On the Hybrid Propositional Plan Encodings *

Amol D. Mali
Dept. of Electr. Engg. & Computer Science, P.O.Box 784
University of Wisconsin, Milwaukee, WI 53201, USA.
mali@miller.cs.uwm.edu, Phone: 414 - 906 - 8942
Fax: 414 - 229 - 2769

Abstract

(Appears in the proceedings of the international conference on Artificial Intelligence and Soft omputing (ASC), Canada, July 2000, pp. 334-342.)

Classical planning is the problem of synthesizing a sequence of actions to reach a goal state starting from an initial state. Recently, classical planning problems were shown to be solvable faster by casting them as satisfiability (SAT). This work is an answer to the important and unaddressed challenge of developing hybrid propositional plan encodings and evaluating their effectiveness, proposed in [3]. We combine key ideas from state space planning and causal planning to develop all hybrid plan encodings, in the planning as satisfiability framework. Our theoretical analysis shows that the hybrid encodings have strictly higher size (number of clauses, sum of the clause lengths and variables) than the smallest state space encoding. Our empirical evaluation shows that hybrid encodings are not easier to solve than the smallest state space encoding. State space encodings are attractive because of their smaller size and the causal encodings are attractive because of their flexibility in reordering steps (which is valuable in incremental planning like plan reuse and plan merging). We show that hybrid encodings have a higher size and a lower flexibility in step reordering, and thus do not combine the best of both the worlds (state space plan encodings and causal plan encodings).

Keywords: Planning, Satisfiability, Propositional reasoning, Constraint satisfaction.

1 Introduction

Classical planning is the problem of synthesizing an executable sequence of actions to reach a partially specified goal state starting from a completely specified initial state, assuming deterministic actions, perfect perception and a static and perfectly observable environment. An action in STRIPS representation has pre-conditions which are atoms (positive literals) and effects which are atoms (positive and negative literals), classified into add effects and delete effects. For example, to execute the action $load(A, R_1, London)$ (loading package A into rocket R_1 at London), the pre-conditions $at(A, London), has\_fuel(R_1)$ and $at(R_1, London)$ must be true and after this action is executed, $in(A, R_1)$ will become true (add effect) and $at(A, London)$ will become false (delete effect), assuming that the action is defined in such a way that a package that is loaded is no longer in the city.

The traditional classical planners (also known as "split & prune" planners) did refinement search [3], starting with an empty plan and refining it by adding more constraints like actions and orderings between these. Based on whether the search is conducted in the space of world states or space of partial plans (sets of constraints about orderings and causal sources of truths of pre-conditions of actions), the planners are broadly classified as "state space" planners and "partial order" planners. More encouraging results were obtained by casting planning as propositional satisfiability [6]. The general idea of this paradigm is to construct a disjunctive structure that contains all action sequences of length k, some of which may be plans. The problem of checking if there exists a plan is posed as satisfiability testing. The SAT instance (encoding of the planning problem) contains constraints that must hold for any specific action sequence to be a solution (a plan). The encoding represents the relevant planning constraints (e.g. action choices, pre-conditions and effects of the actions, mutual exclusivity relations between the actions etc.) by fixing the number of plan steps (k) such that the encoding has a model if and only if a plan of k steps exists. If no model is found, a new encoding is generated by increasing the value of k.

This success [6] generated an interest in developing

* I thank Subbarao Kambhampati for useful comments on the previous draft of this paper. I also thank Henry Kautz, Bart Selman and David McAllester for comments and clarification on planning as satisfiability and providing code for experiments.

different (preferably smaller) propositional encodings of planning problems, as shown by recent attempts like [5] and [1]. These encodings are of interest because of their different sizes (measured in terms of the number of clauses, sum of the clause lengths and the number of variables) that are generally correlated with the ease of finding their models.

If the conjunction of a k length action sequence and a k step encoding is satisfiable, then the action sequence is a plan. Thus encodings are viewed as means of proving plan correctness [7]. The current implementations of planning as satisfiability [1],[6], [2] and several others use only the state space encodings. Kautz et al [5] have reported an encoding based on the causal planning (also known as "partial order" planning or "plan space" planning). This encoding has been empirically shown to be harder to solve [7] and both theoretically and empirically shown to be strictly larger than the smallest encoding (the state space encoding with explanatory frame axioms, reported in [5]). Thus the state space and causal encodings have been viewed to be radically different.

A challenge to synthesize hybrid encodings and analyzing their potential has been posed in [3]. A key difference between state space planning and causal planning is that the state of world is represented at each time step during the state space planning process and it is never represented during the causal planning process. We can bridge these extremes by controlling the number of time steps at which the world state is represented. We show that all the current encodings are subsumed by our hybrid encoding called the *unifying encoding* which is based on the notion of controlling the localization in the interactions among the plan steps. Kambhampati & Srivastava [4] have synthesized a planner that unifies the state space and causal approaches to planning. However, currently no unifying propositional plan encodings exist. We make the following contributions -

- Synthesis of several hybrid encodings and a comparison of their asymptotic sizes with the current encodings (in the non-incremental domain independent planning, where plans are synthesized from scratch).

