# Applying Marginal MAP Search to Probabilistic Conformant Planning: Initial Results

**Junkyu Lee and Rina Dechter**
University of California, Irvine
junkyul@uci.edu, dechter@ics.uci.edu

**Radu Marinescu**
IMB Research, Ireland
radu.marinescu@ie.ibm.com

### Abstract

In this position paper, we present our current progress in applying marginal MAP algorithms for solving the conformant planning problems. Conformant planning problem is formulated as probabilistic inference in graphical models compiled from relational PPDDL domains. The translation from PPDDL into Dynamic Bayesian Network is developed by mapping the SAT encoding of the ground PPDDL into factored representation. We experimented with recently developed AND/OR branch and bound search algorithms for marginal MAP over instances from the international planning competition domains, and we show that several domains were solved efficiently.

## Introduction

We present our current progress in applying a general purpose Marignal MAP search algorithms (Marinescu, Dechter, and Ihler 2014) for solving conformant planning problems by encoding their relational formulae into DBN (Dynamic Bayesian Network) up to fixed time horizon, into a factored representation.

The probabilistic planning problem $P = \langle S, I, G, A \rangle$ we address is based on the probabilistic extension to sequential STRIPS with negation and conditional effects, where the S is the set of states grounded from function free first order predicates, the I is the probabilistic distribution over initial states, the G is the goal that must be satisfied, and the A is a set of actions which are instances of action schemata.

Probabilistic conformant planning is the task of generating a sequence of actions achieving the goal without sensing. In other words, a planner should find a sequence of actions in the belief state space that is resulting from the uncertainty of initial states and action effects. The query to the planner can be formulated in two different ways: $\langle P, \theta \rangle$, which ask for a plan achieving the goal with probability that exceeds a threshold $\theta$, or $\langle P, T \rangle$, which fixes the time horizon $T$ and ask for a plan with maximum probability.

We define an optimal probabilistic conformant planner as a planner which always finds the shortest length plan that exceeds a threshold. Note that a planner which returns a plan from the query $\langle P, T \rangle$ can be transformed into an optimal planner by increasing the time horizon incrementally. For example, COMPLAN (Huang 2006) and CPPlan (Hyafil and Bacchus 2004) answer to the query $\langle P, T \rangle$. On the other hand, Probabilistic Fast Forward (Domshlak and Hoffmann 2007) returns some sequence of actions that exceeds the threshold.

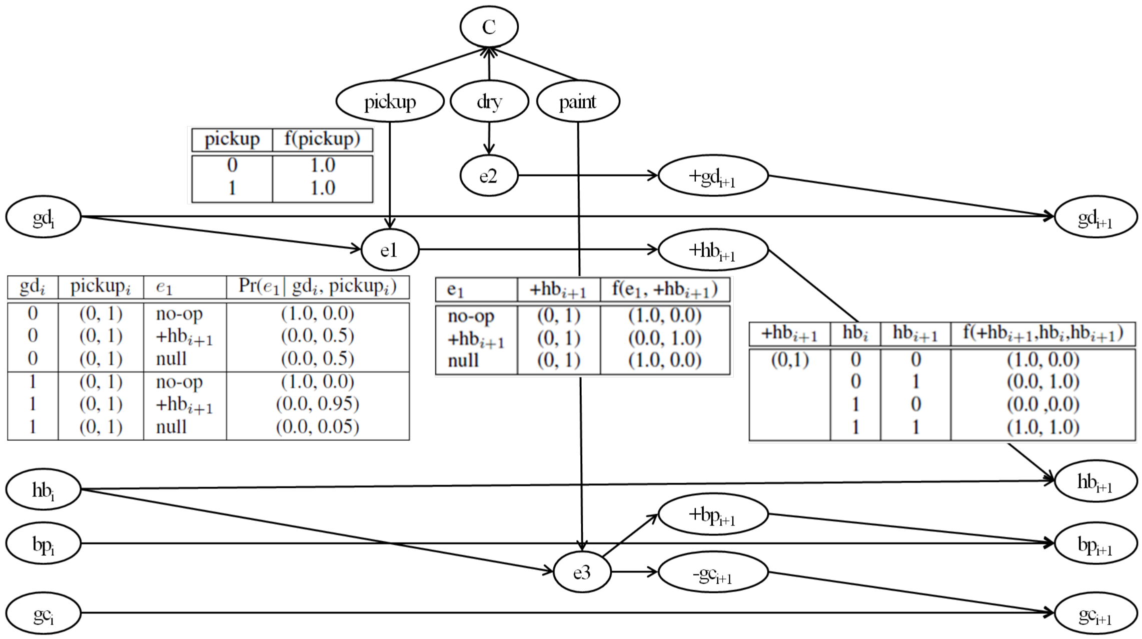
We address probabilistic planning problems expressed in Probabilistic Planning Domain Definition Language (Younes and Littman 2004) with minor extensions to express probability distribution over the initial states. Such problem domains consist of function free first order predicates to encode state variables, and parameterized action schemata with a precondition and an effect, expressed in the first order syntax. The precondition of an action was treated like a condition of conditional effect such that executing an inapplicable action in a state that violates the precondition results in no-op. For simplicity, we don't deal with quantifiers and disjunctions in the condition, and we assumed that there is only single goal state in the goal.

