*Journal of Computer Science and Cybernetics, V.35, N.1 (2019), 1–20*
DOI 10.15625/1813-9663/35/1/13234

# HUPSMT: AN EFFICIENT ALGORITHM FOR MINING HIGH UTILITY-PROBABILITY SEQUENCES IN UNCERTAIN DATABASES WITH MULTIPLE MINIMUM UTILITY THRESHOLDS

TRUONG CHI TIN[1,*], TRAN NGOC ANH[1], DUONG VAN HAI[1,2], LE HOAI BAC[2]

[1]*Department of Mathematics and Computer Science, University of Dalat*
[2]*Department of Computer Science, VNU-HCMC University of Science*
[*]*tintc@dlu.edu.vn*



**Abstract.** The problem of high utility sequence mining (HUSM) in quantitative sequence databases (QSDBs) is more general than that of mining frequent sequences in sequence databases. An important limitation of HUSM is that a user-predefined minimum utility threshold is used to decide if a sequence is high utility. However, this is not suitable for many real-life applications as sequences may differ in importance. Another limitation of HUSM is that data in QSDBs are assumed to be precise. But in the real world, data collected by sensors, or other means, may be uncertain. Thus, this paper proposes a framework for mining high utility-probability sequences (HUPSs) in uncertain QSDBs (UQSDBs) with multiple minimum utility thresholds using a minimum utility. Two new width and depth pruning strategies are also introduced to eliminate low utility or low probability sequences as well as their extensions early, and to reduce the sets of candidate items for extensions during the mining process. Based on these strategies, a novel efficient algorithm named HUPSMT is designed for discovering HUPSs. Finally, an experimental study conducted with both real-life and synthetic UQSDBs shows the performance of HUPSMT in terms of time and memory consumption.

**Keywords.** High utility-probability sequence; Uncertain quantitative sequence database; Upper and lower-bounds; Width and depth pruning strategies.

## 1. INTRODUCTION

Discovering frequent itemsets in transaction databases and frequent sequences in sequence databases (SDBs) are important problems in knowledge discovery in databases (DBs), where the support (occurrence frequency) of patterns is used as measure of interest. However, in real-life (e.g. in business), other criteria, such as the utility (e.g. profit yield by a pattern), are more important than the frequency. Hence, traditional algorithms for mining frequent patterns may miss many important patterns that are infrequent but have a high utility. To overcome this limitation of the frequent pattern mining model, it was proposed to discover high utility patterns in quantitative DB, where each item is associated with a quantity, (internal utility, e.g. indicating the number of items purchased by customer or the time spent on a webpage), and each item has an external utility (e.g. unit profit). Then, based on these two basic utilities, the utility of an item, itemset and sequence can be defined using different

utility functions. The utility measure is more general than the support [20]. A pattern is called high utility (HU) if its utility is no less than a user-specified minimum utility threshold $mu$. In quantitative transaction databases (QTDBs), the utility can be defined using the summation [13, 21] or average form [9, 16]. During the last decade, the problem of high utility sequence mining (HUSM) in quantitative sequence databases (QSDBs) attracted the interest of many researchers and has numerous real-life applications, such as analyzing web logs [1], mobile commerce data [15], gene regulation data [22], and healthcare activity-cost event log data [6]. In the problem of high utility itemset mining (HUIM) in QTDBs, each itemset has a unique utility value, because an itemset can appear at most once in each input transaction. This is different from QSDBs, where itemsets are sequentially ordered (e.g. by time), and a sequence may appear multiple times in each input quantitative sequence. Thus, the utility of a sequence may be calculated in many different ways, and utility calculations in HUSM are more time-consuming than in HUIM and frequent itemset/sequence mining (FIM/FSM).

In FIM/FSM, the support measure satisfies the anti-monotonic ($AM$, or downward-closure) property, a very effective property to reduce the search space. This property states that the support of a pattern $\alpha$ is no less than that of any of its super-patterns $\beta$, i.e. $\text{supp}(\alpha) \geq \text{supp}(\beta)$. Consequently, for a minimum support threshold $ms$, if $\alpha$ is infrequent, i.e. $\text{supp}(\alpha) < ms$, then $\beta$ is also infrequent, and all its super-sequences can be immediately pruned.

A key challenge in HUSM is that, in general, the nice $AM$ property does not hold for a utility measure $u$ such as the sum, maximum or minimum of utilities in HUSM [2, 10, 15, 17]. To deal with this problem, a well-known upper-bound (UB) on $u$ that satisfies $AM$, named the SWU (Sequence-Weighted Utility) [20], has been proposed to prune unpromising patterns. However, for low minimum utility thresholds, that UB is often too large and its pruning effect is thus weak. To overcome this limitation, many tighter UBs satisfying anti-monotone-like properties that can be weaker than $AM$ have been proposed to prune low utility candidates at an early stage. These include SPU and SRU [19], CRoM [4], PEU and RSU [18], and MEU [12] for the maximum utility $u_{\max}$ function, and RBU and LRU for the minimum utility $u_{\min}$ function [17].

However, HUSM has the two following important limitations: *First*, high utility sequences (HUSs) in HUSM are only considered w.r.t. a single minimum utility $mu$ threshold. This is not reasonable in many real-life applications where patterns can differ in importance. *Second*, HUSM assumes that data in QSDBs are precise, so it cannot be used in uncertain QSDBs (UQSDBs) based on the expected support model [5]. Each input sequence collected by sensors in a wireless network, for example, is associated with a probability, because data collected by sensors can be affected by environmental noise (e.g. temperature and humidity) and is therefore more or less accurate. For more details on the motivation and signification of the problem, see [3, 11, 23]. To address these issues, the problem of discovering high utility sequences in QSDBs with multiple minimum utility thresholds has been proposed in [12], where items appearing in QSDBs are associated with different minimum utility thresholds. The problem of mining all high utility-probability sequences (HUPSs) in UQSDBs has been considered in [6]. The maximum $u_{\max}$ utility is used in these two problems. This paper considers the more general problem of mining all high utility-probability sequences (w.r.t. $u_{\min}$) in *UQSDBs* with multiple mu thresholds ($\mathcal{HUPSM}$).

The rest of this paper is organized as follows. Section 2 defines the $\mathcal{HUPSM}$ problem. In Section 3, we propose two depth and width pruning strategies to reduce the search space, and a novel algorithm named HUPSMT (High Utility-Probability Sequence mining with Multiple minimum utility Thresholds) for efficiently mining all HUPSs. An experimental study with both real-life and synthetic UQSDBs is conducted in Section 4 to show the performance of the proposed algorithm. Finally, Section 5 draws conclusions and discusses future work.

## 2. PROBLEM DEFINITION

This section presents the problem of $\mathcal{HUPSM}$, high utility-probability sequence mining in uncertain quantitative sequence databases with multiple *mu* thresholds.

Let $\mathcal{A} = \{a_1, a_2, , a_M\}$ be a set of distinct items. A subset $E$ of these items, $E \subseteq \mathcal{A}$, is called an itemset. Without loss of generality, we assume that items in itemsets are sorted according to a total order relation $\prec$ such as the lexicographical order. A *sequence* $\alpha$ is a list of itemsets $E_k$, $k = 1, 2, .., p$, denoted as $\alpha = E_1 \rightarrow E_2 \rightarrow ... \rightarrow E_p$. In a quantitative database, each item $a$ is associated with an external utility $p(a)$, such as its unit profit, that is a positive real number ($p(a) \in R_+$). A quantitative-item (or briefly $q-item$) is a pair $(a, q)$ of an item $a$ and a positive quantity $q$ (internal utility, e.g. purchase quantity). A $q-itemset$ $E'$, according to an itemset $E$, is a set of $q-items$, $E' \stackrel{\text{def}}{=} \{(a_i, q_i)|a_i \in E, q_i \in R_+\}$, where $E$ is called a projected itemset of $E'$ and denoted as $E = proj(E')$. A $q-sequence$ $\alpha'$ is a list of $q-itemsets$ $E'_k, k = 1, ..., p$, denoted as $\alpha' = E'_1 \rightarrow E'_2 \rightarrow ... \rightarrow E'_p$. Let $length(\alpha') \stackrel{\text{def}}{=} \sum_{k=1..p} |E'_k|$, $size(\alpha') \stackrel{\text{def}}{=} p$, where $|E'_k|$ is the number of items in $E'_k$. If $size(\alpha') = 0$, we obtain the null $q$-sequence, denoted as $\langle\rangle$. An uncertain quantitative sequence database (UQSDB) $\mathcal{D}'$ is a finite set of input $q$-sequences, $\mathcal{D}' = \{\psi'_i,\ i = 1, ..., N\}$, where each $q$-sequence $\psi'_i$ is associated with a probability $P(\psi'_i)$ and a unique sequence identifier, $P(\psi'_i) \in (0; 1]$ and $SID = i$. The projected sequence $\alpha$ of a $q$-sequence $\alpha'$ is defined and denoted as $\alpha = proj(\alpha') \stackrel{\text{def}}{=} proj(E'_1) \rightarrow proj(E'_2) \rightarrow ... \rightarrow proj(E'_p)$. For brevity, we define $\alpha'[k] \stackrel{\text{def}}{=} E'_k$, $\alpha[k] \stackrel{\text{def}}{=} proj(E'_k)$. The projected sequence database (SDB) $\mathcal{D}$ of $\mathcal{D}'$ is defined as $\mathcal{D} = proj(\mathcal{D}') \stackrel{\text{def}}{=} \{proj(\psi'_i)|\psi'_i \in \mathcal{D}'\}$. For the convenience of readers, Table 1 summarizes the notation used in the rest of this paper to denote $(q-)$ items, $(q-)$ itemsets, $(q-)$ sequences and input $q$-sequences.

**Definition 1** (Utility of $q$-elements). The utilities of a $q-item$ $(a, q)$, $q$-itemset $E' = \{(a_{i_1}, q_{i_1}), ..., (a_{i_m}, q_{i_m})\}$, $q$-sequence $\alpha'$ and $\mathcal{D}'$ are defined and denoted as $u((a, q)) \stackrel{\text{def}}{=} p(a)*q$, $u(E') \stackrel{\text{def}}{=} \sum_{j=1..m} u((a_{i_j}, q_{i_j}))$, $u(\alpha') \stackrel{\text{def}}{=} \sum_{i=1..p} u(E'_i)$ and $u(\mathcal{D}') \stackrel{\text{def}}{=} \sum_{\psi' \in \mathcal{D}'} u(\psi')$, respectively.

To avoid repeatedly calculating the utility $u$ of each $q-item$ $(a, q)$ in all $q$-sequences $\psi'$ of $\mathcal{D}'$, we calculate all utility values once, and replace $q$ in $\psi'$ by $u((a, q)) = p(a) * q$. This leads to an equivalent database representation of the UQSDB $\mathcal{D}'$ that is called the integrated UQSDB of $\mathcal{D}'$. For brevity, it is also denoted as $\mathcal{D}'$. Due to space limitations, only integrated UQSDBs are considered in this paper. An integrated UQSDB is depicted in Table 2, which will be used as the running example. The utility of $\alpha' = (d, 50) \rightarrow (a, 4)(c, 10)(f, 36)$ is $u(\alpha') = 50 + 4 + 10 + 36 = 100$.

