# Performance Evaluation of Preference Queries Techniques over a High Multidimensional Database

Ali A. Alwan, Hamidah Ibrahim, Tan Chik Yip, Fatimah Sidi, Nur Izura Udzir, Nurul Husna Mohd Saad
Department of Computer Science
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
43400 Serdang, Selangor D. E
Malaysia
{ali83_upm@yahoo.com, hamidah@fsktm.upm.edu.my, chikyiptan@gmail.com,
{fatimacd, izura}@fsktm.upm.edu.my, n-husna@hotmail.com}

***ABSTRACT:*** *In recent years, there has been much focus on the design and development of database management systems that incorporate and provide more flexible query operators that return data items which are dominating other data items in all attributes (dimensions). This type of query operations is named preference queries as they prefer one data item over the other data item if and only if it is better in all dimensions and not worse in at least one dimension. Several preference evaluation techniques for preference queries have been proposed including top-k, skyline, top-k dominating, k-dominance, and k-frequency. All of these preference evaluation techniques aimed at finding the "best" answer that meet the user preferences. This paper evaluates these five preference evaluation techniques on real application when huge number of dimensions is the main concern. To achieve this, a recipe searching application with maximum number of 60 dimensions has been developed which assists users to identify the most desired recipes that meet their preferences. Two analyses have been conducted, where execution time is the measurement used.*

**Key words:** Preference Queries, Preference Evaluation Techniques, Skyline, Top-*K*, Query Processing

**Received:** 12 April 2011, Revised 29 May 2011, Accepted 3 July 2011



## 1. Introduction

In the recent years, there has been much focus to design and develop database management systems that incorporate and provide more flexible query operators that best fit the user preference and limit the result sets. The preference queries are used in many application domains, like multi-criteria decision making applications [4, 5, 21-23], where many criteria are involved to select the most suitable answer to the user query. Decision support system helps to combine various interests to recommend a strategic decision. Other domains include e-commerce environments like trade off between the price, quality, and efficiency of the products to be assessed; personal preferences of users who request a web service such as hotel recommender [29] and restaurant finder [6, 25]; and peer-to-peer network [16]. In this regards, there are many preference evaluation methods that have been proposed including but not limited to top-*k* [31], skyline [30], *k*-dominance [21], top-*k* dominating [5], and *k*-frequency [4]. The ultimate goal of these preference evaluation methods is to reduce the search space and improve the quality of the given answer by providing the best possible relevant answer with respect to the given conditions (preferences).

This paper is an extension of our previous work [1] which attempts to examine the most recent techniques of preference evaluation of query processing in the database systems, namely: top-*k*, skyline, top-*k* dominating, *k*-dominance, and *k*-frequency when huge number of dimensions is to be considered. The evaluation should be performed on real application. Thus, we have purposely developed a recipe searching application which offers a variety of recipes that best meet the ever-changing demands of user. We focus on various consumers as every user whenever attempts to find the most suitable recipe will consider several sources of information before deciding which recipe to be chosen.

The reasons for choosing the recipe domain to evaluate the performace of the preference evaluation techniques are mainly due to: (i) each recipe normally consists of several components like ingredients, course types, cuisine types, cooking method, occasions, diet and others while the requirements of the end user are multi-dimensional and cannot be easily expressed on discrete scales. In this paper 60 dimensions have been identified. (ii) The main critical issue is a recipe component ratio which is defined by what is known, as the "best" recipe for user. To tackle this, the preference evaluation techniques that consider the ratio and rank results accordingly to the user requirements are the best techniques to be used and evaluated.

This paper is organized as follows. In Section 2, the previous works related to this work is presented and discussed. In Section 3, the recipe searching application is introduced. Performance analysis is presented and discussed in Section 4. Conclusions are presented in the final section, 5.

