The Thirty-Second International Florida
Artificial Intelligence Research Society Conference (FLAIRS-32)

Linear Time and Space Algorithm for Computing All the Fagin-Halpern Conditional Beliefs Generated from Consonant Belief Functions

Lalintha G. Polpitiya, Kamal Premaratne, Manohar N. Murthi
Department of Electrical and Computer Engineering
University of Miami
Coral Gables, Florida, USA
lalintha@umiami.edu, {kamal, mmurthi}@miami.edu

Abstract

This paper proposes a highly efficient exact dynamic programming algorithm for computing *all* the conditionals generated from consonant belief functions. The time and space complexities of this novel algorithm are linear for computing all the conditional beliefs, and hence it significantly outperforms the exponential time and space complexity requirements of the brute force approach and the currently available conditional computation strategies. We provide a thorough analysis and experimental validation of the utility, efficiency, and implementation of the proposed algorithm for carrying out the *Fagin-Halpern conditional*. A new computational library is developed and harnessed in the simulations.

Introduction

The flexibility and expressiveness of Dempster-Shafer (DS) theoretic models make DS evidence theory (Dempster 1967; 1968; Shafer 1976) an ideal framework for reasoning and decision making under uncertainty in Artificial Intelligence (AI) applications (Barnett 1981; Yager and Liu 2008). However, a major criticism cast towards DS theoretic (DST) evidential reasoning is the heavy computational burden it entails. Computing the DST belief functions and the DST conditionals, a critical operation in evidence updating and fusion, are non-deterministic polynomial-time hard (NP-hard) problems (Orponen 1990; Kreinovich et al. 1991). This problem has in fact been identified as an issue that requires increased attention (Shafer 2016). If the advantages offered by DS theory are to be fully realized, it is essential that data structures and algorithms which facilitate the efficient computation of DST operations are developed.

To address the computational limitations of DST methods, different approximation methods have been proposed to restrict the number of focal elements (Bauer 1997; Denœux 2016). *Consonant approximation,* which maps belief functions to possibility measures, is one such method that has been widely examined in the literature (Dubois and Prade 1990; Cuzzolin 2014). The number of focal elements generated from this method is linear with the size of the frame of discernment (FoD), or sample space, Θ. The focal elements of the consonant belief function (CBF) that it produces are nested and form a totally ordered collection of subsets (Shafer 1976). CBFs can be used to model partial ignorance and they are computationally less expensive than general belief functions (Cuzzolin 2014).

How can we efficiently compute conditionals associated with CBFs? The conditional operation plays a fundamental role in evidence updating and fusion and, in general, in reasoning under uncertainty. Of the numerous strategies of DST conditioning that have appeared in the literature (Shafer 1976; 1981; Ichihashi and Tanaka 1989; Fagin and Halpern 1990; Smets 1991; Yu and Arasta 1994), *Dempster's conditional* and *Fagin-Halpern (FH) conditional* can be considered the most widely used two DST conditional notions.

A widely used approach for carrying out precise computation of the Dempster's conditional is a matrix calculus based algorithm which generates the Dempster's conditional masses (Klawonn and Smets 1992; Smets 2002). However, the associated time and space complexity are both $\mathcal{O}(2^{|\Theta|} \times 2^{|\Theta|})$ ($|\Theta|$ refers to the cardinality of the FoD Θ). Therefore, this specialization matrix-based method imposes a prohibitive burden when dealing with larger FoDs. In addition, it cannot be used to compute the FH conditional.

For FH conditional computation, the Conditional Core Theorem (CCT) in (Wickramarathne, Premaratne, and Murthi 2013) can be used to *identify* propositions that retain non-zero support after FH conditioning. However, it does not address how one may *compute* the FH conditionals of these propositions. The work in (Polpitiya et al. 2016; 2017) provides efficient data structures and algorithms for computing belief functions and their conditionals. These are exponential algorithms which work efficiently on general belief functions (when no additional structure is imposed on them) and they can be used to compute both the Dempster's conditional and Fagin-Halpern conditional. However, these algorithms do not exploit the CBF structure which possesses only a small number of focal elements. The brute force approach of computing all the conditionals of CBFs takes exponential time, and exponential space to store the result.

