of the ordering constraints is also a deordering (resp. reordering). *Minimum* deorderings (resp. reorderings) are those where no set of ordering constraints with smaller cardinality yields a valid POP. Finding the minimum deordering or reordering is NP-hard, and cannot be approximated within a constant factor unless $NP \in$ DTIME($n^{poly\ log\ n}$) (Bäckström 1998).

While the notion of a minimum deordering or reordering addresses the commitment to ordering constraints, in the spirit of least commitment we would like to commit to as few actions as possible. We provide an extended criterion of what a least commitment POP (LCP) should satisfy:

Definition 1. Let $P = \langle \mathcal{A}, \mathcal{O} \rangle$ and $Q = \langle \mathcal{A}^{'}, \mathcal{O}^{'} \rangle$ be two valid POPs. Q is a *least commitment POP* (LCP) of P iff Q is the minimum reordering of itself and there is no valid POP $\langle \mathcal{A}^{''}, \mathcal{O}^{''} \rangle$ such that $\mathcal{A}'' \subseteq \mathcal{A}$ and $|\mathcal{A}''| < |\mathcal{A}'|$.

Intuitively, we can compute the LCP of an arbitrary POP by first minimizing the number of actions, and then minimizing the number of ordering constraints. It may turn out that preferring fewer actions causes us to commit to more ordering constraints due to the interaction between the chosen actions. However, in practice we usually place a greater emphasis on minimizing the number of actions, as every action must be executed in the standard interpretation of a POP.

We evaluate the quality of a POP by the number of actions and ordering constraints it contains, as these metrics give us a direct measure of the least commitment nature of a POP. Another property of interest is a POP's *flexibility*; a measure of the robustness inherent in the POP. We measure the flexibility, whenever computationally feasible, as the number of linearizations in the POP. The linearizations serve as a natural measure of the number of ways an agent can execute the plan. Verifying a POP's validity by way of the linearizations is not always practical. As such, we do not attempt to compute POPs that maximize the number of linearizations, but rather compute POPs that adhere to one of the above criteria: minimum deordering, minimum reordering, or LCP.

3 A Partial Weighted MaxSAT Encoding

We encode the task of finding a minimum deordering or reordering as a partial weighted MaxSAT problem (Biere et al. 2009). Solutions to the default encoding correspond to a LCP. That is, no POP exists with a proper subset of the actions, or with a proper subset of the ordering constraints. We add further clauses to produce encodings that correspond to optimal deorderings or reorderings. In contrast to the typical SAT encoding for a planning problem, we do not require the actions to be placed in a particular layer. Instead, we represent each action only once and reason about the ordering between actions. The actions in the encoding come from a provided sequential plan.

We use three types of propositional variables: $\forall a \in \mathcal{A}$, x_a indicates a appears in the POP P; $\forall a_1, a_2 \in \mathcal{A}$, $\kappa(a_1, a_2)$ indicates a_1 occurs before a_2 in P; $\forall a_i \in \mathcal{A}, \forall p \in$ PRE$(a_i), \forall a_j \in$ **adders**(p), $\Upsilon(a_j, a_i, p)$ indicates a_j supports a_i with the fluent p in P.

In a partial weighted MaxSAT encoding there is a distinction between hard and soft clauses. We first present the hard clauses of the encoding, and then describe the soft clauses. We define the formulae that ensures the POP generated is acyclic, and the ordering constraints produced include the transitive closure. Here, actions are universally quantified, and for formula (4) we assume $a_I \neq a_i \neq a_G$.

$$(\neg\kappa(a, a)) \tag{1}$$

$$(x_{a_I}) \wedge (x_{a_G}) \tag{2}$$

$$\kappa(a_i, a_j) \rightarrow x_{a_i} \wedge x_{a_j} \tag{3}$$

$$x_{a_i} \rightarrow \kappa(a_I, a_i) \wedge \kappa(a_i, a_G) \tag{4}$$

$$\kappa(a_i, a_j) \wedge \kappa(a_j, a_k) \rightarrow \kappa(a_i, a_k) \tag{5}$$

(1) ensures that there are no self-loops; (2) ensures that we include the initial and goal actions; (3) ensures that if we use an ordering variable, we include both actions in the POP; (4) ensures that an action cannot appear before the initial action or after the goal; and (5) enforces the transitive closure of ordering constraints. Together, (1) and (5) ensure the POP will be acyclic, while the remaining formulae connects the action occurrence and ordering variables.

