Solving CLQG Influence Diagrams Using Arc-Reversal Operations in a Strong Junction Tree

Anders L Madsen
HUGIN EXPERT A/S, Gasværksvej 5, DK-9000 Aalborg, Denmark
Anders.L.Madsen@hugin.com

Abstract

This paper presents an architecture for solving conditional linear-quadratic Gaussian (CLQG) influence diagrams (IDs) by Lazy Propagation (LP). A strong junction tree (SJT) is used to guide the elimination order whereas the marginalization operation is based on arc-reversal (AR). The use of AR for marginalization simplifies the implementation and gives the architecture a number of advantages. The key benefits of using LP in combination with AR to solve CLQG IDs are illustrated by examples and in experiments. The results of a preliminary performance evaluation are promising.

1 INTRODUCTION

The ID (Howard and Matheson, 1984) is an increasingly popular knowledge representation for decision making under uncertainty. It provides an intuitive graphical representation of a decision problem with a minimum of clutter and confusion for the decision maker and analyst (Shachter and Peot, 1992).

Some of the most popular algorithms for solving IDs, e.g., Olmsted (1983), Shachter (1986), Shenoy (1992), and Jensen et al. (1994), consider the discrete case only. Recently, an increased interest in IDs with continuous as well as mixed continuous and discrete variables has emerged. Kenley (1986) and Shachter and Kenley (1989) introduced the Gaussian ID consisting of continuous variables only and an architecture for its solution. The architecture assumes a single utility function (UF) conditioned on all variables in the ID. The solution process is based on AR operations. Later Poland (1994) introduced an architecture for representing and solving continuous IDs by approximating continuous distributions with Gaussian mixtures. The solution process of this architecture is also based on AR operations. Later Madsen and Jensen (2005) introduced Shenoy-Shafer and LP architectures for solving CLQG IDs. These two architectures are based on a new representation of UFs and representations and operations of Lauritzen and Jensen (2001) and Shachter and Kenley (1989). At the same time, Cobb and Shenoy (2004) introduced an architecture for solving hybrid IDs using mixtures of truncated exponentials (MTEs).

The new architecture introduced in this paper extends LP (Madsen, 2006) with representations for UFs (Madsen and Jensen, 2005) and operations for eliminating random and decision variables (DVs) from UFs. It is simpler than the architecture of Madsen and Jensen (2005). We refer to the proposed architecture as *LARP* as it is based on LP using AR operations (Cowell, 2005; Madsen, 2006) for variable elimination. We make an empirical analysis of the efficiency of LARP using randomly generated IDs and different versions of a famous example. In addition, we compare LARP with the HUGIN algorithm (Jensen et al., 1994).

2 PRELIMINARIES

2.1 CLQG INFLUENCE DIAGRAM

A CLQG ID $\mathcal{N} = (\mathcal{X}, \mathcal{G}, \mathcal{P}, \mathcal{F}, \mathcal{U})$ over variables $\mathcal{X}$ consists of a DAG $\mathcal{G}$, a set of conditional probability distributions $\mathcal{P} = \{\mathsf{P}(\mathsf{X}|\ \mathsf{pa}(\mathsf{X})) : \mathsf{X} \in \Delta_{\mathsf{C}}\}$ where $\mathsf{pa}(\mathsf{X})$ is the set of variables corresponding to the parents of X in $\mathcal{G}$, a set of con-

ditional linear Gaussian (CLG) density functions $\mathcal{F} = \{f(Y|\,pa(Y)) : Y \in \Gamma_C\}$ and a set of quadratic UFs $\mathcal{U}$ where Δ is the set of discrete variables s.t. Δ_C is the set of discrete random variables (RVs) and Δ_D is the set of discrete DVs, Γ is the set of continuous variables s.t. Γ_C is the set of continuous RVs and Γ_D is the set of continuous DVs, i.e., $\mathcal{X} = \Delta_C \cup \Gamma_C \cup \Delta_D \cup \Gamma_D$. We denote the set of RVs as $\mathcal{X}_C = \Delta_C \cup \Gamma_C$ and the set of DVs as $\mathcal{X}_D = \Delta_D \cup \Gamma_D$ s.t. $\mathcal{X} = \mathcal{X}_C \cup \mathcal{X}_D$. The variables $\Delta_C \cup \Gamma_C$ induce a CLG distribution conditional on $\Delta_D \cup \Gamma_D$ s.t. $P(\Delta_C|\Delta_D) \cdot f(\Gamma_C|\Delta, \Gamma_D)$ equals:

$$\prod_{X \in \Delta_C} P(X|\,pa(X)) \cdot \prod_{Y \in \Gamma_C} f(Y|\,pa(Y)).$$

The variables of $\mathcal{N}$ induce an expected UF:

$$EU(\mathcal{X}) = P(\Delta_C|\Delta_D) \cdot f(\Gamma_C|\Delta, \Gamma_D) \cdot \sum_{u \in \mathcal{U}} u. \quad (1)$$

An optimal strategy can be identified by eliminating variables from (1) in the reverse time order. The time order is the order in which variables are observed s.t. $\mathcal{I}_i$ is the set of variables observed after the ith decision and before the $i+1$th decision.

