Efficient Computation of Conditionals in the Dempster–Shafer Belief Theoretic Framework

Lalintha G. Polpitiya, *Member, IEEE*, Kamal Premaratne, *Senior Member, IEEE*, Manohar N. Murthi, *Member, IEEE*, Stephen J. Murrell, *Member, IEEE*, and Dilip Sarkar, *Senior Member, IEEE*

***Abstract*—The Dempster–Shafer (DS) belief theory constitutes a powerful framework for modeling and reasoning with a wide variety of uncertainties due to its greater expressiveness and flexibility. As in the Bayesian probability theory, the DS theoretic (DST) conditional plays a pivotal role in DST strategies for evidence updating and fusion. However, a major limitation in employing the DST framework in practical implementations is the absence of an efficient and feasible computational framework to overcome the prohibitive computational burden DST operations entail. The work in this article addresses the pressing need for efficient DST conditional computation via the novel computational model *DS-Conditional-All.* It requires significantly less time and space complexity for computing the *Dempster's conditional* and the *Fagin–Halpern conditional,* the two most widely utilized DST conditional strategies. It also provides deeper insight into the DST conditional itself, and thus acts as a valuable tool for visualizing and analyzing the conditional computation. We provide a thorough analysis and experimental validation of the utility, efficiency, and implementation of the proposed data structure and algorithms. A new computational library, which we refer to as *DS-Conditional-One* and *DS-Conditional-All (DS-COCA)*, is developed and harnessed in the simulations.**

***Index Terms*—Algorithms, computational complexity, data structures, Dempster's conditional, Dempster–Shafer (DS) belief theory, evidential reasoning, Fagin–Halpern conditional.**

I. Introduction

The Dempster–Shafer (DS) belief theory [1]–[3], also referred to as evidence theory, is a powerful and convenient framework that offers greater expressiveness and flexibility for handling a wider variety of data imperfections [4]–[6]. With the realization of the vital role that human-generated "soft" data play in decision-making systems [7]–[9] and the importance of soft data as a complementary source of information to fuse with sensor-generated "hard" data [10]–[14], the DS-theoretic (DST) framework has become an important tool for handling uncertainty [15]–[19].

Motivation: As in the Bayesian methods [20], the conditional operation plays a fundamental role in DST strategies for evidence updating and fusion and, in general, in reasoning under uncertainty. While numerous strategies of DST conditioning have appeared in [3] and [21]–[26], perhaps the most extensively utilized DST conditional notions are the *Dempster's conditional* [3], [27]–[30] and the *Fagin–Halpern (FH) conditional* [23]. The FH conditional is closely related to the inner and outer conditional probability measures and, as suggested in recent works [12], [31], it can arguably be considered the most natural generalization of the probabilistic conditional notion.

A major criticism directed toward DST implementations is their high computational burden. Computing the DST conditionals, and even the computation of DST belief functions, are nondeterministic polynomial-time hard (NP-hard) problems [32], [33]. The development of an efficient computational framework for DST computations is of critical importance if we are to harness the strengths of the DS theory and make it more widely applicable in practice and in real-world real-time scenarios. This has indeed been identified as an issue that requires increased attention [34].

Challenges: Despite the tremendous advantages DST models offer in terms of their ability to represent a wide variety of data uncertainties, and reason and infer in their presence, DST implementations in current use are restricted to smaller frames of discernments (FoDs). While several approximation techniques to overcome the difficulty associated with computing DST quantities have been proposed [35]–[41], many of these approaches can dramatically affect the quality of approximation, and some lack the ability to be extended for computing DST conditionals. The quality of evidence updating/fusion strategies directly depends on the precision of the underlying computations. Thus, making exact (or precise) conditional computations feasible is of utmost importance for reasoning and inference under uncertainty. A review of existing studies on precise DST computations and their applications [27], [30], [42]–[55] reveals that more work is needed to overcome these limitations associated with DST conditional computation.

Manuscript received June 13, 2019; revised March 6, 2020; accepted August 5, 2020. This work was supported in part by the U.S. Office of Naval Research under Grant N00014-10-1-0140; in part by the U.S. National Science Foundation under Grant 1343430; and in part by the University of Miami U-LINK Initiative on Interdisciplinary Inquiry. This article was recommended by Associate Editor Q. Shen. *(Corresponding author: Kamal Premaratne.)*

Lalintha G. Polpitiya was with the Department of Electrical and Computer Engineering, University of Miami, Coral Gables, FL 33146 USA. He is now with the Department of Control Design Automation, The MathWorks Inc., Natick, MA 01760 USA (e-mail: lpolpiti@mathworks.com).

Kamal Premaratne, Manohar N. Murthi, and Stephen J. Murrell are with the Department of Electrical and Computer Engineering, University of Miami, Coral Gables, FL 33146 USA (e-mail: kamal@miami.edu; mmurthi@miami.edu; s.murrell@miami.edu).

Dilip Sarkar is with the Department of Computer Science, University of Miami, Coral Gables, FL 33146 USA (e-mail: sarkar@miami.edu).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2020.3017340

Previous Work: While the recent work in [52] proffers several DST data structures for highly efficient exact computations of the DST *measures,* it does not address the computation of DST *conditionals.* Perhaps the most thorough discussions regarding precise computation of the Dempster's conditional appear in [27] and [30], where a matrix calculus-based algorithm to compute Dempster's conditional masses is proposed. However, this approach, which employs a certain *specialization matrix,* is feasible only for smaller FoDs because it involves a $2^{|\Theta|} \times 2^{|\Theta|}$ stochastic matrix ($|\Theta|$ refers to the cardinality of the FoD Θ). Indeed, the time and space complexity of this approach are $\mathcal{O}(2^{|\Theta|} \times 2^{|\Theta|})$. So, for example, assuming 10 million computational iterations per second, it can take over 1800 CPU years for an FoD size of 30 and more than 15 CPU hours for an FoD size of 20. Moreover, this method is also not applicable for computing the FH conditional.

Regarding the FH conditional, the conditional core theorem (CCT) in [56] can be used to *identify* propositions that retain nonzero support after FH conditioning. However, the CCT offers no strategy to *compute* the corresponding conditionals. In fact, the time complexity of the CCT can approach $\mathcal{O}(2^{|\Theta|} \times 2^{|\Theta|})$, a prohibitive burden indeed for larger FoDs.

The *DS-Conditional-One* computational model in [53] can be employed to compute the conditional belief of an *arbitrary* proposition. But it does not address the computation of *all* conditional beliefs (or masses).

Contributions: Our main contribution is a novel scalable, generalized computational model, which we refer to as *DS-Conditional-All*, which offers significantly greater flexibility and computational capability for implementation of the DST update/fusion strategies. Its advantages are as follows.

1) *DS-Conditional-All* model can be utilized for efficient computation of *all* the (Dempster's or FH) conditional beliefs.
2) *DS-Conditional-All* is significantly superior to those conditional computation schemes that are currently available (including the use of the *DS-Conditional-One* model to compute each conditioned proposition one by one). By carefully reducing the number of operations being executed, the proposed approach takes significantly less time and lower space complexity. For example, as shown in Table I, our experimental results demonstrate that the average computational time taken to compute the Dempster's or FH conditional beliefs of all propositions by the *DS-Conditional-All* model is less than 22 (μs) for an FoD of size 10 ($\sim$1000 focal elements), 27 (ms) for an FoD of size 20 ($\sim$1 million focal elements), and less than 42.1 (s) for an FoD of size 30 ($\sim$1 billion focal elements). In comparison, the direct use of the *DS-Conditional-One* model [53] to compute each conditioned proposition one by one requires more than 303 (μs) for an FoD of size 10, 8.6 (s) for an FoD of size 20, and the computation becomes prohibitive for an FoD of size 30. The specialization matrix approach [27], [30] requires more than 98.4 (ms) for an FoD of size 10 and the computation becomes prohibitive for an FoD of size 20 or more.
3) *DS-Conditional-All* can be utilized as a common platform to carry out both FH and Dempster's conditional computations. This feature stands in stark contrast to existing computational implementations that apply to the computation of either the Dempster's conditional or the FH conditional.
4) An added advantage that *DS-Conditional-All* offers is that it can also be used to compute Dempster's conditional masses of all propositions. As shown in Table I, this method requires less than 14 (μs) for an FoD of size 10, less than 9.4 (ms) for an FoD of size 20, and less than 25.4 (s) for an FoD of size 30. In comparison, the use of the *DS-Conditional-One* model [53] requires more than 320 (μs) for an FoD of size 10, more than 8.6 (s) for an FoD of size 20, and the computation becomes prohibitive for an FoD of size 30 or more. The specialization matrix approach [27], [30] requires more than 98.4 (ms) for an FoD of size 10 and the computation becomes prohibitive for an FoD of size 20 or more.

