Top-k Skyline: A Unified Approach

Marlene Goncalves and María-Esther Vidal

Universidad Simón Bolívar,
Departamento de Computación,
Caracas, Venezuela
{mgoncalves, mvidal}@usb.ve

Abstract. The WWW has become a huge repository of information. For almost any knowledge domain there may exist thousands of available sources and billions of data instances. Many of these sources may publish irrelevant data. User-preference approaches have been defined to retrieve relevant data based on similarity, relevance or preference criteria specified by the user. Although many declarative languages can express user-preferences, considering this information during query optimization and evaluation remains as open problem. SQLf, Top-k and Skyline are three extensions of SQL to specify user-preferences. The first two filter irrelevant answers following a score-based paradigm. On the other hand, the latter produces relevant non-dominated answers using an order-based paradigm. The main objective of our work is to propose a unified approach that combines paradigms based on order and score. We propose physical operators for SQLf considering Skyline and Top-k features. Properties of those will be considered during query optimization and evaluation. We describe a Hybrid-Naive operator for producing only answers in the Pareto Curve with best score values. We have conducted initial experimental studies to compare the Hybrid operator, Skyline and SQLf.

1 Introduction

The WWW has motivated the definition of new techniques to access information. Currently, there are around three billion of static documents in the WWW. Some of these documents may publish irrelevant data and users have to be aware to discard the useless information based on their preferences. To express user preference queries many declarative languages have been defined. Those languages can be grouped in two paradigms: score-based and order-based. Score-based languages order the top k answers in terms of a score function that induces a total order. The challenge is to identify the top k objects in this totally ordered set, without having to scan all the objects. On the other hand, order-based languages rank answers using multicriteria selections. Multicriteria induce a partially ordered set or strata; in consequence there is no single optimal answer. Thus the main problem is to construct the first stratum or skyline.

Many algorithms have been proposed to evaluate either score-based or order-based languages, however, some problems still remain open. First, user-preferences may be expressed as combinations of top k and multicriteria selections. To process those queries, a physical operator should identify the top k answers among the objects in the strata that fulfil the user-preferences, i.e., the proposed operator should unify the functionalities of the score-based and order-based approaches. Second, user preference algorithms need to be integrated into real world query optimizers. Finally, user preferences should be able to evaluate queries against Web data sources. In this paper we propose the definition and implementation of physical operators to achieve these goals.

Given a multicriteria top k query and a set of objects O, we are interested in defining algorithms to efficiently identify the k objects in strata of O. Strata R is a sequence of subsets $<R_1,...,R_n>$, such that R_i is a stratum, $R_i \subseteq O$ and $\bigcup_{i=1}^{n} R_i = O$, i.e., R is a partition of O. Points in a stratum R_i are non-dominated.

Example 1:
Consider a tourist interested in the top 5 cheap restaurants that are close to his hotel. The tourist can formulate the following top-5 query[1]:

*Select **
From Guide
Skyline of cheap(Price) max, close(Address, HotelAd) max
Order By max(quality(Food))
Stop After 5,

where *cheap* is a user-defined score function that ranks restaurants in terms of their prices. Similarly, *close* is a user-defined function that scores restaurants depending on the distance between a restaurant and the hotel. Finally, *quality* is a user-defined function that measures the quality of the food. Values close to 1 mean that the restaurant is cheap or close or high quality. Note that all the functions are user-dependent. To answer this query, first a query engine should construct the strata of the table *Guide* induced by the multicriteria "cheap(Price) max, close(Address,HotelAd) max". Second, it should construct the first "s" strata $R_1,\ldots,R_s$, such $\left|\bigcup_{i=1}^{s} R_i\right| \geq 5 \geq \left|\bigcup_{i=1}^{s-1} R_i\right|$, i.e., the minimum number of strata $R_1,\ldots,R_s$, where the cardinality of their union is greater or equal than 5. Finally, the top 5 answers will be selected from those strata $R_1,\ldots,R_s$. Then, the 5 tuples that maximize the user-defined quality functions will be in the answer. Note that in case of ties, a new score function will be needed to break them.

To the best of our knowledge none of the existing languages express and efficiently evaluate this type of queries. Thus, the main objective of our proposed work, is the definition and integration in a real DBMS (Data Base Management System) of two hybrid operators, a Top-k Skyline Select and a Top-k Skyline Join, that in conjunction with the relational algebra operators will allow users to express and evaluate queries such as Example 1.