Finally, we use formulae to ensure that every action has its preconditions met, and there are no threats in the solution:

$$\Upsilon(a_j, a_i, p) \rightarrow \bigwedge_{a_k \in \textbf{deleters}(p)} x_{a_k} \rightarrow \kappa(a_k, a_j) \vee \kappa(a_i, a_k) \tag{6}$$

$$x_{a_i} \rightarrow \bigwedge_{p \in PRE(a_i)} \bigvee_{a_j \in \textbf{adders}(p)} \kappa(a_j, a_i) \wedge \Upsilon(a_j, a_i, p) \tag{7}$$

(6) ensures that if a_j is the achiever of precondition p for a_i, then no deleter of p will be allowed to occur between the actions a_j and a_i. (7) ensures that if we include action a_i in the POP, then every precondition p of a_i (the conjunction) must be satisfied by at least one achiever a_j (the disjunction). $\kappa(a_j, a_i)$ orders the achievers correctly, while $\Upsilon(a_j, a_i, p)$ removes threats.

To generate a POP that is least commitment, we prefer solutions that first minimize the actions, and then minimize the ordering constraints. We achieve this by adding a soft unit clause for the negation of every action and ordering variable in our encoding. We weight the negated κ variables with a cost of 1 and weight the negated action variables $a \in \mathcal{A}$ with a cost of $1 + |\mathcal{A}|^2$. A violation of any one of the unit clauses means that the solution includes the action or ordering constraint corresponding to the violated clause's variable.

An optimal solution corresponds to a LCP. Observe that (1)-(7) make no mention of a sequential plan. If our sequential plan is $[a_0, \cdots, a_k]$, we introduce the following to compute minimum deorderings or reorderings:

$$(x_a) \qquad \forall a \in \mathcal{A} \tag{8}$$

$$(\neg\kappa(a_j, a_i)) \qquad 0 \leq i < j \leq k \tag{9}$$

(8) forces every action to be present, while (9) forces the orderings to agree with the original sequential plan. A solution to formulae (1)-(8) corresponds to a reordering, while a solution to formulae (1)-(9) corresponds to a deordering. We refer to the soft constraints along with the encodings (1)-(7), (1)-(8), and (1)-(9) as *LCP*, *MR*, and *MD* respectively.

4 Evaluation

We evaluate the effectiveness of using the partial weighted MaxSAT solver, Sat4j (Le Berre and Parrain 2010), to optimally relax a plan using our proposed encodings. To measure the quality of the POPs we generate, we consider the number of actions, ordering constraints, and linearizations (whenever feasible to compute). Further, we investigate the effectiveness of a method due to Kambhampati and Kedar (1994) that produces a deordering of a plan in polynomial time (denoted KK). For our analysis, we use six domains from the International Planning Competition (IPC): depots, driverlog (driver), logistics, tpp, rovers, and zenotravel (zeno). We chose domains that allow for partial order solutions – many IPC domains are overly constrained and only yield sequential plans (e.g., Sokoban). In such cases, relaxation is not possible and the solver trivially finds the sequential plan we begin with. We conducted the experiments on a Linux desktop with a 2.5GHz processor. Each run of Sat4j was limited to 10 minutes and 2GB of memory. We generated an initial sequential plan by using the FF planner (Hoffmann and Nebel 2001) (we tried multiple planners and found the results to be statistically no better than FF). In the following evaluation, we only report on the problems where FF was able to find a plan within a 30-minute timeout.

POP Quality We begin by examining the relative quality of the POPs produced with different optimization criteria (LCP, MR, and MD), as well as KK. We report the number of actions and ordering constraints in the transitive closure of the generated POP. The number of actions for KK, MR, and MD are equal to those in the sequential plan, so we report the value only for KK and LCP. Table 1 shows the results for all six domains on the problems for which every approach succeeded (98 of the 130 successfully encoded problems).

There are a few items of interest to point out. First, columns 4 and 5 coincide perfectly. Perhaps surprisingly, KK is able to produce the optimal deordering in every case,

Domain	$\lvert\mathcal{A}\rvert$		$\lvert\mathcal{O}\rvert$			
	KK	LCP	KK	MD	MR	LCP
depots (14/22)	34.9	31.0	473.4	473.4	430.9	341.5
driver (15/16)	27.5	26.5	332.6	332.6	326.9	297.3
logistics (30/35)	78.1	77.4	1490.6	1490.6	1462.5	1470.4
tpp (5/30)	13.4	13.4	74.8	74.8	74.8	74.8
rovers (18/20)	31.1	30.3	223.2	223.2	217.6	204.2
zeno (16/20)	29.2	29.2	404.3	404.3	403.5	403.5
ALL (98/143)	44.3	43.2	685.7	685.7	669.0	651.6

Table 1: Mean number of actions and ordering constraints for the various approaches. Numbers next to the domain indicate the number of instances solved by all methods (and included in the mean).

even though it is not guaranteed to do so. Second, we see the number of ordering constraints for the LCP approach is greater than that for the MR approach (on average) in the logistics domain. POPs in the logistics domain require more ordering constraints for a solution with slightly fewer actions. Third, the low coverage in the tpp domain is due to allowing reordering – MD succeeds in 20 problems while LCP and MR succeed in only 5.

