Bounding probabilistic relationships in Bayesian networks using qualitative influences: methods and applications ☆

Chao-Lin Liu [a,*], Michael P. Wellman [b]

[a] *National Chengchi University, 64 Sec. 2, Chih-Nan Road, Taipei 11605, Taiwan*
[b] *University of Michigan, Ann Arbor, MI 48109, USA*

Received 1 December 2002; accepted 1 June 2003

Abstract

We present conditions under which one can bound the probabilistic relationships between random variables in a Bayesian network by exploiting known or induced qualitative relationships. Generic strengthening and weakening operations produce bounds on cumulative distributions, and the directions of these bounds are maintained through qualitative influences. We show how to incorporate these operations in a state-space abstraction method, so that bounds provably tighten as an approximate network is refined. We apply these techniques to qualitative tradeoff resolution demonstrating an ability to identify qualitative relationships among random variables without exhaustively using the probabilistic information encoded in the given network. In an application to path planning, we present an anytime algorithm with run-time computable error bounds.
© 2003 Elsevier Inc. All rights reserved.

1. Introduction

In the past decade, the Bayesian network has emerged as a premier modeling formalism for representing and reasoning about uncertainty in complex applications [19,21]. A Bayesian network captures, respectively, structural and numeric probabilistic relationships among random variables in terms of a directed acyclic graph and conditional probability tables [24,40]. 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 [9]. Approximation algorithms allow us to obtain useful information about the desired probability distributions at reduced computational costs when applications do not need or permit exact inference. Development of improved approximation techniques for Bayesian networks has in consequence become an active subject of research [28,34].

D'Ambrosio [9] classifies approximation procedures into two categories. *Approximate inference methods* compute distributions with special algorithms using the original network [8,41], and *model reduction methods* employ exact algorithms after simplifying the original network [27,33,49]. We can also distinguish approximation procedures in terms of the form of their outcomes: whether they produce point-valued approximations [8,33,49], or upper and/or lower bounds [12,23,25] of the desired probability distributions.

A common way of conducting approximate evaluation of Bayesian networks is to take advantage of special properties of the probability distributions encoded by the given network. Many attempt to find and ignore relatively irrelevant information in the network. For instance, Kjærulff and van Engelen propose to approximate based on weak dependence among random variables [27,49], and Liu and Wellman abstract away relatively unimportant states of random variables [33]. Other approaches, such as variational methods [25], exploit functional properties of the probability distributions.

In this paper, we report methods for computing bounds on probability distributions in Bayesian networks when some random variables have known qualitative influences on others. Given such a Bayesian network, we can construct an approximate network that encodes bounds on the original probability distributions. Evaluating this approximate network can yield bounds on the desired probability distributions at reduced computation time in applications [31]. We illustrate the application of these bounds in two contexts: resolving ambiguous qualitative probabilistic relationships among variables in Bayesian networks, and searching for the best travel plan in stochastic transportation networks. In the first application, it is possible to resolve the ambiguities

without using complete probabilistic information [30]. In the second, it is possible to find travel plans with bounded degrees of suboptimality [29,32].

Section 2 provides background information for the remainder of this paper. In Section 3, we explain how given qualitative influences among random variables allow us to transform a Bayesian network into another that expresses bounds on its original probability distributions. We operationalize the basic concepts in the context of a method for state-space abstraction, with specialized operations to exploit the given qualitative information [33]. We then report applications of the techniques to two different problems in the following sections. In Section 4, we apply the bounding procedures to derive additional qualitative relationships among random variables in Bayesian networks. Section 5 tackles path-planning problems in stochastic transportation networks. Our algorithm compares merits of paths based on bounds on probability distributions over travel times of alternative paths. We also discuss some generalization of our basic bounding concepts.

2. Background

Bayesian networks are directed acyclic graphs augmented with conditional probability tables. Nodes in the network represent variables, and links denote unconditional dependence relationships between variables [40]. We denote variables by capital letters and sets of variables by bold-faced capital letters. Since we use nodes to represent variables, we use "variables" and "nodes" interchangeably henceforth. Values of variables are called *states* and denoted by corresponding small letters. When necessary, we use superscripts to distinguish variables that belong to a set of variables, and subscripts to distinguish states of a variable. For instance, X represents a set of variables, and this set may contain three nodes X^1, X^2, and X^3. The state space of the variable X^1 may contain three possible states x_1^1, x_2^1, and x_3^1. The *cardinality* of a variable is the number of states in its state space. Hence, the cardinality of X^1 is three.

Each node in a Bayesian network has a conditional probability table that specifies the probability distribution of the node given the states of parent nodes of the node. Variables at the tail of the incoming links of X are *parents* of X, denoted $P(X)$, and variables at the head of the outgoing links of X are *children* of X, denoted $C(X)$. The definitions of parents and children have natural extensions. *Ancestors* of a node consist of its parents and, recursively, ancestors of its parents. Analogously, *descendants* of a node consist of its children, and recursively, descendants of its children. We use the shorthand $\Pr(x|p(X))$ to represent an entry, $\Pr(X = x|P(X) = p(X))$, in the conditional probability table associated with node X.

2.1. Qualitative relationships

A qualitative method for reasoning about uncertainty attempts to support valuable inferences without full numeric specification of uncertain relationships [38]. The qualitative relationships used in our algorithms were first defined for qualitative probabilistic networks (QPNs) [50]. QPNs are abstractions of Bayesian networks, with conditional probability tables summarized by the signs of qualitative relationships between variables. Each arc in the network is marked with a sign—positive (+), negative (−), or ambiguous (?)—denoting the sign of the qualitative probabilistic relationship between its terminal nodes.

The interpretation of such qualitative influences is based on *first-order stochastic dominance* (FSD) [15]. Let $F(x)$ and $F'(x)$ denote two cumulative distribution functions (CDFs) of a random variable X. Then $F(x) \text{FSD} F'(x)$ holds if and only if (iff)

$$F(x) \leqslant F'(x) \quad \text{for all } x.$$

We say that one node positively influences another iff the latter's conditional distribution is increasing in the former, all else equal, in the sense of FSD.

Definition 1 [50]. Let $F(y|x_i, \boldsymbol{px}(Y))$ be the cumulative distribution function of Y given $X = x_i$ and the rest of Y's parent nodes $\boldsymbol{PX}(Y) = \boldsymbol{px}(Y)$. We say that node X positively influences node Y, denoted $S^+(X, Y)$ as illustrated in Fig. 1, iff

$$\forall x_i, x_j, \boldsymbol{px}(Y), \quad x_i \leqslant x_j \Rightarrow F(y|x_j, \boldsymbol{px}(Y)) \text{FSD} F(y|x_i, \boldsymbol{px}(Y)).$$

Analogously, we say that node X *negatively influences* node Y, denoted $S^-(X, Y)$, when we reverse the direction of the dominance relationship in Definition 1. The arc from X to Y in that case carries a negative sign. When the dominance relationship holds for both directions, we denote the situation by $S^0(X, Y)$. However, this entails conditional independence, and so we typically do not have a direct arc from X to Y in this case. When none of the preceding relationships between the two CDFs hold, we put a question mark on the arc,
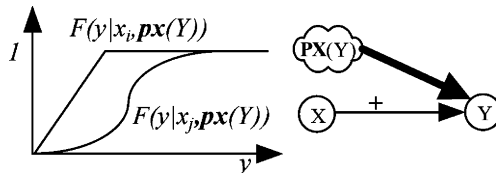

Fig. 1. X positively influences Y ($x_i \leqslant x_j$). The "cloud" represents all parents, excluding X, of Y, and the thick link represents the links from the cloud to Y.

and denote such situations as $S^?(X, Y)$. We may apply the preceding definitions to Boolean nodes under the convention that **false < true**.

A particularly useful implication of the dominance relationship $F_1(x) \text{FSD} F_2(x)$ [53] is that for all monotonically increasing functions g,

$$\int g(x) \, dF_1(x) \geqslant \int g(x) \, dF_2(x). \tag{1}$$

2.2. Bounds of probability distributions

Definition 2. A CDF $\overline{F}(x)$ is an upper bound of $F(x)$, if $F(x) \text{FSD} \overline{F}(x)$. A CDF $\underline{F}(x)$ is a lower bound of $F(x)$, if $\underline{F}(x) \text{FSD} F(x)$.

Our algorithms compute bounds of CDFs that are defined in terms of FSD relationships. As illustrated in Fig. 2, the curves for an upper and a lower bound run through the area, respectively, above and below the curve for the original CDF. Assume that X positively influences Y, and apply $E_H[Y] = \int_x g(y|x) \, dH(x)$ with (1). We can verify that the mean values, $E_{\overline{F}}[Y]$, $E_F[Y]$, and $E_{\underline{F}}[Y]$ of Y computed with $H(x) = \overline{F}(x)$, $H(x) = F(x)$, and $H(x) = \underline{F}(x)$ has the relationship: $E_{\overline{F}}[Y] \leqslant E_F[Y] \leqslant E_{\underline{F}}[Y]$.

Bounds on CDFs induce lower and upper probabilities [6] on particular events. Let M denote the event that $x_i < X \leqslant x_j$. The lower and upper probabilities of M, denoted by $\underline{\Pr}(M)$ and $\overline{\Pr}(M)$, respectively, are:

$$\underline{\Pr}(M) = \max(0, \underline{F}(x_j) - \overline{F}(x_i)) \tag{2}$$

and

$$\overline{\Pr}(M) = \overline{F}(x_j) - \underline{F}(x_i). \tag{3}$$

$\overline{\Pr}(M)$ is guaranteed to be between 0 and 1 since $1 \geqslant \overline{F}(x_j) \geqslant \overline{F}(x_i) \geqslant \underline{F}(x_i) \geqslant 0$. We can show, by simple algebraic manipulation, that the lower and upper probabilities defined in (2) and (3) have several desirable properties as bounds for the probability distribution of X, including superadditivity, subadditivity, and 2-monotone properties [6].
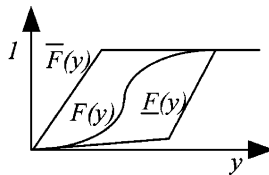

Fig. 2. Bounds of CDFs are defined in terms of FSD relationships.

2.3. State-space abstraction methods

In previous work [33], we reported an iterative state-space abstraction (ISSA) algorithm for approximate evaluation of Bayesian networks. The ISSA algorithm aggregates states of variables into *superstates* to construct abstract versions of the *original Bayesian networks* (OBNs) that specify exact probability distributions. We use these *abstract Bayesian networks* (ABNs) to compute point-valued approximations of the probability distributions of interest. To construct ABNs, we select some nodes, called *abstracted nodes*, from the OBNs, and aggregate their states. As a consequence of state aggregation, we need to assign the CPTs of both the abstracted nodes and their child nodes.

Since a superstate is an aggregation of original, consecutive states, its conditional probability is the sum of conditional probabilities of its constituent original states. Using $\widehat{\Pr}(\cdot)$ to denote probability values in abstract Bayesian networks, we have

$$\widehat{\Pr}([a_{k,l}]|\boldsymbol{p}(A)) = \sum_{j=k}^{l} \Pr(a_j|\boldsymbol{p}(A)), \tag{4}$$

where $[a_{k,l}]$ is the superstate aggregating a_k through a_l, $k \leqslant l$. We determine the CPTs of children of the abstracted nodes according to the *average policy* [33]:

$$\widehat{\Pr}(y|[a_{i,j}], \boldsymbol{px}(Y)) = \frac{1}{j-i+1} \sum_{k=i}^{j} \Pr(y|a_k, \boldsymbol{px}(Y)). \tag{5}$$

We adopt uniform assumptions in assigning probability values of the superstates, and weigh the components of Y's conditioning states equally. If we have information about the relative importance of $\Pr(y|a_k, \boldsymbol{px}(Y))$, such as marginal probabilities $\Pr(a_k, \boldsymbol{px}(Y))$ for some a_k and $\boldsymbol{px}(Y)$, we may assign the new conditional probability tables more precisely. In Section 3.2 we introduce a new policy that enables ISSA to compute bounds of cumulative distribution functions.

Algorithm 1 (*ISSA* [33]). 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.

The methods that we employ to select superstates for refinement at step 4 greatly influence the quality of approximations. The so-called Most Probable SuperState (MPSS) method splits the superstate that has the largest marginal probability [33]. Splitting the most probable superstate has the intuitive implication of making the ABNs more similar to their original OBNs, and has performed well in some of our tests. Some other control heuristics are discussed in the original work [33].

3. Bounding probability distributions

In this section we apply the state space abstraction methods [33] for computing bounds of probability distributions. The first subsection discusses the general concepts for computing bounds of selected conditional CDFs in Bayesian networks. The second subsection provides methods that realize the concepts in the ISSA algorithm. These methods take advantage of qualitative relationships among nodes for bounding probability distributions. We also present methods for tightening the bounds.

3.1. Strengthening and weakening operators

We can compute bounds of some probability distributions by strengthening and weakening selected CDFs in Bayesian networks. Let Y be a child of A, and denote the set of parents of Y excluding A by $\boldsymbol{PX}(Y)$.

Definition 3. We strengthen $F(y|a, \boldsymbol{px}(Y))$ with respect to A by replacing $F(y|a, \boldsymbol{px}(Y))$ with $F'(y|a, \boldsymbol{px}(Y))$ such that

$$F'(y|a, \boldsymbol{px}(Y)) \text{FSD} F(y|a, \boldsymbol{px}(Y)), \quad \text{for all } a.$$

We weaken $F(y|a, \boldsymbol{px}(Y))$ when the FSD relationship is reversed.

