Advances in Applying Stochastic-Dominance Relationships to Bounding Probability Distributions in Bayesian Networks*

Chao-Lin Liu
Department of Computer Science
National Chengchi University
Taipei 11605, Taiwan
chaolin@nccu.edu.tw

Abstract

Bounds of probability distributions are useful for many reasoning tasks, including resolving the qualitative ambiguities in qualitative probabilistic networks and searching the best path in stochastic transportation networks. This paper investigates a subclass of the state-space abstraction methods that are designed to approximately evaluate Bayesian networks. Taking advantage of particular stochastic-dominance relationships among random variables, these special methods aggregate states of random variables to obtain bounds of probability distributions at much reduced computational costs, thereby achieving high responsiveness of the overall system.

The existing methods demonstrate two drawbacks, however. The strict reliance on the particular stochastic-dominance relationships confines their applicability. Also, designed for general Bayesian networks, these methods might not achieve their best performance in special domains, such as fastest-path planning problems. The author elaborates on these problems, and offers extensions to improve the existing approximation techniques.

Keywords: Bayesian Networks, Stochastic Dominance, Approximate Reasoning

1. Introduction

In the past decade, Bayesian networks have become a major formalism for capturing and reasoning about uncertainty in complex applications [6]. A Bayesian network encodes, respectively, qualitative and quantitative probabilistic relationships among random variables in terms of a directed acyclic graph and conditional probability tables [8, 17]. Given observations about some random variables, we *evaluate* the Bayesian network to obtain the conditional probability distributions of random variables of interest. The evaluation process is also known as *inference* in Bayesian networks, and this active research field has seen a wide variety of approaches for computing exact and approximate probability distributions. Approximation algorithms allow us to obtain useful information about the desired probability distributions at reduced computational costs when specific application-dependent constraints do not permit exact inference. D'Ambrosio offers a very informative survey in [3], and some recent developments include [10, 15, 16].

We can classify approximate inference procedures from different perspectives. In terms of how we carry out the approximations, D'Ambrosio comes up with two schools of algorithms: *approximate inference methods* compute distributions with special algorithms using the original network, e.g., [2, 18], and *model reduction methods* employ exact algorithms after simplifying the original network, e.g., [14, 20]. Classifying in terms of types of the outcomes of approximation procedures, we see that some algorithms compute the upper and/or lower bounds, e.g., [4, 7], while others compute point-valued approximations of the desired probability distributions [2, 20].

In the following figure, let E be the curve of the exact cumulative distribution function of a random variable X. Approximate algorithms may compute the upper and lower bounds, U and L, respectively, or the point-valued approximation, A, of E. The curve L is called a lower bound because it suggests the distribution of the random variable X tilts to its lower range, although geometrically L locates on the upside of the curve E.

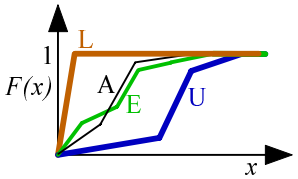

Figure 1. Approximations of the exact distribution E

The state-space abstraction (SSA) methods compute approximate probability distributions by first simplifying the given Bayesian network. Depending on how we simplify the networks, we can compute either the point-value approximations[14] or bounds of the desired probability distributions [12]. For computing the bounds, the SSA methods require that the underlying conditional distributions encoded in the Bayesian networks exhibit the *stochastic dominance* property [23]. Although this property may hold for some applications, there are applications in which this assumption is slightly violated. In this paper, I discuss how we can revise the original SSA methods and expand their applicability into this arena. Also, hoping to find tighter bounds, the SSA methods employ heuristics for selecting the abstract model of the

*This paper appeared in the *Proceedings of the IASTED International Conference on Artificial and Computational Intelligence 2002*, 251-256. Tokyo, Japan, 25-27 September 2002.

original network such that we obtain the best approximations possible. It is, however, very difficult to design heuristics good for all possible probability distributions. This paper reports a new heuristic tailored for fastest path planning problems [13].

