Applying Search Based Probabilistic Inference Algorithms to Probabilistic Conformant Planning: Preliminary Results

Junkyu Lee
University of California, Irvine
junkyul@ics.uci.edu

Radu Marinescu
IBM Research – Ireland
radu.marinescu@ie.ibm.com

Rina Dechter
University of California, Irvine
dechter@ics.uci.edu

Abstract

Probabilistic conformant planning problems can be solved by probabilistic inference algorithms after translating their PPDDL specifications into graphical models. We present two translation schemes that convert probabilistic conformant planning problems as graphical models. The first encoding is based on the probabilistic extension of the serial encoding of PDDL in SatPlan, and the second encoding compiles a graphical model from the finite-domain representation of the SAS+ formalism. We show that a probabilistic conformant plan can be found by answering a marginal MAP inference, and the plan is optimal with respect to the length of the plan as well as the probability of achieving the goal. Since a common task of the conformant planning is to find a plan achieving the goal with a probability that exceeds a threshold, we can consider relaxing of the marginal MAP query to the pure MAP which is far easier to compute. The success probability of the suboptimal plan derived by a pure MAP solver can be re-evaluated by solving a summation problem, also a hard task. The probabilistic inference algorithms for marginal MAP that we evaluated are based on anytime AND/OR branch and bound search guided by weighted mini-bucket heuristics. Our preliminary evaluation highlights the potential and the challenges in this methodology of applying search based probabilistic inference algorithms to probabilistic conformant planning.

Introduction

Graphical models provide a powerful framework for reasoning with probabilistic and deterministic information. These models use graph to capture conditional independencies among variables, allowing a concise representation of knowledge as well as efficient graph-based query processing algorithms. The Maximum a Posteriori (MAP) query asks for the mode of the joint probability distribution, while the Marginal MAP (MMAP) generalizes MAP by allowing a subset of the variables to be marginalized.

It is well-known that the computational complexity of the planning is equivalent to inference. Especially, the complexity of MMAP inference is equivalent to finding the best polynomial-size plan (Littman *et al.* 2001). Therefore, several translation based approaches have been developed by formulating planning under uncertainty as probabilistic inference over graphical models, hoping to apply general purpose probabilistic inference algorithms. For example, a translation from Partially Observable Markov Decision Process (POMDP) to a Dynamic Bayesian Network (DBN) was presented in (Kiselev and Poupart 2014), and the encoding scheme that translates a probabilistic conformant planning into a DBN was shown in (Lee *et al.* 2014), where the probabilistic conformant planning refers to the class of the planning problems with non-observability and stochastic actions.

The probabilistic conformant planning problem $P = \langle S, A, I, G \rangle$ is based on the probabilistic extension of sequential STRIPS with negation and conditional effects, where the S is a set of state variables grounded from first order state predicates, the A is a set of ground action variables indicating whether a ground action were chosen or not, the I is a probability distribution over initial state variables, and the G is a set of goal states, where it is assumed to be a single state for simplicity. In this paper, we address planning problems represented in Probabilistic Planning Domain Definition Language (PPDDL) (Younes and Littman 2004) with minor extension to express probability distributions over the initial states, called initial belief states.

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 belief state space $\{\mathbf{b_0}, \mathbf{b_1}, \cdots, \mathbf{b_t}\}$, where $\mathbf{b_0} = I$ and each $\mathbf{b_t}$ is a probability distribution over states $\mathbf{s_t}$ at time t. The query to the planner can be formulated in two different ways: (1) $\langle P, T \rangle$, which fixes the time horizon T and asks for a plan with maximum probability of success, $argmax_{\{a_0 \cdots a_{T-1}\}} \mathbf{b_T}(\mathbf{s_T} \in G)$, (2) $\langle P, \theta \rangle$, which asks for a plan of arbitrary length L achieving the goal with probability that exceeds a threshold θ, $\mathbf{b_L}(\mathbf{s_L} \in G) > \theta$. 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 such an optimal planner by increasing the time horizon incrementally.

In this paper, we formulate probabilistic conformant planning as MMAP inference (i.e., finding a most likely assignment to a subset of hypothesis variables) over graphical models that compiles a planning problem properly. MMAP is one of the most complex queries in probabilistic inference because it requires the application of summation of a subset of the variables and maximization over the rest in a particular (often unfavorable) order. Therefore, we focus on applying recently developed anytime AND/OR branch and bound search algorithm for MMAP (Lee *et al.* 2016) which explores the context-minimal AND/OR search space for graphical models (Dechter and Mateescu 2007) in depth-first manner, and is guided by static weighted mini-bucket heuristics with variational cost-shifting (Dechter and Rish 2003; Liu and Ihler 2011).

When MMAP task is highly intractable, suboptimal solvers for the tasks should also be considered as approximation schemes. In particular, when we consider the task as finding any plan with probability of reaching the goal with probability higher than the threshold, we consider relaxing of the MMAP query to the pure MAP to find a sub-optimal probabilistic conformant plan and re-evaluate the plan's probability of success by solving a summation problem, treating the action variables as evidence. Thus, candidate probabilistic conformant plans can be drawn from not only the optimal pure MAP assignments (Otten and Dechter 2011) but also sub-optimal solutions from various pure MAP solvers, e.g., stochastic local search algorithms for MPE (Hutter *et al.* 2005).

