Factored Probabilistic Belief Tracking

Blai Bonet
Universidad Simón Bolívar
Caracas, Venezuela
bonet@ldc.usb.ve

Hector Geffner
ICREA & Universitat Pompeu Fabra
Barcelona, SPAIN
hector.geffner@upf.edu

Abstract

The problem of belief tracking in the presence of stochastic actions and observations is pervasive and yet computationally intractable. In this work we show however that *probabilistic beliefs can be maintained in factored form exactly and efficiently* across a number of causally closed beams, when the state variables that appear in more than one beam obey a form of *backward determinism*. Since computing marginals from the factors is still computationally intractable in general, and variables appearing in several beams are not always backward-deterministic, the basic formulation is extended with two approximations: forms of *belief propagation* for computing marginals from factors, and *sampling* of non-backward-deterministic variables for making such variables backward-deterministic *given their sampled history.* Unlike, Rao-Blackwellized particle-filtering, the sampling is not used for making inference *tractable* but for making the factorization *sound*. The resulting algorithm involves sampling *and* belief propagation or just one of them as determined by the structure of the model.

1 Introduction

Keeping track of beliefs when actions and sensors are probabilistic is crucial and yet computationally intractable, with exact algorithms running in time that is exponential in the number of state variables in the worst case. Current approaches rely on samples for approximating probabilistic beliefs by sets of particles [Kanazawa *et al.*, 1995; Thrun *et al.*, 2001] or decompositions where joint beliefs are approximated by products of smaller local beliefs [Boyen and Koller, 1998]. Particle filtering methods however may require too many particles even in the Rao-Blackwellized (RB) version [Murphy, 1999; Doucet *et al.*, 2000], while decomposition approaches may result in poor approximations.

In this work we take a different approach and show that probabilistic beliefs can be maintained in *factored form exactly and efficiently across a number causally closed beams* when the state variables that appear in more than one beam obey a form of *backward determinism* by which the value of a variable at time t is determined by the value of the variable at time $t + 1$, the history, and the prior beliefs. Since computing marginals from a factorized representation is computationally hard in general, and variables appearing in several beams are not always backward-deterministic, the basic formulation is extended through two approximations: forms of *belief propagation* for computing marginals from the factorized representation [Pearl, 1988], and *sampling* of non-backward-deterministic variables for making them backward-deterministic *given their sampled history.* This last part is similar to Rao-Blackwellized particle-filtering with one crucial difference: sampling is not introduced for making inference *tractable* but for making the factorization *sound*. Like the method of Boyen and Koller [1998], the algorithm maintains global beliefs in terms of smaller local factors but the scope of these local factors is determined by the structure of the model and variables usually appear in many factors.

We call the general algorithm *probabilistic beam tracking* (PBT) as it is the probabilistic version of the *beam tracking* scheme of Bonet and Geffner [2014] that deals with beliefs represented as *sets* of states rather than probability distributions. When the size of the beams and the number of particles are bounded, PBT runs in *polynomial time*. In addition, when the structure of the beams results in factored representations that are *acyclic* and where factors share variables that are backward-deterministic only, the algorithm is *exact*.

As an illustration, the 1-line SLAM problem of Murphy [1999] that involves an agent that moves along a line sensing the color of each cell i, results in factors B_i of size two, where one variable represents the agent location, and the other, the color of cell i. PBT reduces then to RB particle-filtering, as the variable that appears in more than one beam (the agent location) is not backward deterministic and must be sampled, leaving each beam with a single, unique variable. On the other hand, if the problem is modified so that the color sensed in a cell depends on the color of the two surrounding cells, PBT would still sample the agent location variable, but in addition would keep track of factors of size three representing the color of each cell and the color of the two surrounding cells. Inference over these factors can be done exactly by the jointree algorithm, as the treewidth of the factor graph is bounded and small [Darwiche, 2014], or approximately but more efficiently in general, by belief propagation. In all cases, the formulation determines the scope of the factors (the

beams) and the set of variables that need to be sampled from the structure of the model.

We begin by discussing the background and related work, and then present the model, the structure, and the beams that follow from them. Next, we derive the equations for factorized belief tracking in *belief decomposable* systems and extend the formulation over non-decomposable systems. We conclude with an approximation algorithm for computing marginals, some experimental results, and discussion.

2 Background and Related Work

In the flat model, a state s is a value assignment to a set of state variables, actions a affect the state through transition probabilities $\mathbb{P}(s'|s,a)$, and observation tokens o provide partial information about the resulting state s' through sensing probabilities $\mathbb{P}(o|s',a)$. Given the prior $\mathbb{P}(s)$, the target belief $\mathbb{P}(s|h)$, where $h = \langle a_0, o_0, \ldots, a_i, o_i\rangle$ is an interleaved sequence of actions and observations, is characterized inductively as $\mathbb{P}(s|h) = \mathbb{P}(s)$ for empty h, $\mathbb{P}(s|h,a) = \sum_{s'} \mathbb{P}(s|s',a) \times \mathbb{P}(s'|h)$, and $\mathbb{P}(s|h,a,o) = \alpha\, \mathbb{P}(o|s,a) \times \mathbb{P}(s|h,a)$ where $\alpha = 1/\mathbb{P}(o|h,a)$ is a normalizing constant.

Keeping track of beliefs using this representation is exponential in the number of state variables. Approaches have thus been developed to exploit problem structure. This structure is often made explicit through the language of *Bayesian networks* [Pearl, 1988]. However, while the posterior $\mathbb{P}(s|h)$ can be obtained from a Dynamic Bayesian network (DBN) with as many slices as time steps [Dean and Kanazawa, 1989], *exact inference* over such networks is hard (yet see [Vlasselaer *et al.*, 2016]), so approximate inference schemes have been pursued instead [Murphy, 2002]. In the method of Boyen and Koller [1998], global joint beliefs are approximated as products of smaller local beliefs, while in particle filtering methods (PF), global beliefs are formed from a set of samples [Kanazawa *et al.*, 1995; Thrun *et al.*, 2001; Koller and Friedman, 2009]. The methods are related to well known approximation techniques for general Bayesian networks like the mini-bucket approximation [Dechter and Rish, 2003], (sampled) cut-set conditioning [Pearl, 1988], and restricted forms of belief propagation [Murphy and Weiss, 2001].

Our work is related to the ideas underlying these methods but it does not build explicitly on them. It is indeed a generalization of the beam tracking (BT) method for keeping track of beliefs given by *sets of states* as opposed to probability distributions [Bonet and Geffner, 2014]. *Probabilistic beam tracking* is aimed at combining the effectiveness of BT with the ability to handle noisy actions and sensing.