The following section presents details of stochastic dominance and its applications with the state-space abstraction methods. Section 3. investigates the situations in which the requirement for stochastic dominance can be relaxed, and provides the revised SSA methods. Section 4. discusses and proposes a new strategy for obtaining the best approximations for applying the SSA methods to fastest path planning problems. Section 5. provides an outline of applications of the new methods, and Section 6. concludes this paper with a brief summary.

2. Background

In this paper, we use capital and small letters to denote random variables and their values, respectively. The possible values of a random variable are also called *states* of the random variable, and we use subscripts to them when necessary. Also, we adopt the shorthand $Pr(x|y,z)$ for $Pr(X=x|Y=y,Z=z)$. Most of the time, we need to refer to a set of random variables, and we use bold letters for sets. We also follow the tradition by calling a node *parent* when it has outgoing arc to another node. The node with an incoming arc is called a *child* of its parent.

2.1 Stochastic dominance

Let $F_1(X)$ and $F_2(X)$ be two possible cumulative distribution functions (CDFs) of the random variable X.

Definition 1 ([21]) *We say that* $F_1(X)$ first-order stochastically dominates $F_2(X)$ *if and only if* $F_1(x) \leq F_2(x)$ *holds for all* x *, and we denote this relationship by* $F_1(x)$ *FSD* $F_2(x)$.

Such a stochastic dominance relationship is used extensively in defining qualitative probabilistic networks (QPNs) in [21]. In a sense, QPNs are special Bayesian networks that take advantage of the dominance relationships to design efficient inference procedures for qualitative relationships. Wellman defines that a node X *positively influences* its child Y if and only if the following inequality holds for all x_i, x_j, and values of the rest of Y's parents $\boldsymbol{px}(Y)$.

$$x_i \leq x_j \Rightarrow F(y|x_j, \boldsymbol{px}(Y)) \text{ FSD } F(y|x_i, \boldsymbol{px}(Y)) \quad (1)$$

An interpretation of this inequality is that, all else being equal, increasing the value of X increases the probability of Y being a larger value as is shown in the following figure. In transportation problems, this is tentative to say that leaving the origin at a later time increases the probability of arriving at the destination later. Analogously, we say that X *negatively influences* Y if the dominance relationship reverses in (1).

Another implication of the dominance relationship $F_1(x)$ *FSD* $F_2(x)$ is that the following relationship holds

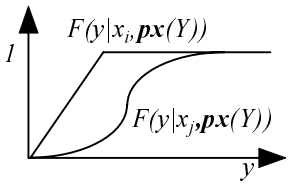

Figure 2. X **positively influences** Y**.**

for all monotonically increasing function $g(X)$ [23].

$$\int g(x)dF_1(x) \geq \int g(x)dF_2(x) \quad (2)$$

This inequality has been applied to fastest-path planning algorithms [22], resolution of ambiguous qualitative relationships in QPNs [11], and computing bounds of probability distributions [12, 13].

2.2 State-space abstraction methods

A Bayesian network encodes the joint probability distribution of a set of nodes.* Each random variable can take on a set of possible values, and each of which is called a *state* of the random variable. It is well known that the computational costs for the evaluation of Bayesian networks increase exponentially with the cardinality of states of the random variables [1]. Hence, reduction of the cardinality emerges as an intuitive approach for computing the desired probability distributions at lower costs. The skeleton of the iterative state-space abstraction (ISSA) algorithm follows.

Algorithm 1 (ISSA [14]) *Iterative State-Space Abstraction*

1. *Abstraction: Construct an approximated network of the original network by aggregating states and reconstructing CPTs.*
2. *Inference: Evaluate the approximated network to obtain approximations of interest.*
3. *Termination?: Check whether the algorithm should stop, using application-dependent criteria. If yes, return the current solution. Otherwise, go to the next step. The algorithm will stop when there is no superstate in the current network.*
4. *Refinement: Select which superstate should be split, return to step 1.*

To evaluate a given Bayesian network approximately, the state-space abstraction methods construct a very coarse version of the original network by aggregating consecutive states of some random variables [14]. Let us call these aggregated states *superstates* henceforth. Since every random variable in a Bayesian network has an associated conditional probability table (CPT) that contains its probability distributions given its parents' states, the state-space abstraction procedure

*Since we represent a random variable with a oval node in Bayesian networks, we will use nodes and random variables exchangeably henceforth.

must also construct CPTs for all affected random variables. Specifically, if a superstate is introduced into either the state space of a random variable or the state spaces of the parents of a random variable, then we need to reconstruct the CPT of this affected random variable.

