NORTH-HOLLAND

Informatics and Computer Science

Aggregate Functions over Probabilistic Data*

CHIH-SHYANG CHANG

and

ARBEE L. P. CHEN

Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan 300, Republic of China

ABSTRACT

Various extended relational data models were proposed to handle uncertain data including *possibilistic* and *probabilistic* data. Query processing involving aggregate functions over uncertain data is rarely considered. In this paper, we define a set of extended aggregate functions over probabilistic data. The time complexity of the computations for these extended aggregate functions is, in general, exponential. We develop two efficient algorithms for the computation of the *maximum* and *minimum* aggregate functions. The worst-case time complexity of the algorithms are $O(n^2)$. These algorithms can be extended to handle the *possibilistic* data. That is, our work is devoted to the accommodation of uncertain data in database systems with an elaboration on speeding up the processing efficiency of the aggregate functions.

1. INTRODUCTION

Incomplete information exists in the real world. This includes *imprecise data* and *uncertain data* [9]. Imprecise data refer to the contents of data and uncertain data refer to the degree of the truth of data [20]. The management of incomplete information has been widely discussed in various extended relational data models. Imprecise data are usually modeled by exclusive disjunctive data [8, 11, 13, 14], such as *partial values* [11]. There are two ways to represent uncertain data, which include the approach based on fuzzy set or possibility theory [19, 22, 23], and the approach based on probability theory [1, 3, 13, 21].

*This work was partially supported by the Republic of China National Science Council under Contract No. NSC 84-2213-E-007-007.

INFORMATION SCIENCES **88**, 15–45 (1996)

655 Avenue of the Americas, New York, NY 10010

0020-0255/96/$15.00
SSDI 0020-0255(95)00128-C

Partial values proposed by Grant [11] have been considered by DeMichiel [8] for resolving domain mismatch problems in heterogeneous database systems, in which an algebraic approach for operating on partial values is proposed. Lipski [14] presented a general discussion on manipulating imprecise information including partial values. Ola [16] presented an approach to processing relations containing exclusive disjunctive data. Regarding the uncertainty aspect, various kinds of fuzzy relational databases based on the possibility theory introduced by Zadeh [22] were proposed. Prade and Testemale [19] presented an extended relational algebra that can be used to deal with partial, uncertain, or fuzzy knowledge and to take vague queries into account. Zemankova and Kandel [23] considered a fuzzy relational model and query language.

Regarding the work with the probability approaches, Barbará et al. [1] presented a probabilistic relational data model and a set of operators. Cavallo and Pittarelli [3] outlined the aspects of a theory of probabilistic databases. Lee [13] extended the relational data model to capture imprecise and uncertain data and defined a set of extended operators for manipulating these data. In [21], we generalized the concept of partial values to *probabilistic partial values* by assigning a probability to each possible value in a partial value. However, none of the probability approaches discussed a way to obtain the probabilities. In [4], we proposed a mechanism based on *Jaccard's similarity coefficients* to estimate the probabilities of the possible values in a partial value from the database contents.

In practice, relational algebra or calculus is inadequate for many important applications involving statistical information or aggregates, such as the decision support systems [12]. Therefore, modern query languages like SQL [7] are equipped with some useful aggregate operators. However, most of the previous works on the manipulation of incomplete information usually discuss the extended relational algebra and ignore the extended aggregate operators. Özsoyoğlu et al. [17] studied some aggregate operators over *set-valued* attributes. Rundensteiner and Bic [20] proposed aggregate operators in possibilistic databases. In [5], we defined a set of extended aggregate functions, namely *sum*, *average*, *count*, *maximum*, and *minimum*, over partial values.

In this paper, we investigate the same set of extended aggregate functions over probabilistic partial values. We first give the *primary definitions* of these extended aggregate functions, which result in a probabilistic partial value once evaluated. By these definitions, users can have the most possible values (i.e., the possible values with the largest probabilities) as the answer. If the users prefer an approximate answer such as *expected values*, we also provide an *alternative definition* for each extended aggregate function.

The time complexity for exhaustively evaluating these extended aggregate functions (under the primary definition) is exponential. Since the cardinality of the results of the two extended aggregate functions *sum* and *average* is exponential in the worst case, no efficient algorithm can be found to evaluate them. To find the minimum number in the possible values, the *count* aggregate function can be reduced to the *minimum cover problem* in graph theory [2], which is *NP-hard* [18]. We propose efficient algorithms for *maximum* and *minimum* aggregate functions, and show their correctness and time complexity for computations. Under the same framework, the extended aggregate functions over possibilistic data [20] or partial values [5] can also be handled. Therefore, our work is devoted to the accommodation of incomplete information in database systems with an elaboration on speeding up its processing efficiency.

This paper is structured as follows. First, we present basic concepts and definitions of probabilistic partial values (Section 2). The primary and alternate definitions of the five extended aggregate functions are introduced in Section 3. The considerations for the *sum*, *average*, and *count* aggregate functions are described in Section 4. Section 5 presents two efficient algorithms for computing the *maximum* and *minimum* aggregate functions. The correctness and the worst-case time complexity for these two algorithms are also presented. With slight modifications of the two algorithms, possibilistic values can also be handled, as discussed in Section 6. We conclude the paper in Section 7.

2. BASIC CONCEPTS AND DEFINITIONS

A *probabilistic partial value* [21] is an extension of *partial values* [8]. That is, imprecise data are extended, to uncertain data, by which more informative query results can be achieved. Theoretically, we can treat the probabilistic partial values as the *probabilistic data* considered in probabilistic data models [1, 10]. In this section, we will describe the concepts of probabilistic partial values and consider the *interpretations* of a set of probabilistic partial values.

2.1. BASIC CONCEPTS OF PROBABILISTIC PARTIAL VALUES

Partial values model data imprecision in databases in the sense that, for an imprecise datum, its *true* value can be restricted in a specific set of possible values [8]. A partial value is represented by a set of *possible* values, in which exactly one of the value is *true*. In this paper, we follow the definition of partial values given in [8], which is formally stated as follows.

DEFINITION 1. A *partial value* [8], denoted $\xi = [a_1, a_2, \ldots, a_m]$, associates with m possible values, $a_1, a_2, \ldots, a_m$, $m \geq 1$, of the same domain, in which exactly one of the values in ξ is the *true* value of ξ.

For a partial value $\xi = [a_1, a_2, \ldots, a_m]$, a function ν is defined [8], where ν maps the partial value to its corresponding finite set of *possible* values; that is, $\nu(\xi) = \{a_1, a_2, \ldots, a_m\}$. Notice that an *applicable null value* [6], $\aleph$, can be considered as a partial value with $\nu(\aleph) = D$, where D is the whole domain.

DEFINITION 2. A *probabilistic partial value*, denoted $\eta = [a_1^{p_1}, a_2^{p_2}, \ldots, a_m^{p_m}]$, associates with m possible values, $a_1, a_2, \ldots, a_m$, of the same domain D, where each a_i associates with a probability $p_i > 0$ such that $\sum_{i=1}^{m} p_i = 1$.

In this paper, since we consider aggregate functions, D is assumed numerical. To facilitate our presentation, we introduce a function μ for a probabilistic partial value to obtain the probability of each possible value a_i in the probabilistic partial value $\eta = [a_1^{p_1}, a_2^{p_2}, \ldots, a_m^{p_m}]$. That is, $\mu_\eta(a_i) = p_i$, $1 \leq i \leq m$. The *cardinality* of a probabilistic partial value η, denoted $|\eta|$, is defined as the number of the possible values in η.

When a probabilistic partial value is used to represent uncertain data in a relation, its associated attribute can be regarded as a discrete random variable [1, 15]. Also, the probability of an attribute value is a conditional probability depending on the key value of the associated tuple (key values are assumed definite). To illustrate, consider the following relation, where *name* is the key attribute.

name	*city*	*specialty*	*age*
Jesse	$[T^{0.4}, H^{0.5}, K^{0.1}]$	SE	30
Annie	K	$[DB^{0.2}, CS^{0.8}]$	27

This relation describes two entities, "Jesse" and "Annie." The probability that Jesse's city is T is

$$prob(city=\text{“}T\text{”} \mid name=\text{“}Jesse\text{”}) = 0.4.$$

2.2. *ALTERNATE WORLDS OF A SET OF PROBABILISTIC PARTIAL VALUES*

For a set of probabilistic partial values Φ, we may enumerate all the possible combinations that Φ represents and compute the probability for each possible combination. A combination, called *interpretation*, represents

a case that exists in the real-world corresponding to the set of probabilistic partial values Φ.

DEFINITION 3. An *interpretation* Δ of a set of probabilistic partial values, $\Phi = \{\eta_1, \eta_2, \ldots, \eta_n\}$, is a pair $\langle \alpha, \theta \rangle$, where $\alpha = (a_1, a_2, \ldots, a_n)$ is an assignment of values from Φ such that $a_i \in \nu(\eta_i)$, $1 \leq i \leq n$, and $\theta = \mu_{\eta_1}(a_1) \times \mu_{\eta_2}(a_2) \times \cdots \times \mu_{\eta_n}(a_n)$ is the probability of the assignment. The *family* of the interpretations of Φ, denoted $\mathcal{I}(\Phi)$, consists of all interpretations of Φ. If $\Phi = \emptyset$, define $\mathcal{I}(\Phi) = \emptyset$.

In Definition 3, we call α and θ the *assignment part* and the *probability part* of the interpretation Δ, respectively. To consider the redundancies among interpretations, the *value set* of an interpretation is defined as follows.

DEFINITION 4. For an interpretation $\Delta = \langle \alpha, \theta \rangle$, $\alpha = (a_1, a_2, \ldots, a_n)$, of a set of probabilistic partial values $\Phi = \{\eta_1, \eta_2, \ldots, \eta_n\}$, the *value set* of Δ is denoted $S_\Delta = \langle S_\alpha, S_\theta \rangle$, where $S_\alpha = \{a_i \mid 1 \leq i \leq n\}$ and $S_\theta = \theta$.

Similarly, in Definition 4, we call S_α and S_θ the *assignment part* and the *probability part* of the value set S_Δ, respectively. An extended union operation, *α–union*, denoted $\widetilde{\cup}$, is defined as follows.