3 Model, Structure, and Beams

Model. The set of all *state variables* is denoted as X. A state x defines a value for each state variable. In general, subsets of variables are denoted with uppercase letters and their values with lowercase letters. Actions a affect the state stochastically with given transition probabilities $tr(x'|x,a)$. The set of all *observation variables* is denoted with O and lowercase o denotes an observation; i.e. a value for each of the observation variables. The sensor model is also Markovian with probabilities $q(o|x,a)$. The joint prior is $\mathbb{P}(x)$.

Histories and Beliefs. A history or execution h is an interleaved sequence of action and observations that begins with an action. A history is complete if it is empty or ends with an observation. The state and observation variables at time t are denoted as X^t and O^t respectively. The observation o_t in the history h encodes the value of the observation variables O^t. For each history h, either complete or incomplete, there is a probability measure $\mathbb{P}_h$ over the events defined by the random variables associated with h. We abuse notation by writing $\mathbb{P}(A|h)$ instead of $\mathbb{P}_h(A)$, and $\mathbb{P}(A|a,h)$ instead of $\mathbb{P}_{ha}(A)$. For a complete execution h for t time steps, the joint over the state variables X^t is denoted by $B^h(x) \doteq \mathbb{P}(x|h)$. The *prior* belief for the *empty* history h is $B^h(x) = \mathbb{P}(x)$.

Causal Structure and 2-DBN. The transition and sensing probabilities are given by a 2-slice DBN whose nodes V, V', and O stand for the state variables before and after the action, and for the observation variables. The transition and sensing probabilities (parameters) are $P(V'|pa(V'),a)$ and $P(O|pa(O),a)$ where the parents $pa(V')$ of V' are among the V-variables, and the parents $pa(O)$ of O are among the V'-variables. A variable W is a *cause* of V if W or W' is a parent of V in the 2-DBN, and W is *causally relevant* to V if it is a cause of V or is causally relevant to a cause of V.

Beams. A *beam* is a non-empty subset of state variables that is *causally closed*; i.e., if a variable belongs to the beam, its causes must belong to the beam as well. A *collection of beams* is *complete* if each state variable is included in some beam and for each observation variable, its set of causes is included in some beam. The size of a beam is the number of variables that it contains, and a complete collection of beams is *minimal* if no beam can be replaced by a collection of beams of smaller size. There is actually a unique minimal complete collection of beams $\mathcal{B}$ that can be constructed as follows: if B_i is the set of state variables that are causally relevant to a state variable V_i or to an observation variable O_i, then $\mathcal{B}$ is the collection of beams B_i with the duplicate beams removed, and the beams B_k properly contained in other beams removed as well. By default, we assume this beam structure $\mathcal{B}$. If we enumerate the beams in this structure as $B_1, \ldots, B_m$, we will refer to beam B_i also by its index i. This beam structure is the same as the one in the non-probabilistic formulation [Bonet and Geffner, 2014] where the *causal width* of the problem is defined as the size of the largest beam. This structural measure is important because beam tracking runs in time that is *exponential in the causal width*.[1] The same will be true for probabilistic beam tracking. Many problems of interest can be formulated so that their causal width is bounded and small.

Example: In Minesweeper, whether in the normal or noisy version, there are hidden state variables V_i encoding whether there is a mine at cell i, and observation variables O_i that

[1] Actually, variables that are *determined*, meaning that their initial value is known and can be affected by deterministic actions only, do not add to the causal width [Bonet and Geffner, 2014].

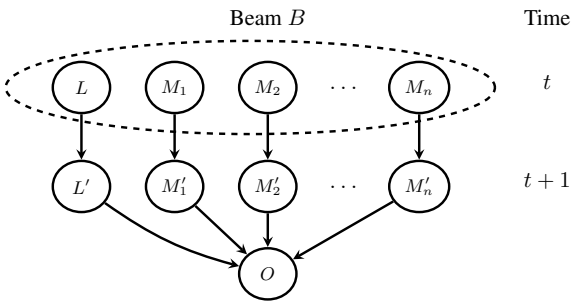

Figure 1: 2-slice DBN and beam structure for the direct formulation of the 1-line SLAM problem. The observation probabilities satisfy $\mathbb{P}(o|\ell, m_1, \ldots, m_n, a) = q(o|\ell, m_\ell)$ that corresponds to a form of context-specific independence. This independence is not exploited and results in a beam structure that contains a single beam B with all the $n+1$ state variables.

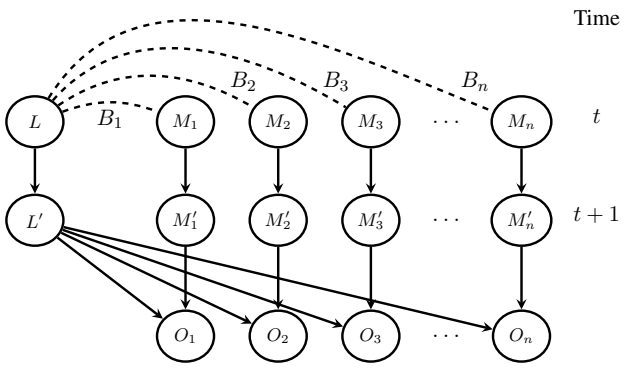

Figure 2: 2-slice DBN and beam structure for the 1-line SLAM problem formulated with multiple "dummy" observation variables O_i that are set to the value of O. In this model, there is one beam B_i for each cell i that contains two variables only, L and M_i, for a causal width of 2.

after probing cell i reveal the total number of mines at the 8 surrounding cells and V_i itself. The causes of O_i are the 9 cell variables V_k that have no further causes (observation variables for border cells have fewer causes). The minimal complete beam structure is given by beams B_i, one for each cell i, each of size no greater than 9. Thus, while tracking beliefs in Minesweeper is NP-hard [Kaye, 2000], the causal width of the problem is 9.