As a result, we need to compute the probability values for the new CPTs from the probability values contained in the original CPTs. Let $\boldsymbol{P}(A)$ and $\boldsymbol{C}(A)$ respectively denote the parents and children of a node A, a_i the i^{th} possible state of A, and $[a_{i,j}]$ the superstate that is the aggregation of states a_i through a_j inclusively. If we replace a_i through a_j by $[a_{i,j}]$, then we need to determine the values in the reconstructed CPTs of A and $\boldsymbol{C}(A)$. Namely, we need to determine $\hat{\Pr}([a_{i,j}]|\boldsymbol{p}(A))$ for every possible value $\boldsymbol{p}(A)$ of $\boldsymbol{P}(A)$, where $\hat{Pr}(\cdot)$ represents an approximated probability. The CPT of every child T of A must be reconstructed as well. Let $\boldsymbol{PX}(T) = \boldsymbol{P}(T) \setminus \{A\}$ denote the parents of T excluding A. We need to determine the conditional probabilities $\hat{\Pr}(t_k|\boldsymbol{px}(T), [a_{i,j}])$ for all k and $\boldsymbol{px}(T)$. By appropriately choosing formula for this probability reassignment task, we control whether we compute the point-value approximations [14] or bounds of the desired probability distributions [12].

Using the following formula in ISSA will give us bounds of distributions of T when A positively influences T.

$$\hat{\Pr}([a_{i,j}]|\boldsymbol{p}(A)) = \sum_{l=i}^{j} \Pr(a_l|\boldsymbol{p}(A)) \tag{3}$$

$$\hat{F}(t_k|\boldsymbol{px}(T), [a_{i,j}]) = \max_{l\in[i,j]} F(t_k|\boldsymbol{px}(T), a_l) \tag{4}$$

(3) is an intuitive assignment, where the probability of a superstate is the sum of the probability of its components. The formula for $\hat{\Pr}(t_k|\boldsymbol{px}(T), [a_{i,j}])$, for all k, is more complex, and we assign them such that (4) holds, where $F(t_k|\boldsymbol{px}(T), a_l) \equiv \Pr(t \le t_k|\boldsymbol{px}(T), a_l)$ represents the conditional, cumulative distribution of T.

After constructing an approximated network, we may employ any exact evaluation algorithm to compute the probability distributions of interest. When random variables positively or negatively influence one another, we can prove that we will obtain bounds of probability distributions if we apply (3) and (4) in the abstraction step [12].

For some applications, we may want to refine the approximated network for better solutions after obtaining the current approximations. We may achieve this goal by splitting superstates in the approximated network, and construct another approximated network for evaluation.

Assume that the state space of an abstracted node[†] contains more than one superstate. We will need to choose which superstate to split. An intuitive strategy is to split the superstate that has the largest approximate, marginal probability for every abstracted node. This so-called most-probable-superstate (MPSS) heuristic led to satisfactory results in some experiments [14]. However, selecting the "best" superstate to split for the new approximated network is not an easy problem, and interested readers are referred to [14] for further details.

[†] Any node whose states are aggregated is an abstracted node.

3. Relaxing stochastic dominance

In previous work, Liu and Wellman report that we can apply the ISSA algorithm to compute bounds of probability distributions when random variables positively or negatively influence others [12]. Such an algorithm allows us to explore complex networks that would render exact computation of probability distributions impractical.

The main purpose of requiring the positive/negative influence relationship between random variables is that we can aggregate states freely and obtain bounds of the exact distributions. The following derivation shows the core basis for computing bounds of travel times via state-space abstraction.