DEFINITION 5. Let $S_{\Delta_1} = \langle S_{\alpha_1}, S_{\theta_1} \rangle$ and $S_{\Delta_2} = \langle S_{\alpha_2}, S_{\theta_2} \rangle$ be two value sets. Then

$$\{S_{\Delta_1}\} \widetilde{\cup} \{S_{\Delta_2}\} = \begin{cases} \{\langle S_{\alpha_1}, S_{\theta_1} + S_{\theta_2} \rangle\}, & \text{if } S_{\alpha_1} = S_{\alpha_2} \\ \{S_{\Delta_1}\} \cup \{S_{\Delta_2}\}, & \text{otherwise.} \end{cases} \quad (1)$$

Definition 5 says that the α–union may merge the interpretations if they have the same assignment parts of the corresponding value sets. The probability part of the merged value set is assigned with the sum of the probability parts of S_{Δ_1} and S_{Δ_2}.

DEFINITION 6. Consider a set of probabilistic partial values $\Phi = \{\eta_1, \eta_2, \ldots, \eta_n\}$. For all interpretations $\Delta_i \in \mathcal{I}(\Phi)$, $1 \leq i \leq p$ ($p = |\eta_1| \times |\eta_2| \times \cdots \times |\eta_n|$), the *family* of the value sets of Φ is denoted $\mathcal{F}(\Phi) = \widetilde{\cup}_{i=1}^{p} \{S_{\Delta_i}\}$. If $\mathcal{I}(\Phi) = \emptyset$, define $\mathcal{F}(\Phi) = \emptyset$.

An example is given in the following to explain the above definitions.

EXAMPLE 1. Consider the relation **Person**, shown in Table 1. In this example, $\Phi = \{\overbrace{[130^{0.6}, 140^{0.4}]}^{\eta_1}, \overbrace{[120^{0.2}, 130^{0.5}, 140^{0.3}]}^{\eta_2}\}$, and the family of

TABLE 1
An Example Relation PERSON

...	*weight*	...
...	$[130^{0.6}, 140^{0.4}]$	...
...	$[120^{0.2}, 130^{0.5}, 140^{0.3}]$	...

the interpretations of Φ, $\mathcal{I}(\Phi)$, consists of the following six interpretations: $\Delta_1 = \langle(130, 120), \theta_1\rangle$; $\Delta_2 = \langle(130, 130), \theta_2\rangle$; $\Delta_3 = \langle(130, 140), \theta_3\rangle$; $\Delta_4 = \langle(140, 120), \theta_4\rangle$; $\Delta_5 = \langle(140, 130), \theta_5\rangle$; and $\Delta_6 = \langle(140, 140), \theta_6\rangle$; where $\theta_1 = 0.6 \times 0.2 = 0.12$, $\theta_2 = 0.6 \times 0.5 = 0.30$, $\theta_3 = 0.6 \times 0.3 = 0.18$, $\theta_4 = 0.4 \times 0.2 = 0.08$, $\theta_5 = 0.4 \times 0.5 = 0.20$, and $\theta_6 = 0.4 \times 0.3 = 0.12$. The corresponding value sets are

$$S_{\Delta_1} = \langle\{120, 130\}, 0.12\rangle, \quad S_{\Delta_2} = \langle\{130\}, 0.30\rangle,$$
$$S_{\Delta_3} = \langle\{130, 140\}, 0.18\rangle, \quad S_{\Delta_4} = \langle\{120, 140\}, 0.08\rangle,$$
$$S_{\Delta_5} = \langle\{130, 140\}, 0.20\rangle, \text{ and } S_{\Delta_6} = \langle\{140\}, 0.12\rangle,$$

and the family of the value sets of Φ is

$$\begin{aligned}\mathcal{F}(\Phi) &= \bigcup_{i=1}^{6} \{S_{\Delta_i}\} \\ &= \{\langle\{120, 130\}, 0.12\rangle, \langle\{130\}, 0.30\rangle, \langle\{130, 140\}, 0.38\rangle, \\ &\quad \langle\{120, 140\}, 0.08\rangle, \langle\{140\}, 0.12\rangle\}.\end{aligned}$$

Notice that S_{Δ_3} and S_{Δ_5} have been merged into one value set. ■

For all the value sets in $\mathcal{F}(\Phi)$, the sum of their probability parts equals 1. We have the following lemma.

LEMMA 1. *Let $\mathcal{F}(\Phi)$ be the family of the value sets of $\Phi = \{\eta_1, \eta_2, \ldots, \eta_n\}$, where $\eta_i = [a_{i_1}^{p_{i_1}}, a_{i_2}^{p_{i_2}}, \ldots, a_{im_i}^{p_{im_i}}]$, $1 \le i \le n$. Then, $\forall\ S_{\Delta_j} = \langle S_{\alpha_j}, S_{\theta_j}\rangle \in \mathcal{F}(\Phi)$, we have*

$$\sum_{j=1}^{|\mathcal{F}(\Phi)|} S_{\theta_j} = 1.$$

Proof.

$$\begin{aligned}\sum_{i=1}^{|\mathcal{F}(\Phi)|} S_{\theta_i} &= \sum_{i=1}^{p} \theta_i, \quad \text{where } p = |\eta_1| \times |\eta_2| \times \cdots \times |\eta_n| \\ &= \sum_{\substack{1 \le i_1 \le m_1 \\ \cdots \\ 1 \le i_n \le m_n}} \mu_{\eta_1}(a_{1i_1}) \times \mu_{\eta_2}(a_{2i_2}) \times \cdots \times \mu_{\eta_n}(a_{ni_n})\end{aligned}$$

$$
\begin{aligned}
&= \sum_{\substack{1\le i_1\le m1\\ \cdots\\ 1\le i_{n-1}\le m_{n-1}}} \mu_{\eta_1}(a_{1i_1}) \times \mu_{\eta_2}(a_{2i_2}) \times \cdots \times \mu_{\eta_{n-1}}(a_{(n-1)i_{n-1}}) \\
&\qquad \times (\mu_{\eta_n}(a_{n1}) + \mu_{\eta_n}(a_{n2}) + \cdots + \mu_{\eta_n}(a_{nm_n})) \\
&= \sum_{\substack{1\le i_1\le m1\\ \cdots\\ 1\le i_{n-1}\le m_{n-1}}} \mu_{\eta_1}(a_{1i_1}) \times \mu_{\eta_2}(a_{2i_2}) \\
&\qquad \times \cdots \times \mu_{\eta_{n-1}}(a_{(n-1)i_{n-1}}) \times 1 \\
&= \cdots \\
&= \sum_{1\le i_1\le m1} \mu_{\eta_1}(a_{1i_1}) \times 1 \times \cdots \times 1 \\
&= (\mu_{\eta_1}(a_{11}) + \mu_{\eta_1}(a_{12}) + \cdots + \mu_{\eta_1}(a_{1m_n})) \times 1 \times \cdots \times 1 \\
&= 1 \times 1 \times \cdots \times 1 \\
&= 1.
\end{aligned}
$$

□

3. AGGREGATE FUNCTIONS OVER PROBABILISTIC PARTIAL VALUES

In this section, we provide two kinds of definitions for each extended aggregate function. Section 3.1 describes the primary definitions for the extended aggregate functions *sum*, *average*, *count*, *maximum*, and *minimum*. The results of these aggregate functions are also probabilistic partial values. If the users prefer the results of these extended aggregate functions as *expected values*, we give alternate definitions for each extended aggregate function in Section 3.2. The results of these functions are approximate values and the computations are linear.

3.1. PRIMARY DEFINITIONS

According to the existence of duplicate values, we classify the set of the extended aggregate functions into two classes:

- The *duplicate value preservation class* includes *count*, *sum*, and *average* aggregate functions.
- The *duplicate value elimination class* includes *count*, *maximum*, and *minimum* aggregate functions.

Consider a set of probabilistic partial values $\Phi = \{\eta_1, \eta_2, \ldots, \eta_n\}$. We use the family of the interpretations of Φ, $\mathcal{I}(\Phi)$ to define the extended aggregate functions that are in the duplicate value preservation class. On the other hand, in order to eliminate the duplicate values, we use the family of the value sets of Φ, $\mathcal{F}(\Phi)$ to define the extended aggregate functions that are

in the duplicate value elimination class. Notice that the *count* aggregate function can be in the duplicate value preservation or elimination class. If it is considered as being in the duplicate value preservation class, the result of the *count* aggregate function over Φ is n; otherwise, its definition is shown in Definition 9.

An element of the family of the interpretations (resp. value sets) of Φ, $\mathcal{I}(\Phi)$ (resp. $\mathcal{F}(\Phi)$), describes a possible case that Φ represents. Hence, $\mathcal{I}(\Phi)$ (resp. $\mathcal{F}(\Phi)$) describes the set of all possible cases in the real world. A conventional aggregate function f, when applied to an interpretation (resp. a value set), say $\Delta = \langle \alpha, \theta \rangle$ (resp. $S_\Delta = \langle S_\alpha, S_\theta \rangle$), in $\mathcal{I}(\Phi)$ (resp. $\mathcal{F}(\Phi)$), returns a definite value, $f(\alpha)$ (resp. $f(S_\alpha)$), with a probability θ (resp. S_θ). Therefore, an extended aggregate function on Φ produces *a set of* possible values with associated probabilities. That is, it merges (i.e., *α–unions*) the aggregation results of all the interpretations (resp. value sets) in $\mathcal{I}(\Phi)$ (resp. $\mathcal{F}(\Phi)$).