## 2. Related Works

Many types and variations of preference evaluation techniques of preference queries have been described in the database literature. These techniques include but not limited to: top-*k*, skyline, *k*-dominant skyline, skyline frequency, top-*k* dominating, ranked skyline, skycube, sort and limit skyline algorithm (SaLSa), SUBSKY, sort-filter-skyline (SFS), linear elimination sort for skyline (LESS), and Z-Sky. Most of these preference evaluation techniques aim to improve the search performance by terminating the process of searching the data items as early as possible in obtaining the "best" answer that satisfies the conditions as indicated in the submitted query. In the following we present the most important types of preference evaluation techniques in preference queries.

**Top-*K*:** Given a set of data items with *d* dimensions (attributes) and a monotonic preference ranking function *F*, top-*k* technique retrieves a selected set of data items (*k*) that dominates the data items according to the best scoring value based on *F*.

The basic concept of this technique is to give score (weight) to each data item in the database. Thus, in order to produce the scoring results a preference ranking function (monotone function) is involved to accumulate the values of dimensions for each particular data item. Then depends on the final results of the preference ranking function, the *k*-data items with the best scores are considered the preferred data items [3, 9-10, 15, 17, 20, 23, 24, 26-27, 31]. Several algorithms have been proposed based on the top-*k* preference evaluation technique such as *Onion* [33], PREFER [32], *Mpro* [17], Top-*k* Monitoring Algorithm (TMA) [20], SPEERTO [3], and Skyband Top-*k* Monitoring Algorithm (SMA) [20]. However, these algorithms are being evaluated on small scale of dimensions within the range 2-7.

**Skyline:** The skyline preference evaluation method produces the set of data items in a way such that the set of data items *S* are the superior among the other data items in the dataset. In other words, a data item *p* is preferred over another data item *q* if and only if *p* is as good as *q strictly* in at least one possible dimension (attribute) and in all other *n* dimensions (attributes) [2, 4, 5, 8, 13, 15, 21, 23, 25, 29, 30]. Skyline queries are considered as one of the most widely used queries in preference queries for several types of multidimensional database systems. Several algorithms have been proposed based on the skyline preference evaluation technique such as Block-Nested-Loop (BNL) and Divide-and-Conquer (DC) [30], Sort-Filter-Skyline (SFS) [12], Linear Elimination Sort for Skyline (LESS) [28], Nearest Neighbor (NN) [12], Branch-Bound-Skyline (BBS) [7], Bitmap and Index [19], SkyCube [30] Sort and Limit Skyline algorithm (SaLSa) [11], and Z-Sky [18] but these algorithms are being evaluated on small scale of dimensions within the range 2-10.

**Top-*K* Dominating:** Top-*k* dominating technique retrieves the set of data items *k* which are dominating the largest number of data items in the dataset. That means data item *p* is preferred over another data item *q* if and only if the domination power of *p* is greater than the domination power of *q*. The value of domination power of data item *p* comes from the total number of data items in the dataset that are dominated by *p*. Top-*k* dominating technique is a very significant tool for multi-criteria application such as decision making system and decision support system, since it identifies the most significant data items in an intuitive way [14, 15, 21].

***K*-Dominance:** *K*-dominance skyline technique prefers one data item *p* over another data item *q* in the dataset *D* if and only if *p* is as good as *q strictly* in at least one possible *k*-dimension (attribute) and in the subset of *k* dimensions (attributes) where *k* is less than the total number of dimensions.

*K*-dominance exhibits some characteristics over the traditional skyline. The size of *k*-dominance skyline answer is less than the size of conventional skyline answer, particularly when the considered dimensions are few. Furthermore, *k*-dominance has some similar characteristics with skyline technique especially when $k = d$ (*d* is the total number of dimensions in the dataset). However, *k*-dominance skyline suffers from a significant problem which is *circular dominance* that leads to loss the transitivity property [5, 15, 34-35].

***K*-Frequency Skyline:** *K*-frequency skyline technique retrieves a set of skyline data items $D'$ from the given dataset *D* in a space *S*, where a data item *p* in $D'$ has the lowest dominating score, denoted as $S(p)$, which represents the number of available sub-dimensions where *p* is not a skyline.