In this paper we consider the optimal probabilistic conformant plan that maximizes the probability of achieving the goal given a fixed time horizon. We show that the optimal plan can be found by marginal MAP inference over the DBN that compiles a planning problem properly. Such a translation of PPDDL into DBN was developed based on the SAT encoding of planning problems (Kautz, McAllester, and Selman 1996), where the translation is applied after grounding the lifted expressions by grounding each predicate and action schema into state and action variables. Benchmark problems were selected from the international planning competitions to assess the applicability of marignal MAP inference for planning.

## Conformant Planning as Marignal MAP

Our task is to solve the conformant planning problem $P = \langle \mathbf{s_0}..\mathbf{s_T}, \mathbf{a_0}..\mathbf{a_{T-1}}, T \rangle$. Namely to find a length $T$ sequence of action assignments $\mathbf{a_0}..\mathbf{a_{T-1}}$ that maximizes the conditional probability of reaching the goal $\mathbf{s_T}$ given the initial state $\mathbf{s_0}$ and the action assignments as follows:

$$\arg \max_{\mathbf{a_0}..\mathbf{a_{T-1}}} \sum_{\mathbf{s_1}..\mathbf{s_{T-1}}} P(\mathbf{s_{1..T-1}} | \mathbf{s_0}, \mathbf{s_T}, \mathbf{a_{0..T-1}}) \quad (1)$$




| pickup | f(pickup) |
|---|---|
| 0 | 1.0 |
| 1 | 1.0 |

| gd$_i$ | pickup$_i$ | $e_1$ | Pr($e_1$\| gd$_i$, pickup$_i$) |
|---|---|---|---|
| 0 | (0, 1) | no-op | (1.0, 0.0) |
| 0 | (0, 1) | +hb$_{i+1}$ | (0.0, 0.5) |
| 0 | (0, 1) | null | (0.0, 0.5) |
| 1 | (0, 1) | no-op | (1.0, 0.0) |
| 1 | (0, 1) | +hb$_{i+1}$ | (0.0, 0.95) |
| 1 | (0, 1) | null | (0.0, 0.05) |

| e$_1$ | +hb$_{i+1}$ | f(e$_1$, +hb$_{i+1}$) |
|---|---|---|
| no-op | (0, 1) | (1.0, 0.0) |
| +hb$_{i+1}$ | (0, 1) | (0.0, 1.0) |
| null | (0, 1) | (1.0, 0.0) |

| +hb$_{i+1}$ | hb$_i$ | hb$_{i+1}$ | f(+hb$_{i+1}$,hb$_i$,hb$_{i+1}$) |
|---|---|---|---|
| (0,1) | 0 | 0 | (1.0, 0.0) |
| | 0 | 1 | (0.0, 1.0) |
| | 1 | 0 | (0.0 ,0.0) |
| | 1 | 1 | (1.0, 1.0) |

Figure 1: 2TDBN for the slippery gripper domain

where the $\mathbf{a_i}$ is the vector of action variables at the $i$-th stage, and the $\mathbf{s_i}$ is the vecor of state variables at the $i$-th stage. If multiple goal states were allowed in the goal, we would add addtional layer $r_{T+1}$ qualifying the final state $s_T$, and the summation would subsume the final state $s_T$.

## Encoding PPDDL into DBN

We developed translation from PPDDL domain into 2 stage DBN (2TDBN) encoding state transitions in a single time step. Our 2TDBN encoding extends the translation given in (Younes and Littman 2004) by introducing additional variables and functional relations to incorporate action variables and bound the maximum number of parents of a node.