DEFINITION 7. The extended *sum*, denoted $sum1(\Phi)$, is defined as

$$sum1(\Phi) \equiv [y_1^{\theta_1}, y_2^{\theta_2}, \ldots, y_{|T|}^{\theta_{|T|}}], \text{ where } \forall \langle \{y_i\}, \theta_i \rangle \in T \text{ and}$$

$$T = \widetilde{\bigcup}_{\Delta=\langle \alpha,\theta \rangle \in \mathcal{I}(\Phi)} \left\{ \left\langle \left\{ \sum_{j=1}^{n} a_j \mid \alpha = (a_1, \ldots, a_j, \ldots, a_n), \right. \right. \right.$$

$$\left. \left. \left. 1 \le j \le n \right\}, \theta \right\rangle \right\}.$$

EXAMPLE 2. Consider the relation **Person** shown in Table 1 again. We have

$$T = \{\langle \{250\}, 0.12 \rangle, \langle \{260\}, 0.30 \rangle, \langle \{270\}, 0.38 \rangle, \langle \{260\}, 0.08 \rangle, \langle \{280\}, 0.12 \rangle\}$$

for the extended aggregate function *sum*. Hence,

$$sum1(weight) = [250^{0.12}, 260^{0.38}, 270^{0.38}, 280^{0.12}]. \quad \blacksquare$$

DEFINITION 8. The extended *average*, denoted $avg1(\Phi)$, is defined as

$$avg1(\Phi) \equiv [y_1^{\theta_1}, y_2^{\theta_2}, \ldots, y_{|T|}^{\theta_{|T|}}], \text{ where } \forall \langle \{y_i\}, \theta_i \rangle \in T \text{ and}$$

$$T = \widetilde{\bigcup}_{\Delta=\langle \alpha,\theta \rangle \in \mathcal{I}(\Phi)} \left\{ \left\langle \left\{ \sum_{j=1}^{n} a_j/n \mid \alpha = (a_1, \ldots, a_j, \ldots, a_n), \right. \right. \right.$$

$$\left. \left. \left. 1 \le j \le n \right\}, \theta \right\rangle \right\}.$$

EXAMPLE 3. For the relation **Person**, we have

$$T = \{\langle\{125\}, 0.12\rangle, \langle\{130\}, 0.38\rangle, \langle\{135\}, 0.38\rangle, \langle\{140\}, 0.12\rangle\}$$

for the extended aggregate function *average*. Hence,

$$avg1(weight) = [125^{0.12}, 130^{0.38}, 135^{0.38}, 140^{0.12}].$$ ■

DEFINITION 9. The extended *count*, denoted $count1(\Phi)$, is defined as

$$count1(\Phi) \equiv \left[y_1^{\theta_1}, y_2^{\theta_2}, \ldots, y_{|T|}^{\theta_{|T|}}\right], \text{ where } \forall\ \langle\{y_i\}, \theta_i\rangle \in T \text{ and}$$
$$T = \widetilde{\bigcup}_{S_\Delta = \langle S_\alpha, S_\theta\rangle \in \mathcal{F}(\Phi)} \{\langle\{|S_\alpha|\}, S_\theta\rangle\}.$$

EXAMPLE 4. For the relation **Person**, we have

$$\begin{aligned} T &= \{\langle\{2\}, 0.12\rangle, \langle\{1\}, 0.30\rangle, \langle\{2\}, 0.38\rangle, \langle\{2\}, 0.08\rangle, \langle\{1\}, 0.12\rangle\} \\ &= \{\langle\{1\}, 0.42\rangle, \langle\{2\}, 0.58\rangle\} \end{aligned}$$

for the extended aggregate function *count*. Hence,

$$count1(weight) = [1^{0.42}, 2^{0.58}].$$ ■

DEFINITION 10. The extended *maximum*, denoted $max1(\Phi)$, is defined as

$$max1(\Phi) \equiv \left[y_1^{\theta_1}, y_2^{\theta_2}, \ldots, y_{|T|}^{\theta_{|T|}}\right], \text{ where } \forall\ \langle\{y_i\}, \theta_i\rangle \in T \text{ and}$$
$$T = \widetilde{\bigcup}_{S_\Delta = \langle S_\alpha, S_\theta\rangle \in \mathcal{F}(\Phi)} \{\langle\{\max S_\alpha\}, S_\theta\rangle\}.$$

EXAMPLE 5. For the relation **Person**, we have

$$\begin{aligned} T &= \{\langle\{130\}, 0.12\rangle, \langle\{130\}, 0.30\rangle, \langle\{140\}, 0.38\rangle, \langle\{140\}, 0.08\rangle, \langle\{140\}, 0.12\rangle\} \\ &= \{\langle\{130\}, 0.42\rangle, \langle\{140\}, 0.58\rangle\} \end{aligned}$$

for the extended aggregate function *maximum*. Hence,

$$max1(weight) = [130^{0.42}, 140^{0.58}].$$ ■

DEFINITION 11. The extended *minimum*, denoted $min1(\Phi)$, is defined as

$$min1(\Phi) \equiv \left[y_1^{\theta_1}, y_2^{\theta_2}, \ldots, y_{|T|}^{\theta_{|T|}}\right], \text{ where } \forall\ \langle\{y_i\}, \theta_i\rangle \in T \text{ and}$$
$$T = \widetilde{\bigcup}_{S_\Delta = \langle S_\alpha, S_\theta\rangle \in \mathcal{F}(\Phi)} \{\langle\{\min S_\alpha\}, S_\theta\rangle\}.$$

EXAMPLE 6. For the relation **Person**, we have

$$\begin{aligned} T &= \{\langle\{120\}, 0.12\rangle, \langle\{130\}, 0.30\rangle, \langle\{130\}, 0.38\rangle, \\ &\quad \langle\{120\}, 0.08\rangle, \langle\{140\}, 0.12\rangle\} \\ &= \{\langle\{120\}, 0.20\rangle, \langle\{130\}, 0.68\rangle, \langle\{140\}, 0.12\rangle\} \end{aligned}$$

for the extended aggregate function *minimum*. Hence,

$$min1(weight) = [120^{0.20}, 130^{0.68}, 140^{0.12}]. \quad \blacksquare$$

3.2. ALTERNATE DEFINITIONS

There are two reasons to provide an alternate definition for each extended aggregate function. First, users may prefer an approximate but definite answer for the extended aggregate functions. Second, the computation of alternate definitions is linear. For a probabilistic partial value, we can use an *expected value* to approximate it. The expected value of a probabilistic partial value is defined as follows.

DEFINITION 12. For a probabilistic partial value, $\eta = [a_1^{p_1}, a_2^{p_2}, \ldots, a_m^{p_m}]$, the *expected value*, denoted $\bar{\eta}$, of η is defined as

$$\bar{\eta} = \sum_{i=1}^{m} a_i \times p_i.$$

For a set of probabilistic partial values Φ, we first compute the expected values of the corresponding probabilistic partial values in Φ. Then, the alternate definition of each extended aggregate function is defined on the set of expected values.

DEFINITION 13. Let $\Phi = \{\eta_1, \eta_2, \ldots, \eta_n\}$, where $\eta_i = [a_{i1}^{p_{i1}}, a_{i2}^{p_{i2}}, \ldots, a_{im_i}^{p_{im_i}}]$, $1 \le i \le n$, be a set of probabilistic partial values, and $\bar{\eta}_i$ be the expected value of η_i, $1 \le i \le n$. The alternate definitions of *count*, *sum*, *average*, *maximum*, and *minimum*, denoted $count2(\Phi)$, $sum2(\Phi)$, $avg2(\Phi)$, $max2(\Phi)$, and $min2(\Phi)$, respectively, are defined as follows.

$$\begin{aligned} count2(\Phi) &= n, \\ sum2(\Phi) &= \sum_{i=1}^{n} \bar{\eta}_i, \\ avg2(\Phi) &= \frac{sum2(\Phi)}{n}, \\ max2(\Phi) &= \max_{i=1}^{n} \bar{\eta}_i, \text{ and} \\ min2(\Phi) &= \min_{i=1}^{n} \bar{\eta}_i. \end{aligned}$$

EXAMPLE 7. Consider the **Person** relation shown in Table 1 again. The alternate definitions of the extended aggregate functions are evaluated below.

$$\begin{aligned}
\bar{\eta_1} &= 130 \times 0.6 + 140 \times 0.4 = 134, \\
\bar{\eta_2} &= 120 \times 0.2 + 130 \times 0.5 + 140 \times 0.3 = 131. \\
count2(weight) &= 2, \\
sum2(weight) &= \bar{\eta_1} + \bar{\eta_2} = 265, \\
avg2(weight) &= 265/2 = 132.5, \\
max2(weight) &= \max\{134, 131\} = 134, \text{ and} \\
min2(weight) &= \min\{134, 131\} = 131.
\end{aligned}$$

■

4. CONSIDERATIONS FOR *count*1, *sum*1, AND *avg*1 AGGREGATE FUNCTIONS

*4.1. sum*1 *AND avg*1

When $|\Phi| = n$ and $|\eta_i| = m$, $1 \leq i \leq n$, there are m^n interpretations of Φ. Since the cardinalities of $\nu(sum1(\Phi))$ and $\nu(avg1(\Phi))$ are equal to m^n in the worst case (see Example 1), no polynomial time algorithms can be found for *sum*1 and *avg*1.

*4.2. count*1

Although the cardinality of $\nu(count1(\Phi))$ is $O(n)$ in the worst case, to find the minimum number in $\nu(count1(\Phi))$ is difficult. In this section, we use techniques in graph theory [2] to consider $\nu(count1(\Phi))$.

DEFINITION 14. For a set S of vertices in a graph $G = (V, E)$, $S \subseteq V$, the *neighbor set* of S, denoted $N(S)$, is defined to be the set of all vertices adjacent to the vertices in S.

DEFINITION 15. For a bipartite graph $G = (X \cup Y, E)$, we say a set S, $S \subseteq X$, *covers* Y if $N(S) = Y$.

DEFINITION 16. For a set of probabilistic partial values $Y = \Phi = \{\eta_1, \eta_2, \ldots, \eta_n\}$, let $X = \cup_{i=1}^{n} \nu(\eta_i) = \{a_1, a_2, \ldots, a_q\}$. The *membership graph* of Y over X is a bipartite graph $G = (X \cup Y, E)$, where

$$E = \{(a_i, \eta_j) \mid a_i \in \nu(\eta_j), 1 \leq i \leq q, 1 \leq j \leq n\}.$$

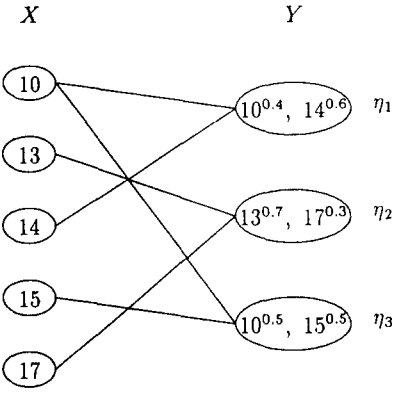

Fig. 1. The membership graph of Y over X.

EXAMPLE 8. Consider a set $Y = \Phi = \{\eta_1, \eta_2, \eta_3\}$, where $\eta_1 = [10^{0.4}, 14^{0.6}]$, $\eta_2 = [13^{0.7}, 17^{0.3}]$ and $\eta_3 = [10^{0.5}, 15^{0.5}]$. Then, $X = \{10, 13, 14, 15, 17\}$. The membership graph of Y over X is shown in Figure 1. ■

THEOREM 1. *Let $Y = \Phi = \{\eta_1, \eta_2, \ldots, \eta_n\}$, $X = \cup_{i=1}^{n} \nu(\eta_i)$ and $G = (X \cup Y, E)$ be the membership graph of Y over X. Let l denote the cardinality of the minimum set S, $S \subseteq X$, that covers Y. Then, for all $c \in \nu(count1(\Phi))$, $c \geq \ell$.*

Proof. Suppose there is an element c of $\nu(count1(\Phi))$ such that $c < l$. That is, there is an interpretation $\alpha = (a_{i_1}, a_{i_2}, \ldots, a_{i_n})$, $a_{i_j} \in \nu(\eta_j), 1 \leq j \leq n$, of Φ with $|S_\alpha| = c < l$, where $S_\alpha = \{a_{i_j} \mid 1 \leq j \leq n\}$. Since $a_{i_j} \in \nu(\eta_j)$, for $1 \leq j \leq n$, S_α covers Y (i.e., $N(S_\alpha) = Y$). But $|S_\alpha| < l$, which contradicts the assumption that the cardinality of the minimum set that covers Y is l. Hence, for all $c \in \nu(count1(\Phi))$, $c \geq l$. □

The *minimum cover problem* [18] on a family F is described as follows:

> Given a family $F = \{A_1, A_2, \ldots, A_n\}$ of subsets of a finite set U. Find a subfamily C of F such that $\cup_{A_j \in C} A_j = U$ and there does not exist another subfamily C', $|C'| < |C|$, such that $\cup_{A_j \in C'} A_j = U$.