- An adaptation of the current and hybrid encodings to a generalized incremental planning, the specific problem of plan completion and a comparison of their asymptotic sizes.

- A comparison of the asymptotic sizes of the encodings (current and hybrid) with a restriction on a particular potential non-minimality in the plans found.

- An empirical evaluation of several instances of the unifying encoding for domain independent planning which shows that the hybrid instances are neither smaller nor easier to solve, than the smallest encoding.

- We show that the hybrid encodings do not combine the best of both worlds (smallest size of the state space encoding with explanatory axioms and the flexibility of causal encoding in allowing steps to be reordered) and thus do not look promising.

The paper is organized as follows. We explain the basics of the state space and causal encodings from [5] in section 2. We then explain how the current encodings differ and introduce the notion of the unifying encoding (in section 3). We also explain in this section, the notation and terminology used to describe the constraints in the encodings, show how the unifying encoding can be automatically generated and report its asymptotic size. In section 4, we discuss other hybrid encodings and show that none of them is smaller than the currently smallest encoding. In section 5, we discuss how the state space, causal and unifying encodings can handle the generalized incremental planning and the specific problem of plan completion and report their asymptotic sizes. In this section, we also compare the sizes of these encodings when they are augmented with constraints to control a certain type of non-minimality in the plans found, in incremental planning. In section 6, we discuss our empirical results and show why the hybrid encodings do not look promising in the non-incremental and incremental planning (like plan merging and plan reuse). We report conclusions in section 7.

2 Background

In this section, we provide the necessary basics of the encodings from [5], starting with some notation.

2.1 Notation

p_i denotes a plan step and o_j denotes a ground executable action. k is the number of steps in an encoding and U is the set of ground pre-condition and effect propositions u_j in the domain. $u_j(t)$ denotes that u_j is true at time t. ϕ denotes the null action (no-op) that does not cause any change to the world state. O is the set of non-null ground actions in the domain. $(p_i = o_j)$ denotes the step→action mapping. $p_i \xrightarrow{f} p_j$ denotes the causal link where p_i adds the condition f (makes f true (is the contributor)), p_j needs (is the consumer) it and p_i precedes p_j. $Adds(p_i, u_j)$ denotes that p_i has the effect of making u_j true. $Needs(p_i, u_s)$ denotes that u_s is a pre-condition of p_i. $Dels(p_i, u_q)$ denotes that p_i has the effect of making u_q false. $p_i * p_j$ denotes the contiguity of the two steps and thus no step can occur between p_i and p_j. $p_i \prec p_j$ denotes the temporal precedence relation between the two steps. A "threat" is said to exist if the pre-condition f of a step p_i is made true by a step p_j, $p_j \prec p_i$ and there is a step p_s, such that $p_j \prec p_s, p_s \prec p_i$ and p_s deletes f. $o_i(t)$ denotes that the action o_i occurs at time t.

2.2 State space encoding

To prove that a sequence of actions is a plan, state space methods essentially try to progress the initial state I (or regress the goal state G) through the sequence to see if the goal state (or initial state) is reached. The following constraints are represented in the explanatory frame axiom-based state space encoding from [5].

1. The initial state is true at time 0 and the goal must be true at time k. **2.** Conflicting actions (one action deleting the pre-condition or effect of another or needing negation of pre-condition of another) cannot occur at the same time step. **3.** The "explanatory frame axioms" state that if the truth of a fluent changes over the interval $[t, t+1]$, some action changing that must occur at t. **4.** If an action occurs at time t, its pre-conditions are true at t and effects are true at $(t+1)$.

2.3 Causal encoding

The plan space (causal) encodings (based on partial order planning) are based on the ideas of proving the correctness of a plan using causal reasoning about the establishment and preservation of goals and the pre-conditions of individual actions. The following constraints are represented in the causal encoding from [5].

1. Each step in the encoding is bound to a single action or the no-op. This is stated as a disjunctive *step* $\rightarrow$ *action* binding and mutual exclusion constraints. **2.** The conditions true or false in the initial state are the effects of step I and the conditions in the goal state are the preconditions of step F. I has no pre-conditions and it is always the first step in a plan and F has no effects and it is always the last step in a plan. **3.** A step inherits the preconditions and effects of its action binding. **4.** The only way a step can add, delete or need a condition is if the condition is added, deleted or needed (respectively) by its action binding. **5.** Each precondition of each step must have a causal link supporting it. **6.** The contributor step of a causal link precedes the consumer step, and if a step is bound to an action that deletes the condition supported by the causal link (threat), that step either precedes the contributor or succeeds the consumer. **7.** The $\prec$ relation is irreflexive, asymmetric and transitive.