The semantics of 2TDBN is based on the linear encoding of SATPLAN with additional multivalued effect variables to address probabilistic outcomes of a probabilistic effect. The $j$-th boolean state variables $s_i^j$ of the state $\mathbf{s}_i$, and the $k$-th boolean action variables $a_i^k$ of the action $\mathbf{a}_i$ are obtained by grounding the lifted predicates and action schemata, where the $i$ is the time step.

A ground action consists of a precondition $\phi$, where the $\phi$ is a conjunction of equality predicates on ground objects and literals of state variables, and an effect which can be either a simple effect, a conditional effect, a probabilistic effect or a conjunction of other effects. We assumed that a planner would execute a no-op action if the precondition didn't meet. Thus, an action is implicitly converted to a conditional effect, which is the so called weak precondition. From now on, we do not distinguish between a precondition of an action and a condition in the conditional effect.

A simple effect is a conjunction of state variables where positive literals corresponds to add list and negative literals corresponds to delete list of STRIPS formalism. A conditional effect is a pair $(\phi, e)$, where the $\phi$ is a condition defined eariler, and the $e$ is an effect. A probabilistic effect is a list of pairs $(p_i, e_i)$, where the $p_i$ is the probability value for the $i$-th effect $e_i$. Thus, a PPDDL effect can be nested in arbitrary depth forming a tree of which simple effects placed at the leaf. For each effect we introduce an effect variable capturing the outcomes of the effect. Specifically, a probabilistic effect having $n$ outcomes maps to an effect variable with $n+1$ values where the additional value is reserved for the no-op. Finally, addtional variables +s$_{i+1}^k$ and -s$_{i+1}^k$ are defined to indicate whether the state variable $s_{i+1}^k$ will be added or deleted.

A 2TDBN $G = \langle V, F \rangle$ with a set of variables $V$ and a set of conditional probability table $F$ for each variable can be obtained by converting following clauses:

- $a_i^j \wedge \phi_i^j \rightarrow (e_i^j \neq$ no-op$)$, $\neg(a_i^j \wedge \phi_i^j) \rightarrow (e_i^j =$ no-op$)$, where $\phi_i^j$ is the precondition of $a_i^j$
- $\vee_{k \in K}$ (e$_i^j = v$) $\equiv$ +/-s$_{i+1}$, where K = $\{k|$ s$_{i+1}^j \in \cup_v$ add/del(e$_i^k = v$) $\}$
- $\forall_{k_1 \neq k_2}$ (e$_i^{k_1} = v_1$) $\wedge$ (e$_i^{k_2} = v_2$) $\rightarrow \neg$+/-s$_{i+1}$, if$s_{i+1} \in$add/del($e_i^{k_1} = v_1$) and $s_{i+1} \in$add/del($e_i^{k_2} = v_2$)
- $\forall_j \vee a_i^j$, $\quad \forall_{j \neq k}\ a_i^j \rightarrow \neg a_i^k$
- $\neg$ +$s_{i+1}^j \wedge \neg$ -$s_{i+1}^j \rightarrow (s_i^j \wedge s_{i+1}^j) \vee (\neg s_i^j \wedge \neg s_{i+1}^j)$

Note that above clauses encode a planning doamin where each action has a single probabilistic effect without nesting. So, an outcome $v$ of the $j$-th effect variable at the $i$-th time stage ($e_i^j = v$) corresponds to a simple effect. For the nested effects, we consult in normal forms in (Rintanen 2003). The first clauses correspond to the CPT for an effect variable, where each outcome occurs only if the action triggering the effect was executed and the precondition was satisfied. The second and the third clauses encode the CPT for an indicator variable to indicate the result of executing some action as well as to depricate inconsistent combination of effects. The last two clauses encode the mutual exclusivity on action variables and the frame axioms. Concurrent actions can be easily adopted by modifying the previous propositional encoding.

Consider slippery gripper domain (Kushmerick, Hanks, and Weld 1995) which consists of four state variables, gripper-dry (gd), holding-block (hb), block-painted (bp), and gripper-clean (gc), and three actions, “pick up”, “dry”, and “paint”. Initially, the gc is true, the gd is true with probability 0.7, and other two state variables are false. The goal is a conjunction of lieterals, gd $\wedge$ hb $\wedge$ bp. If gripper-dry were true, the “pick up” action would add hold-block with probability 0.95. Otherwise hold-block would be added with probability 0.5. The “dry” action would add gripper-dry with probability 0.8. Lastly, If holding-block were true, gripper-clean will be deleted by “paint” action, and if it were false, gripper-clean would be deleted with probability 0.1. The “paint” action also adds block-painted.

