A Novel Dynamic Skyline Operation for Multicriteria Decision Support

Junyi Chai¹, James N. K. Liu¹, Man Lung Yiu¹, Hongwei Wang², and Anming Li¹

¹Department of Computing
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong SAR
{csjchai, csnkliu, csmlyiui, csamli}@comp.polyu.edu.hk
²School of Economics and Management
Tongji University, Shanghai, China
hwwang@mail.tongji.edu.cn

Abstract

The skyline operation is a typical Multicriteria Decision Making (MCDM) that has been well studied in database context. The basic assumption of conventional skyline operator is preference-ordered values within multi-dimensional decision tables. This assumption utilizes preference-ordered values to simplify the complex human preference, which is subject to big challenges in real applications. In this paper, we firstly investigate preference relations on skyline operation and lead to a dynamic preference model. We then introduce the concept of preference intensity and propose a new decision model, called Tolerant Skyline Operation (T-skyline). We study the method for computation of T-skyline and address the issue of continuous T-skyline maintenance. Through a detailed case study related to the NBA player evaluation in 2010-11 regular seasons, we demonstrate the effectiveness and advantages of the proposed decision model.

1. Introduction

Multicriteria Decision Making (MCDM) is the process in which decision makers evaluate each object according to multiple criteria. It consists of four issues including criteria analysis; sorting; ranking; and choice [6]. We regard the first issue as the essential procedure of optimization. And the latter three can produce specific decision outcomes. Conventional skyline operation [1] can be regarded as a multicriteria ranking procedure aiming to retrieve a set of qualified objects. It requires a settled preference system as the prerequisite. A dominant B if its values are not inferior to B’s values in any dimensions and at least superior to B’s values in one dimension. Skyline is regarded as an elementary set in which objects cannot be dominated by any other object in the universe. The skyline operation aims to obtain such object subset that fulfills the requirement of predefined preference.

In the past decade, many papers have contributed to skyline-related analysis. Based on the naive pairwise comparison, major methods include Block-Nest-Loop (BNL) and Divide-and-Conquer (D&C) [1], Sort-filter Skyline (SFS) [5], Linear-estimation-sorts (LESS) [7], and so on. For meeting various needs of real applications, many interesting studies also appeared such as subspace skyline operators [14], R-tree based skyline queries [13], and constrained skyline queries [12]. A detailed survey of related works is provided in section 4. Although existing works have provided efficient algorithms to conventional skyline computation, they rarely take human dynamic preference into account. First, most of existing methods assume a settled preference system before operating a skyline. This assumption is subject to challenge since users may adjust their preference dynamically. Second, a well-known weakness of conventional skyline is that the outputting size of skyline is uncontrollable. Once a preference system has been confirmed, the size is accordingly settled. This problem comes from two aspects. For one thing, the predefined preference system strongly relies on decision-makers’ subjective judgment which might be imperfect. For the other thing, nonflexible skyline results inevitably omit the users’ desirable size.

In order to overcome these weaknesses, this paper provides a new decision-oriented skyline decision model. We firstly carry out an analysis on preference relations of conventional skyline operations through a running case. Then we introduce a new concept of preference intensity for dynamically modeling human
2. The Tolerant Skyline Operator

2.1. Presentation of the Problem

First, we propose the challenging problems via a running case of NBA player evaluation. Table 1 gives technical statistic of NBA players in 2008-09 regular seasons (data collected from \url{http://espn.go.com/nba/}). Ten players are described by SEVEN attributes, including last name (LN), team (TM), playing games (G), minutes per game (MPG), assists (A), turnovers (T), and steal (S).

<table>
<thead>
<tr>
<th>No.</th>
<th>LN</th>
<th>TM</th>
<th>G</th>
<th>MPG</th>
<th>A</th>
<th>T</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>Nash</td>
<td>Sun</td>
<td>68</td>
<td>35.3</td>
<td>786</td>
<td>264</td>
<td>56</td>
</tr>
<tr>
<td>P2</td>
<td>Williams</td>
<td>Jazz</td>
<td>71</td>
<td>37.5</td>
<td>672</td>
<td>218</td>
<td>78</td>
</tr>
<tr>
<td>P3</td>
<td>Kidd</td>
<td>Nets</td>
<td>71</td>
<td>36.9</td>
<td>645</td>
<td>188</td>
<td>117</td>
</tr>
<tr>
<td>P4</td>
<td>Paul</td>
<td>Hornets</td>
<td>57</td>
<td>36.5</td>
<td>495</td>
<td>143</td>
<td>109</td>
</tr>
<tr>
<td>P5</td>
<td>Davis</td>
<td>Warriors</td>
<td>55</td>
<td>35.5</td>
<td>451</td>
<td>170</td>
<td>114</td>
</tr>
<tr>
<td>P6</td>
<td>Ford</td>
<td>Raptors</td>
<td>76</td>
<td>30.4</td>
<td>535</td>
<td>215</td>
<td>93</td>
</tr>
<tr>
<td>P7</td>
<td>Miller</td>
<td>76ers</td>
<td>71</td>
<td>36.9</td>
<td>562</td>
<td>200</td>
<td>99</td>
</tr>
<tr>
<td>P8</td>
<td>Wade</td>
<td>Heat</td>
<td>46</td>
<td>38.9</td>
<td>362</td>
<td>193</td>
<td>96</td>
</tr>
<tr>
<td>P9</td>
<td>Iverson</td>
<td>Nuggets</td>
<td>57</td>
<td>42.7</td>
<td>417</td>
<td>238</td>
<td>112</td>
</tr>
<tr>
<td>P10</td>
<td>Billups</td>
<td>Pistons</td>
<td>64</td>
<td>36.7</td>
<td>460</td>
<td>132</td>
<td>78</td>
</tr>
</tbody>
</table>