Example: In the direct formulation of Murphy's 1-line SLAM problem mentioned above, there are color variables M_i for each cell i, an agent location variable L, and one observation variable O. The observation o that results after applying an action a is determined by the probabilities $q(o|\ell, m_1, \ldots, m_n, a) = q(o|\ell, m_\ell)$ that encode a form of context-specific independence [Boutilier *et al.*, 1996] in which the observation o depends only on the color m_ℓ of the cell ℓ for $L = \ell$. In this representation, all the state variables L and $\{M_i\}_i$ are causally relevant to the observation variable O, determining a beam structure with a single beam B of size $n + 1$ (Fig. 1). It is possible, however, to take advantage of context-specific independence to reformulate the model so that its *causal width becomes bounded and small*. For this, it suffices to split the single observation variable O into n observation variables $O_1, \ldots, O_n$, one for each cell i, such that the parents of variable O_i are the variables L and M_i only, and to set the probability $q_i(o_i|\ell, m_i)$ for variable O_i equal to $q(o_\ell|\ell, m_\ell)$ when $\ell = i$ and to $1/2$ (a normalizing constant) when $\ell \neq i$. The value of all the "artificial" observation variables O_i at time t is set to the value of the real observation variable O at time t; that is, if the observation O has value o at time t then all variables O_i are set to o at time t. The sensor model of the reformulated task is defined as $q(\langle o_1, \ldots, o_n\rangle|\ell, m_1, \ldots, m_n, a) = \prod_{i=1}^{n} q_i(o_i|\ell, m_i)$. While the two models are equivalent, the first has one beam of size $n + 1$, while the latter has n beams B_i of size 2, each containing the agent location variable L and the cell variable M_i, $i = 1, \ldots, n$, for a *causal width* of 2 (Fig. 2).

Internal and External Variables. A variable that appears in more than one beam is called *external*, while one that appears in one beam only is called *internal*. The internal and external variables for beam X_j are denoted as Y_j and Z_j respectively. The set X of all (state) variables is partitioned as $X = YZ$ where Y are all the internal variables and Z are all the external variables. In the second formulation of 1-line SLAM, the color variables are all internal, and the agent location variable is external. In Minesweeper, all variables are external.

Factored Model. Given the beam structure determined by the 2-DBN structure, the transition and sensing probabilities $tr(x'|x, a)$ and $q(o'|x', a)$ can be factorized as $tr(x'|x, a) = tr(y'z'|x, a) = tr(y'|z', x, a)tr(z'|x, a)$ with $tr(y'|z', x, a) = \prod_j tr_j(y'_j|z'_j, x_j, a)$, $tr(z'|x, a) = \prod_j tr_j(z'_j|x_j, a)$, and $q(o'|x', a) = \prod_j q_j(o'_j|x'_j, a)$, where j ranges over the beam indices. All the tr_j and q_j probabilities are determined by the conditional probabilities $P(V'|pa(V'), a)$ in the 2-DBN.[2] It is also assumed without loss of generality that the prior belief $B^h(x)$ for the empty history factorizes across the beams as $\prod_j B^h_j(x_j)$. When this is not the case, the model can be extended with an extra action-observation pair that must start any non-empty history.

4 Tracking in Belief Decomposable Systems

Given the model structure and beams, our task is to show that the posterior joint beliefs $B^h(x) = B^h(X^t = x)$ for histories $h = \langle a_0, o_0, \ldots, a_t, o_t\rangle$ can be expressed as the normalized product of *belief factors* $\prod_j B^h_j(x_j)$, one for each beam j, that are tracked independently. For this, we assume a form of *determinism* over the state variables that appear in more than

[2] If Y_j is $\{V_1, \ldots, V_k\}$ where the variables V_i are ordered topologically, $tr_j(y'_j|z'_j, x_j, a)$ is $\prod_{i=1}^{k} P(V'_i|pa(V'_i), a)$. Associating each external variable V with the smallest j such that B_j contains V, and each observation variable W with the smallest j such that B_j contains all of its causes, $tr_j(z'_j|x_j, a)$ factorizes as $\prod_{i=1}^{k} P(V'_i|pa(V'_i), a)$, where $V_1, \ldots, V_k$ are the external variables in Z_j associated with the beam j in topological order, and $q_j(w'|x'_j, a)$ is $P(w'|pa(w'), a)$ when the observation variable W is associated with the beam j, and else $q_j(w'|x'_j, a) = 1$.

one beam; namely, the external variables. The following key definition is from Bonet and Geffner [2014]:

Definition 1 (Backward Determinism) *A state variable V is* backward deterministic *if the value of V at any time t is determined by the value of V at time $t+1$, the action at time t, the history h up to time t, and the priors.*

Notice that static variables and variables that are initially fully known and are affected by deterministic actions only are backward deterministic, as are the variables that are fully observable and the variables V affected by deterministic actions that map different values of V into different values of V' [Amir and Russell, 2003]. When V is backward deterministic, we write $\mathcal{R}_a(v|h)$ to denote the value of V at time t that is determined by the value v of V at time $t+1$, the complete history h up to time t, the next action a, and the priors. We refer to $\mathcal{R}_a(v|h)$ as the *regression* of $V = v$ given h, a, and the priors (left implicit in the notation).

Definition 2 (Belief Decomposable Model) *A model is* belief decomposable *when all the external variables are backward deterministic.*

It will be convenient to use the abbreviation BD to refer to both *belief decomposable models* and to *backward deterministic* variables. The notation $\mathcal{R}_a(z|h)$ for the set Z of external BD variables denotes the regression of the value vector z through h and a. If the value v for a variable or set of BD variables V is impossible given h and a, we write $\mathcal{R}_a(z|h) = \bot$. Clearly, $\mathcal{R}_a(z|h) = \bot$ iff $\mathcal{R}_a(z_j|h) = \bot$ for some beam j, and $\mathcal{R}_a(z|h)_j = \mathcal{R}_a(z_j|h)$.

4.1 Equations for the Belief Factors

We want to show that the distribution $B^h(x) = \mathbb{P}(x|h)$ after history h for a BD model is the normalized product of factors $B^h_j(x_j)$:

$$B^h(x) \;=\; \beta \prod_j B^h_j(x_j) \tag{1}$$

where $B^h_j(x_j)$ denotes the belief factor over the variables in beam j, and $\beta = \beta(h)$ is a normalization factor that only depends on h. We show this inductively by using the assumption of factorized priors and by expressing the factors that define the joint belief $B^{h'}(x)$ for $h' = \langle h, a, o\rangle$ in terms of the factors that define the joint belief $B^h(x)$, where a is an action and o an observation such that $P(o|a,h) > 0$. The factors $B^h_j(x_j)$ for the empty history h are given.