A complete conditional computation library in C++ (DS-COCA) that includes all relevant software routines are available to the interested reader [57]. We believe that this computational framework and the associated implementations constitute a significant step toward closing the gap between the promise of DST methods for reasoning under uncertainty and its practical implementation. This computational framework can also serve as a tool for visualizing how conditional computations progress within conditioning and updating operations.

The remainder of this article is organized as follows. Section II provides a review of essential DST notions. Section III introduces the *DS-Conditional-All* computational model. Section IV contains algorithms for the efficient computation of DST conditionals. Section V contains comparisons and experimental results. Section VI provides the concluding remarks.

II. Preliminaries: DS Belief Theory

A. Basic Notions

In the work to follow, we only consider the case where the FoD Θ, which refers to the set of all possible mutually exclusive and exhaustive propositions [3], is finite. Thus, let $\Theta = \{\theta_0, \theta_1, \ldots, \theta_{n-1}\}$. The proposition $\{\theta_i\}$, referred to as a *singleton,* represents the lowest level of discernible information. In the DS theory, the power set 2^Θ of Θ represents all propositions of interest. 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 $B \subseteq \Theta$, that is, $A \backslash B = \{\theta_i \in \Theta \mid \theta_i \in A, \theta_i \notin B\}$, $\overline{A}$ denotes $\Theta \backslash A$, and $|A|$ denotes the cardinality of A.

In the DS theory, the *basic belief assignment (BBA)* or *mass* represents the "support" strictly allocated to a proposition.

Definition 1 (BBA or Mass Assignment): The mapping $m : 2^\Theta \mapsto [0, 1]$ is said to be a *BBA* or a *mass assignment* if

$$\sum_{A \subseteq \Theta} m(A) = 1, \quad \text{with} \quad m(\emptyset) = 0.$$

The mass of a composite proposition is free to move into its subsets (including individual singletons), which allows one to model the notion of *ignorance.* For example, complete ignorance can be modeled via the *vacuous BBA,* where $m(A) = 1$ for $A = \Theta$ and $m(A) = 0$ for $A \neq \Theta$. 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* $\mathfrak{F}$, that is, $\mathfrak{F} = \{A \subseteq \Theta \mid m(A) > 0\}$. Note that $|\mathfrak{F}|$ is the number of focal elements. The triple $\mathcal{E} = \{\Theta, \mathfrak{F}, m(\cdot)\}$ 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. On the other hand, the *plausibility* measures the extent to which a proposition is plausible, that is, the amount of belief not strictly supporting the complement of the proposition.

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

$$Bl(A) = \sum_{B \subseteq A} m(B); \quad Pl(A) = \sum_{B \cap A \neq \emptyset} m(B).$$

Note that $0 \leq Bl(A) \leq Pl(A) = 1 - Bl(\overline{A}) \leq 1\ \forall A \subseteq \Theta$. 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 [3]

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

We will use $\widehat{\mathfrak{F}}$ to identify the set of propositions that possess nonzero belief, that is, $\widehat{\mathfrak{F}} = \{A \subseteq \Theta \mid Bl(A) > 0\}$. For convenience, we will also employ the following notation:

$$\mathcal{S}(A; B) = \sum_{\substack{\emptyset \neq C \subseteq A;\\ \emptyset \neq D \subseteq B}} m(C \cup D); \quad \mathcal{T}(A; B) = \sum_{C \subseteq A} m(C \cup B). \tag{2}$$

Note that $\mathcal{S}(A; B)$ involves the masses of all those propositions that "straddle" *both* $A \subseteq \Theta$ and $B \subseteq \Theta$; it contains neither $m(A)$ nor $m(B)$. On the other hand, $\mathcal{T}(A; B)$ involves the masses of all those propositions that properly contain B and "straddle" $A \subseteq \Theta$; it contains $m(B)$.

Our computational models for computing the DST conditionals rely on several important results that we now state.

Proposition 1: Consider the BoE $\mathcal{E} = \{\Theta, \mathfrak{F}, m(\cdot)\}$ and $A \subseteq \Theta$. For $B \subseteq \Theta$, consider the mappings $\Gamma_A : 2^\Theta \mapsto [0, 1]$ and $\Pi_A : 2^\Theta \mapsto [0, 1]$, where

$$\Gamma_A(B) = \sum_{\emptyset \neq X \subseteq \overline{A}} m((A \cap B) \cup X); \quad \Pi_A(B) = \sum_{Y \subseteq (A \cap B)} \Gamma_A(Y).$$

Then, the following are true.

1) $\Gamma_A(A \cap B) = \Gamma_A(B)$ and $\Pi_A(A \cap B) = \Pi_A(B)$.
2) $\Gamma_A(\emptyset) = \Pi_A(\emptyset) = Bl(\overline{A})$.
3) $\Gamma_A(B) = \mathcal{T}(\overline{A}; A \cap B) - m(A \cap B)$.
4) $\Pi_A(B) = \Pi_A(\emptyset) + \mathcal{S}(\overline{A}; A \cap B)$.

Proof: 1)–3) follow by direct substitution. To show 4), note that

$$\Pi_A(B) = \sum_{\substack{Y \subseteq (A \cap B)\\ \emptyset \neq X \subseteq \overline{A}}} m((A \cap Y) \cup X).$$

When $Y = \emptyset$, the right-hand side yields $\Gamma_A(\emptyset) = \Pi_A(\emptyset) = Bl(\overline{A})$; else, it yields $\mathcal{S}(\overline{A}; A \cap B)$. This establishes 4). ■

Note that in light of 1), w.l.o.g., we can assume that $B \subseteq A$ and write

$$\Gamma_A(B) = \sum_{\emptyset \neq X \subseteq \overline{A}} m(B \cup X); \quad \Pi_A(B) = \sum_{Y \subseteq B} \Gamma_A(Y). \tag{3}$$

Thus, $\Gamma_A(B)$ involves the masses of all those propositions that "straddle" *both* $\overline{A}$ and *precisely* B. On the other hand, $\Pi_A(B)$ involves the masses of all those propositions that "straddle" *both* $\overline{A}$ and B (including its subsets).

B. Fagin–Halpern Conditional

As mentioned earlier, the FH conditional [23] can arguably be considered the most natural generalization of the probabilistic conditional [12], [31].

Definition 3 (Fagin–Halpern (FH) Conditional [23]): Consider BoE $\mathcal{E} = \{\Theta, \mathfrak{F}, m(\cdot)\}$ and $A \in \widehat{\mathfrak{F}}$. The *conditional belief* $Bl(B|A) : 2^\Theta \mapsto [0, 1]$ of B given the conditioning proposition A is

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

The conditional plausibility follows from $Pl(B|A) = 1 - Bl(\overline{B}|A)$. Once the conditional beliefs of all the propositions are computed, the corresponding conditional BBA $m(\cdot|A)$ can be obtained via a Möbius transform of the type in (1).

Given the proposition $A \in \widehat{\mathfrak{F}}$, the propositions that retain a nonzero mass after being conditioned by A are referred to as the *conditional focal elements*; the set of all such conditional focal elements is referred to as the *conditional core* $\mathfrak{F}_A$, that is, $\mathfrak{F}_A = \{B \subseteq A \in \widehat{\mathfrak{F}} \mid m(B|A) > 0\}$. In our work, we will exploit several previous results related to the conditional core [56], [58]. Of particular importance is the following result.

Lemma 1 [58]: Consider BoE $\mathcal{E} = \{\Theta, \mathfrak{F}, m(\cdot)\}$ and $A \in \widehat{\mathfrak{F}}$. Then, the following are true.

1) $m(B|A) = 0$ whenever $\overline{A} \cap B \neq \emptyset$.
2) $Bl(B|A)$ can be expressed as

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

Note that 1) states that FH conditioning annuls those propositions that have a nonempty intersection with the complement of the conditioning proposition. 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.

Proposition 2: Consider the BoE $\mathcal{E} = \{\Theta, \mathfrak{F}, m(\cdot)\}$ and $A \in \widehat{\mathfrak{F}}$. Then, we may express $Bl(B|A)$ as

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

Proof: This follows directly from Lemma 1-2) by using the fact that $Bl(A) = 1 - Pl(\overline{A})$. ■

C. Dempster's Conditional

Dempster's conditional is perhaps the most widely employed DST conditional notion [3].