Let $Y \in \Gamma_C$ with $I = pa(Y) \cap \Delta$ and $Z = pa(Y) \cap \Gamma$, then Y has a CLG distribution if:

$$f(Y|I = i, Z = z) = \mathcal{N}(\alpha(i) + \beta(i)z, \sigma^2(i)), \quad (2)$$

where the mean value is linear in the values of Z, while the covariance matrix is independent of Z. In (2), $\alpha(i)$ is a table of real numbers, $\beta(i)$ is a table of $|Z|$-dimensional row vectors and $\sigma^2(i)$ is a table of non-negative values.

A quadratic UF has the form $u(X = x, I = i) = x^T Q(i) x + R(i) x + S(i)$ where X is a $|X| \times 1$-dimensional vector of continuous variables, $I \subseteq \Delta$, $Q(i) := (q_{jk})_{|X| \times |X|}(i)$ is a table of $|X| \times |X|$ symmetric negative semi-definite matrices, $R(i) = (r_k)_{|X|}(i)$ is a table of $1 \times |X|$ vectors and $S(i)$ is a table of constants. Thus, a UF is represented as a triple $[Q, R, S]$.

We assume the UF to be a negative quadratic function as a weighted average of quadratic functions is quadratic. This implies that optimization of DVs and elimination of RVs from (1) have closed form solutions. Notice that the UFs specified in the model need not be negative quadratic. It is sufficient if the UF over which a continuous DV is maximized is negative quadratic (or constant).

We let $\mathcal{G}(\mathcal{P} \cup \mathcal{F} \cup \mathcal{U})$ denote the domain graph spanned by $\mathcal{P} \cup \mathcal{F} \cup \mathcal{U}$ and $C_{\mathcal{G}}(X)$ denote the conditioning variables of X in $\mathcal{G}$ where subscript $\mathcal{G}$ is omitted when no confusion is possible.

A RV X in $\mathcal{G}(\mathcal{P} \cup \mathcal{F} \cup \mathcal{U})$ is *probabilistic barren* w.r.t. $T \subseteq \mathcal{X}$ if it is barren w.r.t. T in $\mathcal{G}(\mathcal{P} \cup \mathcal{F})$. Let D_i be the ith decision in an ID. A variable $X \in \mathcal{I}_{k>=i}$ is *irrelevant* for D_i if $X \perp de(D_i) \cap \mathcal{U} |\, pa(D_i)$ where $\perp$ denotes d-separation and $de(D_i)$ are the descendants of D_i while a variable $Y \in pa(D_i)$ is *non-requisite* for D_i if $X \perp \mathcal{U} \cap de(D_i) |\, pa(D_i) \setminus \{Y\}$, see, e.g., Nielsen (2001) for more details.

2.2 THE AR OPERATION

The AR operation (Shachter, 1986; Cowell, 2005) changes the direction of an edge between two variables. Let $X_i, X_j \in \Delta$ with $C(X_i) = \{Z_1, \ldots, Z_n\} \subseteq \Delta$ and $C(X_j) = \{X_i, Z_1, \ldots, Z_n\} \subseteq \Delta$ s.t. $p(X_j|X_i, Z_1, \ldots, Z_n)$ and $p(X_i|Z_1, \ldots, Z_n)$ are the corresponding probability distributions of X_i and X_j, respectively. The edge (X_i, X_j) is reversed by setting:

$$p(X_j|Z_1, \ldots, Z_n) = \sum_{X_i} p(X_j|X_i, Z_1, \ldots, Z_n) p(X_i|Z_1, \ldots, Z_n),$$

$$p(X_i|X_j, Z_1, \ldots, Z_n) = \frac{p(X_j|X_i, Z_1, \ldots, Z_n) p(X_i|Z_1, \ldots, Z_n)}{p(X_j|Z_1, \ldots, Z_n)}. \quad (3)$$

The AR operation can also be applied to a pair of density functions (Cowell, 2005). Let $Y_i, Y_j \in \Gamma$ with $C(Y_j) = \{Z_1, \ldots, Z_n\} \subseteq \Gamma$ and $C(Y_i) = \{Y_j, Z_1, \ldots, Z_n\} \subseteq \Gamma$ s.t.:

$$Y_i|Y_j, Z_1, \ldots, Z_n \sim \mathcal{N}(\alpha_{Y_i} + \beta_{Y_j} Y_j + \sum_{i=1}^{n} \beta_i Z_i, \sigma^2_{Y_i}),$$

$$Y_j|Z_1, \ldots, Z_n \sim \mathcal{N}(\alpha_{Y_j} + \sum_{i=1}^{n} \delta_i Z_i, \sigma^2_{Y_j}).$$

The distributions of Y_i and Y_j after AR are:

$$Y_i|Z_1,\ldots,Z_n \sim$$
$$\mathcal{N}((\alpha_{Y_i}+\beta_{Y_j})+\sum_{i=1}^{n}(\beta_i+\delta_i)Z_i, \sigma^2_{Y_i}+\beta^2_{Y_j}\sigma^2_{Y_j}),$$
$$Y_j|Y_i,Z_1,\ldots,Z_n \sim \mathcal{N}\left(\frac{\alpha_{Y_j}\sigma^2_{Y_i}+\mu}{d}, \frac{\sigma^2_{Y_j}\sigma^2_{Y_i}}{d}\right),$$

where $d = \sigma^2_{Y_i}+\beta^2_{Y_j}\sigma^2_{Y_j}$ and μ equals:

$$-\alpha_{Y_i}\beta_{Y_j}\sigma^2_{Y_j}+\beta_{Y_j}\sigma^2_{Y_j}Y_i+\sum_{i=1}^{n}(\delta_i\sigma^2_{Y_i}-\beta_i\beta_{Y_j}\sigma^2_{Y_j})Z_i.$$

See Cowell (2005) and Madsen (2006) for more details. If $C(Y_i)\cap\Delta = C(Y_j)\cap\Delta = K$, then the formulas are indexed by k.