The most important effect of strengthening the CDF $F(y|a, \boldsymbol{px}(Y))$ with respect to A is to increase the probability of Y being a larger value for some states of A. For instance, if we replace $F(y|x_i, \boldsymbol{px}(Y))$ by $F(y|x_j, \boldsymbol{px}(Y))$ in Fig. 1, the cumulative probability for Y being smaller than any y is either reduced or not changed. Analogously, weakening $F(y|a, \boldsymbol{px}(Y))$ with respect to A implies that we decrease the probability of Y being a larger value for some states of A. Using these strengthening and weakening operations, we may compute the bounds of selected conditional probability distributions when some links in a Bayesian network can be marked with decisive qualitative signs: "+" or "−".
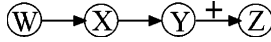
Fig. 3. A Bayesian network with a link marked with a decisive qualitative sign.

Example 1. Consider the network in Fig. 3, and assume that we strengthen $F(y|x)$ with respect to X by replacing $F(y|x)$ with $F'(y|x)$ such that $F'(y|x)\,\mathrm{FSD}\,F(y|x)$ for all states of X. Since $F(z|x) = \int_y F(z|y)\,\mathrm{d}F(y|x)$, we can apply (1) to analyze the effects of strengthening $F(y|x)$. Given that $S^+(Y, Z)$ implies that $F(z|y)$ is decreasing in y, our strengthening $F(y|x)$ will increase the probability of Y being a larger value, thereby decreasing $F(z|x)$ and obtaining a lower bound of $F(z|x)$ for all x. As a result, we also have obtained a lower bound of $F(z|w)$ for all w because $F(z|w) = \int_x F(z|x)\,\mathrm{d}F(x|w)$.

This example illustrates that we may compute bounds of CDFs by *locally* strengthening selected CDFs. Specifically, given $S^+(Y, Z)$, we are able to compute a lower bound of $F(z|w)$ by strengthening $F(y|x)$ with respect to X. The strengthening of $F(y|x)$ can be carried out by using the values in the conditional probability tables (CPT) associated with Y.

In the following theorem statements, we use *ancestral ordering* of nodes as defined below.

Definition 4 (cf. [35]). Let $\boldsymbol{J}$ denote a set of nodes $\{J^1, \ldots, J^n\}$ in a Bayesian network. $[J^1, \ldots, J^n]$ is an ancestral ordering of the nodes in $\boldsymbol{J}$ if, for every $J^i \in \boldsymbol{J}$, all the ancestors of J^i are ordered before J^i.

The following theorem presents conditions for computing bounds of a conditional CDF of a variable Z given the evidence $\boldsymbol{E} = \boldsymbol{e}$ by strengthening and weakening the distributions of the children of a distinguished node A. We call nodes whose values are instantiated *evidence nodes*, and we denote the set of evidence nodes by $\boldsymbol{E}$. Let $\boldsymbol{Y}$ be the children of A and Y^i be a node in $\boldsymbol{Y}$. The theorem is applicable when children of A meet the stated requirements. We denote the subset of the siblings of Y^i in $\boldsymbol{Y}$, $\boldsymbol{Y} \setminus \{Y^i\}$, by $\boldsymbol{SB}(Y^i)$, and we use the notation $S^{\sigma^i}(Y^i, Z \| \boldsymbol{e}, \boldsymbol{X})$ to represent that $S^{\sigma^i}(Y^i, Z)$ given $\boldsymbol{E} = \boldsymbol{e}$ and all possible instantiations of $\boldsymbol{X}$, where σ^i is a sign for the qualitative relationship between Y^i and Z. The notation $CI(X, Y, Z)$ means that X and Z are independent conditional on the values of Y.

Theorem 1. *Assume that*:

1. *For all i, $S^{\sigma^i}(Y^i, Z \| \boldsymbol{e}, \boldsymbol{SB}(Y^i))$, where σ^i is either +, −, or 0.*
2. *$CI(Z, \{\boldsymbol{E}, \boldsymbol{Y}\}, A)$.*
3. *$\boldsymbol{E}$, A, and $\boldsymbol{Y}$ appear in order in an ancestral ordering.*
4. *For all i, Y^i is not a descendant of nodes in $\boldsymbol{SB}(Y^i)$. $\boldsymbol{SB}(Y^j)$ for all j.*

When $\sigma^i = -$, we obtain, respectively, a lower bound and an upper bound of $F(z|e)$ by weakening and strengthening $F(y^i|a, px(Y^i))$ with respect to A. When $\sigma^i = +$, we obtain, respectively, an upper bound and a lower bound of $F(z|e)$ by weakening and strengthening $F(y^i|a, px(Y^i))$ with respect to A. When $\sigma^i = 0$, neither strengthening nor weakening $F(y^i|a, px(Y^i))$ with respect to A affects $F(z|e)$.

Proof. Proof for this theorem is in Appendix A.   □

Example 2. In the network shown in Fig. 4, the following conditions hold: (a) $S^+(Y^1, Z\|e, Y^2)$ and $S^-(Y^2, Z\|e, Y1)$ for all e, (b) $CI(Z, \{E, Y^1, Y^2\}, A)$, (c) $[E, A, Y^1, Y^2]$ is an ancestral ordering, and (d) Y^1 is not a descendant of Y^2 and vice versa. Therefore, we can obtain a lower bound of $F(z|e)$ for any $E = e$ by strengthening $F(y^1|a)$ or weakening $F(y^2|a)$ with respect to A.

Theorem 1 can be applied to cases where we strengthen and weaken multiple such $F(y^i|a, px(Y^i))$, as long as these strengthening and weakening operations are coordinated consistently so that the effects are to find a lower or an upper bound of $F(z|e)$. This extended interpretation of the theorem can be proved inductively as follows. The theorem dictates that we obtain a bound of the exact $F(z|e)$ by strengthening or weakening one particular $F(y^i|a, px(Y^i))$. Then, we can obtain a new bound of $F(z|e)$ by strengthening or weakening a $F(y^j|a, px(Y^j))$ in the ABN where some $F(y^k|a, px(Y^k))$ has been strengthened or weakened. Therefore, by induction, we may coordinate the strengthening and weakening operations to obtain lower (or upper) bounds of $F(z|e)$ by strengthening or weakening the conditional probability distributions of all nodes in Y with respect to A.

The first condition of Theorem 1 requires that Y^i have a nonambiguous qualitative relationship with Z. This qualitative relationship determines the selection of strengthening and weakening operations for computing desired bounds. The remaining conditions ensure that we can compute desired bounds by locally modifying $F(y^i|a, px(Y^i))$. Specifically, when A has multiple child nodes Y, we can, simultaneously and independently, strengthen or weaken the conditional probability distribution of each node in Y to obtained bounds of $F(z|e)$. Notice that this and the following theorem do not require a decisive
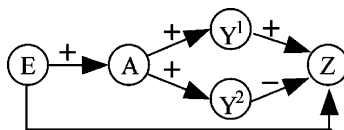

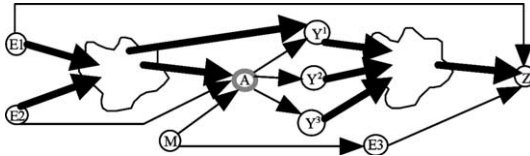Fig. 4. An applicable example for Theorem 1.

Fig. 5. Another applicable example for Theorem 1.

qualitative relationship between the evidence nodes E and the node of interest Z.

The third and the fourth restrictive conditions are required, sufficient conditions in the proof for Theorem 1. In the previous and the following examples, although E, A, and Y form different topologies, we can compute the bounds of the $F(z|e)$. However, when the restrictions are violated, e.g., abstracting ancestors of E, we might not obtain desired bounds.

Example 3. Theorem 1 may be applicable to networks that are as complex as the one shown in Fig. 5. In this network, we assume all links point from the left to the right-hand side, and we use thick links to represent bunches of links that might exist between clouds of nodes and individual nodes. With $E = \{E1, E2, E3\}$, we can verify that $CI(Z, \{E, Y\}, A)$ holds in this network. Also, $[E, A, Y]$ is an ancestral ordering, and Y^i is not a descendant of $SB(Y^i)$ for all Y^i. Therefore, Theorem 1 is applicable to any Y^i that satisfies the first condition in the theorem.

Theorem 1 cannot be applied to cases where A is a parent node of Z because Y and Z represent distinct nodes. The following theorem specifies conditions and methods for abstracting the parents of Z to compute bounds of $F(z|e)$.

Theorem 2. *In addition to conditions* 3 *and* 4 *of Theorem* 1, *assume that* $Z \in Y$. *We obtain, respectively, a lower and an upper bound of* $F(z|e)$ *by strengthening and weakening* $F(z|a, px(Z))$ *with respect to A.*

Proof. Proof for this theorem is in the Appendix A. □

Example 4. In the network shown in Fig. 4, we have (a) $[E, Y^1, Z]$ is an ancestral ordering and (b) Z is the only descendant of Y^1. Therefore, Theorem 2 is applicable, and we can obtain a lower bound of $F(z|e)$ by strengthening $F(z|y^1)$ with respect to Y^1. Analogously, we obtain a lower bound of $F(z|e)$ by strengthening $F(z|y^2)$ with respect to Y^2.

Theorems 1 and 2 also provide guidelines for obtaining tighter bounds. For convenience, we say that $G(x)$ is *less dominating than* $H(x)$ if $H(x)\mathrm{FSD}\,G(x)$

FSD $F(x)$. Roughly speaking, the following corollary, which follows from Theorem 1, states that we can obtain tighter bounds of $F(z|e)$ by setting $F(y^i|a, \boldsymbol{px}(Y^i))$ to a less (or more) dominating alternative.

Corollary 1. *Let $G(y^i|a, \boldsymbol{px}(Y^i))$ and $H(y^i|a, \boldsymbol{px}(Y^i))$ be alternatives for weakening $F(y^i|a, \boldsymbol{px}(Y^i))$ with respect to A. Assume that $H(y^i|a, \boldsymbol{px}(Y^i))$ is less dominating than $G(y^i|a, \boldsymbol{px}(Y^i))$ for all a and $\boldsymbol{px}(Y^i)$. Then, weakening $F(y^i|a, \boldsymbol{px}(Y^i))$ by $G(y^i|a, \boldsymbol{px}(Y^i))$ rather than $H(y^i|a, \boldsymbol{px}(Y^i))$ provides a tighter lower bound of $F(z|e)$ when $\sigma^i = -$, and a tighter upper bound of $F(z|e)$ when $\sigma^i = +$.*

Analogously, strengthening $F(y^i|a, \boldsymbol{px}(Y^i))$ by a less dominating CDF provides a tighter upper bound of $F(z|e)$ when $\sigma^i = -$, and a tighter lower bound of $F(z|e)$ when $\sigma^i = +$.

Similarly, we may derive the following corollary from Theorem 2.

Corollary 2. *Applying Theorem 2, we obtain tighter lower (upper) bounds of $F(z|e)$ by setting $F(z|a, \boldsymbol{px}(Z))$ to a less (more) dominating alternative.*

Notice that neither Theorem 1 nor Theorem 2 requires any particular qualitative relationship between A and nodes in Y. The existence of a particular qualitative relationship between A and nodes in Y facilitates, but is not required for, the application of the theorems.

3.2. State-space abstraction with the dominance policy

We can exploit these bounding relationships in ISSA by replacing the policy for determining the CPTs of children of abstracted nodes. In place of the average policy (5), we adopt the *dominance policy*, which sets the CPTs to extreme values. We may choose to strengthen or weaken selected conditional probability distributions, depending on whether we want to compute lower or upper bounds of the desired CDFs. Let Y be a child node of abstracted node A, and $\boldsymbol{PX}(Y)$ be the subset of parent nodes of Y excluding A. If we choose to *strengthen* the conditional probability distributions of Y with respect to A, we assign the CPT of Y as follows:

$$\widehat{F}(y|[a_{k,l}], \boldsymbol{px}(Y)) = \min_{j \in [k,l]} F(y|a_j, \boldsymbol{px}(Y)). \tag{6}$$

Similarly, to *weaken* the conditional probability distributions of Y with respect to A, assign

$$\widehat{F}(y|[a_{k,l}], \boldsymbol{px}(Y)) = \max_{j \in [k,l]} F(y|a_j, \boldsymbol{px}(Y)). \tag{7}$$

Application of (6) and (7) becomes easier when the links from the abstracted node to its child nodes can be marked with decisive qualitative signs. For

instance, if the sign from A to Y is "+", we can move the min operator into $F(\cdot)$ in (6). The operator becomes $\max_{j\in[k,l]} a_j$, so the right-hand side becomes $F(y|a_l, \boldsymbol{px}(Y))$. Analogously, using the same rewriting procedure, the right-hand side of (7) becomes $F(y|a_k, \boldsymbol{px}(Y))$. However, the application of (6) and (7) does not require any particular qualitative relationship between A and Y.