3 Unifying Encoding

In the state space encodings, the closest contributor of the pre-conditions of an action o_i (consumer) is zero time steps behind o_i (since the state at time t also provides the pre-conditions, though the actions that made those pre-conditions true could have occurred at any time step between 0 and $(t-1)$). The closest provider of the pre-condition can be arbitrarily far behind a consumer in the causal encoding. If we control the distance between consumers and closest contributors by varying the number of time steps at which the world state is represented, we can synthesize a series of encodings that contain a partial order on the steps that are "sandwiched" in the "regions" between two consecutive world states, as in the unifying encoding shown in Figure 1. Purely causal and purely state space encodings can be viewed as instances of the unifying encoding with 1 region and k regions respectively, k being the number of steps. The world states eliminate the need to consider all potential interactions between all the steps (like threats and causal links) and we then need to focus only on the local interactions (between the steps in a region).

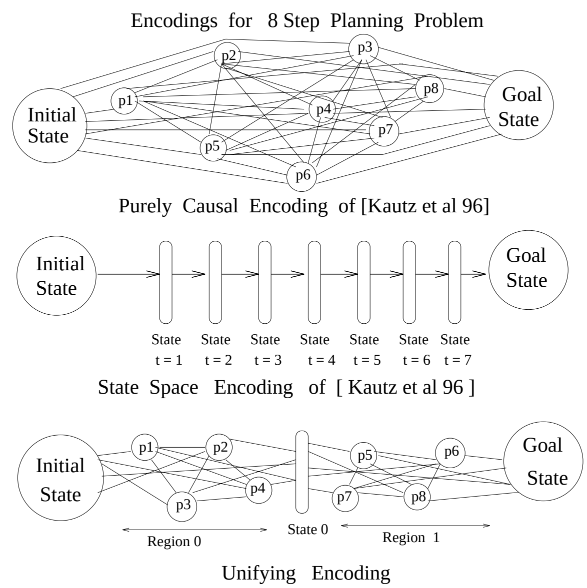

Figure 1: Propositional Encodings of Classical Planning Problems.

We assume that there are p regions in the encoding. Hence there are $(p-1)$ world states between the initial state and the goal state, $1 \leq p \leq k$. The step indices range from 1 to k. The region indices range from 0 to

$(p-1)$ and the immediately preceding state for region 0 is the initial state and the immediately succeeding state for region $(p-1)$ is the goal state. Without loss of generality, we assume that all regions contain equal, that is $m = \frac{k}{p}$ steps. Thus i th region contains steps with indices ranging from $(i*m+1)$ to $(i+1)*m$. $u_q(i) \longrightarrow p_j$ denotes the causal link where u_q is true in state i and remains undeleted till the occurrence of the step p_j which also has u_q in the list of its pre-conditions. $p_i \longrightarrow u_j(q)$ denotes the causal link where u_j is true in state q and undeleted after the occurrence of p_i which also has u_q in the list of of its add effects. $u_i(t) \longrightarrow u_i(t+1)$ denotes the causal link where u_i is true in the states t and $(t+1)$ and not deleted by any step which occurs between these states.

3.1 Schemas for generating the Unifying Encoding

We explain here the schemas required to generate the unifying encoding (only selected parts of important schemas are shown in Figure 2). Though we do not separately describe the basics of purely state space and purely causal encodings from [5], the basics as well as the schemas needed to generate these encodings can be derived from the schemas of the unifying encoding here, since these encodings are just special cases of the unifying encoding. **(1)** Each step must be mapped to some action. **(2)** All propositions in the initial state description are true in the initial state, remaining propositions are false in the initial state. The goal must be true at time p. **(3)** A step inherits the pre-conditions and effects of the action it is mapped to. **(4)** A step adds, deletes or needs a proposition only if it is mapped to an action which adds, deletes or needs that proposition respectively. **(5)** Each pre-condition of each step p_{i*m+j} (this is the j th step in i th region) must be made available either by some other step in the same region as the step p_{i*m+j} or by the immediately preceding state. **(6)** A unifying encoding contains 4 types of causal links, since both the consumer and contributor can be a step or a state (step→state, state→step, step→step and state→state). The threats to these causal links (where either the consumer or the contributor is a step) must be resolved by promotion or demotion (reordering the clobberer). The threats to the causal links from state i to state $(i+1)$ are resolved by demanding that no step in the region should delete the contributed condition (protection method). **(7)** Each of the four types of causal link symbols needs to be translated into constraints on the establisher of the pre-condition and the consumer. **(8)** The state at the end of each region needs to be computed, by computing the truth of each fluent $u_j \in U$ in that state, like the explanatory frame axioms in [5]. A fluent is true in state i (that immediately succeeds the region $(i-1)$) only if it was either true in the state $i-1$ and not made false by any step in the region or if it was made true by some step in the region and not deleted later. Similar explanation holds for the false value of a fluent in state i. **(9)** The relation $\prec$ is transitive, asymmetric and irreflexive.