Conventional skyline operation will firstly predefine a preference order on the attributes. Generally speaking, there are two types of attributes: GAIN (large values are preferred) and COST (small values are preferred). In this case, attributes “LN” and “TM” generally cannot be preference-ordered. The value in attributes “G” and “MPG” can be preference-ordered in some situations. Normally, “A” and “S” are GAIN type and “T” belongs to COST. If just taking “A”, “T” and “S” into account, we can easily obtain the skyline that includes P1, P2, P3, P4, P5, P10. However, many existing methods will confront challenges in case like a NBA team manager who would like to choose qualified players under the following situations.

1. The original attributes cannot be directly used to choose the qualified player. Alternatively, the manager would like to consider another three dimensions including (i) Assist per 48 min: \( (G \times MPG) / (A + T) \); (ii) Ratio of assist over turnover: \( A / T \); (iii) Ratio of steal over turnover: \( S / T \). Then, how to compute skyline under such user-specific preference system?

2. This user-specified evaluation (preference) system is not one hundred percent suitable or trustworthy. In this case, users may want to correct such preference system through adjusting some coefficients.

3. In a very large numerical database, the difference between two records (attribute values) might be so small that can be ignored. For example, records \( <P3, S> = 117 \) and \( <P5, S> = 115 \) can be regarded as the same in this decision process. Therefore, the conventional skyline operation needs appropriate extensions.

In the next section, we firstly construct a dynamic preference model of skyline operation. Then a new concept of preference intensity is introduced to model users’ dynamic preference. We finally provide a new skyline operator for our concerned decision problem.

2.2. Dynamic Preference System

Skyline operation arouses great attentions because of its ability in analyzing data table. Such table consists of finite objects (also called tuples, alternatives, points, items) which are described by multiple dimensions (also called attributes, criteria, features). The values of the cell are called attribute values (or called information function, records, etc.). Formally, Data Table (DT) is a 4-tuple \( DT = (U, Q, V, g) \), where \( U \) is a finite object set for \( x \in U \); \( Q \) is an attribute set for \( q \in Q \). The scale of values of attribute \( q \) is denoted by \( v \), for \( v = [v_q : q \in Q] \). And attribute values can be represented as \( g_q(x) : U \times Q \rightarrow V \) for \( g_q(x) \in V \). In general, attribute values can be in various forms like numbers, symbols and linguistic terms. But, they should be homogeneous with respect to a specific attribute.

Formally, Preference Table (PT) can be represented as a 3-tuple \( PT = (U, P, f) \). It includes a set \( U \) of objects \( x \); a set \( P \) of criteria \( p \); and a set of preference function \( f \). Criterion values related to \( x \) and \( P \) are the values of preference function with attribute values \( g_q(x) \) as independent variables. Thus, criterion values
also can be denoted as $f_j [g_j (x)]$ instead. Each singleton criterion $p_i$ is with a specific $f_j$. We call the set of $f_j$ as the preference system. For instance, denoting attribute values $g_j (x) = v(x, q)$ and criterion values $f_j [g_j (x)] = w(x, p)$, we understand $w(x, p)$ is the value of preference function $f$ with independent variable $v(x, q)$, hence $w(x, p) = f(v(x, q))$.

Remark that the simplest preference system is a set of unary linear functions. If the function is monotonic increasing, we call it GAIN type like $f(q) = q$. Otherwise, we call it COST type like $f(q) = -q$. In other words, values are preference-ordered. To our running case, the criteria “A” and “S” are with $f(q) = q$, and criterion “T” is with $f(q) = -q$. Most of literatures assume this simplest preference system in skyline computation.

2.3. Preference Intensity

Skyline operation strongly relies on subjective judgments. Thus once the defined preference system is imprecise, the corresponding skyline is hard to be satisfied. In this paper, we define the concept of preference intensity. Through specifying preference intensity of criteria, users have opportunities to adjust the settled preference system and thus control the resulting skylines. We define the preference intensity function as follows.

**Definition (preference intensity):**

Considering dominance relations of two criterion values $w(x, p_j)$ and $w(y, p_j)$ on criterion $p_j$, preference intensity is defined as:

$$G(x, y) = \rho(d) \quad \text{s.t. } d = w(x, p_j) - w(y, p_j)$$

The variable $d$ is the difference (D-values for short) of criterion value $x$ over $y$. The function value $G(x, y)$ is between 0 and 1. It requires users to preset certain intensity functions that reflect the users’ preference. In this paper, we recommend six generalized intensity functions [2, 17] as the commonly used types for users’ selection, as shown in Table 2.