In general, the LCP has fewer actions and ordering constraints than the optimal reordering, which in turn has fewer ordering constraints than the optimal deordering. If the LCP has the same number of actions as the sequential plan, then the LCP and minimum reordering coincide. We can see this effect in tpp and zenotravel.

Encoding Difficulty To measure the difficulty of solving the encoded problems, we show the number of problems solved by Sat4j as a function of time (including encoding phase) in Figure 1. Sat4j consistently produced a feasible solution almost immediately. While we do not consider it here, early solutions serve as approximations to the optimal plan and are valid POPs (Sat4j finds the optimal solution if given sufficient time and memory). For comparison, we additionally include the aggregate time for KK. When Sat4j had difficulty, it was due to the number of transitivity clauses included (cubic in the number of actions).

We find that MD is generally easier to solve and the majority of the problems are readily handled by Sat4j: 62% solved in under 5 seconds. Being a polynomial algorithm, KK consistently finds a solution faster than any encoding.

Reordering Flexibility To further evaluate the flexibility of the optimal deordering, we compare the number of linearizations it induces with the number of linearizations the optimal reordering induces. We found that of the 78 problems we could successfully compute the linearizations for, approximately 40% of the problems exhibited a difference between the optimal deordering and optimal reordering. Figure 2 shows the number of linearizations for the

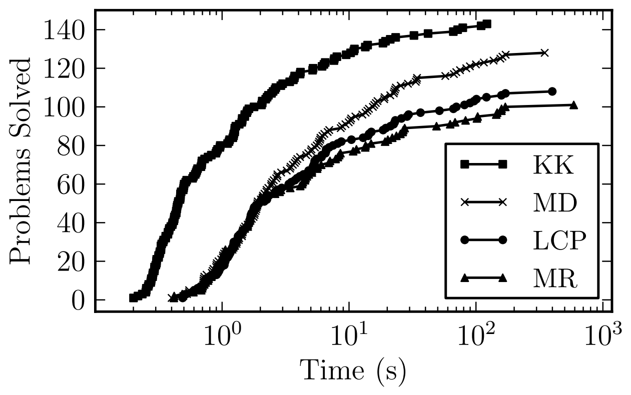

Figure 1: The number of problems solved by Sat4j if given a limited amount of time per problem.

optimal reordering divided by the number of linearizations for the optimal deordering. For readability, we omit the 47 instances where the linearizations matched.

The ratio of linearizations ranges from 0.9 (an anomaly discussed below) to over two million. While KK is proficient at finding the optimal deordering, there are still significant gains in terms of flexibility by using the optimal reordering.

5 Discussion

The results paint an overall picture of how the optimization criteria compare to one another. We find that the KK is extremely adept at finding the optimal deordering, despite its lack of theoretical guarantee. In contrast, in many of the domains we see gains in terms of the number of actions or ordering constraints in the POP if we compute the optimal reordering or a least commitment POP. The single ratio under 1 in Figure 2 occurs due to the rare case in which two valid POPs with an equal number of actions and ordering constraints have a different number of linearizations.

The standard SAT-based planning encodings also produce a POP, but a significant difference from our work is that we avoid encoding an action for every layer in a planning graph by appealing to the fact that we already know the (superset of) actions in the solution. The core of our encoding is similar to Variant-II of Robinson *et al.* (2010) and the causal encodings of (Kautz, McAllester, and Selman 1996). We similarly encode the ordering between any pair of actions as a variable ($\kappa(a_i, a_j)$), but rather than encoding a relaxed planning graph or every potential action occurrence, we encode the formulae that must hold for a valid POP on the specific set of actions we start with. There are also similarities between our work and that of Do and Kambhampati (2003). In particular, the optimization criteria for minimizing the number of ordering constraints coincide, as does the optional use of constraints to force a deordering. However, while Do and Kambhampati focus on temporal relaxation in the context of action ordering, we take the orthogonal view of minimizing the number of actions.

It is natural to consider the impact that the choice of initial plan has on the final POP. We found that starting from an existing POP from a planner such as POPF or Blackbox had little impact on the quality of the optimally relaxed POP.