Let $\Pi = L_1 \to L_2 \to \cdots \to L_n$ be a traveling path, and $\Pr(t_i)$ be the probability of arriving at location L_i at time $T_i = t_i$. Thus, given a departure time from L_1, say $\bar{t}_1$, we can infer the distribution of the arrival time at L_2 easily, and it is $F(t_2|\bar{t}_1)$. As we expand the partial path in a search algorithm, we compute the CDF $F(t_{j+1})$ of arrival time at L_{j+1} based on the arrival time at L_j:[‡]

$$F(t_{j+1}) = \sum_{t_j} F(t_{j+1}|t_j) \Pr(t_j), \tag{5}$$

where $\Pr(t_j)$ is actually a shorthand for $\Pr(t_j|\bar{t}_1)$. The conditioning on $\bar{t}_1$ will not be shown explicitly for notational simplicity henceforth. One way to control the time for computing the distributions of arrival times is to confine the growth of the number of states of L_j [13]. We can achieve this by aggregating the state of L_j before we compute the distribution of L_{j+1}. Therefore, in general, we would still like to abstract the state space of T_j in computing the distribution of T_{j+1} after obtaining $\hat{F}(t_j)$. We apply (3) and (4) as follows.

$$\hat{\hat{\Pr}}(\hat{t_j}) = \sum_{t_j \in \hat{t_j}} \hat{\Pr}(t_j) \tag{6}$$

$$\hat{F}(t_{j+1}|\hat{t_j}) = \max_{t_j \in \hat{t_j}} F(t_{j+1}|t_j) \tag{7}$$

We can let $\hat{\Pr}(t_2) \equiv \Pr(t_2)$ without any loss, although we have obtained the exact distribution for T_2 already. Hence we can apply (6) and (7) to T_j for all $j \ge 2$. In (6), the double "hats" imply that the approximate probabilities are determined based on other approximated probabilities. For simplicity, single "hat" rather than double "hats" will be used to denote any approximate probabilities when there is no risk of confusion. Also we use the "hat" symbol over t_j to denote that the state space of T_j is aggregated when we compute an approximate distribution of T_{j+1}.

[‡] For simplicity, we assume that one would not stop at intermediate locations, so there is really no need to distinguish arrival and departure time for intermediate locations. As a result, we will use arrival time for both.

Using (6) and (7), we show $F(t_{j+1})$ *FSD* $\hat{F}(t_{j+1})$ as follows.

$$\begin{aligned}
&\hat{F}(t_{j+1}) \qquad (8)\\
&= \sum_{\hat{t_j}} \hat{F}(t_{j+1}|\hat{t_j})\,\hat{\Pr}(\hat{t_j})\\
&= \sum_{\hat{t_j}} [[\max_{t_j \in \hat{t_j}} F(t_{j+1}|t_j)] \sum_{t_j \in \hat{t_j}} \Pr(t_j)]\\
&\geq \sum_{t_j} F(t_{j+1}|t_j)\Pr(t_j) = \sum_{t_j} F(t_{j+1}|t_j)d\hat{F}(t_j)\\
&\geq \sum_{t_j} F(t_{j+1}|t_j)dF(t_j) = F(t_{j+1})
\end{aligned}$$

Since every t_j is covered by exactly one $\hat{t_j}$ when we aggregate states, $\hat{\Pr}(t_j)$ will occur exactly once after we completely expand the summations in the second equality. Also each component $\Pr(t_j)$ of $\hat{\Pr}(\hat{t_j})$ is multiplied by $\max_{t_j \in \hat{t_j}} F(t_{j+1}|t_j)$ which must be larger than $F(t_{j+1}|t_j)$ for all t_j covered by $\hat{t_j}$, so we obtain the first inequality in (8) after recollecting all terms. Now, as we have assumed that T_j positively influences T_{j+1}, we have $F(t_{j+1}|t_j)$ *FSD* $F(t_{j+1}|t'_j)$ if $t_j \geq t'_j$. In other words, $F(t_{j+1}|t_j)$ is a non-increasing function of t_j. Also recall that at L_2, $\hat{F}(t_2)$ is actually an exact distribution, so it is trivially true that $F(t_2)$ *FSD* $\hat{F}(t_2)$. Using proof by induction, we can assume that $F(t_j)$ *FSD* $\hat{F}(t_j)$, and go on to show that $F(t_{j+1})$ *FSD* $\hat{F}(t_{j+1})$. Now given that $F(t_j)$ *FSD* $\hat{F}(t_j)$ and that $F(t_{j+1}|t_j)$ is a non-increasing function of t_j, we can apply (2) to obtain the second inequality after a simple algebraic manipulation, and establish $F(t_{j+1})$ *FSD* $\hat{F}(t_{j+1})$.