Let $x' = y'z'$ be a valuation for X^{t+1}. Using Bayes' rule, the posterior can be expressed as

$$\mathbb{P}(x'|o,a,h) \;=\; \alpha\, \mathbb{P}(o|x',a,h)\, \mathbb{P}(x'|a,h) \tag{2}$$

where $\alpha = 1/\mathbb{P}(o|a,h)$ is a normalizing constant, and the second term is

$$\mathbb{P}(x'|a,h) \;=\; \textstyle\sum_y \mathbb{P}(y'|z',y,a,h)\, \mathbb{P}(z',y|a,h)\,. \tag{3}$$

Assume now that $\mathcal{R}_a(z'|h) \neq \bot$. Using backward determinism and factored transitions, the first term in (3) becomes:

$$\begin{aligned}\mathbb{P}(y'|z',y,a,h) \;&=\; tr(y'|z',y,\mathcal{R}_a(z'|h),a) &(4)\\ &=\; \textstyle\prod_j tr_j(y'_j|z'_j,y_j,\mathcal{R}_a(z'|h)_j,a)\,. &(5)\end{aligned}$$

For the second term, we use the inductive hypothesis:

$$\begin{aligned}\mathbb{P}(z',y|a,h) \;&=\; \mathbb{P}(z',y,\mathcal{R}_a(z'|h)|a,h) &(6)\\ &=\; \mathbb{P}(z'|y,\mathcal{R}_a(z'|h),a,h)\, \mathbb{P}(y,\mathcal{R}_a(z'|h)|a,h) &(7)\\ &=\; tr(z'|y,\mathcal{R}_a(z'|h),a)\, \mathbb{P}(y,\mathcal{R}_a(z'|h)|h) &(8)\\ &=\; tr(z'|y,\mathcal{R}_a(z'|h),a)\, \beta \textstyle\prod_j B^h_j(y_j,\mathcal{R}_a(z'|h)_j) &(9)\\ &=\; \beta \textstyle\prod_j tr_j(z'_j|y_j,\mathcal{R}_a(z'_j|h),a)\, B^h_j(y_j,\mathcal{R}_a(z'_j|h))\,. &(10)\end{aligned}$$

Substituting these expressions back into (3), abbreviating $\mathcal{R}_a(z'_j|h)$ as $\mathcal{R}(z'_j)$, and using $Y_i \cap Y_j = \emptyset$ for $i \neq j$:

$$\begin{aligned}&\mathbb{P}(x'|a,h)\,\beta^{-1} \;=\\ &=\; \textstyle\sum_y \prod_j tr_j(y'_j,z'_j|y_j,\mathcal{R}(z'_j),a)\, B^h_j(y_j,\mathcal{R}(z'_j)) &(11)\\ &=\; \textstyle\prod_j \sum_{y_j} tr_j(y'_j,z'_j|y_j,\mathcal{R}(z'_j),a)\, B^h_j(y_j,\mathcal{R}(z'_j)) &(12)\\ &=\; \textstyle\prod_j \sum_{y_j} tr_j(x'_j|y_j,\mathcal{R}(z'_j),a)\, B^h_j(y_j,\mathcal{R}(z'_j))\,. &(13)\end{aligned}$$

Finally, from the factorization of the observations

$$\mathbb{P}(o|x',a,h) \;=\; q(o|x',a) \;=\; \textstyle\prod_j q_j(o_j|x'_j,a)\,, \tag{14}$$

and the inductive hypothesis, the factors $B^{h'}(x'_j)$ become:

$$\begin{aligned}B^{h'}_j(y'_j,z'_j) \;&=\; \alpha'\, q_j(o_j|y'_j,z'_j,a) \times\\ &\textstyle\sum_{y_j} tr_j(x'_j \mid y_j,\mathcal{R}_a(z'_j|h),a)\, B^h_j(y_j,\mathcal{R}_a(z'_j|h))\end{aligned} \tag{15}$$

when $\mathcal{R}_a(z'|h) \neq \bot$, and $B^{h'}_j(y'_j,z'_j) = 0$ otherwise. That is, the belief factors B^h_j are progressed and filtered *independently* for each beam j. The complexity of updating each belief factor, i.e., mapping B^h_j into $B^{h'}_j$ for $h' = \langle h,a,o\rangle$, is exponential in the beam j size, and more precisely, in the number of internal variables in beam j.

While the factors B^h_j determine the joint distribution B^h, the computation of marginals from such a joint is hard in general. We will come back to this issue but focus now on extending the formulation to models where not all external variables are backward deterministic. For this, like in Rao-Blackwellized PF methods, we sample such variables to make them backward deterministic given their sampled history.[3]

5 Tracking in Non-Decomposable Systems

We assume now that the set X of state variables is partitioned into three sets YZU where Y stands for the internal variables (variables appearing in one beam only), Z for the external variables that are backward deterministic, and the new set U for the external variables that are not BD and need to be sampled. If the context above was provided by the history h of actions and observations, the new context is provided by h

[3]The assumption of backward determinism appears in (15) through the use of the regressions $\mathcal{R}_a(z'_j|h)$. A different type of approximation can be defined by replacing such regressions by values z_j that are summed over with weights given by $\mathbb{P}(z_j|z'_j,h,a)$. When the BD assumption holds, these weights are either 0 or 1, and the new formula reduces to the old formula. When BD does not hold, the new formula encodes an approximation. We tested this approximation empirically, but in the examples considered, it does not run faster than the particle-based approximation and its quality does not appear to be better either.

and the sampled history $\bar{u}$ of the U variables. The expression $\mathcal{R}_a(z', u'|\bar{u}', h)$ denotes the pair of values $\mathcal{R}_a(z'|h)$ and $\mathcal{R}_a(u'|\bar{u}')$ where the latter denotes u, the value preceding the last value u' in the sampled history $\bar{u}'$. The joint over the Y and Z variables can be expressed as

$$\mathbb{P}(y, z|h) \;=\; \textstyle\sum_{\bar{u}} \mathbb{P}(y, z|\bar{u}, h)\, \mathbb{P}(\bar{u}|h) \tag{16}$$

which can be approximated as:

$$\mathbb{P}(y, z|h) \;\approx\; \textstyle\sum_{i=1}^{N} \mathbb{P}(y, z|\bar{u}_i, h) \tag{17}$$

where the histories $\bar{u}_i$ are sampled with probability $\mathbb{P}(\bar{u}_i|h)$. It is often convenient however to sample $\bar{u}_i$ with a different distribution $\pi(\bar{u}|h)$, called the *sampling* or *proposal distribution*, as long as no possible U history is made impossible. In this method, called *importance-based sampling* [Bishop, 2007] the approximation becomes:

$$\mathbb{P}(y, z|h) \;\approx\; \alpha \textstyle\sum_{i=1}^{N} w_i \times \mathbb{P}(y, z|\bar{u}_i, h) \tag{18}$$

where α is a normalizing constant, and the weights w_i are

$$w_i \;=\; \mathbb{P}(\bar{u}_i|h)/\pi(\bar{u}_i|h)\,. \tag{19}$$

Provided with the sampled histories $\bar{u}$, it can be shown that $\mathbb{P}(y, z|\bar{u}, h)$ in (16) becomes:

$$\mathbb{P}(y, z|\bar{u}, h) \;=\; B^h(y, z|\bar{u}) \;=\; \beta \textstyle\prod_j B^h_j(y_j, z_j|\bar{u}) \tag{20}$$

where the factors $B^h_j(\,\cdot\,|\,\bar{u})$ can be progressed to $h' = \langle h, a, o\rangle$ and $\bar{u}' = \langle \bar{u}, u'\rangle$ as:

$$\begin{aligned} &B^{h'}_j(y'_j, z'_j|\bar{u}') \;=\; \alpha''\, q_j(o_j|x'_j, a) \times \\ &\textstyle\sum_{y_j} tr_j(x'_j|y_j, \mathcal{R}_a(z'_j|h), u_j, a)\, B^h_j(y_j, \mathcal{R}_a(z'_j|h)|\bar{u}) \end{aligned} \tag{21}$$

when $\mathcal{R}_a(z'_j|h) \neq \bot$, else $B^{h'}_j(y'_j, z'_j|\bar{u}') = 0$. This equation is indeed exactly like (15) except for the sampled history $\bar{u}$ included in the context, and the new component u'_j in x'_j.

In summary, the filter for approximating the target joint distribution $\mathbb{P}(y, z|h)$ is given by the sequence of triplets $\mathcal{F}_h = \{(\bar{u}_i, w_i, \{B^h_j\}_j)\}_i$ where h is the action-observation history, $\bar{u}_i$ is the sampled history of the U variables, $i = 1, \ldots, N$, w_i is the weight associated with $\bar{u}_i$, and $B^h_j(y_j, z_j|\bar{u}_i)$ represents the belief factor given h and $\bar{u}_i$ for each beam j. These belief factors determine the probability $\mathbb{P}(y, z|\bar{u}, h)$ via (20), that provides the approximation for $\mathbb{P}(y, z|h)$ via (18). The filter $\mathcal{F}_h$ is extended to $\mathcal{F}_{h'}$ for $h' = \langle h, a, o\rangle$ by 1) extending the sampled history, 2) computing its associated weight, and 3) updating the belief factors with the new action-observation pair and the new sample. This last operation is defined by (21). We focus next on the other two operations. Initially, $\bar{u}_i$ is empty and $w_i = 1$ for $i = 1, \ldots, N$.

5.1 Marginals, Samples, and Weights

The factors $B^h_j(x_j|\bar{u})$, where x_j represents the valuations $y_j z_j$ over the non-sampled variables in beam b_j, do not represent themselves the probabilities $\mathbb{P}(x_j|h, \bar{u})$, that stand actually for the marginals:

$$\mathbb{P}(x_j|h, \bar{u}) \;=\; \beta \textstyle\sum_w \prod_i B^h_i(x_i|\bar{u}) \tag{22}$$

where β is a normalizing constant and the sum ranges over all the variables W that are not in the beam j. Such marginals can be computed from the factors $B^h_j(x_j|\bar{u})$ by standard methods such as the jointree algorithm or belief propagation. We show next how to compute the proposal distribution and weights from such marginals.

We consider two proposal distributions π: the so-called *motion* and *optimal* distributions [Doucet, 1998; Grisetti *et al.*, 2005]. The motion distribution uses only the transition probabilities to generate new histories $\bar{u}' = \langle \bar{u}, u'\rangle$ from previous histories $\bar{u}$ (not using the information provided by o):

$$\pi_{motion}(u'|h, a, o, \bar{u}) \;=\; \mathbb{P}(u'|h, a, \bar{u})\,, \tag{23}$$

while the optimal proposal makes use of both a and o:

$$\pi_{opt}(u'|h, a, o, \bar{u}) \;=\; \mathbb{P}(u'|h, a, o, \bar{u})\,. \tag{24}$$

For computing these probabilities effectively, we will *assume* that there is at least one beam j that contains the whole set of variables U, i.e. for which $U = U_j$. This assumption holds automatically when U is a singleton as in many SLAM problems. In addition and without loss of generality, we assume that the observation $o = o^t$ at time t, for each time step, falls into a single beam only; i.e., $q(o|x, a) = q_j(o_j|x_j, a)$ for some beam j, and $q_k(o_k|x_k, a) = 1$ for $k \neq j$. This is true when one observation variable is observed at each time point, and when this is not true, it can be made true by serializing the observations, using dummy actions in between. Under these assumptions, the proposal distributions and their weights can be computed from the marginals $P(x_j|h, \bar{u})$ associated with one beam. Indeed, if $U = U_j$ and $x_j = y_j, z_j, u_j$:

$$\mathbb{P}(u'|h, a, \bar{u}) \;=\; \textstyle\sum_{y_j, z_j} \mathbb{P}(u'|x_j, a)\, \mathbb{P}(y_j, z_j|h, \bar{u}) \tag{25}$$

and $\mathbb{P}(u'|h, a, o, \bar{u}) \propto \mathbb{P}(o|h, a, \bar{u}')\mathbb{P}(u'|h, a, \bar{u})$ where

$$\mathbb{P}(o|h, a, \bar{u}') \;=\; \textstyle\sum_{y'_k, z'_k} q_k(o_k|x'_k, a)\, \mathbb{P}(x'_k|h, a, \bar{u}') \tag{26}$$

if o falls into the beam k. The marginal $\mathbb{P}(x'_k|h, a, \bar{u}') = \mathbb{P}(y'_k, z'_k|h, a, \bar{u}')$ excludes the observation o, and hence it is obtained with (21) by setting all $q_j(o_j|x'_j, a)$ to 1.