The experiment result of applying search based probabilistic inference algorithms reveals opportunities as well as challenges for solving probabilistic conformant planning problem by probabilistic inference. The rest of the paper is organized as follows. Section 2 formulates the probabilistic conformant planning task as a probabilistic inference, Section 3 review the search based probabilistic inference algorithms we evaluated, Section 4 presents two encoding schemes that convert PPDDL planning domains into graphical models, Section 5 shows the experiment results of applying search based probabilistic inference algorithms to the blocks world domain, and we conclude in Section 6.

Conformant Planning as Marginal MAP

A graphical model is a tuple $\mathcal{M} = \langle \mathbf{X}, \mathbf{D}, \mathbf{F} \rangle$, where $\mathbf{X} = \{X_i : i \in V\}$ is a set of variables indexed by set V and $\mathbf{D} = \{D_i : i \in V\}$ is the set of their finite domains of values. $\mathbf{F} = \{F_j\}$ is a set of local functions defined on subsets of variables called its scope. The function scopes yield a primal graph whose vertices are the variables and whose edges connect any two variables that appear in the scope of the same function.

Let $\mathbf{X}_M = \{X_1, ..., X_m\}$ be a subset of $\mathbf{X}$ called MAP variables and $\mathbf{X}_S = \mathbf{X} \setminus \mathbf{X}_M$ be the complement of $\mathbf{X}_M$, called sum variables. The MMAP task seeks an assignment $\mathbf{x}^*_M$ to variables $\mathbf{X}_M$ having maximum probability. This requires access to the marginal distribution over $\mathbf{X}_M$, which is obtained by summing out variables $\mathbf{X}_S$:

$$\mathbf{x}^*_M = \underset{\mathbf{X}_M}{\operatorname{argmax}} \sum_{\mathbf{X}_S} \prod_{F_j \in \mathbf{F}} F_j \tag{1}$$

Note that, the MMAP task reduces to a pure MAP task if $\mathbf{X}_S$ is empty and it also reduces to a PR task (i.e., evaluating the probability of evidence) if $\mathbf{X}_M$ is empty.

Our task is to solve the conformant planning problem formulated either as to $\langle P, T \rangle$ or $\langle P, \theta \rangle$, where $P = \langle S, A, I, G \rangle$. The graphical model for P can be defined as follows. The set of boolean variables $\mathbf{X}$ is $\{\mathbf{s_0}, \mathbf{s_1}, \cdots, \mathbf{s_T}\} \cup \{\mathbf{a_0}, \mathbf{a_1}, \cdots, \mathbf{a_{T-1}}\}$, where the $\mathbf{s_t}$ is a vector over state variables grounded from first order state predicates at time t, and the $\mathbf{a_t}$ is a vector over latent action variables indicating that some action was chosen at time t. For each ground action, probabilistic state transition function can be defined as $F_t(\mathbf{a_t}) = Pr(\mathbf{s_{t+1}}|\mathbf{s_t}, \mathbf{a_t})$, where $F_t(\mathbf{a_t})$ is a local factor over 1 step transition between time t and $t+1$ given an action $\mathbf{a_t}$. Usually, each F_t can be further factored by preconditions and post conditions of each action $\mathbf{a_t}$ because each ground action only affects a small fraction of state variables. The initial belief state I is $F_0 = Pr(\mathbf{s_0})$. Finally, we have a constraint that characterizes the final state that $C(\mathbf{s_T}) = I(\mathbf{s_T} \in G)$, if $\mathbf{s_T} \notin G$, $C(\mathbf{s_T}) = 0$. The overall functions can be combined as,

$$Pr(\mathbf{s_0}..\mathbf{s_T}|\mathbf{a_0}..\mathbf{a_{T-1}}) = \prod_{t=0..T} F_t(\mathbf{a_t}) C(\mathbf{s_T}). \tag{2}$$

The probabilistic conformant planning task can be reformulated as probabilistic inference over the graphical model as follows.

$\langle P, T \rangle$: Finding the length T sequence of actions,

$$argmax_{\{\mathbf{a_0}..\mathbf{a_{T-1}}\}} \sum_{\mathbf{s_0}..\mathbf{s_T}} \prod_{t=0..T} F_t(\mathbf{a_t}) C(\mathbf{s_T}).$$

$\langle P, \theta \rangle$: Finding any length L sequence of actions,

$$\sum_{\mathbf{s_0}..\mathbf{s_L}} \prod_{t=0..L} F_t(\mathbf{a_t}) C(\mathbf{s_L}) > \theta.$$

The first query is equivalent to MMAP, where the hypothesis variables are action variables and the summation variables are state variables. In contrast to the first query, it requires finding the maximum probability of success given a fixed time horizon T. The second query is searching for an arbitrary length L plan that satisfies the inequality. Thus, it is equivalent to evaluating the probability of evidence of the distribution defined in equation 2 after treating a length L plan as evidence, and testing if it is greater than θ. Such length L plan could be found by arbitrary methods, e.g., action variables truncated from MAP solution to the problem.

AND/OR Search for Marginal MAP

Recent advances in search for MMAP inference have been achieved by using AND/OR search spaces (Dechter and Mateescu 2007), and depth-first or best-first AND/OR search algorithms guided by mini-bucket heuristics enhanced with variational cost-shifting ideas (Marinescu *et al.* 2014; 2015). In this section, we review AND/OR search space and the depth-fisrt AND/OR search algorithms for MMAP.