3 POTENTIALS

Definition 1. A *potential* $\pi_W = (\mathcal{P},\mathcal{F},\mathcal{U})$ on $W \subseteq \mathcal{X}$ consists of a set of conditional probability potentials $\mathcal{P}$ over subsets of W, a set of CLG density functions $\mathcal{F}$ over subsets of $W\cap\Gamma$ conditional on subsets of $W\cap\Delta$ and a set of UFs $\mathcal{U}$ over subsets of W.

Elements of $\mathcal{P}$ are referred to as *factors* and elements of $\mathcal{F}$ as *density functions*. Furthermore, we call the potential $\pi_W = (\emptyset,\emptyset,\emptyset)$ *vacuous* and denote it $\pi_\emptyset$. The domain of $\mathcal{P}$ is $\mathsf{dom}(\mathcal{P}) = \bigcup_{p\in\mathcal{P}}\mathsf{dom}(p)$ where $\mathsf{dom}(p)$ denotes the set of variables over which p is defined (similarly for $\mathsf{dom}(\mathcal{F})$ and $\mathsf{dom}(\mathcal{U})$). The domain of a potential $\pi = (\mathcal{P},\mathcal{F},\mathcal{U})$ is defined as $\mathsf{dom}(\pi) = \mathsf{dom}(\mathcal{P})\cup\mathsf{dom}(\mathcal{F})\cup\mathsf{dom}(\mathcal{U})$. Finally, we define $\mathcal{G}(\pi) = \mathcal{G}(\mathcal{P}\cup\mathcal{F}\cup\mathcal{U})$.

Definition 2. The *combination* of potentials $\pi_{W_1} = (\mathcal{P}_1,\mathcal{F}_1,\mathcal{U}_1)$ and $\pi_{W_2} = (\mathcal{P}_2,\mathcal{F}_2,\mathcal{U}_2)$ denotes the potential on $W_1\cup W_2$ given by $\pi_{W_1}\otimes\pi_{W_2} = (\mathcal{P}_1\cup\mathcal{P}_2,\mathcal{F}_1\cup\mathcal{F}_2,\mathcal{U}_1\cup\mathcal{U}_2)$.

Potential combination is simply set union.

Definition 3. A *projection* of $\pi_W = (\mathcal{P}_W,\mathcal{F}_W,\mathcal{U}_W)$ to a subset $V\subseteq W$ denotes the potential $\pi_V = \pi_W^{\downarrow V} = (\mathcal{P}_V,\mathcal{F}_V,\mathcal{U}_V)$ obtained by performing a sequence of variable eliminations of $W\setminus V$.

In projection $\downarrow$, variables $\Gamma\cap V$ are eliminated before $\Delta\cap V$ in reverse time order.

3.1 MARGINALIZATION

Madsen and Jensen (2005) define the necessary and sufficient marginalization operations for solving a CLQG ID. These operations assume the $[Q,R,S]$-representation of UFs and the $p\cdot[A,B,C]$-representation of (multivariate) CLG distributions where p is the discrete potential and A is a vector of αs, B is a matrix of βs, and C is a covariance matrix (this representation is equal to the $[p,A,B,C]$-representation introduced by Lauritzen and Jensen (2001) except for the decomposition of the potential into the discrete p and the continuous $[A,B,C]$ parts). The main contribution of this paper is that the proposed solution method considers only univariate CLG distributions, i.e., a density function has exactly one head variable. This is a significant simplification over the $[A,B,C]$-representation as it simplifies the operations related to variable elimination and eliminates the need for complex matrix operations.

Madsen (2006) describes an architecture for belief update in CLG Bayesian networks using AR operations to eliminate variables. These operations are applicable for the elimination of RVs in the process of solving a CLQG ID.

When solving an ID, variables are eliminated in the reverse temporal order and continuous variables are eliminated before discrete variables. In the process we may take advantage of irrelevance and non-requisite variables. In the following subsections we assume X is the variable to eliminate, $\mathcal{U}_X = \{u\in\mathcal{U} : X\in\mathsf{dom}(u)\}$ and $\mathsf{dom}(\mathcal{U}_X) = \{X,Z_1,\ldots,Z_n\}$. In the process of eliminating variables it may be necessary to perform straightforward domain extensions.