Definition 4 (Dempster's Conditional [3]): Consider BoE $\mathcal{E} = \{\Theta, \mathfrak{F}, m(\cdot)\}$ and $A \subseteq \Theta$ s.t. $Bl(\overline{A}) \neq 1$ or, equivalently, $Pl(A) \neq 0$. The *conditional belief* $Bl(B\|A) : 2^{\Theta} \mapsto [0, 1]$ of B given the conditioning proposition A is

$$Bl(B\|A) = \frac{Bl(\overline{A} \cup B) - Bl(\overline{A})}{1 - Bl(\overline{A})}.$$

The *conditional mass* $m(B\|A) : 2^{\Theta} \mapsto [0, 1]$ of B given the conditioning proposition A is

$$m(B\|A) = \begin{cases} \dfrac{\sum_{C \subseteq \overline{A}} m(B \cup C)}{1 - Bl(\overline{A})}, & \text{for } \emptyset \neq B \subseteq A; \\ 0, & \text{otherwise.} \end{cases}$$

The conditional plausibility follows from $Pl(B\|A) = 1 - Bl(\overline{B}\|A)$. Similar to FH conditioning, Dempster's conditioning also annuls the masses of all those propositions that have a nonempty intersection with the complement of the conditioning. So w.l.o.g., one may again consider only those propositions $B \subseteq A$.

For our work, we will need the following alternate expression for Dempster's conditional.

Proposition 3: Consider BoE $\mathcal{E} = \{\Theta, \mathfrak{F}, m(\cdot)\}$ and $A \subseteq \Theta$ s.t. $Bl(\overline{A}) \neq 1$. Then, $Bl(B\|A)$ can be expressed as

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

Proof: This follows directly from Definition 4 by using the fact that $Bl(\overline{A} \cup B) = Bl(\overline{A} \cup (A \cap B)) = Bl(\overline{A}) + Bl(A \cap B) + \mathcal{S}(\overline{A}; A \cap B)$. ■

Propositions 2 and 3 highlight an important fact: the three quantities $Bl(\overline{A})$, $Bl(A \cap B)$, and $\mathcal{S}(\overline{A}; A \cap B)$ fully determine both FH and Dempster's conditionals $Bl(B|A)$ and $Bl(B\|A)$, respectively. It is this observation that we exploit in our computational model.

Direct substitution yields the following result.

Proposition 4: Consider BoE $\mathcal{E} = \{\Theta, \mathfrak{F}, m(\cdot)\}$ and $A \subseteq \Theta$ s.t. $Bl(\overline{A}) \neq 1$. Then, $m(B\|A)$ can be expressed as

$$m(B\|A) = \begin{cases} \dfrac{\mathcal{T}(\overline{A}; A \cap B)}{1 - Bl(\overline{A})}, & \text{for } \emptyset \neq B \subseteq A; \\ 0, & \text{otherwise.} \end{cases}$$

D. REGAP Procedure

The work in [52] proposes new data structures—*DS-Vector, DS-Matrix, and DS-Tree*—and computationally efficient algorithms for computing the basic DST operations of belief and plausibility. For this purpose, the authors utilize what is referred to as the *REGAP* (*recursive generation of and access to propositions*) procedure.

To explain, consider the FoD $\Theta = \{\theta_0, \theta_1, \ldots, \theta_{n-1}\}$. Suppose we desire to determine the belief potential $Bl(A)$ associated with $A = \{\theta_{k_0}, \theta_{k_1}, \ldots, \theta_{k_{|A|-1}}\} \subseteq \Theta$. Then, *REGAP*$(A)$ recursively generates all the $2^{|A|}-1$ propositions whose masses are required to compute $Bl(A)$, *viz.*, all subsets of A (including A itself). We will refer to the subsets of A and A itself as *subset propositions* of A. *REGAP*(A) is implemented in the following manner. Start with $\{\emptyset\}$. First, insert the singleton $\{\theta_{k_0}\} \in A$. Only one proposition is associated with this singleton, *viz.*, $\{\emptyset\} \cup \{\theta_{k_0}\} = \{\theta_{k_0}\}$ itself. Next, insert another singleton $\{\theta_{k_1}\} \in A$. The new propositions that are associated with this singleton are $\{\emptyset\} \cup \{\theta_{k_1}\} = \{\theta_{k_1}\}$ and $\{\theta_{k_0}\} \cup \{\theta_{k_1}\} = \{\theta_{k_0}, \theta_{k_1}\}$. Inserting the next singleton $\{\theta_{k_2}\} \in A$ brings the new propositions $\{\emptyset\} \cup \{\theta_{k_2}\} = \{\theta_{k_2}\}$, $\{\theta_{k_0}\} \cup \{\theta_{k_2}\} = \{\theta_{k_0}, \theta_{k_2}\}$, $\{\theta_{k_1}\} \cup \{\theta_{k_2}\} = \{\theta_{k_1}, \theta_{k_2}\}$, and $\{\theta_{k_0}, \theta_{k_1}\} \cup \{\theta_{k_2}\} = \{\theta_{k_0}, \theta_{k_1}, \theta_{k_2}\}$. In essence, when a new singleton is added, new propositions associated with it can be recursively generated by adding the new singleton to each existing proposition. Of course, all propositions of interest within the FoD Θ can be generated by *REGAP*(Θ).

The propositions recursively generated via the REGAP procedure can be represented as a vector, *DS-Vector,* a matrix, *DS-Matrix,* or a tree, *DS-Tree,* and utilized to capture a BoE. We will utilize this REGAP procedure and the DS-Vector and DS-Matrix structures in this work.

III. DS-Conditional-All Computational Model

In this section, we develop our *DS-Conditional-All* computational model. It is important to distinguish between the *DS-Conditional-One* computational model in [53] and the *DS-Conditional-All* model that is being developed below. The *DS-Conditional-One* model can be used to compute the conditional belief of a *specific arbitrary* proposition; on the other hand, the *DS-Conditional-All* model can be used to compute the FH and Dempster's conditional beliefs of *all* propositions. As it turns out, it can also be used to compute Dempster's conditional masses of *all* propositions. As we will presently show, the *DS-Conditional-All* model incorporates strategies that facilitates the representation, access, and efficient computation of the quantities that are needed to compute these conditionals (see Propositions 1–4).

Henceforth, we will denote the conditioning proposition A, its complement $\overline{A}$, and the conditioned proposition B as $\{a_0, a_1, \ldots, a_{|A|-1}\}$, $\{\alpha_0, \alpha_1, \ldots, \alpha_{|\overline{A}|-1}\}$, and $\{b_0, b_1, \ldots, b_{|B|-1}\}$, respectively. Here, $\Theta = \{\theta_0, \theta_1, \ldots, \theta_{n-1}\}$ denotes the FoD and $a_i, \alpha_j, b_k \in \Theta$, that is, they are singletons. When dealing with FH and Dempster's conditioning, it is implicitly assumed that $A \in \widehat{\mathfrak{F}}$ and $Bl(\overline{A}) \neq 1$, respectively.

We will represent singletons of the conditioning proposition $A = \{a_0, a_1, \ldots, a_{|A|-1}\}$ as *column singletons* and singletons of the complement of conditioning proposition $\overline{A} = \{\alpha_0, \alpha_1, \ldots, \alpha_{|\overline{A}|-1}\}$ as *row singletons* in a DS-Matrix. See segment 1 in Fig. 1. Note that this DS-Matrix is of size $(2^{|A|} \times 2^{|\overline{A}|})$, meaning that it has $2^{|\Theta|}$ number of entries.

The proposed *DS-Conditional-All* computational model allows direct identification of *REGAP*(A), *REGAP*$(\overline{A})$, (*REGAP*$(\overline{A})\times$*REGAP*(A)), (*REGAP*$(\overline{A})\times B$), $\Gamma_A(B)\ \forall B \subseteq A$, and $\Pi_A(B)\ \forall B \subseteq A$. Among these, the following four quantities are required to compute both FH and Dempster's conditional beliefs of all propositions, as well as to compute