The main value of preference intensity is to model users’ imprecise preference. In real applications, we consider that the conventional skyline operation is of type I, where $\rho(d) = 1$ means it is different between two criterion values. For other five types, they need a user-specified threshold on $G(x, y)$, which offers an opportunity to measure the similarity via various preference intensity functions $\rho(d)$. For example, Quasi type can tolerate the similarity of continuous criterion values. Gaussian type is able to integrate group preference via controlling the parameter $\epsilon (\epsilon > 0)$ (e.g., $\epsilon$ can be obtained according to the normal distribution of preference intensities, if multiple participants are involved). The last three types can be used in various problem domains by setting a threshold on $\rho(d)$. The preference intensity is considered in our definition of tolerant skyline. We will further demonstrate its effectiveness in section 5.1.

**Table 2** The six types of generalized criteria

<table>
<thead>
<tr>
<th>Type</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>I: Usual Criterion</td>
<td>$\rho(d) = \begin{cases} 1 &amp; \text{if } d \neq 0 \ 0 &amp; \text{if } d = 0 \end{cases}$</td>
</tr>
<tr>
<td>II: Quasi Criterion</td>
<td>$\rho(d) = \begin{cases} 1 &amp; \text{if } d &lt; -\epsilon \text{ or } d &gt; \epsilon \ 0 &amp; \text{if } -\epsilon \leq d \leq \epsilon \end{cases}$</td>
</tr>
<tr>
<td>III: Gaussian Criterion</td>
<td>$\rho(d) = 1 - \exp(-d^2/2\epsilon^2)$</td>
</tr>
<tr>
<td>IV: Level Criterion</td>
<td>$\rho(d) = \begin{cases} 1 &amp; \text{if }</td>
</tr>
<tr>
<td>V: Linear Criterion (i)</td>
<td>$\rho(d) = \begin{cases} 1 &amp; \text{if } d &lt; -\epsilon \text{ or } d &gt; \epsilon \ d/\epsilon &amp; \text{if } -\epsilon \leq d \leq \epsilon \end{cases}$</td>
</tr>
<tr>
<td>VI: Linear Criterion (ii)</td>
<td>$\rho(d) = \begin{cases} 1 &amp; \text{if }</td>
</tr>
</tbody>
</table>

2.4. The Tolerant Skyline

In this section, we define a new Tolerant Skyline Operation (T-skyline). It consists of three main concepts: dominance relations, dominance granules, and (non-)preferred T-skyline.

At first, dominance relations can be represented as follows. (1) For $x, y \in U$, object $x$ is dominating object $y$ under singleton criterion $p_j$ if $w(x, p_j)$ is superior or equal to $w(y, p_j)$ on preference function $f_j$, denoted as $x \preceq_D y$. (2) For $x, y \in U$, object $x$ is dominated by object $y$ under singleton criterion $p_j$ if $w(x, p_j)$ is inferior or equal to $w(y, p_j)$ on preference function $f_j$, denoted as $y \succeq_D x$. Hereinto, three terms need to be remarked. “Equal” means criterion values are same. “Superior” and “inferior” are typical outranking relations regarding the value of preference function.

The dominance relations are preserving the preference intensities. If adopting generalized criteria for identification, an expected D-value $d_{\epsilon}$ can be founded. It is subject to threshold of $G(x, y)$ and/or parameter(s) $\epsilon$. Supposing $G(x, y) = \rho(d)$ s.t. $d = w(x, p_j) - w(y, p_j)$, dominance relation can be given like: If $d \geq d_{\epsilon}$, then $w(x, p_j)$ is superior to $w(y, p_j)$, or $w(y, p_j)$ is inferior to
w(x, y). For example, users can specify $d_e = \varepsilon$ in Type III or $d_e = 0.2\varepsilon + 0.7e_0$ in Type IV. Again, if users specify a group-agreed threshold $G(x, y) \geq 0.8$ on Gaussian criterion with $\varepsilon = 0.5$, it is easy to obtain $|d| \geq 0.5\sqrt{2\ln 3}$ Thus, dominance relation can be defined as: If $d \geq d_e$ or $w(x, y) = w(y, x)$, then $x \not< y$.

According to dominance relations, we can define the dominance granules as: (i) the superiority set $D_s(x)$ where $D_s(x) = \{y \in U: \forall p, yD, x\}$, (ii) the inferiority set $D_i(x)$ where $D_i(x) = \{y \in U: \forall p, xD, y\}$. Then, the definition of tolerant skyline includes two parts: The preferred skyline $S_p(\lambda)$ and the non-preferred skyline $S_n(\lambda)$, where: $S_p(\lambda) = \{x \in U: |D_s(x)| \leq \lambda\}$ and $S_n(\lambda) = \{x \in U: |D_i(x)| \leq \lambda\}$. The number of objects in dominance granules is denoted by $|\cdot|$. The natural number $\lambda$, where $\lambda \in [1, \infty)$, is called tolerant degree. This coefficient is installed by users and usually alterable along with different needs and the size of data set. Let us remark this new definition below.

1. The dominance relation of T-skyline is based on the relaxed outranking relation. It eliminates the requirement of the conventional skyline operator, that is “criterion values should strictly be superior or inferior to that of any other objects at least on one criterion”.

2. Each object $x$ from the universe $U$ has two dominance granules. The superiority set includes object $x$ itself and all dominating objects which are with superior values at least on one criterion and not with inferior values on all criteria. Meanwhile, the inferiority set includes object $x$ itself and all dominated object which are with inferior value at least on one criterion and not with superior values on all criteria. Thus, we can obtain a property as: $D_s(x) \cap D_i(x) = [x]$.