3.1.1 Discrete Random Variables

The marginalization of $X\in\Delta_C$ from a potential $\pi = (\mathcal{P},\emptyset,\mathcal{U})$ s.t. $\mathsf{dom}(\mathcal{U})\subseteq\Delta$ produces a new potential $\pi^*_{\mathsf{dom}(\pi)-\{X\}} = (\mathcal{P}^*,\emptyset,\mathcal{U}^*)$ where the components $\mathcal{P}^*$ and $\mathcal{U}^*$ are computed as follows. A sequence of AR operations to make X probabilistic barren in $\mathcal{G}(\mathcal{P})$ is performed. Let $\mathcal{P}_X$ denote the resulting set of factors, the set of UFs $\mathcal{U}$ is unchanged by this operation. Since X is probabilistic barren only a single factor $p\in\mathcal{P}_X$ has X in its

domain. Let $p(X|\,C(X))$ be this potential and let $u(X, C(X)) = \sum_{u\in\mathcal{U}_X} u$. Finally, we set:

$$\begin{aligned}\mathcal{P}^* &= \mathcal{P}_X \setminus \{p(X|\,C(X)) \in \mathcal{P}_X\},\\ \mathcal{U}^* &= \mathcal{U} \setminus \mathcal{U}_X \cup \mathcal{U}_{-X},\end{aligned} \quad (4)$$

where $\mathcal{U}_{-X} = \{\sum_X p(X|\,C(X)) \cdot u(X, C(X))\}$ denotes the resulting set of UFs (even though $|\mathcal{U}_{-X}| = 1$). Notice that if $\mathcal{U}_X = \emptyset$, then $\mathcal{U}^* = \mathcal{U}$. Later the marginalization operation is extended to produce a set of UFs.

3.1.2 Continuous Random Variables

The marginalization of $X \in \Gamma_C$ from a potential $\pi = (\mathcal{P}, \mathcal{F}, \mathcal{U})$ produces a new potential $\pi^*_{\mathsf{dom}(\pi)-\{X\}} = (\mathcal{P}, \mathcal{F}^*, \mathcal{U}^*)$ where $\mathcal{F}^*$ and $\mathcal{U}^*$ are computed as follows. A sequence of AR operations to make X probabilistic barren in $\mathcal{G}(\mathcal{P}\cup\mathcal{F})$ is performed. Let $\mathcal{F}_X$ denote the resulting set of density functions, the factors $\mathcal{P}$ and set of UFs $\mathcal{U}$ are unchanged by this operation. Since X is probabilistic barren a single density function $f \in \mathcal{F}_X$ has X in its domain. Let $f(X|\,C(X))$ be this density function and let $u(X, C(X)) = \sum_{u\in\mathcal{U}_X} u$. Finally, we set:

$$\begin{aligned}\mathcal{F}^* &= \mathcal{F}_X \setminus \{f(X|\,C(X)) \in \mathcal{F}_X\},\\ \mathcal{U}^* &= \mathcal{U} \setminus \mathcal{U}_X \cup \mathcal{U}_{-X}.\end{aligned} \quad (5)$$

where $\mathcal{U}_{-X} = \left\{(f(X|\,C(X)) \cdot u(X, C(X)))^{\downarrow C(X)}\right\}$ again denotes the resulting set of UFs (again $|\mathcal{U}_{-X}| = 1$). Notice that if $\mathcal{U}_X = \emptyset$, then $\mathcal{U}^* = \mathcal{U}$. Later the marginalization of $X \in \Gamma_C$ is revised s.t. it may produce a set of UFs.

The operation $(f(X|\,C(X)) \cdot u(X, C(X)))^{\downarrow C(X)}$ produces a UF $u(C(X)) = [(q^*_{ij}), (r^*_i), S^*]$ where:

$$\begin{aligned}q^*_{ij} &= q_{ij} + q_{ik}\beta_j + \beta_i q_{kj} + q_{kk}(\beta_i\beta_j),\\ r^*_i &= r_i + 2\alpha q_{ki} + (2q_{kk}\alpha + r_k)\beta_i,\\ S^* &= S + q_{kk}(\alpha^2 + \sigma^2) + r_k\alpha,\end{aligned}$$

where subscript k specifies the matrix/vector entry corresponding to variable X. If $I = C(X) \cap \Delta \neq \emptyset$, the formulas are indexed by the configuration i of I (similarly for continuous decisions).

3.1.3 Discrete Decision Variables

The marginalization of $X \in \Delta_D$ from a potential $\pi = (\emptyset, \emptyset, \mathcal{U})$ s.t. $\mathsf{dom}(\mathcal{U}) \subseteq \Delta$ produces a new potential $\pi^*_{\mathsf{dom}(\pi)-\{X\}} = (\emptyset, \emptyset, \mathcal{U} \setminus \mathcal{U}_X \cup \{\max_D \sum_{u\in\mathcal{U}_X} u\})$.

An optimal policy δ_X for X is determined as the maximizing arguments of $\sum_{u\in\mathcal{U}_X} u$.

3.1.4 Continuous Decision Variables

The marginalization of $X \in \Gamma_D$ from a potential $\pi = (\emptyset, \emptyset, \mathcal{U})$ s.t. $\mathsf{dom}(\mathcal{U}) \subseteq \Delta \cup \Gamma$ produces a new potential $\pi^*_{\mathsf{dom}(\pi)-\{X\}} = (\emptyset, \emptyset, \mathcal{U}^*)$. $\mathcal{U}^*$ is computed as:

$$\mathcal{U}^* = \mathcal{U} \setminus \mathcal{U}_X \cup \{(\sum_{u\in\mathcal{U}_X} u)^{\downarrow\{Z_1,\ldots,Z_n\}}\},$$

where $(\sum_{u\in\mathcal{U}_X} u)^{\downarrow\{Z_1,\ldots,Z_n\}} = [(q^*_{ij}), (r^*_i), S^*]$,

$$q^*_{ij} = q_{ij} - \frac{q_{ik}q_{kj}}{q_{kk}}, r^*_i = r_i - \frac{r_k q_{ki}}{q_{kk}}, S^* = S - \frac{r_k^2}{4q_{kk}}.$$

An optimal policy $\delta_X(z)$ for X is

$$\delta_X(z_1, \ldots, z_n) = -\frac{r_k + 2\sum_{i=1}^n q_{ki}z_i}{2q_{kk}}.$$

A sufficient condition for the marginalization to be well-defined is $q_{kk} < 0$ (for all discrete configurations with non-zero probability) as this implies that the second order polynomial has a unique maximum with respect to X (Madsen and Jensen, 2005).