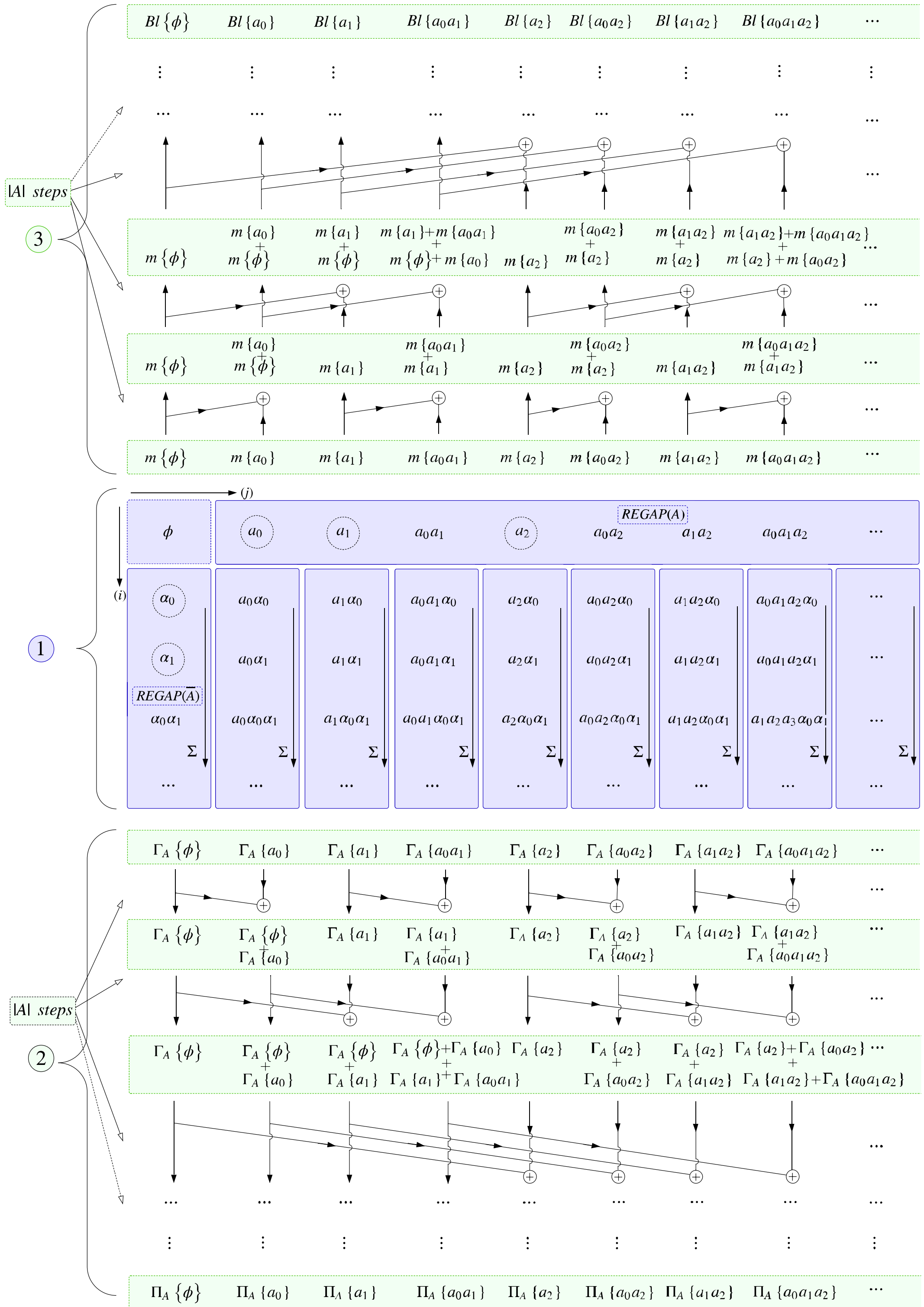

Fig. 1. DS-Conditional-All model. Quantities related to $Bl(B|A)$ [or $Bl(B\|A)$] computation for *all* $B \subseteq A$ when $A = \{a_0, a_1, \ldots, a_{|A|-1}\}$ and $\overline{A} = \{\alpha_0, \alpha_1, \ldots, \alpha_{|\overline{A}|-1}\}$. Segment ①, which employs a $(2^{|A|} \times 2^{|\overline{A}|})$-sized DS-Matrix data structure, illustrates the computation of $\Gamma_A(B)\ \forall B \subseteq A$; segment ② illustrates the computation of $\Pi_A(B)\ \forall B \subseteq A$; and segment ③ illustrates the computation of $Bl(B)$, $B \subseteq A$.

Dempster's conditional masses of all propositions (see Propositions 1–4).

1) *REGAP($\overline{A}$):* Use this to compute $Bl(\overline{A})$ which is $\Gamma_A(\emptyset)$ or $\Pi_A(\emptyset)$ ($\Gamma_A(\emptyset)$ can be obtained from the output of Procedure 1).

2) *(REGAP($\overline{A}$)×B) $\forall B \subseteq A$:* Add the BBA of each column except the BBA of the top element to compute $\Gamma_A(B)\ \forall B \subseteq A$. Segment ① of Fig. 1 shows this computation (see Procedure 1).

Procedure 1 Compute all Γ_A ($\mathcal{O}(2^{|\Theta|})$)

1: **procedure** ALLΓ_A(Singletons $\overline{A}$, Singletons A, DS-Matrix $BBA[\,][\,]$)
2: **for** $j \leftarrow 0,\ power[|A|] - 1$ **do**
3: $\Gamma_A[j] \leftarrow 0$
4: **end for**
5: **for** $j \leftarrow 0,\ power[|A|] - 1$ **do**
6: **for** $i \leftarrow 1,\ power[|\overline{A}|] - 1$ **do**
7: $\Gamma_A[j] \leftarrow \Gamma_A[j] + BBA[i][j]$
8: **end for**
9: **end for**
10: Return $\Gamma_A[\,]$
11: **end procedure**

Procedure 2 Compute all Π_A ($\mathcal{O}(2^{|A|} \times |A|)$)

1: **procedure** ALLΠ_A(Singletons A, DS-Vector $\Gamma_A[\,]$)
2: **for** $j \leftarrow 0,\ power[|A|] - 1$ **do**
3: $\Pi_A[j] \leftarrow \Gamma_A[j]$
4: **end for**
5: **for** $J \leftarrow 0,\ |A| - 1$ **do**
6: **for** $t \leftarrow 0,\ power[|A| - J] - 2$ step 2 **do**
7: **for** $s \leftarrow 0,\ power|J| - 1$ **do**
8: $\Pi_A[(t+1) * power[J] + s] \leftarrow \Pi_A[(t+1) * power[J] + s] + \Pi_A[t * power[J] + s]$
9: **end for**
10: **end for**
11: **end for**
12: Return $\Pi_A[\,]$
13: **end procedure**

3) Use *REGAP*(A) to identify the propositions relevant for $\Pi_A(B)\ \forall B \subseteq A$, computation and apply the fast Möbius transform (FMT) to obtain the $\Pi_A(\cdot)$ values from $\Gamma_A(B)\ \forall B \subseteq A$. Segment ② of Fig. 1 illustrates this transformation (see Procedure 2).
4) Use *REGAP*(A) and apply the FMT to obtain the belief values $Bl(B)$, $B \subseteq A$, from the BBA $m(B)\ \forall B \subseteq A$. Segment ③ of Fig. 1 illustrates this transformation (steps of this computation are similar to Procedure 2).

Fig. 1 depicts these quantities related to the computation of $Bl(B|A)$ [or $Bl(B\|A)$] for *all* $B \subseteq A$.

Note that segments ② and ③ of Fig. 1, which are employed within 3) and 4) above, utilize the FMT in [42], [43], [45], and [46]. The FMT, which is analogous to the fast Fourier transform (FFT), allows one to convert between a BBA and its corresponding belief potential [as in Definition 2 and (1)]. The functional relationship between $\Gamma(\cdot)$ and $\Pi(\cdot)$ (see Proposition 1) and the relationship between a BBA and its corresponding belief (see Definition 2) are the same, and hence, this same FMT can be used to convert between $\Gamma(\cdot)$ and $\Pi(\cdot)$.

In the procedures to follow, we use a lookup table called *power* to enhance computational efficiency. It contains 2 to the power of singleton indexes in increasing order and it is implemented using a dynamic array that replaces runtime computation of 2 to the power values with a simpler array indexing operation. The Jth entry of the *power* table, $power[J]$, refers to 2^J.

Time Complexity of Procedure 1: This computes all Γ_A in $\mathcal{O}(2^{|\Theta|})$ complexity.

Line #1: The procedure inputs are the complement of conditioning proposition $\overline{A}$, conditioning proposition A, and the DS-Matrix $BBA[\,][\,]$.

Lines #2–4: The required number of iterations is $(2^{|A|})$ and the complexity of this segment is $\mathcal{O}(2^{|A|})$.

Lines #5–9: The outer loop is executed $(2^{|A|})$ times. *Lines #6–8:* The inner loop is executed $(2^{|\overline{A}|} - 1)$ times. The complexity of an access operation is $\mathcal{O}(1)$. Thus, the time complexity of lines #5–9 is $(2^{|\overline{A}|} - 1)\,(2^{|A|}) = \mathcal{O}(2^{|\overline{A}|+|A|}) = \mathcal{O}(2^{|\Theta|})$.

Line #10: The procedure output is $\Gamma_A[\,]$.

Space Complexity of Procedure 1: The matrix in Fig. 1 (see segment ①) is of size $2^{|A|} \times 2^{|\overline{A}|}$. Hence, the space complexity associated with the above procedure is $\mathcal{O}(2^{|\Theta|})$.

Time Complexity of Procedure 2: This computes all Π_A in $\mathcal{O}(2^{|A|} \times |A|)$ complexity.

Line #1: The procedure inputs are the conditioning proposition A and the DS-Vector $\Gamma_A[\,]$.