3. Tolerant degree $\lambda$ is user-specified. It is used to control the extent of skyline for meeting users’ needs. In particular, we have $|D_s(x)| \leq 1$ and $|D_i(x)| \leq 1$ when $\lambda = 1$. It implies $D_s(x) = [x]$ and $D_i(x) = [x]$.

This new operation has two features. First, it considers two boundaries according to predefined preference: the preferred one and the non-preferred one. The former is used to provide available alternatives. And the latter provides adverse alternatives for users. Secondly, it employs the user-specified thresholds to control the level of skyline membership. It offers an opportunity to discover marginal skyline objects for consideration. Our empirical study in section 5.1 will demonstrate the advantage of such mechanism.

3. Tolerant Skyline Computation and Maintenance

This section provides algorithms to compute T-skyline and address continuous skyline maintenance. In the following, we just consider preferred T-skyline, since non-preferred skyline can be computed in the same manner.

3.1. The UA Method

Pairwise comparison operation is the naïve method in computing skylines. Each object in the universe compares its values of preference function with all other objects on all criteria, in order to obtain dominance granules of each object. In a table $PT:n \times k$, this operation can be done under time complexity $O(n^2)$. For $\exists x \in U$, one step of the iteration can partition the universe $U$ into two subsets. We define them as the comparable set $C_r(x)$ and the incomparable set $I_r(x)$.

The properties include: $C_r(x) \cup I_r(x) = U$ and $C_r(x) \cap I_r(x) = \emptyset$. Set $C_r(x)$ is constituted by superiority set $D_s(x)$ and inferiority set $D_i(x)$. According to their definitions, we can obtain the properties, including $D_s(x) \cup D_i(x) = C_r(x)$ and $D_s(x) \cap D_i(x) = [x]$. Then, the following assertions can be easily proved to be valid: (i) For objects $x, y \in U$, if $y \in D_s(x)$ is satisfied, we then have $D_s(y) \subseteq D_s(x)$ and $D_i(y) \supseteq D_i(x)$. (ii) For objects $x, y \in U$, if $y \in D_i(x)$ is satisfied, we then have $D_s(y) \subseteq D_i(x)$ and $D_i(y) \supseteq D_s(x)$. With respect to tolerant degree $\lambda$, the following assertions can be easily proved to be valid for $\exists x \in U$:

1. If $|D_s(x)| \leq \lambda$, we have $\bigcup_{y \in D_s(x)} D_i(y) \subset S_p(\lambda)$.
2. If $|D_i(x)| \leq \lambda$, we have $\bigcup_{y \in D_i(x)} D_s(y) \subset S_n(\lambda)$.
3. If $|D_s(x)| > \lambda$, we have $\bigcup_{y \in D_s(x)} D_i(y) \not\subset S_p(\lambda)$.
4. If $|D_i(x)| > \lambda$, we have $\bigcup_{y \in D_i(x)} D_s(y) \not\subset S_n(\lambda)$.

Based on above analysis, we can compute the T-skyline through frequently updating the universe. For $\exists x \in U$ and a given $\lambda$, if $|D_s(x)| \leq \lambda$ is satisfied in an iteration, then the inferiority set $D_i(x)$ can be eliminated from the universe for next iteration. If $|D_s(x)| \leq \lambda$ is satisfied in an iteration, then the superiority set $D_s(x)$ can be eliminated from the universe for next iteration, and also $D_s(x)$ can be accepted as T-skyline $S_p(\lambda)$. If $|D_s(x)| > \lambda$ is satisfied in an iteration, then the comparable set $C_r(x) = D_s(x) \cup D_i(x)$ can be eliminated from the universe for next iteration, and also $D_s(x)$ can be accepted as T-skyline $S_p(\lambda)$.

This update-approaching (UA) method aims to minimize the number of iterations by frequently updating the sets of objects. We present the UA method via pseudocode. Algorithm 1 calculates the dominance
granule of singleton object in consideration of dynamic preference system. This operation is called by other algorithms. Algorithm II conducts the UA computation.

Algorithm I: Calculation of dominance granules
Input: The singleton object \(x\); its criterion value \(w(x, p_j)\); each criterion \(p_j\) is with preference function \(f_j\) and preference intensity \(\rho\).
Output: Dominance granules \(D_p^\lambda(x)\) and \(D_p^\lambda(y)\).

1. for \(\exists j \in P\)
2. for \(\exists y \in U\)
3. compare \(w(x, p_j)\) with \(w(y, p_j)\) on \(f_j\) and \(\rho\)
4. if \(w(x, p_j)\) is superior or equal to \(w(y, p_j)\)
5. \(D_p^\lambda(x) \leftarrow y\)
6. if \(w(x, p_j)\) is inferior or equal to \(w(y, p_j)\)
7. \(D_p^\lambda(y) \leftarrow y\)
8. end if
9. end for
10. compute dominance granule
11. \(D_p^\lambda(x) = \bigcap_{j=1}^u D_p^\lambda(x)\) and \(D_p^\lambda(y) = \bigcap_{j=1}^u D_p^\lambda(y)\)
12. end for
13. return \(D_p^\lambda(x)\) and \(D_p^\lambda(y)\).