*K*-frequency has many common characteristics with skyline technique such as transitivity property is preserved, and the *k*-frequency queries' answers are obtained by merely comparing the actual values for each identical dimension in two different data items. Further, this technique can be applied in the full space and subspace dataset. However, *k*-frequency needs a powerful data structure that saves the dominated sub-dimensions for every data item *p* in order to precisely determine the score of every data item *p* [4, 15].

**Example:** The following example illustrates the five preference evaluation techniques that are considered in this paper. Assuming a database consists of 7 data items with 3 dimensions as depicted in Figure 1. Suppose that the preference ranking function *F* is the sum of these dimensions' values of each data item (i.e., $F = d1 + d2 + d3$) and the smallest value is the preferable. Thus, based on the proposed scoring function and assuming $k = 2$, the data items *t*1 and *t*2 are the results of top-*k* technique as their scores is the smallest.

Referring to the same example and by applying the skyline technique, the skyline answers are the data items *t*1 and *t*2 (smaller values are preferable) as these data items are the best in all dimensions over the other data items. Since *t*1 is better than *t*2 in dimension *d*1, while *t*2 dominates *t*1 in dimension *d*2, thus both of them are skyline as none of them completely dominate each other.

Applying the top-*k* dominating technique produces the data items *t*1 and *t*2 as the dominating score of data item *t*1 is 5 since *t*1 dominates *t*3, *t*4, *t*5, *t*6, and *t*7 in all dimensions, and the dominating score of *t*2 is 4 since *t*2 dominates *t*3, *t*4, *t*5, and *t*7 in all dimensions.

Implementing *k*-dominance technique on the example database by assuming the value of $k = 2$ which indicates the number of preferred dimensions, results into *t*1 as the *k*-dominance answer, if we prefer the data items that have the lowest values in the dimensions *d*1 and *d*3. However, if the value of $k = 3$ *k*-dominance reverts a skyline technique.

Applying *k*-frequency preference evaluation technique on the example database produced *t*1 and *t*2 data items as the results. The score of data item *t*1 is 1, as it can only be dominated in a single sub-dimension *d*2 by the data item *t*2. Furthermore, the score of data item *t*2 is 3 since it is dominated by the data item *t*1 in the sub-dimensions *d*1 and *d*3 (i.e., {*d*1}, {*d*3}, {*d*1, *d*3}). Lastly, the data item *t*7 has a score of 7 as it is dominated by the data item *t*1 in all possible sub-dimensions.

## 3. The Recipe Searching Application

The proposed recipe searching application has been successfully implemented using PHP web programming language and SQL server. Each preference technique has been developed and tested with respect to different type of recipes. We have identified six elements which are important in searching and later selecting a particular recipe. These elements are type of ingredients, courses, cooking methods, occasions, diet restrictions, and type of cuisines. Each element has its own set of dimensions (attributes) that can be selected. All together there are 60 dimensions. A range of 0-5 has been prepared for each dimension which indicates the degree of interest by a user towards a particular dimension. The smallest scale, 0, denotes no interest at all while the scale 5 denotes the highest preferences. Table 1 summarizes these dimensions. We use the notation *di* to denote the *i*th dimension.

| *id* | *d1* | *d2* | *d3* | *Score* |
|---|---|---|---|---|
| *t*1 | 4 | 2 | 2 | 8 |
| *t*2 | 7 | 1 | 3 | 11 |
| *t*3 | 9 | 5 | 4 | 18 |
| *t*4 | 8 | 6 | 3 | 17 |
| *t*5 | 9 | 5 | 7 | 21 |
| *t*6 | 6 | 7 | 9 | 22 |
| *t*7 | 10 | 7 | 12 | 29 |

Figure 1. Example of Database

| Element | Number of dimensions |
|---|---|
| Main Ingredient | 16 (*d*1 – *d*16) |
| Course | 12 (*d*17 – *d*28) |
| Cooking Method | 8 (*d*29 – *d*36) |
| Occasion | 8 (*d*37 – *d*44) |
| Diet | 8 (*d*45 – *d*52) |
| Cuisine | 8 (*d*53 – *d*60) |