Lines #2–4: The required number of iterations is $(2^{|A|})$ and the complexity of this segment is $\mathcal{O}(2^{|A|})$.

Lines #5–11: The outer loop is executed $(|A|)$ times.

Lines #6–10: The middle loop is executed $(2^{|A|-J-1})$ times.

Lines #7–9: The inner loop is executed (2^J) times. The complexity of an access operation is $\mathcal{O}(1)$. Thus, the time complexity of lines #5–11 is $(|A|)(2^{|A|-J-1})\,(2^J) = \mathcal{O}(2^{|A|} \times |A|)$.

Line #12: The procedure output is $\Pi_A[\,]$.

Space Complexity of Procedure 2: The $\Gamma_A[\,]$ and $\Pi_A[\,]$ vectors in Fig. 1 are of size $2^{|A|}$. Hence, the space complexity associated with the above procedure is $\mathcal{O}(2^{|A|})$.

Note that in the *DS-Conditional-All* model, *REGAP*(A) captures all propositions that may contribute to the conditional core $\mathfrak{F}_A$. *REGAP*($\overline{A}$) and (*REGAP*($\overline{A}$)$\times$*REGAP*(A)), which is the Cartesian product of *REGAP*($\overline{A}$) and *REGAP*(A), capture all propositions whose masses are annulled (as identified by Lemma 1). See Fig. 1.

IV. Efficient Computation of DST Conditionals

In this section, we discuss efficient computation of DST conditionals using the *DS-Conditional-All* computational model (see Fig. 1), which can be employed to compute both the FH and Dempster's conditional beliefs and Dempster's conditional masses of *all* propositions. To obtain the FH conditional masses, one may apply the FMT to the FH conditional beliefs that have been computed.

A. Algorithm 1: FH Conditional Beliefs of All Propositions

The *DS-Conditional-All* model provides $\Pi_A(A \cap B)$ and $Bl(A \cap B)$ for all $B \subseteq A$ via Procedures 1 and 2. Now, one may use these in Propositions 1 and 2 to obtain $Bl(B|A)$ as

$$Bl(B|A) = \frac{Bl(A \cap B)}{1 - \Pi_A(A \cap B)}. \tag{4}$$

We may compute all conditional belief values by iterating on the propositions in *REGAP*(A) and applying the above equation. The time complexity is $\mathcal{O}(\max\{2^{|A|} \times |A|, 2^{|\Theta|}\})$.

B. Algorithm 2: Dempster's Conditional Beliefs of All Propositions

As before, we can obtain $\Pi_A(A \cap B)$ and $Bl(A \cap B)$ for all $B \subseteq A$ via Procedures 1 and 2. Then, use the expressions in Propositions 1 and 3 to obtain $Bl(B\|A)$ as

$$Bl(B\|A) = \frac{Bl(A \cap B) + \Pi_A(A \cap B) - \Gamma_A(\{\emptyset\})}{1 - \Gamma_A(\{\emptyset\})}. \tag{5}$$

We can compute all conditional belief values by iterating on the propositions in *REGAP*(A) and applying the above equation. The time complexity is $\mathcal{O}(\max\{2^{|A|} \times |A|, 2^{|\Theta|}\})$.

C. Algorithm 3: FH (or Dempster's) Conditional Masses of All Propositions With the FMT

FH (or Dempster's) conditional masses of *all* propositions can be obtained by applying the FMT to the FH (or Dempster's) conditional beliefs that have been computed via Algorithm 1 (or Algorithm 2) above. The time complexity is $\mathcal{O}(\max\{2^{|A|} \times |A|, 2^{|\Theta|}\})$.

D. Algorithm 4: Dempster's Conditional Masses of All Propositions With No Recourse to the FMT

We can obtain $\Gamma_A(A \cap B)$ for all $B \subseteq A$ via Procedure 1 and then use the expressions in Propositions 1 and 4 to obtain

$$m(B\|A) = \frac{m(A \cap B) + \Gamma_A(A \cap B)}{1 - \Gamma_A(\{\emptyset\})}. \tag{6}$$

All conditional mass values are computed by iterating on the propositions in *REGAP*(A) and applying (6). The time complexity of iterating over *REGAP*(A) is $\mathcal{O}(2^{|A|})$. $m(A \cap B)$ can be accessed in $\mathcal{O}(1)$. Also, after applying Procedure 1, we can access $\Gamma_A(\{\emptyset\})$ in $\mathcal{O}(1)$. Thus, the time complexity of computing all conditional masses is $\mathcal{O}(2^{|\Theta|})$.

It is noteworthy that the work in [27] and [30] offers a matrix calculus-based algorithm for *exact* computation of Dempster's conditional *masses.* It employs a $2^{|\Theta|} \times 2^{|\Theta|}$-sized stochastic matrix $\mathfrak{S}_A$ (with each entry "0" or "1") referred to as the *conditioning specialization matrix* and a $2^{|\Theta|} \times 1$-sized vector $m(\cdot)$ containing the BoE's focal elements. Then, $m(\cdot\|A) = \mathfrak{S}_A \cdot m(\cdot)$ yields Dempster's conditioning masses *without normalization.* The time and space complexity of the specialization matrix multiplication is $\mathcal{O}(2^{|\Theta|} \times 2^{|\Theta|})$, a prohibitive burden even for modest FoD sizes.

V. Experiments

We now report the results of our experiments that were conducted to verify the performance of the *DS-Conditional-All* computational model.

Procedure: Note that Procedures 1 and 2 yield all the parameters [*viz.*, $Bl(A \cap B)$, $\Gamma_A(\{\emptyset\})$, and $\Pi_A(A \cap B)$] required for both FH and Dempster's conditional belief computations of *all* propositions. Once these quantities are computed, computational times for both conditional belief computations (FH and Dempster's) are similar because the final operations require only a constant time (see Propositions 1–3).

With the *DS-Conditional-All* model, we use Algorithms 1 and 2 to compute all the conditional beliefs; this applies to *both* FH and Dempster's conditionals. We next use Algorithm 3, which employs the FMT to obtain the conditional masses for all the propositions. We also use Algorithm 4 to compute all Dempster's conditional masses directly, without FMT.

With the *DS-Conditional-One* model, which again applies to *both* FH and Dempster's conditionals, we first use a "brute force" approach to compute all the conditional beliefs one by one. We next use FMT to obtain the conditional masses for all the propositions.

The specialization matrix-based method, which applies to Dempster's conditional *only*, yields the conditional masses of *all* propositions, but the time taken already far exceeds that taken by the *DS-Conditional-One* and *DS-Conditional-All* models (even including the FMT). So we did not compute the conditional beliefs with the specialization matrix-based method (which would have required FMT).

Average CPU Times: Table I lists the average computational times taken by the *DS-Conditional-All* model, direct use of the *DS-Conditional-One* model [53], and the specialization matrix-based method in [27] and [30]. The results reported are the average CPU times obtained by executing the algorithms for 10 000 randomly chosen conditioning (A) and conditioned ($B \subseteq A$) propositions from FoD. A random set of focal elements was generated in the core for each FoD size.

All conditional computations for smaller FoDs were simulated on a Macintosh desktop computer (iMac) running Mac OS X 10.14.3 (with 2.9-GHz Intel Core i5 processor and 8 GB of 1600-MHz DDR3 RAM). Conditional computations for larger FoDs were done on a supercomputer (https://ccs.miami.edu/pegasus) (underlined in Table I).

The significant speed advantage offered by the proposed computational model over the specialization matrix-based approach and *DS-Conditional-One* model is evident from Table I. For larger FoDs, the computational burden associated with the specialization matrix-based approach becomes prohibitive because of its $\mathcal{O}(2^{|\Theta|} \times 2^{|\Theta|})$ time and space complexities. For example, an FoD of size 22 would need 2 TB ($= 2^{22} \times 2^{22}/8$ bytes) of memory to represent the specialization matrix, if each matrix entry occupies only 1 bit. This is a prohibitive space requirement for practical applications. For large FoDs, the computational time requirement of the specialization matrix-based approach rapidly becomes infeasible (see Table I). With increasing FoD size, the computational time requirements of the *DS-Conditional-All* and *DS-Conditional-One* models are significantly less compared to what the specialization matrix-based approach requires.

Bounds on the CPU Times: The time complexities of the *DS-Conditional-All* and specialization matrix method are $\mathcal{O}(\max\{2^{|A|} \times |A|, 2^{|\Theta|}\})$ and $\mathcal{O}(2^{|\Theta|} \times 2^{|\Theta|})$, respectively (see Section III). We can bound the CPU times associated with these two methods as follows.