In this paper, we propose a highly efficient exact dynamic programming algorithm for computing all the FH conditionals generated from CBFs. A similar algorithm to compute Dempster's conditionals can also be developed. However, due to page limitations, our discussion is restricted to the FH conditionals case only. The FH conditional can in fact be considered the most natural generalization of the probabilistic conditional notion because of its close connection with the inner/outer conditional probability measures (Fagin and Halpern 1990; Wickramarathne, Premaratne, and Murthi 2013).

Computation of all the conditional beliefs are achieved in linear time and space complexity. It significantly outperforms the exponential time and space complexity requirements of the brute force approach and the currently available conditional computation strategies. We provide a thorough analysis and experimental validation of the utility, efficiency, and implementation of the proposed algorithm when computing the Fagin-Halpern conditional. A new computational library, which we refer to as *FH-CBF (Fagin-Halpern conditional beliefs generated from Consonant Belief Functions)* is developed and harnessed in the simulations (ProFuSELab 2019).

This paper is organized as follows: First, we provide a review of essential DST notions. This is followed by the proposed efficient exact dynamic programming algorithm. The experimental results are provided next, followed by concluding remarks.

Preliminaries

DST Basic Notions

In DS theory, the *frame of discernment (FoD)* refers to the set of all possible mutually exclusive and exhaustive propositions (Shafer 1976). We consider the case where the FoD is finite and we denote it as $\Theta = \{\theta_0, \theta_1, \ldots, \theta_{n-1}\}$. Proposition $\{\theta_i\}$, which is referred to as a *singleton,* represents the lowest level of discernible information. The power set of Θ, denoted by 2^Θ, form all the propositions of interest in DS theory. A proposition that is not a singleton is referred to as a *composite.* The set $A \backslash B$ denotes all singletons in $A \subseteq \Theta$ that are not included in $B \subseteq \Theta$, i.e., $A \backslash B = \{\theta_i \in \Theta \mid \theta_i \in A, \theta_i \notin B\}$. We use $\overline{A}$ to denote $\Theta \backslash A$ and $|A|$ to denote the cardinality of A.

Definition 1 (Basic Belief Assignment (BBA) or Masses)**.** *The mapping* $m : 2^\Theta \mapsto [0, 1]$ *is said to be a* basic belief assignment (BBA) *or a* mass assignment *if* $m(\emptyset) = 0$ *and* $\sum_{A \subseteq \Theta} m(A) = 1.$ ■

So, the mass captures the 'support' that is strictly allocated to a given proposition. The mass of a composite proposition (a general focal element) is free to move into its subsets (e.g., into individual singletons), which allows one to model the notion of *ignorance.* Complete ignorance is captured via the *vacuous BBA* 1_Θ*:* $m(A) = 1$ for $A = \Theta$, and $m(A) = 0$ for $A \subset \Theta$. Propositions that possess nonzero mass are referred to as *focal elements;* the set of all focal elements in an FoD is referred to as its *core* $\mathfrak{F}$*,* i.e., $\mathfrak{F} = \{A \subseteq \Theta \mid m(A) > 0\}$. Note that $|\mathfrak{F}|$ is the number of focal elements. $\mathcal{E} = \{\Theta, \mathfrak{F}, m(\cdot)\}$ is referred to as the *body of evidence (BoE).*

The *belief* assigned to a proposition takes into account the support for all of its subsets.

Definition 2 (Belief)**.** *Given a BoE* $\mathcal{E} = \{\Theta, \mathfrak{F}, m(\cdot)\}$*, the* belief *assigned to* $A \subseteq \Theta$ *is* $Bl : 2^\Theta \mapsto [0, 1]$ *where* $Bl(A) = \sum_{B \subseteq A} m(B).$ ■

Propositions that possess non-zero belief are denoted by $\hat{\mathfrak{F}}$, i.e., $\hat{\mathfrak{F}} = \{A \subseteq \Theta \mid Bl(A) > 0\}$.

The *plausibility* measures the extent to which a proposition is plausible, i.e., the amount of belief not strictly supporting the complement of the proposition.