Table 1. Dimensions of the Recipe Searching Application

Figure 2 illustrates the main design interface of the proposed recipe searching application. The application provides several features for the user before a particular recipe is selected. These features include (i) users can select the preference evaluation tecnique they prefer; (ii) users are free to ignore any dimensions that are not interest to them. By default all dimensions are assigned the value 0; and (iii) users may rank the dimensions according to their needs by manipulating the scale to be assigned to the needed dimensions. For example, the following table represents a query submitted by a user.

| Element | Dimensions selected |
|---|---|
| Main Ingredient | *d*1 = 5; *d*2 = 3 |
| Course | *d*18 = 4 |
| Cooking Method | *d*29 = 4 |
| Occasion | *d*43 = 5 |
| Diet | *d*46 = 4 |
| Cuisine | *d*54 = 5 |

Note: *d*1 (chicken); *d*2 (rice); *d*18 (dinner); *d*29 (baking); *d*43 (Christmas); *d*46 (healthy); *d*54 (Italian)

Table 2. Example of dimensions selected in a User Query

After selecting the appropriate dimensions by giving the suitable preference value, then user is required to determine the type of preference technique before the application finds the recipes. The default preference evaluation is the skyline. The best 20 first results will be shown based on the preference feature scoring and the preference evaluation technique that has been chosen. For the aim of this paper, 150 recipes have been collected and saved in a database called the Recipe Database (*RDb*).

Several steps are initially performed before the preference evaluation techniques are being applied. These steps mainly aim at removing the irrelevant data items (records) from the Recipe Database from being considered in the evaluation process as they will not contribute to the final result. The steps are discussed below:


Figure 2. The Main Interface Design of the Recipe Searching Application

1. Each recipe from the *RDb* is mapped into a two dimesional array, *RA*, with the following format:

Structure of *RA*

| Index | 0 | 1 | 2 | 3 | … | 60 |
|---|---|---|---|---|---|---|
| Dimension | *Id* | *d*1 | *d*2 | *d*3 | … | *d*60 |

Where *Id* is the identifier of the recipe and *di* is a score given to the *i*th dimension. We use the notation $r_k.di$ to denote the *i*th dimension of the *k*th recipe. An example of a recipe stored in the array is as follow:

An instance of *RA*

| Index | 0 | 1 | 2 | 3 | … | 60 |
|---|---|---|---|---|---|---|
| Element | 101 | 5 | 0 | 5 | … | 5 |

The above is an information about the recipe 101 which uses chicken (*d*1) as the main ingredient, vegetable (*d*3), …, and Southwestern (*d*60) is the main cuisine.

2. Given a query, *Q*, with a set of *n* selected dimensions, *dq* = {*dq*1, *dq*2, …, *dqn*} only those recipes in the *RA* that matched with these dimensions are selected and stored in a temporary array, *TRA*. The following definition defined the match criteria.

*Definition* 1: A recipe $r_k$ is said to be matched to a query *Q* if $\exists dqi \in dq, \exists dj \in r_k$ and $r_k.dj > 0$ where *j* is the equivalent dimension as *i*.

This gives the following definition which defined the unmatched criteria.

*Definition* 2: A recipe $r_k$ is said to be unmatched to a query *Q* if $dqi \in dq, \exists dj \in r_k$ and $r_k.dj = 0$ where *j* is the equivalent dimension as *i*.

The following example clarifies this point. Consider the query given in Table 2 and the following instances of *RA*.