AND/OR Search Space

The AND/OR search space is defined relative to a pseudo tree of the primal graph, which captures problem decomposition. A pseudo tree of an undirected graph $G = (V, E)$ is a directed rooted tree $\mathcal{T} = (V, E')$ such that every arc of G not included in E' is a back-arc in $\mathcal{T}$ connecting a node in $\mathcal{T}$ to one of its ancestors. The arcs in E' may not all be included in E. A pseudo tree $\mathcal{T}$ of G is valid for MAP variables $\mathbf{X}_M$ if the restricted pseudo tree $\mathcal{T}'$ which prunes all nodes except $\mathbf{X}_M$ is a connected pseudo tree with the same root as $\mathcal{T}$.

Given a graphical model $\mathcal{M} = \langle \mathbf{X}, \mathbf{D}, \mathbf{F} \rangle$ with primal graph G and valid pseudo tree $\mathcal{T}$ of G, the AND/OR search tree $S_{\mathcal{T}}$ based on $\mathcal{T}$ has alternating levels of OR nodes corresponding to the variables, and AND nodes corresponding to the values of the OR parent's variable, with edge weights extracted from the original functions $\mathbf{F}$. Identical sub-problems, identified by their context (the partial instantiation that separates the sub-problem from the rest of the problem graph), can be merged, yielding an AND/OR search graph. Merging all context-mergeable nodes yields the context minimal AND/OR search graph, denoted $C_{\mathcal{T}}$. The size of $C_{\mathcal{T}}$ is exponential in the induced width of G along a depth-first traversal of $\mathcal{T}$ (i.e., the constrained induced width).

A solution subtree $\hat{x}_M$ of $C_{\mathcal{T}}$ relative to the MAP variables $\mathbf{X}_M$ is a subtree of $C_{\mathcal{T}}$ restricted to $\mathbf{X}_M$ such that it contains the root of $C_{\mathcal{T}}$, n is labeled with a MAP variable and exactly one of its children is in $\hat{x}_M$ if an internal OR node $n \in C_{\mathcal{T}}$ is in $\hat{x}_M$, and if an internal AND node $n \in C_{\mathcal{T}}$ is in $\hat{x}_M$ then all its OR children which denote MAP variables are in $\hat{x}_M$. Each node n in $C_{\mathcal{T}}$ can be associated with a value $v(n)$; for MAP variables $v(n)$ captures the optimal marginal MAP value of the conditioned sub-problem rooted at n, while for a sum variable it is the likelihood of the partial assignment denoted by n.

Anytime AND/OR Branch and Bound Search

AOBB-MMAP (Marinescu *et al.* 2014) explores the context minimal AND/OR search graph in a depth-first manner and therefore takes advantage of problem decomposition. During search, AOBB-MMAP keeps track of the value of the best solution found so far and uses this value and the heuristic function to prune away portions of the search space that are guaranteed not to contain the optimal solution in a typical branch and bound manner. AOBB often lacks good anytime behavior because AOBB will solve to completion all but one independent subproblems rooted at an AND node during search. This behavior was first observed by (Otten and Dechter 2011) in the context of pure MAP inference. To recover the anytime behavior, breadth rotating technique was introduced as an anytime AND/OR branch and bound search scheme that rotates through different subproblems in a round-robin manner. Breadth Rotate AND/OR Branch and Bound Search (BRAOBB-MMAP) (Lee *et al.* 2016) extends the same principle to AND/OR search space for MMAP.

Compilation into DBN

In this section, we show two encoding schemes that translate probabilistic conformant planning problem into DBN.

Direct Encoding from PPDDL into DBN

The direct encoding from PPDDL into DBN first encodes single step state transitions for each ground action schema as shown in (Younes and Littman 2004). Then, we combine all state transition models from each ground action schema into a single DBN with additional variables and deterministic relations which will be defined soon after. The semantics of the resulting graphical model is based on the serial encoding of SatPlan (Kautz *et al.* 1996), and we extended it to encode probabilistic effects by introducing effect variables for each probabilistic effect.

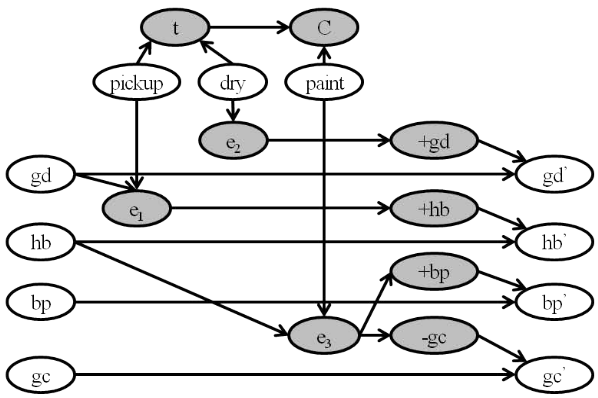
A graphical model $G_{i \to i+1} = \langle V, F \rangle$ for a single time step transition from i to $i+1$ can be obtained as follows. The set of variables V is defined as $V = S_i \cup S_{i+1} \cup A_i \cup E_{A_i} \cup S_{add_{i+1}} \cup S_{del_{i+1}}$, where the $S_i = \{s_i^j\}$ is a set of all ground predicate at i, the $A_i = \{a_i^k\}$ is a set of all ground actions at i, the $E_{A_i} = \{e_{a_i^k}\}$ is a set of effect variables for each ground action variable, the $S_{add_{i+1}} = \{+s_{i+1}^j\}$, and the $S_{del_{i+1}} = \{-s_{i+1}^j\}$.