*Table 1.* Notation

| Type | Representation | Example |
|---|---|---|
| Item | Roman letter | $a$, $b$, $c$ |
| q-item | (Roman letter, number) | $(a, 2)$, $(b, 5)$, $(c, 3)$ |
| Itemset | Capitalized roman letter | $A$, $B$, $C$ |
| q-Itemset | Capitalized roman letter followed by $'$ | $A'$, $B'$, $C'$ |
| Sequence | Greek letter | $\alpha$, $\beta$, $\gamma$ |
| q-sequence | Greek letter followed by $'$ | $\alpha'$, $\beta'$, $\gamma'$ |
| Input sequence | Captialized Greek letter | $\psi$, $\psi_{index}$ |
| Input q-sequence | Captialized Greek letter followed by $'$ | $\psi'$, $\psi'_{index}$ |

*Table 2.* Integrated UQSDB $\mathcal{D}'$

| SID | Input $q$-sequence | Probability |
|---|---|---|
| $\psi'_1$ | $(c,5)(e,6) \rightarrow (a,3) \rightarrow (d,50) \rightarrow (a,5)(c,40) \rightarrow (a,4)(c,10)(f,36)$ | 0.5 |
| $\psi'_2$ | $(b,12) \rightarrow (c,20)(e,6) \rightarrow (d,20) \rightarrow (a,1)(f,9)$ | 0.2 |
| $\psi'_3$ | $(d,8) \rightarrow (a,7)(c,35)(e,15) \rightarrow (g,50) \rightarrow (a,9)(f,72)$ | 0.9 |

Let $\alpha' = E'_1 \rightarrow E'_2 \rightarrow ... \rightarrow E'_p$, $\beta' = F'_1 \rightarrow F'_2 \rightarrow ... \rightarrow F'_q$ be two arbitrary $q$-sequences, and $\alpha = E_1 \rightarrow E_2 \rightarrow ... \rightarrow E_p$, $\beta = F_1 \rightarrow F_2 \rightarrow ... \rightarrow F_q$ be their respective projected sequences.

**Definition 2** (Extensions of a sequence). The $i-$extension (or $s-$extension) of $\alpha$ and $\beta$ is defined and denoted as $\alpha \diamond_i \beta \overset{\text{def}}{=} E_1 \rightarrow E_2 \rightarrow ... \rightarrow (E_p \cup F_1) \rightarrow F_2 \rightarrow ... \rightarrow F_q$, where $a \prec b, \forall a \in E_p, \forall b \in F_1$ (or $\alpha \diamond_s \beta \overset{\text{def}}{=} E_1 \rightarrow E_2 \rightarrow ... \rightarrow E_p \rightarrow F_1 \rightarrow F_2 \rightarrow ... \rightarrow F_q$, respectively). A forward extension (or briefly extension) of $\alpha$ with $\beta$, denoted as $\gamma = \alpha \diamond \beta$, can be either $\alpha \diamond_i \beta$ or $\alpha \diamond_s \beta$. Moreover, any sequence $\beta = \alpha \diamond y$ where $\alpha$ is a non-null prefix can be extended in a backward manner using a sequence $\varepsilon$. The sequence $\gamma = \alpha \diamond \varepsilon \diamond y$ such that $\gamma \sqsupseteq \beta$ is called a backward extension of $\beta$ (by $\varepsilon$ w.r.t. the last item $y = lastItem(\beta)$). Note that if $\gamma = \alpha \diamond_i \varepsilon \diamond_i y$ and $size(\varepsilon) = 1$, then $\gamma \sqsupseteq \alpha \diamond_i y$, otherwise, $\gamma \sqsupseteq \alpha \diamond_s y$.

For instance, $d \rightarrow af$ and $d \rightarrow a \rightarrow c$ are respectively $i-$ and $s-$extensions of $d \rightarrow a$; $d \rightarrow acf$, $d \rightarrow a \rightarrow acf$ and $d \rightarrow ac \rightarrow g \rightarrow af$ are backward extensions of $d \rightarrow af$.

**Definition 3** (Partial order relations over $q$-sequences and sequences). Consider any two $q$-itemsets $E' = \{(a_{i_1}, q_{i_1}), ..., (a_{i_m}, q_{i_m})\}$, $F' = \{(a_{j_1}, q_{j_1}), ..., (a_{j_n}, q_{j_n})\}$, $m \leq n$. The $q$-itemset $E'$ is said to be contained in $F'$ and denoted as $E' \sqsubseteq F'$, if there exist natural numbers $1 \leq k_1 < k_2 < ... < k_m \leq n$ such that $a_{i_l} = a_{j_{k_l}}$ and $q_{i_l} = q_{j_{k_l}}, \forall l = 1, ..., m$. Then, $\alpha'$ is said to be contained in $\beta'$ and denoted as $\alpha' \sqsubseteq \beta'$ (or $\beta'$ is called a super-$q$-sequence of $\alpha'$) if $p \leq q$ and there exist $p$ positive integers, $1 \leq j_1 < j_2 < ... < j_p \leq q : E'_k \sqsubseteq F'_{j_k}, \forall k = 1, ..., p$; and $\alpha' \sqsubset \beta' \Leftrightarrow (\alpha' \sqsubseteq \beta' \wedge \alpha' \neq \beta')$. Similarly, for simplicity, we also use $\sqsubseteq$ to define the containment relation over all sequences as follows: $\alpha \sqsubseteq \beta$ or $\beta \sqsupseteq \alpha$ ($\beta$ is called a super-sequence of $\alpha$) if there exist $p$ positive integers, $1 \leq j_1 < j_2 < ... < j_p \leq q : E_k \sqsubseteq F_{j_k}, \forall k = 1, ..., p$, and $\alpha \sqsubset \beta \Leftrightarrow (\alpha \sqsubseteq \beta \wedge \alpha \neq \beta)$. The $q$-sequence $\beta'$ contains the sequence $\alpha$ (or $\alpha$ is a sub-sequence of $\beta'$), denoted as $\alpha \sqsubseteq \beta'$ or $\beta' \sqsupseteq \alpha$, if $proj(\beta') \sqsupseteq \alpha$. Let

$\rho(\alpha) \stackrel{\text{def}}{=} \{\psi' \in \mathcal{D}' | \psi' \sqsupseteq \alpha\}$ denote the set of all input $q$-sequences containing $\alpha$. The support of $\alpha$ is defined as the number of super-$q$-sequences of $\alpha$, that is $\text{supp}(\alpha) = |\rho(\alpha)|$.

For example, for $\beta = d \to ac \to af$ and $\psi_3 = proj(\psi'_3) = d \to ace \to g \to af$, then $\psi'_3 \sqsupseteq \beta$. Similarly, $\psi'_1 \sqsupseteq \beta$ and $\rho(\beta) = \{\psi'_1, \psi'_3\}$, so $\text{supp}(\beta) = 2$. Note that a sequence may have multiple occurrences in an input $q$-sequence. For instance, $\alpha = d \to ac$ appears twice in $\psi'_1$, because $(d, 50) \to (a, 5)(c, 40) \sqsupseteq \alpha$ and $(d, 50) \to (a, 4)(c, 10) \sqsupseteq \alpha$ with two different utility values (95 and 64).

Let $U(\alpha, \psi'_i) \stackrel{\text{def}}{=} \{\alpha' | \alpha' \sqsubseteq \psi'_i \wedge proj(\alpha') = \alpha\}$ be the set of all occurrences $\alpha'$ of $\alpha$ in $\psi'_i$. Because this set may contain more than one occurrence, the utility of $\alpha$ in $\psi'_i$ can be defined in many different ways. For example, it can be calculated as the maximum or minimum of the utilities of $\alpha$ in $\psi'_i$, as in many studies [4, 12, 17, 18, 19]. Formally, they are defined as follows.

**Definition 4** (Minimum utility of sequences [17]). The minimum utility of a sequence $\alpha$ in an input $q$-sequence $\psi'_i$ (or in $\mathcal{D}'$) is defined and denoted as $u_{\min}(\alpha, \psi'_i) \stackrel{\text{def}}{=} \min\{u(\alpha') | \alpha' \in U(\alpha, \psi'_i)\}$ (or $u_{\min}(\alpha, \mathcal{D}')$ or more briefly $u_{\min}(\alpha) \stackrel{\text{def}}{=} \sum_{\psi'_i \in \rho(\alpha)} u_{\min}(\alpha, \psi'_i)$). As a convention, we define $u_{\min}(\langle\rangle, \psi'_i) \stackrel{\text{def}}{=} u(\psi'_i),\ \forall \psi'_i \in \mathcal{D}'$.

Similarly, we also have the definition of the maximum utility of $\alpha$ in $\psi'_i$ (or in $\mathcal{D}'$) [20], $u_{\max}(\alpha, \psi'_i) \stackrel{\text{def}}{=} \max\{u(\alpha') | \alpha' \in U(\alpha, \psi'_i)\}$ (or $u_{\max}(\alpha) \stackrel{\text{def}}{=} \sum_{\psi'_i \in \rho(\alpha)} u_{\max}(\alpha, \psi'_i)$). In this paper, we consider the minimum $u_{\min}$ utility. The reason for using $u_{\min}$ and its advantages compared to $u_{\max}$ were discussed in [17].

For example, for $\alpha = d \to ac$, we have $\rho(\alpha) = \{\psi'_1, \psi'_3\}$ and $U(\alpha, \psi'_1) = \{(d, 50) \to (a, 5)(c, 40), (d, 50) \to (a, 4)(c, 10)\}$, so $u_{\min}(\alpha, \psi'_1) = \min\{95, 64\} = 64$. Similarly, $u_{\min}(\alpha, \psi'_3) = 50$. Hence, $u_{\min}(\alpha) = 114$. Besides, for another $\alpha = ce \to f$, $\beta = ce \to af$ and $\delta = ce \to a \to f$, then $\delta \sqsupset \alpha \sqsubset \beta$ and $u_{\min}(\beta) = 218 > u_{\min}(\alpha) = 204 > u_{\min}(\delta) = 50$. In other words, $u_{\min}$ is neither anti-monotonic nor monotonic. In this context, a measure $u$ of sequences is said to be anti-monotonic or briefly $AM$ (or monotonic) if $u(\beta) \leq u(\alpha)$ (or $u(\beta) \geq u(\alpha)$, respectively), for any sequences $\alpha$ and $\beta$ such that $\beta \sqsupseteq \alpha$.

Unlike the support measure, the maximum and minimum utility functions are not $anti-monotonic$. Thus, it is necessary to devise UBs satisfying $AM$ or weaker properties to efficiently reduce the search space. For example, USpan [19, 20] is a popular and well-known, but unfortunately incomplete, algorithm for mining high utility sequences (w.r.t. $u_{\max}$). The reason is that USpan utilizes a measure named SPU to deeply prune candidate sequences, but the SPU is not an UB on $u_{\max}$ (see more details in [17]). Other UBs on $u_{\max}$ (or $u_{\min}$) are REU and LAS [12] (or RBU and LRU [17], respectively).

**Definition 5** (Minimum utility threshold of sequences). Let $Mu \stackrel{\text{def}}{=} \{mu(x), x \in \mathcal{A}\}$ be the set of minimum utility thresholds of all items in $\mathcal{A}$. Then, the minimum utility threshold of a sequence $\alpha$ is defined and denoted as $mu(\alpha) \stackrel{\text{def}}{=} \min\{mu(x), x \in \alpha\}$.

For instance, consider minimum utility thresholds of all items in $\mathcal{A}$ as shown in Table 3 and $\beta = d \to ac \to af$. Then, $mu(\beta) = \min\{320, 260, 270, 350\} = 260$.