We want to reduce the problem to find l defined in Theorem 1 to the minimum cover problem.

THEOREM 2. *Let $Y = \Phi = \{\eta_1, \eta_2, \ldots, \eta_n\}$, $X = \cup_{i=1}^{n} \nu(\eta_i)$ and $G = (X \cup Y, E)$ be the membership graph of Y over X. Let ℓ denote the cardinality of the minimum set S, $S \subseteq X$ that covers Y. To find the number l is a minimum cover problem.*

Proof. Consider the membership graph of Y over X, $G = (X \cup Y, E)$. Let $U = Y$, $F = X$, and $C = S$ in the description of the minimum cover problem, the proof follows. □

From Theorem 2, we know that one can find l using an algorithm with polynomial time if and only if one can solve the minimum cover problem. However, the minimum cover problem have been proved as an *NP-hard* problem [18]. Therefore, we cannot find an efficient algorithm for $\nu(count1(\Phi))$. Further, we cannot find an efficient algorithm for $count1(\Phi)$ either.

To sum up, we have no efficient algorithms for the extended aggregate functions *sum*1, *avg*1, and *count*1. To compute *sum*1, *avg*1, and *count*1 on a set of probabilistic partial values Φ, one can generate all the interpretations of Φ based on the definitions shown in Definition 7, Definition 8, and Definition 9, respectively.

5. ALGORITHMS FOR *max*1 AND *min*1 AGGREGATE FUNCTIONS

To evaluate the extended aggregate functions on a set of probabilistic partial values Φ based on the primary definitions is time consuming. If we generate all interpretations of Φ for the evaluation, the time complexity will be exponential. In Section 5.1, we provide a dual efficient algorithms for *max*1 and *min*1 aggregate functions. Section 5.2 shows the correctness of the dual efficient algorithms. The worst-case time complexity for the dual algorithms are also provided in Section 5.3.

5.1. ALGORITHMS

From Definition 10, $max1(\Phi)$[1] is a probabilistic partial values. In order to compute $max1(\Phi)$ efficiently, we divide the computation algorithm into two phases, *value phase* and *probability phase.*

- *Value phase*: the possible values of $max1(\Phi)$, i.e., $\nu(max1(\Phi))$, is obtained in this phase.
- *Probability phase*: the probability to each value in $\nu(max1(\Phi))$ is assigned in this phase.

Value Phase

A lemma to obtain $\nu(max1(\Phi))$ is shown in the following. This lemma describes that each value in $\nu(max1(\Phi))$ is greater or equal than the

[1] $min1(\Phi)$ can be considered analogously.

maximal value of a set A, which consists of the minimal values of each $\nu(\eta_i)$, $1 \leq i \leq n$.

LEMMA 2. *Let* $\Phi = \{\eta_1, \eta_2, \ldots, \eta_n\}$, *where* $\eta_i = [a_{i1}^{p_{i1}}, a_{i2}^{p_{i2}}, \ldots, a_{im_i}^{p_{im_i}}]$, $1 \leq i \leq n$, *be a set of probabilistic partial values. The possible values of the result of* $max1(\Phi)$ *can be evaluated as follows.*

$$\nu(max1(\Phi)) = \Big\{a \mid a \in \bigcup_{1 \leq i \leq n} \nu(\eta_i),\ a \geq \max_{1 \leq i \leq n} \min \nu(\eta_i)\Big\}.$$

Proof. For all $S_{\Delta_k} = \langle S_{\alpha_k}, S_{\theta_k} \rangle \in \mathcal{F}(\Phi)$, let $S_{\alpha_k} = \{a_{ik} \mid a_{ik} \in \eta_i,\ 1 \leq i \leq n\}$, $1 \leq k \leq |\mathcal{F}(\Phi)|$. By Definition 10, we can conclude $\nu(max1(\Phi)) = \cup_{\langle S_{\alpha_i}, S_{\theta_i} \rangle \in \mathcal{F}(\Phi)} \{\max S_{\alpha_i}\}$.

Let $A = \cup_{\langle S_{\alpha_i}, S_{\theta_i} \rangle \in \mathcal{F}(\Phi)} \{\max S_{\alpha_i}\}$ and

$$B = \Big\{a \mid a \in \bigcup_{1 \leq i \leq n} \nu(\eta_i),\ a \geq \max_{1 \leq i \leq n} \min \nu(\eta_i)\Big\}.$$

We want to show $A = B$.

"$\subseteq$": For any $\max S_{\alpha_k} \in A$, we have

$$\begin{aligned}
\max S_{\alpha_k} &= \max\{a_{ik} \mid a_{ik} \in \nu(\eta_i), 1 \leq i \leq n\} \\
&= \max\{a_{ik} \mid a_{ik} \in \nu(\eta_i), a_{ik} \geq \min \nu(\eta_i), 1 \leq i \leq n\} \\
&= \max\Big\{a_{ik} \mid a_{ik} \in \nu(\eta_i), a_{ik} \geq \max_{1 \leq j \leq n} \min \nu(\eta_j), 1 \leq i \leq n\Big\} \\
&\in \Big\{a \mid a \in \bigcup_{1 \leq i \leq n} \nu(\eta_i), a \geq \max_{1 \leq j \leq n} \min \nu(\eta_j)\Big\} = B.
\end{aligned}$$

"$\supseteq$": Assume $a \in B$, then we have $a \geq \max_{1 \leq i \leq n} \min \nu(\eta_i)$ and $a \in \cup_{1 \leq i \leq n} \nu(\eta_i)$. Hence, there exists an $\eta_l \in \Phi$, such that $a \in \eta_l$. Also, there exists an $\langle S_{\alpha_k}, S_{\theta_k} \rangle \in \mathcal{F}(\Phi)$, $S_{\alpha_k} = \{a_{ik} \mid a_{ik} \in \eta_i,\ 1 \leq i \leq n\}$ such that

$$a_{ik} = \begin{cases} a, & \text{if } i = l \\ \min \nu(\eta_i), & \text{otherwise.} \end{cases}$$

This implies $a \geq \max_{1 \leq j \leq n} \min \nu(\eta_j) \geq \min \nu(\eta_i) = a_{ik}$, for all $1 \leq i \leq n$. Hence, $\max S_{\alpha_k} = a$. By $\max X_k \in A$, we have $a \in A$. □

Obviously, the time complexity for obtaining the set $\nu(max1(\Phi))$ from Lemma 2 is linear, i.e., $O(\sum_{i=1}^{n} m_i)$. We give an example to demonstrate how the lemma works.

EXAMPLE 9. Consider the relation **Person** shown in Table 1 again. We have $\Phi = \{\eta_1, \eta_2\}$, where $\eta_1 = [130^{0.6}, 140^{0.4}]$ and $\eta_2 = [120^{0.2}, 130^{0.5},$

$140^{0.3}]$. The possible values of $max1(\Phi)$ can be obtained by Lemma 2 with the following steps. Since

$$\bigcup_{1\le i\le 2} \nu(\eta_i) = \{120, 130, 140\}$$

and

$$\begin{aligned}\max_{1\le i\le 2} \min \nu(\eta_i) &= \max\{\min\{130, 140\}, \min\{120, 130, 140\}\} \\ &= \max\{130, 120\} = 130,\end{aligned}$$

we have

$$\begin{aligned}\nu(max1(\Phi)) &= \Big\{a \mid a \in \bigcup_{1\le i\le 2} \nu(\eta_i),\ a \ge \max_{1\le i\le 2} \min \nu(\eta_i)\Big\} \\ &= \{a \mid a \in \{120, 130, 140\}, a \ge 130\} \\ &= \{130, 140\}.\end{aligned}$$

■

Similarly, we can evaluate $\nu(min1(\Phi))$ as follows.

LEMMA 3. *Let* $\Phi = \{\eta_1, \eta_2, \ldots, \eta_n\}$, *where* $\eta_i = [a_{i1}^{p_{i1}}, a_{i2}^{p_{i2}}, \ldots, a_{im_i}^{p_{im_i}}]$, $1 \le i \le n$, *be a set of probabilistic partial values. The possible values of the result of* $min1(\Phi)$ *can be evaluated as follows.*

$$\nu(min1(\Phi)) = \Big\{a \mid a \in \bigcup_{1\le i\le n} \nu(\eta_i),\ a \le \min_{1\le i\le n} \max \nu(\eta_i)\Big\}.$$

Proof. The proof can be obtained in an analogous way as that of Lemma 2 by replacing "max," "min," and "$\ge$" with "min," "max," and "$\le$," respectively. □

Probability Phase After $\nu(max1(\Phi))$[2] is obtained in the value phase, the corresponding probability for each value in $\nu(max1(\Phi))$ has to be assigned. To explain how this phase works, we first give an example as follows.

EXAMPLE 10. Consider Example 9. In order to facilitate the following discussions, we label i to each possible value and its associated probability in η_i. That is,

$$\begin{aligned}\Phi &= \{\eta_1, \eta_2\}, \\ \eta_1 &= \big[130_1^{0.6_1}, 140_1^{0.4_1}\big], \text{and} \\ \eta_2 &= \big[120_2^{0.2_2}, 130_2^{0.5_2}, 140_2^{0.3_2}\big].\end{aligned}$$

[2] $min1(\Phi)$ can be considered analogously.