The weights $w' = \mathbb{P}(\bar{u}'|h')/\pi(\bar{u}'|h')$ can be computed incrementally from the weights $w = \mathbb{P}(\bar{u}|h)/\pi(\bar{u}|h)$ when $h' = \langle h, a, o\rangle$ and $\bar{u}' = \langle \bar{u}, u'\rangle$. For the motion proposal, the weight is the ratio between:

$$\mathbb{P}(\bar{u}'|h') \;=\; \alpha'\, \mathbb{P}(o|\bar{u}', h, a)\, \mathbb{P}(u'|h, a, \bar{u})\, \mathbb{P}(\bar{u}|h)$$

and $\pi_{motion}(\bar{u}'|h') = \mathbb{P}(u'|h, a, \bar{u})\pi(\bar{u}|h)$, which results in:

$$w'_{motion} \;=\; \alpha'\, w_{motion} \times \mathbb{P}(o|\bar{u}', h, a)\,. \tag{27}$$

For the optimal proposal, the weight is the ratio between

$$\mathbb{P}(\bar{u}'|h') \;=\; \alpha''\, \mathbb{P}(u'|h, a, o, \bar{u})\, \mathbb{P}(o|h, a, \bar{u})\, \mathbb{P}(\bar{u}|h)$$

and $\pi_{opt}(\bar{u}'|h') = \mathbb{P}(u'|h, a, o, \bar{u})\pi(\bar{u}|h)$ which results in:

$$w'_{opt} \;=\; \alpha''\, w_{opt} \times \mathbb{P}(o|h, a, \bar{u})\,. \tag{28}$$

If the observation o falls into beam k, the marginals required can be computed as:

$$\begin{aligned} &\mathbb{P}(o|\bar{u}', h, a) \;=\; \textstyle\sum_{x'_k} q_k(o|x'_k, a)\, \mathbb{P}(x'_k|h, a, \bar{u}')\,, \text{ and} \\ &\mathbb{P}(o|\bar{u}, h, a) \;= \\ &\qquad \textstyle\sum_{x'_k} q_k(o|x'_k, a) \sum_{u'} \mathbb{P}(x'_k|h, a, \bar{u}')\mathbb{P}(u'|h, a, \bar{u}) \end{aligned}$$

where $\mathbb{P}(x'_j|h, a, \bar{u}') = \mathbb{P}(y'_j, z'_j|h, a, \bar{u}')$ is obtained as indicated above, and $\mathbb{P}(u'|h, a, \bar{u})$ is given by (25).

6 Faster Approximation of Marginals

The main bottleneck of the algorithm is the computation of the marginals $\mathbb{P}(x_j|h,\bar{u})$ from the factors $B_j^h(x_j|\bar{u})$ following (22). Such marginals are needed for computing the samples and weights, and for answering queries. The marginals can be computed using the jointree algorithm or belief propagation (BP). However, since scalability is crucial, we introduce a third method that will be evaluated in comparison with the other two. It is motivated by the results reported by beam tracking that uses arc consistency (AC) [Mackworth, 1977; Dechter, 2003] for progressing logical (non-probabilistic) beliefs. The relation between BP and AC is well-known: both methods are exact for trees and BP propagates the zero-probability entries in agreement with AC [Dechter and Mateescu, 2002]. We call the new method Iterated AC which is aimed at combining the speed and monotonic convergence of AC with the ability to approximate probabilistic beliefs, even if roughly.

Iterated AC approximates the marginals from the belief factors by using arc consistency along with order-of-magnitude probabilities also called κ-rankings [Spohn, 1988; Goldszmidt and Pearl, 1996; Darwiche and Goldszmidt, 1994]. For this, it follows three steps that we sketch briefly with no much justification. The algorithm uses two parameters ϵ and m. *First*, real values $p \in (0,1]$ are mapped into the smallest non-negative integer $\kappa = \kappa_\epsilon(p)$ for which $\epsilon^{\kappa+1} < p$, while $p = 0$ is mapped into $\kappa_\epsilon(p) = \infty$. This mapping is used to transform the belief factor $B_j^h(\cdot|\bar{u})$ into tables D_j^i that contain all the valuations x_j (for the variables X_j in beam j) that satisfy $\kappa_\epsilon(B_j^h(x_j|\bar{u})) \leq i + \eta_j$ for a given non-negative integer i where $\eta_j = \min_{x_j} \kappa_\epsilon(B_j^h(x_j|\bar{u}))$ is a normalization constant. *Second*, the tables D_j^i associated with the different beams j and the same i are made *arc consistent* (the tables share variables). The κ-marginal $\kappa(x_j|h,\bar{u})$ is defined then as i iff i is the minimum non-negative integer for which the tuple x_j belongs to D_j^i after running AC, while $\kappa(x_j|h,\bar{u}) = \infty$ when there is no such i. *Finally*, the $\kappa(x_j|h,\bar{u})$ marginals are used to approximate the marginals as $\mathbb{P}(x_j|h,\bar{u}) = \alpha\epsilon^{\kappa(x_j|h,\bar{u})}$ where α is a normalizing constant.

A further simplification is that κ measures that are greater than the m parameter but less than ∞, are treated as if they were equal to m. This means that the approximation of the marginals $\mathbb{P}(x_j|h,\bar{u})$ from the belief factors $B_j^h(\cdot|\bar{u})$ are computed by running arc consistency $m+1$ times. Iterated AC will be denoted as AC_m where m is the parameter used. The parameter ϵ is fixed to 0.1.

7 Experimental Results

The general PBT algorithm can use different algorithms for computing marginals from the factors. We experiment with the jointree (JT), belief propagation (BP), and AC_m algorithms for $m \in \{0,1\}$. For JT and BP we use `libdai` [Mooij, 2010] while AC_m is ours. The experiments were performed on Intel Xeon E5-2666 CPUs running at 2.9GHz with a memory cap of 10Gb (exhausted/approached only by JT).

		JT		BP		AC_0	
board	mines	%succ	time	%succ	time	%succ	time
6×6	6	84.1	.002	68.5	.046	84.2	.000
8×8	10	83.3	.070	66.3	.069	84.6	.002
16×16	40	—	—	41.4	.232	79.9	.005
30×16	99	—	—	1.8	.991	33.4	.003

Table 1: PBT in Minesweeper using three methods for computing marginals. Time is average time per decision in seconds. JT runs out of memory in larger maps. Figures are averages over 500 runs.

Minesweeper. Minesweeper cannot be fully solved by pure inference and requires guessing in certain situations, even if variables are static and sensing is noiseless. Since all variables are external but static, and hence backward deterministic, no sampling is required. The beam decomposition has one beam B_i for each cell i that contains up to 9 variables. Table 1 shows the results for PBT using JT, BP, and AC_0, with a policy that chooses to tag (resp. open) the cell that is most certain to contain a mine (resp. to be clear). The success ratio indicates the percentage of maps solved, i.e., without doing a wrong action. In this noise-free example, it can be shown that the marginals computed by PBT with AC_0 are equivalent to the ones computed by beam tracking where the belief factors represent sets of states [Bonet and Geffner, 2014]. Actually, PBT with AC_0 scales up best with a quality that matches the quality of JT in the small instances. JT does not scale up to larger instances and BP does but achieves a much lower score.