1) *DS-Conditional-All:* The number of operations required for segments ①, ②, and ③ in Fig. 1 is $2^{|A|}(2^{|\overline{A}|} - 1)$,

TABLE I

DS-Conditional-All Model Versus *DS-Conditional-One* Model and Specialization Matrix-Based Method. Average Computational Times (ms) (*** Denotes Computations not Completed Within a Feasible Time or Space Requirement)

Method →		***DS-Conditional-All* Model**			***DS-Conditional-One* Model**		**Specialization Matrix**
Conditional →		**FH or Dempster's**		**Dempster's**	**FH or Dempster's**		**Dempster's**
		$Bl(B\|A)$ or $Bl(B\\|A)$	$m(B\|A)$ or $m(B\\|A)$	$m(B\\|A)$	$Bl(B\|A)$ or $Bl(B\\|A)$	$m(B\|A)$ or $m(B\\|A)$	$m(B\\|A)$
FoD		**Algo. 1, 2**	**Algo. 3**	**Algo. 4**	**Brute Force**	**with FMT**	
$\|\Theta\|$	**Max.** $\|\mathfrak{F}\|$	**(All)**	**(All)**	**(All)**	**(All)**	**(All)**	**(All)**
2	3	0.0011	0.0014	0.0015	0.0016	0.0024	0.0015
4	15	0.0014	0.0018	0.0020	0.0057	0.0068	0.0070
6	63	0.0026	0.0034	0.0031	0.0189	0.0208	0.0767
8	255	0.0067	0.0091	0.0054	0.0707	0.0758	1.1264
10	1,023	0.0211	0.0303	0.0135	0.3038	0.3208	98.4795
12	4,095	0.0770	0.1133	0.0427	1.5535	1.6206	1,581.8300
14	16,383	0.2950	0.4378	0.1532	15.0000	17.1429	24,847.0000
16	65,535	1.1592	1.7243	0.5814	131.8750	136.8750	396,860.0000
18	262,143	6.5901	9.2096	2.3430	1,072.2200	1,077.7800	1.7637 CPU hours
20	1,048,575	26.7221	39.0397	9.3537	8,670.0000	8,698.0000	***
22	4,194,303	112.4180	166.0070	43.5348	71,115.9000	73,942.3000	***
24	16,777,215	500.3420	689.8700	233.6080	653,268.0000	660,883.0000	***
26	67,108,863	2,239.2400	2,908.7000	1,118.9500	1.6334 CPU hours	1.6915 CPU hours	***
28	268,435,455	9,273.8100	12,406.4000	4,976.9700	***	***	***
30	1,073,741,823	42,087.2000	52,055.8000	25,354.9000	***	***	***

TABLE II

CPU Times (ms) for *DS-Conditional-All* Model (Upper Bound) Versus Specialization Matrix Method (Lower Bound). Computational Times Are Calculated Assuming 10×10^6 Computational Iterations per Second

	***DS-Conditional-All* Model**		**Specialization Matrix Method**	
Number of Elements in FoD, i.e., $\|\Theta\|$	**Number of Elements in *DS-Matrix***	**CPU Time (Upper Bound)**	**Number of Elements in *Specialization* Matrix**	**CPU Time (Lower Bound)**
10	2^{10}	1.02	2^{20}	52.42
20	2^{20}	2097.15	2^{40}	15 (hours)
30	2^{30}	3221 (s)	2^{60}	1827 (years)

$|A| \cdot 2^{|A|-1}$, and $|A| \cdot 2^{|A|-1}$, respectively. So, using the fact that $|\overline{A}| = |\Theta| - |A|$, we obtain the total number of operations as $2^{|\Theta|} - 2^{|A|} + A \cdot 2^{|A|-1}$. The maximum value of this occurs when $|A| = |\Theta|$, thus giving the following upper bound on the number of operations required for *DS-Conditional-All:*

$$\mathrm{OPS}(\mathrm{DSConditionalAll}) \leq |\Theta| \cdot 2^{|\Theta|}. \tag{7}$$

2) *Specialization Matrix Method:* This method contains three main steps in the computation. Initializing the specialization matrix with respect to the conditioning event, specialization matrix multiplication with the BoE, and normalization (if normalized BBA values are required). The specialization matrix multiplication itself requires $2^{|\Theta|} \cdot 2^{|\Theta|}/2$ number of operations, yielding the following lower bound on the number of operations required for the specialization matrix method:

$$\mathrm{OPS}(\mathrm{SpecializationMatrix}) \geq 2^{|\Theta|} \cdot 2^{|\Theta|}/2. \tag{8}$$

A comparison of these bounds appears in Table II.

VI. Conclusion

This article provides a scalable, generalized framework for the efficient and exact computation of DST conditionals. The proposed *DS-Conditional-All* model can also serve as a tool for visualization and further analysis of the conditional computation process. We believe that this computational framework constitutes an important step forward in harnessing the strengths of DST methods in practical applications.

By carefully reducing the number of operations being executed, the proposed approach takes significantly less time and space complexity when compared to other approaches for conditional computation. This reduction in the number of operations is achieved primarily by the following four improvements.

1) The matrix utilized in the *DS-Conditional-All* model is generally much smaller than the matrix used in the specialization matrix method, *viz.*, the DS-Matrix is of size $(2^{|A|} \times 2^{|\overline{A}|})$ (see segment 1 of Fig. 1), whereas the specialization matrix is of size $(2^{|\Theta|} \times 2^{|\Theta|})$.
2) Matrix multiplications are avoided. Instead, only additions, which are computationally less expensive than multiplications, are employed.
3) Algorithms avoid repetitive computations.

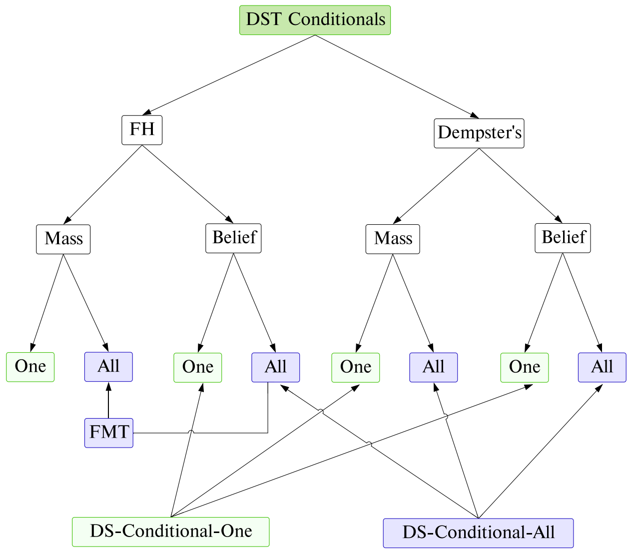

Fig. 2. Best use of *DS-Conditional-One* and *DS-Conditional-All* computational models. For example, to compute all the Dempster's conditional masses, use the *DS-Conditional-All* model; to compute all the FH conditional masses, use the *DS-Conditional-All* model to first compute all the FH conditional beliefs and then apply the FMT to obtain the conditional masses.

4) The access operation of a focal element takes constant time.

For both FH and Dempster's conditionals, this framework provides algorithms to compute all conditional beliefs in $\mathcal{O}(\max\{2^{|A|} \times |A|, 2^{|\Theta|}\})$. It also provides an algorithm to compute all Dempster's conditional masses in $\mathcal{O}(2^{|\Theta|})$. In contrast, the specialization matrix approach [30] for the Dempster's conditional requires a time complexity of $\mathcal{O}(2^{|\Theta|} \times 2^{|\Theta|})$. As an example, for an FoD of size 30 ($\sim$1 billion focal elements), the proposed framework can compute all conditional masses or beliefs within 1 min while the specialization matrix will take more than 1800 CPU years. The space complexity of the proposed algorithms is $\mathcal{O}(2^{|\Theta|})$, again a significant improvement over the prohibitive $\mathcal{O}(2^{|\Theta|} \times 2^{|\Theta|})$ space complexity of the specialization matrix approach.

Another advantage of our model is that it can be utilized for both the FH and Dempster's conditional belief computations. Fig. 2 summarizes the best use of *DS-Conditional-One* and *DS-Conditional-All* computational models.

An outcome of this research is a completely new computational library (in C++) that we refer to as *DS-COCA* [57]. It includes the implementation of required data structures and algorithms along with simulation tools for both the *DS-Conditional-One* model (developed in [53] for computing the conditional belief of an *arbitrary* proposition) and *DS-Conditional-All* model (developed in this article for computing the conditional beliefs of *all* propositions). We hope that *DS-COCA* will influence the development and practical implementation of real-time evidence fusion and uncertainty reasoning algorithms based on the DST framework.

Our current research work is focused on conditional computations on potentially dynamic FoDs (where the singletons may have to be removed or new singletons may have to be appended as operations are being carried out). From a computational perspective, the ability to work with a dynamic FoD is highly important for enhanced resource utilization. It also appears that the underlying matrix structure may allow further improvement of the algorithms that we have developed via parallel computing optimizations.