Fig. 1 shows the 2TDBN for the slippery gripper domain. Each value of the CPT for “pick up” is normalized to 1.0 because the joint conditional distribution is conditioned on the action variable. The effect variable $\mathrm{e}_1$ takes three values, the no-op for the case where the “pick up” action was not executed, the $+\mathrm{hb}_{i+1}$ for adding the state variable $hb_{i+1}$, and the null effect derived from the semantics of probabilistic effect, i.e., when $\sum_i p_i < 1$, the probabilistic effect takes an additional null effect with probability $1 - \sum_i p_i$. The probability of each outcome is mapped to the corresponding row of the CPT. Note that the effect CPTs from probabilistic effects are only probabilistic. The rest of the two tables in Fig. 1 are simply rewritten in tabular form of the propositional encoding. Finally, the CPT for the node $C$ in Fig. 1 is also in tabular form of the mutual exclusivity clauses.

The DBN with time horizon $T$ will be obtained by replicating the 2TDBN over $T$ times. The probability distribution over the initial state variables and the single goal state can be assigned to the state variables $\mathbf{s_0}$, and $\mathbf{s_T}$.

In practice, in-degree of the nodes for the constraint on action variables, and the indicator variables $\pm s_{i+1}^{j}$ grows prohibitively for storing the tabular form of the CPT, so additional hidden variables were introduced to bound the maximum in-degrees.

The complexity of the resulting 2TDBN is similar to that of SATPLAN, and it can be summarized as follows. Let the number of action schemata in the planning domain be $|as|$, the number of predicates be $|pre|$, the maximum arity of the action schemata be $p$, the maximum arity of the predicates be $q$, and the number of ground objects in the instance be $k$. The total number of the action variables is $O(|as|k^p)$, and the total number of the state variables is $O(|pre|k^q)$. Assuming that we have only flat effects, the number of effect variables are the same as the number of action variables. Thus, the total number of variables in a 2TDBN is $O(2|as|k^p + 3|pre|k^q)$. In case of introducing hidden variables to bound the maxmum in-degree, we would introduce $|as|k^p - 1$ hidden variables to bound the maximum in-degree of the constraint variable to be 2. Similarly, if the maximum number of parents for the indicator variables was $|E|$, we would introduce $|E| - 1$ hidden variables. In such a case, the maximum arity of the CPTs is $max(3, s+1)$, where the s is the maximum number of state variables shown in a precondition which contributes to the arity of the CPT for an effect variable. The maximum domain size is $n+1$, where the $n$ is the maximum number of outcomes of a probabilistic effect. Finally, the total number of variables is $O(3|as|k^p + (1+2|E|)|pre|k^q)$.

## Marginal MAP Algorithms

We are using recently proposed AND/OR Branch and Bound (AOBB) search algorithms for marginal MAP (Marinescu, Dechter, and Ihler 2014), where heuristics were created by Weighted Mini Bucket elimination (WMB) from the constrained variable ordering. AOBB search heuristics are:

- Pure MB (Dechter and Rish 2003)
- MB with moment matching (Ihler et al. 2012)
- WMB with join graph cost shifting (Ihler et al. 2012)

Constraint propagation also employed on the conditioned subproblems by CNF encoding of zero probability tuples and applying unit resolution or full satisfiability (Marinescu and Dechter 2009). In addition, we tested our other algorithm BBBT, which is an OR branch and bound search guided by MCTE (Mini Cluster Tree Elimination) heuristic as well as Yuan's algorithm (Yuan and Hansen 2009).

## Experiment Result

The benchmark instances were selected from the planing domains provided by past international planning competitions: “slippery gripper”, “comm”, and “blocks world”. For the blocks world domain the instances are from probabilistic track of IPC 2006, and we removed three “tower actions” that can move more than two blocks at once. The problem instances for the blocks world domain were recreated such that there exists shorter length plan. The slippery gripper instances and “comm” domain were not modified from the original release. Note that the “comm” domain has only uncertain inital state. In contrast, blocks world domain has only probabilistic transitions, and the slippery gripper domain contains uncertain initial state and probabilistic state transitions.