The j-th boolean state variable s_i^j of the state $\mathbf{s}_i$, and the k-th boolean action variable a_i^k of the action $\mathbf{a}_i$ are obtained by grounding the state predicates and action schemata. A ground action consists of a precondition ϕ and an effect e, where ϕ is a conjunction of equality predicates on ground objects or literals of state variables, and the effect e 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. A simple effect is a conjunction of state variables where positive literals correspond to add list and negative literals corresponds to delete list of STRIPS formalism. A conditional effect is a pair (ϕ, e), where the ϕ is a condition defined earlier, and the e is an effect. A probabilistic effect e is a list of pairs (p_i, e_i), where p_i is the probability value for the ith 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 of an action a_i^k, we introduce an effect variable $e_{a_i^k}$ capturing the outcomes of the effect. Specifically, an effect having n outcomes maps to the effect variable with domain size $n+1$, where the additional value is reserved for the no-op. Finally, additional indicator variables $+\mathrm{s}_{i+1}^j$ and $-\mathrm{s}_{i+1}^j$ are introduced to encode addition and deletion of a state variable s_{i+1}^j.

The set of functions F are obtained by converting the following expressions into tabular forms.

1. $a_i^j \wedge \phi_i^j \leftrightarrow (e_{a_i^j} \neq \text{no-op})$, where ϕ_i^j is precondition of a_i^j
2. $\vee_{e_{a_i^j} \in E^+} (e_{a_i^j}) \leftrightarrow +s_{i+1}^j, \vee_{e_{a_i^j} \in E^-} (e_{a_i^j}) \leftrightarrow -s_{i+1}^j$
   where $E^+ = \{e_{a_i^j} | \exists v \in Dom(e_{a_i^j}) \text{ s.t. } s_{i+1}^j \in \text{add}(e_{a_i^j})\}$,
   $E^- = \{e_{a_i^j} | \exists v \in Dom(e_{a_i^j}) \text{ s.t. } s_{i+1}^j \in \text{del}(e_{a_i^j})\}$
3. $\forall_{k_1 \neq k_2} (\mathrm{e}_i^{k_1} = v_1) \wedge (\mathrm{e}_i^{k_2} = v_2) \rightarrow \neg\text{+/-}\mathrm{s}_{i+1}$,
   if$s_{i+1} \in$add/del$(e_i^{k_1} = v_1)$ and $s_{i+1} \in$add/del$(e_i^{k_2} = v_2)$
4. $\forall_j \vee a_i^j, \quad \forall_{j \neq k}\ a_i^j \rightarrow \neg a_i^k$
5. $\neg + s_{i+1}^j \wedge \neg - s_{i+1}^j \rightarrow (s_i^j \wedge s_{i+1}^j) \vee (\neg s_i^j \wedge \neg s_{i+1}^j)$

The first clauses correspond to the Conditional Probability Table (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



pickup	f(pickup)
0	1
1	1

pickup	gd	e_1	Pr(e_1|)
0	0	noop	1
0	0	hb	0
0	0	null	0
0	1	noop	1
0	1	hb	0
0	1	null	0
1	0	noop	0
1	0	hb	0.5
1	0	null	0.5
1	1	noop	0
1	1	hb	0.95
1	1	null	005

e_1	+hb	f(+hb)
noop	0	1
noop	1	0
hb	0	0
hb	1	1
null	0	1
null	1	0

+hb	hb	hb'	f(hb')
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Figure 1: Dynamic Bayesian Network for the Slippery Gripper Domain : The graphical model describes a single state transitions resulting from each action. The state variables are *gd*, *hb*, *bp*, and *gc*. The *pickup*, *dry*, and *paint* are the action variables. Additional effect variables are introduced for each ground action variables, and additional variables for encoding add effect and delete effect for each state variable are also introduced. The constraint variable C is introduced to encode mutual exclusivity of action variables, and hidden node t bounds the number of in-degree of the variable C. On the right, we show the tables that representing CPTs.

the result of executing some action as well as to deprecate inconsistent combination of effects. The fourth set of clauses encode the mutual exclusivity on action variables and the last encodes frame axioms.

The overall graphical model with time horizon T can be obtained by replicating G over T time steps. 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 $\pm s^j_{i+1}$ grow prohibitively for storing the tabular form of the CPT, so additional hidden variables were introduced to bound the maximum in-degrees.

Translation Example: Slippery Gripper Domain Consider slippery gripper domain (Kushmerick *et al.* 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* ∧ *hb* ∧ *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*.

Figure 1 shows the DBN for the slippery gripper domain. The CPT for the action variable pickup shows that the value of selecting the action is normalized to 1. The effect variable e_1 takes three values, the no-op for the case where the *pick up* action was not executed, the *+hb* for adding the state variable *hb'*, 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 CPT for the variable e_1 is tabular representation of the *pickup* action based on the precondition on the state variable *gd*. Note that the only probabilistic functions are effect CPTs from probabilistic effects. The CPT for the auxiliary variable *+hb* is a truth table that encodes *+hb* is true when the outcome of the e_1 was adding the state variable *hb'*. The last table is also a truth table that encodes the frame axiom between the state variable *hb* and *hb'*. Finally, the CPT for the node C is also a tabular representation of the mutual exclusivity constraint.

Complexity of the translation The complexity of the resulting graphical model is similar to that of serial encoding in 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 G is $O(2|as|k^p + 4|pre|k^q)$. 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 scope size of the CPTs is $max(3, s+1)$, where the s is the maximum number of a 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. In summary, the total number of variables and functions in G is $O(3|as|k^p + 2(1 + |E|)|pre|k^q)$.