User query

| Index | *d*1 | *d*2 | … | *d*18 | … | *d*29 | … | *d*43 | … | *d*46 | … | *d*54 | … |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Element | 5 | 3 | … | 4 | … | 4 | … | 5 | … | 4 | … | 5 | … |

Note: The other dimensions have the value 0

Instances of *RA*

| Index | *Id* | *d*1 | *d*2 | … | *d*18 | … | *d*29 | … | *d*43 | … | *d*46 | … | *d*54 | … |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Element | 102 | 5 | 5 | … | 5 | … | 5 | … | 5 | … | 5 | … | 5 | … |
| | 103 | 0 | 0 | … | 0 | … | 0 | … | 0 | … | 0 | … | 0 | … |
| | … | … | … | … | … | … | … | … | … | … | … | … | … | … |
| | 110 | 0 | 5 | … | 0 | … | 0 | … | 0 | … | 0 | … | 0 | … |

Note: The other dimensions that are not listed in the table might have value 0, 1, 2, 3, 4, or 5 while for 103 we assume that all values are zero

From the above instances of *RA*, recipe *r*.102 and *r*.110 satisfied the Definition 1 and are selected while *r*.103 is ignored as for all the dimensions requested by the user have the value = 0 (satisfied the Definition 2).

3. Those dimensions in the temporary array, *TRA*, which are not considered in the query, *Q*, are then eliminated to reduce the size of dimensions to be considered. Based on the example given in Step 2 above, the following is the result of Step 3.

Instances of TRA

| Index | Id | d1 | d2 | d18 | d29 | d43 | d46 | d54 |
|---|---|---|---|---|---|---|---|---|
| Element | 102 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| | … | … | … | … | … | … | … | … |
| | 110 | 0 | 5 | 0 | 0 | 0 | 0 | 0 |

4. The preference evaluation techniques are then applied towards the recipes that have been saved in the TRA. Figure 3 illustrates the interface design of the searching result of the skyline, top-*k*, top-*k* dominating, *k*-dominance, and *k*-frequency skyline techniques where the selected dimensions and the given rate is the same for each technique.


(a) The Result of Skyline Query


(b) The Result of Top-k Query

## 4. Performance Evaluation

We have conducted two analyses. The first analysis aims at analyzing the performance of the preference evaluation techniques with respect to the total number of dimensions that represents the user's preferences. In this paper we vary the number of dimensions from 10 – 60 dimensions, while the size of the recipe database is fixed. The second analysis aims at comparing the preference evaluation techniques with respect to the size of recipe database while the number of dimensions is fixed during the process of searching the best recipes that meet the user's request. In this paper we focused exclusively on the number of dimensions and the size of databases as they are the most critical factors which influence the process of finding preference answer.

### 4.1 Results of Analysis 1

Figure 4 shows the results of applying different number of dimensions with fixed number of data items (recipe),which is 100. The initial number of dimensions is 10 and it is incrementally increased by 10, until the number of dimensions reached 60, which is the maximum number of dimensions considered in this paper. All together there are 6 experiments that have been conducted whereby in each experiment the number of dimensions considered is different. For each experiment 10 queries have been randomly generated where each query selects the appropriate number of dimensions (see Step 2 of Section 3). The execution time of each query is measured when Step 4 as described in Section 3 is performed. Averaging the execution time of these 10 queries gives the final execution time of the experiment. Thus, six different sets of queries have been designed for this analysis. The following table summarizes our experiment set up for this analysis.


(c) The Result of Top-*k* Dominating Query


(d) The Result of k-Dominance Query

| Experiment | Query | Number of dimensions | Number of Recipes |
|---|---|---|---|
| Experiment 1 | Q1, Q2, …, Q10 | 10 | 100 |
| Experiment 2 | Q11, Q12, …, Q20 | 20 | 100 |
| Experiment 3 | Q21, Q22, …, Q30 | 30 | 100 |
| Experiment 4 | Q31, Q32, …, Q40 | 40 | 100 |
| Experiment 5 | Q41, Q42, …, Q50 | 50 | 100 |
| Experiment 6 | Q51, Q52, …, Q60 | 60 | 100 |
| Experiment 6 | Q51, Q52, …, Q60 | 60 | 100 |