Consider all the value sets of the interpretations of Φ. They are

$$\begin{aligned}
S_{\Delta_1} &= \langle\{130_1, 120_2\}, 0.6_1 \times 0.2_2\rangle, \\
S_{\Delta_2} &= \langle\{130_1, 130_2\}, 0.6_1 \times 0.5_2\rangle, \\
S_{\Delta_3} &= \langle\{130_1, 140_2\}, 0.6_1 \times 0.3_2\rangle, \\
S_{\Delta_4} &= \langle\{140_1, 120_2\}, 0.4_1 \times 0.2_2\rangle, \\
S_{\Delta_5} &= \langle\{140_1, 130_2\}, 0.4_1 \times 0.5_2\rangle, \text{ and} \\
S_{\Delta_6} &= \langle\{140_1, 140_2\}, 0.4_1 \times 0.3_2\rangle.
\end{aligned}$$

We want to assign a probability for each value in the set $\nu(max1(\Phi)) = \{130, 140\}$ obtained in Example 9. First, we consider the assignment of the probability of 130. From Definition 10 and the definition of α–*union*, we have

$$\begin{aligned}
\mu_{\max 1(\Phi)}(130) &= \sum_{\substack{x \in \{S_\theta | S_{\Delta_i} = \langle S_\alpha, S_\theta\rangle \wedge \\ \max S_\alpha = 130,\, 1 \le i \le 6\}}} x \\
&= S_{\theta_1} + S_{\theta_2} \\
&= 0.6_1 \times 0.2_2 + 0.6_1 \times 0.5_2 \\
&= 0.42. \qquad (2)
\end{aligned}$$

In equation (2), three operations (two multiplications and one addition) are needed. We can reduce the number of operations as

$$\begin{aligned}
\mu_{\max 1(\Phi)}(130) &= 0.6_1 \times (0.2_2 + 0.5_2) \\
&= 0.42. \qquad (3)
\end{aligned}$$

From equation (3) we only need two operations (one multiplication and one addition). Now, consider the assignment of the probability of 140.

$$\begin{aligned}
\mu_{\max 1(\Phi)}(140) &= \sum_{\substack{x \in \{S_\theta | S_{\Delta_i} = \langle S_\alpha, S_\theta\rangle \wedge \\ \max S_\alpha = 140, 1 \le i \le 6\}}} x \\
&= S_{\theta_3} + S_{\theta_4} + S_{\theta_5} + S_{\theta_6} \\
&= 0.6_1 \times 0.3_2 + 0.4_1 \times 0.2_2 + 0.4_1 \times 0.5_2 + 0.4_1 \times 0.3_2 \\
&= 0.58. \qquad (4)
\end{aligned}$$

We can reduce the number of operations as

$$\begin{aligned}
\mu_{\max 1(\Phi)}(140) &= 0.6_1 \times 0.3_2 + 0.4_1 \times (0.2_2 + 0.5_2 + 0.3_2) \\
&= 0.58. \qquad (5)
\end{aligned}$$

■

From Example 10, we have the following observation:

> The probability of x, which is in $\nu(\max 1(\Phi))$, can be obtained by adding the probability parts of the interpretations of Φ, say $\Delta = \langle \alpha, \theta \rangle$, where for the corresponding value set, $S_\Delta = \langle S_\alpha, S_\theta \rangle$, $\max S_\alpha = x$. Further, for computing the probability of x, we can use the *distribution law* to reduce the number of operations.

Let $\Phi = \{\eta_1, \eta_2, \ldots, \eta_n\}$ be a set of probabilistic partial values. For each element $u \in \nu(max1(\Phi))$, we define a set $I(u)$ to record the probabilistic partial values in Φ, in which u is a possible value. That is, $\forall\, k \in I(u), u \in \nu(\eta_k)$. The cardinality of $I(u)$ can be used to know how many addition operations are needed to compute $max1(\Phi)$. In fact, it is $|I(u)|-1$. Consider Example 10. Since $\nu(max1(\Phi)) = \{130, 140\}$, we have $I(130) = \{1, 2\}$ and $I(140) = \{1, 2\}$. Therefore, the number of addition operations for $max1(\Phi)$ is 1.

For a probabilistic partial value, $\eta = [a_1^{p_1}, a_2^{p_2}, \ldots, a_m^{p_m}]$, without loss of generality, we assume that $a_1 < a_2 < \cdots < a_m$.

DEFINITION 17. For a probabilistic partial value, $\eta = [a_1^{p_1}, a_2^{p_2}, \ldots, a_m^{p_m}]$, two functions l and l^* from $\boldsymbol{R}$ to a range 0 to 1 are defined as follows (for the $max1$ aggregate function).

$$l_\eta(x) = \begin{cases} \sum_{i=1}^{k} p_i, & a_k \le x < a_{k+1}, \quad 1 \le k \le m-1 \\ 1, & a_m \le x \\ 0, & x < a_1 \end{cases} \qquad x \in \boldsymbol{R} \tag{6}$$

$$l^*_\eta(x) = \begin{cases} 0 & x \le a_1 \\ l_\eta(a_j) & \exists j, a_j < x \le a_{j+1}, \quad 1 \le j \le m-1 \\ 1, & x > a_m \end{cases} \qquad x \in \boldsymbol{R}. \tag{7}$$

The above definition describes that the $l_\eta(x)$ and $l^*_\eta(x)$ functions have the same values except for $x \in \{a_1, a_2, \ldots, a_m\}$.

EXAMPLE 11. According to the above definition, we have the following representations. From equation (3) in Example 10, we have

$$\begin{aligned} \mu_{\max 1(\Phi)}(130) &= 0.6_1 \times (0.2_2 + 0.5_2) + 0.0_1 \times 0.5_2 \\ &= \mu_{\eta_1}(130) \times l_{\eta_2}(130) + l^*_{\eta_1}(130) \times \mu_{\eta_2}(130). \end{aligned} \tag{8}$$

In equation (8), we add the term $0.0_1 \times 0.5_2$ to here in order to unify the representation of the equation, because the number of needed addition

operations is 1. Also, we replace each value term with their corresponding meaning. That is, $\mu_{\eta_1}(130)$, $l_{\eta_2}(130)$, $l^*_{\eta_1}(130)$, and $\mu_{\eta_2}(130)$ mean the probability of 130 in η_1, the accumulative probability for the corresponding possible values, which are less than or equal to 130 in η_2, the accumulative probability for the corresponding possible values, which are less than 130 in η_1 (we use $l^*_{\eta_1}(130)$ instead of $l_{\eta_1}(130)$ to prevent the duplication of computing the probabilities), and the probability of 130 in η_2, respectively.

From Equation (5) in Example 10, we have

$$\begin{aligned}\mu_{\max 1(\Phi)}(140) &= 0.4_1 \times (0.2_2 + 0.5_2 + 0.3_2) + 0.6_1 \times 0.3_2 \\ &= \mu_{\eta_1}(140) \times l_{\eta_2}(140) + l^*_{\eta_1}(140) \times \mu_{\eta_2}(140). \quad \blacksquare\end{aligned}$$

DEFINITION 18. For a probabilistic partial value, $\eta = [a_1^{p_1}, a_2^{p_2}, \ldots, a_m^{p_m}]$, two functions t and t^* from $\boldsymbol{R}$ to a range 0 to 1 are defined as follows (for the $min1$ aggregate function).

$$t_\eta(x) = \begin{cases} \sum_{i=k}^{m} p_i, & a_{k-1} < x \le a_k, \quad 2 \le k \le m \\ 1, & x \le a_1 \\ 0, & a_m < x \end{cases} \qquad x \in \boldsymbol{R} \tag{9}$$

$$t^*_\eta(x) = \begin{cases} 0 & x \ge a_m \\ t_\eta(a_j) & \exists j, a_{j-1} \le x < a_j, \quad 2 \le j \le m \\ 1, & x < a_1 \end{cases} \qquad x \in \boldsymbol{R}. \tag{10}$$

The above definition describes that the $t_\eta(x)$ and $t^*_\eta(x)$ functions have the same values except for $x \in \{a_1, a_2, \ldots, a_m\}$.

Algorithm 1 and Algorithm 2 for evaluating $max1(\Phi)$ and $min1(\Phi)$, respectively, are given in the following.

ALGORITHM 1: **Maximum**
Input: a set of probabilistic partial values, $\Phi = \{\eta_1, \eta_2, \ldots, \eta_n\}$ and $\eta_i = [a_{i1}^{p_{i1}}, a_{i2}^{p_{i2}}, \ldots, a_{im_i}^{p_{im_i}}]$, $1 \le i \le n$. Let $Z = \nu(\eta_1) \cup \cdots \cup \nu(\eta_n)$.
Output: a maximal probabilistic partial value of Φ
Comments: value phase: step 1, probability phase: steps 2–16 R, U, W are temporal variables.

0: **begin**
1: **let** $x = \max_{1 \le i \le n} \min \nu(\eta_i)$
2: **for each** a_{ij}, evaluates its corresponding $l_{\eta_i}(a_{ij})$, $1 \le j \le m_i$, $1 \le i \le n$
3: **for each** $u \in Z$ **and** $u \ge x$

4: **begin**
5: $W = \phi$
6: $R = \phi$
7: $I(u) = \{k \mid \mu_{\eta_k}(u) > 0, 1 \leq k \leq n\}$
8: **for each** $k \in I(u)$
9: **begin**
10: $U = \{l_{\eta_i}(u) \mid \forall\, i \neq k, i \notin R, 1 \leq i \leq n\} \bigcup$
$\{l^*_{\eta_i}(u)^3 \mid \forall\, i \neq k,\ i \in R,\ 1 \leq i \leq n\} \bigcup \{\mu_{\eta_k}(u)\}$
11: $w = \mathbf{TIMES}\{U\}$
12: $W = W \cup \{w\}$
13: $R = R \cup \{k\}$
14: **end**
15: $\mu_{max1(\Phi)}(u) = \mathbf{ADDS}\{W\}$
16: **end**
17: **end**

ALGORITHM 2: **Minimum**
Input: a set of probabilistic partial values, $\Phi = \{\eta_1, \eta_2, \ldots, \eta_n\}$ and $\eta_i = [a_{i1}^{p_{i1}}, a_{i2}^{p_{i2}}, \ldots, a_{im_i}^{p_{im_i}}]$, $1 \leq i \leq n$. Let $Z = \nu(\eta_1) \cup \cdots \cup \nu(\eta_n)$.
Output: a minimal probabilistic partial value of Φ
Comments: value phase: step 1, probability phase: steps 2–16. R, U, W are temporal variables.

0: **begin**
1: **let** $x = \min_{1 \leq i \leq n} \max \nu(\eta_i)$
2: **for each** a_{ij}, evaluates its corresponding $t_{\eta_i}(a_{ij})$, $1 \leq j \leq m_i$, $1 \leq i \leq n$
3: **for each** $u \in Z$ **and** $u \leq x$
4: **begin**
5: $W = \phi$
6: $R = \phi$
7: $I(u) = \{k \mid \mu_{\eta_k}(u) > 0, 1 \leq k \leq n\}$
8: **for each** $k \in I(u)$
9: **begin**
10: $U = \{t_{\eta_i}(u) \mid \forall i \neq k, i \notin R, 1 \leq i \leq n\} \bigcup$
$\{t^*_{\eta_i}(u)^4 \mid \forall\, i \neq k, i \in R, 1 \leq i \leq n\} \bigcup \{\mu_{\eta_k}(u)\}$
11: $w = \mathbf{TIMES}\{U\}$
12: $W = W \cup \{w\}$
13: $R = R \cup \{k\}$

[3] l^* function can be computed by l function.
[4] t^* function can be computed by t function.