Encoding from SAS+ into DBN

The second encoding scheme compiles planning problems represented by finite-domain representation (Helmert 2009) into DBN. The finite-domain representation can be obtained

instance blocks, horizon	ppddl to dbn n, a, w_c, h_c	braobb-mmap i_{best}	time (sec)	pr(G)	sas+ to dbn n, a, w_c, h_c	braobb-mmap i_{best}	time (sec)	pr(G)
2, 5	299, 40, 48, 76	10	1.56	0.703125	406, 5, 22, 64	2	1.65	0.703125
2, 8	473, 64, 72, 112	10	2990.73	0.91626	646, 8, 24, 76	14	1857.33	0.91626
2, 11	647, 88, 96, 149	16	oot	0.966007	886, 11, 24, 86	6	oot	0.943176
2, 14	821, 112, 120, 169	2	oot	0.91626	1126, 14, 28, 100	8	oot	0.91626
2, 17	995, 136, 144, 199	10	oot	0.91626	1366, 17, 28, 108	10	oot	0.91626
2, 20	1169, 160, 168, 237	2	oot	0.870117	1606, 20, 26, 103	2	oot	0.870117
3, 5	741, 90, 132, 182	6	2.53	0.079102	833, 5, 44, 85	4	0.96	0.079102
3, 8	1176, 144, 159, 251	6	5767.69	0.494385	1328, 8, 45, 125	4	4382.65	0.494385
3, 11	1611, 198, 213, 328	10	oot	0.494385	1823, 11, 46, 132	2	oot	0.494385
3, 14	2046, 252, 267, 401	10	oot	0.454834	2318, 14, 46, 146	2	oot	0.494385
3, 17	2481, 306, 326, 474	2	oot	0.395508	2813, 17, 44, 183	4	oot	0.494385
3, 20	2916, 360, 380, 545	2	oot	0.395508	3308, 20, 44, 178	6	oot	0.494385
4, 8	2185, 256, 370, 477	10	108.7	0.177979	2266, 8, 67, 164	6	55.04	0.177979
4, 9	2455, 288, 415, 520	12	5717.1	0.222473	2548, 9, 68, 188	2	2291.27	0.222473
4, 10	2725, 320, 397, 556	2	oot	0.222473	2830, 10, 68, 179	2	oot	0.222473
4, 11	2995, 352, 491, 624	2	oot	0.222473	3112, 11, 68, 214	2	oot	0.222473
4, 13	3535, 416, 541, 716	2	oot	0.222473	3676, 13, 68, 222	2	oot	0.222473
4, 15	4075, 480, 672, 841	10	oot	0.222473	4240, 15, 82, 263	2	oot	0.222473

Table 1: Experiment Results from BRAOBB-MMAP Algorithm: Each instance refers to the blocks world problem instance with specified number of blocks and time horizon. The problem statistics from direct encoding from PPDDL into DBN and SAS+ into DBN is presented, where n is the number of variables, a is the number of MAP variables (action variables), w_c is the constrained induced width, and h_c is the constrained pseudo tree height. Each group of columns corresponds to the results from two encoding schemes, where i_{best} is the best i-bound that produced the plan, $Pr(G)$ is the probability of success for the plan found. The *oot* indicates the algorithm encountered 2 hour time limit. Otherwise BRAOBB-MMAP found the optimal plan.

by transforming planning problems specified in PDDL formalism into SAS+ formalism (Bäckström and Nebel 1995) which aggregates mutually exclusive binary state variables as a multi-valued state variable and represents each ground action as a conjunction of state transitions. In classical planning, a SAT encoding based on SAS+ formalism was proposed in (Huang *et al.* 2010), and it shows superior performance compared to SatPlan (Kautz *et al.* 1996).

Similar to the direct encoding from PPDDL into DBN, the encoding from SAS+ into DBN also first compile a graphical model $G_{i\to i+1} = \langle V, F\rangle$ for a single time step from i to $i+1$, and replicate it over T time steps. The set of variables V is defined as $V = S_i \cup S_{i+1} \cup \{a_i\} \cup E_i \cup \delta^{pre} \cup \delta^{post}$, where the $S_i = \{s_i^j\}$ is a set of multi-valued state variables at time i, the a_i is a multi-valued action variable at time i, the $E_i = \{e_i^k\}$ is a set of effect variables at time i, where each effect variable e_i^k corresponds to the k-th ground action, the $\delta^{pre} = \{\delta^{pre}_{e_{kl}}\}$ is a set of precondition variables at time i, where each variable $\delta^{pre}_{e_{kl}}$ encodes the truth value of the precondition of e_{kl}, and the $\delta^{post} = \{\delta^{post}_{s_{i+1}}\}$ is a set of post condition variables at time i, where each variable $\delta^{post}_{s_{i+1}}$ encodes the transition of the state variable s_{i+1}.