Algorithm II: The UA method
Input: Database and the user-specified tolerant degree \(\lambda\).
Output: Preferred T-skyline \(S_\lambda^\star\) with degree \(\lambda\).

1. Initialization: \(S_\lambda^\star = \emptyset\) and goal set \(\Delta = U\)
2. for \(\exists x \in \Delta\)
3. call Algorithm I on \(\exists y \in \Delta\)
4. if \(|D_p^\lambda(x)| > \lambda\) update \(\Delta = \Delta - D_p^\lambda(x)\)
5. then go to 2
6. else if \(|D_p^\lambda(x)| < \lambda\) update \(\Delta = \Delta - D_p^\lambda(x)\)
7. \(S_\lambda^\star = S_\lambda^\star \cup D_p^\lambda(x)\)
8. end if
9. end for

3.2. Continuous T-skyline Maintenance
Continuous skyline maintenance aims to keep the calculated skyline up-to-date after deleting “old” data. In general, this computation includes two aspects: (i) objects deletion when the criteria set is fixed, and (ii) criteria deletion when the objects are fixed. Many literatures have contributed to the issue of subspace skyline analysis (for example [14]). The concepts such as skyline group and decisive subspace are provided to explore relations among original criteria sets, criteria subsets and criteria supersets. These works have partly solved the second aspect. Nevertheless, to the best of our knowledge, rarely any paper tackle the first aspect. In this section, we provide the solutions for continuous maintenance of T-skyline when deleting objects with fixed criteria.

Suppose the deleted object set is denoted as \(V_r(\neq \emptyset)\) and the calculated T-skyline is denoted as \(S_\lambda^\star(\lambda)\). Obviously, the updated skyline should still be \(S_\lambda^\star(\lambda)\) if \(S_\lambda^\star(\lambda) \cap V_r = \emptyset\). However, in case \(S_\lambda^\star(\lambda) \cap V_r \neq \emptyset\), it is a bit complicated. We suppose the set \(N = S_\lambda^\star(\lambda) \cap V_r\). After deleting objects \(V_r\), the rest of T-skyline will be \(S_\lambda^\star(\lambda) - N\). Then, the dominance granules of objects from the set \(N\) will vary correspondingly. More specifically, the inferiority sets of objects from \(N\) need to be considered for the updated skyline. The union of these sets is \(\bigcup_{\lambda \in \lambda} D_p^\lambda(x)\). Therefore, the updated skyline can be obtained in consideration of a new object set \(\bigcup_{\lambda \in \lambda} D_p^\lambda(x) \cup (S_\lambda^\star(\lambda) - N)\). Algorithm III shows the pseudocode for this T-skyline maintenance.

Algorithm III: Continuous T-skyline maintenance
Input: Known preferred skyline \(S_\lambda^\star(\lambda)\); The user-specified tolerant degree \(\lambda\) where \(\lambda \leq \lambda\); Deleted object set \(V_r\).
Output: Updated preferred skyline \(\tilde{S}_\lambda^\star(\lambda)\)

1. Initialization: \(\tilde{S}_\lambda^\star(\lambda) = \emptyset\); goal set \(\Delta = \emptyset\); let set \(N = S_\lambda^\star(\lambda) \cap V_r\).
2. for \(\exists x \in N\)
3. \(D_p^\lambda(x), \tilde{D}_p^\lambda(x) \leftarrow \text{Algorithm I on } \exists y \in U\)
4. compute goal set \(\Delta = \bigcup_{\lambda \in \lambda} \tilde{D}_p^\lambda(x) \cup (S_\lambda^\star(\lambda) - N)\)
5. end for
6. for \(\exists x \in \Delta\)
7. \(D_p^\lambda(x), \tilde{D}_p^\lambda(x) \leftarrow \text{Algorithm I on } \exists y \in \Delta\)
8. if \(|D_p^\lambda(x)| > \lambda\) update \(\Delta = \Delta - D_p^\lambda(x)\)
9. else if \(|D_p^\lambda(x)| < \lambda\) update \(\Delta = \Delta - D_p^\lambda(x)\)
10. do \(\tilde{S}_\lambda^\star(\lambda) = \tilde{S}_\lambda^\star(\lambda) \cup \tilde{D}_p^\lambda(x)\)
11. then go to 6
12. else if \(|D_p^\lambda(x)| = \lambda\) update \(\Delta = \Delta - D_p^\lambda(x)\)
13. then go to 6
14. end for

There is a prerequisite for this method. Suppose the that the known T-skyline is \(S_\lambda^\star(\lambda \leq \lambda)\), the updated skyline can be obtained while \(\lambda = 1, 2, 3\). In another words, the setting degree \(\lambda\) of updated sky-
lines should not be larger than the degree \( \lambda \) of the known skylines, denoted as \( \lambda \leq \lambda \).

4. Related Works

This paper proposes a tolerant skyline operation and studies the related computing methods. In this operation, we attempt to model human preference and further make the resulting skyline more controllable. The literature related to our work can be classified into three categories: work that is related to MCDM and decision support system [3, 4, 6, 8, 9, 18, 19, 20, 23]; work that is related to the preference issue in skyline operation [10, 16, 22] and attempts to control the outputting size of skyline [11, 12, 15, 21]; and work that refers to basic computing methods of skyline operation [1, 5, 7].