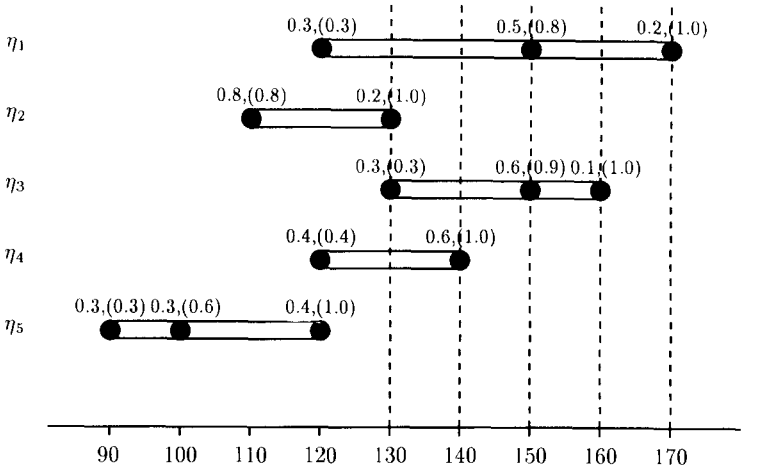

Fig. 2. The explanation graph of $max1(weight)$.

14: **end**
15: $\mu_{min1(\Phi)}(u) = \mathbf{ADDS}\{W\}$
16: **end**
17: **end**

Notice that, in the two algorithms, **ADDS** and **TIMES** are two numerical operators. **ADDS** and **TIMES** accept any number of arguments and return the sum and product of these arguments, respectively. That is, the two operators are the conventional summation (+) and multiplication (×).

In the following, we define an *explanation graph* to explain the process of the algorithm. Consider a set of probabilistic partial values, $\Phi = \{\eta_1, \eta_2, \ldots, \eta_n\}$, where $\eta_i = [a_{i1}^{p_{i1}}, a_{i2}^{p_{i2}}, \ldots, a_{im_i}^{p_{im_i}}]$, $1 \leq i \leq n$. An explanation graph is a graph in which each η_i, $1 \leq i \leq n$, is represented by a bar with a bullet for each possible value in $\nu(\eta_i)$ on the corresponding position. Each bullet representing a possible value, say $a_{ij}^{p_{ij}}$, is associated with a pair of values, i.e., $\mu_{\eta_i}(a_{ij}), (l_{\eta_i}(a_{ij}))$. According to the set $Z = \cup_{i=1}^{n} \nu(\eta_i)$ and the value x obtained from step 1 in the algorithm, we compute the probability for each element in $\nu(max1(\Phi)) = \{a \mid a \in Z \wedge a \geq x\}$ (a dashed line is indicated on the corresponding position for each element in $\nu(max1(\Phi))$). As an example, an explanation graph for Example 12 is given in Figure 2.

EXAMPLE 12. Consider the example relation **Person** shown in Table 2. We construct the explanation graph of the set of probabilistic partial values, $weight = \{\eta_1, \ldots, \eta_5\}$, as shown in Figure 2. $max1(weight)$ is evaluated

TABLE 2
A Probabilistic Relation PERSON

...	*weight*	...
.	$\eta_1 = [120^{0.3}, 150^{0.5}, 170^{0.2}]$	.
.	$\eta_2 = [110^{0.8}, 130^{0.2}]$	.
.	$\eta_3 = [130^{0.3}, 150^{0.6}, 160^{0.1}]$	.
.	$\eta_4 = [120^{0.4}, 140^{0.6}]$	.
.	$\eta_5 = [90^{0.3}, 100^{0.3}, 120^{0.4}]$	.

by the following steps. We have

$$\begin{aligned} Z &= \nu(\eta_1) \cup \cdots \cup \nu(\eta_5) \\ &= \{90, 100, 110, 120, 130, 140, 150, 160, 170\}, \\ x &= \max\{\min \nu(\eta_1), \min \nu(\eta_2), \ldots, \min \nu(\eta_5)\} \\ &= \max\{120, 110, 130, 120, 90\} = 130. \end{aligned}$$

Hence,

$$\begin{aligned} \nu(max1(weight)) &= \{a \mid a \in Z \wedge a \geq x\} \\ &= \{130, 140, 150, 160, 170\}. \end{aligned}$$

Then

$$\begin{aligned} \mu_{max1(weight)}(130) &= \mathbf{ADDS}\{\mathbf{TIMES}\{(0.3), 0.2, (0.3), (0.4), (1.0)\}, \\ &\qquad \mathbf{TIMES}\{(0.3), (0.8), 0.3, (0.4), (1.0)\}\} \\ &= 0.036, \\ \mu_{max1(weight)}(140) &= \mathbf{ADDS}\{\mathbf{TIMES}\{(0.3), (1.0), (0.3), 0.6, (1.0)\}\} \\ &= 0.054, \\ \mu_{max1(weight)}(150) &= \mathbf{ADDS}\{\mathbf{TIMES}\{0.5, (1.0), (0.9), (1.0), (1.0)\}, \\ &\qquad \mathbf{TIMES}\{(0.3), (1.0), 0.6, (1.0), (1.0)\}\} \\ &= 0.63, \\ \mu_{max1(weight)}(160) &= \mathbf{ADDS}\{\mathbf{TIMES}\{(0.8), (1.0), 0.1, (1.0), (1.0)\}\} \\ &= 0.08, \\ \mu_{max1(weight)}(170) &= \mathbf{ADDS}\{\mathbf{TIMES}\{0.2, (1.0), (1.0), (1.0), (1.0)\}\} \\ &= 0.2, \text{ and} \\ max1(weight) &= [130^{0.036}, 140^{0.054}, 150^{0.63}, 160^{0.08}, 170^{0.2}]. \end{aligned}$$

■

EXAMPLE 13. To evaluate $min1(weight)$, we first compute x, which is equal to 120. The possible values for $min1(weight)$ are $\{a \mid a \in Z \wedge a \leq x\} = \{90, 100, 110, 120\}$. The explanation graph is the same as the one in Figure 2 except that the dashed lines are now indicated on the corresponding positions of the possible result values, $\{90, 100, 110, 120\}$, and the pair of values associated with each bullet is computed by the t function instead of the l function. The result of $min1(weight)$ is

$$min1(weight) = [90^{0.3}, 100^{0.3}, 110^{0.32}, 120^{0.08}]. \quad \blacksquare$$

5.2. CORRECTNESS

In this section, we show the correctness of the two aggregate algorithms. We say the algorithms are correct if they return the same probabilistic partial values as defined in Definition 10 and Definition 11.

THEOREM 3. *Algorithm* 1 *is correct.*

Proof. Consider a set of probabilistic partial values, $\Phi = \{\eta_1, \eta_2, \ldots, \eta_n\}$, $1 \leq i \leq n$. Let the result of $max1(\Phi)$ by computing all interpretations of Φ be denoted $\mathcal{B} = max1(\Phi) = [b_1^{\mu_{\mathcal{B}}(b_1)}, \ldots, b_r^{\mu_{\mathcal{B}}(b_r)}]$, and that by using Algorithm 1 be denoted $\mathcal{E} = max1(\Phi) = [e_1^{\mu_{\mathcal{E}}(e_1)}, \ldots, e_s^{\mu_{\mathcal{E}}(e_s)}]$. Recall that, we assume $b_1 < b_2 < \cdots < b_r$ and $e_1 < e_2 < \cdots < e_s$. We want to show that $\mathcal{B} = \mathcal{E}$. That is, we will prove that $b_i = e_i$ and $\mu_{\mathcal{B}}(b_i) = \mu_{\mathcal{E}}(e_i)$, $1 \leq i \leq s$ (or r).

From Lemma 2 and step 1 in Algorithm 1, we have obtained $\nu(\mathcal{B}) = \nu(\mathcal{E})$. That is, $b_i = e_i$ and $r = s$, $1 \leq i \leq s$. We show $\mu_{\mathcal{B}}(b_i) = \mu_{\mathcal{E}}(e_i)$, $1 \leq i \leq s$ by mathematical induction on n as follows.

Basis step: $n = 1$. In this case, $\Phi = \{\eta\}$ and $\eta = [a_1^{p_1}, \ldots, a_m^{p_m}]$. Clearly, $\mu_{\mathcal{B}}(b_i) = \mu_{\mathcal{E}}(e_i) = p_i$, $1 \leq i \leq m$. Hence, $\mathcal{E} = \mathcal{B} = max1(\Phi)$.

Inductive hypothesis: Suppose that the claim holds when $n = k$. That is, $\Phi = \{\eta_1, \eta_2, \ldots, \eta_k\}$ and $\mathcal{E} = \mathcal{B} = max1(\Phi)$.

Inductive step: Consider $n = k + 1$. That is, $\Phi = \{\eta_1, \eta_2, \ldots, \eta_{k+1}\}$. We have

$$\begin{aligned} max1(\Phi) &= max1(\{\eta_1, \eta_2, \ldots, \eta_{k+1}\}) \\ &= max1(\{\eta_1, \eta_2, \ldots, \eta_k\} \cup \{\eta_{k+1}\}) \\ &= max1(\{max1(\{\eta_1, \eta_2, \ldots, \eta_k\}), \eta_{k+1}\}) \\ &= max1(\{C, \eta_{k+1}\}) \end{aligned} \tag{11}$$

where $C = max1(\{\eta_1, \eta_2, \ldots, \eta_k\})$, which can be correctly computed by Algorithm 1 by the **inductive hypothesis**. Let $C = [a_{11}^{p_{11}}, a_{12}^{p_{12}}, \ldots, a_{1m_1}^{p_{1m_1}}]$ and $\eta_{k+1} = [a_{21}^{p_{21}}, a_{22}^{p_{22}}, \ldots, a_{2m_2}^{p_{2m_2}}]$. Recall that $\nu(\mathcal{E}) = \nu(\mathcal{B})$. For each $e \in$

$\nu(\mathcal{E})$ $(= \nu(\mathcal{B}))$, there exists a corresponding $b \in \nu(\mathcal{B})$ such that $b = e$, and vice versa. We show $max1(\{C, \eta_{k+1}\})$ can be correctly computed by Algorithm 1 by considering the following three cases.