Table 1 summarizes our benchmark results from three marginal MAP algorithms: 1. AOBB search with the WMB-JG heuristic ran for 10 iterations and full constraint propagation for deterministic tuples in the CPTs (AOBB-WMB-JG), 2. BBBT search with the incremental MCTE heuristic (BBBTi), and 3. Yuan's algorithm. The tuple ($n$, $k$, $s$, $|S|$, $|A|$) is problem statistics showing the total number of variables in the 2TDBN, the maximum domain size, the maximum scope size, and the number of the states and action variables per time step.

The number of variables $n$ is much higher than sum of grounded state/action variables since we introduced hidden

Table 1: Experiment results from marignal MAP search algorithms. Time/Memory limit 6 Hr/4GB.

| | Solution | AOBB-JG | BBBTi | YUAN | Solution | AOBB-JG | BBBTi | YUAN | Solution | AOBB-JG | BBBTi | YUAN |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Domain Name | max prob | time | time | time | max prob | time | time | time | max prob | time | time | time |
| (n, k, s, $\|S\|$, $\|A\|$) | (L, w, h) | (i, c) | (i, c) | - | (L, w, h) | (i, c) | (i, c) | - | (L, w, h) | (i, c) | (i, c) | - |
| slippery gripper | 0.7335 | 0.01 | 0.01 | 0.01 | 0.9 | 210 | 54 | 12 | 0.9 | oot | 8906 | 3734 |
| (23, 3, 3, 4, 3) | (2, 6, 3) | (6, 6) | (6, 6) | - | (14, 44, 58) | (22, 22) | (30, 30) | - | (21, 65, 82) | (24, 24) | (26, 26) | - |
| | 0.8985 | 8.33 | 0.08 | 0.03 | 0.9 | 8840 | 2329 | 881 | 0.9 | oot | oot | 7334 |
| | (6, 20, 29) | (20, 20) | (20, 20) | - | (19, 59, 74) | (26, 26) | (30, 30) | - | (22, 68, 88) | (24, 24) | (26, 26) | - |
| | 0.8999 | 5.49 | 1.99 | 0.65 | 0.9 | 19262 | 7481 | 2121 | 0.9 | oot | oot | 19607 |
| | (10, 32, 44) | (20, 20) | (30, 30) | - | (20, 62, 78) | (26, 26) | (30, 30) | - | (23, 71, 86) | (24, 24) | (26, 26) | - |
| comm | 0.25 | 11 | oom | oom | 0.25 | 74 | oom | oom | 0.25 | 54 | oom | oom |
| (349, 2, 2, 45, 46) | (5, 269, 324) | (8,8) | (2,2) | - | (7, 363, 436) | (2, 2) | - | | (9, 467, 540) | (2, 2) | (2, 2) | - |
| bw-24 | 0.1406 | 0.57 | 0.35 | 7 | 0.7031 | 5 | 8.93 | 92 | 0.8701 | 879 | 9513 | 18809 |
| (73, 3, 5, 9, 8) | (3, 32, 55) | (10,10) | (12, 1) | - | (5, 51, 80) | (10, 10) | (10, 10) | - | (7, 69, 101) | (16, 16) | (14,14) | - |
| | 0.5625 | 1 | 1.95 | 10 | 0.8085 | 67 | 242 | 1029 | 0.9162 | 9930 | oot | oot |
| | (4, 40, 66) | (10, 10) | (12, 12) | - | (6, 57, 90) | (12, 12) | (12, 12) | - | (8, 72, 115) | (18, 18) | (22, 22) | - |
| bw-34 | 0.0351 | 1.71 | 2.19 | oom | 0.1406 | 225 | 217 | oom | 0.5273 | 1572 | 7681 | oom |
| (170, 3, 5, 16, 18) | (4, 101, 142) | (4,4) | (8,8) | - | (6, 135, 191) | (8, 8) | (8, 8) | - | (7, 156, 221) | (8, 8) | (8, 8) | - |

variables for high in-degree CPTs due to the action constraints and the frame axioms. The solution is reported for the maximum probability with length $L$, the induced width $w$, and the pseudo tree height $h$ for the topological variable ordering. AOBB-WMB-JG and BBBTi shares the same algorithmic parameters, the $i$ bound for the WMB and the $c$ bound for the caching, which are shown below the running time for each algorithms.