**Definition 6** (Probability of sequences). The probability of a sequence $\alpha$ in $\mathcal{D}'$ is defined

*Table 3.* Minimum utility thresholds of items

| Item | $a$ | $b$ | $c$ | $d$ | $e$ | $f$ | $g$ |
|---|---|---|---|---|---|---|---|
| $mu(it)$ | 260 | 5 | 270 | 320 | 50 | 350 | 31 |

and denoted as $P(\alpha) \stackrel{\text{def}}{=} \sum_{\psi'_i \in \rho(\alpha)} P(\psi'_i)/PS$, where $PS \stackrel{\text{def}}{=} \sum_{\psi'_i \in \mathcal{D}'} P(\psi'_i)$ is a standardized coefficient. Then, $P(\alpha) \in [0, 1]$.

For example, for $\beta = d \to ac \to af$, then $\rho(\beta) = \{\psi'_1, \psi'_3\}$ and $PS = 1.6$, so $P(\beta) = 1.4/1.6 = 0.875$.

**Problem Definition**. For a user-predefined minimum probability threshold $mp$ and minimum utility thresholds $Mu$, a sequence $\alpha$ is said to be a high utility-probability (HUP) sequence if $u_{\min}(\alpha) \geq mu(\alpha)$ and $P(\alpha) \geq mp$. The problem of high utility-probability sequence mining ($\mathcal{HUPSM}$) in a UQSDB $\mathcal{D}'$ with multiple minimum utility thresholds is to discover the set $\mathcal{HUPS} \stackrel{\text{def}}{=} \{\alpha | u_{\min}(\alpha) \geq mu(\alpha) \wedge P(\alpha) \geq mp\}$.

For example, for $mp = 0.875$, $Mu$ of Table 2 and $\beta = d \to ac \to af$, then $\rho(\beta) = \{\psi'_1, \psi'_3\}, u_{\min}(\beta) = u_{\min}(\beta, \psi'_1) + u_{\min}(\beta, \psi'_3) = 135 + 131 = 266$, so $u_{\min}(\beta) \geq mu(\beta)$ and $P(\beta) \geq mp$. Hence, $\beta$ is a HUP sequence.

## 3. PRUNING STRATEGIES AND PROPOSED ALGORITHM

### 3.1. Prunning strategies

Since $u_{\min}$ is not anti-monotonic ($AM$), devising upper-bounds satisfying anti-monotone-like properties that can be weaker than $AM$ is necessary and useful to efficiently reduce the search space.

Firstly, we introduce the concepts of ending and remaining $q$-sequence of a sub-sequence in a q-sequence. Assume that $\alpha = E_1 \to E_2 \to ... \to E_p \sqsubseteq \beta' = F'_1 \to F'_2 \to ... \to F'_q$, i.e. there exist $p$ positive integers, $1 \leq i_1 < i_2 < ... < i_p \leq q : E_k \sqsubseteq proj(F'_{i_k}), \forall k = 1, ..., p$. Then, the index $i_p$ is said to be an ending of $\alpha$ in $\beta'$, denoted as $end(\alpha, \beta')$ and the last item of $\alpha$ in $F'_{i_p}$ is called an ending item and denoted as $e_{i_p}$. The remaining $q$-sequence of $\alpha$ in $\beta'$ w.r.t. the ending $i_p$ is the rest of $\beta'$ after $\alpha$ (or after the ending item $e_{i_p}$) and denoted as $rem(\alpha, \beta', i_p)$. Let $i^*_p \stackrel{\text{def}}{=} FEnd(\alpha, \beta')$ denote the first ending of $\alpha$ in $\beta'$, $e_{i^*_p} \stackrel{\text{def}}{=} FEItem(\alpha, \beta')$ - the first ending item of $\alpha$ in $\beta'$, and $ub_{\min}(\alpha, \beta') \stackrel{\text{def}}{=} u(\alpha, \beta', i^*_p) + u(rem(\alpha, \beta', i^*_p))$ as an upper-bound on $u_{\min}(\alpha, \beta')$ for $\alpha \neq \langle\rangle$, and $ub_{\min}(\langle\rangle, \beta') \stackrel{\text{def}}{=} u(\beta')$ if $\alpha = \langle\rangle$, where $u(\alpha, \beta', i^*_p) \stackrel{\text{def}}{=} \min\{u(\alpha') | \alpha' \in U(\alpha, \beta') \wedge end(\alpha, \beta') = i^*_p\}$. If $\alpha = \langle\rangle$, then as a convention, $i^*_p = (\langle\rangle, \beta') \stackrel{\text{def}}{=} 0$ and $rem(\langle\rangle, \beta', i_p) = \beta'$.

For instance, the sequence $\gamma = a \to ac$ has two endings of 4 and 5 in $\psi'_1$, so its first ending $i^*_p = FEnd(\gamma, \psi'_1)$ is 4, $rem(\gamma, \psi'_1, i^*_p) = (a, 4)(c, 10)(f, 36)$, $rem(\gamma, \psi'_1, 5) = (f, 36)$, $u(\gamma, \psi'_1, i^*_p) = u((a, 3) \to (a, 5)(c, 40)) = 48$ and $u(\alpha, \psi'_1, 5) = \min\{u((a, 3) \to (a, 4)(c, 10)), u((a, 5) \to (a, 4)(c, 10))\} = 17$.

### 3.1.1. Designing upper-bounds on $u_{\min}$

**Definition 7** (Upper-bounds on $u_{\min}$).

a. A measure $ub$ (of sequences) is said to be an upper-bound (UB) on $u_{\min}$, denoted as $u_{\min} \ll ub$, if $u_{\min}(\alpha) \leq ub(\alpha),\ \forall \alpha$.

b. For two measures $ub_1$ and $ub_2$, $ub_1$ is said to be tighter than $ub_2$, denoted as $ub_1 \ll ub_2$, if $ub_1(\alpha) \leq ub_2(\alpha),\ \forall \alpha$. Given two UBs on $u_{\min}$, $ub_1$ and $ub_2$, $ub_1$ is called tighter than $ub_2$ if $u_{\min} \ll ub_1 \ll ub_2$.

c. ($UBs$ on $u_{\min}$ [17]) For any sequence $\alpha$ and its extension sequence $\beta = \alpha \diamond y$, we define and denote three UBs on $u_{\min}$, $SWU$ (Sequence-Weighted Utility), $RBU$ (Remaining-Based Utility) and $LRU$ (Looser Remaining Utility), as $SWU(\alpha) \stackrel{\text{def}}{=} \sum_{\psi'_i \in \rho(\alpha)} u(\psi'_i)$, $RBU(\alpha) \stackrel{\text{def}}{=} \sum_{\psi'_i \in \rho(\alpha)} ub_{\min}(\alpha, \psi'_i)$ and $LRU(\beta) \stackrel{\text{def}}{=} \sum_{\psi'_i \in \rho(\beta)} ub_{\min}(\alpha, \psi'_i)$. Obviously, if $\alpha = \langle\rangle$, $LRU(y) = SWU(y),\ \forall y \in \mathcal{A}$.

The $SWU$ UB was proposed in [20], and two new tighter $LRU$ and $RBU$ UBs on $u_{\min}$ were presented in [17]. As shown in the following theorem, the two $LRU$, $RBU$ UBs are tighter than $SWU$, but their pruning ability is weaker compared to the largest $SWU$ UB.

**Theorem 1** (Anti-monotone-like ($\mathcal{AML}$) properties of RBU, LRU and SWU UBs on $u_{\min}$ [17]).

a. *$u_{\min} \ll RBU \ll LRU \ll SWU$, i.e. $SWU$, $LRU$ and $RBU$ are gradually tighter $UBs$ on $u_{\min}$.*

b. *(i) $\mathcal{AM}(SWU)$ or $SWU$ is anti-monotonic, i.e. $SWU(\beta) \leq SWU(\alpha)$ for any super-sequence $\beta$ of $\alpha$, $\beta \sqsupseteq \alpha$.*

*(ii) $\mathcal{AMF}(RBU)$ or $RBU$ is anti-monotonic w.r.t. forward extension, i.e. $RBU(\beta) \leq RBU(\alpha)$ for any forward extension $\beta = \alpha \diamond \delta$ of $\alpha$ (with $\delta$).*

*(iii) $\mathcal{AMB}i(LRU)$ or $LRU$ is anti-monotonic w.r.t. bi-direction extension, i.e. $\mathcal{AMF}(LRU)$ and for any backward extension $\gamma = \alpha \diamond \varepsilon \diamond y$ of $\delta = \alpha \diamond y$, if $\gamma = \alpha \diamond_i \varepsilon \diamond_i y$ and $size(\varepsilon) = 1$, then $LRU(\gamma) \leq LRU(\alpha \diamond_i y)$, otherwise, $LRU(\gamma) \leq LRU(\alpha \diamond_s y)$.*

It is observed that, for any UB $ub$ on $u_{\min}$, $\mathcal{AM}(ub) \Rightarrow \mathcal{AMB}i(ub) \Rightarrow \mathcal{AMF}(ub)$, i.e. the three anti-monotone-like properties $\mathcal{AM}$, $\mathcal{AMB}i$ and $\mathcal{AMF}$ are gradually weaker.

For example, for an $i$-extension $\beta = c \to ac = \alpha \diamond_i c$ of $\alpha = c \to a$ with $c$, since $\rho(\beta) = \{\psi'_1\}$, $u_{\min}(\beta) = u_{\min}(\beta, \psi'_1) = \min\{u((c,5) \to (a,5)(c,40)), u((c,5) \to (a,4)(c,10)), u((c,40) \to (a,4)(c,10))\} = 19$. Besides, $i^*_p = FEnd(\beta, \psi'_1) = 2, u(\beta, \psi'_1, i^*_p) = u((c,5) \to (a,5)(c,40)) = 50$ and $u(rem(\beta, \psi'_1, i^*_p)) = u((a,4)(c,10)(f,36)) = 50$, so $RBU(\beta) = ub_{\min}(\beta, \psi'_1) = 50 + 50 = 100$ and similarly, $LRU(\beta) = ub_{\min}(\alpha, \psi'_1) = u((c,5) \to (a,3)) + u((d,50) \to (a,5)(c,40) \to (a,4)(c,10)(f,36)) = 8 + 145 = 153, SWU(\beta) = u(\psi'_1) = 159$. Thus, $u_{\min}(\beta) < RBU(\beta) < LRU(\beta) < SWU(\beta)$. Moreover, in the same way, since $\rho(\alpha) = \{\psi'_i, i = 1,2,3\}$, $SWU(\alpha) = \sum_{i=1,2,3} u(\psi'_i) = 159 + 68 + 196 = 423 > SWU(\beta)$, $LRU(\alpha) =$

$159 + 56 + 181 = 396 > LRU(\beta)$ and $RBU(\alpha) = \sum_{i=1,2,3} ub_{\min}(\alpha, \psi'_i) = 153 + 30 + 116 = 299 > RBU(\beta)$.

Similarly, since the *mu* measure of sequences in Definition 5 is not monotonic, devising its lower-bounds (LBs) to satisfy $monotone-like$ ($\mathcal{ML}$) properties that can be weaker than the *monotonic* property is also useful to efficiently reduce the search space. As shown in the Remarks and Discussion section below, designing such LBs is important. That is, missing some lower-bounds or using them incorrectly may result in false results.

### 3.1.2. Designing lower-bounds on $u_{\min}$

For any two items $x$ and $z$ in $\psi'$, we write $x \lhd z$ if $z$ follows $x$, and $x \trianglelefteq z$ if either $z$ is $x$ or $x \lhd z$. For example, since $a$ appears firstly in the $2^{nd}$ itemset of $\psi'_3$, we have $FEItem(a, \psi'_3) = a_2$, where $x_i$ indicates that item $x$ appears in the $i^{th}$ itemset of $\psi'_3$. In $\psi'_3$, the set $\{x | a_2 \lhd x \trianglelefteq f_4\}$ of all items which follow $a_2$ and do not follow $f_4$ are $c_2$, $e_2$, $g_3$, $a_4$ and $f_4$. Then, three lower-bounds (LBs) on $\min Mes$, $lbF$ (LB monotone w.r.t. forward extension), $lbBi$ (looser LB monotone w.r.t. bi-direction extension) and $lbM$ (LB monotone), can be defined as follows.