For networks that have more complex structure than the linear Π, the proof is similar but more involved. In short, when a node in the Bayesian network positively or negatively influences its child, we can apply the state-space abstraction methods to compute bounds of probability distributions. Moreover, the SSA methods will give us bounds as long as we aggregate consecutive states.

What do positive and negative influences really imply in reality? Let X and Y in Figure 2 be the departure time and arrival time of a trip, respectively. An interpretation of positive influence in transportation networks is that departing from the origin later will not increase the probability of arriving at the destination at an earlier time. This assumption seems reasonable, and arguably holds in real world applications. Nevertheless, it is an assumption at best.

When the relationships of positive or negative influence do not hold, the second inequality in (8) will not hold because we lose the condition that $F(t_{j+1}|t_j)$ is a non-increasing function of t_j. This, in turn, destroys the applicability of SSA methods.

However, if the assumption of positive or negative influence is slightly violated, we can still apply the SSA methods, with some modifications, to find bounds of probability distributions. Consider the distributions shown in Figure 3, where t_{jk} represents the k^{th} state of T_j. The crossing curves show that T_j does not positively influence T_{j+1}. However, the trend of the curves seems

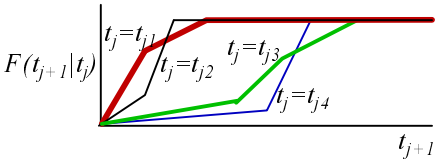

Figure 3. T_j *weakly* positively influences T_{j+1}.

to support that T_j *weakly* positively influences T_{j+1}. In particular, both $F(t_{j+1}|t_{j3})$ and $F(t_{j+1}|t_{j4})$ first order dominate $F(t_{j+1}|t_{j1})$ and $F(t_{j+1}|t_{j2})$. A formal definition follows.

Definition 2 *Assume that a random variable X has m states: $x_1, x_2, \cdots, x_m$ and that these states form $n \geq 1$ groups: $G_1 = \{x_1, \cdots, x_{b_1}\}$, $G_2 = \{x_{b_1+1}, \cdots, x_{b_2}\}$, $\cdots$, and $G_n = \{x_{b_{n-1}+1}, \cdots, x_m\}$. A node X* weakly positively influences *its child Y if and only if, $F(y|x_i, \boldsymbol{px}(Y))$ FSD $F(y|x_k, \boldsymbol{px}(Y))$, for all $x_i \in G_j$, $x_k \in G_l$, and $\boldsymbol{px}(Y)$, where $j > l$ and $\boldsymbol{px}(Y)$ denotes value of other parents $\boldsymbol{PX}(Y)$ of Y.*

When X weakly positively influences its child Y, we can apply the SSA methods to obtain bounds of the distributions of Y. Let $g(x_i)$ be the state group that contains x_i. To compute the bounds, we approximate the conditional cumulative distribution functions $F(y|x_i, \boldsymbol{px}(Y))$ for all x_i by the following formula before computing the approximations with Algorithm 1.

$$\hat{F}(y|x_i, \boldsymbol{px}(Y)) = \max_{x_j \in g(x_i)} F(y|x_j, \boldsymbol{px}(Y)) \qquad (9)$$

Consider the example shown in Figure 3. After we apply (9) to the distributions of $F(t_{j+1}|t_j)$, both $F(t_{j+1}|t_{j1})$ and $F(t_{j+1}|t_{j2})$ are set to the values of the upper, thick curve, while $F(t_{j+1}|t_{j3})$ and $F(t_{j+1}|t_{j4})$ to the lower, thick curve in Figure 4.