3.2.1. Single node abstraction

We discuss the application of Theorem 1 and Corollary 1. The application of Theorem 2 and Corollary 2 is analogous. We operationalize Theorem 1 using the dominance policy in the state-space abstraction methods. To compute bounds of the desired CDFs $F(z|\boldsymbol{e})$, we can abstract the state space of any node A that meets the conditions of the theorem. We may apply the inference algorithms for QPNs [14] to locate those nodes whose children have unambiguous qualitative relationships with Z as specified in the first condition of the theorem. We apply (4) to assign the CPT of A, and we apply (6) or (7) to assign the CPTs of the child nodes of A. The selection of (6) or (7) depends on whether we want to compute lower or upper bounds of the desired CDFs, and Theorem 1 provides guidelines for the selection.

The CDF $F(z|\boldsymbol{e})$ specified in the ABNs constructed with the dominance policy is a bound of the CDFs of the $F(z|\boldsymbol{e})$ specified in the OBNs. This is due to Theorem 1 and the fact that we can show that the effects of applying the dominance policy in abstracting nodes are equivalent to strengthening (or weakening) the conditional probability distributions of the children of the abstracted nodes.

To prove this, we need to compare the probability values of the states a_j, $j = k, \ldots, l$, aggregated in a superstate $[a_{k,l}]$ in the OBNs and ABNs in order to apply the aforementioned theorems in analyzing the effects of the dominance policy. In terms of Theorem 1, we want to strengthen or weaken $F(y^i|a, \boldsymbol{px}(Y^i))$ to obtain desired bounds when we abstract node A. However, the state space of A in an ABN is smaller than that of A in the OBN due to state space abstraction, so we do not have corresponding $\Pr(y^i|a_j, \boldsymbol{px}(Y^i))$ for all $j = k, \ldots, l$ in the ABN. Fortunately, we may show that the strengthening and weakening operations in the dominance policy have the effects of strengthening and weakening distributions that we defined in Section 3.1. We show this by transforming the ABNs into equivalent abstract Bayesian networks and comparing these equivalent networks with the OBNs.

Definition 5 [33]. An equivalent abstract Bayesian network (EABN) adopts the state space of the OBN, and preserves the joint probability distribution of the unabstracted nodes in the ABN.

We have shown that it is already possible to construct an EABN based on the given OBN and its ABN [33]. When we abstract only one node, A, in the

OBN to obtain the ABN, we construct the EABN by duplicating the graphical structure of the OBN. Then we set the CPTs of A and its children in the EABN, denoted by $\widetilde{\Pr}(\cdot)$, by the following formula when $l \in [s_i, t_i]$.

$$\widetilde{\Pr}(a_l|\boldsymbol{p}(X)) = \Pr(a_l|\boldsymbol{p}(X)) \quad \text{and}$$
$$\widetilde{\Pr}(y^j|a_l, \boldsymbol{px}(Y^j)) = \widehat{\Pr}(y^j|[a_{s_i,t_i}], \boldsymbol{px}(Y^j)). \tag{8}$$

Notice that the CPTs of the children of the abstracted node in the EABN are copied from those in the ABN. The procedure can be generalized to multiple nodes as well [33]. Given the preservation of the joint distribution of the un-abstracted nodes, we can use the EABN as a surrogate for the ABN when we look into the probability distributions of the unabstracted nodes.

Now that the OBN and the EABN have the same state space, we can compare their probabilities. As described above, the CPTs of the OBN and the EABN differ only in the CPTs of the children of the abstracted node. Specifically, (8) reveals the difference. The right-hand side of (8) comes from either (6) and (7), depending on whether we strengthen or weaken the target distributions. If we apply (6), we will have, for all a_j, $\widetilde{F}(y|a_j, \boldsymbol{px}(Y)) \leqslant F(y|a_j, \boldsymbol{px}(Y))$. Namely, the main difference between the OBN and the EABN is that $\widetilde{F}(y|a_j, \boldsymbol{px}(Y))\text{FSD}\,F(y|a_j, \boldsymbol{px}(Y))$. Analogously, if we apply (7), the difference will be that $F(y|a_j, \boldsymbol{px}(Y))\text{FSD}\,\widetilde{F}(y|a_j, \boldsymbol{px}(Y))$.

Recall that we have proved that the EABN and the ABN have the same joint distribution of the unabstracted nodes in [33]. Hence, we infer that, as long as we do not abstract the evidence nodes, $\boldsymbol{E}$, and the target node, Z, we can obtain lower and upper bounds of $F(z|e)$ by respectively employing (6) and (7) when we abstract A that meets all conditions in Theorem 1.

Furthermore, we can show that ISSA returns bounds that tighten in each iteration using Corollary 1. The tightening bounds are due to the fact that, as we split superstates, the reassigned CDFs, respectively, become less and more dominating when we strengthen and weaken the original CDFs. Consider the case in which we want to strengthen $F(y|a, \boldsymbol{px}(Y))$ with respective to A. When we split the superstate $[a_{k,l}]$ into two superstates $[a_{k,m}]$ and $[a_{m+1,l}]$, $m \in (k, l)$, $F(y|[a_{k,l}], \boldsymbol{px}(y))$ for every y is replaced with corresponding $F(y|[a_{k,m}], \boldsymbol{px}(y))$ and $F(y|[a_{m+1,l}], \boldsymbol{px}(y))$. Using (6), we can easily verify that $F(y|[a_{k,l}], \boldsymbol{px}(y))$ is not larger than $F(y|[a_{k,m}], \boldsymbol{px}(y))$ and $F(y|[a_{m+1,l}], \boldsymbol{px}(y))$. As a result, the newly reassigned CDFs are less dominating, and, according to Corollary 1, the bounds of $F(z|e)$ tighten in each iteration of ISSA.

3.2.2. Multiple node abstraction

We may compute bounds of CDFs by abstracting multiple nodes that do not share child nodes. With an analogous method used in the previous section, we also can show that the bounds obtained by evaluating network with multiple abstracted nodes tighten as we split superstates.

We assume that the purpose of abstracting nodes is to compute an upper bound of $F(z|e)$. The steps and analysis for computing lower bounds are again analogous. We analyze the effects of abstracting multiple nodes by assuming that we abstract one node at a time and that we abstract nodes A^i, $i = 1, 2, \ldots, m$. Let ABN^i denotes the ABN that is constructed by sequentially abstracting node A^1 up to node A^i. Clearly, by applying the result from the previous section, the $F(z|e)$ specified in ABN^i is an upper bound of the $F(z|e)$ specified in ABN^{i-1} since one more node is abstracted in ABN^i than in ABN^{i-1}. Hence, by induction, we can show that the $F(z|e)$ specified in ABN^m is an upper bound of the $F(z|e)$ specified in the OBN.

The remaining problem is to show that the ABN^m that is constructed by sequentially abstracting node A^1 through A^m is the same as the ABN that is constructed by simultaneously abstracting all A^is. Recall that we use maximal and minimal operations for strengthening and weakening CDFs in (6) and (7). As a result, the order that we abstract the nodes matters only when the abstracted nodes share child nodes. When abstracted nodes share child nodes, abstracting nodes in different orders may result in different abstract networks due to the fact that maximal and minimal operations are not commutative. However, when the abstracted nodes do not share child nodes as in our case, the ordering of these nodes being abstracted will not affect the resulting network, so the ABN^m that is constructed by sequentially abstracting node A^1 through A^m is the same as the ABN that is constructed by abstracting all A^is in any order. Therefore, we have shown that we can abstract multiple nodes that do not share child nodes to obtain bounds of CDFs.

4. Application: tradeoff resolution in Bayesian networks

In this and the following sections we demonstrate the applications of the techniques of bounding probability distributions to the tasks of tradeoff resolution in Bayesian networks and path planning in stochastic transportation networks, respectively.

Researchers in uncertain reasoning regularly observe that to reach a desired conclusion (e.g., a decision), full precision in probabilistic relationships is rarely required, and that in many cases purely qualitative information (for some conception of "qualitative") is sufficient [16]. In consequence, the literature has admitted numerous schemes attempting to capture various forms of qualitative relationships [37,43,51], useful for various uncertain reasoning tasks. Unfortunately, we generally lack a robust mapping from tasks to the levels of precision required, and indeed, necessary precision is inevitably variable across problem instances. As long as some potential problem might require precision not captured in the qualitative scheme, the scheme is potentially inadequate for the associated task. Advocates of qualitative uncertain rea-

soning typically acknowledge this, and sometimes suggest that one can always revert to full numeric precision when necessary. However specifying a numerically precise probabilistic model as a fallback preempts any potential model-specification benefit of the qualitative scheme, and so it seems that one may as well use the precise model for everything. [1] This is perhaps the primary reason that qualitative methods have not seen much use in practical applications of uncertain reasoning to date.

The case for qualitative reasoning in contexts where numerically precise models are available must appeal to benefits other than specification, such as computation. Cases where qualitative properties justify computational shortcuts are of course commonplace (e.g., independence), though we do not usually consider this to be qualitative reasoning unless some inference is required to establish the qualitative property itself in order to exploit it. Since pure qualitative inference can often be substantially more efficient than its numeric counterpart (e.g., in methods based on infinitesimal probabilities [17] or ordinal relationships [14]), it is worth exploring any opportunities to exploit qualitative methods even where some numeric information is required.

4.1. The tradeoff resolution task

We consider the task of deriving the qualitative relationship between a pair of variables in a Bayesian network. From an abstracted version of the network, where all local relationships are described qualitatively, we can derive the entailed sign between the variables of interest efficiently using propagation techniques [14].

If we are fortunate, we may acquire decisive answers from the qualitative inference algorithms. Often, however, the results of such qualitative reasoning are ambiguous. This might be because the relationship in question actually is ambiguous (i.e., nonmonotone or context-dependent), or due to loss of information in the abstraction process.

This can happen, for instance, when there are competing influential paths from the source node—whose value is tentatively modified—to the target node—whose change in value is of interest. For example, whereas accept flu shots may decrease the probability of get flu, it also increases the probability and degree of feel pain. On the other hand, increasing either get flu or feel pain decreases overall bodily well-being, all else equal. [2] As a result, qualitative

---

[1] If the qualitative formalism is a strict abstraction, then any conclusions produced by the precise model will agree at the qualitative level. Even in such cases, qualitative models may have benefits for explanation or justification [22], as they can indicate something about the robustness of the conclusions (put another way, they can concisely convey broad classes of conclusions).

[2] We interpret low < medium < high as ordinal values.

AFS		GF	AFS=true	AFS=false	FP	AFS=true	AFS=false
true	0.7	true	0.1	0.4	true	0.8	0
false	0.3	false	0.9	0.6	false	0.2	1

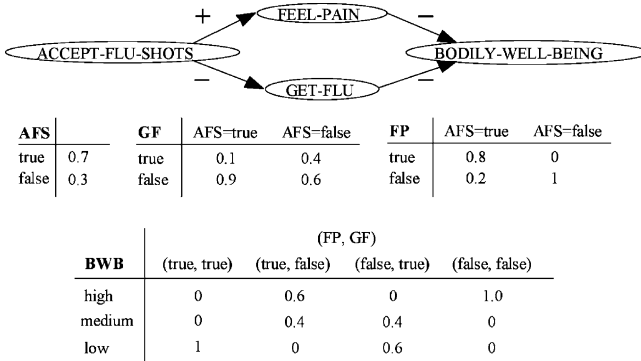|	(FP, GF)			
BWB	(true, true)	(true, false)	(false, true)	(false, false)
high	0	0.6	0	1.0
medium	0	0.4	0.4	0
low	1	0	0.6	0

Fig. 6. A simple case of qualitative ambiguity.

reasoning about the problem of whether we should accept flu shots will yield only an ambiguous answer. The situation is illustrated by the QPN in Fig. 6, where there is one positive path and one negative path from accept flu shots to bodily well-being. The combination of these two paths is qualitatively ambiguous. Worse, the ambiguity of this relationship would propagate within any network for which this pattern forms a subnetwork. For example, if this issue plays a role in a decision whether to go to a doctor, the result would be ambiguous regardless of the other variables involved.

Had we applied more precise probabilistic knowledge, such as a numerically specified Bayesian network, the result may have been decisive. Indeed, if accept flu shots and bodily well-being are binary, then a fully precise model is by necessity qualitatively unambiguous. However, performing all inference at the most precise level might squander some advantages of the qualitative approach. In the developments below, we consider some ways to apply numeric inference incrementally, to the point where qualitative reasoning can produce a decisive result.

Definition 6. Tradeoff resolution is the inference task that computes the qualitative relationships between two random variables in a given Bayesian network.

This task of obtaining more specific qualitative relationships between variables has attracted the interest of several researchers. Parsons and Dohnal discuss a semiqualitative approach for inference using Bayesian networks [39], based on a calculus for computing probability intervals for network variables. Osseiran [36] quantizes ranges of variables into seven intervals in a style similar to interval-based probabilities [18], and propose a calculus for this special class of Bayesian networks.

Some tackle the task by introducing more qualitative relationships, hoping to find decisive qualitative relationships that would be hidden otherwise. Parsons exploits more special numerical relationships among variables, and define *positively categorically influences* and *negatively categorically influences* [37]. Renooij and van der Gaag take a similar approach, and introduce more specific qualitative relationships such as *strongly positive qualitative influence* (++) and *strongly negative qualitative influence* (−−) in QPNs [43]. Renooij et al. also explore the application of context-specific qualitative influences [44], extending the concept and employing it in an augmented sign-propagation algorithm [45].

Before presenting our techniques, we demonstrate that in the worst case, qualitative tradeoff resolution in Bayesian networks may be no easier than full numerical inference.