The state variables s_i^j and the action variable a_i are obtained by the multi-valued state variables and state transition based ground actions that are generated by the translator in fast downward planning system (Helmert 2006). The effect variable e_i^k corresponds to the k-th ground action, which is a collection of pairs of a deterministic ground action and the probability value including the no-op action. The CPT of the e_i^k encodes the probability value of each deterministic ground action. A deterministic ground action in SAS+ formalism composed of a set of preconditions, and a set of post conditions that describes the transition of state variables. Thus, the k-th value of the action variable a_i incurs a random selection from the value of the effect variable e_i^k, where the l-th value maps to the l-th deterministic state transition of the corresponding ground action schema. The boolean variable $\delta^{pre}_{e_{kl}}$ for the precondition expression of the l-th deterministic effect e_{kl} must be true to be executed. Otherwise, we assumed that a planner would execute a no-op action. The post condition variable $\delta^{post}_{s_{i+1}}$ for the state variable s_{i+1} can be influenced by all ground actions that can change the value of s_{i+1} and the precondition variables of them. The CPT of the $\delta^{pre}_{e_{kl}}$ simply encodes the truth table of the precondition expression of e_{kl}, and the CPT of the $\delta^{post}_{s_{i+1}}$ encodes the deterministic relation over the effect variables and precondition variables affecting the state variable s_{i+1}.

Experiment Result

We evaluate search based probabilistic inference algorithms for solving probabilistic conformant planning problems. In particular, we apply BRAOBB-MMAP to find the optimal plan and BRAOBB-MAP and stochastic local search (Hutter *et al.* 2005) for pure MAP to find a suboptimal plan. The heuristic function for the search algorithm was generated by mini-bucket elimination with moment matching, WMB-MM(i), whose strength can be controlled by a parameter i-bound (Dechter and Rish 2003; Liu and Ihler 2011). Note that both algorithms for pure MAP task only produce a MAP assignment, so the probability of success was evaluated by computing the probability of evidence.

The problem instances were generated from the blocks world domain of the IPC 2006 probabilistic planning track. We modified the original domain by removing 3 *tower ac-*

instance blocks, horizon	ppddl to dbn n, a, w_c, h_c, w_u, h_u	BRAOBB-MAP time (sec)	 pr(G)	GLS+ time (sec)	 pr(G)
2, 5	299, 40, 48, 76, 17, 56	4.34	0.5625	0.33	0.5625
2, 8	473, 64, 72, 112, 17, 88	6.63	0.5625	1.82	0.5625
2, 12	647, 88, 96, 149, 17, 111	9.71	0.5625	3.63	0.74707
2, 14	821, 112, 120, 169, 17, 163	15.54	0.5625	25.45	0.823975
2, 17	995, 136, 144, 199, 17, 241	29.29	0.316406	84.71	0.922302
2, 20	1169, 160, 168, 237, 17, 292	35.92	0.624023	25.2	0.886399
3, 5	741, 90, 132, 182, 34, 129	9.49	0.079102	4.85	0.079102
3, 8	1176, 144, 159, 251, 34, 157	53.77	0.316406	26.64	0.316406
3, 11	1611, 198, 213, 328, 34, 317	128.28	0.316406	69.57	0.454834
3, 14	2046, 252, 267, 401, 34, 382	495.64	0.316406	671.34	0.415283
3, 18	2481, 306, 326, 474,36, 350	oot	na	1743.19	0.395508
3, 20	2916, 360, 380, 545, 34, 518	oot	na	126.36	10.23731
4, 8	2185, 256, 370, 477, 60, 347	106.88	0.177979	257.99	0.177979
4, 9	2455, 288, 415, 520, 61, 321	556	0.177979	146.52	0.222473
4, 10	2725, 320, 397, 556, 61, 382	418.75	0.044495	413.68	0.222473
4, 11	2995, 352, 491, 624, 61, 497	552.66	0.177979	2003.72	0.222473
4, 13	3535, 416, 541, 716, 62, 598	oot	na	186.66	0.125141
4, 15	4075, 480, 672, 841, 62, 533	oot	na	720.01	0.044495

Table 2: Experiment Results from BRAOBB-MAP and GLS+ Algorithm: In addition to the problem statistics presented in 1, w_u is the unconstrained induced width and h_u is the unconstrained pseudo tree height. Pure MAP algorithms first produce MAP assignment to all variables, and the probability of success was evaluated by computing the probability of evidence by processing action variables as evidence. The time in each column is the total time for evaluating the pure MAP and PR.

tion schemata so that the resulting domain has 4 action schemata, and we created a task of reversing the initial configuration of blocks, where the blocks are stacked on the table, and the goal is to reverse the order of the stack.

The experiment supports the time bounds up to 2 hours within 4 GB memory. In addition, constraint propagation is also employed on the conditioned subproblems to detect partial solution trees with zero probability (Marinescu 2008).

Results from Marginal MAP Inference Algorithms

Table 1 shows the result of applying BRAOBB-MMAP algorithm to two encoding schemes introduced in the earlier section. For both encoding schemes, BRAOBB-MMAP found the optimal conformant plan up to 8 time horizon for 2 and 3 blocks, and 9 time horizon for 4 blocks. For the longer time horizon, BRAOBB-MMAP reported suboptimal plans from the anytime solutions.

In addition, we can see that the encoding scheme based on SAS+ improves the performance of BRAOBB-MMAP. Especially, the constrained induced width and the constrained pseudo tree height from the encoding based on SAS+ is much better than direct encoding from PPDDL.

Results from pure MAP Inference Algorithms

Table 2 shows the result of applying pure MAP algorithms, BRAOBB-MAP and GLS+ to the problem instances generated by direct encoding from PPDDL.

The first group shows the result of applying BRAOBB-MAP in conjunction with AND/OR search algorithm for computing the probability of evidence. Such a combination can be used to find a probabilistic conformant plan with a proper time horizon expanding strategy, when we consider the query $\langle P, \theta \rangle$. (In this paper, a strategy for expanding time horizon is out of scope and we assumed that such good time horizons were given in advance.) We can see that the BRAOBB-MAP does not always improves the plan with longer time horizons since the values of Pr(G) are not monotonically increased with time horizon.