Table 3. Experiments for the analysis 1

From the above figure, the following can be concluded: in general the amount of execution time to retrieve the query answer increased for all the preference evaluation techniques when the number of dimensions increased. Top-*k* technique is the best as the increment rate of the execution time to obtain the query result is the lowest while skyline, *k*-dominance, and *k*-frequency achieved almost the same execution time. However, top-*k* dominating technique performs the worst compared to the other techniques as the execution time increased dramatically when the number of dimensions increased. From this analysis, we can conclude that the number of dimensions involved in the process of preference queries has significant impact on the execution time in searching the “best” answer that meet the users’ preferences for most of the preference evaluation techniques.

Moreover, this simple analysis shows that applying different type of preference evaluations give different impacts to the performance of the preference queries.

### 4.2 Results of Analysis 2

Figure 5 shows the results of applying different number of recipes which reflects the size of database with fixed number of dimensions, which is 10. The initial number of recipes is 10 and it is incrementally increased by 10, until the number of recipes reached 100, which is the maximum number of recipes considered in this analysis. All together there are 10 experiments that have been conducted whereby in each experiment the number of recipes considered is different. For each experiment 10 queries have been randomly generated where each query selects 10 dimensions (see Step 2 of Section 3). The execution time of each query is measured when Step 4 as described in Section 3 is performed. Averaging the execution time of these 10 queries gives the final execution time of the experiment. The following table summarizes our experiment set up for this analysis.


(e) The Result of k-Frequency Skyline Query

Figure 3. The Results of Preference Evaluation Techniques

From the above figure, it is obvious that the top-k technique has the lowest amount of execution time compared to the other four techniques. This is due to the fact that most of the process in finding the best query answer is performed without needing to compare the individual dimensions at the data item level to determine the query results. i.e. accumulate the values of all dimensions as a single value. However, k-dominance, k-frequency and skyline techniques achieved almost the same amount of increment in the execution time when the number of recipes (the size of database) is increased. However, top-k dominating has the worst performance compared to the other techniques.

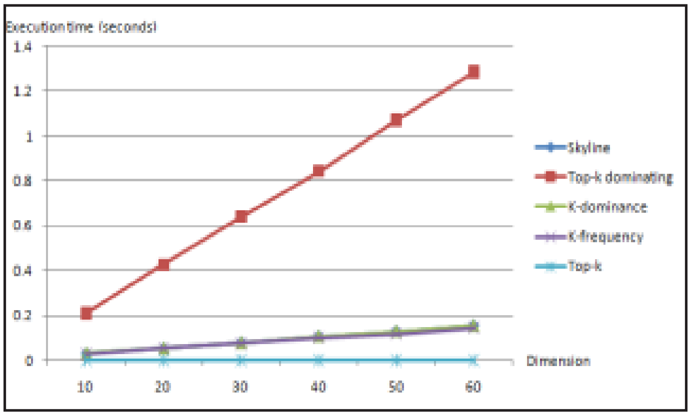


Figure 4. The Amount of Execution Time with Respect to the Number of Dimensions

| Experiment | Query | Number of dimensions | Number of Recipes |
|---|---|---|---|
| Experiment 1 | Q1, Q2, …, Q10 | 10 | 10 |
| Experiment 2 | Q1, Q2, …, Q10 | 10 | 20 |
| Experiment 3 | Q1, Q2, …, Q10 | 10 | 30 |
| Experiment 4 | Q1, Q2, …, Q10 | 10 | 40 |
| Experiment 5 | Q1, Q2, …, Q10 | 10 | 50 |
| Experiment 6 | Q1, Q2, …, Q10 | 10 | 60 |
| Experiment 7 | Q1, Q2, …, Q10 | 10 | 70 |
| Experiment 8 | Q1, Q2, …, Q10 | 10 | 80 |
| Experiment 9 | Q1, Q2, …,Q10 | 10 | 90 |
| Experiment 10 | Q1, Q2, …, Q10 | 10 | 100 |