4.2. Tradeoff resolution in Bayesian networks is NP-hard

Theorem 3. *Tradeoff resolution in Bayesian networks is NP-hard.*

Proof. We reduce the problem of computing absolute approximations to the task of qualitative tradeoff resolution. An estimate γ is an absolute approximation of $\Pr(y)$ if

$$\Pr(y) - \delta \leqslant \gamma \leqslant \Pr(y) + \delta,$$

where δ is the range of error. The problem of computing absolute approximations has been shown NP-hard [7].

Consider the task of computing absolute approximations for $\Pr(y)$ in a given Bayesian network in which Y is a Boolean variable. We construct a corresponding tradeoff resolution problem for this task as follows. The network for this tradeoff resolution problem includes the given Bayesian network and two Boolean variables D and T. The tradeoff resolution task is to determine the qualitative influence of D on T in the network shown in Fig. 7. The cloud where Y resides represents the given Bayesian network. We use x and $\bar{x}$ to denote that X is true and false, respectively.
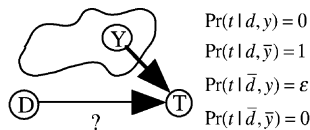

$$\Pr(t \mid d, y) = 0$$
$$\Pr(t \mid d, \bar{y}) = 1$$
$$\Pr(t \mid \bar{d}, y) = \varepsilon$$
$$\Pr(t \mid \bar{d}, \bar{y}) = 0$$

Fig. 7. Resolving qualitative influence of D on T yields an approximate probability for Y.

To check the overall influence of D on T, we need to know whether $\Pr(t|d) \geqslant \Pr(t|\bar{d})$. Using the data shown in Fig. 7,

$$
\begin{aligned}
&\Pr(t|d) \geqslant \Pr(t|\bar{d}) \\
&\iff \Pr(t|d,y)\Pr(y) + \Pr(t|d,\bar{y})\Pr(\bar{y}) \geqslant \Pr(t|\bar{d},y)\Pr(y) + \Pr(t|\bar{d},\bar{y})\Pr(\bar{y}) \\
&\iff \Pr(\bar{y}) \geqslant \varepsilon\Pr(y) \\
&\iff 1 - \Pr(y) \geqslant \varepsilon\Pr(y)
\end{aligned}
\tag{9}
$$

Thus, the overall influence of D on T is positive iff $\Pr(y) \leqslant \frac{1}{1+\varepsilon}$. Notice that the range of $\frac{1}{1+\varepsilon}$ is $[1/2, 1]$ when we vary ε between 1 and 0.

Therefore, using an efficient algorithm for tradeoff resolution, we can efficiently determine the range of $\Pr(y)$ if its range is in $[1/2, 1]$. We set ε to a very small number $\alpha \in [0, 1]$ in Fig. 7 such that (9) holds. We can apply the efficient algorithm for tradeoff resolution to verify whether a selected α satisfies (9), since the resolved qualitative relationship dictates the truth of $\Pr(t|d) \geqslant \Pr(t|\bar{d})$ and $\Pr(y) \leqslant \frac{1}{1+\alpha}$. Once we have this very small α, we can gradually set ε to multiples of α, i.e., $2\alpha, 3\alpha, \ldots, n\alpha$, and 1, where $n = \lfloor 1/\alpha \rfloor$, until $\Pr(t|d) \geqslant \Pr(t|\bar{d})$ does not hold. Let $m\alpha$ be the smallest multiple of α that makes $\Pr(t|d) \geqslant \Pr(t|\bar{d})$ fail. We have $\frac{1}{1+m\alpha} < \Pr(y) \leqslant \frac{1}{1+(m-1)\alpha}$.

As a result, when the range of $\Pr(y)$ is in $[1/2, 1]$, we will find a more precise range of $\Pr(y)$ by executing the efficient algorithm for tradeoff resolution $O(\frac{1}{\alpha})$ times. Similarly, we can determine the range of $\Pr(y)$ when the range is in $[0, 1/2]$, by executing the efficient algorithm for tradeoff resolution $O(\frac{1}{\alpha})$ times. This can be done by setting $\Pr(t|d,\bar{y})$ and $\Pr(t|\bar{d},y)$ to ε and 1 in Fig. 7, respectively. Therefore, the problem of computing absolute approximations in Bayesian networks can be solved efficiently if we have an efficient algorithm for qualitative tradeoff resolution. $\quad\square$

4.3. Tradeoff resolution via node reduction

4.3.1. Node reduction

The idea of incremental marginalization is to reduce the network node-by-node until the result is qualitatively unambiguous. The basic step is Shachter's arc reversal operation.

Theorem 4 [47]. *If there is an arc from node X to node Y in the given Bayesian network, and no other directed paths from X to Y, then we may transform the network to one with an arc from Y to X instead. In the new network, X and Y inherit each other's parent nodes.*

Let $\boldsymbol{P}_X$, $\boldsymbol{P}_Y$, and $\boldsymbol{P}_{XY}$ respectively denote X's own parent nodes, Y's own parent nodes, and X and Y's common parent nodes in the original network, and let $\boldsymbol{P}_{Y'} = \boldsymbol{P}_Y - \{X\}$. The new conditional probability distributions of Y and X are:

$$\mathrm{Pr}^{\mathrm{new}}(y|\boldsymbol{p}_X,\boldsymbol{p}_{Y'},\boldsymbol{p}_{XY}) = \sum_X \mathrm{Pr}^{\mathrm{old}}(y|\boldsymbol{p}_Y,\boldsymbol{p}_{XY})\,\mathrm{Pr}^{\mathrm{old}}(x|\boldsymbol{p}_X,\boldsymbol{p}_{XY}),$$

$$\mathrm{Pr}^{\mathrm{new}}(x|y,\boldsymbol{p}_X,\boldsymbol{p}_{Y'},\boldsymbol{p}_{XY}) = \frac{\mathrm{Pr}^{\mathrm{old}}(y|\boldsymbol{p}_Y,\boldsymbol{p}_{XY})\,\mathrm{Pr}^{\mathrm{old}}(x|\boldsymbol{p}_X,\boldsymbol{p}_{XY})}{\mathrm{Pr}^{\mathrm{new}}(y|\boldsymbol{p}_X,\boldsymbol{p}_{Y'},\boldsymbol{p}_{XY})}. \tag{10}$$

On reversing all the outgoing arcs from node X, the node becomes *barren* and can be removed from the network. By appropriate arrangements, we can remove a node by local operations. The net effect of reversing arcs and removing barren nodes as described is equivalent to marginalizing node X from the network [47].

4.3.2. Incremental node reduction

Consider the QPN shown on the left-hand side of Fig. 8. Since there exist both a positive path (through X) and a negative path (direct arc) from W to Z, the qualitative influence of W on Z is ambiguous. This local "?" would propagate throughout the network, necessarily ambiguating the relationship of any predecessor of W to any successor of Z.

Once we have detected the source of such a local ambiguity, we may attempt to resolve it by numerically marginalizing node X. The new sign on the direct arc from W to Z can be determined by inspecting the new conditional probability table of Z, given by (10). If we are fortunate, the qualitative sign σ may turn out to be decisive, in which case we have resolved the tradeoff.

This example illustrates the main idea of the incremental marginalization approach to resolving tradeoffs in QPNs. If we obtain an unambiguous answer to the desired qualitative relationship from the reduced network after marginalizing a selected node, then there is no need to do further computation. If the answer is still ambiguous, we may select other nodes to marginalize. The iteration continues until a decisive answer is uncovered. We present the skeleton of the *I*ncremental *T*rade*O*ff *R*esolution algorithm below. The algorithm is designed to answer queries about the qualitative influence of a decision node on a target node, using a given strategy for selecting the next node to reduce.

Algorithm 2. ITOR(decision, target, strategy)

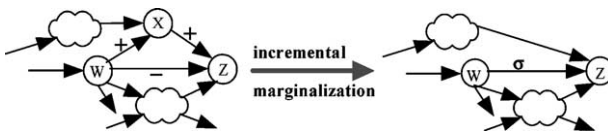1. Remove nodes that are irrelevant to the query about decision's influence on target [48].

Fig. 8. Marginalizing X potentially resolves the qualitative influence of W on Z.

2. Attempt to answer the query via qualitative inference [14].
3. If the answer to the query is decisive, exit; otherwise continue.
4. Select a node to reduce according to strategy. If there is no node that can be reduced, return "?", else perform the node reduction, and calculate the qualitative abstractions of the transformed relationships. Return to step 2.

We expect the incremental approach to improve performance over purely numeric inference on average. Since qualitative inference is quadratic whereas exact inference in Bayesian networks is exponential in the worst case, the qualitative inference steps do not add appreciably to computation time. On the other hand, when the intermediate results suffice to resolve the tradeoff, we save numeric computation over whatever part of the network is remaining.

4.3.3. Prioritizing node reduction operations

The objective of carrying out node-reduction operations in ITOR is to resolve qualitative tradeoffs. The optimal strategies for respective tasks will differ, in general. For example, a node that is very expensive to reduce at a certain stage of the evaluation might offer the best prospect for resolving the tradeoff.

We exploit intermediate information provided in qualitative belief propagation [14] in determining which node to reduce next. If we can propagate a decisive qualitative influence from the decision node D all the way to the target node T, we will be able to answer the query. Otherwise, there must be a node X that also has an indecisive relationships with D. Recall that we have pruned nodes irrelevant to the query, so any nodes that have indecisive relationship with D will eventually make the relationship between D and T indecisive. We have identified several conceivable strategies based on this observation, and report on experience with two of them.

The first strategy is to reduce node X, as long as X is not the target node T. When X is actually T, we choose to reduce the node Y that passed the message to X changing its qualitative sign from a decisive one to "?". However, this Y cannot be D itself. If it is, then either (1) there are only two nodes remaining in the network, and there is no decisive answer to the query, or (2) there are other nodes, and we randomly pick among those adjacent to D or T.

The second strategy is similar to the first, except that we exchange the priority of reducing X and Y. We handle the situations where X and/or Y happen to be D and/or T in the same manner as in the first strategy.

These strategies have the advantage that finding the next node to reduce does not impose extra overhead in the ITOR algorithm. The selection is a by-product of the qualitative inference algorithm. However neither of these strategies (nor any that we know) is guaranteed to minimize the cost of resolving the tradeoff.

Renooij and colleague also exploit the existence of the node X with an indecisive relationship with D, when there is an indecisive qualitative relationship between D and T for resolving qualitative tradeoffs in QPNs [46]. They identify the *pivot node* that has an ambiguous relationship with D, and, as an extra qualification, this pivot node must have a decisive qualitative relationship with T if the pivot node itself had a decisive node sign. It is shown that, assuming that there is only one new observation and that we are interested in the resulting qualitative sign of one node, the pivot node is unique. Having identified the pivot node, they then look for a set of special nodes, called *resolution frontier*, that would lead to a decisive qualitative sign of the pivot node, if all nodes in the resolution frontier have decisive node signs. The qualitative sign of the pivot node is determined by comparing the strengths of the competing influences over the pivot node from the nodes in the resolution frontier. The preprocessing steps, including the identification of the pivot node and the resolution frontier, systematically pinpoint the subnetwork that is directly relevant to the tradeoff resolution task. They then infer the qualitative influence of interest using information in the subnetwork that contains the pivot node and the resolution frontier as a whole. The result is the qualitative sign of the pivot node conditional on qualitative signs of the nodes in the resolution frontier. In contrast, we incrementally marginalize selected nodes that have ambiguous relationships with D in the relevant subnetwork until the resolution of the tradeoffs. Also, as a result of our not employing the concept of the pivot node, it is possible that we may marginalize some nodes that will not affect the resulting qualitative relationship of interest, although these nodes will affect the degree of influence quantitatively.

4.3.4. Experimental study

We have tested the effectiveness of the algorithm using randomly generated network instances. The experiments are designed to examine how connectivity of the network, sizes of state spaces, and strategies for scheduling node reduction affect the performance of the algorithm.

In the experiments, we use Bayesian networks in which arcs can be assigned decisive qualitative signs. To this end, we construct QPNs with only decisive signs on arcs, and then use the signs to govern the way we assign conditional probability values for nodes in their corresponding Bayesian network. The conditional probability distributions of nodes and the qualitative signs on arcs must agree with each other.

To create a random QPN with n nodes and l arcs, we first create a complete directed acyclic graph (DAG) with n nodes. Each arc in this DAG is assigned a random number that is sampled from a uniform distribution. We then attempt to remove the arc with the largest assigned number, under the constraint that the DAG remains connected. If removing the arc with the largest assigned number will make the DAG disconnected, we will attempt to remove the arc

with the next largest number. We remove arcs until the DAG contains only l arcs. After creating the network structure, we randomly assign qualitative signs (positive or negative) to the arcs.

We then build a Bayesian network that corresponds to the generated QPN, that is, respects its structure and qualitative signs. We select the cardinality of each node by sampling from a uniform distribution over the range $[2, \mathrm{MC}]$, where MC denotes the maximum state-space cardinality. For nodes without parents, we assign prior probabilities by selecting parameters from a uniform distribution and then normalizing.