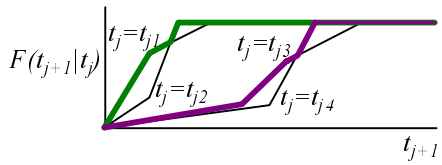

Figure 4. Using weakly positive influence for bounding distributions

We can prove that the approximate CDF computed with these approximations first order dominates the exact CDF.

$$\begin{aligned}
\hat{F}(t_{j+1}) &= \sum_{t_j} (\max_{t_j \in g(t_j)} F(t_{j+1}|t_j))\,\Pr(t_j)\\
&\geq \sum_{t_j} F(t_{j+1}|t_j)\,\Pr(t_j) = F(t_{j+1})
\end{aligned}$$

After we apply (9) to the conditional CDFs, the resulting approximate CDFs make the involved random variables assume the positive influence relationship. As a result, we can apply the SSA methods to compute bounds of the already approximated distributions using even less number of states. Due to transitivity, the new bounds are also bounds of the exact distributions.

When the *FSD* relationship in Definition 2 reverses, we say that X *weakly negatively influences* Y. Under such circumstances, we replace the $\max$ operator in (9) by the $\min$ operator to obtain bounds of probability distributions analogously.

4. Superstate selection strategies

At step 4 of Algorithm 1, we have to split selected superstates for improving the quality of approximations. As we split the superstates, we recover some distinction among the original states, and expect the results of evaluating Bayesian networks to improve. Liu and Wellman report theoretic and experimental analysis for this superstate selection problem for Bayesian networks [14].

As we discuss in the previous section, the superstate selection problem for computing the fastest path is not the same as that for evaluating Bayesian networks. As we gradually expand partial paths $\Pi = L_1 \rightarrow L_2 \rightarrow \cdots \rightarrow L_n$ to the next intermediate location L_{n+1}, we have an approximate probability distribution for T_n already. In order to confine the growth of the state spaces of arrival times for intermediate locations, we aggregate the states of T_n before computing the CDF of T_{n+1}. The problem is how we group states into a set of superstates, not selecting superstates for splitting. In the previous work on fastest path planning problems, this superstate selection problem was unaddressed [13].

Liu and Wellman assume that the departure time positively influences the arrival time for any trip in applying the SSA methods for computing bounds of travel times. Figure 2 shows such an example, letting X and Y be the departure and arrival times, respectively. For computing the travel times of a path Π, they assume that the departure time from L_j positively influences the arrival time at L_{j+1}. Namely, $F(t_{j+1}|t'_j) \geq F(t_{j+1}|t_j)$ for all t_{j+1} and $t_j > t'_j$.

Let t_{jk} be the k^{th} state of T_j. We make the assumption more specific by assuming that the values of $F(t_{j+1}|t_{jk})$ will not deviate from those of $F(t_{j+1}|t_{j(k+1)})$ significantly. Namely, for a small ε and for all k and t_{j+1}, we assume

$$F(t_{j+1}|t_{jk}) - F(t_{j+1}|t_{j(k+1)}) \leq \varepsilon. \tag{10}$$

This assumption should hold for transportation networks, as we typically do not expect normal traffic conditions to change drastically within a short time period. The assumed inequality deviates from reality when t_{j+1} is extremely small or large. $F(t_{j+1}|t_{jk})$ will be 0 and 1, respectively, for all t_{jk}, and the differences should be 0. Nevertheless, the inequality still holds.

Now, although we are computing bounds for the desired distributions, we would like to make the bounds as close to the actual distributions as possible. Assume that T_j has m states: $t_{j1}, t_{j2}, \cdots, t_{jm}$ and that we aggregate these states into n groups: $S_1 = \{t_{j1}, \cdots, t_{jb_1}\}$, $S_2 = \{t_{j(b_1+1)}, \cdots, t_{jb_2}\}$, $\cdots$, and $S_n = \{t_{j(b_{n-1}+1)}, \cdots, t_{jm}\}$. Let $g'(t_{jk})$ denote the group that contains t_{jk}. To minimize the errors, referring to (8) and its derivation, we would like to minimize the following difference.