**Definition 8** (Lower-bounds on *mu*).

a. A measure $lb$ (of sequences) is said to be a lower-bound (LB) on *mu*, denoted as $lb \ll mu$, if $mu(\alpha) \geq lb(\alpha)$, $\forall \alpha$. Given two LBs on *mu*, $lb_1$ and $lb_2$, $lb_1$ is called tighter than $lb_2$ if $lb_2 \ll lb_1 \ll mu$.

b. (LBs on *mu*) For any sequence $\alpha$ and its extension sequence $\beta = \alpha \diamond y$, we define and denote three LBs on *mu* as $lbF(\alpha) \stackrel{\text{def}}{=} \min\{mu(x) | x \in \alpha \vee (x \in \psi' \wedge \psi' \in \rho(\alpha) \wedge e_{i_p^*} \lhd x)\}$, $lbM(\alpha) \stackrel{\text{def}}{=} \min\{mu(x) | x \in \psi' \wedge \psi' \in \rho(\alpha)\}$, $lbBi(\beta) \stackrel{\text{def}}{=} \{mu(x) | x \in \alpha \vee (x \in \psi' \wedge \psi' \in \rho(\beta) \wedge e_{i_p^*} \lhd x)\}$ if $\alpha \neq \langle\rangle$ and $lbBi(y) \stackrel{\text{def}}{=} lbM(y)$ if $\alpha = \langle\rangle$, where $e_{i_p^*} \stackrel{\text{def}}{=} FEItem(\alpha, \psi')$.

The following theorem states that $lbM$, $lbBi$ and $lbF$ are gradually tighter LBs on *mu* that satisfy gradually *weaker* monotone-like properties ($\mathcal{M}$, $\mathcal{MB}i$ and $\mathcal{MF}$).

**Theorem 2** (Monotone-like properties ($\mathcal{ML}$) of LBs on *mu*).

*a.* $lbM \ll lbBi \ll lbF \ll mu$ *, i.e.* $lbM$*,* $lbBi$ *and* $lbF$ *are gradually tighter LBs on mu.*

*b.* $\mathcal{AM}(mu)$ *or mu is anti-monotonic, i.e.* $mu(\beta) \leq mu(\alpha)$*, for any super-sequence* $\beta$ *of* $\alpha$*,* $\beta \sqsupseteq \alpha$*.*

*c.* *(i)* $\mathcal{M}(lbM)$ *or* $lbM$ *is monotonic, i.e.* $lbM(\beta) \geq lbM(\alpha)$*, for any super-sequence* $\beta$ *of* $\alpha$*,* $\beta \sqsupseteq \alpha$*.*

*(ii)* $\mathcal{MF}(lbF)$ *or* $lbF$ *is monotonic w.r.t. forward extension, i.e.* $lbF(\beta) \geq lbF(\alpha)$*, for any forward extension* $\beta = \alpha \diamond \delta$ *of* $\alpha$*.*

*(iii)* $\mathcal{MB}i(lbBi)$ *or* $lbBi$ *is monotonic w.r.t. bi-direction extension, i.e.* $\mathcal{MF}(lbBi)$ *and for any backward extension* $\gamma = \alpha \diamond \varepsilon \diamond y$ *of* $\delta = \alpha \diamond y$*, if* $\gamma = \alpha \diamond_i \varepsilon \diamond_i y$ *and* $size(\varepsilon) = 1$*, then* $lbBi(\gamma) \geq lbBi(\alpha \diamond_i y)$*, otherwise,* $lbBi(\gamma) \geq lbBi(\alpha \diamond_s y)$*.*

*Obviously, for any LB lb on mu, $\mathcal{M}(lb) \Rightarrow \mathcal{MB}i(lb) \Rightarrow \mathcal{MF}(lb)$, i.e. the three monotone-like properties $\mathcal{M}$, $\mathcal{MB}i$ and $\mathcal{MF}$ are gradually weaker.*

*Proof.* For any super-sequence $\beta$ of $\alpha$, $\beta \sqsupseteq \alpha$, since $\{x \in \alpha\} \subseteq \{x \in \beta\}, \rho(\beta) \subseteq \rho(\alpha)$, we have $mu(\beta) \leq mu(\alpha)$ and $lbM(\beta) \geq lbM(\alpha)$, i.e. $\mathcal{AM}(mu)$ and $\mathcal{M}(lbM)$. The assertions b and c.(i) are proven.

Now we will prove two assertions a and c.(ii)-(iii). For any forward extension $\beta$ of $\alpha$, $\beta = \alpha \diamond \delta \sqsupseteq \alpha$ and $\psi' \in \rho(\beta) \subseteq \rho(\alpha), i_p \stackrel{\text{def}}{=} FEnd(\alpha, \psi') \leq i_q \stackrel{\text{def}}{=} FEnd(\beta, \psi')$, so $\{x \in \beta \vee (x \in rem(\beta, \psi', i_q) \wedge \psi' \in \rho(\beta))\} \subseteq \{x \in \alpha \vee (x \in rem(\alpha, \psi', i_p) \wedge \psi' \in \rho(\alpha))\} \supseteq \{x \in \alpha\}$. Thus, $lbF(\alpha) \leq lbF(\beta)$ and $lbF(\alpha) \leq mu(\alpha)$, i.e. $\mathcal{MF}(lbF)$ and $lbF \ll mu$.

Similarly, to prove $\mathcal{MF}(lbBi)$, without loss of generality, we only need to consider any forward extension $\beta = \delta \diamond z$ of $\delta = \alpha \diamond y$ with an item $z$. Then, $\beta \sqsupseteq \delta$ and $\forall \psi' \in \rho(\beta) \subseteq \rho(\delta), FEnd(\alpha, \psi') \leq FEnd(\delta, \psi')$. For $e_{i_p^*} \stackrel{\text{def}}{=} FEItem(\alpha, \psi'), e_{i_q^*} \stackrel{\text{def}}{=} FEItem(\delta, \psi')$, we have $e_{i_p^*} \lhd e_{i_q^*}$, so $S_\beta \subseteq T_\delta \subseteq U_\delta$ and $T_\delta \supseteq R_\delta$, where $S_\beta \stackrel{\text{def}}{=} \{x \in \delta \vee (x \in \psi' \wedge \psi' \in \rho(\beta) \wedge e_{i_q^*} \lhd x)\}$, $T_\delta \stackrel{\text{def}}{=} \{x \in \alpha \vee (x \in \psi' \wedge \psi' \in \rho(\delta) \wedge e_{i_p^*} \lhd x)\}$, $R_\delta \stackrel{\text{def}}{=} \{x \in \delta \vee (x \in \psi' \wedge \psi' \in \rho(\delta) \wedge e_{i_q^*} \lhd x)\}$, $U_\delta \stackrel{\text{def}}{=} \{x \in \psi' \wedge \psi' \in \rho(\delta)\}$. Thus, $lbBi(\delta) \leq lbBi(\beta)$ and $lbM(\delta) \leq lbBi(\delta) \leq lbF(\delta)$, i.e. $\mathcal{MF}(lbBi)$ and $lbM \ll lbBi \ll lbF$.

To prove $\mathcal{MB}i(lbBi)$, consider any backward extension $\gamma = \alpha \diamond \varepsilon \diamond y$ of $\delta = \alpha \diamond y$ such that $\gamma \sqsupset \delta$. Then, $FEnd(\alpha, \psi') \leq FEnd(\alpha \diamond \varepsilon, \psi'), \forall \psi' \in \rho(\gamma) \subseteq \rho(\delta)$. For $e_{i_p^*} \stackrel{\text{def}}{=} FEItem(\alpha, \psi')$, $e_{i_q^*} \stackrel{\text{def}}{=} FEItem(\alpha \diamond \varepsilon, \psi')$, we have $e_{i_p^*} \trianglelefteq e_{i_q^*}$, so $\{x \in \alpha \diamond \varepsilon \vee (x \in \psi' \wedge \psi' \in \rho(\gamma) \wedge e_{i_q^*} \lhd x)\} \subseteq \{x \in \alpha \vee (x \in \psi' \wedge \psi' \in \rho(\delta) \wedge e_{i_p^*} \lhd x)\}$, and $lbBi(\delta) \leq lbBi(\gamma)$. Hence, if $\gamma = \alpha \diamond_i \varepsilon \diamond_i y$ and $size(\varepsilon) = 1$, then $\gamma \sqsupseteq \alpha \diamond_i y$ and $lbBi(\gamma) \geq lbBi(\alpha \diamond_i y)$; otherwise, $\gamma \sqsupseteq \alpha \diamond_s y$ and $lbBi(\gamma) \geq lbBi(\alpha \diamond_s y)$. Thus, $\mathcal{MB}i(lbBi)$. ∎

For example, for $\gamma = af \sqsupset \delta = a$, we have $mu(\gamma) = \min\{mu(a), mu(f)\} = \min\{260; 320\} = 260$. Since $\rho(\gamma) = \rho(\delta) = \mathcal{D}'$, $lbM(\gamma) = \min\{mu(x), x \in \psi'_i, i \in \{1, 2, 3\}\} = 5$, $lbF(\gamma) = \min\{mu(a), mu(f)\} = 260$ and similarly, $lbBi(\gamma) = 31$. Hence, $mu(\gamma) \geq lbF(\gamma) > lbBi(\gamma) > lbM(\gamma)$. In the same way, we also have $lbF(\delta) = 31 < lbF(\gamma)$ and $lbBi(\delta) = lbM(\delta) = 5 \leq lbM(\gamma) < lbBi(\gamma)$.

### 3.1.3. Designing pruning strategies

In the process of mining $\mathcal{HUPS}$, all candidate sequences are stored in a prefix tree that contains the null sequence as its root, where each node represents a candidate sequence, and each child node of a node *nod* is an extension of *nod*. In the following, $branch(\alpha)$ denotes the set consisting of $\alpha$ and all its extensions.

The process of extending a sequence with single items may generate many sequences that do not appear in any input $q$-sequence. Considering these sequences is a waste of time. To deal with this issue, projected databases (PDBs) [14] of sequences are often used. However, creating and scanning multiple PDBs is very costly. To overcome this challenge, it is observed that if $\alpha \diamond_i y$ is a HUP sequence, then $lbBi(\alpha \diamond_i y) \leq mu(\alpha \diamond_i y) \leq u_{\min}(\alpha \diamond_i y) \leq LRU(\alpha \diamond_i y)$ and $P(\alpha \diamond_i y) \geq mp$, i.e. $y$ belongs to the set $I_{LRU,lbBi,P}(\alpha)$ or briefly $I(\alpha) \stackrel{\text{def}}{=} \{y \in \mathcal{A} | y \succ lastItem(\alpha) \wedge LRU(\alpha \diamond_i y) \geq lbBi(\alpha \diamond_i y) \wedge P(\alpha \diamond_i y) \geq mp\}$. Similarly, we define $S_{LRU,lbBi,P}(\alpha) = S(\alpha) \stackrel{\text{def}}{=} \{y \in \mathcal{A} | LRU(\alpha \diamond_s y) \geq lbBi(\alpha \diamond_s y) \wedge P(\alpha \diamond_s y) \geq mp\}$. Then, $I(\alpha)$ and $S(\alpha)$ are two sets of candidate items for $i-$ and $s-extensions$ of $\alpha$.