For a node X with parent nodes $\boldsymbol{P}(X)$, the qualitative signs in the QPN dictate a partial ordering of the conditional probability distributions for various values of X, where the distributions are ordered based on the FSD relationship. Let $\boldsymbol{pa}_i(X)$ denote an instantiation of the parent nodes of X. To enforce this ordering, we identify the $\boldsymbol{pa}_i(X)$ such that the distribution $F(x|\boldsymbol{pa}_i(X))$ must dominate distributions $F(x|\boldsymbol{pa}_j(X))$ for all other $\boldsymbol{pa}_j(X)$. We assign the parameters for $F(x|\boldsymbol{pa}_i(X))$ (as for priors) by sampling from a uniform distribution. We then assign the remaining distributions in stages, at each stage setting only those distributions dominated by the previously assigned distributions. We make these assignments using the same random procedure, but under the constraint that the resulting distribution must respect the qualitative signs given the previous assignments.

In each experiment, we specify the number of nodes, the number of arcs, and maximum cardinality of state spaces for the randomly generated networks. In all experiments, we create networks with 10 nodes before pruning. We query the qualitative influence from the node 1 to node 10, and disregard the instances in which the answer is ambiguous after exact evaluation of the network.

Since the first step of the ITOR algorithm prunes nodes irrelevant to the query, the network actually used in inference is usually simpler than the original network. Table 1 shows the statistics collected for the first and the second node selection strategy. In both tables, we record the *average* number of nodes and links after the pruning step. There is no significant difference between the performance achieved by these two strategies.

We measure the performance of ITOR with two metrics. The first metric, R_{nodes}, is the ratio of the number of reduced nodes when the decisive answer is found to the number of nodes that would be reduced in exact numerical evaluation. The second metric, $R_{\mathrm{reversals}}$, is the ratio of number of arc reversal operations already done when the solution is found to the number of arc reversal operations that would be carried out for exact numerical evaluation. The latter figure is based on an arbitrary strategy for reducing the remaining network after the tradeoff is resolved, however, and so would tend to be an optimistic estimate of the saving. Table 1 reports averages for each metric. The savings due to incremental tradeoff resolution are $1 - R_{\mathrm{nodes}}$ and $1 - R_{\mathrm{reversals}}$, respectively, and so lower values of the metrics indicate better performance.

Table 1
Experimental results for the prioritizing strategies

Nodes	Links	MC	R_{nodes}	$R_{\text{reversals}}$
(a) The first strategy				
8.0	14.2	2	0.697	0.722
8.0	14.4	3	0.730	0.754
9.2	26.1	2	0.846	0.869
9.4	26.8	3	0.855	0.874
(b) The second strategy				
7.9	14.2	2	0.697	0.734
8.0	14.4	3	0.731	0.767
9.2	26.2	2	0.848	0.886
9.4	26.8	3	0.861	0.895

Each experiment runs ITOR over 10000 random networks with decisive influence from node 1 to node 10.

The statistics in both tables suggest that ITOR offers greater performance for sparsely connected networks and smaller state spaces. A possible reason for this phenomenon is that, as the cardinality and connectivity increased, the "numerical cause" that made the qualitative relationship of interest appear to be ambiguous became more complex. For instance, resolving the qualitative ambiguity might require us to fully utilize numerical information of more nodes due to the increased connectivity. Although these experimental results support the intuition that ITOR provides chance to resolve qualitative ambiguities at reduced computational costs, we do not consider the results as being conclusive. Further experimentation may lead us to more precise characterization of the expected savings achievable through incremental marginalization.

4.4. Tradeoff resolution via approximation

Since only qualitative relationships among variables are of interest, exact calculation of the values of the CDFs may not be necessary if we can use approximate CDFs to determine whether FSD holds. In this section, we apply the dominance policy in ISSA for resolving ambiguous qualitative relationships.

4.4.1. Motivation

Consider the task of determining whether $F(x|d_i)$ dominates $F(x|d_j)$. Assume that we have ways to control approximation methods to obtain approximate CDFs $\widehat{F}(x|d_i)$ and $\widehat{F}(x|d_j)$ such that $F(x|d_i) \leqslant \widehat{F}(x|d_i)$ and $\widehat{F}(x|d_j) \leqslant F(x|d_j)$ for all x. Given these approximate CDFs, $F(x|d_i)\,\text{FSD}\,F(x|d_j)$ will hold if we have
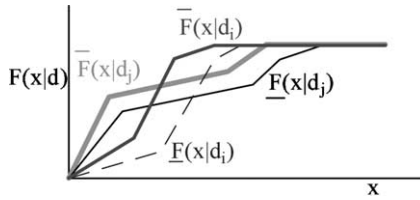
Fig. 9. Bounds of $F(x|d)$ imply $S^?(D,X)$.

$\widehat{F}(x|d_i) \leqslant \widehat{F}(x|d_j)$ for all x. In other words, it is possible to determine qualitative relationship using bounds of probability distributions.

In terms of bounds defined in Definition 2, $S^-(D,X)$ holds if there exist $\overline{F}(x|d_i)$ and $\underline{F}(x|d_j)$ such that

$$\text{for all } x, \quad d_i < d_j \Rightarrow \overline{F}(x|d_i) \leqslant \underline{F}(x|d_j). \tag{11}$$

Similarly, $S^+(D,X)$ holds if there exist $\overline{F}(x|d_j)$ and $\underline{F}(x|d_i)$ such that

$$\text{for all } x, \quad d_i < d_j \Rightarrow \overline{F}(x|d_j) \leqslant \underline{F}(x|d_i). \tag{12}$$

In addition, we may be able to tell that D neither positively nor negatively influences T by examining bounds. Specifically, a sufficient condition for $S^?(D,X)$ is that there exist x_r, x_s, and bounds such that, for some $d_i < d_j$,

$$\overline{F}(x_r|d_i) < \underline{F}(x_r|d_j) \quad \text{and} \quad \overline{F}(x_s|d_j) < \underline{F}(x_s|d_i). \tag{13}$$

When (13) holds, the curves for $F(x|d_i)$ and $F(x|d_j)$ must intersect as illustrated in Fig. 9.

4.4.2. Tradeoff resolution via SSA methods

The approximation theorems and methods discussed in Section 3 are applicable to the tradeoff resolution problem.

Example 5. Consider the network in Fig. 10. Theorem 1 is applicable to this network: We have (a) Y^1 positively influences X given D and Y^2, (b) Y^2 negatively influences X given D and Y^1, (c) X and A are independent given D, Y^1, and Y^2, (d) $[D, A, Y^1, Y^2]$ is an ancestral ordering, and (e) Y^1 is not a descendant of Y^2 and vice versa in the network. Therefore, we can compute the bounds of $F(x|d)$ when we abstract A with the dominance policy. Specifically, we obtain lower (upper) bounds of $F(x|d)$ by weakening (strengthening) $F(y^1|a,y^2)$ and
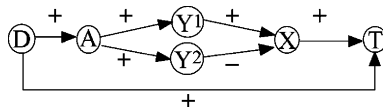

Fig. 10. We may use qualitative relationships for bounding probability distributions.

strengthening (weakening) $F(y^2|a, y^1)$ with respect to A when we abstract A. In addition, We can obtain lower (upper) bounds of $F(x|d)$ by strengthening (weakening) $F(x|y^1)$ with respect to Y^1 when we abstract Y^1, given that (a) $[D, Y^1, X]$ is an ancestral ordering and (b) X is the only child node of Y^1. Analogously, we may abstract Y^2 in computing bounds of $F(x|d)$ to further reduce computation time.

In addition, Corollary 1 guarantees that bounds computed by the ISSA algorithm tighten as we refine the state space of the abstracted nodes. Therefore, we are more likely to resolve qualitative tradeoffs as we carry out more iterations of the ISSA algorithm. The computation can terminate whenever we determine the qualitative relationship of interest.

As we discussed in Section 4.3.3, if the relationship between D and T is ambiguous, there must be a node X such that X is marked with "?" when we propagate the sign from D toward T. As an alternative to incremental marginalization, we may apply any approximate evaluation algorithm for Bayesian networks at step 4 of ITOR, if the approximation algorithm can return bounds of conditional probability distributions. We use ISSA with the dominance policy as such an alternative in the following discussion.

Using the ISSA algorithm with dominance policy can save computation time for the tradeoff resolution task. As we mentioned, the ISSA algorithm may find the correct qualitative relationship by the time it needs to exactly evaluate $F(x|d)$ using the conditions specified in (11)–(13). In addition, the ISSA algorithm may compute the bounds of $F(x|d)$ by evaluating a portion of the given Bayesian network. For instance, T in Fig. 10 is barren and can be ignored for the computation of $F(x|d)$.

Notice that we should not terminate ITOR when ISSA returns a "?" for the relationship of D and X at step 4. When this occurs, we need to continue ITOR as usual. It is possible that we find a decisive relationship between D and T even when some nodes in the network have an ambiguous relationship with D. For instance, using ISSA, we might find that D positively influences T even if D neither negatively nor positively influences X in the network shown in Fig. 11. Therefore, ITOR should run until either a decisive relationship between D and T has been found at step 2 or an ambiguous relationship between D and T is confirmed by ISSA at step 4.

Some results obtained by ISSA may be reused in later iterations of ITOR. Take the network in Fig. 10 as an example. Assume that we find that D negatively influences X by exactly computing the values of $F(x|d)$, and that we are
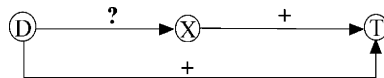

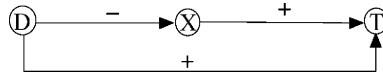Fig. 11. ITOR should not terminate when ISSA reports $S^?(D, X)$.

Fig. 12. A possible reduced version of the network in Fig. 10.

about to use the ISSA algorithm to determine the qualitative relationship between D and T. In this case, if we do not reuse previous results, we may abstract A, Y^1, Y^2, and X in computing the bounds of $F(t|d)$ at step alg1:step4. However, given that $CI(T, \{D, X\}, \{A, Y^1, Y^2\})$ and that we have computed the exact values of $F(x|d)$, the network has been reduced to the one shown in the following figure. Therefore, we could save computation time by running ISSA over the network in Fig. 12, and should not run ISSA over the network in Fig. 10 from scratch.

Whether we reuse the information about the bounds that are obtained from running ISSA is a design issue. For instance, assume that we find that D negatively influences X in Fig. 10 by applying (11). Given this result, the qualitative relationship between D and T is still ambiguous since there are still two competing influential paths from D to T as indicated in Fig. 12. Upon locating this ambiguity, ITOR uses ISSA to resolve the ambiguity, and the issue is whether we reuse the quantitative information we have about $F(x|d)$ that is obtained from the previous execution of ISSA. Given that $CI(T, \{D, X\}, \{A, Y^1, Y^2\})$ and that D decisively influences X in the network, a heuristic is that we use the knowledge about $F(x|d)$ obtained in the previous run of ISSA first. To do so, we set the conditional probability of X given D according to bounds of $F(x|d)$ in the network shown in Fig. 12, and use this approximate network for resolving the ambiguity. It is possible that we find a decisive qualitative relationship using the bounds in the network. If this opportunistic approach does not resolve the qualitative ambiguity between D and T, we then use ISSA to evaluate a network that includes A, Y^1 and Y^2. Another alternative is to directly compute bounds of $F(t|d)$ using the original network. It is possible that we can resolve the ambiguity when A, Y^1, Y^2, and X have very small numbers of states in ISSA. The optimal choice for this design issue varies from network to network, depending on their underlying probability distributions.

5. Application: path planning in stochastic networks

Path planning—the problem of finding an optimal route in a transportation network—has been widely studied, resulting in a variety of well-known algorithms [1,10,11,13]. We denote a transportation network with a weighted graph $G = (V, E, w)$, where the set of vertices V represents locations, the set of edges E represents roads that connect locations, and the cost function w encodes the

total costs of traversing an edge. Following the traditional terminology, we call an edge in the transportation network a *link*, and the cost of traversing across an edge the *link travel time*.

Many path-planning algorithms assume that the link traversal costs are deterministic, that is, w is a function associating edges with travel times. This is typical for algorithms seeking plans that will travel the least surface distance. In actual transportation networks, however, link travel times are rarely certain, and thus should be treated as random variables. Moreover, in such *stochastic networks*, the probability distributions over link travel times also depend on the time one traverses the link. Hall [20] shows that such dependent relationships among link travel times violate the optimality principle underlying standard shortest-path algorithms.

Kaufman and Smith [26] study time-dependent networks, in which w is a deterministic function of both the link being considered and the time the traveler enters the link. They show that algorithms such as Dijkstra's remain applicable when the link travel times satisfy the following *consistency* property. Let $t_1 \leqslant t_2$ be alternative departure times that one may enter a link e, the consistency property requires the following inequality holds for the arrival time at the other end of the link for all links:

$$t_1 + w(e, t_1) \leqslant t_2 + w(e, t_2).$$

Wellman et al. [52] generalize this approach, and consider networks in which costs are time-dependent random variables. They show that standard search algorithms, such as A^*, can find the shortest path when link travel times have the *stochastic consistency* property. This property requires that the following hold for all links e, and all times $t_1 \geqslant t_2$:

$$F(t_1 + w(e, t_1)) \text{FSD} F(t_2 + w(e, t_2)).$$

In other words, the probability of arriving by any particular time cannot increase by leaving later.

Boyan and Littman [3] present a general model of time-dependent Markov Decision Processes, which can solve stochastic path-planning problems with piecewise linear time dependencies. Chabini and Lan [4] report and analyze an application of the A^* algorithm to stochastic transportation networks, which replies on a first-in-first-out property that is similar to the stochastic consistent property.