In this paper we formalize the problem and present our initial results. The paper comprises 5 sections. In Section 2 we define the problem and provide a naive solution. In Section 3, we briefly describe the existing approaches. In Section 4 we report our initial experimental results. Finally, in Section 0, the concluding remarks and future work are pointed out.

[1] The query is expressed using a combination of the languages defined in [9][18].

2 Motivating Example

Suppose that a research company has two vacancies and has received applications from 5 candidates. Candidates are described by their names, degrees, publications, years of professional experience, grade point averages, and their main research areas. Consider the following relational table that represents this candidate information:

Candidate(Name,Degree,Publications,Experience, GPA, Area)

Additionally, consider the following instances of this relational table with the information of the 5 candidates:

Table 1. Candidates for two vacancies of a research company

Name	Degree	Publications	Experience	GPA	Area
Joseph Lieberman	Post Doctorate	9	2	3.75	Databases
Steve Studer	Post Doctorate	10	1	4	Systems
Margaret Stoffel	PhD.	12	2	3.75	Computer Graphics
Ann Grant	MsC.	13	4	3.6	Networks
Joe Grys	Engineer	6	3	3.25	Databases

According to the company policy, a criterion is not more important than any other, and all of them are equally relevant, hence either a weight or a score function cannot be assigned. A candidate can be chosen for the job if and only if there is no other candidate with a higher degree, number of publications, and years of experience. To nominate a candidate, one must identify the set of all the candidates that are not dominated by any other candidate in terms of these criteria. Thus, tuples in table Candidate must be selected in terms of the values: Degree, Publications, and Experience. For example, Anna Grant dominates Joe Grys because he has worse values in the Degree, Publications and Experience attributes. Thus, the nominates are as follows:

Table 2. Nominate Candidates for two vacancies of a research company

Name	Degree	Publications	Experience	GPA	Area
Joseph Lieberman	Post Doctorate	9	2	3.75	Databases
Steve Studer	Post Doctorate	10	1	4	Systems
Margaret Stoffel	PhD.	12	2	3.75	Computer Graphics
Ann Grant	MsC.	13	4	3.6	Networks

Since the company only has two vacancies, it must apply another criteria to select the two new staff members and discard the other two. Staff members will be selected among nominates in terms of the top two values of one or more overall preference functions that combine values of either the first criteria or the other attributes.

First, considering the maximum GAP as a new preference function, three candidates are the new nominates: Steve Studer, Joseph Lieberman, and Margaret Stoffel. Then, taking into account the degree, the tie between Joseph Lieberman and Margaret Stoffel can be broken, and the selected members will be: Steve Studer and Joseph Lieberman.

Intuitively, to select the staff members, queries based on user preferences have been posted against the table Candidates. There are several databases languages to express preference queries. Skyline, Top-k and SQLf are three user preference languages that could be used to identify some of the staff members. However, none of them will provide the complete set, and post-processing will be needed to identify all the members.

Skyline offers a set of operators to build an approximation of a Pareto curve (strata) or set of points that are not dominated by any other point in the dataset (skyline or first stratum). Thus, by using Skyline, one could just obtain the nominated candidates.

On the hand, SQLf will allow referees to implement a score function and filter some of the winners in terms of the combined function. In order to choose staff members, SQLf computes the score for each tuple without checking dominance relationship between tuples in the dataset. Finally, also Top-k query approaches rank a set of tuples according to some provided functions and do not check dominance relationships. However, it is not possible to define such score function, because all criteria are equally important. Thus, the problem of selecting the staff members corresponds to the problem of identifying the top k elements in partially ordered set.

On one hand, Skyline constructs partially ordered sets induced by the equally important criteria. On the other hand, TopK or SQLf select the best k elements in terms of a score function that induce a totally ordered set. In consequence, to identify the members, a hybrid approach that combines the benefits of Skyline, and SQLf or Top-k is required. Thus, tuples in the answer will be chosen among the stratum induced by a multiple criteria and then, ties will be broken using user-defined functions that eventually induce a total order.

2.1 Research Problem

Our main objective is the definition and integration in a real DBMS of two hybrid operators, a Top-k Skyline Select and a Top-k Skyline Join. We plan to extend a traditional relational cost model with statistics about these operators and make them accessible by a query optimizer.