Note that the $P$ probability is also anti-monotonic and denoted as $\mathcal{AM}(P)$, i.e. $P(\beta) \leq P(\alpha), \forall \beta \sqsupseteq \alpha$. Based on $\mathcal{AM}(P)$, two $\mathcal{AML}$ and $\mathcal{ML}$ properties of pairs $(RBU, lbF)$ and $(LRU, lbBi)$, we can design two depth and width pruning strategies and a tightening strategy as shown in Theorem 3 below. For brevity, denote $DepthPC_{RBU,lbF}(\alpha) \stackrel{\text{def}}{=} (RBU(\alpha) < lbF(\alpha))$ and $WidthPC_{LRU,lbBi,P}(\alpha) \stackrel{\text{def}}{=} (LRU(\alpha) < lbBi(\alpha) \vee P(\alpha) < mp)$ as depth and width pruning conditions, respectively.

**Theorem 3** (Depth, width pruning strategies).

a. *Depth pruning strategy based on RBU and lbF* $\mathcal{DPS}(RBU, lbF)$ *(or briefly* $\mathcal{DPS}$*). If* $DepthPC_{RBU,lbF}(\alpha)$*, then* $u_{\min}(\beta) < mu(\beta)$*, for all (forward) extensions* $\beta$ *of* $\alpha$*, i.e. the* $branch(\alpha)$ *can be deeply pruned.*

b. *Width pruning strategy based on LRU, lbBi and P* $\mathcal{WPS}(LRU, lbBi, P)$ *(or briefly* $\mathcal{WPS}$*). If* $WidthPC_{LRU,lbBi,P}(\beta)$ *then* $(u_{\min}(\gamma) < mu(\gamma) \vee P(\gamma) < mp)$*, for all (forward) extensions* $\gamma$ *of* $\beta$*, i.e. the* $branch(\beta)$ *is deeply pruned. Moreover, we can apply additionally the following Tightening strategy -* $\mathcal{TS}(LRU, lbBi, P)$*:* $I(\alpha \diamond_i x) \subseteq I(\alpha)$ *and* $I(\alpha \diamond_s x) \cup S(\alpha \diamond_i x) \cup S(\alpha \diamond_s x) \subseteq S(\alpha)$*, i.e. the two* $I$ *and* $S$ *sets of candidate items for extensions of sequences are gradually tightened during the mining process.*

*Similarly, we also have* $\mathcal{WPS}(SWU, lbM, P)$*,* $\mathcal{WPS}(SWU, lbM)$ *and* $\mathcal{WPS}(P)$ *according to the width pruning conditions:* $WidthPC_{SWU,lbM,P}(\alpha) \stackrel{\text{def}}{=} (SWU(\alpha) < lbM(\alpha) \vee P(\alpha) < mp)$*,* $WidthPC_{SWU,lbM}(\alpha) \stackrel{\text{def}}{=} (SWU(\alpha) < lbM(\alpha))$ *and* $WidthPC_{P(\alpha)} \stackrel{\text{def}}{=} (P(\alpha) < mp)$*, respectively.*

*Proof.*

a. If $RBU(\alpha) < lbF(\alpha)$, then $\forall \beta = \alpha \diamond \varepsilon \sqsupseteq \alpha$, by Theorem 1 and Theorem 2, $u_{\min}(\beta) \leq RBU(\beta) \leq RBU(\alpha) < lbF(\alpha) \leq lbF(\beta) < mu(\beta)$.

b. If $(LRU(\beta) < lbBi(\beta) \vee P(\beta) < mp)$, then $\forall \gamma = \beta \diamond \varepsilon \sqsupseteq \beta, \rho(\gamma) \subseteq \rho(\beta), u_{\min}(\gamma) \leq LRU(\gamma) \leq LRU(\beta) < lbBi(\beta) \leq lbBi(\gamma) < mu(\gamma)$ or $P(\gamma) \leq P(\beta) < mp$. Since $\mathcal{AMB}i(LRU)$ and $\mathcal{MB}i(lbBi)$, the remaining assertions also hold. Indeed, for example, for any $y \in I(\alpha \diamond_i x)$, then $y \succ x, P(\alpha \diamond_i x \diamond_i y) \geq mp$ and $LRU(\alpha \diamond_i x \diamond_i y) \geq lbBi(\alpha \diamond_i x \diamond_i y)$. Hence, $P(\alpha \diamond_i x) \geq P(\alpha \diamond_i x \diamond_i y) \geq mp$ and $size(x) = 1, LRU(\alpha \diamond_i y) \geq LRU(\alpha \diamond_i x \diamond_i y) \geq lbBi(\alpha \diamond_i x \diamond_i y) \geq lbBi(\alpha \diamond_i x)$, so $y \in I(\alpha)$, i.e. $I(\alpha \diamond_i x) \subseteq I(\alpha)$. The remaining assertions are similarly proven. ∎

For example, for the above sequence $\beta = c \rightarrow ac$, we have $RBU(\beta) = 100$ and $LRU(\beta) = 153$. On other hand, $lbF(\beta) = lbBi(\beta) = 260$. Since $RBU(\beta) < LRU(\beta) < lbBi(\beta) \leq lbF(\beta)$, the whole $branch(\beta)$ is pruned and we can apply the $\mathcal{TS}(LRU, lbBi, P)$ strategy for the sequence $\beta$.

**Remarks and Discussion**

a. The Reducing UQSDB Strategy - $\mathcal{R}ed\mathcal{S}(SWU, lbM, P)$ (or briefly $\mathcal{R}ed\mathcal{S}$, used additionally in $\mathcal{WPS}$). For any item $x$ of $\mathcal{A}$ such that the width pruning condition $WidthPC_{SWU,lbM,P}(x)$ holds, we can apply the following reducing strategy, denoted

as $\mathcal{R}ed(SWU, lbM, P)$: not only the original UQSDB $\mathcal{D}'$ can be reduced by removing all such irrelevant items $x$ from $\mathcal{D}'$, but also values of all bounds of remaining items are updated and can also be tightened. Indeed, since $\mathcal{AM}(P)$, $\mathcal{AM}(SWU)$ and $\mathcal{M}(lbM)$ properties are true, for any sequence $\alpha$ containing $x$, $\alpha = \varepsilon \diamond x \diamond \delta$, we have $P(\alpha) \leq P(x) < mp$ or $u_{\min}(\alpha) \leq SWU(\alpha) \leq SWU(x) < lbM(x) \leq lbM(\alpha) \leq mu(\alpha)$, i.e. $\alpha$ cannot be a HUP sequence. For example, since $P(b) = 0.125$, $P(g) = 0.5625 < mp = 0.875$, we can remove $b$ and $g$ from $\mathcal{D}'$. For the sequence $\alpha = e$, before removing $b$ and $g$, its ($lbM$, $lbF$, $RBU$, $LRU$, $SWU$) values are respectively (5, 31, 346, 423, 423), and after removing, the corresponding updated values are (50, 50, 296, 361, 361), i.e. the updated $lbM$, $lbF$ LB values increase and the $RBU$, $LRU$, $SWU$ UB values decrease. In other words, these updated values really are more tightened.

b. The tightening strategy - $CMAPS(SWU, lbM, P)$ (or briefly $CMAPS$) for speeding up the mining process. Note that the $\mathcal{AM}(P)$, $\mathcal{AM}(SWU)$ and $\mathcal{M}(lbM)$ properties are true. Inspired by the $CMAP$ technique using the co-occurrence information of two items based on the support measure [7], for each item $x$ of $\mathcal{A}$, let us define $iCMAP(x) \stackrel{\text{def}}{=} \{y \in \mathcal{A} | y \succ x \wedge P(xy) \geq mp \wedge SWU(xy) \geq lbM(xy)\}$ and $sCMAP(x) \stackrel{\text{def}}{=} \{y \in \mathcal{A} | P(x \to y) \geq mp \wedge SWU(x \to y) \geq lbM(x \to y)\}$. Then, the two $iCMAP(x)$ and $sCMAP(x)$ sets contain candidate items for extensions of any sequence $\alpha$ such that $lastItem(\alpha) = x$, i.e. $I(\alpha) \subseteq iCMAP(x)$ and $S(\alpha) \subseteq sCMAP(x)$. For example, to prove $I(\alpha) \subseteq iCMAP(x)$ for any $\alpha$ such that $lastItem(\alpha) = x$, consider any item $y \in I(\alpha)$, i.e. $y \succ x$, $lbBi(\alpha \diamond_i y) \leq LRU(\alpha \diamond_i y)$ and $P(\alpha \diamond_i y) \geq mp$. Then, define $\beta = \alpha \diamond_i y$, $\delta = x \diamond_i y$, since $\beta \sqsupseteq \delta$, $lbM(\delta) \leq lbM(\beta) \leq lbBi(\beta) \leq LRU(\beta) \leq SWU(\beta) \leq SWU(\delta)$ and $P(\delta) \geq P(\beta) \geq mp$, i.e. $y \in iCMAP(x)$. The inclusion $S(\alpha) \subseteq sCMAP(x)$ is proven similarly. Note that for all items $x$ of $\mathcal{A}$, the two $iCMAP(x)$ and $sCMAP(x)$ sets are only calculated once. Thus, to improve the process of mining $\mathcal{HUPS}$ by the tightening $\mathcal{TS}(LRU, lbBi, P)$ strategy, we can additionally use the following $CMAPS$ strategy: if $y \notin iCMAP(x)$ or $y \notin sCMAP(x)$, then $y \notin I(\alpha)$ or $y \notin S(\alpha)$, without wasting much time for computing the $lbBi$ and $LRU$ bounds of $\alpha \diamond y$, which are used in $I(\alpha)$ or $S(\alpha)$. In other words, we obtain two tighter sets of candidate items for extensions: $I(\alpha) = \{y \in iCMAP(x) | y \succ x \wedge LRU(\alpha \diamond_i y) \geq lbBi(\alpha \diamond_i y) \wedge P(\alpha \diamond_i y) \geq mp\}$ and $S(\alpha) = \{y \in sCMAP(x) | LRU(\alpha \diamond_s y) \geq lbBi(\alpha \diamond_s y) \wedge P(\alpha \diamond_s y) \geq mp\}$.

In short, the tuple ($SWU$, $lbM$, $RBU$, $lbF$, $LRU$, $lbBi$, $P$) used in the five $\mathcal{DPS}$, $\mathcal{WPS}$, $\mathcal{TS}$, $\mathcal{RedS}$ and $CMAPS$ strategies is called a solution of $\mathcal{HUPSM}$.

c. Note that applying the tightening $\mathcal{TS}(LRU, lbBi, P)$ strategy, based on the two smaller $I(\alpha)$ and $S(\alpha)$ sets in this paper, is better compared to $IS(\alpha) \stackrel{\text{def}}{=} I(\alpha) \cup S(\alpha)$ as shown in [17]. Moreover, replacing $lbM$, $lbBi$ LBs with $lbF$, or $SWU$, $LRU$ UBs with $RBU$ may result in incorrect results. Since the $\mathcal{MB}i(lbF)$, $\mathcal{M}(lbF)$ and $\mathcal{AMB}i(RBU)$ properties do not hold, $\mathcal{RedS}(SWU, lbF, P)$, $\mathcal{WPS}(SWU, lbF, P)$, $\mathcal{TS}(RBU, lbBi, P)$ as well as $\mathcal{TS}(SWU, lbF, P)$ are incorrect. Indeed, we consider the following counter example with $\mathcal{D}' = \{\psi' = (d, 50) \to (a, 5)(c, 40) \to (a, 1)(b, 1)(f, 30) \to (e, 1)\}$, where the $mu$ thresholds of all items in $\mathcal{A} = (a, b, c, d, e, f)$ are respectively (130, 30, 125, 132, 140, 133) and $P(\psi') = mp = 1$. Assume conversely that $\mathcal{RedS}(SWU, lbF, P)$ is true.