In this section, we study the problem of computing travel times in stochastic transportation networks. In particular, we consider situations in which computation time is insufficient for obtaining exact probability distributions over travel times of interest. To this end, we propose a path-planning algorithm that applies results, including Theorem 1 and Corollary 2, reported in Section 3 for computing gradually tightening bounds on probability distributions over travel times. We show that our anytime algorithm may find the actual shortest path,

and that we can compute the maximum deviation of the distribution over the travel time on the path selected by our algorithm from the distribution over the travel time on the actual shortest path. Moreover, we exploit special properties of distributions over travel times to design specific state-space abstraction methods for tighter bounds of travel times, and extend applicability of the proposed algorithm to situations in which the FSD relationships among travel times do not hold perfectly.

5.1. Approximate travel times

As in typical path-planning problems, we assume that we have specifications of the link travel times, which in our case comprise distributions given each possible time that one may enter the links. Let $(1, 2, \ldots, n)$ index locations on a path, and $\Pr(t_i)$ be the probability of arriving at location i at time $T_i = t_i$. Given a departure time from location 1, say $\bar{t}_1$, the distribution of the arrival time at location 2 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 location $j + 1$ based on the arrival time at location j: [3]

$$F(t_{j+1}) = \sum_{t_j} F(t_{j+1}|t_j) \Pr(t_j). \tag{14}$$

Since the uncertainty about arrival time increases with the traveling distance, the number of possible values of T_j grows with j, and makes computing of $\Pr(T_j)$ for all candidate paths with (14) very time consuming. One way to limit the computational cost is to restrict the number of states of T_j [32]. We can achieve this by aggregating the states of T_j, say into β states, before we compute the distribution of T_{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 $\widehat{F}(t_j)$. We apply (4) and (7) discussed in Sections 2.3 and 3.2 to obtain the following formula.

$$\widehat{\widehat{\Pr}}(\hat{t}_j) = \sum_{t_j \in \hat{t}_j} \widehat{\Pr}(t_j), \tag{15}$$

$$\widehat{F}(t_{j+1}|\hat{t}_j) = \max_{t_j \in \hat{t}_j} F(t_{j+1}|t_j). \tag{16}$$

We can let $\widehat{\Pr}(t_2) \equiv \Pr(t_2)$ without any loss, although we have obtained the exact distribution for T_2 already. Hence we can apply (15) and (16) to T_j for all

---

[3] For simplicity, we assume that one would not stop at intermediate locations. As a result, there is no need to distinguish arrival and departure times; we use arrival time for both. Moreover, since all expressions are conditional on the original departure time, $\bar{t}_1$, we express this implicitly henceforth.

$j \geqslant 2$. In (15), the double "hats" signify that the approximate probabilities are determined based on other approximated probabilities. For simplicity, a 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}. [4]

Theorem 5. *We obtain an optimistic approximation of $F(t_{j+1})$, i.e., $F(t_{j+1}) \text{FSD} \widehat{F}(t_{j+1})$, when we use (15) and (16) in computing $\widehat{F}(t_{j+1})$.*

Proof

$$\widehat{F}(t_{j+1}) = \sum_{\hat{t}_j} \widehat{F}(t_{j+1}|\hat{t}_j) \widehat{\widehat{\Pr}}(\hat{t}_j)$$

$$= \sum_{\hat{t}_j} \left[\left[\max_{t_j \in \hat{t}_j} F(t_{j+1}|t_j) \right] \sum_{t_j \in \hat{t}_j} \widehat{\Pr}(t_j) \right]$$

$$\geqslant \sum_{t_j} F(t_{j+1}|t_j) \widehat{\Pr}(t_j) = \sum_{t_j} F(t_{j+1}|t_j) \, \mathrm{d}\widehat{F}(t_j)$$

$$\geqslant \sum_{t_j} F(t_{j+1}|t_j) \, \mathrm{d}F(t_j) = F(t_{j+1}). \qquad (17)$$

Since every t_j is covered by exactly one $\hat{t}_j$ when we aggregate states, $\widehat{\Pr}(t_j)$ will occur exactly once after we completely expand the summations in the second equality. Also each component $\widehat{\Pr}(t_j)$ of $\widehat{\widehat{\Pr}}(\hat{t}_j)$ 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 (17) after recollecting all terms. Now, as we have assumed that T_j positively influences T_{j+1}, we have $F(t_{j+1}|t_j) \text{FSD} F(t_{j+1}|t'_j)$ if $t_j \geqslant t'_j$. In other words, $F(t_{j+1}|t_j)$ is a nonincreasing function of t_j. Also recall that $\widehat{F}(t_2)$ is actually an exact distribution, so it is trivially true that $F(t_2) \text{FSD} \widehat{F}(t_2)$. Using proof by induction, we can assume that $F(t_j) \text{FSD} \widehat{F}(t_j)$, and go on to show that $F(t_{j+1}) \text{FSD} \widehat{F}(t_{j+1})$. Now given that $F(t_j) \text{FSD} \widehat{F}(t_j)$ and that $F(t_{j+1}|t_j)$ is a nonincreasing function of t_j, we can apply (1) to obtain the second inequality, and establish $F(t_{j+1}) \text{FSD} \widehat{F}(t_{j+1})$. $\quad \square$

Using an analogous procedure, we can show by induction that replacing (16) with the following assignment:

---

[4] Theorem 5 is a special case of Theorem 1. We repeat the theorem and proof in a context specifically for the path-planning problem in stochastic transportation networks.

$$\widehat{F}(t_{j+1}|\hat{t}_j) = \min_{t_j \in \hat{t}_j} F(t_{j+1}|t_j)$$

yields *pessimistic* approximations of the true distributions, that is, $\widehat{F}(t_{j+1})$ FSD $F(t_{j+1})$.

Recall that an upper bound, as defined in Section 2.2, is an optimistic estimation of the actual arrival time because probability mass is shifted toward earlier arrival times. Similarly, a lower bound provides a pessimistic approximation. To avoid possible confusion, we use optimistic and pessimistic bounds of travel times in places of upper and lower bounds in the path-planning context. In summary, applying different assignments of the dominance policy we can choose to compute the optimistic or pessimistic bounds of the exact travel times by the state-space abstraction methods.

5.2. Path planning based on optimistic approximations

We can apply our approximate arrival-time calculations to find shortest paths from a given origin O to a given destination D. The algorithm attempts to find the best solution, without necessarily computing exact distributions over the travel times in comparing paths.

5.2.1. Algorithm

Our algorithm is a form of priority-first search, maintaining a priority queue (*PQ*) of partially expanded paths from O toward D. We prioritize paths by bounds on their *projected arrival times* at D. A projected arrival time at D of a path $O \to \cdots \to X$ is the random variable $T_X + h(X)$, where $h(X)$ is a lower bound on the time required to travel from X to D. We can estimate $h(X)$ from standard information such as straight-line distance and maximal travel rate.

In addition, we use the *optimistic envelope* of the projected arrival times of all paths in *PQ* to determine whether the algorithm should terminate. Let $OB_i(t)$ be the optimistic bound on the distribution over the projected arrival time at D of path P_i in *PQ*. The optimistic envelope $OE(t)$ of the projected arrival times of all paths in *PQ* is defined as:

$$OE(t) = \max_{P_i \in PQ} OB_i(t).$$

Maintaining the optimistic envelope is easy. When new paths are added to *PQ*, we compare the current envelope with the lower bounds of the newly added paths, and update the optimistic envelope when necessary. When a path P_i is being removed, we only need to update $OE(t)$ for those ts such that $OE(t) = OB_i(t)$ before the removal of P_i. We describe the algorithm in the following elements:

Algorithm 3. Path planning with bounds of travel times

Input: a stochastic network, the origin O with departure time, the destination D, and the maximal number of states β used to specify arrival times at intermediate locations after state-space aggregation.

Output: a selected path from O to D along with the maximum deviation of the distribution over its arrival time from the distribution over the arrival time of the actual shortest path.

Data structures:
- *PQ*, a priority queue: Items associate a partial path with bounds on its path cost, with path costs prioritized by the method explained below.
- Optimistic envelope: The optimistic bound of all paths in *PQ* as is explained.
- Solution list (SL): Undominated candidate paths that connect O and D.

Procedure:
1. Add $O \rightarrow O$ to *PQ* with cost 0.
2. If *PQ* is empty and the solution list is empty, report an error. If *PQ* is empty and the solution list is not empty, terminate and return the best candidate in the solution list.
3. Remove the highest-priority path $P = O \rightarrow \cdots \rightarrow X$ from *PQ*, and update the optimistic envelope. If $X \neq D$, go to the next step. Otherwise, add P into the solution list unless there is a candidate solution with dominant priority in the solution list. Go to step 5.
4. Construct new paths for each possible next link the traveler could take from X, and insert the resulting items into *PQ*. Update the optimistic envelope.
5. If the optimistic envelope dominates the pessimistic bound of a known path in the solution list, return the best candidate in the solution list. Otherwise, go to step 3.

Notice that one path does not necessarily dominate the other, or vice versa, even if we have exact distributions over arrival times. This could happen when the curves of the CDFs of their arrival times intersect.

We employ heuristics for prioritizing paths in *PQ* and the solution list. We can choose an arbitrary measure that is consistent with the dominance ordering. For instance, we may prioritize paths based on expected values of their optimistic bounds.

The procedure terminates when the optimistic envelope of projected arrival times of paths in *PQ* dominates the upper bound of a known path in the solution list. When this condition does occur, the known path must be better than all the paths in *PQ* since the projected arrival times are the earliest arrival time achievable by traveling on all paths that are not yet fully expanded. We show this proposition next.

5.2.2. Properties of the algorithm

We show that, when the procedure terminates, the solution list must include the actual shortest path. Then, based on this result, we discuss how to compute maximum deviation of the distribution over the arrival time of the recommended path from the distribution over the arrival time of the actual shortest path.

Theorem 6. *The solution list must include the optimal path when the algorithm terminates.*

Proof. We show this by contradiction. First, notice that all partial paths departing from O must reside in the priority queue or in the solution list. Let Q be the known path found at step 5, $AQ(t)$ denote the actual distribution over the arrival time at D of Q, and $UQ(t)$ the pessimistic bound of $AQ(t)$. Denote the optimistic envelope of all paths in PQ by $OE(t)$. Fig. 13 shows a generic picture of the scenario. If the actual shortest path R is not in the solution list and is still in PQ, then the actual distribution of R must fall into the lower-right hand side to the curve of $OE(t)$. This implies that the actual arrival time at D of R is a pessimistic bound of that of Q. However, if the actual arrival time of R is a pessimistic bound of that of Q, Q will be better than R, and R cannot be a shortest path. Therefore, we have shown that, if R is a shortest path, it must be in the solution list when the procedure terminates.   □

As a result, when the procedure terminates, we can compute the maximum deviation of the distribution over the arrival time of the recommended path, RP, from the distribution over the arrival time of the actual shortest path. Let $OBS_i(t)$ and $PBS_i(t)$ be the optimistic and pessimistic bounds on path P_i in the solution list, respectively. Let $OESL(t)$ and $PESL(t)$ be the lower and upper envelopes of all paths in the solution list, respectively. Namely, we define

$$OESL(t) = \max_{P_i \in SL} OBS_i(t) \quad \text{and} \quad PESL(t) = \min_{P_i \in SL} PBS_i(t).$$

Since the exact distribution over the arrival time of the actual shortest path must fall in between $OESL(t)$ and $PESL(t)$, these envelopes can be used to
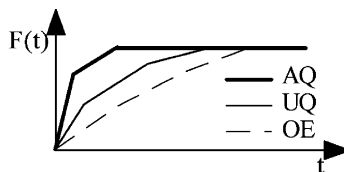

Fig. 13. The actual shortest path must be in the solution list when Algorithm 3 terminates.

compute the maximum deviation. For instance, assume that we choose to use the expected arrival time at D as the measure for the deviation. We can compute the expected arrival time calculated with $\mathrm{OESL}(t)$, and report the difference between this expected arrival time and that of RP as the maximum deviation.

Corollary 3. *The distribution of arrival time of the actual shortest path is bounded by* $\mathrm{OESL}(t)$ *and* $\mathrm{PESL}(t)$.

5.2.3. Approximation strategies

The superstate selection problem for computing the fastest path in Algorithm 3 is not the same as that for evaluating Bayesian networks in Algorithm 1. 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.

In contrast, as we gradually expand partial paths $L_1 \rightarrow L_2 \rightarrow \cdots \rightarrow L_n$ to the next intermediate location L_{n+1} in Algorithm 3, 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.