References

[1] A. P. Dempster, "Upper and lower probabilities induced by a multivalued mapping," *Ann. Math. Stat.*, vol. 38, no. 2, pp. 325–339, Apr. 1967.

[2] A. P. Dempster, "A generalization of Bayesian inference," *J. Roy. Stat. Soc. B, (Methdol.)*, vol. 30, no. 2, pp. 205–247, Jul. 1968.

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

[4] G. Shafer, "Perspectives on the theory and practice of belief functions," *Int. J. Approx. Reason.*, vol. 4, nos. 5–6, pp. 323–362, Sep.–Nov. 1990.

[5] R. R. Yager, J. Kacprzyk, and M. Fedrizzi, Eds., *Advances in the Dempster-Shafer Theory of Evidence*. New York, NY, USA: Wiley, 1994.

[6] P. Smets, "Practical uses of belief functions," in *Proc. 15th Conf. Uncertainty Artif. Intell. (UAI)*, Stockholm, Sweden, Jul. 1999, pp. 612–621.

[7] C. Laudy and B. Goujon, "Soft data analysis within a decision support system," in *Proc. 12th Int. Conf. Inf. Fusion (FUSION)*, Seattle, WA, USA, Jul. 2009, pp. 1889–1896.

[8] T. L. Wickramarathne, K. Premaratne, M. N. Murthi, and N. V. Chawla, "Convergence analysis of iterated belief revision in complex fusion environments," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 4, pp. 598–612, Aug. 2014.

[9] T. Wickramarathne, "Evidence updating for stream-processing in big-data: Robust conditioning in soft and hard data fusion environments," in *Proc. 20th Int. Conf. Inf. Fusion (FUSION)*, Xi'an, China, Jul. 2017, pp. 1–7.

[10] K. Sambhoos, J. Llinas, and E. Little, "Graphical methods for real-time fusion and estimation with soft message data," in *Proc. 11th Int. Conf. Inf. Fusion (FUSION)*, Cologne, Germany, Jun./Jul. 2008, pp. 1–8.

[11] M. A. Pravia, O. Babko-Malaya, M. K. Schneider, J. V. White, C.-Y. Chong, and A. S. Willsky, "Lessons learned in the creation of a data set for hard/soft information fusion," in *Proc. 12th Int. Conf. Inf. Fusion (FUSION)*, Seattle, WA, USA, Jul. 2009, pp. 2114–2121.

[12] T. L. Wickramarathne, K. Premaratne, M. N. Murthi, M. Scheutz, S. Kübler, and M. Pravia, "Belief theoretic methods for soft and hard data fusion," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Prague, Czechia, May 2011, pp. 2388–2391.

[13] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, "Multisensor data fusion: A review of the state-of-the-art," *Inf. Fusion*, vol. 14, no. 1, pp. 28–44, Jan. 2013.

[14] R. C. Nunez, M. N. Murthi, and K. Premaratne, "Efficient computation of DS-based uncertain logic operations and its application to hard and soft data fusion," in *Proc. 17th Int. Conf. Inf. Fusion (FUSION)*, Salamanca, Spain, Jul. 2014, pp. 1–8.

[15] R. R. Yager and L. Liu, Eds., *Classic Works of the Dempster-Shafer Theory of Belief Functions* (Studies in Fuzziness and Soft Computing). Berlin, Germany: Springer-Verlag, 2008.

[16] T. Denœux, "40 years of Dempster–Shafer theory," *Int. J. Approx. Reason.*, vol. 79, no. C, pp. 1–6, Dec. 2016.

[17] J. N. Heendeni, K. Premaratne, M. N. Murthi, J. Uscinski, and M. Scheutz, "A generalization of Bayesian inference in the Dempster–Shafer belief theoretic framework," in *Proc. 19th Int. Conf. Inf. Fusion (FUSION)*, Heidelberg, Germany, Jul. 2016, pp. 798–804.

[18] L. G. Polpitiya, "A framework for efficient implementation and effective visualization of Dempster–Shafer belief theoretic computations for reasoning under uncertainty," Ph.D. dissertation, Dept. Elect. Comp. Eng., Univ. Miami, Coral Gables, FL, USA, Aug. 2019.

[19] L. G. Polpitiya and K. Premaratne, "Real-time detection and prediction of relative motion of moving objects in autonomous driving," in *Proc. 33rd Int. Florida Artif. Intell. Res. Soc. Conf. (FLAIRS)*, Miami, FL, USA, May 2020, pp. 136–141.

[20] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA, USA: Morgan Kaufmann, 1988.

[21] G. Shafer, "Jeffrey's rule of conditioning," *Philos. Sci.*, vol. 48, no. 3, pp. 337–362, Sep. 1981.

[22] H. Ichihashi and H. Tanaka, "Jeffrey-like rules of conditioning for the Dempster–Shafer theory of evidence," *Int. J. Approx. Reason.*, vol. 3, no. 2, pp. 143–156, Mar. 1989.