Definition 3 (Plausibility)**.** *Given a BoE* $\mathcal{E} = \{\Theta, \mathfrak{F}, m(\cdot)\}$*, the* plausibility *assigned to* $A \subseteq \Theta$ *is* $Pl : 2^\Theta \mapsto [0, 1]$ *where*

$$Pl(A) = 1 - Bl(\overline{A}).$$ ■

It is easy to see that, for all $A \subseteq \Theta$,

$$Pl(A) = \sum_{\substack{B \subseteq \Theta \\ B \cap A \neq \emptyset}} m(B) = 1 - Bl(\overline{A}) \geq Bl(A), \forall A \subseteq \Theta. \tag{1}$$

Also, given a valid belief function $Bl : 2^\Theta \mapsto [0, 1]$, one may generate the corresponding BBA $m : 2^\Theta \mapsto [0, 1]$ via the Möbius transform (Shafer 1976):

$$m(A) = \sum_{B \subseteq A} (-1)^{|A \backslash B|} Bl(B), \ \forall A \subseteq \Theta. \tag{2}$$

For convenience, we will employ the following notation:

$$\mathcal{S}(A; B) = \sum_{\substack{\emptyset \neq C \subseteq A; \\ \emptyset \neq D \subseteq B}} m(C \cup D). \tag{3}$$

Note that $\mathcal{S}(A; B)$ denotes the sum of all mass values of propositions that 'straddle' both $A \subseteq \Theta$ and $B \subseteq \Theta$.

Fagin-Halpern (FH) DST Conditional

Definition 4 (Fagin-Halpern (FH) Conditional (Fagin and Halpern 1990))**.** *Consider the BoE* $\mathcal{E} = \{\Theta, \mathfrak{F}, m(\cdot)\}$*. The* conditional belief $Bl(B|A) : 2^\Theta \mapsto [0, 1]$ *of* B *given the conditioning event* $A \subseteq \Theta$ *s.t.* $A \in \hat{\mathfrak{F}}$ *is*

$$Bl(B|A) = \frac{Bl(A \cap B)}{Bl(A \cap B) + Pl(A \cap \overline{B})}.$$ ■

Suppose the BoE $\{\Theta, \mathfrak{F}, m(\cdot)\}$ is being conditioned w.r.t. the proposition $A \in \hat{\mathfrak{F}}$. The propositions that retain a nonzero mass after conditioning are referred to as the *conditional focal elements;* the set of all such conditional focal elements is referred to as the *conditional core* $\mathfrak{F}_A$, i.e., $\mathfrak{F}_A = \{B \subseteq A \in \hat{\mathfrak{F}} \mid m(B|A) > 0\}$ (Wickramarathne, Premaratne, and Murthi 2013). Propositions that possess non-zero conditional belief are denoted by $\hat{\mathfrak{F}}_A$, i.e., $\hat{\mathfrak{F}}_A = \{B \subseteq A \in \hat{\mathfrak{F}} \mid Bl(B|A) > 0\}$.

In our work, we will exploit several results in the literature related to the Fagin-Halpern conditional (Kulasekere et al. 2004; Polpitiya et al. 2017). Of particular importance are the following results:

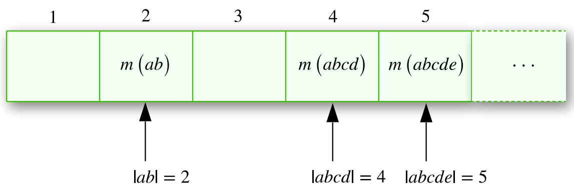

Figure 1: *Consonant BoE as a dynamic array.* Representation of three focal elements $\{a, b\}$, $\{a, b, c, d\}$, and $\{a, b, c, d, e\}$ of a CBF when $\Theta = \{a, b, c, d, e, \ldots\}$.

Lemma 1 ((Kulasekere et al. 2004))**.** *Consider the BoE* $\mathcal{E} = \{\Theta, \mathfrak{F}, m(\cdot)\}$ *and* $A \in \hat{\mathfrak{F}}$. *Then, the following are true:*
(i) $m(B|A) = 0$ *whenever* $\overline{A} \cap B \neq \emptyset$*, and*
(ii) $Bl(B|A)$ *can be expressed as*

$$Bl(B|A) = \frac{Bl(A \cap B)}{Pl(A) - \mathcal{S}(\overline{A}; A \cap B)}, \; B \subseteq A. \qquad \blacksquare$$

Note that, **(i)** states that FH conditioning annuls those propositions that 'straddle' the conditioning proposition A and its complement $\overline{A}$. So, w.l.o.g., for FH conditioning, one may consider only those propositions $B \subseteq A$.