Case 1: $\exists i, j$ such that $e(= a_{1i}) \in \nu(C)$ **and** $e(= a_{2j}) \in \nu(\eta_{k+1})$.

The probability of e, $\mu_{\mathcal{E}}(e)$, computed by Algorithm 1 is shown as

$$\begin{aligned}\mu_{\mathcal{E}}(e) &= \mu_C(e) \times l_{\eta_{k+1}}(e) + l^*_C(e) \times \mu_{\eta_{k+1}}(e) \\ &= \mu_C(a_{1i}) \times l_{\eta_{k+1}}(a_{2j}) + l^*_C(a_{1i}) \times \mu_{\eta_{k+1}}(a_{2j}) \\ &= p_{1i} \times \left(\sum_{k=1}^{j} p_{2k}\right) + \left(\sum_{k=1}^{i-1} p_{1k}\right) \times p_{2j}.\end{aligned}$$

Let

$$\Theta = \{S_\theta \mid \forall S_\Delta = \langle S_\alpha, S_\theta \rangle \in \mathcal{F}(\Phi) \wedge \max S_\alpha = b\}.$$

In contrast to our algorithm, the probability of b $(= e)$ in $\nu(\mathcal{B})$ as computed by generating all interpretations is shown as

$$\begin{aligned}\mu_{\mathcal{B}}(b) &= \sum_{\theta \in \Theta} \theta \\ &= \mu_C(a_{1i}) \times \mu_{\eta_{k+1}}(a_{21}) + \mu_C(a_{1i}) \times \mu_{\eta_{k+1}}(a_{22}) + \cdots + \mu_C(a_{1i}) \\ &\quad \times \mu_{\eta_{k+1}}(a_{2j}) + \mu_C(a_{11}) \times \mu_{\eta_{k+1}}(a_{2j}) + \mu_C(a_{12}) \\ &\quad \times \mu_{\eta_{k+1}}(a_{2j}) + \cdots + \mu_C(a_{1(i-1)}) \times \mu_{\eta_{k+1}}(a_{2j}) \\ &= p_{1i} \times p_{21} + p_{1i} \times p_{22} + \cdots + p_{1i} \times p_{2j} \\ &\quad + p_{11} \times p_{2j} + p_{12} \times p_{2j} + \cdots + p_{1(i-1)} \times p_{2j} \\ &= p_{1i} \times (p_{21} + p_{22} + \cdots + p_{2j}) + (p_{11} + p_{12} + \cdots + p_{1(i-1)}) \times p_{2j} \\ &= p_{1i} \times \left(\sum_{k=1}^{j} p_{2k}\right) + \left(\sum_{k=1}^{i-1} p_{1k}\right) \times p_{2j} \\ &= \mu_{\mathcal{E}}(e).\end{aligned}$$

Case 2: $\exists i$ such that $e(= a_{1i}) \in \nu(C)$ **and** $e \notin \nu(\eta_{k+1})$ (without loss of generality, assume $\exists j$ such that $a_{2j} < e < a_{2(j+1)}$).

The probability of e, $\mu_{\mathcal{E}}(e)$, computed by Algorithm 1 is shown as

$$\begin{aligned}\mu_{\mathcal{E}}(e) &= \mu_C(a_{1i}) \times l(a_{2j}) \\ &= p_{1i} \times \sum_{k=1}^{j} p_{2k}.\end{aligned}$$

Let

$$\Theta = \{S_\theta \mid \forall S_\Delta = \langle S_\alpha, S_\theta \rangle \in \mathcal{F}(\Phi) \wedge \max S_\alpha = b\}.$$

In contrast to our algorithm, the probability of b $(= e)$ in $\nu(\mathcal{B})$ as computed by generating all interpretations is shown as

$$\begin{aligned}
\mu_{\mathcal{B}}(b) &= \mu_C(b) \times \mu_{\eta_{k+1}}(b) \\
&= \mu_C(a_{1i}) \times \mu_{\eta_{k+1}}(a_{21}) + \mu_C(a_{1i}) \times \mu_{\eta_{k+1}}(a_{22}) \\
&\quad + \cdots + \mu_C(a_{1i}) \times \mu_{\eta_{k+1}}(a_{2j}) \\
&= p_{1i} \times \sum_{k=1}^{j} p_{2k} \\
&= \mu_{\mathcal{E}}(e).
\end{aligned}$$

Case 3: $\exists j$ such that $e \notin \nu(C)$ **and** $e(= a_{2j}) \in \nu(\eta_{k+1})$.

In this case, the discussion is the same as that in case 2. By the use of mathematical induction, we complete the proof. □

THEOREM 4. *Algorithm* 2 *is correct.*

Proof. The proof can be obtained analogously as that of Theorem 3 by replacing "l," "l^*," and "max" with "t," "t^*," and "min," respectively. □

5.3. *TIME COMPLEXITY*

In this section, we show the **Maximum** and **Minimum** algorithms are *efficient.* That is, the time complexity of the computations is polynomial in worst case.

THEOREM 5. *The worst-case time complexity of Algorithm* 1 *is* $O(n^2)$.

Proof. Consider a set of probabilistic partial values, $\Phi = \{\eta_1, \eta_2, \cdots, \eta_n\}$. We make a reasonable assumption, $n \gg |\nu(\eta_i)|$, $1 \leq i \leq n$, and furthermore, without loss of generality, we assume that $|\nu(\eta_i)| = m$. From step 7 in Algorithm 1, the worst-case time complexity occurs when the set $I(u)$ contains n elements, i.e., $I(u) = \{1, 2, \ldots, n\}$, $\forall\, u \in \cup_{i=1}^{n} \nu(\eta_i)$. That is, $\nu(\eta_1) = \nu(\eta_2) = \cdots = \nu(\eta_n)$. We construct the explanation graph for this case in Figure 3. The time complexity of Algorithm 1 is computed

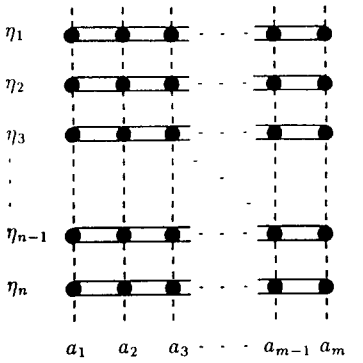

Fig. 3. An explanation graph for time complexity analysis.

step by step as follows.

step 1:	mn
step 2:	mn
step 3:	**for** $i = 1$ **to** m
step 5,6,7:	$n + 2$
step 8:	**for** $j = 1$ **to** n
step 10:	n
step 11,12,13:	$n + 2$
step 14:	**end for**
step 15:	1
step 16:	**end for**

Hence, the time complexity is

$$\begin{aligned} & mn + mn + \sum_{i=1}^{m} \left(n + 3 + \sum_{j=1}^{n} (2n + 2) \right) \\ &= 2mn + m(n + 3 + n(2n + 2)) \\ &= 2mn^2 + 5mn + 3m. \end{aligned}$$

Consequently, we conclude that the worst-case time complexity of Algorithm 1 is $O(n^2)$, under the assumption $n \gg m$. □

THEOREM 6. *The worst-case time complexity of Algorithm* 2 *is* $O(n^2)$.

Proof. The proof can be obtained in the analogous way as that of Theorem 5. □

6. EXTENSIONS FOR HANDLING POSSIBILISTIC DATA

A framework for handling aggregates in *possibilistic data* was proposed in [20]. In this section, we show that the Algorithm 1 and Algorithm 2 can be adapted for the *maximum* and *minimum* aggregate functions on possibilistic data, respectively. We follow the related definitions of Rundensteiner and Bic's work [20]. The detailed presentation of possibility theory can be found in [19].

DEFINITION 19. A *possibilistic value* $\gamma = p_1/a_1 + p_2/a_2 + \cdots + p_m/a_m$, where a_i is a possible value of γ and p_i is the possibility of a_i, $1 \leq i \leq m$.

The ν and μ functions for possibilistic values can also be used here. That is, $\nu(\gamma) = \{a_1, a_2, \ldots, a_m\}$ and $\mu_\gamma(a_i) = p_i$, $1 \leq i \leq n$. The assumption $a_1 < a_2 < \cdots < a_m$ for probabilistic partial values also holds for possibilistic values.

DEFINITION 20 [20]. Given a set of possibilistic values $\Phi = \{\gamma_1, \gamma_2, \ldots, \gamma_n\}$, where $\gamma_i = p_{i1}/a_{i1} + p_{i2}/a_{i2} + \cdots + p_{im_i}/a_{im_i}$, $1 \leq i \leq n$, the *maximum* aggregate function *fmax* and the *minimum* aggregate function *fmin* on possibilistic data are defined as

$$fmax(\Phi) = \Big\{u/y \mid \Big(\Big(y = \max_{ki=k1}^{kn} a_{ki}\Big) \wedge \Big(u = \min_{i=1}^{n} \mu_{\gamma_i}(a_{ki})\Big)\Big) \times (\forall k1, \ldots, kn: 1 \leq ki \leq m_i)\Big\}$$

$$fmin(\Phi) = \Big\{u/y \mid \Big(\Big(y = \min_{ki=k1}^{kn} a_{ki}\Big) \wedge \Big(u = \min_{i=1}^{n} \mu_{\gamma_i}(a_{ki})\Big)\Big) \times (\forall k1, \ldots, kn: 1 \leq ki \leq m_i)\Big\}.$$

In order to use the **Maximum** and **Minimum** algorithms to handle possibilistic values, we have to modify the functions defined in Definition 17 and Definition 18.

DEFINITION 21. For a possibilistic value, $\gamma = p_1/a_1 + p_2/a_2 + \cdots + p_m/a_m$, two functions $\bar{l}$ and $\bar{l}^*$ from $\boldsymbol{R}$ to a range 0 to 1 are defined as follows (for the $fmax$ aggregate function).

$$\bar{l}_\eta(x) = \begin{cases} \max_{i=1}^{k} p_i, & a_k \leq x < a_{k+1}, \quad 1 \leq k \leq m-1 \\ 1, & a_m \leq x \\ 0, & x < a_1 \end{cases} \qquad x \in \boldsymbol{R} \tag{12}$$

$$\bar{l}^*_\eta(x) = \begin{cases} 0 & x \le a_1 \\ \bar{l}_\eta(a_j) & \exists j, a_j < x \le a_{j+1}, \quad 1 \le j \le m-1 \\ 1, & x > a_m \end{cases} \qquad x \in \boldsymbol{R}. \quad (13)$$

The above definition describes that the $\bar{l}_\eta(x)$ and $\bar{l}^*_\eta(x)$ functions have the same values except for $x \in \{a_1, a_2, \ldots, a_m\}$.