$$\begin{aligned}
\delta &= \sum_{\hat{t'_j}} \hat{F}(t_{j+1}|\hat{t'_j})\,\hat{\hat{\Pr}}(\hat{t'_j}) - \sum_{t_j} F(t_{j+1}|t_j)\hat{\Pr}(t_j) \\
&= \sum_{\hat{t'_j}} [[\max_{t_j \in \hat{t'_j}} F(t_{j+1}|t_j)] \sum_{t_j \in \hat{t'_j}} \hat{\Pr}(t_j)] - \\
&\quad \sum_{t_j} F(t_{j+1}|t_j)\hat{\Pr}(t_j) \qquad (11) \\
&= \sum_{t_j} [[F(t_{j+1}|\min_{t_k \in g'(t_j)} t_k) - F(t_{j+1}|t_j)]\hat{\Pr}(t_j)] \\
&\leq [(0 + \varepsilon\hat{\Pr}(t_{j2}) + \cdots + (b_1 - 1)\varepsilon\hat{\Pr}(t_{jb_1}))] + \\
&\quad [(0 + \varepsilon\hat{\Pr}(t_{j(b_1+2)}) + \cdots \\
&\quad + (b_2 - b_1 - 1)\varepsilon\hat{\Pr}(t_{jb_2}))] + \cdots \\
&\quad + [(0 + \varepsilon\hat{\Pr}(t_{j(b_{n-1}+2)}) + \cdots \\
&\quad + (b_n - b_{n-1} - 1)\varepsilon\hat{\Pr}(t_{jm}))]
\end{aligned}$$

The first two equalities follow directly from (8). The third equality also follows from (8), adding that $\max_{t_j \in t'_j} F(t_{j+1}|t_j)$ is equal to the CDF of T_{j+1} given the smallest t_k in $g'(t_j)$ because T_j positively influences T_{j+1}. Applying (10) will give us the inequality in (11), where the zeros result from the fact that in each S_i, one and only one CDF will subtract itself.

The right hand side of the inequality in (11) gives us an upper bound of the difference δ. Therefore, one way to minimize δ is to minimize the upper bound. Let $b_0 = 0$ and $b_n = m$. When (10) holds, a heuristic for determining how we aggregate the states of T_j into n group is to minimize the following quantity.

$$\sum_{k=0}^{n-1} \sum_{i=b_k+1}^{b_{k+1}} (i - b_k - 1)\hat{\Pr}(t_{ji})$$

Notice that the contribution of each t_{jk} is $\hat{\Pr}(t_{jk})$ multiplied by a weighting factor that is determined by the location of t_{jk} in its group $g'(t_{jk})$. In contrast, the MPSS heuristic, that was proposed for general Bayesian networks and discussed in Section 2.2, leads us to use $\sum_{t_j \in t'_j} \hat{\Pr}(t_j)$ as the guidance for superstate selection.

5. Applications

The techniques presented in this paper should be helpful for any applications that need to compute probability distributions of random variables. Random variables can represent travel times in path planning problems, job processing times in task planning problems, etc. One may also apply the methods to resolving tradeoffs in QPNs [11, 19]. Due to space limitation, we provide only the outline of an application to path planning problems below.

Hall shows that we cannot directly apply the principle of dynamic programming to path planning in networks with time-dependent arc weights [5]. Kaufman and Smith extend the applicability of the Dijkstra's algorithm to transportation networks with time-dependent travel costs by introducing the concept of *consistent* link travel times [9]. Wellman et al. generalize the concept to

stochastic consistency in stochastic networks [22]. Liu and Wellman apply the SSA methods to tackle path planning problems when we have large stochastic networks [13]. In both [22] and [13], strict stochastic dominance relationships must hold among related distributions of travel times, which, as discussed in [9], is a good approximation to the reality at best. The concepts of *weakly positive/negative influences* in Definition 2 extends the applicability of algorithms proposed in the previous extensions.

Assume that we are given a transportation network in which the departure times weakly positively influences the arrival times as shown in Figure 3. As discussed in Section 3., we can create an approximation of the transportation network by formula (9) so that we can apply the path planning algorithm proposed in [13]. Also, the technique discussed in Section 4. provides context-specific guidelines for improving qualities of bounds for the path planning problems, and we can implement the guidelines in [13] as well.