For our work, we will need the following alternate expression for the FH conditional which we will later exploit for computing all the conditional beliefs generated from CBFs.

Proposition 1 ((Polpitiya et al. 2017))**.** *Consider the BoE* $\mathcal{E} = \{\Theta, \mathfrak{F}, m(\cdot)\}$ *and* $A \in \hat{\mathfrak{F}}$. *Then,*

$$Bl(B|A) = \frac{Bl(A \cap B)}{1 - Bl(\overline{A}) - \mathcal{S}(\overline{A}; A \cap B)}, \; B \subseteq \Theta. \qquad \blacksquare$$

Consonant Belief Functions: Computing All FH Conditional Beliefs

Definition 5 (Consonant BoE)**.** *(Shafer 1976) The BoE* $\mathcal{E} = \{\Theta, \mathfrak{F}, m(\cdot)\}$ *is said to be a* consonant BoE *if its core is* $\mathfrak{F} = \{A_0, A_1, \ldots, A_{n-1}\}$, *where* $A_0 \subset A_1 \subset \ldots \subset A_{n-1} \subseteq \Theta$. *The corresponding belief function is said to be a* consonant belief function (CBF). $\blacksquare$

Representing a Consonant BoE

The focal elements of a consonant BoE can be represented in a dynamic array. The indexes of the array can be mapped to the cardinality of the relevant proposition and masses can be represented as respective array elements. See Fig. 1.

Linear Time and Space Algorithm for Computing All the FH Conditional Beliefs

Algorithm. The highly efficient exact dynamic programming algorithm for computing all the FH conditional beliefs generated from a CBF that we propose appears in Algorithm 1. It employs four dynamic arrays (see Fig. 2):

(a) $BBA[]$: This array, which contains masses of the CBF, is used in Step 1. Its size is the cardinality $|\Theta|$ of the FoD.

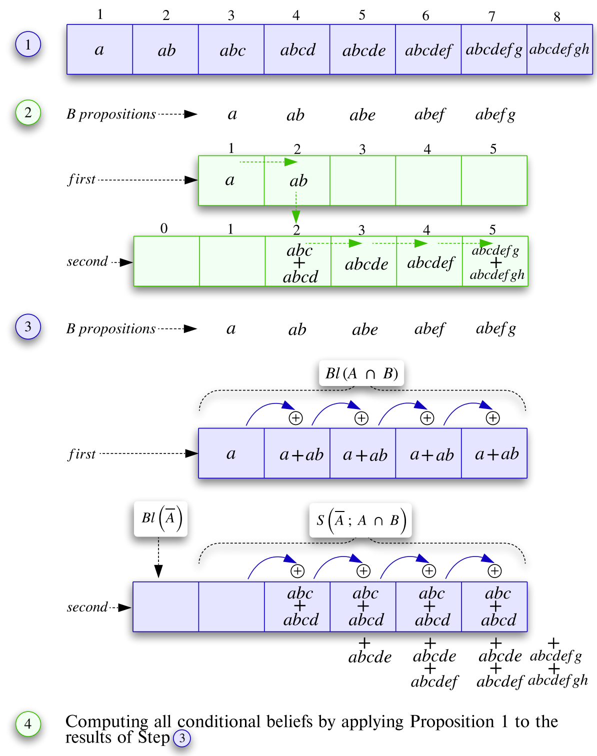

Figure 2: *Computation of all FH conditional beliefs.* Steps 1-4 of computing all conditional beliefs generated from CBFs when $\Theta = \{a, b, c, d, e, f, g, h\}$ and $A = \{a, b, e, f, g\}$.

(b) $first[]$: This array, which keeps the $Bl(A \cap B)$ values at the end of Step 3, is used in Steps 2, 3, and 4. Its size is the cardinality $|A|$ of the conditioning event.

(c) $second[]$: This array, which keeps the computed $\mathcal{S}(\overline{A}; A \cap B)$ values at the end of Step 3, is used in Steps 2, 3, and 4. Its size is $|A| + 1$.

(d) $FHcondbel[]$: This array keeps all the computed FH conditional beliefs. Its size is $|A|$.