1-line-3-SLAM. This is a variant of the 1-line SLAM task considered before in which the observation received by the agent when in a certain cell depends also on the colors of the two *adjacent* cells. In such version of the problem, sampling the agent location is no longer sufficient for decoupling the color cell variables, but in our formulation makes the belief factors acyclic, so that exact inference over such factors is not exponential in the total number of variables but in the size of the largest factor. The observation is a 0/1 token equal to the color of the current cell with probability $p = 0.9^{noise}$ where $noise$ is one plus the number of adjacent cells with a different color. For a line of n cells, there are n beams of size up to 4, one for each cell, that include the agent location and the color of the three cells that influence the observation at the location. The agent location is the only external variable and is not backward deterministic, so it must be sampled. Table 2 shows results for instances of size $n = 64$ and $n = 512$, using 16 and 256 particles sampled with the optimal proposal distribution. The executions choose actions randomly until each cell of the grid is visited 10 times. The table shows averages with 97.5% confidence intervals over 100 executions for percentages of cells labeled correctly (one of two colors), not labeled at all (no sufficiently certainty), and times per step and per execution. For this problem, a cell is assumed "labeled" when for one of the possible colors its marginal probability is 0.55 of higher. Since the resulting factorized model has bounded and small treewidth, JT scales up well to provide a good baseline with few errors, although 40% of the cells

$1\times n$	#p	inf.	%good	%unknowns	time / step	time / exec
$1\times$ 64	16	JT	97.7 ± 0.2	41.7 ± 0.7	0.0 ± 0.0	2.9 ± 0.0
$1\times$ 64	16	BP	98.5 ± 0.2	43.0 ± 0.7	0.0 ± 0.0	8.9 ± 0.0
$1\times$ 64	16	AC_0	82.9 ± 0.6	36.5 ± 0.7	0.0 ± 0.0	1.1 ± 0.0
$1\times$ 64	16	AC_1	84.0 ± 0.5	36.5 ± 0.7	0.0 ± 0.0	1.1 ± 0.0
$1\times$ 64	256	JT	97.9 ± 0.2	41.0 ± 0.8	0.0 ± 0.0	42.8 ± 0.4
$1\times$ 64	256	BP	98.5 ± 0.2	40.8 ± 0.7	0.1 ± 0.0	110.9 ± 0.9
$1\times$ 64	256	AC_0	82.5 ± 0.6	34.5 ± 0.7	0.0 ± 0.0	19.1 ± 0.1
$1\times$ 64	256	AC_1	83.9 ± 0.6	33.3 ± 0.8	0.0 ± 0.0	19.6 ± 0.1
$1\times$512	16	JT	97.8 ± 0.1	47.8 ± 0.2	0.0 ± 0.0	319.7 ± 1.0
$1\times$512	16	BP	98.3 ± 0.1	47.8 ± 0.2	0.1 ± 0.0	681.7 ± 2.8
$1\times$512	16	AC_0	81.8 ± 0.2	42.3 ± 0.2	0.0 ± 0.0	88.7 ± 0.3
$1\times$512	16	AC_1	81.7 ± 0.2	42.1 ± 0.2	0.0 ± 0.0	91.0 ± 0.3
$1\times$512	256	JT	98.2 ± 0.1	47.5 ± 0.2	0.7 ± 0.0	$4,012.9 \pm 11.5$
$1\times$512	256	BP	98.4 ± 0.1	47.7 ± 0.2	1.4 ± 0.0	$8,193.1 \pm 22.6$
$1\times$512	256	AC_0	82.7 ± 0.2	41.7 ± 0.2	0.2 ± 0.0	$1,117.9 \pm 3.1$
$1\times$512	256	AC_1	82.6 ± 0.2	42.1 ± 0.2	0.2 ± 0.0	$1,121.0 \pm 3.0$

Table 2: PBT in 1-line SLAM using three methods for computing marginals. In this task, observations depend on the color of the current and adjacent cells. Figures are averages over 100 random executions, each of length roughly 10 times the number of cells.

are not labeled. The quality for BP in this case is similar but runs slower. AC_0 is an order-of-magnitude faster but of lower quality. The results for AC_1 are similar.

Minemapping. This problem is a version of Minesweeper that involves an agent that moves stochastically and receives noisy information about the presence/absence of mines in the cell and surrounding cells. The task for the agent is to map the minefield instead of clearing it as in Minesweeper, and the observations are similar but noisy. More precisely, movement actions have .9 probability of success and .1 probability of doing nothing, and when the current location is i, the observation token o is generated by summing stochastic indicator variables $I(j)$ for each adjacent cell j and for $j = i$, where the variable $I(j)$ is equal to 1 (resp. 0) with probability .9 if the cell j contains a mine (resp. no mine). The resulting beam structure is similar to the one for Minesweeper except that the agent location belongs to all the beams and must be sampled. Results for this problem are shown Table 3 in a format similar to Table 2. This time, JT is feasible for the small instances only, and BP appears to be the best choice for approximating the marginals: it doesn't make many mistakes when labeling cells (with mines or not) and runs faster than AC methods that make many more mistakes.

$n\times n$	#p	inf .	%good	%unknowns	time / step	time /exec
6×6	32	JT	97.3 ± 0.6	12.1 ± 1.6	0.2 ± 0.0	73.6 ± 1.0
6×6	32	BP	98.4 ± 0.6	16.6 ± 1.8	0.2 ± 0.0	81.0 ± 2.2
6×6	32	AC_0	68.0 ± 1.5	18.1 ± 0.8	0.1 ± 0.0	57.4 ± 3.2
6×6	32	AC_1	69.0 ± 1.4	17.6 ± 0.8	0.1 ± 0.0	56.8 ± 3.3
6×6	256	JT	97.9 ± 0.6	8.5 ± 1.4	1.7 ± 0.0	515.9 ± 5.9
6×6	256	BP	97.9 ± 0.6	10.3 ± 1.5	1.6 ± 0.0	506.7 ± 9.8
6×6	256	AC_0	67.0 ± 1.6	17.2 ± 0.8	1.3 ± 0.0	395.9 ± 20.6
6×6	256	AC_1	66.8 ± 1.6	17.4 ± 0.7	1.3 ± 0.0	397.8 ± 20.9
10×10	32	BP	98.9 ± 0.5	33.9 ± 1.3	0.8 ± 0.0	814.6 ± 8.3
10×10	32	AC_0	55.6 ± 0.9	23.7 ± 0.6	0.9 ± 0.0	852.3 ± 53.3
10×10	32	AC_1	55.0 ± 0.9	22.5 ± 0.6	0.9 ± 0.0	883.9 ± 48.7
10×10	256	BP	97.4 ± 0.6	27.9 ± 1.3	5.7 ± 0.0	$5,326.6 \pm 48.9$
10×10	256	AC_0	54.7 ± 1.1	23.8 ± 0.6	7.0 ± 0.3	$6,531.9 \pm 333.7$
10×10	256	AC_1	54.3 ± 1.2	23.5 ± 0.7	7.1 ± 0.3	$6,643.6 \pm 307.2$

Table 3: PBT in Minemapping using three methods for computing marginals. Figures are averages over 100 random executions. Each execution consists of roughly 5 times the number of cells in the grid.