**slippery gripper** The optimal plan for the slippery gripper domain should satisfy two partial order relations: the "paint" action precedes the "pick up" action, and the "dry" action precedes the "pick up" action. One such valid plan is (paint, dry$^i$, pick-up$^j$), where the $i \geq 0$ and the $j \geq 1$ denotes the number of "dry" and "pick up" actions executed in the plan, and $T = 1 + i + j$. The probability of success for the plans satisfying above partial order relations is $0.9(d_i p_j + (1 - d_i) q_j)$, where $d_0 = 0.7$, $d_i = d_{i-1} + 0.8(1 - d_{i-1})$, $p_1 = 0.95$, $p_j = p_{j-1} + 0.95(1 - p_{j-1})$, $q_1 = 0.5$, and $q_j = q_{j-1} + 0.5(1 - q_{j-1})$. For example, the optimal length 3 plan is (paint, pick up, pick up) with the probability $0.830925$. Thus, we can assure that the plans found by marginal MAP search algorithms were truely optimal by checking above equations on various $i$ and $j$ values.

Yuan's algorithm performed best among three algorithms, and it was able to find conformant plans up to 23 time horizon in 6 hour time limit. On the other hand, BBBT search with incremental MCTE heuristics was able to find up to 21 time horizon.

**comm** For the "p01" problem from the IPC 2006 competition, AOBB-WMB-JG solved problems up to 9 time horizon. We suspect that the full constraint propagation incorporated in AOBB-WMB-JG played a significant role for solving the problems. Compared to other domains, induced width of the constrained ordering was prohibitively huge, for example the induced width was 467 for the 9 time horizon instance.

**simple blocks world** Two blocks world domain were reported. Each of them has 2 and 3 blocks with 4 action schemata. AOBB-WMB-JG was able to find plans for up to 8 time horizon for bw-24 and 7 time horizon for bw-34. Since the number of state/action variables is exponentially increasing, the algorithms couldn't find a solution for the problems having more than 3 blocks.

## Comparison with Other Planners

In this section, we compare the marginal MAP search algorithms with COMPLAN and Probabilistic-FF. The performance of Probabilistic-FF was evaluated in the same benchmark envrionment.

To the best of our knowledge, COMPLAN is the most recent probabilistic conformant planning algorithm that maximizes the probability of success for a plan with a fixed time horizon. COMPLAN extends linear encoding of SATPLAN with chance variables that encode the uncertainity of a planning problem, and compiles CNFs into deterministic decomposable negation normal form (Darwiche 2004), and then apply the marginal MAP query. The conformant plan is found by depth first branch and bound search guided by heuristics generated by a relaxed marginal MAP query on the d-DNNF compilation. Note that the induced width of constrained variable orderings for a planning problem is usually prohibited as shown in the Table 1. AOBB-WMB-JG relaxes the problem based on the weighted mini-buckets with cost shifting, whereas COMPLAN allows an unconstrained variable ordering to yield lower induced width.

Probabilistic-FF finds a conformant plan that achieves the goal higher than a desired threshold by heuristic forward search in a belief state space. The belief states $b_{\bar{a}}$ is a probability distribution over states reachable from initial belief state $b_I$ by applying a sequence of actions $\bar{a}$. Probabilistic-FF also represents belief states by DBN $N_{b_{\bar{a}}}$ (Boutilier, Dean, and Hanks 1999), and the DBN is further compiled into weighted CNFs, to retrieve probabilitic queries via weigthed model counting (Sang, Beame, and Kautz 2005).

In our representation, all the action variables were nodes of the DBN such that the probabilistic inference directly query over the action variables after marginalizing the rest of the variables. On the other hand, each DBN for the belief states $b_{\bar{a}}$ of the Probabilistic-FF unrolls only portion of the entire DBN that can be instantiated by applying a specific sequence of actions $\bar{a}$, and the weighted model counting computes only posterior probability so that the heuristic function evaluates the probability of achieving the goal.