4 STRONG JUNCTION TREE

A SJT $\mathcal{T}$ with cliques $\mathcal{C}$ and separators $\mathcal{S}$ is used to solve $\mathcal{N}$. Basically, $\mathcal{T}$ is used as a computational caching structure to guide the solution process, i.e., the order in which variables are eliminated, as $\mathcal{T}$ induce a partial order on the elimination order. $\mathcal{T}$ is constructed by moralization and strong triangulation of $\mathcal{G}$.

4.1 INITIALIZATION

The initialization of $\mathcal{T}$ consists of the following steps: (1) associate $\pi_\emptyset$ with each clique $C \in \mathcal{C}$, (2) for each $X \in \Delta, Y \in \Gamma$, assign $P(X|\,\mathsf{pa}(X)) \in \mathcal{P}, f(Y|\,\mathsf{pa}(Y)) \in \mathcal{F}$ to the clique C closest to strong root R s.t. $\mathrm{fa}(X), \mathrm{fa}(Y) \subseteq C$ where $\mathrm{fa}(X) = \mathsf{pa}(X) \cup \{X\}, \mathrm{fa}(Y) = \mathsf{pa}(Y) \cup \{Y\}$ and (3) for each UF $u \in \mathcal{U}$, assign u to the clique C closest to R s.t. $\mathsf{dom}(u) \subseteq C$. After initialization each

clique C holds a potential $\pi_C = (\mathcal{P}, \mathcal{F}, \mathcal{U})$. The potential $\pi_{\mathcal{X}} = \bigotimes_{C \in \mathcal{C}} \pi_C$ on $\mathcal{T}$ is a decomposition of the expected UF over $\mathcal{X}$:

$$\Big(\bigcup_{X \in \Delta_C} \{P(X|\, \mathsf{pa}(X))\}, \bigcup_{Y \in \Gamma_C} \{f(Y|\, \mathsf{pa}(Y))\}, \bigcup_{u \in \mathcal{U}} \{u\} \Big).$$

4.2 MESSAGE PASSING

The solution process in $\mathcal{T}$ proceeds by message passing via the separators $\mathcal{S}$. The separator $S = A \cap B$ between two adjacent cliques A and B stores the message passed between A and B. Messages are passed from leaf cliques toward R by recursively letting each clique A pass a message to its parent B whenever A has received a message from each $C \in \mathrm{adj}(A) \setminus \{B\}$ where $\mathrm{adj}(C)$ is the set of cliques adjacent to C.

Hence, a clique A sends a message $\pi_{A \to B}$ to its parent B when it has received messages from all its children s.t. $\pi_{A \to B} = \left(\pi_A \otimes \left(\otimes_{C \in \mathrm{adj}(A) \setminus \{B\}} \pi_{C \to A}\right)\right)^{\downarrow B}$ where $\pi_{C \to A}$ is the message passed from C to A.

5 SOLVING A CLQG ID

The solution of a CLQG ID $\mathcal{N}$ using LARP proceeds in two steps: 1) construction and initialization of a SJT representation $\mathcal{T}$ of $\mathcal{N}$ and 2) a round of message passing from the leaves of $\mathcal{T}$ to its root R. In the process of eliminating a DV X from a UF u, an optimal policy δ_X for X is recorded as the maximizing arguments of u as described in Sections 3.1.3 and 3.1.4.

Once the variables of $\mathcal{X}$ have been eliminated, an optimal strategy $\widehat{\Delta} = \{\delta_X : X \in \mathcal{X}_D\}$ has been identified and $\mathsf{EU}(\widehat{\Delta})$ computed.

Since an ID over discrete variables only is a special case of a CLQG ID, LARP can also be used to solve discrete IDs.

5.1 DISTRIBUTIVE LAW

The distributive law of algebra (DL) may be exploited in the solution process, see e.g., (Madsen and Jensen, 1999; Dechter, 2000; Pralet et al., 2006). Consider the example where $X \in \Delta_C$ is to be eliminated from a sum of UFs:

$$U(Y, T, Z) = \sum_X P(X) \left(U(X, Y, Z) + U(X, T)\right).$$

Using DL the expression is rewritten:

$$U(Y, Z) + U(T) = \sum_X P(X) U(X, Y, Z) + \sum_X P(X) U(X, T).$$

This use of DL reduces the number of arithmetic operations in the marginalization of X and supports the decompositions of a UF into a set of UFs. This may reduce the total number of arithmetic operations. In (4) and (5) $\mathcal{U}_{-X}$ is specified as a set of UFs to indicate a possible use of DL. Figure 1 shows an example where the application of DL reduces the computational cost significantly.