MCDM aims at providing users (also known as decision makers) a knowledge recommendation amid a finite number of objects (also known as alternatives, actions, candidates) being evaluated from multiple viewpoints called criteria (also known as dimensions, attributes, features). A useful and comprehensive survey on MCDM was provided in [6]. Considering the latest developments of MCDM, we can outline the directions with the perspective of operational techniques: (i) dominance-based rough set approach [8], (ii) intuitionistic fuzzy methodology [4, 19], (iii) modeling preference in MCDM [18, 20]. This work, decision-oriented tolerant skyline with dynamic preference, can be located in the category (iii). Although the foundation of skyline operation in the database contexts has nearly completed via the developments in the last decade, it gradually starts to support other related fields like: using skyline operation as aggregation function to build data cubes for fast OLAP [23], reviewing the skyline processing in Peer-to-Peer systems [9], etc. In this paper, we concern the preference MCDM and have developed a tolerant skyline operation taking dynamic decision-making preference into account. As far as our knowledge, it is the first time to develop skyline operation for handling the MCDM problems. The proposed techniques in this paper are supposed to be implemented in various kinds of decision support system, for example [3].

Although previous studies have contributed various skyline operators, they always assume the fixed preference-ordered dataset. As shown before, this assumption is the simplest preference model, that will be subject to challenges in practice. The most straightforward method for computation is to transfer dynamic skylines to conventional skylines, thus making existing algorithms feasible. However, it needs full materializations of datasets and a time-consuming preprocessing. Wong et al. [16] pointed out such method is prohibitive in real applications, and hence provided a semi-materialization method via consideration of an implicit preference rather than ordered values. It has partly solved dynamic preference problems, yet is still limited in unitary linear preference function. Jiang et al. [10] studied preference relations in a specific problem domain. Yiu et al. [22] proposed preference query techniques for spatial database. All these literatures partly refer to the dynamic preference issue of skyline operations.

An inherent weakness of the conventional skyline operator is that the outputting size of skyline is nonflexible and uncontrollable. Yet our proposed model (i.e. dynamic preference and preference intensities) can make the resulting skyline be more flexible and adjustable. In addition, the T-skyline can tolerate several marginal points, and consequently it is more possible to acquire valuable points which are desired by users. Previous studies [11, 15] developed various methods for finding a representative subset of skylines. They can surely control the size of skyline but just works when skyline size is much larger than users’ desires. The recent works [12, 21] systematically studied how to constrain skyline, but both of them are still under the assumption of a preference-ordered data model.

Pairwise comparison is the basic method for skyline operation. The well known [1] provided two baseline algorithms. BNL compares every object with each other object and identifies its skyline membership once it cannot be dominated. D&C retrieves partial skyline from several subsets of data sets and merges all obtained partial skylines into a final result. SFS [5] developed BNL through firstly sorting the objects by a monotone function. And, LESS [7] further improve SFS through eliminating a part of objects in the sorting process. In this paper, the UA method requires us to calculate dominance granules and promptly update goal sets via eliminating objects. Although it seems very similar to the above-mentioned methods, they are different fundamentally. It is user-specified tolerant degree as the threshold value to decide whether superiority set or inferiority set should be eliminated. All these operations are based on relaxed dominance relations that are established on preference functions and preference intensities. Therefore, as primary computation for T-skyline, they are actually the development of naïve pairwise comparisons.

5. An Empirical Study

This section provides a detailed empirical study related to a real decision problem: NBA player evalua-
tion. In this study, we establish two cases to evaluate our method. We firstly use case (i) to compare T-skyline with different preference intensities. We secondly use case (ii) to illustrate the continuous maintenance of T-skyline. All programs are conducted on an Intel Core2 Duo CPU (T5750 @ 2.00GHz) CPU with 4.00 GB memory.

Player evaluation in NBA is a very frequent activity. This study employs the real NBA dataset in 2010-11 regular seasons. The original dataset contains 468 players with 26 attributes (data collected from http://espn.go.com/nba/). In accordance with practice, we only consider the player if and only if he has played 25 games or more (G ≥ 25). Then, 383 NBA players are enrolled in both case (i) and case (ii).

5.1. T-skyline with Preference Intensities

In this section, we use case (i) to demonstrate the resulting T-skyline with various preference intensities. Suppose the user is interested to know: Who is/are the most efficient player(s)? She firstly establish the dynamic preference system that involves three preference functions \( f_1, f_2, f_3 \). A criteria set is defined in \( P = \{ A, B, C \} \) as below.

1. Criterion A: Personal Efficiency per Game (EFF):
   
   \[
   \text{EFF} = f_1(\text{PV}, \text{NV}, G) = (\text{PV} - \text{NV}) / G, \text{ where } \\
   \text{Positive Values: } \text{PV} = \text{PTS} + \text{REB} + \text{STL} + \text{AST} + \text{BLK} \\
   \text{Negative Values: } \text{NV} = (\text{FGA} - \text{FG}) + (\text{FTA} - \text{FT}) + \text{TO} 
   \]

2. Criterion B: Shot Efficiency (SE):
   
   \[
   \text{SE} = f_2(\text{PTS}, \text{FT}, \text{FGA}) = (\text{PTS} - \text{FT}) / \text{FGA} 
   \]

3. Criterion C: Score per game AVG:
   
   \[
   \text{AVG} = f_3(\text{AVG}) = \text{AVG} 
   \]

Remark that criterion A is used to detect the overall efficiency of basketball players. It is actually a common-used criterion in real NBA scenario. Criterion B is to detect the efficiency of shot. It means the average non-Free-Throw (FT) personal total score (PTS) of each field goal attempt (FGA). These criterion values should be around 1. Criterion C is the total score per game (GAIN type).