Time Complexity. Algorithm 1 computes all the FH conditional beliefs in $\mathcal{O}(n)$ complexity. Fig. 2 provides an illustration of this algorithm when $\Theta = \{a, b, c, d, e, f, g, h\}$ and $A = \{a, b, e, f, g\}$.

Line #1: The algorithm inputs are the conditioning event A, the FoD Θ, and the consonant BoE $BBA[]$.

Lines #4-15: $BBA[]$ contains the CBF as in Step 1. In Step 2, $BBA[]$ is transformed to $first[]$ and $second[]$ arrays. After Step 2, $first[]$ contains masses relevant to possible B propositions. The element $second[0]$ contains masses which do not straddle B propositions; the remaining elements of $second[]$ contain masses which straddle the B propositions. The required number of iterations and the computational complexity of this segment are $|\Theta|$ and $\mathcal{O}(n)$, respectively.

Lines #16-19: This segment corresponds to Step 3. After Step 3, $first[]$ contains $Bl(A \cap B)$ values, $second[0]$ keeps $Bl(\overline{A})$ and the remaining elements of $second[]$ contain $\mathcal{S}(\overline{A}; A \cap B)$ values. The required number of iterations

Algorithm 1 Compute All FH Conditional Beliefs

1: **procedure** ALLFHCONDBEL(Singletons $A[]$, Singletons $\Theta[]$, ConsonantBoE $BBA[]$)
2: $ptr \leftarrow 0$
3: $ck \leftarrow true$
4: **for** each θ_i in Θ **do**
5: **if** ck **and** $ptr < |A|$ **and** $A[ptr] = \Theta[i]$ **then**
6: $ptr \leftarrow ptr + 1$
7: $first[ptr] \leftarrow BBA[i+1]$
8: **else**
9: $ck \leftarrow false$
10: **if** $ptr < |A|$ **and** $A[ptr] = \Theta[i]$ **then**
11: $ptr \leftarrow ptr + 1$
12: **end if**
13: $second[ptr] \leftarrow second[ptr] + BBA[i+1]$
14: **end if**
15: **end for**
16: **for** $i \leftarrow 2, |A|$ **do**
17: $first[i] \leftarrow first[i] + first[i-1]$
18: $second[i] \leftarrow second[i] + second[i-1]$
19: **end for**
20: **for** $i \leftarrow 1, |A|$ **do**
21: $FHcondbel[i] \leftarrow first[i]/(1 - second[0] - second[i])$
22: **end for**
23: **return** $FHcondbel[]$
24: **end procedure**

and the computational complexity of this segment are $|A|-1$ and $\mathcal{O}(|A|)$ $(\leq \mathcal{O}(|n|)$, respectively.

Lines #20-22: In Step 4, the algorithm computes all conditional beliefs by applying Proposition 1 to the results of Step 3. The required number of iterations and the computational complexity of Step 4 are $|A|$ and $\mathcal{O}(|A|)$ $(\leq \mathcal{O}(|n|)$, respectively.

Line #23: The algorithm output, $FHcondbel[]$, contains all the computed conditional belief values. $Bl(B|A)$ can be obtained by accessing the index of $|A \cap B|$ in $FHcondbel[]$.

Space Complexity of Algorithm 1. All the arrays used in the proposed algorithm are linear with the cardinality of FoD. Hence, the space complexity associated with Algorithm 1 is $\mathcal{O}(n)$.

Note that each conditioning event A may generate $2^{|A|}$ number of possible conditioned propositions. Any of these conditional beliefs can be accessed by selecting the largest cardinality of the possible B proposition (according to Fig. 2) which is a subset of the considering proposition. For example, in Fig. 2, the possible B propositions are $\{a\}$, $\{a,b\}$, $\{a,b,e\}$, $\{a,b,e,f\}$, and $\{a,b,e,f,g\}$. Consider computing the conditional belief $Bl(B|A)$, when $B = \{a,b,c\}$. Here, $\{a,b\}$ possesses the largest cardinality satisfying the above condition. So accessing the index $|\{a,b\}|$ of $FHcondbel[]$ yields the required result. Conditional beliefs $Bl(B|A)$, are equal for all of the following B propositions: $\{a,b\}$, $\{a,b,c\}$, $\{a,b,d\}$, $\{a,b,h\}$, $\{a,b,c,d\}$, $\{a,b,c,h\}$, $\{a,b,d,h\}$, and $\{a,b,c,d,h\}$. The value of $FHcondbel[|\{a,b\}|]$ represents all of them. Therefore, only linear space complexity is required to store all possible conditional values; these are stored in $FHcondbel[]$, an $|A|$-sized array.