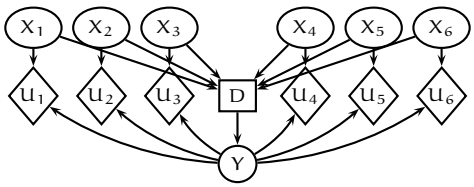

Figure 1: Example of DL utilization.

If we assume $\|Y\| = 100$, $\|X_i\| = 5$ and $\|D\| = 10$, then the advantage of applying DL becomes apparent. The optimal SJT for $\mathcal{N}$ consists of a single clique including all variables. Hence, there is no structure to exploit in the graph $\mathcal{G}$. The state space size of the largest factor when DL is not used is $15,626,000$ whereas it is $156,250$ when DL is used. In the former case $\mathcal{U}_{-X} = \{U(X_1, \ldots, X_6, D)\}$ and in the latter case $\mathcal{U}_{-X} = \{U(X_1, D), \ldots, U(X_6, D)\}$.

The SPI principle (Shachter et al., 1990) may be applied to combine UFs pairwise. The variable X may be eliminated at any step of the process, i.e., X may be eliminated from each UF separately, from the sum of a subset of UFs or from the sum of all UFs. The latter is the traditional approach. Notice that the result of eliminating X is a set of UFs with zero to n UFs where $n = |\mathcal{U}_X|$ (zero only when $n = 0$).

5.2 DECOMPOSITION

Decomposition of clique and separator potentials facilitates, for instance, the exploitation of irrelevant variables in the solution process. Figure 2 shows the *jjd* network (Jensen et al., 1994)

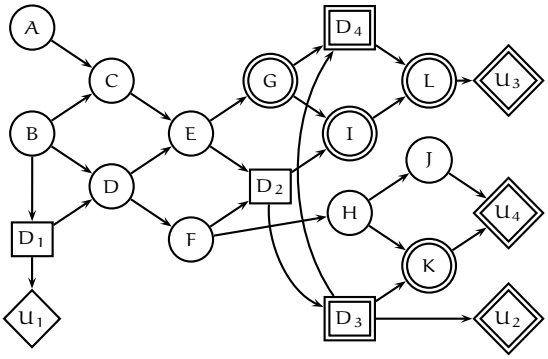

Figure 2: The *jjd* network with mixed variables.

and Figure 3 shows a SJT representation (separators are not shown while link directions specify the flow of messages towards the strong root during the solution process).

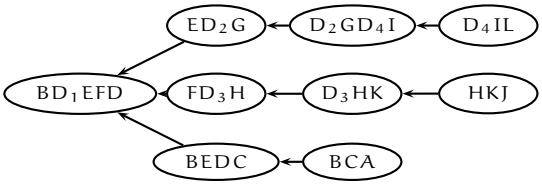

Figure 3: A SJT for *jjd*.

From Figure 2 and the definition of relevance, it is clear that RV D is irrelevant for decisions D_2, D_3 and D_4. In the root clique the elimination process may proceed as:

$$\begin{aligned}\mathrm{EU}(\hat{\Delta}) \;=\; & \sum_{\mathsf{B}} \mathsf{P}(\mathsf{B}) \max_{\mathsf{D}_1} \big(\mathsf{U}_1(\mathsf{D}_1) \\ & + \sum_{\mathsf{D}} \mathsf{P}(\mathsf{D}|\mathsf{B},\mathsf{D}_1) \big(\sum_{\mathsf{E}} \mathsf{P}(\mathsf{E}|\mathsf{D}) \mathsf{U}(\mathsf{E}) \\ & + \sum_{\mathsf{F}} \mathsf{P}(\mathsf{F}|\mathsf{D}) \mathsf{U}(\mathsf{F}) \big) \big) .\end{aligned}$$

This more efficient calculation violates the strong elimination order, but it is facilitated by the decomposition which enables the algorithm to exploit the irrelevance of D, Figure 4 shows $\mathcal{G}(\pi_{\mathsf{BD}_1\mathsf{EFD}})$.

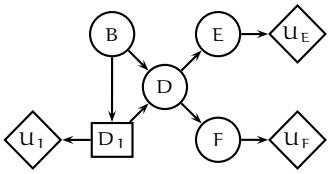

Figure 4: Domain graph for $\pi_{\mathsf{BD}_1\mathsf{EFD}}$.

6 COMPARISON

Due to space constraints comparisons are brief.

6.1 JENSEN ET AL

The HUGIN architecture solves an ID by message passing in a SJT representation. The absorption algorithm is derived based on a variable elimination approach. The initialization process combines all probability distributions assigned to a clique to form the initial clique probability potential. Similarly, for the initial utility potential. Messages are passed from the leaf cliques towards the strong root. Since an ID is solved by a collect operation from the leaves to the root, the HUGIN architecture extended to CLQG IDs reduces to a scheme, which in principle is equivalent to the Shenoy-Shafer architecture (Madsen and Jensen, 2005).