Table 4. Experiments for the analysis 2

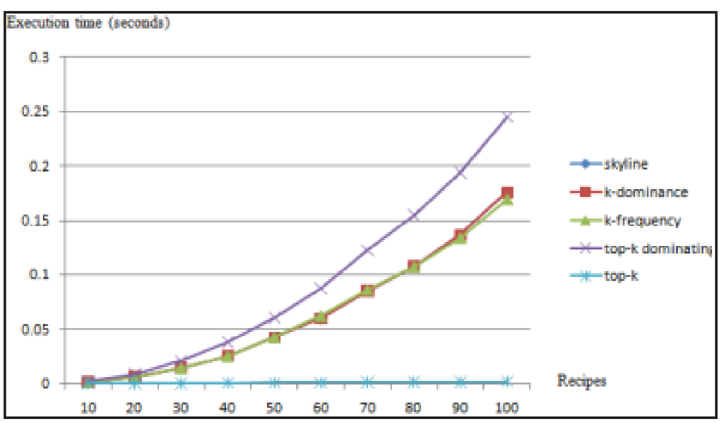


Figure 5. The Amount of Execution Time with Respect to the Database Size

**5. Conclusion**

In this paper we have presented and discussed a recipe searching application which has been developed with the aim to evaluate the various types of preference evaluation techniques for preference queries. Two analyses with different aims have been conducted by considering various numbers of dimensions and sizes of the databases. These are the most significant factors that impact the execution time of the preference evaluation techniques in searching for the "best" query answer that meets the users' preferences. We have also shown that the best preference technique in term of execution time is top-*k*, while the worst is top-*k* dominating.

## 6. References


[1] Ali, A.A., Hamidah, I., Tan C.Y., Fatimah, S., Nur Izura, U. (2011). Preference evaluation of preference queries techniques over a high multidimensional database. *In*: Proc. of the International Conference on Networked Digital Technologies (NDT2011), p. 212–223. Springer-Verlag Berlin Heidelberg 2011.

[2] Ali, A.A., Hamidah, I., Nur Izura, U., Fatimah, S. (2011). Skyline queries over incomplete multidimensional database. *In*: Proc. of the International Conference on Computing and Informatics (ICOCI2001), p.201-206.

[3] Akrivi, V., Christos, D., Kjetil, N., Michalis, V. (2008). On efficient top-k query processing in highly distributed environments. In: Proc. of the International Conference on Management of Data (ICMD08), p. 753-764. ACM SIGMOD.

[4] Chee-Yong, C., Jagadish, H.V., Kian-Lee, T., Anthony, K.H.T., Zhenjie, Z. (2006). On high dimensional skylines. *In*: Proc. Of the 10th International Conference on Extending Database Technology, p. 478- 495.

[5] Chee-Yong, C., Jagadish, H.V., Kian-Lee, T., Anthony, K.H.T., Zhenjie, Z. (2006). Finding k-dominant skylines in high dimensional space. In: Proc. of the International Conference on Management of Data (ICMD06), p. 503- 514. ACM SIGMOD.

[6] Dana, A., Bouchra, S., Erick, L., Florence, S. (2008). LA-GPS: A location-aware geographical pervasive system. *In*: Proc. of the 24th International Conference on Data Engineering Works (ICDEW08), p. 160-163.

[7] Dimitris, P., Yufei, T., Greg, F., Bernhard, S. (2003). An optimal and progressive algorithm for skyline queries. *In*: Proc. of the International Conference on Management of Data (ICMD), p. 467- 478.

[8] Donald, K., Frank, R., Steffen, R. (2002). Shooting stars in the sky: an online algorithm for skyline queries. *In*: Proc. of the 28th International Conference on Very Large Data Bases (VLDB 28), p. 275- 286.

[9] Ihab, F.I., Walid, G. A., Ahmed, K.E. (2003). Supporting top-*k* join queries in relational databases. *In*: Proc. of the 29th International Conference on Very Large Data Bases (VLDB 29), p. 754-765.