Comparison With Other Methods

We emphasize that the proposed algorithm computes *all* the FH conditional beliefs, not *one* FH conditional belief.

Focal Elements-Based Brute Force Algorithms. *One* conditional can of course be computed in linear time $\mathcal{O}(|\mathfrak{F}|)$ and constant space $\mathcal{O}(1)$ using the brute force approach. However, the brute force approach cannot be used to compute/store *all* the conditionals in linear time/space.

To compute *all* the conditionals associated with a conditioning event A, we must consider $2^{|A|}$ number of possible conditioned propositions. The complexity then becomes $\mathcal{O}(|\mathfrak{F}|2^{|A|})$ in time and $\mathcal{O}(2^{|A|})$ in space for the brute force approach. In other words, the brute force algorithms based on focal elements require exponential time/space.

DS-Conditional-One Model. The DS-Conditional-One approach proposed in (Polpitiya et al. 2017) provides an efficient approach to compute the conditionals associated with general BoEs. In this approach the computational complexity associated with conditional belief computation of an arbitrary proposition is $\mathcal{O}(2^{|\overline{A}|+|A\cap B|})$, and of all propositions is $\mathcal{O}(2^{|A|+|\overline{A}|+|A\cap B|})$. It requires $\mathcal{O}(2^{|\Theta|})$ space complexity. This method is faster for working with general belief functions, but it does not exploit the consonant structure of CBFs.

Experiments

We have developed a new computation library in C++ which we refer to as *FH-CBF (Fagin-Halpern conditional beliefs generated from Consonant Belief Functions)* (ProFuSELab 2019). This library includes the proposed algorithm along with simulation tools. All conditional computations were carried out on a Macintosh desktop computer (iMac) running Mac OS X 10.13.6 (with 2.9GHz Intel Core i5 processor and 8GB of 1600MHz DDR3 RAM) for smaller FoDs and on a supercomputer (http://ccs.miami.edu/pegasus) for larger FoDs (underlined in Table 1).

Results were obtained by executing the algorithms for 10,000 randomly chosen conditioning (A) and conditioned ($B \subseteq A$) propositions from the FoD and noting the average CPU time. A random set of focal elements were generated in the core for each FoD size. Table 1 compares the average computational times taken by three algorithms that can be used for exact conditional computation: the proposed algorithm, focal elements-based brute force approach, and the DS-Conditional-One model (Polpitiya et al. 2017). Fig. 3 is a plot of these average computational times.

With the proposed algorithm, we computed all the FH conditional beliefs generated from CBFs. Similarly, we computed all the FH conditional beliefs with a focal elements-based brute force approach as well. With the DS-Conditional-One model, which is an algorithm to compute an arbitrary FH conditional, we again employed a brute force approach to get all the conditional beliefs. As evident from Table 1 and Fig. 3, the speed advantage provided by the pro-

$\lvert\Theta\rvert$	**Max.** $\lvert\hat{\mathfrak{F}}_A\rvert$	**Proposed Algorithm**	**Focal Elements-Based Brute Force Algorithm**	**DS-Conditional-One Based Brute Force Algorithm**
2	3	0.0012	0.0013	0.0016
4	15	0.0011	0.0015	0.0057
6	63	0.0012	0.0027	0.0189
8	255	0.0012	0.0076	0.0707
10	1,023	0.0012	0.0247	0.3038
12	4,095	0.0013	0.0836	1.5535
14	16,383	0.0013	0.3825	15.0000
16	65,535	0.0014	1.3027	131.8750
18	262,143	0.0014	5.1290	1,072.2200
20	1,048,575	0.0015	19.2470	8,670.0000
22	4,194,303	0.0016	75.0794	71,115.9000
24	16,777,215	0.0016	298.5110	653,268.0000
26	67,108,863	0.0016	1,193.4700	1.6334 cpu hours
28	268,435,455	0.0017	4,809.4100	***
30	1,073,741,823	0.0017	18,987.5000	***

Table 1: Average computational times (ms) of the three exact conditional computation methods: proposed algorithm (Algorithm 1), the focal elements-based brute force approach, and the DS-Conditional-One based brute force approach. (*** denotes computations not completed within a feasible time or space requirement).