6. Conclusions

Assuming the positive and negative influence relationships among random variables, we can apply the state-space abstraction methods to computing bounds of probability distributions [12, 14]. Applications of such bounds include inferring qualitative relationships in qualitative probabilistic networks [11] and searching fastest paths in stochastic transportation systems [13].

However, the assumption of positive and negative influences limits the applicability of the existing methods. This paper defines the concepts of *weakly positive influence* and *weakly negative influence* among random variables. When random variables have such relationships, the state-space abstraction methods remain applicable after a few revisions. Also we show that the superstate selection strategies proposed for general Bayesian networks may not work well for special domains, and find a better heuristic designed for the fastest path planning problems.

References

[1] Cooper, G. F., The computational complexity of probabilistic inference using Bayesian belief networks, *Artificial Intelligence* 42, 393–405, 1990.

[2] Dagum, P. and Luby, M., An optimal approximation algorithm for Bayesian inference, *Artificial Intelligence* 93, 1–27, 1997.

[3] D'Ambrosio, B., Inference in Bayesian networks, *AI Magazine* 20(2), 21–36, Summer 1999.

[4] Draper, D. L. and Hanks, S., Localized partial evaluation of belief networks, *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, 170–177, 1994.

[5] Hall, R. W., The fastest path through a network with random time-dependent travel times, *Transportation Science* 20(3), 182–188, 1986.

[6] Heckerman, D. E., Mamdani, A., and Wellman, M. P., Real-world applications of Bayesian networks, *Communications of the ACM* 38(3), 24–26, 1995.

[7] Horvitz, E., Suermondt, H. J., and Cooper, G. F., Bounded conditioning: Flexible inference for decisions under scarce resources, *Proceedings of the 5th Workshop on Uncertainty in Artificial Intelligence*, 182–193, 1989.

[8] Jensen, F. V., *An Introduction to Bayesian Networks*, Springer-Verlag, New York, 1996.

[9] Kaufman, D. E. and Smith, R. L., Fastest paths in time-dependent networks for intelligent vehicle-highway systems applications, *IVHS Journal* 1(1), 1–11, 1993.

[10] Koller, D., Lerner, U., and Angelov, D., A general algorithm for approximate inference and its application to hybrid Bayes nets, *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, 324–333, 1999.

[11] Liu, C.-L. and Wellman, M. P., Incremental tradeoff resolution in qualitative probabilistic networks, *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, 338–345, 1998.

[12] Liu, C.-L. and Wellman, M. P., Using qualitative relationships for bounding probability distributions, *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, 346–353, 1998.

[13] Liu, C.-L. and Wellman, M. P., Using stochastic-dominance relationships for bounding travel times in stochastic networks, *Proceedings of the 2nd International IEEE Conference on Intelligent Transportation Systems*, 55–60, 1999.

[14] Liu, C.-L. and Wellman, M. P., Evaluation of Bayesian networks with flexible state-space abstraction methods, *International Journal of Approximate Reasoning* 30(1), 1–39, 2002.

[15] Minka, T. P., Expectation propagation for approximate Bayesian inference, *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence*, 362–369, 2001.

[16] Nicholson, A. E. and Jitnah, N., Using mutual information to determine relevance in Bayesian networks, *Proceedings of the 5th Pacific Rim International Conference on Artificial Intelligence*, 399–410, 1998.

[17] Pearl, J., *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers, Inc., 1988.

[18] Poole, D., Average-case analysis of a search algorithm for estimating prior and posterior probabilities in Bayesian networks with extreme probabilities, *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, 606–612, 1993.

[19] Renooij, S. and van der Gaag, L. C., Enhancing QPNs for trade-off resolution, *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, 559–566, 1999.

[20] van Engelen, R. A., Approximating Bayesian belief networks by arc removal, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(8), 916–920, 1997.

[21] Wellman, M. P., Fundamental concepts of qualitative probabilistic networks, *Artificial Intelligence* 44, 257–303, 1990.

[22] Wellman, M. P., Ford, M., and Larson, K., Path planning under time-dependent uncertainty, *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, 532–539, 1995.

[23] Whitmore, G. A. and Findlay, M. C., (Eds.), *Stochastic Dominance: An Approach to Decision Making Under Risk*, D. C. Heath and Company, 1978.