3.2 Asymptotic Size of the Unifying Encoding

Since each of the k steps may be mapped to any of the $\mid O \mid$ actions from the domain and hence may add, delete or need almost any fluent from U, $O(k*(\mid O \mid + \mid U \mid))$ variables relating the steps to the actions and fluents are needed. (Because of these step→action mapping possibilities, the state space encoding from [5] contains $O(k*(\mid U \mid + \mid O \mid))$ variables and because of the explanatory frame axioms and the fact that an action occurring at time t implies the truth of its pre-conditions at t and effects at $(t+1)$ and mutual exclusions for each pair of conflicting actions for each time step, it contains $O(k*(\mid U \mid + \mid O \mid) + k* \mid O \mid^2)$ clauses.) Variables $(p_i \prec p_j)$ that represent the partial order on the steps are also needed. Since $\prec$ relations are generated only for the steps in the same region and there are p such regions, the number of such $(p_i \prec p_j)$ variables is $O(\frac{k^2}{p})$. Similar argument shows that the number of causal link variables is $O(\frac{k^2}{p} * \mid U \mid)$. Thus the number of variables in the unifying encoding is $O(k*(\mid O \mid + \mid U \mid) + \frac{k^2}{p} * \mid U \mid)$.

Since each of the $O(\frac{k^2}{p^2} * \mid U \mid)$ causal links in a region may be threatened by any of the other $O(\frac{k}{p})$ steps in that region, and there are p such regions, $O(\frac{k^3}{p^2} * \mid U \mid)$ clauses are required to resolve the potential threats. Ignoring the clauses generated by schemas that do not dominate the asymptotic size and considering the clauses needed to specify mutual exclusions that a step cannot be bound to more than one action, the unifying encoding contains $O(\frac{k^3}{p^2} * \mid U \mid + k* \mid O \mid^2)$ clauses. The maximum clause length is a constant and hence the sum of the clause lengths is also $O(\frac{k^3}{p^2} * \mid U \mid)$. The number of clauses and variables is thus minimum when $p = k$ and maximum when $p = 1$. Note that since the number of actions and fluents are constants for a given domain, we use the power of the variable k to compare the sizes of different encodings of a problem.

4 Other Hybrid Encodings

Two key notions in state space planning are (1) World state and (2) Contiguity of steps. Two key notions in causal planning are (1) Partial order on the steps and

$$\mathbf{5}.\quad \wedge_{i=0}^{p-1} \wedge_{j=1}^{m} \wedge_{q=1}^{|U|} (Needs(p_{i*m+j}, u_q) \Rightarrow ((\vee_{s=i*m+1, s\neq(i*m+j)}^{m*(i+1)} (p_s \xrightarrow{u_q} p_{i*m+j})) \vee (u_q(i) \longrightarrow p_{i*m+j})))$$

$$\mathbf{6}.\wedge_{i=0}^{p-1} \wedge_{j_1=1}^{m} \wedge_{j_2=1, j_1\neq j_2}^{m} \wedge_{j_3=1, j_3\neq j_2, j_3\neq j_1}^{m} \wedge_{q=1}^{|U|} (((p_{j_1} \xrightarrow{u_q} p_{j_2}) \wedge Dels(p_{j_3}, u_q)) \Rightarrow ((p_{j_3} \prec p_{j_1}) \vee (p_{j_2} \prec p_{j_3})))$$

$$\wedge_{i=0}^{p-1} \wedge_{j=1}^{|U|} ((u_j(i) \longrightarrow u_j(i+1)) \Rightarrow (\wedge_{q=i*m+1}^{(i+1)*m} \neg Dels(p_q, u_j)))$$

$$\mathbf{7}.\wedge_{i=0}^{p-1} \wedge_{j_1=(i*m+1)}^{(i+1)*m} \wedge_{j_2=(i*m+1), j_1\neq j_2}^{(i+1)*m} \wedge_{q=1}^{|U|} ((p_{j_1} \xrightarrow{u_q} p_{j_2}) \Rightarrow (Adds(p_{j_1}, u_q) \wedge Needs(p_{j_2}, u_q) \wedge (p_{j_1} \prec p_{j_2})))$$

$$\wedge_{i=0}^{p-1} \wedge_{j=1}^{|U|} ((u_j(i) \longrightarrow u_j(i+1)) \Rightarrow (u_j(i) \wedge u_j(i+1)))$$

$$\mathbf{8}.\wedge_{i=1}^{p} \wedge_{j=1}^{|U|} (u_j(i) \Rightarrow ((\vee_{q=(i-1)*m+1}^{i*m} (p_q \longrightarrow u_j(i))) \vee (u_j(i-1) \longrightarrow u_j(i))))$$

Figure 2: Selected parts of important schemas for generating the unifying encoding.

(2) Causal links (or a step must rely on other steps and not the world state, to make its pre-conditions available). To consider all hybridization possibilities, we define an encoding to be hybrid if it contains at least one key notion from state space planning and at least one key notion from causal planning. This view allows our conclusions to be applicable to all hybrid encodings.