We consider an approximation strategy that considers a specific condition in transportation networks in this section, although the strategies we reported in [33] can be applicable for the problems of bounding travel times too. Let t_{jk} be the kth state of T_j. We assume 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,

$$F(t_{j+1}|t_{jk}) - F(t_{j+1}|t_{j(k+1)}) \leqslant \varepsilon \quad \text{for all } k \text{ and } t_{j+1}. \tag{18}$$

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: $t_{j1}, t_{j2}, \ldots, t_{jm}$ and that we aggregate these states into n groups: $S_1 = \{t_{j1}, \ldots, t_{jb_1}\}$, $S_2 = \{t_{j(b_1+1)}, \ldots, t_{jb_2}\}$, $\ldots$, and $S_n = \{t_{j(b_{n-1}+1)}, \ldots, t_{jm}\}$. Let $g'(t_{jk})$ denote the group that contains t_{jk}. To minimize the errors of approximations, referring to (17) and its derivation, we would like to minimize the following difference.

$$\delta = \sum_{\hat{t}_j} \widehat{F}(t_{j+1}|\hat{t}_j) \widehat{\widehat{\mathrm{Pr}}}(\hat{t}_j) - \sum_{t_j} F(t_{j+1}|t_j) \widehat{\mathrm{Pr}}(t_j)$$

$$= \sum_{\hat{t}_j} \left[\left[\max_{t_j \in \hat{t}_j} F(t_{j+1}|t_j) \right] \sum_{t_j \in \hat{t}_j} \widehat{\mathrm{Pr}}(t_j) \right] - \sum_{t_j} F(t_{j+1}|t_j) \widehat{\mathrm{Pr}}(t_j)$$

$$= \sum_{t_j} \left[\left[F(t_{j+1}| \min_{t_k \in g'(t_j)} t_k) - F(t_{j+1}|t_j) \right] \widehat{\mathrm{Pr}}(t_j) \right]$$

$$\leqslant (0 + \varepsilon \widehat{\mathrm{Pr}}(t_{j2}) + \cdots + (b_1 - 1)\varepsilon \widehat{\mathrm{Pr}}(t_{jb_1})) + (0 + \varepsilon \widehat{\mathrm{Pr}}(t_{j(b_1+2)}) + \cdots$$
$$+ (b_2 - b_1 - 1)\varepsilon \widehat{\mathrm{Pr}}(t_{jb_2})) + \cdots + (0 + \varepsilon \widehat{\mathrm{Pr}}(t_{j(b_{n-1}+2)}) + \cdots$$
$$+ (b_n - b_{n-1} - 1)\varepsilon \widehat{\mathrm{Pr}}(t_{jm})). \tag{19}$$

The first two equalities follow directly from (17). The third equality also follows from (17), 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 (18) will give us the inequality in (19), 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 (19) gives us an upper bound of the difference δ. Therefore, one way to minimize δ is to minimize the upper bound. Letting $b_0 = 0$ and $b_n = m$, we ignore the common factor ε, and rewrite the bound in a more compact form. As a result, a heuristic for determining how we aggregate the states of T_j into n group is to minimize the following quantity.

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

Notice that the contribution of each t_{jk} is $\widehat{\mathrm{Pr}}(t_{jk})$ 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 2.3, leads us to use $\sum_{t_j \in t'_j} \widehat{\mathrm{Pr}}(t_j)$ as the guidance for superstate selection.

5.3. Anytime extensions by incremental state-space refinement

We may apply the algorithm in the context in which the deadlines for returning recommended paths are unknown at the time the algorithm is implemented. We may apply Algorithm 3 in an iterative manner, setting the maximal number of states a random variable can have, β, to 2 initially. If we can finish the procedure with $\beta = 2$, we will have a candidate solution at our disposal. We may then set β to a larger quantity, say 3, and run the algorithm again to obtain another candidate solution. We can repeat the algorithm until the

deadlines are reached, and return the candidate solution found in the latest completed run of the algorithm. If the deadlines are long enough, we may choose to terminate the computation if the candidate solution is good enough. This can happen, for example, when the maximum deviation of the candidate solution from the actual shortest path is considered small enough.

Several factors influence the variation of the computation time of the algorithm from iterations to iterations. On one hand, increasing β tends to increase the costs for computing travel times of individual paths; on the other hand, increasing β may also reduce the number of candidate paths because we gather more information about path costs in each iteration. Also, Theorem 6 guarantees that the solution list must contain the actual shortest path when the procedure terminates at step 5. Therefore, in the subsequent run of the procedure with a larger β, we need not consider any path that cannot be expanded into paths in the solution list in a previous run of the algorithm. Moreover, Corollary 2 provides that, as we use more states in the computation when we increase β, the bounds will become tighter than their counterparts in previous runs of the algorithm. Together, these factors make nonoptimal paths less likely to survive. In other words, the number of alternative paths that may be considered as candidate solutions should decrease as we increase β, thereby helping to contain the growth of the computational costs.

This extended algorithm can demonstrate the desirable *anytime* property as discussed in [2]. Our algorithm can be interrupted and return a solution anytime after we finish the first run of the algorithm. Also, the quality of the recommended path will improve monotonically as we finish more iterations, if we define the quality as the maximum deviation between $\mathrm{OESL}(t)$ and $\mathrm{PESL}(t)$. However, the actual deviation between the recommended and the shortest path is *not* guaranteed to dwindle. There are always chances that the algorithm recommends the actual shortest path in early iterations and suboptimal paths in subsequent iterations.

5.4. Weak stochastic dominance

When the relationships of positive or negative influence do not hold, the theorems reported in Section 3 and methods reported in this section become inapplicable. The second inequality in (17) would not hold because we lose the condition that $F(t_{j+1}|t_j)$ is a nonincreasing function of t_j. 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 Fig. 14, where t_{jk} represents the kth state of T_j. Assume that $t_{j1} < t_{j2} < t_{j3} < t_{j4}$. An interpretation of the curves in this figure is that leaving the origin early almost guarantees earlier arrival at the destination. The crossing curves show that T_j does not positively influence T_{j+1}.
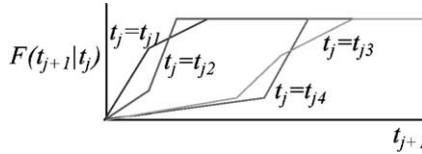
Fig. 14. T_j *weakly* positively influences T_{j+1}.

Nevertheless, the trend of the curves seems to support that T_j *weakly* positively influences [5] 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 7. Assume that a random variable X has m states: $x_1, x_2, \ldots, x_m$ and that these states form $n \geqslant 1$ groups: $G_1 = \{x_1, \ldots, x_{b_1}\}$, $G_2 = \{x_{b_1+1}, \ldots, x_{b_2}\}, \ldots$, and $G_n = \{x_{b_{n-1}+1}, \ldots, x_m\}$. A node X *weakly positively influences* its child Y if and only if, $F(y|x_i, \boldsymbol{px}(Y)) \mathrm{FSD} F(y|x_k, \boldsymbol{px}(Y))$, 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 T_j weakly positively influences T_{j+1} in the stochastic transportation network, methods reported in this section remain applicable after we approximate the probability distributions of the link travel times. Let $g(t_{jk})$ be the state group that contains a particular value t_{jk} of T_j. To compute the bounds, we approximate the CDF $F(t_{j+1}|t_{jk})$ for all possible value t_{jk} of T_j by the following formula before applying the SSA methods.

$$\widehat{F}(t_{j+1}|t_{jk}) = \max_{t_{jl} \in g(t_{jk})} F(t_{j+1}|t_{jl}). \tag{20}$$

Consider the example shown in Fig. 14, where we have $G_1 = \{t_{j1}, t_{j2}\}$ and $G_2 = \{t_{j3}, t_{j4}\}$. After we apply (20) 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 Fig. 15.

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

$$\widehat{F}(t_{j+1}) = \sum_{t_j} \left(\max_{t_j \in g(t_j)} F(t_{j+1}|t_j) \right) \Pr(t_j) \geqslant \sum_{t_j} F(t_{j+1}|t_j) \Pr(t_j) = F(t_{j+1}).$$

---

[5] Our notion of weakly qualitative influences are extensions of the *generalized qualitative influences* proposed by the authors in [31]. The basic concept is not related to Renooij and van der Gaag's notion of weakly qualitative influences that are defined based on the magnitude of probability-mass fluctuation of a random variable caused by changing the state of another variable [43].
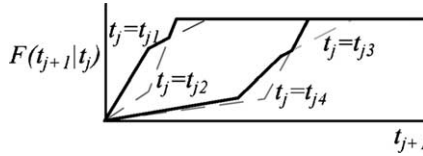
Fig. 15. Using weakly positive influence for bounding distributions.

After we apply (20) 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 7 reverses, we say that X *weakly negatively influences* Y. Under such circumstances, we replace the max operator in (20) by the min operator to obtain bounds of probability distributions analogously.

6. Conclusion

Although the results of probabilistic inference can be quite sensitive to tiny changes in the probability values in Bayesian networks [5], there are situations where inaccuracy or nonspecificity in probability values will not affect quality of the reached conclusions [42]. When the latter statement holds, we may be able to do without full numeric precision, thus saving computational effort or other costs of purely quantitative inference. The approximation methods presented here take advantage of decisive qualitative relationships between random variables to find useful solutions without necessarily relying on all the numbers, even when they are available.

When variables positively or negatively influence others, one can ensure bounds of probability distributions through controlled adjustment of the original probability parameters. One embodiment of this idea is our use of a specialized aggregation policy for abstracting states of variables in ways that preserve bounding relationships. We show when the approximation techniques work and how the acquired approximations behave over computation time. One application of our state-space methods is an incremental tradeoff resolution technique that determines qualitative relationships of interest using appropriate bounds of involved probability distributions. A second application finds fastest paths in stochastic transportation networks using gradually tightening upper and lower bounds of travel times. This algorithm can respond to queries of uncertain deadlines, and is able to compute the worst-case deviation in travel times between the proposed travel plan and the unknown fastest plan.

Acknowledgements

Appendix A

A.1. Proof of Theorem 1

Theorem 1. *Assume that*

1. *For all i, $S^{\sigma^i}(Y^i, Z \| e, \boldsymbol{SB}(Y^i))$, where σ^i is either $+$, $-$, or 0.*
2. *$CI(Z, \{\boldsymbol{E}, \boldsymbol{Y}\}, A)$.*
3. *$\boldsymbol{E}$, A, and $\boldsymbol{Y}$ appear in order in an ancestral ordering of nodes of the given Bayesian network.*
4. *For all i, Y^i is not a descendant of nodes in $\boldsymbol{SB}(Y^i)$.*

When $\sigma^i = -$, we obtain, respectively, a lower bound and an upper bound of $F(z|e)$ by weakening and strengthening $F(y^i|a, \boldsymbol{px}(Y^i))$ with respect to A. When $\sigma^i = +$, we obtain, respectively, an upper bound and a lower bound of $F(z|e)$ by weakening and strengthening $F(y^i|a, \boldsymbol{px}(Y^i))$ with respect to A. When $\sigma^i = 0$, neither strengthening nor weakening $F(y^i|a, \boldsymbol{px}(Y^i))$ with respect to A will affect $F(z|e)$.

Proof. We show the case that we compute a lower bound of $F(z|e)$ by weakening $F(y^i|a, \boldsymbol{px}(Y^i))$ when $S^-(Y^i, Z \| e, \boldsymbol{SB}(Y^i))$. Other cases can be shown analogously.

Assume that we construct a Bayesian network, XBN, by copying all information that specifies the OBN to the XBN and then weakening the conditional probability distribution of Y^i given its parent nodes with respect to A. We can show that the $F(z|e)$ computed from this XBN is a lower bound of the $F(z|e)$ computed from the OBN. We denote CDF functions in the XBN by $\widetilde{F}(\cdot)$, and expand $\widetilde{F}(z|e)$ as follows. (Notice that this XBN is actually the EABN, introduced in Section 3.2, of the ABN in which A is the abstracted node.)

$$
\widetilde{F}(z|e) = \int_A \mathrm{d}\widetilde{F}(a|e) \int_{\boldsymbol{SB}(Y^i)} \mathrm{d}\widetilde{F}(\boldsymbol{sb}(Y^i)|e, a)
$$
$$
\times \int_{Y^i} \widetilde{F}(z|a, \boldsymbol{y}, e) \, \mathrm{d}\widetilde{F}(y^i|e, a, \boldsymbol{sb}(Y^i)). \tag{A.1}
$$

We need to show the following properties of the factors in (A.1).

1. $\widetilde{F}(a|e) = F(a|e)$

Given the third condition, Y^i is a barren node for the task of computing $\widetilde{F}(a|e)$ and $F(a|e)$. Therefore, removing Y^i and its descendants from XBN and OBN will not affect $\widetilde{F}(a|e)$ and $F(a|e)$. After removing Y^i and its descendants from the networks, XBN and OBN become exactly the same, so we have $\widetilde{F}(a|e) = F(a|e)$.

2. $\widetilde{F}(\boldsymbol{sb}(Y^i)|a, e) = F(\boldsymbol{sb}(Y^i)|a, e)$

Given the third and the fourth conditions, Y^i is barren for the task of computing $\widetilde{F}(\boldsymbol{sb}(Y^i)|a, e)$, and $F(\boldsymbol{sb}(Y^i)|a, e)$. Therefore, using an analogous reasoning we use to show that $\widetilde{F}(a|e) = F(a|e)$ in this proof, we have this equality.

3. $\widetilde{F}(\boldsymbol{px}(Y^i)|e, a, \boldsymbol{sb}(Y^i)) = F(\boldsymbol{px}(Y^i)|e, a, \boldsymbol{sb}(Y^i))$

Given the third condition, the weakening of $F(y^i|a, \boldsymbol{px}(Y^i))$ cannot affect the CDF $F(\boldsymbol{px}(Y^i)|e, a, \boldsymbol{sb}(Y^i))$ by Theorem 3 in [33], so we have the equality.