Since $\mathcal{SWU}(f) = 128 < lbF(f) = 133$, we can remove $f$ from $\mathcal{D}'$, so $\gamma = d \to ac \to f$ containing $f$ cannot be present in $\mathcal{HUPS}$, while $u_{\max}(\gamma) = u_{\min}(\gamma) = mu(\gamma) = 125$ and $P(\gamma) = 1 = mp$, i.e. $\gamma$ is a $\mathcal{HUP}$ sequence w.r.t. $u_{\min}$ as well as $u_{\max}$. In other words, replacing $lbM$ in $\mathcal{RedS}(SWU, lbM, P)$ with $lbF$ may lead to the incompleteness of Algorithm 1 in [12] for mining $\mathcal{HUPS}$ using $u_{\max}$. The reason is that $\mathcal{MBi}(lbF)$ does not hold. Indeed, for the backward extension $\gamma$ of $\delta = d \to a \to f$ (with c), we have $\rho(\gamma) = \rho(\delta) = \mathcal{D}'$ and $lbF(\gamma) = 125 < lbF(\delta) = 130$. Using the correct width pruning condition, we have $WidthPC_{SWU,lbM,P}(f) = (SWU(f) = 128 < lbM(f) = 30)$, which does not hold, and thus, we are not allowed to remove $f$ from $\mathcal{D}'$.

d. Consider the two following particular cases of $\mathcal{HUPSM}$. *First*, if all values $P(\psi')$ are identical with a constant (e.g. $P(\psi') \equiv 1$), $\forall \psi' \in \mathcal{D}'$ and $mp = 1/|\mathcal{D}'|$, then $P(\alpha) = \text{supp}(\alpha)/|\mathcal{D}'|$ is the relative support measure of $\alpha$ and $P(\alpha) \geq mp$ for all sequences $\alpha$. Thus, $\mathcal{HUPSM}$ becomes the problem $\mathcal{HUSM}$ of high utility sequence mining in (certain) QSDB with multiple minimum utility thresholds (using $u_{\min}$). *Second*, if all $mu(x)$ are identical to a constant $mu$, $\forall x \in \mathcal{A}$, i.e. all items have the same importance, then we obtain the problem of high utility-probability sequence mining in UQSDB with a single minimum utility $mu$ threshold. Thus, by replacing $u_{\min}$ with $u_{\max}$, we obtain the two corresponding problems proposed in [12] and [23] (using two $MEU$ and $LAS$ UBs on $u_{\max}$ in [12] instead of $RBU$ and $LRU$).

Based on the above theoretical results, a novel algorithm named HUPSMT (High Utility-Probability Sequence mining with Mulitiple minimum utility Thresholds) is designed for the $\mathcal{HUPSM}$ problem using the minimum $u_{\min}$ utility.

### 3.2. The HUPSMT algorithm

The proposed HUPSMT algorithm is based on a novel vertical data structure named Extended Utility List (EUL). Given a sequence $\alpha$ and an input $q$-sequence $\psi'_i \in \rho(\alpha)$, for each ending $end \stackrel{\text{def}}{=} end(\alpha, \psi'_i)$ of $\alpha$ in $\psi'_i$, let $u^{end}_{\min} = u(\alpha, \psi'_i, end)$, $u_{rem} = u_{rem}(end) \stackrel{\text{def}}{=} u(rem(\alpha, \psi'_i, end))$ and $mu_{rem} = mu_{rem}(end) \stackrel{\text{def}}{=} \min\{mu(x)|x \in rem(\alpha, \psi'_i, end)\}$ be respectively the minimum utility, remaining utility and remaining minimum utility $mu$ threshold of $\alpha$ in $\psi'_i$ according to the ending $end$. Furthermore, we denote the list of tuples $tup(end) \stackrel{\text{def}}{=} (end, u^{end}_{\min}, u_{rem}, mu_{rem})$ as $tl(\alpha, \psi'_i)$ or briefly $tl \stackrel{\text{def}}{=} \{tup(end)|end = end(\alpha, \psi'_i)\}$. Without loss of generality, we can assume that the $tl$ list is sorted in ascending order by $end$. Then, the structure $EUL$ of $\alpha$ is defined as $EUL(\alpha) \stackrel{\text{def}}{=} \{(i, tl(\alpha, \psi'_i))|\psi'_i \in \rho(\alpha)\}$. This structure allows us to quickly calculate the probability $P$, $u_{\min}$, $RBU$, $LRU$, $SWU$ UBs and $lbF$, $lbBi$, $lbM$ LBs of $\alpha$ as well as its extensions. Due to space limitations, formulas for calculating them quickly are skipped.

The pseudo-code for the HUPSMT algorithm is shown in Figure 1. It takes as input a UQSDB $\mathcal{D}'$, a probability threshold $mp$ and a set of minimum utility thresholds $Mu \stackrel{\text{def}}{=} \{mu(x), x \in \mathcal{A}\}$. At the first level of the prefix-tree, the algorithm applies reducing and width pruning strategies, $\mathcal{Red}(SWU, lbM, P)$ and $\mathcal{WPS}(LRU, lbBi, P)$. It scans UQSDB $\mathcal{D}'$ once to calculate the set $S \stackrel{\text{def}}{=} \{x \in \mathcal{A}|P(x) \geq mp \wedge SWU(x) \wedge lbM(x)\}$ of relevant HUP candidate items. Then, irrelevant items in $\mathcal{A} \setminus S$ are removed from $\mathcal{D}'$ (lines 1-2) and all

bounds (UBs and LBs) of all remaining items in $S$ are updated (line 3) and can be tightened. Next, the procedure SearchHUPS is called for each item $x \in S$ (line 4).

**HUPSMT** $(\mathcal{D}', mp, Mu)$
. *Input*: UQSDB $\mathcal{D}'$, probability $mp$ threshold, set $Mu$ of utility thresholds $\{mu(x), x \in \mathcal{A}\}$.
. *Output*: all HUP sequences in $\mathcal{HUPS}$.
1. Scan once $\mathcal{D}'$ to find $S := \{x \in \mathcal{A} \mid P(x) \geqslant mp \wedge SWU(x) \geqslant lbM(x)\}$;
2. **Remove** from $\mathcal{D}'$ all *irrelevant* items in $\mathcal{A} \backslash S$; // *using* $\mathcal{Red}(SWU, lbM, P)$ *strategy*
3. **Update** bounds of all *remaining* items in $S$; // *using* $\mathcal{WPS}(SWU, lbM, P)$ *strategy*
4. **for each** item $x \in S$ **do SearchHUPS** $(x, S, S, mp)$;

*Figure 1.* Algorithm HUPSMT for mining the $\mathcal{HUPS}$ set

**SearchHUPS** $(\alpha, I, S, mp)$
. *Input*: sequence $\alpha$, two candidate item $I$ and $S$ sets for extensions of the prefix of $\alpha$, $mp$ threshold.
. *Output*: $\alpha$ if it is a HUP sequence.
1. **if** $(RBU(\alpha) < lbF(\alpha))$ **then return;** // *using* $\mathcal{DPS}(RBU, lbF)$ *strategy*
2. **if** $(u_{min}(\alpha) \geqslant mu(\alpha))$ **then** output HUP sequence $\alpha$;
3. $newI := newS := \emptyset$; // *candidate item sets for extensions of* $\alpha$
// *using* $\mathcal{TS}(LRU, lbBi, P)$ *in* $\mathcal{WPS}(LRU, lbBi, P)$ *and CMAPS strategies:*
4. $x := lastItem(\alpha)$;
5. **for each** item $y \in I$ such that $y \succ x$ **do**
6. **if** $(y \in iCMAP(x) \wedge P(\alpha \diamond_i y) \geqslant mp \wedge LRU(\alpha \diamond_i y) \geqslant lbBi(\alpha \diamond_i y))$ **then** $newI = newI \cup \{y\}$;
7. **for each** item $y \in S$ **do**
8. **if** $(y \in sCMAP(x) \wedge P(\alpha \diamond_s y) \geqslant mp \wedge LRU(\alpha \diamond_s y) \geqslant lbBi(\alpha \diamond_s y))$ **then** $newS = newS \cup \{y\}$;
9. **for each** item $y \in newI$ **do SearchHUPS** $(\alpha \diamond_i y, newI, newS, mp)$;
10. **for each** item $y \in newS$ **do SearchHUPS** $(\alpha \diamond_s y, newI, newS, mp)$;

*Figure 2.* Procedure SearchHUPS

The recursive SearchHUPS procedure (Figure 2) takes as input a sequence $\alpha$, two $I$ and $S$ sets of candidate items for $i-$ and $s-extensions$ of the $\alpha$s prefix, and the $mp$ threshold. The procedure uses the depth pruning strategy $\mathcal{DPS}(RBU, lbF)$ in line 1. If $u_{\min}(\alpha) \geq mu(\alpha)$, the HUP sequence $\alpha$ is output (line 2). Next, in lines 3-8, the width pruning and tightening up strategies, $\mathcal{WPS}(LRU, lbBi, P)$, $\mathcal{TS}(LRU, lbBi, P)$ and $CMAPS$, are applied for extensions of $\alpha$. Finally, the SearchHUPS procedure is recursively called for each item in the two $newI$ and $newS$ sets (lines 9-10). Theorems 1-3 guarantee the correctness of HUPSMT, which allows to prune non-HUP candidate branches early without missing any HUP sequence.

## 4. EXPERIMENTAL EVALUATION

Experiments were performed on an Intel Core i5-2320 CPU, 3.0 GHz PC with 8 GB of memory, running Windows 8.1. All algorithms used in the experiments are implemented in Java SE 1.8 and compared on four real-life SDBs named BMS, SIGN, FIFA and BIBLE, and one synthetic SDB named D4C7T5N5S6I4 generated using the IBM Quest data generator

(obtained from [8]) with parameters described in Table 4. The characteristics of the databases are shown in Table 5. For the four real-life SDBs, BMS is dense, while the three remaining SDBs are sparse. SIGN is a smaller SDB, and FIFA and BIBLE are larger than BMS. To obtain integrated UQSDBs from SDBs, we have used the IBM Quest data generator. Then, the minimum probability thresholds and the quantities of all items of input $q$-sequences in the SDBs were randomly generated in the [0; 1] interval and [1; 5] interval, respectively. The external utilities of all distinct items in the SDBs are created using a log-normal distribution in the range of 1 and 1000. Similar to [12], to avoid creating an enormous number of HUP sequences, the minimum utility thresholds of all different items in the databases are set as $mu(item) = \max\{u_{\min}(item) * \beta, LMU * u(D')\}$ such that they are not too low, where $\beta$ is a constant (often greater than one) and $LMU$ (%) is a least minimum utility threshold, specified by users.