We list the complete set of hybridization possibilities (that are soundness and completeness preserving) next. **1.** Contiguity and precedence constraints, only 1 region and no world state representation. One can augment the causal encoding in [5] with the contiguity constraint. One has to then also state the relation between contiguity and partial order to avoid inconsistency. This adds $O(k^2)$ more variables of type $p_i * p_j$ and $O(k^3)$ more clauses to state the semantics of contiguity (that is, if any of the $O(k^2)$ variables of type $p_i * p_j$ is true, none of the remaining $(k-2)$ steps can occur between p_i and p_j). **2.** More than one region. There are several variants of this. These variants have variants of their own. **2.1** No world state representation. **2.1.1** Equal/Unequal distribution of steps among the regions. **2.1.1.1** All steps in a region are partially ordered. **2.1.1.2** All steps in a region are contiguous. **2.1.1.3** Some steps in a region are contiguous, others are partially ordered. **2.2** World state representation present. **2.2.1** Equal/Unequal distribution of steps among the regions. **2.2.1.1** All steps in a region are partially ordered. **2.2.1.2** All steps in a region are contiguous. **2.2.1.3** Some steps in a region are contiguous, others are partially ordered. The variant 2.2.1.1 with an equal distribution of steps among the regions is the unifying encoding in Figure 1.

The following theorem holds since having fewer regions than the number of plan steps requires more clauses and variables to represent the planning constraints and removing the representation of state increases the size further, since all interactions between the steps in region i with the steps in the regions preceding i need to be represented. If steps are not equally distributed among the regions, the size of the encoding worsens (increases).

Theorem 1. No hybrid encoding has fewer variables or fewer clauses or lower sum of the clause lengths than the smallest encoding (state space encoding with explanatory frame axioms), for the non-incremental domain independent planning.

5 Incremental Planning

Satisfiability has been hitherto applied to only non-incremental (purely generative) planning. We adapt the encodings to handle incremental planning (where new actions can be added to an existing plan and existing actions can be removed, to achieve new goal from the new initial state). Just as the number of plan steps is bounded (k) in the non-incremental planning, we bound the number of plan steps in the incremental planning as well ($k+q$). We do not consider the criterion of maximally recycling the old plan. This criterion can be handled as a restriction on the models of the propositional encodings that we synthesize next.

To allow the addition of new actions to a k step plan, we include q new steps in an encoding, so that q new actions can be added (some state space encodings allow parallel actions and in these cases, more than q actions may be added). Note that when comparing the sizes of encodings for incremental planning, we consider the powers of both k and q from the asymptotic expressions. An encoding for incremental planning has three parts - (i) constraints from the old plan (ii) constraints between the new steps and (iii) constraints to capture the interaction between the steps of the old plan and the new steps. The actions from the old plan can be removed by changing the truth assignments of the variables, e.g. an occurrence of o_j can be removed by making $(p_i = o_j)$ false and $(p_i = \phi)$ true. We next discuss the key modifications (or adaptations) that various current encodings need, to handle incremental planning.

5.1 Adapting the State Space Encodings

We discuss two schemes to achieve this.

5.1.1 Scheme 1 We specify that each action o_i from the old plan will either occur at any of the $[0, k+q-1]$ time steps or will never occur (for this case, we create a special variable ϕ'_i) and new actions (from O) may occur at any time step.

5.1.2 Scheme 2 Since the actions from the old plan may have to be removed or reordered to allow the incorporation of new actions at arbitrary places, we not only make k copies of the old plan, but also reserve $(k+1)$ blocks, each of q steps for accomodating new actions (as shown in Figure 3), to simulate in state space planning, the ability of causal planning to allow the inclusion of new steps anywhere as per need. This means that to synthesize a $(k+q)$ step plan, we have $(k^2 + (k+1)*q)$ steps in the encoding. Due to the representation of world state at each time step, the possibility of occurrence of any of the $\mid O \mid$ actions at the $(k+1)*q$ steps reserved for new actions and the explanatory frame axioms, the encoding has $O((k^2+(k+1)*q)* \mid U \mid + (k+1)*q* \mid O \mid)$ variables and the number of clauses is the number of variables plus the term $(k+1).q. \mid O \mid^2$ (for mutual exclusion constraints for pairs of conflicting actions for each of the $(k+1).q$ new steps for allowing the inclusion of new actions).

5.2 Adapting the Unifying Encoding

Since the q new steps may be arbitrarily distributed among the existing p regions of the encoding, we need to include extra steps to account for this distribution. Hence when q new steps are to be added to an existing

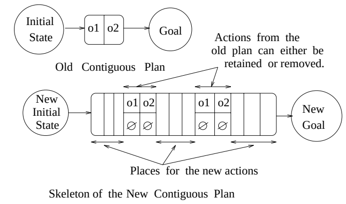

Figure 3: Scheme 2 - Incremental planning with a contiguous plan by making multiple copies of it and reserving multiple blocks of steps for the new actions, $k = 2, q = 3$.

plan, we add $q*p$ new steps to the existing encoding. Since each region in the encoding now has q more steps, there will be $O((\frac{k}{p}+q)^2 * p* \mid U \mid)$ variables (causal links) and $O((\frac{k}{p}+q)^3 * p* \mid U \mid)$ clauses (for resolving the threats). These can be considered as functions $v(p)$ and $c(p)$ respectively.