6.2 MADSEN AND JENSEN

Madsen and Jensen (2005) describe Shenoy-Shafer and LP schemes for solving a CLQG ID by message passing in a SJT. The two architectures are based on the $[\mathsf{p}, \mathsf{A}, \mathsf{B}, \mathsf{C}]$ CG potential and the $\mathsf{p} \cdot [\mathsf{A}, \mathsf{B}, \mathsf{C}]$ CG potential representation, respectively. Madsen and Jensen (2005) assume that appropriate conditioning operations have been performed as a preprocessing step before eliminating a continuous RV as CG potentials, in general, may have multiple head variables. This may require complex matrix operations. In LARP each regression is uni-variate due to the use of the AR operation. This eliminates the need for complicated combination, conditioning and matrix operations and it simplifies the implementation of the architecture.

6.3 COBB AND SHENOY

Since the class of CLQG IDs is quite restricted, MTEs have been considered for solving hybrid IDs (Cobb and Shenoy, 2004; Cobb and Shenoy, 2007) using approximation. The use of MTEs implies that there are no constraints on the relation between $\mathcal{X}_\Gamma$ and $\mathcal{X}_\Delta$ as long as the distributions of the model can be approximated using MTEs. The approach requires a multiplicative decomposition of the UF and $|\mathsf{pa}(\mathsf{X})| \leq 1$

for $X \in \Gamma_D$. These are important limitations. Also, the numerical stability and the number of terms in mixtures are major concerns.

7 PERFORMANCE ANALYSIS

7.1 RANDOM NETWORKS

We compare the performance of LARP and the HUGIN algorithm (Jensen et al., 1994) as implemented in the HUGIN Decision Engine (HDE) on a set of randomly generated networks. The random networks — generated using a revised version of the algorithm used by Vomlelová (2003) — are all discrete as the HDE does not support CLQG IDs. We include this performance comparison as LARP can be used to solve discrete IDs. Table 1[1] shows the results of experiments on eight selected networks. Networks where both algorithms ran out of memory and networks that were solved in less than a few milliseconds were disregarded. In the table, an *N/A* specifies that the solution algorithm ran out of memory. The results indicate

Table 1: Statistics on random networks.

	Time		Space	
$\|\mathcal{X}\|$	LARP	HDE	LARP	HDE
20	4.27	N/A	1,953,125	N/A
20	0.93	1.25	390,625	1,953,125
20	0.03	0.24	3,125	390,625
25	0.13	N/A	15,625	N/A
25	0.64	1.74	78,125	1,953,125
50	4.67	10.16	1,048,576	8,388,608
50	24.31	N/A	4,194,304	N/A
50	7.22	28.64	1,048,576	16,777,216

that LARP is most efficient on the networks solved by both algorithms and that it is able to solve more complex networks. The HDE performs both a collect and a distribute on the SJT though, but it has more efficient data structures and operations. LARP achieves its efficiency by maintaining decompositions of potentials.

7.2 MIXED NETWORKS

Since the HDE does not support CLQG IDs (in fact we are not aware of any system implementing CLQG IDs), we assess the performance of LARP by solving a set of CLQG IDs with the same structure, but different fractions of continuous and discrete variables. For the case of discrete variables only, the network is solved with both LARP and HDE. This will give a rough estimate on the performance of LARP.

Table 2 shows statistics on the IDs *jjd_d*, *jjd_m* and *jjd_c* where $s(C) = \prod_{X \in \Delta \cap C} \|X\|$ and $s(\mathcal{C}) = \sum_{C \in \mathcal{C}} s(C)$. The structure of the network is shown in Figure 2 where $\|X\| = 25$ for $X \in \Delta$. *jjd_d* has discrete variables only, *jjd_m* has mixed variables as indicated in Figure 2 and *jjd_c* has continuous variables only. In Table 3 the average time cost of the solution process is shown for each network. Only for *jjd_d* a value is specified for HDE. The performance of the LARP on the *jjd_d* network is a factor of seven better than the performance of HDE. The performance of LARP improves as the fraction of continuous variables increases. Each network is solved 25 times and the average time cost is computed.

Table 2: Statistics on *jjd_d*, *jjd_m* and *jjd_c*.

jjd	$\|\mathcal{C}\|$	$\max_{C \in \mathcal{C}} s(C)$	$s(\mathcal{C})$
d	9	9,765,625	10,640,625
m	9	9,765,625	10,188,826
c	9	1	1

Table 3: Average time costs in seconds.

	Time		Space	
jjd	HDE	LARP	HDE	LARP
d	3.87	0.53	9,765,625	390,625
m	-	0.35	-	390,625
c	-	0.03	-	1

7.3 DISTRIBUTE LAW

To illustrate the potential impact of DL during the solution process we solved the ID of Figure 1 using HDE, LARP and LARP with DL. The average time costs in seconds (over ten runs) are 2.91, 16.73, and 0.49, respectively. Exploiting DL produced an improvement of a factor of 34 in the time cost when $\|Y\| = 100$, $\|X_i\| = 5$ and $\|D\| = 10$. This (naïve) example was designed to illustrate the potential improvement offered by exploiting DL in the solution process.

[1] Networks with $|\mathcal{X}| \leq 25$ has $\|X\| = 5$ while networks with $|\mathcal{X}| = 50$ has $\|X\| = 2$

A simple heuristic is used to guide the application of DL on UFs. The rule is based on the score $s_{\mathrm{DL}}(\mathsf{p}_Y, \mathcal{U}) = \sum_{\mathsf{u} \in \mathcal{U}_Y} \| \mathsf{dom}(\mathsf{p}_Y) \cup \mathsf{dom}(\mathsf{u}) \|$. If the sum of the state spaces is less than the total state space, i.e., $s_{\mathrm{DL}}(\mathsf{p}_Y, \mathcal{U}) < \| \mathsf{dom}(\mathsf{p}_Y) \cup \mathsf{dom}(\mathcal{U}) \|$, then DL is applied.