[23] R. Fagin and J. Y. Halpern, "A new approach to updating beliefs," in *Proc. 6th Conf. Uncertainty Artif. Intell. (UAI)*, Cambridge, MA, USA, Jul. 1990, pp. 347–374.
[24] P. Smets, "About updating," in *Proc. 7th Conf. Uncertainty Artif. Intell. (UAI)*, Los Angeles, CA, USA, Jul. 1991, pp. 378–385.
[25] C. Yu and F. Arasta, "On conditional belief functions," *Int. J. Approx. Reason.*, vol. 10, no. 2, pp. 155–172, Feb. 1994.
[26] R. Meester and T. Kerkvliet, "A new look at conditional probability with belief functions," *Statistica Neerlandica*, vol. 73, no. 2, pp. 274–291, May 2019.
[27] F. Klawonn and P. Smets, "The dynamic of belief in the transferable belief model and specialization-generalization matrices," in *Proc. 8th Conf. Uncertainty Artif. Intell. (UAI)*, Stanford, CA, USA, Jul. 1992, pp. 130–137.
[28] H. T. Nguyen and P. Smets, "On dynamics of cautious belief and conditional objects," *Int. J. Approx. Reason.*, vol. 8, no. 2, pp. 89–104, Feb. 1993.
[29] H. Xu and P. Smets, "Reasoning in evidential networks with conditional belief functions," *Int. J. Approx. Reason.*, vol. 14, nos. 2–3, pp. 155–185, Feb.–Apr. 1996.
[30] P. Smets, "The application of the matrix calculus to belief functions," *Int. J. Approx. Reason.*, vol. 31, nos. 1–2, pp. 1–30, Oct. 2002.
[31] K. Premaratne, M. Murthi, J. Zhang, M. Scheutz, and P. Bauer, "A Dempster–Shafer theoretic conditional approach to evidence updating for fusion of hard and soft data," in *Proc. 12th Int. Conf. Inf. Fusion (FUSION)*, Seattle, WA, USA, Jul. 2009, pp. 2122–2129.
[32] P. Orponen, "Dempster's rule of combination is #P-complete," *Artif. Intell.*, vol. 44, nos. 1–2, pp. 245–253, Jul. 1990.
[33] V. Y. Kreinovich, A. Bernat, W. Borrett, Y. Mariscal, and E. Villa, "Monte–Carlo methods make Dempster–Shafer formalism feasible," NASA Johnson Space Center, Houston, TX, USA, Rep. NASA-CR-192945, Jan. 1991. [Online]. Available: https://ntrs.nasa.gov/search.jsp?R=19930015951
[34] G. Shafer, "A mathematical theory of evidence turns 40," *Int. J. Approx. Reason.*, vol. 79, pp. 7–25, Dec. 2016.
[35] F. Voorbraak, "A computationally efficient approximation of Dempster–Shafer theory," *Int. J. ManMach. Stud.*, vol. 30, no. 5, pp. 525–536, May 1989.
[36] D. Dubois and H. Prade, "Consonant approximations of belief functions," *Int. J. Approx. Reason.*, vol. 4, nos. 5–6, pp. 419–449, Sep.–Nov. 1990.
[37] B. Tessem, "Approximations for efficient computation in the theory of evidence," *Artif. Intell.*, vol. 61, no. 2, pp. 315–329, Jun. 1993.
[38] M. Bauer, "Approximation algorithms and decision making in the Dempster–Shafer theory of evidence—An empirical study," *Int. J. Approx. Reason.*, vol. 17, nos. 2–3, pp. 217–237, Aug.–Oct. 1997.
[39] D. Harmanec, "Faithful approximations of belief functions," in *Proc. 15th Conf. Uncertainty Artif. Intell. (UAI)*, Stockholm, Sweden, Jul. 1999, pp. 271–278.
[40] T. Denœux, "Inner and outer approximation of belief structures using a hierarchical clustering approach," *Int. J. Uncertainty Fuzziness Knowl. Based Syst.*, vol. 9, no. 4, pp. 437–460, Aug. 2001.
[41] R. Haenni and N. Lehmann, "Resource bounded and anytime approximation of belief function computations," *Int. J. Approx. Reason.*, vol. 31, nos. 1–2, pp. 103–154, Oct. 2002.
[42] H. M. Thoma, "Factorization of belief functions," Ph.D. dissertation, Dept. Stat., Harvard Univ., Cambridge, MA, USA, 1989.
[43] R. Kennes and P. Smets, "Fast algorithms for Dempster–Shafer theory," in *Proc. 3rd Int. Conf. Inf. Process. Manag. Uncertainty Knowl. Based Syst. (IPMU)*, Paris, France, Jul. 1990, pp. 14–23.
[44] R. Kennes and P. Smets, "Computational aspects of the Mobius transform," in *Proc. 6th Conf. Uncertainty Artif. Intell. (UAI)*, Cambridge, MA, USA, Jul. 1990, pp. 401–416.
[45] H. M. Thoma, "Belief function computations," in *Conditional Logic in Expert Systems*, I. R. Goodman, M. M. Gupta, H. T. Nguyen, and G. S. Rogers, Eds. Amsterdam, The Netherlands: Elsevier, 1991, ch. 9, pp. 269–308.
[46] R. Kennes, "Computational aspects of the Mobius transformation of graphs," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, no. 2, pp. 201–223, Mar./Apr. 1992.
[47] H. Xu and R. Kennes, "Steps toward efficient implementation of Dempster–Shafer theory," in *Advances in the Dempster-Shafer Theory of Evidence*, R. R. Yager, J. Kacprzyk, and M. Fedrizzi, Eds. New York, NY, USA: Wiley, 1994, ch. 8, pp. 153–174.
[48] N. Wilson, "Algorithms for Dempster–Shafer theory," in *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, vol. 5. D. M. Gabbay and P. Smets, Eds. Dordrecht, The Netherlands: Kluwer, 2000, ch. 10, pp. 421–475.
[49] R. Haenni and N. Lehmann, "Implementing belief function computations," *Int. J. Intell. Syst.*, vol. 18, no. 1, pp. 31–49, Jan. 2003.
[50] N. Lehmann, "Fast projection of focal sets," in *Proc. 3rd Int. Conf. Intell. Sens. Sens. Netw. Inf. Process. (ISSNIP)*, Melbourne, VIC, Australia, Dec. 2007, pp. 191–196.
[51] T. Augustin, F. P. A. Coolen, G. de Cooman, and M. C. M. Troffaes, Eds., *Introduction to Imprecise Probabilities*. Chichester, U.K.: Wiley, 2014.
[52] L. G. Polpitiya, K. Premaratne, M. N. Murthi, and D. Sarkar, "A framework for efficient computation of belief theoretic operations," in *Proc. 19th Int. Conf. Inf. Fusion (FUSION)*, Heidelberg, Germany, Jul. 2016, pp. 1570–1577.
[53] L. G. Polpitiya, K. Premaratne, M. N. Murthi, and D. Sarkar, "Efficient computation of belief theoretic conditionals," in *Proc. 10th Int. Symp. Imprecise Probab. Theor. Appl. (ISIPTA)*, Lugano, Switzerland, Jul. 2017, pp. 265–276.
[54] L. G. Polpitiya, K. Premaratne, and M. N. Murthi, "Linear time and space algorithm for computing all the Fagin–Halpern conditional beliefs generated from consonant belief functions," in *Proc. 32nd Int. Florida Artif. Intell. Res. Soc. Conf. (FLAIRS)*, Sarasota, FL, USA, May 2019, pp. 482–487.
[55] (2019). *The Society for Imprecise Probability: Theories and Applications (SIPTA) Software Tools for Imprecise Probabilities*. [Online]. Available: http://www.sipta.org/index.php?id=sfw
[56] T. L. Wickramarathne, K. Premaratne, and M. N. Murthi, "Toward efficient computation of the Dempster–Shafer belief theoretic conditionals," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 712–724, Apr. 2013.
[57] L. G. Polpitiya. (2019). *DS-COCA: DS-Conditional-One and DS-Conditional-All*. [Online]. Available: https://profuselab.github.io/DS-COCA/
[58] E. C. Kulasekere, K. Premaratne, D. A. Dewasurendra, M.-L. Shyu, and P. H. Bauer, "Conditioning and updating evidence," *Int. J. Approx. Reason.*, vol. 36, no. 1, pp. 75–108, Apr. 2004.

Lalintha G. Polpitiya (Member, IEEE) received the B.Sc.Eng. (Hons.) degree in electrical engineering from the University of Moratuwa, Moratuwa, Sri Lanka, in 2011, the B.I.T. degree in information technology from the University of Colombo, Colombo, Sri Lanka, in 2013, and the Ph.D. degree in electrical and computer engineering from the University of Miami, Coral Gables, FL, USA, in 2019.

He is currently a Senior Software Engineer with The MathWorks Inc., Natick, MA, USA. His research interests include artificial intelligence, autonomous robotics, reasoning under uncertainty, Dempster–Shafer belief theory, evidence fusion, deep learning, data structures and algorithm design, computational complexity, combinatorics, and visualization.

Dr. Polpitiya was a recipient of the IEEE Computer Society Richard E. Merwin Student Award in 2014. He has served as a Student Ambassador of the IEEE Computer Society in 2015. He was a member of the IEEEXtreme 3.0 winning team and IEEEXtreme 5.0 runner-up team in 2009 and 2011, respectively. He is a member of ACM, AAAI, SIPTA, and SAE.

Kamal Premaratne (Senior Member, IEEE) received the B.Sc. degree (First-Class Hons.) in electronics and telecommunication engineering from the University of Moratuwa, Moratuwa, Sri Lanka, in 1982, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Miami, Coral Gables, FL, USA, in 1984 and 1988, respectively, under the supervision of Prof. E. I. Jury.

He is currently the Victor P. Clarke Professor with the University of Miami. His research interests include uncertainty modeling, machine learning and knowledge discovery from imperfect data, opinion and consensus dynamics in social networks, and network science and graph theory.

Dr. Premaratne was a recipient of the 1992/1993 Mather Premium; the 1999/2000 Heaviside Premium from the Institution of Electrical Engineers, London, U.K.; the Eliahu I. Jury Excellence in Research Award in 1991, 1994, and 2001; and the Johnson A. Edosomwan Researcher of the Year Award in 2014 from the College of Engineering, University of Miami. He has served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING from 1994 to 1996 and the *Journal of the Franklin Institute* from 1993 to 2005. He is a Fellow of IET (formerly, IEE).

Manohar N. Murthi (Member, IEEE) received the B.S. degree in electrical engineering and computer science from the University of California at Berkeley, Berkeley, CA, USA, in 1990, and the M.S. and Ph.D. degrees in electrical engineering from the University of California at San Diego, San Diego, CA, USA, in 1992 and 1999, respectively.

He was previously with Qualcomm, San Diego, and Global IP Sound, San Francisco, CA, USA. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, University of Miami, Coral Gables, FL, USA. His research interests are in the general areas of speech and audio signal processing, statistical signal processing, data fusion and learning, and networks.

Dr. Murthi was a recipient of the National Science Foundation CAREER Award.

Stephen J. Murrell (Member, IEEE) received the B.Sc. degree (Hons.) in mathematics and computer science from the University of Essex, Colchester, U.K., in 1981, and the D.Phil. degree in computation from Oxford University, Oxford, U.K., in 1988.

He is currently a Lecturer with the Department of Electrical and Computer Engineering, University of Miami, Coral Gables, FL, USA. His research interests are in programming languages and operating systems.

Dilip Sarkar (Senior Member, IEEE) received the B.Tech. (Hons.) degree in electronics and electrical communication engineering from the Indian Institute of Technology Kharagpur, Kharagpur, India, in 1983, the M.S. degree in computer science from the Indian Institute of Science Bangalore, Bengaluru, India, in 1984, and the Ph.D. degree in computer science from the University of Central Florida, Orlando, FL, USA, in 1988.

From 1985 to 1986, he was a Ph.D. student with Washington State University, Pullman, WA, USA. He is currently an Associate Professor of computer science with the University of Miami, Coral Gables, FL, USA. His research interests include concurrent transport protocols, parallel and distributed processing, neural networks, wireless sensors networks and their applications, and security of virtual machines. He has guided several theses and has authored numerous papers in the above areas.

Dr. Sarkar has served on the Program Committees of the IEEE International Conference on Communications, the IEEE Globecom, the IEEE International Conference on Multimedia and Expo, the International Conference on Computer Communications and Networks, and the IEEE INFOCOM for many years.