Given a set $O=\{o_1,\ldots,o_m\}$ of m database objects, where each object o_i is characterized by *p* attributes $(A_1,\ldots,A_p)$; n score-functions $s_1,\ldots,s_n$ defined over some of those attributes, with $s_i : O \rightarrow [0,1]$; a combined score function f defined over subsets of $\{s_1,\ldots,s_n\}$ that induces a total order of the objects in O; and n sorted lists $S_1,\ldots,S_n$ containing all database objects in descending order by score-function s_i respectively, we define *Pareto Points* through recurrence in Definition 1.

Definition 1a (Base Case – First *Pareto Point*):

$$P_1 = \left\{ o_i \in O \middle/ \neg\exists o_j \in O : \begin{pmatrix} s_1(o_i) \le s_1(o_j) \wedge \ldots \wedge s_r(o_i) \le s_r(o_j) \wedge r \le n \\ \wedge \exists q \in [1,\ldots,r] : s_q(o_i) < s_q(o_j) \end{pmatrix} \right\}$$

Definition 1b (Inductive Case: *Pareto Point*):

$$P_i = \left\{ o_l \in O \middle/ \begin{array}{l} o_l \notin P_{i-1} \wedge \neg\exists\ o_t \in \left(O - \bigcup_{j=1}^{i-1} P_j\right): \\ \left(\begin{array}{l} s_1(o_l) \leq s_1(o_t) \wedge \ldots \wedge s_r(o_l) \leq s_r(o_t) \wedge r \leq n \\ \wedge\ \exists\ q \in [1, \ldots, r]: s_q(o_l) < s_q(o_t) \end{array} \right) \end{array} \right\}$$

We define *Stratum Top-k* in Definition 2 as the minimum number of strata or points of a Pareto Curve that contain the top k answers based on a combined score function f.

Definition 2 (*Stratum Top-k*):

$$StratumTop-k = \left\{ \bigcup_{i=1}^{j} P_i \middle/ \left| \bigcup_{i=1}^{j} P_i \right| \geq k > \left| \bigcup_{i=1}^{j-1} P_i \right| \right\}$$

Finally, the conditions to be satisfied by the answers of a top-k skyline query are given in Definition 3.

Definition 3 (*The Top-k Skyline Problem*):

$$Top-k = \left\{ o_i \in O \middle/ \begin{array}{l} o_i \in StratumTop-k \wedge \\ \neg\exists^k o_j \in StratumTop-k: \\ \left(f(o_j) > f(o_i)\right) \end{array} \right\}$$

We have extended the Basic Distributed Skyline Algorithm introduced in [6] in order to construct a set of objects that satisfy the conditions in Definition 3. It is presented in Algorithm 1. Our algorithm first builds all the strata R following Definition 1 and 2, and then, it breaks ties by using function f. Elements are considered with respect to the order induced by R, i.e., the top k answers correspond to the best K objects in the topological sort of R. Topological sorting is done considering the combined score function f.

Algorithm 1. (The naive Top-k Skyline Problem)

```
1. Initialize $P_1:=\phi$, n lists $K_1,\ldots,K_n:=\phi$, and $p_1,\ldots,p_n:=\phi$.
2. Initialize counters i:=1, nroStrata:=1, s:=1;
   2.1. Get the next object $o_{new}$ by sorted access on list S.
   2.2. If $o_{new} \in P_1$, update its record's i-th real value with $s_i(o_{new})$, else create such a record in $P_1$.
   2.3. Append $o_{new}$ with $s_i(o_{new})$ to list $K_i$.
   2.4. Set $p_i:=s_i(o_{new})$ and i:=(i mod n) + 1
   2.5. If all scores $s_i(o_{new})$ $(1\leq j\leq n)$ are known, proceed with step 3 else with step 2.1.
3. For i=1 to n do
   3.1. While $p_i=s_i(o_{new})$ do sorted on list $S_i$ and handle the retrieved objects like in step 2.2. to 2.3
```

```
4. If more than one object is entirely known, compare
pairwise and remove the dominated objects from $P_1$.
   4.1. For j=1 to nroStrata do
      4.1.1 If there are dominated objects in $P_j$,
initialize $P_{j+1}$:=$\phi$, add dominated objects to $P_{j+1}$ and
nroStrata:= nroStrata+1.
5. For i=1 to n do
   5.1. Do all necessary random access for the objects
in $K_i$ that are also in all of initialized stratum $P_i$ and
discard objects that are not in $P_i$.
   5.2. Take the objects of $K_i$ and compare them pairwise
to the objects in $K_i$. If an object is dominated by
another object remove it from $K_i$ and $P_i$. Add the
dominated object to stratum $P_{i+1}$.
6. Calculate and order all non-dominated objects by the
combined function f in stratum $P_s$.
7. Output the first K non-dominated objects.
8. While there are not K non-dominated objects,
increase s by 1; repeat step 6.
```