The second group summarizes the result of applying GLS+ (Hutter *et al.* 2005) algorithm for finding a MAP assignment and evaluating the probability of success. Here, we show the highest probability of success obtained from each problem instance after evaluating suboptimal solutions generated by GLS+. Thus, the time spent for finding a conformant plan is highly favorably skewed, and should yield a potential integration, while the times should be increased. Overall GLS+ can reach longer time horizon than BRAOBB-MAP and it could find relatively good solution in short time bounds.

Comparison with Probabilistic Fast Forward

In Table 3, we compare the results of applying BRAOBB-MMAP and BRAOBB-MAP with Probabilistic Fast Forward(Domshlak and Hoffmann 2007). Probabilistic Fast Forward (PFF) finds a conformant plan with respect to the task $\langle P, \theta \rangle$ by heuristic forward search in a belief state space. PFF represents belief states by DBN $N_{b_{\bar{a}}}$ for some sequence of actions $\bar{a}$, and the DBN is compiled into weighted CNFs to retrieve probabilistic queries via weighted model counting. In our encoding schemes, all the action variables are present in a single DBN, so that the probabilistic inference can directly query the action variables after marginalizing the rest of the variables. On the other hand, each DBN for the belief states $b_{\bar{a}}$ in PFF 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.

For problem instances having 2 blocks, PFF was the best until θ = 0.5. For the blocks world instances with 3 blocks,

BW2	θ	0.1	0.15	0.2	0.35	0.4	0.5	0.7	0.8	0.85
	PFF	**0.06 (4)**	**0.04 (4)**	**0.05 (4)**	**0.05 (4)**	**0.06 (4)**	**0.05 (4)**	err	err	err
	BRAOBB-MMAP	1.55(5)	1.55(5)	1.55(5)	1.55(5)	1.55(5)	1.55(5)	1.55(5)	25.07 (6)	247.08 (7)
	BRAOBB-MAP	4.12 (3)	4.22 (4)	4.22 (4)	4.22 (4)	4.22 (4)	4.22 (4)	8.31 (10)	65.21 (30)	65.26 (30)
	GLS+	0.12 (3)	0.22 (4)	0.22 (4)	0.22 (4)	0.22 (4)	0.22 (4)	**0.71 (7)**	**5.25 (14)**	18.26 (17)
BW3	θ	0.1	0.15	0.2	0.35	0.4	0.5	0.7	0.8	0.85
	PFF	**0.07 (6)**	**0.06 (6)**	**0.09 (6)**	**0.12 (7)**	err	err	err	err	err
	BRAOBB-MMAP	65.79 (6)	65.79 (6)	65.79 (6)	212.01 (7)	5647.65 (8)	oot	oot	oot	oot
	BRAOBB-MAP	14.47 (6)	14.47 (6)	14.47 (6)	2069.56 (15)	oot	oot	oot	oot	oot
	GLS+	18.88 (10)	18.88 (10)	18.88 (10)	18.88 (10)	**69.57 (11)**	oot	oot	oot	oot
BW4	θ	0.1	0.15	0.2	0.35	0.4	0.5	0.7	0.8	0.85
	PFF	**0.15 (8)**	**0.14 (8)**	err	err	err	err	err	err	err
	BRAOBB-MMAP	90.32 (8)	90.32 (8)	6489.2 (9)	oot	oot	oot	oot	oot	oot
	BRAOBB-MAP	106.88 (8)	106.88 (8)	oot	oot	oot	oot	oot	oot	oot
	GLS+	146.52 (9)	146.52 (9)	**146.52 (9)**	oot	oot	oot	oot	oot	oot

Table 3: Comparison with Probabilistic Fast Forward: Each row is grouped by the number of blocks, from 2 blocks to 4 blocks. The query to the planner is $\langle P, \theta \rangle$, and the length of the plan found by each algorithm is shown inside the parentheses next to the total time.

PFF performed the best until θ = 0.35. PFF and BRAOBB-MAP failed to find a plan with higher thresholds while BRAOBB-MMAP and GLS+ did. In case of problem instances with 4 blocks, PFF was able to find a plan until θ = 0.15. Similar to the previous cases, it also failed at larger threshold.

We observe that BRAOBB-MMAP were competitive and were able to generate as good, or better threshold plans than PFF. Yet, BRAOBB-MMAP did so using far more time in most cases. There were cases that PFF failed. For instance with 2 blocks, PFF was unable to generate plans with threshold 0.7 or 0.8 where others did. BRAOBB-MMAP accomplished this using 1.55 sec and 25.07 seconds respectively, while BRAOBB-MAP and even GLS+ did so with less time. Still, MMAP provided shortest plan for the task, as expected.

Conclusion

The paper explores the potential of probabilistic inference algorithms to conformant planning. First, we explored two formulations of probabilistic conformant planning. The first seeks the most likely plan that achieves the goal. The second assumes a constant probability threshold and seeks a plan whose probability of success exceeds that threshold. In both formulations shorter plans are preferred. Then, we provided two encoding schemes that converts probabilistic conformant planning problem into a graphical model such that a most likely plan maps to the most likely assignment to the corresponding action variables of the graphical model. Therefore, a most likely conformant plan can be generated by solving what is known as a marginal MAP task.

We applied several of the most competitive probabilistic algorithms for solving the marginal MAP, pure MAP, and PR (Probability of Evidence), in order to generate exact and approximate algorithms for conformant planning and tested their performance empirically. Our empirical evaluation, though preliminary, highlights the potential and the current boundaries of this methodology.