8 Discussion

We have introduced a formulation and algorithm PBT for tracking probabilistic beliefs in the presence of stochastic actions and sensors in the form of local belief factors that can be progressed independently in time when variables appearing in more than one beam are backward deterministic. In such a case, the local belief factors provide an exact decomposition of the joint distribution at any time point, and progressing the factors in time is exponential in the size of the beams. The beams are fully determined by the 2-DBN model structure, and are usually bounded and small. For computing marginal probabilities, however, the local belief factors need to be merged. This computation can be performed exactly using the jointree algorithm or approximately by belief propagation or other local consistency methods. In addition, when the beams share variables that are not backward deterministic, such variables must be sampled to make them backward deterministic given their sampled histories.

As far as we know, there are no other general, principled approaches for dealing effectively with problems such as Minemapping or even 1-line-3-SLAM. RB particle-filtering methods would need to consider too many particles for making inference tractable in the first problem, and would need to be programmed to exploit the resulting tractable factorization in the second. In our setting, this all follows from the problem structure and the general formulation. At the same time, decomposition methods that operate over disjoint factors result in poor approximations, and standard grid SLAM algorithms [Grisetti *et al.*, 2005] involve a number of domain-dependent tricks that would not apply to more general problems, like the idea of associating one particular map to each particle, drastically simplifying the uncertainty about the map.

In the future, we want to develop new ideas for scaling PBT further so that it can be applied to more realistic SLAM problems. The bottleneck is not the progression of factors but the computation of marginals from factors that is done from scratch at every time point. We want to explore ways for making such computation incremental. Likewise, the performance of PBT needs to be compared with other approaches, in particular, approaches that also manage to exploit forms of context-specific independence and determinism [Vlasselaer *et al.*, 2016].

Acknowledgments

We thank the reviewers for useful comments. This work is partially funded by grant TIN2015-67959, MEC, Spain.

References

[Amir and Russell, 2003] E. Amir and S. Russell. Logical filtering. In *Proc. IJCAI*, pages 75–82, 2003.

[Bishop, 2007] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007.

[Bonet and Geffner, 2014] B. Bonet and H. Geffner. Belief tracking for planning with sensing: Width, complexity and approximations. *J. of AI Research*, pages 923–970, 2014.

[Boutilier *et al.*, 1996] C. Boutilier, N. Friedman, M. Goldszmidt, and Da. Koller. Context-specific independence in Bayesian networks. In *Proc. Int. Conf. Uncertainty in AI (UAI)*, pages 115–123, 1996.

[Boyen and Koller, 1998] X. Boyen and D. Koller. Tractable inference for complex stochastic processes. In *Proc. UAI*, pages 33–42, 1998.

[Darwiche and Goldszmidt, 1994] A. Darwiche and M. Goldszmidt. On the relation between kappa calculus and probabilistic reasoning. In *Proc. UAI*, pages 145–153, 1994.

[Darwiche, 2014] A. Darwiche. *Modeling and Reasoning with Bayesian Networks*. Cambridge, 2014.

[Dean and Kanazawa, 1989] T. Dean and K. Kanazawa. A model for reasoning about persistence and causation. *Computational intelligence*, 5(2):142–150, 1989.

[Dechter and Mateescu, 2002] R. Dechter and R. Mateescu. A simple insight into iterative belief propagation's success. In *Proc. UAI*, pages 175–183, 2002.

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

[Dechter, 2003] R. Dechter. *Constraint Processing*. Morgan Kaufmann, 2003.

[Doucet *et al.*, 2000] A. Doucet, N. De Freitas, K. Murphy, and S. Russell. Rao-Blackwellised particle filtering for dynamic bayesian networks. In *Proc. UAI*, pages 176–183, 2000.

[Doucet, 1998] A. Doucet. *On sequential simulation-based methods for Bayesian filtering*. PhD thesis, U. of Cambridge, 1998.

[Goldszmidt and Pearl, 1996] M. Goldszmidt and J. Pearl. Qualitative probabilities for default reasoning, belief revision, and causal modeling. *Artificial Intelligence*, 84(1):57–112, 1996.

[Grisetti *et al.*, 2005] G. Grisetti, C. Stachniss, and W. Burgard. Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling. In *Proc. Robotics and Automation (ICRA)*, pages 2432–2437, 2005.

[Kanazawa *et al.*, 1995] K. Kanazawa, D. Koller, and S. Russell. Stochastic simulation algorithms for dynamic probabilistic networks. In *Proc. UAI*, pages 346–351, 1995.

[Kaye, 2000] R. Kaye. Minesweeper is NP-Complete. *Mathematical Intelligencer*, 22(2):9–15, 2000.

[Koller and Friedman, 2009] D. Koller and N. Friedman. *Probabilistic graphical models: Principles and techniques*. MIT press, 2009.

[Mackworth, 1977] A. Mackworth. Consistency in networks of relations. *Artificial intelligence*, 8(1):99–118, 1977.

[Mooij, 2010] Joris M. Mooij. libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *Journal of Machine Learning Research*, 11:2169–2173, August 2010.

[Murphy and Weiss, 2001] K. Murphy and Y. Weiss. The factored frontier algorithm for approximate inference in DBNs. In *Proc. Int. Conf. Uncertainty in AI (UAI)*, pages 378–385, 2001.

[Murphy, 1999] K. Murphy. Bayesian map learning in dynamic environments. In *NIPS*, pages 1015–1021, 1999.

[Murphy, 2002] K. Murphy. *Dynamic bayesian networks: representation, inference and learning*. PhD thesis, University of California, Berkeley, 2002.

[Pearl, 1988] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

[Spohn, 1988] W. Spohn. A general non-probabilistic theory of inductive reasoning. In *Proceedings UAI*, pages 315–322, 1988.

[Thrun *et al.*, 2001] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *Artificial intelligence*, 128(1):99–141, 2001.

[Vlasselaer *et al.*, 2016] J. Vlasselaer, W. Meert, G. Van den Broeck, and L. De Raedt. Exploiting local and repeated structure in dynamic bayesian networks. *Artificial Intelligence*, 232:43–53, 2016.