Table 2 compares Probabilistic-FF and AOBB-WMB-JG on the planning problems: "slippery gripper", and "simple blocks world". We set threshold values of the Probabilistic-

Table 2: Comparison between Probabilistic-FF and AOBB-WMB-JG. Time/Memory limit 6 Hr/4GB.

| s-gripper | $\theta$ | 0.7335 | 0.830925 | 0.884385 | 0.895077 | 0.898539 | 0.899618 | 0.899859 | 0.899967 | 0.899989 | 0.899999 | 0.9 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| pff | time | 0.04 | 0.03 | 0.04 | 0.05 | 0.04 | 0.05 | 0.04 | 0.04 | 0.07 | 0.11 | oot |
| (h1,w0) | length | 3 | 4 | 5 | 6 | 8 | 10 | 11 | 13 | 15 | 18 | - |
| pff | time | 0.03 | 0.19 | 0.42 | 1.22 | 1.27 | 3.05 | 6.29 | 13.89 | 31.56 | 156.86 | oot |
| (h2, w1) | length | 2 | 4 | 5 | 6 | 6 | 7 | 8 | 9 | 10 | 12 | - |
| AOBB | pre, tot | 0.01, 0.01 | 0.02, 0.02 | 0.13, 0.13 | 0.98, 0.98 | 8.33, 8.33 | 66.23, 66.23 | 3.37, 4.13 | 4.27, 5.19 | 46.94, 49.29 | 9.07 ,37.23 | 90.59, 126.41 |
| WMB-JG | L, w, i | 2, 6, 6 | 3, 10, 10 | 4, 14, 14 | 5, 17, 17 | 6, 20, 20 | 7, 23, 10 | 8, 27, 22 | 9, 29, 22 | 10, 32, 28 | 12,38,24 | 13, 41, 24 |

| bw-24 | $\theta$ | 0.140625 | 0.5625 | 0.703125 | 0.808594 | 0.870117 |
|---|---|---|---|---|---|---|
| pff | time | 0.04 | 0.05 | oom | oom | oom |
| (h1, w0) | length | 4 | 4 | - | - | - |
| AOBB | pre, tot | 0.56, 0.57 | 0.9, 1.06 | 1.28, 5 | 3.62, 67 | 55.6, 879 |
| WMB-JG | L, w, i | 3, 32, 10 | 4, 40, 10 | 5, 48, 10 | 6, 57, 12 | 7, 67, 16 |

| bw-34 | $\theta$ | 0.140625 | 0.2 | 0.4 | 0.527344 |
|---|---|---|---|---|---|
| pff | time | 0.09 | 0.09 | 0.1 | error |
| (h2, w0) | length | 6 | 6 | 6 | - |
| AOBB | pre, tot | 5, 225 | - | - | 7.4, 1572 |
| WMB-JG | L, w, i | 6, 135, 8 | - | - | 7, 156, 8 |

FF from optimal conformant planning solutions for comparison. For the Probabilistic-FF we present the best result from 2 heuristic functions and 3 weight propagation schemes. For the detail about Probabilistic-FF planner, we refer to the Probabilistic-FF codes available at the author's website. For the AOBB-WMB-JG, we present preprocessing time for compiling WMB-JG static heuristics with 10 iterations for the join graph cost shifting, the total time that covers both preprocessing time and AOBB search time, and triplet $(L, w, i)$ for the shortest plan length, the induced width of a constrained variable ordering, and the i-bound parameter for WMB.

**slippery gripper** Probabilistic-FF solved all the instances around 0.1 seconds except the one with threshold 0.9. The length of the plan is typically longer than the optimal, but plans that are closer to the optimal could be found with increased time. For the slippery gripper problem, the threshold 0.9 is the maximum probability of success achievable.

Considering the length of the plan, the performance of AOBB-WMB-JG is comparable to Probabilistic-FF in many problem instances. The WMB relaxation was exact, i.e., the i-bound was same as the induced width when the time horizon was less than 7. Thus, AOBB-WMB-JG was solved by inference.

**simple blocks world** In two problem domains, bw-24 and bw-34, Probabilistic-FF was not able to find a plan longer than 4 and 6 due to the 4GB memory limit. In our experiments, Probabilistic-FF showed cut-off thresholds for both domains, 0.6 and 0.5 for each. As we discussed in previous sections, AOBB-WMB-JG ran out of the time when the time horizon grows higher than 8 and 7 for each domain.

## Discussion and Future Work

We applied state of the art marginal MAP solver to probabilistic conformant planning problems by converting SAT encoding of a planning instance into DBN. We were able to solve instances from three domains that are orginated from the past planning competitions. When the size of the factored representation at the propositional level was manageable, our marginal MAP solver could find the optimal plan given a fixed time horizon. On the other hand, the heuristics based on the approximate probabilistic inference becomes inaccurate as we extend the time horizon since the induced width of constrained variable ordering increases linearly.