Solving $\frac{d(v(p))}{dp} = 0$, $\frac{d(c(p))}{dp} = 0$ to find the lowest size, we get $p = \frac{k}{q}$ and $p = \frac{2*k}{q}$ respectively. Let us consider $p = \frac{k}{q}$. Since $1 \le p \le k$ and $p = \frac{k}{q}$, $q \ge 1$ and $q \le k$. Since each action from the old plan remains in the same region, one may worry that the adapted unifying encoding may not have models of $(k+q)$ steps that can be found by reordering the actions from the old plan (this reordering involves moving an action from one region to other). However it should be noted that since each of the $q*p$ new steps may be mapped to any of the $\mid O \mid$ actions, plans of $(k+q)$ steps can still be found, as long as $q*p \ge (k+q)$. This however means that $p \ge (\frac{k}{q}+1)$, different than the minimum. A comparison of the sizes of the encodings is given in Figure 4.

Encoding	# Variables	# Clauses
State space	$O((k+q)*(\mid U \mid + \mid O \mid))$	$O((k+q)*(\mid U \mid + \mid O \mid))$
Causal	$O((k+q)^2* \mid U \mid)$	$O((k+q)^3* \mid U \mid)$
Unifying	$O((\frac{k}{p}+q)^2 * p* \mid U \mid)$	$O((\frac{k}{p}+q)^3 * p* \mid U \mid)$
Unifying $(p = \frac{k}{q})$	$O(k*q* \mid U \mid)$	$O(k*q^2* \mid U \mid)$

Figure 4: Sizes of the encodings for incremental planning. The size of the state space encoding is based on the scheme in section 5.1.1. To get the exact expressions for the number of clauses, the number of clauses needed to specify mutual exclusions between actions need to be added to the expressions in the table, though these are linear in k and q.

5.3 Plan Completion

Plan completion is a specific case of incremental planning, where an incomplete plan is extended to solve the problem, respecting certain constraints from it (the presence of actions and the precedence relations between them). To prevent a violation of these constraints, their conjunction is included in the encoding, so that any model of the encoding will respect these constraints. As shown in Figure 5, a contiguous plan of k steps which is to be completed by adding actions at arbitrary places, can be completed, by solving an encoding in which $(k+1)$ blocks, each long enough to accomodate q new actions (and thus $(k+1)*q$ additional steps) are included. A comparison of the sizes of the encodings is shown in Figure 6. As in section 5.2, we add q new steps to each region of the unifying encoding, to solve the plan completion problem.

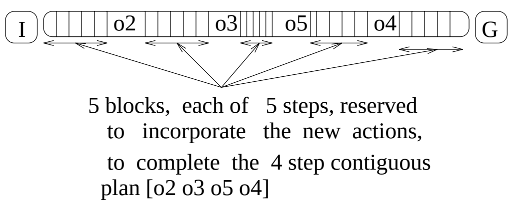

Figure 5: Completion of a contiguous plan.

Encoding	# Variables	# Clauses
State space	$O((k+1)*q*(\mid U\mid + \mid O\mid))$	$O((k+1)*q*(\mid U\mid + \mid O\mid))$
Causal	$O((k+q)^2 * \mid U\mid)$	$O((k+q)^3 * \mid U\mid)$
Unifying ($p=\frac{k}{q}$)	$O(k*q*\mid U\mid)$	$O(k*q^2*\mid U\mid)$

Figure 6: Sizes of the encodings for plan completion. As in Fig. 4, the number of clauses needed to specify mutual exclusions needs to be added to the expressions in the table, to get a more accurate expression for asymptotic number of clauses.

5.4 Plan Non-minimality

Let us consider the generalized incremental planning. If an action occurs s times in the old plan, since q new steps are added, it should certainly not occur more than $(q+s)$ times in the final plan (because actions from the old plan are either kept or removed, the steps from the old plan are not mapped to new non-null actions). Violation of this may lead to non-minimal plans, e.g. repeatedly loading and unloading the same packages. One has to add extra clauses to the state space encodings and the unifying encoding, to control the number of occurrences of an action (unless one decides to remove this non-minimality by post-processing the plans, which is not necessarily computationally cheaper), as shown in Figure 7.

Encoding	# Variables	# Clauses
State space (Scheme 1)	$O((k+q)*(\mid U\mid + \mid O\mid))$	$O((k+q)*(\mid U\mid + \mid O\mid) + (k+q)^{(q+s+1)})$
Causal	$O((k+q)^2 * \mid U\mid)$	$O((k+q)^3 * \mid U\mid)$
Unifying ($p=\frac{k}{q}$)	$O(k*q*\mid U\mid)$	$O(k*q^2*\mid U\mid + (k+p*q)^{(q+s+1)})$
State space (Scheme 2)	$O((k^2+(k+1)*q)*\mid U\mid + (k+1)*q*\mid O\mid)$	$O((k^2+(k+1)*q)*\mid U\mid + k^3 + (k+1)^2*q)$

Figure 7: Sizes of the encodings for the incremental planning, with the restriction on the number of occurrences of an action that occurs s times in the old plan. As in Fig. 4, the number of clauses needed to specify mutual exclusions needs to be added to the expressions in the table, to get a more accurate expression for asymptotic number of clauses.