4. $\widetilde{F}(z|a, \boldsymbol{y}, e) = F(z|a, \boldsymbol{y}, e)$

Since $CI(Z, \boldsymbol{E} \cup \boldsymbol{Y}, A)$, we have $\widetilde{F}(z|a, \boldsymbol{y}, e) = \widetilde{F}(z|\boldsymbol{y}, e)$ and $F(z|a, \boldsymbol{y}, e) = F(z|\boldsymbol{y}, e)$. Also because of $CI(Z, \boldsymbol{E} \cup \boldsymbol{Y}, A)$, the abstraction of A will not affect the condition probability $F(z|\boldsymbol{y}, e)$ by Theorem 1 in [33]. Namely, we have $\widetilde{F}(z|\boldsymbol{y}, e) = F(z|\boldsymbol{y}, e)$, and that $\widetilde{F}(z|a, \boldsymbol{y}, e) = F(z|a, \boldsymbol{y}, e)$ follows.

5. $F(y^i|e, a, \boldsymbol{sb}(Y^i))\text{FSD}\,\widetilde{F}(y^i|e, a, \boldsymbol{sb}(Y^i))$

Recall that when we weaken a CDF, we make its values larger, so $\widetilde{F}(y^i|a, \boldsymbol{px}(Y^i)) \geqslant F(y^i|a, \boldsymbol{px}(Y^i))$. Using this inequality and the third fact $\widetilde{F}(\boldsymbol{px}(Y^i)|e, a, \boldsymbol{sb}(Y^i)) = F(\boldsymbol{px}(Y^i)|e, a, \boldsymbol{sb}(Y^i))$, we have the following.

$$
\begin{aligned}
\widetilde{F}(y^i|e, a, \boldsymbol{sb}(Y^i)) &= \int_{\boldsymbol{PX}(Y^i)} \widetilde{F}(y^i|e, a, \boldsymbol{px}(Y^i))\,\mathrm{d}\widetilde{F}(\boldsymbol{px}(Y^i)|e, a, \boldsymbol{sb}(Y^i)) \\
&= \int_{\boldsymbol{PX}(Y^i)} \widetilde{F}(y^i|a, \boldsymbol{px}(Y^i))\,\mathrm{d}\widetilde{F}(\boldsymbol{px}(Y^i)|e, a, \boldsymbol{sb}(Y^i)) \\
&\geqslant \int_{\boldsymbol{PX}(Y^i)} F(y^i|a, \boldsymbol{px}(Y^i))\,\mathrm{d}F(\boldsymbol{px}(Y^i)|e, a, \boldsymbol{sb}(Y^i)) \\
&= F(y^i|e, a, \boldsymbol{sb}(Y^i)).
\end{aligned}
$$

6. $\int_{Y^i} F(z|a, \boldsymbol{y}, e)\,\mathrm{d}\widetilde{F}(y^i|e, a, \boldsymbol{sb}(Y^i)) \leqslant \int_{Y^i} F(z|a, \boldsymbol{y}, e)\,\mathrm{d}F(y^i|e, a, \boldsymbol{sb}(Y^i))$

$S^-(Y^i, Z\|e, \boldsymbol{SB}(Y^i))$ implies that $F(z|\boldsymbol{y}, e)$ is an increasing function in y^i, so is $F(z|a, \boldsymbol{y}, e)$ because $F(z|\boldsymbol{y}, e) = F(z|a, \boldsymbol{y}, e)$ due to $CI(Z, \boldsymbol{E} \cup \boldsymbol{Y}, A)$. Given these and the fifth fact, the inequality (1) implies this inequality.

Given these six facts, we show that $\widetilde{F}(z|e)$ is a lower bound of $F(z|e)$ by continuing expansion of (A.1) as follows. The first equality results from the

application of the first, the second, and the fourth facts. The inequality is a direct result of the sixth result.

$$\widetilde{F}(z|e) = \int_A \mathrm{d}F(a|e) \int_{\boldsymbol{SB}(Y^i)} \mathrm{d}F(\boldsymbol{sb}(Y^i)|e,a) \int_{Y^i} F(z|a,\boldsymbol{y},\boldsymbol{e}) \,\mathrm{d}\widetilde{F}(y^i|e,a,\boldsymbol{sb}(Y^i))$$

$$\leqslant \int_A \mathrm{d}F(a|e) \int_{\boldsymbol{SB}(Y^i)} \mathrm{d}F(\boldsymbol{sb}(Y^i)|e,a) \int_{Y^i} F(z|a,\boldsymbol{y},\boldsymbol{e}) \,\mathrm{d}F(y^i|e,a,\boldsymbol{sb}(Y^i))$$

$$= F(z|e). \qquad \square$$

A.2. Proof for Theorem 2

Theorem 2. *In addition to conditions* 3 *and* 4 *in Theorem* 1*, assume that* $Z \in \boldsymbol{Y}$*. We obtain, respectively, a lower and an upper bound of* $F(z|e)$ *by strengthening and weakening* $F(z|a, \boldsymbol{px}(Z))$ *with respect to A.*

Proof. We show the case for computing lower bounds of $F(z|e)$. The case for computing upper bounds of $F(z|e)$ can be done analogously. This proof is similar to that for Theorem 1. First, we can expand $F(z|e)$ as follows:

$$\widetilde{F}(z|e) = \int_A \mathrm{d}\widetilde{F}(a|e) \int_{\boldsymbol{SB}(Z)} \widetilde{F}(z|a,\boldsymbol{sb}(Z),\boldsymbol{e}) \,\mathrm{d}\widetilde{F}(\boldsymbol{sb}(Z)|e,a). \qquad (A.2)$$

Applying analogous reasons used in the proof for Theorem 1, we can show the following results.

1. $\widetilde{F}(a|e) = F(a|e)$.
2. $\widetilde{F}(\boldsymbol{sb}(Z)|a,\boldsymbol{e}) = F(\boldsymbol{sb}(Z)|a,\boldsymbol{e})$.
3. $\widetilde{F}(z|e,a,\boldsymbol{sb}(Z)) \leqslant F(z|e,a,\boldsymbol{sb}(Z))$ when we strengthen $F(z|a,\boldsymbol{px}(Z))$ with respect to A.

Given these, we continue the derivation of (A.2) to show that $\widetilde{F}(z|e)$ is a lower bound of $F(z|e)$ as follows.

$$\widetilde{F}(z|e) = \int_A \mathrm{d}F(a|e) \int_{\boldsymbol{SB}(Z)} \widetilde{F}(z|a,\boldsymbol{sb}(Z),\boldsymbol{e}) \,\mathrm{d}F(\boldsymbol{sb}(Z)|e,a)$$

$$\leqslant \int_A \mathrm{d}F(a|e) \int_{\boldsymbol{SB}(Z)} F(z|a,\boldsymbol{sb}(Z),\boldsymbol{e}) \,\mathrm{d}F(\boldsymbol{sb}(Z)|e,a) = F(z|e). \qquad \square$$

References

[1] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, Network Flows: Theory, Algorithms and Applications, Prentice-Hall, 1993.

[2] M. Boddy, T.L. Dean, Deliberation scheduling for problem solving in time-constrained environments, Artificial Intelligence 67 (2) (1994) 245–285.

[3] J.A. Boyan, M.L. Littman, Exact solutions to time-dependent MDPs, Advances in Neural Information Processing Systems 13 (2001) 1026–1032.

[4] I. Chabini, S. Lan, Adaptation of the A^* algorithm for the computation of fastest paths in the discrete-time dynamic networks, IEEE Transactions on Intelligent Transportation Systems 3 (1) (2002) 60–74.

[5] H. Chan, A. Darwiche, When do numbers really matter? Journal of Artificial Intelligence Research 17 (2002) 265–287.

[6] L. Chrisman, Incremental conditioning of lower and upper probabilities, International Journal of Approximate Reasoning 13 (1) (1995) 1–25.

[7] P. Dagum, M. Luby, Approximating probabilistic inference in Bayesian belief networks is NP-hard, Artificial Intelligence 60 (1) (1993) 141–153.

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

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

[10] N. Deo, C. Pang, Shortest path algorithms: Taxonomy and annotation, Networks 14 (1984) 275–323.

[11] E.W. Dijkstra, A note on two problems in connexion with graphs, Numerische Mathematik 1 (1959) 269–271.

[12] D.L. Draper, S. Hanks, Localized partial evaluation of belief networks, in: Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, 1994, pp. 170–177.

[13] S.E. Dreyfus, An appraisal of some shortest-path algorithms, Operations Research 17 (1969) 395–412.

[14] M.J. Druzdzel, M. Henrion, Efficient reasoning in qualitative probabilistic networks, in: Proceedings of the Eleventh National Conference on Artificial Intelligence, 1993, pp. 548–553.

[15] P.C. Fishburn, R.G. Vickson, Theoretical foundations of stochastic dominance, in: Whitmore and Findlay [53].

[16] M. Goldszmidt, Research issues in qualitative and abstract probability, AI Magazine 15 (4) (1994) 63–65.

[17] M. Goldszmidt, J. Pearl, Reasoning with qualitative probabilities can be tractable, in: Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence, 1992, pp. 112–120.

[18] V. Ha, P. Haddawy, Geometric foundations for interval-based probabilities, in: KR'98: Principles of Knowledge Representation and Reasoning, 1998, pp. 582–593.

[19] P. Haddawy, An overview of some recent developments in Bayesian problem-solving techniques, AI Magazine 20 (2) (1999) 11–19.

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

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

[22] M. Henrion, M.J. Druzdzel, Qualitative propagation and scenario-based approaches to explanation of probabilistic reasoning, in: P. Bonissone, M. Henrion, L. Kanal, J. Lemmer (Eds.), Uncertainty in Artificial Intelligence 6, Elsevier, 1991, pp. 17–32.

[23] E. Horvitz, Computation and Action Under Bounded Resources, PhD Thesis, Stanford University, Stanford, California, USA, 1990.

[24] F.V. Jensen, Bayesian Networks and Decision Graphs, Springer-Verlag, New York, 2001.

[25] M.I. Jordan, Z. Ghahramani, T. Jaakkola, L.K. Saul, An introduction to variational methods for graphical models, Machine Learning 37 (2) (1999) 183–233.

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

[27] U. Kjærulff, Reduction of computational complexity in Bayesian networks through removal of weak independences, in: Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, 1994, pp. 374–382.

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

[29] C.-L. Liu, Advances in applying stochastic-dominance relationships to bounding probability distributions in Bayesian networks, in: Proceedings of the IASTED International Conference on Artificial and Computational Intelligence 2002, 2002, pp. 251–256.

[30] C.-L. Liu, M.P. Wellman, Incremental tradeoff resolution in qualitative probabilistic networks, in: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, 1998, pp. 338–345.

[31] C.-L. Liu, M.P. Wellman, Using qualitative relationships for bounding probability distributions, in: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, 1998, pp. 346–353.

[32] C.-L. Liu, M.P. Wellman, Using stochastic-dominance relationships for bounding travel times in stochastic networks, in: Proceedings of the Second International IEEE Conference on Intelligent Transportation Systems, 1999, pp. 55–60.

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

[34] T.P. Minka, Expectation propagation for approximate Bayesian inference, in: Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence, 2001, pp. 362–369.

[35] R.E. Neapolitan, Probabilistic Reasoning in Expert Systems: Theory and Algorithms, Wiley, 1990.

[36] A.C. Osseiran, Qualitative Bayesian networks, Information Sciences 131 (1–4) (2001) 87–106.

[37] S. Parsons, Refining reasoning in qualitative probabilistic networks, in: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, 1995, pp. 427–433.

[38] S. Parsons, Qualitative Methods for Reasoning under Uncertainty, MIT Press, 2001.

[39] S. Parsons, M. Dohnal, A semiqualitative approach to reasoning in probabilistic networks, Applied Artificial Intelligence 7 (3) (1993) 223–235.

[40] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, 1988.

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

[42] D. Poole, Context-specific approximation in probabilistic inference, in: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, 1998, pp. 447–454.

[43] S. Renooij, L.C. van der Gaag, Enhancing QPNs for trade-off resolution, in: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, 1999, pp. 559–566.

[44] S. Renooij, L.C. van der Gaag, Exploiting non-monotonic influences in qualitative belief networks, in: Proceedings of the Eighth International Conference on Information Processing and Management of Uncertainty, 2000, pp. 1285–1290.

[45] S. Renooij, L.C. van der Gaag, S. Parsons, Context-specific sign-propagation in qualitative probabilistic networks, Artificial Intelligence 140 (1–2) (2002) 207–230.

[46] S. Renooij, L.C. van der Gaag, S. Parsons, S. Green, Pivotal pruning of trade-offs in QPNs, in: Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence, 2000, pp. 515–522.

[47] R.D. Shachter, Probabilistic inference and influence diagrams, Operation Research 36 (4) (1988) 589–604.

[48] R.D. Shachter, Bayes-ball: The rational pastime (for determining irrelevance and requisite information in belief networks and influence diagrams), in: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, 1998, pp. 480–487.

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

[50] M.P. Wellman, Fundamental concepts of qualitative probabilistic networks, Artificial Intelligence 44 (3) (1990) 257–303.

[51] M.P. Wellman, Some varieties of qualitative probability, in: Proceedings of the Fifth International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, 1994, pp. 437–442.

[52] M.P. Wellman, M. Ford, K. Larson, Path planning under time-dependent uncertainty, in: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, 1995, pp. 532–539.

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