The MMAP algorithms we experimented find the most likely plan for a given horizon and thus can yield shortest plan that achieves a particular threshold when applied to growing horizons. As the horizon extend such algorithms can find plans having increasing probability of success. Thus, our MMAP algorithm provides an optimal scheme (perhaps the first of its kind) that can solve the conformant planning problem for the formulation query $\langle P, T \rangle$.

Our empirical evaluation illustrated the algorithm's ability to solve optimally relatively small planning problems having a limited horizon. Competing approximate algorithms show somewhat a wider but still limited range of solving ability, yet they are often significantly suboptimal. The specific algorithm we used is based on a recently developed marginal map scheme that proved to be currently one of the most effective algorithms for this task. Note that MMAP not only provides a best quality plan, but also proved its optimality. The approximate algorithms generate a plan and its probability of success for a given horizon. The idea is to generate a sequence of candidates plan somehow and evaluate their probability of success by solving (exactly) the PR task. The generated plans having the highest probability is returned. Namely compute candidate plans somehow and evaluate their probability of success by solving the summation problem exactly. We tested 2 such schemes, one that solve the pure MAP (a far easier task than MMAP) task optimally and the other that generates suboptimal MAP assignments by stochastic local search solver, GLS+.

Overall, the marginal MAP solver, BRAOBB-MMAP, produced shortest plans with the highest probability of success, whenever it could find a solution. Yet the suboptimal solvers were far faster. Note that the only scheme that proves optimality is MMAP. Comparing the two new approximate schemes, BRAOBB-MAP with PR vs. GLS+ with PR, the latter seems superior as it produced plans with higher probability of success due to its ability to solve instances having a larger horizon. Comparing with Probabilistic Fast Forward (PFF), our probabilistic inference based approaches became competitive as the threshold grew. Indeed, PFF was overall fastest compared to all solvers, when it could find a solution. Nevertheless, in a few cases the (approximate) probabilistic inference approaches were able to find plans for thresholds for which PFF completely failed.

Our results are clearly preliminary and should be considered as initial attempts in this direction. Still, even at this preliminary stage we see the potential of tapping into probabilistic inference algorithms for the conformant planning task. This exploration already highlights that in the future we should develop more efficient *anytime* marginal map algorithms for the task, that we should explore more effective translations, and that we should consider partial horizons evaluation that incrementally increase in the style of PFF.

References

Christer Bäckström and Bernhard Nebel. Complexity results for sas+ planning. *Computational Intelligence*, 11(4):625–655, 1995.

Rina Dechter and Robert Mateescu. AND/OR search spaces for graphical models. *Artificial Intelligence*, 171(2–3):73–106, 2007.

Rina Dechter and Irina Rish. Mini-buckets: A general scheme of approximating inference. *Journal of ACM*, 50(2):107–153, 2003.

Carmel Domshlak and Jörg Hoffmann. Probabilistic planning via heuristic forward search and weighted model counting. *Journal of Artificial Intelligence Research*, 30:565–620, 2007.

Malte Helmert. The fast downward planning system. *Journal of Artificial Intelligence Research*, 26:191–246, 2006.

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

Ruoyun Huang, Yixin Chen, and Weixiong Zhang. A novel transition based encoding scheme for planning as satisfiability. In *AAAI Conference on Artificial Intelligence*, 2010.

Frank Hutter, Holger H Hoos, and Thomas Stützle. Efficient stochastic local search for mpe solving. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2005.

Henry Kautz, David McAllester, and Bart Selman. Encoding plans in propositional logic. *KR*, 96:374–384, 1996.

Igor Kiselev and Pascal Poupart. Policy optimization by marginal-map probabilistic inference in generative models. In *International conference on Autonomous agents and multi-agent systems*, pages 1611–1612, 2014.

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

Junkyu Lee, Radu Marinescu, and Rina Dechter. Applying marginal map search to probabilistic conformant planning: Initial results. In *International Workshop on Statistical Realtional AI*, 2014.

Junkyu Lee, Radu Marinescu, Rina Dechter, and Alexander Ihler. From exact to anytime solutions for marginal MAP. In *to appear in AAAI Conference on Artificial Intelligence*, 2016.

Michael L. Littman, Stephen M. Majercik, and Toniann Pitassi. Stochastic boolean satisfiability. *Journal of Automated Reasoning*, 27(3):251–296, 2001.

Qiang Liu and Alexander Ihler. Bounding the partition function using Hölder's inequality. In *International Conference on Machine Learning (ICML)*, pages 849–856, 2011.

Radu Marinescu, Rina Dechter, and Alexander Ihler. AND/OR search for marginal MAP. In *Uncertainty in Artificial Intelligence (UAI)*, pages 563–572, 2014.

Radu Marinescu, Rina Dechter, and Alexander Ihler. Pushing forward marginal MAP with best-first search. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 696–702, 2015.

Radu Marinescu. *AND/OR Search Strategies for Combinatorial Optimization in Graphical Models*. PhD thesis, University of California, Irvine, 2008.

Lars Otten and Rina Dechter. Anytime AND/OR depth-first search for combinatorial optimization. In *International Symposium on Combinatorial Search (SoCS)*, pages 117–702, 2011.

Håkan LS Younes and Michael L Littman. Ppddl 1.0: The language for the probabilistic part of ipc-4. In *International Conference on Automated Planning and Scheduling (ICAPS)*, 2004.