DEFINITION 22. For a possibilistic value, $\gamma = p_1/a_1 + p_2/a_2 + \cdots + p_m/a_m$, two functions $\bar{t}$ and $\bar{t}^*$ from $\boldsymbol{R}$ to a range 0 to 1 are defined as follows (for the *fmin* aggregate function).

$$\bar{t}_\eta(x) = \begin{cases} \max\limits_{i=k}^{m} p_i, & a_{k-1} < x \le a_k, \quad 2 \le k \le m \\ 1, & x \le a_1 \\ 0, & a_m < x \end{cases} \qquad x \in \boldsymbol{R} \quad (14)$$

$$\bar{t}^*_\eta(x) = \begin{cases} 0 & x \ge a_m \\ \bar{t}_\eta(a_j) & \exists j, a_{j-1} \le x < a_j, \quad 2 \le j \le m \\ 1, & x < a_1 \end{cases} \qquad x \in \boldsymbol{R}. \quad (15)$$

The above definition describes that the $\bar{t}_\eta(x)$ and $\bar{t}^*_\eta(x)$ functions have the same values except for $x \in \{a_1, a_2, \ldots, a_m\}$.

Our algorithms on computing the *maximum* and *minimum* aggregate functions can be modified to handle possibilistic values. The needed modifications are described as follows.

1. **ADDS** and **TIMES** in Algorithm 1 and Algorithm 2 are replaced by **MAX** and **MIN**, respectively. **MAX** and **MIN** accept any number of arguments and return the maximum value and minimum value of these arguments, respectively.
2. Replace l and l^* functions with $\bar{l}$ and $\bar{l}^*$, respectively, in Algorithm 1 for *fmax* aggregate function, and replace t and t^* functions with $\bar{t}$ and $\bar{t}^*$, respectively, in Algorithm 2 for *fmin* aggregate function.

EXAMPLE 14. Consider a possibilistic relation **Person** shown in Table 3, which is similar to the probabilistic relation **Person** as shown in Table 2. An analogous explanation graph for the set of possibilistic values, *weight* $= \{\gamma_1, \ldots, \gamma_5\}$, is constructed in Figure 4. In Figure 4, the number in parentheses is the corresponding $\bar{t}$ value of each possible value.

TABLE 3
A Possibilistic Relation PERSON

...	*weight*	...
·	$\gamma_1 = 0.8/120 + 1./150 + 0.9/170$	·
·	$\gamma_2 = 0.9/110 + 0.8/130$	·
·	$\gamma_3 = 0.9/130 + 0.8/150 + 0.7/160$	·
·	$\gamma_4 = 1./120 + 0.8/140$	·
·	$\gamma_5 = 0.7/90 + 0.9/100 + 1./120$	·

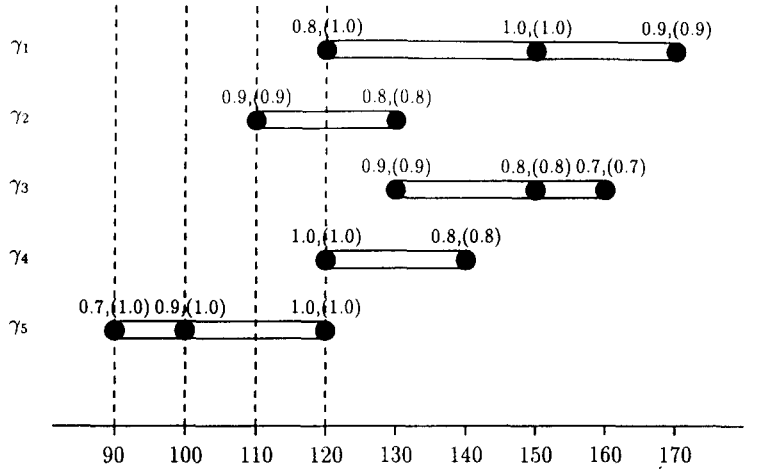

Fig. 4. The explanation graph of $fmin(weight)$.

Then $fmin(weight)$ can be evaluated by the following steps.

$$\begin{aligned} Z &= \nu(\gamma_1) \cup \cdots \cup \nu(\gamma_5) \\ &= \{90, 100, 110, 120, 130, 140, 150, 160, 170\}, \\ x &= \min\{\max \nu(\gamma_1), \ldots, \max \nu(\gamma_5)\} \\ &= \min\{170, 130, 160, 140, 120\} = 120. \end{aligned}$$

Hence,

$$\begin{aligned} \nu(fmin(weight)) &= \{a \mid a \in Z \wedge a \leq x\} \\ &= \{90, 100, 110, 120\}. \end{aligned}$$

$$\mu_{fmin(weight)}(90) = \mathbf{MAX}\{\mathbf{MIN}\{(1.0), (0.9), (0.9), (1.0), 0.7\}\} = 0.7,$$

$$\mu_{fmin(weight)}(100) = \mathbf{MAX}\{\mathbf{MIN}\{(1.0),(0.9),(0.9),(1.0),0.9\}\} = 0.9,$$

$$\mu_{fmin(weight)}(110) = \mathbf{MAX}\{\mathbf{MIN}\{(1.0),0.9,(0.9),(1.0),(1.0)\}\} = 0.9,$$

$$\begin{aligned}\mu_{fmin(weight)}(120) = \mathbf{MAX}\{&\mathbf{MIN}\{0.8,(0.8),(0.9),(1.0),(1.0)\},\\ &\mathbf{MIN}\{(1.0),(0.8),(0.9),1.0,(1.0)\},\\ &\mathbf{MIN}\{(1.0),(0.8),(0.9),(0.8),1.0\}\} = 0.8.\end{aligned}$$

$$fmin(weight) = 0.7/90 + 0.9/100 + 0.9/110 + 0.8/120. \quad \square$$

EXAMPLE 15. The result of $fmax(weight)$ is

$$fmax(weight) = 0.8/130 + 0.8/140 + 0.9/150 + 0.7/160 + 0.9/170. \quad \blacksquare$$

7. CONCLUSIONS

This paper studies a set of extended aggregate functions, namely *sum*, *average*, *count*, *maximum*, and *minimum*, over probabilistic data. For a set of probabilistic values, we can define extended aggregate functions based on its interpretations. The results of these extended aggregate functions are also probabilistic values. The users may prefer an approximate but definite answer for the extended aggregate functions, we give alternate definitions for the extended aggregate functions. The alternate definitions return approximate definite values. The time complexity of the computations is linear.

We show the computations of *sum*, *average*, and *count* are exponential, and develop two efficient algorithms for the *maximum* and *minimum*. The worst-case time complexity of these algorithms are $O(n^2)$. These two algorithms can be adapted for *possibilistic data* with slight modifications. If we ignore the probability phase in the algorithms, the exclusive disjunctive data (e.g., partial values) can also be handled. Therefore, our work is devoted to the accommodation of uncertain data in database systems with an elaboration on speeding up the processing efficiency of the aggregate functions.

The authors wish to thank the anonymous referees whose comments and suggestions helped improve this paper.

REFERENCES

1. D. Barbará, H. Garcia-Molina, and D. Porter, The management of probabilistic data, *IEEE Trans. Knowledge Data Eng.* 4(5):487–502 (1992).

2. J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*, Macmillan Press, New York, 1976.
3. R. Cavallo and M. Pittarelli, The Theory of probabilistic databases, in *Proceedings of the 13th VLDB Conference*, 1987, pp. 71–81.
4. C. S. Chang and A. L. P. Chen, Determining probabilities for probabilistic partial values, in *Proceedings of the International Conference on Data and Knowledge Systems for Manufacturing and Engineering (DKSME)*, 1994, pp. 277–284.
5. A. L. P. Chen, J. S. Chiu, and F. S. C. Tseng, Evaluating aggregate operations over imprecise data, *IEEE Trans. Knowledge Data Eng.* (to appear).
6. E. F. Codd, Missing information (applicable and inapplicable) in relational databases, *ACM SIGMOD Record.* 15(4):53–78 (1986).
7. C. J. Date, *A Guide to the SQL Standard*, Addison-Wesley, Reading, MA, 1989.
8. L. G. DeMichiel, Resolving database incompatibility: An approach to performing relational operations over mismatched domains, *IEEE Trans. Knowledge Data Eng.* 1(4):485–493 (1989).
9. D. Dubois and H. Prade, *Possibility Theory: An Approach to Computerized Processing of Uncertainty*, Plenum Press, New York, 1986.
10. N. Fuhr, A probabilistic framework for vague queries and imprecise information in databases, in *Proceedings of the 16th VLDB Conference*, 1990, pp. 696–707.
11. J. Grant, Partial values in a tabular database model, *Information Processing Letters* 9(2):97–99 (1979).
12. P. G. W. Keen and M. S. S. Morton, *Decision Support Systems: An Organizational Perspective*, Addison-Wesley, Reading, MA, 1978.
13. S. K. Lee, An extended relational database model for uncertain and imprecise information, in *Proceedings of the 18th VLDB Conference*, 1992, pp. 211–220.
14. W. Lipski, On semantic issues connected with incomplete information databases, *ACM Trans. Database Systems* 4(3):262–296 (1979).
15. P. L. Meyer, *Introductory Probability and Statistical Applications*, 2nd ed., Addison-Wesley, Reading, MA, 1970.
16. A. Ola, Relational databases with exclusive disjunctions, in *Proceedings of the IEEE International Conference on Data Engineering*, 1992, pp. 328–336.
17. G. Özsoyoğlu, Z. M. Özsoyoğlu, and V. Matos, Extending relational algebra and relational calculus with set-valued attributes and aggregate functions, *ACM Trans. Database Systems* 12(4):566–592 (1987).
18. C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, NJ, 1982.
19. H. Prade and C. Testemale, Generalizing database relational algebra for the treatment of incomplete or uncertain information and vague queries, *Information Sciences* 34:115–143 (1984).
20. E. A. Rundensteiner and L. Bic, Evaluating aggregates in possibilistic relational databases, *Data Knowledge Eng.* 7:239–267 (1992).

21. F. S. C. Tseng, A. L. P. Chen, and W. P. Yang, Answering heterogeneous database queries with degrees of uncertainty, *Distributed Parallel Databases* 1(3):281–302 (1993).
22. L. A. Zadeh, Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets System* 1(1):3–28 (1978).
23. M. Zemankova and A. Kandel, Implementing imprecision in information systems, *Information Sciences* 37:107–141 (1985).

Received 3 August 1993; revised 29 July 1994 and 19 December 1994