Using UA method, we can compute the T-skyline with respect to \( \lambda = 1, 2, 3, \ldots \) as shown in the first row of Table 3. With setting \( \lambda = 1 \), it can return 9 players which is as the same as the conventional skyline after materializing 3-dimensional preference table. Besides, the proposed T-skyline operation additionally provides 7 players when setting \( \lambda = 2 \) (e.g. R.Allen) and 6 players when setting \( \lambda = 3 \) (e.g. C.Anthony). Outperforming conventional skyline operators, the T-skyline operator can additionally reveal some favored NBA players why they can or cannot turn into the skyline membership. For example, the user is interested in the skyline membership of the famous player Kobe Bryant. The proposed T-skyline operator can return the result as: K.Bryant belongs to the 6-skylines*. Hereinto, in addition to the 22 players shown in the first row of Table 3 (from \( \lambda = 1 \) to \( \lambda = 3 \)), there are 9 players as 4-skylines (e.g. S.Nash) and 4 players as 5-skylines (e.g. D.Rose). All of them are more efficient than B.Kobe in 2010-11 NBA regular seasons.

The second advantage of T-skyline operator is that the hierarchical results can make up the possibly defective (predefined) preference system. In the real world, the user-specified preference system might be imperfect. Taken case (i) as example, obviously obtaining one rebound (REB)/assist (AST)/block shot (BLK) is much harder than getting one score from shot. That is to say, under such player-evaluation system, it may be unfair for those good defensive players (particularly the CENTERs like A.Bynum and P.Gasol). They are qualified as the skyline membership with \( \lambda = 3 \) via T-skyline operation, although they are neglected by previous skyline operations.

In above experiments, the used preference intensity is PI-1 as shown in Table 4. In the following, we calculate T-skyline with another two assemblies of Preference Intensities (i.e. PI-2 and PI-3), and make the comparison. The preferred T-skyline with PI-2 is illustrated in Figure 1(a). We show the first three hierarchical T-skyline as 1-skylines (shown via the symbol of point), 2-skylines (shown via the symbol of star), and 3-skylines (shown via the symbol of circle). The name of the corresponding players is also marked in this figure. Figure 1(b) illustrates the preferred T-skyline with PI-3. In this figure, 5 players (T.Chandler, N.Hilario, D.Howard, L.James, and D.Jordan) are marked as the 1-skylines. It further includes 3 players as 2-skylines and 4 players as 3-skylines. By contrast, we can clearly view the changes with respect to the different PIs.

As illustrated in Figure 1, under the same preference system, it brings the multiple results due to the changes of intensity function \( \rho(\lambda) \). Table 3 shows the detailed results with three kinds of PIs. Compared with PI-1, both PI-2 and PI-3 narrowed down the number of

* For simplicity, the objects in each hierarchical T-skyline set with respect to the tolerant degree \( \lambda \) can be represented as \( \lambda \)-skylines. Thus, 6-skylines means the objects with the skyline membership when \( \lambda = 6 \) excluding the objects with the skyline membership when \( \lambda = 5 \).
qualified skyline points in each hierarchy. Particularly, 4 players (i.e. K.Durant, S.Novak, D.Nowitzki, S.O’Neal) are no longer as 1-skylines. Some players also vary their skyline membership along with different PIs. For example, the player D.Nowitzki is as 1-skylines of PI-1, 2-skylines of PI-3, and 3-skylines of PI-2.

With respect to the same preference system, the different settings of PIs can generate the multiple skyline results, which bring the benefits for better decision performances. Generally speaking, skyline operations contain an inherent property: an object, which is with the most preferred value in at least one criterion, tends to be the 1-skylines. Such property is adverse in some situations. For instance, the user may feel hard to accept the experimental result that S.Novak is as good as D.Nowitzki (being 1-skylines of PI-1). The reason is that S.Novak is with the most preferred values in both Criteria B and Criterion C. Such mechanism makes the user-specified evaluation system be questionable. In T-skyline, via setting preference intensity functions, the expected D-value \( d_0 \) can control the similarity in dominance relations. Thus, various kinds of human factors (e.g. group opinions, statistical results, etc.) can be further taken into account in the process of player evaluation. And, extreme values that have strong influence to the final resulting skyline are eliminated. In our experiments of PI-2 and PI-3, S.Novak is no longer with any skyline membership when setting \( \lambda = 1 \), \( \lambda = 2 \) or \( \lambda = 3 \) (see Figure 1). This mechanism is beneficial to many real-world applications.