It is clear that the size of the state space encoding with explanatory frame axioms (scheme 2) then approaches the size of the causal encoding, in presence of these additional constraints. Though k copies of each action o_i from the old plan are made, only one should be used, as stated in the following $O(k^3)$ clauses.

$\wedge_{q_1=1}^{k} \wedge_{i=1}^{k} (o_i(q_1*q+(q_1-1)*k+i-1) \Rightarrow (\wedge_{q_2=1,q_2\neq q_1}^{k} \neg o_i(q_2*q+(q_2-1)*k+i-1)))$. Similar restriction on the new actions requires $O((k+1)^2*q)$ more clauses.

In the state space encoding based on scheme 1 (in section 5.1.1) as well as the unifying encoding, we have to state that $(q+s+1)$ occurrences of an action are not allowed. This makes the space required to store the encodings impractical. Causal encodings do not have this problem. A step in a causal encoding is not bound to more than one action. Exactly s occurrences of o_i are included in the representation of the old plan in the causal encoding and since o_i is either retained or removed and not replaced by some other non-null action, the number of occurrences of the newly introduced actions as well as the old actions are automatically controlled. Causal encodings are naturally compatible with incremental planning, due to the partial order on the steps and need no adaptation like other encodings (addition of extra steps etc.). (State space encodings with explanatory frame axioms allow non-conflicting actions to occur in parallel. One can restrict them with $O(k* \mid O \mid^2)$ clauses to ensure that only one action occurs at a time, however this restriction still does not control the number of occurrences of individual actions.)

6 Discussion

To answer one of the challenges in [3], we developed hybrid encodings. We computed their sizes in both the incremental and the non-incremental planning and showed that the hybrid encodings are not the smallest (have fewer clauses, fewer variables and lower sum of the clause lengths). (We did not report the asymptotic sum of clause lengths in Figures 4, 6 and 7 since these can be derived by multiplying the number of clauses by the maximum clause lengths.)

The sizes of different encodings for domain independent planning differ because of different ways of declaratively specifying the constraints (clauses) that will be true if and only if there is a plan of certain length. Thus a larger encoding does not contain more information that can be propagated to simplify it more. The empirical evaluation [6], [1],[7] and [2] reports that generally smaller encodings can be solved faster.

Our empirical evaluation of various instances of the unifying encoding on several benchmark domains is shown in Figure 8. The number of steps in the encodings were same as the number of actions in the plans. We found that simplifying the encodings by unit propagation (e.g. deriving $(\alpha \wedge \beta)$ from $(\alpha, \alpha \Rightarrow \beta)$ with propagation of the unit clause α) did not have an appreciable impact on the sizes and solving times. The results clearly show that the hybrid encodings were neither smaller nor faster to solve than the smallest encoding. Our expressions for the asymptotic sizes of the instances of the unifying encoding indicate that the ratios of the number of variables and clauses in the encodings of a problem with p_1 and p_2 regions are close to $\frac{p_2}{p_1}$ and $(\frac{p_2}{p_1})^2$ respectively. This is not the case as shown by our results in Figure 8, because of the contributions of terms that do not dominate the asymptotic size. Our implementation did contain the three types of causal link variables $u_i(t) \longrightarrow u_i(t+1)$, $u_q(t) \longrightarrow p_i$ and $p_i \longrightarrow u_q(t)$ in the purely state space encodings, where in fact these links are not needed. Modifying the implementation to avoid the generation of these links considerably lowered both the sizes and solving times of the purely state space encodings.

Many times domain specific or problem specific constraints can be propagated to reduce the size of an encoding and this may make it faster to solve. In [7], it has been shown that the purely causal encodings are strictly larger and harder to solve than the purely state space encodings. However it has been shown in [8] and [9], that in presence of causal domain specific knowledge (like the causal links and precedences in the task reduction schemas in hierarchical planning) or in case of plan merging or plan reuse (where steps may have to be reordered to allow new steps to be incorporated), the domain specific knowledge or the constraints from the plans being merged or reused can be propagated to significantly simplify the causal encodings making them faster to solve than the state space encodings, on some problems. Do these results transfer to hybrid encodings? We examine this next.

The purely causal encodings offer more freedom in step reordering than the hybrid encodings, in incremental planning or the use of domain specific causal knowledge. The hybrid encodings either contain world state or contiguity on some steps, both of which prohibit certain types of reordering of steps. All steps before a world state clearly precede all steps after the world state, limiting the reordering options, opening up the possibilities of incompleteness. To prevent this, one has to add several extra steps to the encoding and this raises the concern that the plans may be not only non-minimal, but also may not show conformance to the user intent. To prevent these problems, one has to either include a large number of constraints in the hybrid encodings, or post-process the plans to remove these problems and iteratively generate and solve the encodings till the desired plans are obtained. Hybrid encodings may be useful in very restricted cases where it is guaranteed that certain steps reorderings are not going to be required for solving the problems.