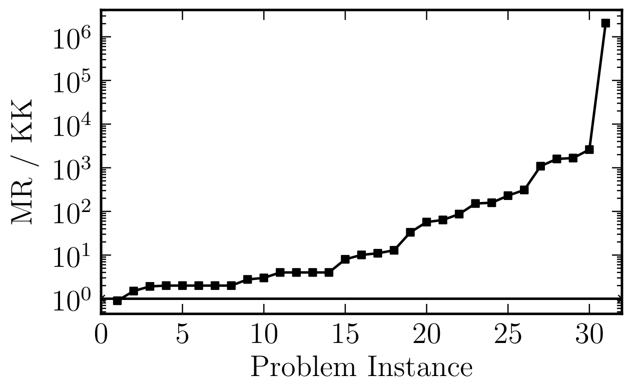

Figure 2: Ratio of Linearizations. The y-axis represents the number of linearizations induced by the POP for the optimal reordering divided by the number of linearizations induced by the POP for the optimal deordering. The x-axis ranges over all problems where the number of linearizations differed (~40%), and is sorted based on the y-axis value.

Our approach significantly improved the quality of the initial POP with respect to our proposed criteria, but the final solution was quite similar to the optimal POP produced with FF plans as input. The question remains open, however, on how to best compute an initial set of actions for our encoding. We also hope to investigate versions of partial weighted MaxSAT solvers tailored to problems in which only unit clauses are soft (as is the case with our encoding).

6 Conclusion

In this paper we proposed a practical method for computing the optimal deordering and reordering of a sequential plan. Despite the theoretical complexity of computing the optimal deordering or reordering, we are able to compute the optimal solution by leveraging the power of modern MaxSAT solvers. We further proposed an extension to the classical least commitment criterion that considers the number of actions in a solution and demonstrated the added flexibility of a POP that satisfies this criterion.

Our approach uses a family of novel encodings for partial weighted MaxSAT where a solution corresponds to a POP satisfying one of the three least commitment criteria. We found that the majority of problems are readily handled by the MaxSAT solver, Sat4j, but also found that two domains presented a problem for the encoding phase of our approach.

We also investigated an existing polynomial algorithm for deordering sequential plans, and discovered that it successfully computes the optimal deordering in every problem we tested despite its lack of theoretical guarantee. Since the algorithm is fast in practice, it is well suited for relaxing a POP if we require a deordering. If a reordering or least commitment POP is desired, then we can produce a far more flexible POP by using one of the proposed encodings.

Acknowledgements

The authors gratefully acknowledge funding from the Ontario Ministry of Innovation and the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

Bäckström, C. 1998. Computational aspects of reordering plans. *Journal of Artificial Intelligence Research (JAIR)* 9(1):99–137.

Biere, A.; Heule, M.; van Maaren, H.; and Walsh, T. 2009. *Handbook of Satisfiability, Frontiers in Artificial Intelligence and Applications*. IOS Press, Amsterdam.

Coles, A.; Coles, A.; Fox, M.; and Long, D. 2010. Forward-chaining partial-order planning. In *Proceedings of the 20th International Conference on Automated Planning and Scheduling (ICAPS)*.

Do, M., and Kambhampati, S. 2003. Improving the temporal flexibility of position constrained metric temporal plans. In *AIPS Workshop on Planning in Temporal Domains*.

Hoffmann, J., and Nebel, B. 2001. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research (JAIR)* 14(1):253–302.

Kambhampati, S., and Kedar, S. 1994. A unified framework for explanation-based generalization of partially ordered and partially instantiated plans. *Artificial Intelligence (AIJ)* 67(1):29–70.

Kautz, H.; McAllester, D.; and Selman, B. 1996. Encoding plans in propositional logic. In *Proceedings of the 5th International Conference on the Principles of Knowledge Representation and Reasoning (KR)*, 374–385.

Le Berre, D., and Parrain, A. 2010. The sat4j library, release 2.2 system description. *Journal on Satisfiability, Boolean Modeling and Computation (JSAT)* 7:59–64.

Robinson, N.; Gretton, C.; Pham, D. N.; and Sattar, A. 2010. Partial weighted MaxSAT for optimal planning. In *Proceedings of the 11th Pacific Rim International Conference on Artificial Intelligence (PRICAI)*.

Russell, S., and Norvig, P. 2009. *Artificial intelligence: A modern approach*. Prentice hall.

Veloso, M.; Pollack, M.; and Cox, M. 1998. Rationale-based monitoring for planning in dynamic environments. In *Proceedings of the 4th International Conference on Artificial Intelligence Planning Systems (AIPS)*, 171–179.

Weld, D. 1994. An introduction to least commitment planning. *AI Magazine* 15(4):27.