*Table 4.* Parameters of the IBM quest synthetic data generator

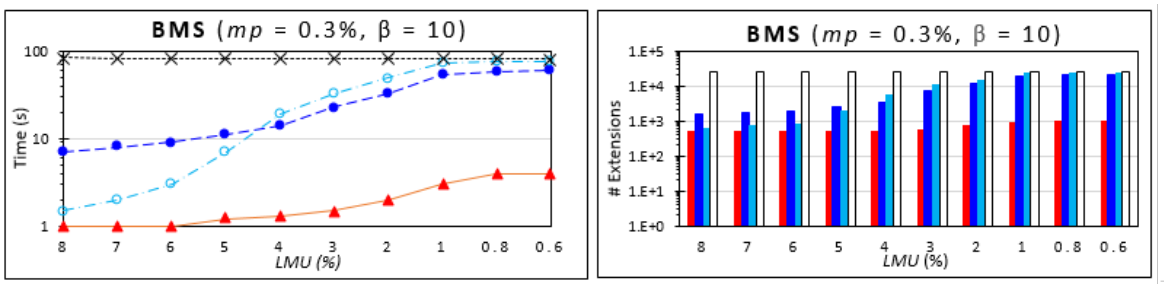
| Parameter | Meaning |
|---|---|
| D | Number of sequences (in thousands) in the database |
| C | Average number of item-sets per sequence |
| T | Average number of items per item-set |
| N | Number of different items (in thousands) in the database |
| S | Average number of item-sets in maximal sequences |
| I | Average number of items in maximal sequences |

*Table 5.* Characteristics of databases

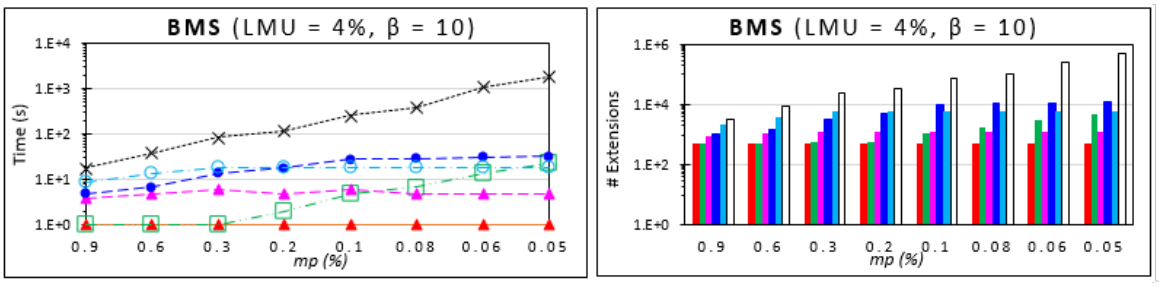
| Database | #sequences | #items | avg. seg. length | type of data |
|---|---|---|---|---|
| BMS | 59,601 | 497 | 2.51 | web click stream |
| SIGN | 730 | 267 | 51.99 | language utterances |
| FIFA | 20,450 | 2,990 | 36.24 | web click stream |
| BIBLE | 36,369 | 13,905 | 21.64 | book |
| D4C7T5N5S6I4 | 4000 | 5000 | 28.68 | synthetic |

First, we consider the influence of the three depth, width and CMAP pruning strategies on HUPSMT, which is the first algorithm for solving the $\mathcal{HUPSM}$ problem in UQSDBs using the minimum $u_{\min}$ utility. For comparing their pruning effect, we compared the performance of HUPSMT for the six following cases using $\mathcal{WPS}(P)$ and additionally: (1) using the three $\mathcal{DPS}$, $\mathcal{WPS}$ and CMAPS strategies (All), (2) only using CMAPS (CMAP), (3) using both $\mathcal{DPS}$ and $\mathcal{WPS}$ (Both), (4) only using $\mathcal{DPS}$ (Depth), (5) only using $\mathcal{WPS}$ (Width), and (6) without using any strategy related to utility (Non). The following experimental results show that the runtime of the algorithm for the above cases always depends on the number of performed extensions. We utilize the real-life BMS database as an illustration (Figure 3) with the coefficient $\beta = 10$.

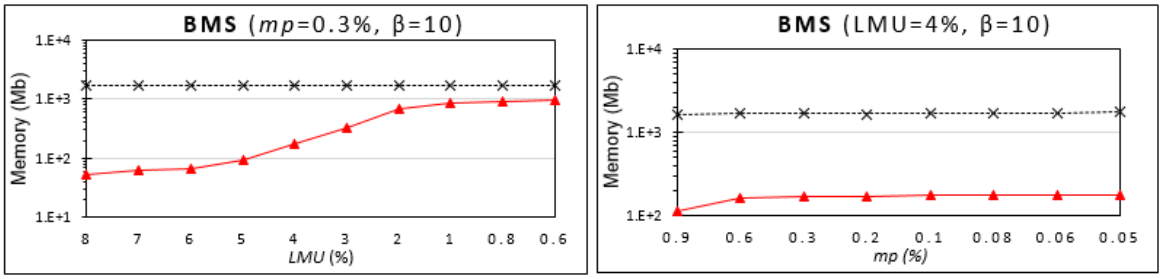
We fixed the $mp$ threshold to 0.3% and decreased $LMU$. The runtime and number of extensions are shown in Figure 3a. For high $LMU$ values (greater than 4%), the pruning effect ($PE$) of $\mathcal{WPS}$ is better than $\mathcal{DPS}$ because the width pruning condition using the ($LRU$, $lbBi$) bounds has more chance to be applied compared to ($RBU$, $lbF$), so the number




(a) Varying $LMU(\%)$ for a fixed $mp = 0.3\%$




(b) Varying $mp(\%)$ for a fixed $LMU = 4\%$




(c) Memory usage in (**All**) and (**Non**) for fixed $mp = 0.3\%$ and $LMU = 4\%$

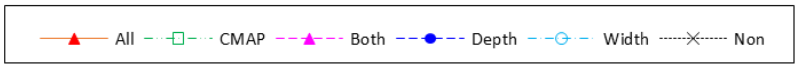



*Figure 3.* Comparison of pruning strategies on BMS

of pruned candidate sequences (or extensions) is greater (or smaller, respectively). Otherwise, for lower $LMU$, $PE$ of $\mathcal{DPS}$ using the tighter ($RBU$, $lbF$) bounds is better than $\mathcal{WPS}$. (All) is faster by 12, 14 and 55 times on average compared to (Depth), (Width) and (Non), respectively.

To further analyze the $PE$ of different strategies, we consider a fixed $LMU = 4\%$ while decreasing $mp$. The resulting runtime and number of extensions are shown in Figure 3b. For high $mp$ (larger than 0.2%), since $PE$ of $\mathcal{WPS}$ using the $P$ probability is stronger than $LRU$, and $RBU$ is tighter than $LRU$, $PE$ of $\mathcal{DPS}$ is better than $\mathcal{WPS}$. Otherwise, $\mathcal{WPS}$

is better than $\mathcal{DPS}$. However, using both $\mathcal{WPS}$ and $\mathcal{DPS}$ (Both) is always better than using only one of them. The $PE$ of (Both) is better than (CMAP) for low $mp$ (less than 0.1%) because $PE$ of $\mathcal{WPS}$ using the $P$ probability becomes weaker, and (CMAP) uses the pair of ($SWU$, $lbM$) bounds, while (Both) uses two pairs of tighter ($LRU$, $lbBi$) and ($RBU$, $lbF$) bounds. Thus, applying simultaneously all of the above strategies is really necessary. Finally, (All) is always the best. In terms of average, it is about 7, 5, 21, 17 and 472 times faster than (CMAP), (Both), (Depth), (Width) and (Non), respectively.

In Figure 3c, the memory consumption of (All) is about 10 or 12 times on average less than (Non) when $LMU$ or $mp$ are fixed, respectively.

Furthermore, Figure 4 shows that on average, using multiple $mu$ thresholds for different items in the $\mathcal{HUPSM}$ problem significantly decreases the cardinality of $\mathcal{HUSP}$ and mining time (by more 700 and 200 times, respectively) compared to using the P-HUSPM algorithm [23] for mining all high utility-probability sequences with a single common threshold for all items, $mu = \min\{mu(x)|x \in \mathcal{A}\}$. In this experiment with BMS and $\beta = 10$, we have $mu = 0.01\%$ (of $u(\mathcal{D})$).

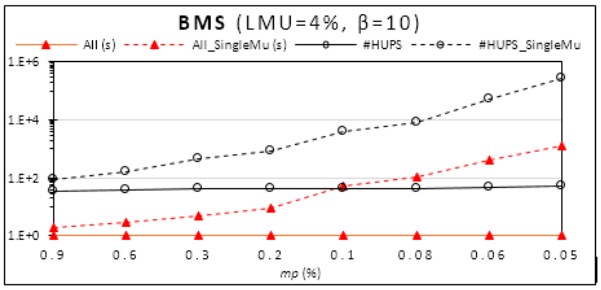
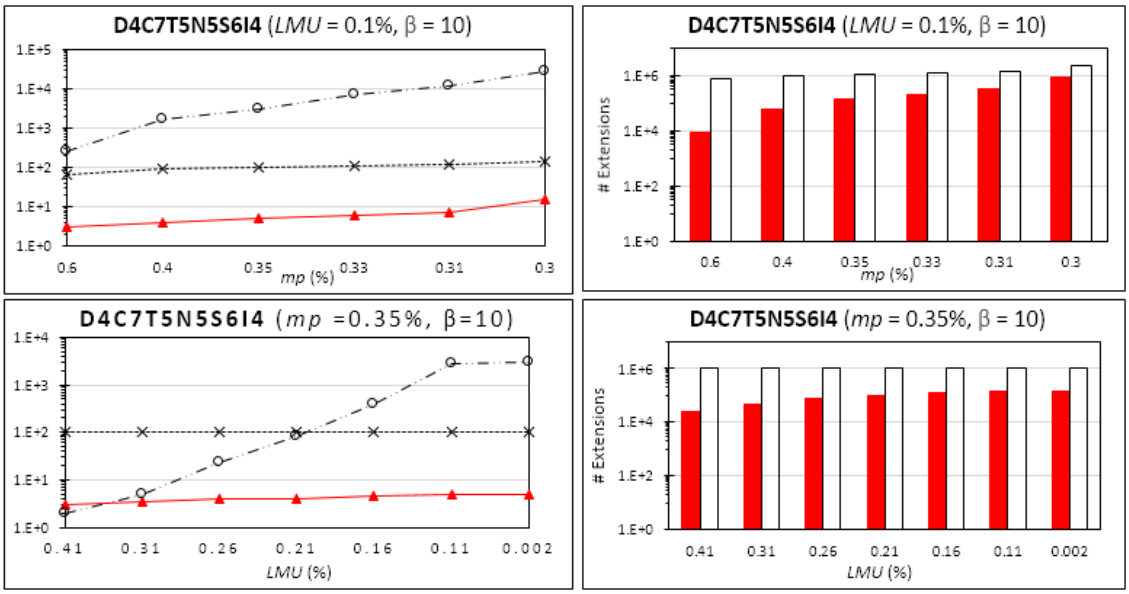



*Figure 4.* Runtime (sec), cardinality of $\mathcal{HUPS}$ using multiple or single $mu$ threshold(s)