7 Conclusion

To address one of the challenges in [3], we synthesized a complete set of hybrid encodings and showed that no hybrid encoding can have fewer clauses or fewer variables or lower sum of the clause lengths than the smallest encoding, in the domain independent non-incremental planning. We adapted the encodings to incremental planning and showed that no hybrid encoding is smaller than the smallest encoding. We empirically showed that the hybrid encodings are harder to solve as well. Considering plan merging, plan reuse and use of domain specific causal knowledge, we showed why hybrid encodings do not look promising for these paradigms. The hybrid encodings neither combine the best of both worlds (lowest size of the state space encoding and flexibility in step reordering in the causal encodings) nor preserve an advantage of an either world.

Domain (Steps)	**p = 1**	**p = k/6**	**p = k/4**	**p = k/3**	**p = k/2**	**p = k**
	V, C, T	V, C, T	V, C, T	V, C, T	V, C, T	V, C, T
Blocks (12)	6111	3918	3230	2916	2662	2648
	59703	19857	11975	9093	6967	5867
	16.6	46.98	1.24	1.03	0.45	0.26
Ferry (12)	5598	3605	2983	2699	2469	2455
	53670	18913	12083	9592	7762	6836
	16.95	50.72	0.83	1.15	0.25	0.2
Ferry (18)	6499	3074	-	2294	2094	2074
	99055	16134	-	7605	5924	5049
	*	2.42	-	3.49	0.41	0.28
Ferry (8)	1003	-	690	-	562	552
	6123	-	2408	-	1413	1205
	0.64	-	0.13	-	0.02	0.02
Tsp(12)	5149	3303	2725	2461	2247	2233
	49321	16426	9942	7575	5833	4941
	8.44	5.04	0.52	0.27	0.13	0.1
Tsp(8)	1801	-	1243	-	1019	1009
	10817	-	4208	-	2431	2061
	0.74	-	0.16	-	0.04	0.04
Log(12)	3790	2439	2015	1821	1663	1649
	36574	12380	7614	5876	4598	3942
	27.57	54.69	0.64	0.69	0.27	0.24
Log(6)	696	696	-	508	460	452
	3066	3066	-	1396	1068	914
	0.14	0.14	-	0.05	0.02	0.02
Tireworld(12)	5127	3282	2704	2440	2226	2212
	49431	16424	9914	7537	5787	4889
	10.57	6.02	0.68	0.61	0.28	0.21

Figure 8: Empirical results on various (hybrid) instances of the unifying encoding from Figure 1. $p = 1$ corresponds to the purely causal encoding and $p = k$ corresponds to the purely state space encoding. V, C, T denote the number of variables, clauses and time needed to solve the encodings respectively. Times are in CPU seconds. A "*" indicates that the encoding was not solved within 5 minutes of CPU time. A "-" denotes that the encoding was not generated since the number of regions was not an integer. The descriptions of the benchmark domains used and the "satz" systematic SAT solver used are available at http://www.cs.yale.edu/HTML/YALE/CS/HyPlans/ ∼ mcdermott.html and http://aida.intellektik.informatik.th-darmstadt.de/ ∼ hoos/SATLIB respectively. Log is the transportation logistics domain and Tsp is the traveling salesperson domain.

References

[1] M. Ernst, T. Millstein and D. Weld, Automatic SAT compilation of planning problems, Proccedings of the International Joint Conference on Artificial Intelligence (IJCAI), 1997.

[2] E. Giunchiglia, A. Massarotto and R. Sebastiani, Act and the rest will follow: Exploiting determinism in planning as satisfiability, Proceedings of the National Conference on Artificial Intelligence (AAAI), 1998.

[3] S. Kambhampati, Challenges in bridging the plan synthesis paradigms, Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 1997.

[4] S. Kambhampati and B. Srivastava, Unifying state space and plan space approaches to classical planning, Technical Report, Dept. of computer science, Arizona state university, 1996.

[5] H. Kautz, D. McAllester and B. Selman, Encoding plans in propositional logic, Proceedings of the Knowledge Representation and Reasoning conference (KRR), Cambridge, USA, 1996.

[6] H. Kautz and B. Selman, Pushing the envelope: Planning, Propositional logic and Stochastic search, Proceedings of the National Conference on Artificial Intelligence, Portland, USA, (AAAI), 1996.

[7] A. D. Mali and S. Kambhampati, On the utility of plan-space (causal) encodings, Proceedings of the National Conference on Artificial Intelligence (AAAI), Orlando, USA, 1999.

[8] A. D. Mali, Hierarchical task network planning as satisfiability, Proceedings of the European Conference on Planning (ECP), Durham, UK, 1999.

[9] A. D. Mali, Plan merging and plan reuse as satisfiability, Proceedings of the European Conference on Planning (ECP), Durham, UK, 1999.