**Benchmark on harder problems** The three planning domains we addressed in this paper are easy domains; "slippery gripper" domain was already defined at the propositional level and it contains 4 state variables and 3 action variables. "Blocks world" domains were simpified by removing three action schemata in the orginal release to bound the number of ground variables at each time stage. The problem instances solved by marginal MAP algorithms were limited to contain at most three blocks and 7 time horizon. Therefore, our goal is to translate the planning domains in more compact way, and improve the probabilistic inference to exploit symmetry and determinism present in the planning problem.

**Compact translation** More compact translation at the propositional level is desired to push the marginal MAP search algorithms to solve planning instances leaving a larger number of constant objects and longer time horizon at the ground level. The SAS+ based SAT encoding for the classical planning is known to produce more compact translation compared to the linear encoding of SATPLAN (Huang, Chen, and Zhang 2010). Thus, the straightforward extension of our current work is to convert a STRIPS like PPDDL domain into FDR (finite domain representation) (Helmert 2009), and translate the new SAT encoding into the graphical model with proper extension addressing the uncertainty.

**Need for lifting** Planning problems have two source of lifting: the time horizon which implies replicated structure over the time axis, and the state/action representation parameterized by obejct variables. It was shown that MDP/POMDP planning can be reformulated as a mixture model of unbounded number finite length DBNs (Toussaint 2009) with flat state variables. Still, lifted inference for the parameterized random variables declared in both STRIPS like PPDDL and SAS+ based FDR has not gained much attention in the literature. We leave improving current algorithms in conjunction with lifted inference on parfactor graphs possibly translated from lifted planning probelm representations as future work.

## References


Boutilier, C.; Dean, T.; and Hanks, S. 1999. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research* 1:1–93.

Darwiche, A. 2004. New advances in compiling cnf to decomposable negation normal form. In *Proc. of ECAI*, 328–332. Citeseer.

Dechter, R., and Rish, I. 2003. Mini-buckets: A general scheme for bounded inference. *Journal of the ACM (JACM)* 50(2):107–153.

Domshlak, C., and Hoffmann, J. 2007. Probabilistic planning via heuristic forward search and weighted model counting. *J. Artif. Intell. Res.(JAIR)* 30:565–620.

Helmert, M. 2009. Concise finite-domain representations for pddl planning tasks. *Artificial Intelligence* 173(5):503–535.

Huang, R.; Chen, Y.; and Zhang, W. 2010. A novel transition based encoding scheme for planning as satisfiability. In *AAAI*.

Huang, J. 2006. Complan: A conformant probabilistic planner. *ICAPS 2006* 63.

Hyafil, N., and Bacchus, F. 2004. Utilizing structured representations and csps in conformant probabilistic planning. In *ECAI*, volume 16, 1033.

Ihler, A.; Flerova, N.; Dechter, R.; and Otten, L. 2012. Join-graph based cost-shifting schemes. In *UAI*.

Kautz, H.; McAllester, D.; and Selman, B. 1996. Encoding plans in propositional logic. *KR* 96:374–384.

Kushmerick, N.; Hanks, S.; and Weld, D. S. 1995. An algorithm for probabilistic planning. *Artificial Intelligence* 76(1):239–286.

Marinescu, R., and Dechter, R. 2009. And/or branch-and-bound search for combinatorial optimization in graphical models. *Artificial Intelligence* 173(16):1457–1491.

Marinescu, R.; Dechter, R.; and Ihler, A. 2014. Improving marginal map search in graphical models. In *submitted to UAI*.

Rintanen, J. 2003. Expressive equivalence of formalisms for planning with sensing. In *ICAPS*, 185–194.

Sang, T.; Beame, P.; and Kautz, H. 2005. Solving bayesian networks by weighted model counting. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI-05)*, volume 1, 475–482.

Toussaint, M. 2009. Probabilistic inference as a model of planned behavior. *Künstliche Intelligenz* 3(9):23–29.

Younes, H. L., and Littman, M. L. 2004. Ppddl1. 0: The language for the probabilistic part of ipc-4. In *Proc. International Planning Competition*.

Yuan, C., and Hansen, E. A. 2009. Efficient computation of jointree bounds for systematic map search. In *IJCAI*, 1982–1989.