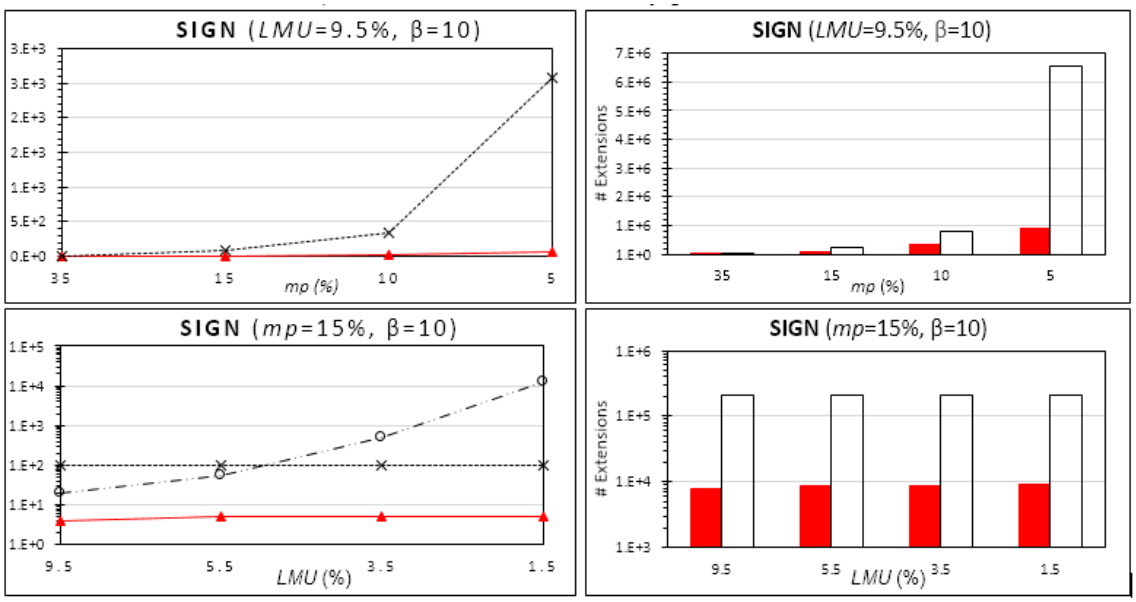
We also have similar remarks for the experimental results on the synthetic dataset D4C7T5N5S6I4, the smaller real-life SIGN and two larger FIFA, BIBLE datasets. Figures 5a and 5b show the runtime, number of extensions and $\mathcal{HUPS}$ in D4C7T5N5S6I4 and SIGN. Figure 6 shows the runtime and cardinality of $\mathcal{HUPS}$ when varying the $mp$ parameter for the FIFA and BIBLE datasets, generated by HUPSMT for (All) and (Non). When fixing $LMU$ and varying $mp$ on the four above datasets, the execution time of (All) is faster than (Non) on average by about 21, 17, 24 and 34 times.

## 5. CONCLUSIONS

This paper proposes *depth* and *width* pruning strategies, *reducing* and *tightening* strategies, which rely on the anti-monotonic property of the probability $P$, anti-monotone-like properties of $RBU$, $LRU$ and $SWU$ upper-bounds on the minimum $u_{\min}$ utility, and monotone-like properties of three novel $lbM$, $lbBi$ and $lbF$ lower-bounds on a minimum utility threshold $mu$ measure of items. These strategies allow us to prune non high utility-probability branches of the prefix search tree early and to reduce databases as well as tighten the set of candidate items to be considered for extensions. The strategies are integrated into the novel $EUL$ data structure and HUPSMT algorithm. It is the first algorithm for discovering




(a) Influence of the $LMU$ and $mp$ parameters on D4C7T5N5S6I4




(b) Influence of the $LMU$ and $mp$ parameters on SIGN


*Figure 5.* Influence of the $LMU$ and $mp$ parameters on D4C7T5N5S6I4 and SIGN

all high utility-probability sequences in UQSDBs with multiple minimum utility thresholds using $u_{\min}$. An experimental study shows the efficiency of the proposed algorithm in both real-life and synthetic UQSDBs in terms of time and memory consumption.

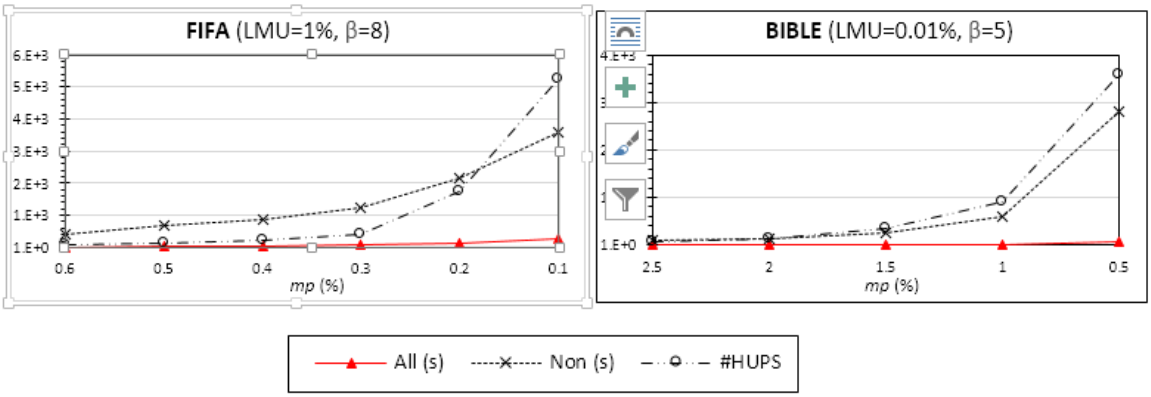



*Figure 6.* Influence of the *mp* parameter on FIFA and BIBLE

In the future, we will consider the similar problem of using the average utility measure of sequences instead of $u_{\min}$ and the problem of mining the top-$k$ high utility-probability sequences in UQSDBs.

## ACKNOWLEDGMENT

This work is funded by Vietnams National Foundation for Science and Technology Development (NAFOSTED) under Grant Number 102.05-2017.300.

## REFERENCES

[1] C.F. Ahmed, S.K. Tanbeer, and B.S Jeong, "Mining high utility web access sequences in dynamic web log data," in *2010 11th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, London, UK, June 9-11, 2010, pp. 76-81. Doi: 10.1109/SNPD.2010.21.

[2] C. F. Ahmed, S. K. Tanbeer, and B. S. Jeong, "A novel approach for mining high-utility sequential patterns in sequence databases," *ETRI Journal*, vol. 32, no. 5, pp. 676-686, 2010.

[3] A. U. Ahmed, C. F. Ahmed, M. Samiullah, N.Adnan, and C. K. S. Leung, "Mining interesting patterns from uncertain databases," *Information Sciences Journal*, vol. 354, pp. 60-85, 2016.

[4] O.K. Alkan, and P. Karagoz, "CRoM and HuspExt: Improving efficiency of high utility sequential pattern extraction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 10, pp. 2645–2657, 2015. Doi: 10.1109/TKDE.2015.2420557.

[5] C.K. Chui, B. Kao, and E. Hung, "Mining frequent itemsets from uncertain data," in *Proceedings 11th Pacific-Asia Conference, PAKDD 2007,* Nanjing, China, May 22-25, 2007, pp. 47–58.

[6] B. Dalmas, P. Fournier-Viger, and S. Norre, "TWINCLE: A constrained sequential rule mining algorithm for event logs," *Procedia Computer Science*, vol. 112, pp. 205–214, 2017.

[7] P. Fournier-Viger, A. Gomariz, M. Campos, and R. Thomas, "Fast vertical mining of sequential patterns using co-occurrence information," in *Proc. 18th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD '2014, Part I*, Tainan, Taiwan, May 13-16, 2014, pp. 40-52.

[8] P. Fournier-Viger, A. Gomariz, T. Gueniche, A. Soltani, C. Wu, and V.S. Tseng, "SPMF: a Java open-source pattern mining library," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3389-3393, 2014.

[9] T.P. Hong, C.H. Lee, and S.L. Wang, "Mining high average-utility itemsets," in *2009 IEEE International Conference on Systems, Man and Cybernetics*, San Antonio, TX, USA, Oct. 11-14, 2009, pp. 2526–2530. Doi: 10.1109/ICSMC.2009.5346333.

[10] G.C. Lan, T.P. Hong, V.S. Tseng, and S.L. Wang, "Applying the maximum utility measure in high utility sequential pattern mining," *Expert Systems with Applications*, vol. 41, no. 11, pp. 5071–5081, 2014.

[11] M. Li, Y. Liu, "Underground coal mine monitoring with wireless sensor networks", *ACM Transactions on Sensor Networks*, vol. 5, no. 2, 2009.

[12] J.C.W. Lin, J. Zhang, and P. Fournier-Viger, "High-utility sequential pattern mining with multiple minimum utility thresholds," in *First International Joint Conference, APWeb-WAIM 2017, Proceedings, Part I*, Beijing, China, July 79, 2017, pp. 215-229. https://doi.org/10.1007/978-3-319-63579-8_17.

[13] J. Liu, K. Wang, and B. Fung, "Direct discovery of high utility itemsets without candidate generation," in *2012 IEEE 12th International Conference on Data Mining*, Location: Brussels, Belgium, Dec. 10-13, 2012, pp. 984989. Doi: 10.1109/ICDM.2012.20.

[14] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M. Hsu, "Mining sequential patterns by pattern-growth: the PrefixSpan approach," *Journal IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 11, pp. 1424-1440, 2004.

[15] B.E. Shie, P.S. Yu, and V.S. Tseng, "Mining interesting user behavior patterns in mobile commerce environments," *Appl. Intell.*, vol. 38, no. 3, pp. 418-435, 2013.

[16] T. Truong, H. Duong, B. Le, and P. Fournier-Viger, "Efficient Vertical Mining of High Average-Utility Itemsets based on Novel Upper-Bounds," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 2, pp. 301–314, 2019. Doi: 10.1109/TKDE.2018.2833478.

[17] T. Truong, A. Tran, H. Duong, B. Le, and P. Fournier-Viger, "EHUSM: Mining high utility sequences with a pessimistic approach," in *Proc. UDM 2018, 24th ACM SIGKDD (KDD 2018)* [Online]. Available: http://philippe-fournier-viger.com/utility_mining_workshop_2018/paper5_pessimistic.pdf.

[18] J. Z. Wang, J.L. Huang, and Y.C. Chen, "On efficiently mining high utility sequential patterns," *Knowledge and Information Systems*, vol. 49, no. 2, pp. 597–627, 2016.

[19] J. Yin, Z. Zheng, L. Cao, Y. Song, and W. Wei, "Efficiently mining top-k high utility sequential patterns," in *2013 IEEE 13th International Conference on Data Mining*, Dallas, TX, USA, Dec. 7–10, 2013, pp. 1259–1264. Doi: 10.1109/ICDM.2013.148.

[20] J. Yin, Z. Zheng, and L. Cao, "USpan: an efficient algorithm for mining high utility sequential patterns," in *Proceeding KDD '12 Proceedings of The 18th ACM SIGKDD International Conference on Knowledge Discovery And Data Mining* , Beijing, China, August 12–16, 2012, pp. 660–668. Doi:10.1145/2339530.2339636.

[21] S. Zida, P. Fournier-Viger, JC.W. Lin, C.W. Wu, and V.S. Tseng, "EFIM: A highly efficient algorithm for high-utility itemset mining," in *Advances in Artificial Intelligence and Soft Computing 14th Mexican International Conference on Artificial Intelligence, MICAI 2015, Proceedings, Part I*, Cuernavaca, Morelos, Mexico, October 25-31, 2015, pp. 530-546.

[22] M. Zihayat, H. Davoudi, and A. An, "Top-k utility-based gene regulation sequential pattern discovery," in *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Shenzhen, China, Dec. 15-18, 2016, pp. 266–273. Doi: 10.1109/BIBM.2016.7822529.

[23] B. Zhang, J.C.W. Lin, P. Fournier-Viger, and T. Li, "Mining of high utility-probability sequential patterns from uncertain databases," *PLoS ONE*, vol. 12, no. 7, pp. 1-21, 2017.

*Received on November 14, 2018*
*Revised on February 14, 2019*