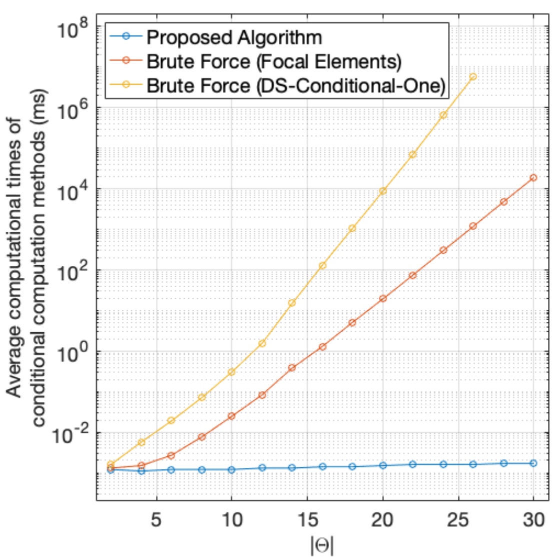

Figure 3: Average computational times (ms) of exact conditional computation methods for computing all FH conditional beliefs.

posed algorithm is significantly higher than the other strategies.

Table 2 lists the average computational times taken by the proposed algorithm for larger FoDs. Results were obtained by executing the algorithms for 10,000 randomly chosen conditioning (A) and conditioned ($B \subseteq A$) propositions from the FoD and noting the average CPU times. A random set of focal elements were generated in the core for each FoD size. Algorithm 1 computes all FH conditional beliefs of an FoD of size 10,000 within 0.2 ms. The significant speed advantage is clear from the values in Table 2.

$\lvert\Theta\rvert$	10	100	1,000	10,000
Avg. Time (ms)	0.0012	0.0038	0.0230	0.1996

Table 2: Average computational times of the proposed algorithm.

Concluding Remarks

This paper provides a highly efficient exact dynamic programming algorithm for computing all the FH conditional beliefs generated from CBFs. As an outcome of this research work, *FH-CBF* library (in C++) is made available to efficiently work with CBFs and for computing the FH conditional beliefs. We strongly believe that this algorithm constitutes a significant step toward the practical utility of the strengths offered by DST methods.

The time and space complexities of the proposed algorithm are linear for computing all conditional beliefs, thus

Conditional Computation Method	**Complexity** Time	Space
Proposed Algorithm	$\mathcal{O}(n)$	$\mathcal{O}(n)$
Brute Force on $\mathfrak{F}$	$\mathcal{O}(\lvert\mathfrak{F}\rvert 2^{\lvert A\rvert})$	$\mathcal{O}(2^{\lvert A\rvert})$
DS-Conditional-One	$\mathcal{O}(2^{\lvert A\rvert+\lvert\overline{A}\rvert+\lvert A\cap B\rvert})$	$\mathcal{O}(2^{\lvert\Theta\rvert})$

Table 3: Time and space complexity requirements of exact conditional computation methods for computing all FH conditional beliefs.

serving to reduce the memory usage and to significantly improve computational efficiency. In particular, it offers a significant improvement over the currently available exact computation algorithms which require exponential time and space complexity. For DS-Conditional-One, the complexity to compute conditional belief of an arbitrary proposition is $\mathcal{O}(2^{|\overline{A}|+|A\cap B|})$. Since it employs a brute force approach to compute all conditional beliefs, computational time becomes $\mathcal{O}(2^{|A|+|\overline{A}|+|A\cap B|})$, which is expensive for larger BoEs. Storing the BoE requires $\mathcal{O}(2^{|\Theta|})$ space. Applying brute force approach on focal elements is also exponential and the computational complexity is $\mathcal{O}(|\mathfrak{F}|2^{|A|})$. If it is required to store the computed results, the brute force approach needs $\mathcal{O}(2^{|A|})$ space. Table 3 summarizes the time and space complexity requirements of exact conditional computation methods which can be utilized to compute all FH conditional beliefs on CBFs.

Considering the uncertainty inherent in AI reasoning and the advantages of DST methods in reasoning with uncertainty, we believe that the proposed algorithm constitutes an important step toward the utility of DST methods in real AI systems.