<table>
<thead>
<tr>
<th>Table 3</th>
<th>The results of T-skyline with various preference intensities (i.e. PI-1, PI-2, and PI-3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PI</td>
<td>1-skylines</td>
</tr>
<tr>
<td>PI-3</td>
<td>T.Chandler, N.Hilario, D.Howard, L.James, D.Jordan</td>
</tr>
</tbody>
</table>

Table 4 Three assemblies of preference intensity

<table>
<thead>
<tr>
<th>PI</th>
<th>Criterion A</th>
<th>Criterion B</th>
<th>Criterion C</th>
</tr>
</thead>
<tbody>
<tr>
<td>PI-1</td>
<td>Type I and user-specified ( G(x,y) = 1 )</td>
<td>Type I and user-specified ( G(x,y) = 1 )</td>
<td>Type I and user-specified ( G(x,y) = 1 )</td>
</tr>
<tr>
<td>PI-2</td>
<td>Type III with ( \varepsilon = 0.5 ), and user-specified ( G(x,y) \geq 0.8 )</td>
<td>Type II with ( \varepsilon = 0.05 ), and user-specified ( G(x,y) = 1 )</td>
<td>Type V with ( \varepsilon = 0.5 ), and user-specified ( G(x,y) \geq 0.8 )</td>
</tr>
<tr>
<td>PI-3</td>
<td>Type III with ( \varepsilon = 0.5 ), and user-specified ( G(x,y) \geq 0.9 )</td>
<td>Type II with ( \varepsilon = 0.08 ), and user-specified ( G(x,y) = 1 )</td>
<td>Type V with ( \varepsilon = 0.5 ), and user-specified ( G(x,y) \geq 0.9 )</td>
</tr>
</tbody>
</table>

Figure 1 Tolerant skyline with different preference intensities

5.2. T-skyline and Continuous Maintenance

In this section, we use case (ii) to demonstrate continuous T-skyline maintenance. Suppose the user is interested to know: Who is/are the most efficient...
ball-stealer(s)? We firstly establish the dynamic preference system that includes three preference functions \( \{ f_1, f_2, f_3 \} \). A criteria set is defined in \( P = \{ A, B, C \} \) below.

1. **Criterion A:** \( f_1(\text{STL}, \text{TO}) = \frac{\text{STL}}{\text{TO}} \)
2. **Criterion B:** \( f_2(\text{STL}, \text{PF}) = \frac{\text{STL}}{\text{PF}} \)
3. **Criterion C:** \( f_3(\text{STL}, \text{MIN}) = \frac{\text{STL}}{\text{MIN}} \times 48 \)

Remark that criterion A is the ratio of the number of steal (STL) and the number of turnover (TO). Criterion B is the ratio of the number of steal and the number of personal foul (PF). Criterion C is the number of steal per 48 minutes. In practice, this preference system is commonly used for evaluation of the GUARD player’s ability in ball-stealing.

### Table 5 The comparison of T-skyline after object deletion

<table>
<thead>
<tr>
<th>T-skyline</th>
<th>1-skylines</th>
<th>2-skylines</th>
<th>3-skylines</th>
<th>n-skylines</th>
</tr>
</thead>
<tbody>
<tr>
<td>n=383</td>
<td>B.Ronnie, L.Jeremy, R.Jason, W.Jason</td>
<td>A.Tony, K.Jason, P.Chris</td>
<td>B.Corey, D.Carlos, S.Thabo</td>
<td>...</td>
</tr>
<tr>
<td>n=332</td>
<td>A.Tony, B.Ronnie, P.Chris, R.Jason, W.Jason</td>
<td>B.Corey, D.Carlos, K.Jason, S.Thabo</td>
<td>A.Ron, E.Monta, J.Jared, W.Julian</td>
<td>...</td>
</tr>
</tbody>
</table>

Figure 2 illustrates the comparable T-skyline results. The preferred T-skyline without object deletion is illustrated in Figure 2(a). We show the first three hierarchical T-skyline as 1-skylines (using the symbol of star), 2-skylines (using the symbol of box), and 3-skylines (using the symbol of circle). The names of the corresponding players are also marked in this figure. Several players are also marked to be non-preferred T-skyline.

### 6. Conclusion

This paper proposes a dynamic tolerant skyline operator for decision supports. We firstly investigate preference relations on conventional skyline operation and thus establish a dynamic preference model through introduction of the concept of preference intensity. Based on the proposed model, we define a new decision-oriented skyline operator called T-skyline. The results of this operator are a set of hierarchical skylines with respect to the coefficients called tolerant degree. This new operator provides multiple parameters which make the outputting size of skyline adjustable and partly controllable. In the meanwhile, such flexible mechanism offers the opportunity for decision-makers to correct the possibly imperfect (predefined) preference system. For T-skyline computation, we provide the polynomial-time UA algorithm. And also, we study...
the method for continuous T-skyline maintenance. Finally, we elaborate an empirical study on NBA player evaluation of the 2010-11 regular seasons. We use it for demonstrating the effectiveness and advantages of this new operation, as well as illustrating its continuous maintenance. As the conclusion, the new T-skyline operator outperforms conventional skyline operators in solving the real-world decision problems and the developed methods are effective and practical. The limitation is that the outputting size of T-skyline is still not fully controllable. This is a challenging problem that will be a part of our future work.

7. Acknowledgements

The authors are grateful for the partial support of CRG grants G-U756, G-YL14 of The Hong Kong Polytechnic University.

8. References