8 DISCUSSION

The two main reasons for considering AR operations in a SJT are: 1) the structure of the SJT serves as an efficient caching structure where elimination orders are reconsidered dynamically and 2) the SJT offers an opportunity to distribute information. The first point is relevant if the decision problem is solved on-line (for instance, if unexpected observations have become available or the model is too large to be solved off-line) while the second point is relevant for computing probabilities of future decisions, i.e., the decision policies are encoded as conditional probability distributions, which makes it possible to compute probabilities of future decisions and events under the encoded strategy (Nilsson and Jensen, 1998).

If the UFs of the model meet the requirements of CLQG IDs, then expectation and maximization calculations have closed form solutions. This implies that the ID can be solved. In general, a CLQG IDs can be solved in closed form when all continuous variables can be eliminated before discrete variables using closed form operations. Shenoy (2006) describes an approach to modeling hybrid Bayesian networks using mixtures of Gaussian distributions. This approach can also be applied to approximate continuous distributions in mixed IDs. The resulting mixed ID should meet the requirements of a CLQG ID in order to be solvable with LARP.

Despite a significant difference in the efficiency of table operations LARP is as efficient as the HDE. The results of the performance evaluation indicate a large potential of LARP.

References

B. R. Cobb and P. P. Shenoy. 2004. Decision Making with Hybrid Influence Diagrams Using Mixtures of Truncated Exponentials. In *Proc. of UAI*, pages 85–93.

B. R. Cobb and P. P. Shenoy. 2007. Influence Diagrams with Continuous Decisions Variables and Non-Gaussian Uncertainties. *Decision Analysis*, pages 136–155.

R. G.. Cowell. 2005. Local Propagation In Conditional Gaussian Bayesian Networks. *Journal of Machine Learning Research*, 6:1517–1550.

R. Dechter. 2000. A New Perspective on Algorithms for Optimizing Policies under Uncertainty. In *Artificial Intelligence Planning Systems*, pages 72–81.

R. A. Howard and J. E. Matheson. 1984. Influence Diagrams. In *Readings in Decision Analysis*, chapter 38, pages 763–771.

F. Jensen, F. V. Jensen, and S. L. Dittmer. 1994. From Influence Diagrams to Junction Trees. In *Proc. of UAI*, pages 367–373.

C. R. Kenley. 1986. *Influence Diagram Models with Continuous Variables*. Ph.D. thesis, EES Department, Stanford University.

S. L. Lauritzen and F. Jensen. 2001. Stable Local Computation with Mixed Gaussian Distributions. *Statistics and Computing*, 11(2):191–203.

A. L. Madsen and F. V. Jensen. 1999. Lazy Evaluation of Symmetric Bayesian Decision Problems. In *Proc. of UAI*, pages 382–390.

A.L. Madsen and F. Jensen. 2005. Solving linear-quadratic conditional Gaussian influence diagrams. *International Journal of Approximate Reasoning*, 38(3):263–282.

A. L. Madsen. 2006. Variatoins Over the Message Computation Algorithm of Lazy Propagation. *IEEE Transactions on Systems, Man. and Cybernetics Part B*, 36(3):636–648.

T. D. Nielsen. 2001. Decomposition of Influence Diagrams. In *Proc. of ECSQARU*, pages 144–155.

D. Nilsson and F. V. Jensen. 1998. Probabilities of future decisions. In *Proceedings from the International Conference on Informational Processing and Management of Uncertainty in knowledge-based Systems (IPMU)*, pages 144–1462.

S.M. Olmsted. 1983. *On representing and solving decision problems*. Phd thesis, Department of Engineering-Economic Systems, Stanford University, Stanford, CA.

W. B. Poland. 1994. *Decision Analysis with Continuous and Discrete Variables: A Mixture Distribution Approach*. Ph.D. thesis, Engineering-Economic Systems, Stanford University, Stanford, CA.

C. Pralet, T. Schiex, and G. Verfaillie. 2006. From Influence Diagrams to Multioperator Cluster DAGs. In *Proc. of UAI*, pages –.

R. D. Shachter and C. R. Kenley. 1989. Gaussian influence diagrams. *Management Science*, 35(5):527–549.

R. D. Shachter and M. A. Peot. 1992. Decision making using probabilistic inference methods. In *Proc. of UAI*, pages 276–283.

R. D. Shachter, B. D'Ambrosio, and B. Del Favero. 1990. Symbolic probabilistic inference in belief networks. In *Proc. of 8th National Conference on AI*, pages 126–131.

R. D. Shachter. 1986. Evaluating influence diagrams. *Operations Research*, 34(6):871–882.

P. P. Shenoy. 1992. Valuation-Based Systems for Bayesian Decision Analysis. *Operations Research*, 40(3):463–484.

P. P. Shenoy. 2006. Inference in Hybrid Bayesian Networks Using Mixtures of Gaussians. In *Proc. of UAI*, pages 428–436.

M. Vomlelová. 2003. Unconstrained Influence Diagrams - Experiments and Heuristics. T. R, Faculty of Mathematics, Charles University.