Acknowledgments

This work is based on research supported by the U.S. Office of Naval Research (ONR) via grant #N00014-10-1-0140 and the U.S. National Science Foundation (NSF) via grant #1343430.

References

Barnett, J. A. 1981. Computational methods for A mathematical theory of evidence. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, 868–875.

Bauer, M. 1997. Approximation algorithms and decision making in the Dempster-Shafer theory of evidence — An empirical study. *International Journal of Approximate Reasoning* 17(2-3):217–237.

Cuzzolin, F. 2014. L_p Consonant Approximations of Belief Functions. *IEEE Transactions on Fuzzy Systems* 22(2):420–436.

Dempster, A. P. 1967. Upper and lower probabilities induced by a multivalued mapping. *The Annals of Mathematical Statistics* 38(2):325–339.

Dempster, A. P. 1968. A generalization of Bayesian inference. *Journal of the Royal Statistical Society. Series B* 30(2):205–247.

Denœux, T. 2016. 40 years of Dempster-Shafer theory. *International Journal of Approximate Reasoning* 79(C):1–6.

Dubois, D., and Prade, H. 1990. Consonant approximations of belief functions. *International Journal of Approximate Reasoning* 4(5-6):419–449.

Fagin, R., and Halpern, J. Y. 1990. A new approach to updating beliefs. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence (UAI)*, 347–374.

Ichihashi, H., and Tanaka, H. 1989. Jeffrey-like rules of conditioning for the Dempster-Shafer theory of evidence. *International Journal of Approximate Reasoning* 3(2):143–156.

Klawonn, F., and Smets, P. 1992. The dynamic of belief in the transferable belief model and specialization-generalization matrices. In *Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence (UAI)*, 130–137.

Kreinovich, V. Y.; Bernat, A.; Borrett, W.; Mariscal, Y.; and Villa, E. 1991. Monte-Carlo methods make Dempster-Shafer formalism feasible. Technical report, NASA Johnson Space Center, Houston, TX.

Kulasekere, E. C.; Premaratne, K.; Dewasurendra, D. A.; Shyu, M.-L.; and Bauer, P. H. 2004. Conditioning and updating evidence. *International Journal of Approximate Reasoning* 36(1):75–108.

Orponen, P. 1990. Dempster's Rule of Combination is #P-complete. *Artificial Intelligence* 44(1-2):245–253.

Polpitiya, L. G.; Premaratne, K.; Murthi, M. N.; and Sarkar, D. 2016. A framework for efficient computation of belief theoretic operations. In *Proceedings of the Nineteenth International Conference on Information Fusion (FUSION)*, 1570–1577.

Polpitiya, L. G.; Premaratne, K.; Murthi, M. N.; and Sarkar, D. 2017. Efficient computation of belief theoretic conditionals. In *Proceedings of the Tenth International Symposium on Imprecise Probability: Theories and Applications (ISIPTA)*, 265–276.

ProFuSELab. 2019. FH-CBF: Fagin-Halpern conditional beliefs generated from Consonant Belief Functions. https://profuselab.github.io/FH-CBF/.

Shafer, G. 1976. *A Mathematical Theory of Evidence*. Princeton, NJ: Princeton Univ. Press.

Shafer, G. 1981. Jeffrey's Rule of Conditioning. *Philosophy of Science* 48(3):337–362.

Shafer, G. 2016. A Mathematical Theory of Evidence turns 40. *International Journal of Approximate Reasoning* 79:7–25.

Smets, P. 1991. About updating. In *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence (UAI)*, 378–385.

Smets, P. 2002. The application of the matrix calculus to belief functions. *International Journal of Approximate Reasoning* 31(1-2):1–30.

Wickramarathne, T. L.; Premaratne, K.; and Murthi, M. N. 2013. Toward efficient computation of the Dempster-Shafer belief theoretic conditionals. *IEEE Transactions on Cybernetics* 43(2):712–724.

Yager, R. R., and Liu, L., eds. 2008. *Classic Works of the Dempster-Shafer Theory of Belief Functions*. Berlin Heidelberg: Springer-Verlag.

Yu, C., and Arasta, F. 1994. On conditional belief functions. *International Journal of Approximate Reasoning* 10(2):155–172.