3 Related Work

There exist two paradigms for expressing user preferences: score-based and order-based. Score-based languages rank the top k answers in terms of a score function that induces a total order. The challenge is to identify the top k objects in this totally ordered set, without having to scan all the objects. On the other hand, order-based languages rank answers using multicriteria selections. Multicriteria induce a partially ordered set stratified into subsets of non-dominated objects.

3.1 Order-Based Paradigm

During the 70's and the 80's, people have already studied and proposed user-preference query languages. DEDUCE [17] offers a declarative query language for relational databases including preferences. In [37] DRC (Domain Relational Calculus) is extended with Boolean preference mechanisms and score functions cannot be easily expressed.

Chomicky [20][21] introduced a general logic framework to formalize preference formulas and a preference relational operator called *winnow*. This new operator is integrated into the relational algebra. This approach is more expressive than DRC and implements a combined mechanism between operators. However, the operator *winnow* is not simple for writing and composing preferences. Moreover it is not clear how to implement the operator in a relational system [22].

Preference SQL [33] is an algebraic approach that extends SQL by means of preference operators. Kiessling and Köstler [34] proposed how to extend SQL and XPATH with Preference SQL operators and introduced various types of queries that can be composed with these operators. This language is implemented on the top of a SQL engine. The problem of defining physical operators is not considered.

Skyline operator is introduced in [9] as another SQL extension. This operator expresses preference queries and can be combined with traditional relational operators. Kung et al. defined the first Skyline algorithm in [36], referred to as the maximum vector problem [8][40] and it is based on the divide & conquer principle. Skyline was formalized using a partial order semantic and there are efficient algorithms for relational databases [9][23][25][26][29][35]. Although, the problem of computing the first stratum or skyline is solved, all these algorithms have high time complexity.

3.2 Score-Based Paradigm

Agraval et. al explored how to combine numeric and categorical attributes in [2] and, Agraval and Wimmers [1] introduced a framework for combining preferences. In [41], the preferences are expressions stored as data and evaluated during query execution.

[31] showed how to evaluate preference queries using materialized views that must be defined off-line. This approach does not scale if users define queries and score functions dynamically.

In [4][5][27][28] algorithms are introduced to answer ranking queries. These algorithms assume that inputs are sorted and do not support sources that provide a random access of the data, although these are on common the WWW. Some algorithms for evaluating top-k queries over relational databases are proposed in [3][15]. In [32] a top-k query optimization framework that fully integrates rank-join operators into relational engines is introduced. The algorithms Upper [16], MPro [18] and Framework NC [19] do support random access but are not able to check dominance relationships.

On the other hand, Fagin et al. studied those utility functions that are better with respect to minimization of discrepancy between partial order and weak orders of preferences [28]. In [17], membership functions are defined for measuring tuple adherence to preference conditions. Motro uses functions that measure distance between tuples to measure adherence to goals expressed in the query [38].

Bosc and Pivert integrate the fuzzy set theory with relational algebra. Fuzzy conditions indicate membership grade to the preferences [11]. Later, some query processing mechanisms were proposed [10][12][13][14]; the most relevant is the Derivation Principle due to its lower evaluation cost.

Finally, works that propose to integrate these two paradigms are [7][30]. However, they does not consider the identification of the K best objects in the strata.

4 Initial Results

So far, we have explored the integration of SQLf and Skyline approaches to implement multicriteria top k queries. In this hybrid approach, answers are filtered by means of SQLf, and then a Skyline algorithm is executed. The initial filtering reduces Skyline algorithm complexity time because the Skyline algorithm only has to discard solutions that are not better across all the criteria and that are produced by SQLf. Also, we limit this study only to identify the best objects over the first stratum or skyline.

Our initial experimental study was performed on Oracle 8i. The study consisted of experiments running over one relational table with 100,000 and 1,000,000 tuples. The table has 10 integer columns and one string column. Values of integer columns vary from 1 to 30, where 30 corresponds to the best value. A column may have duplicated values. Duplications are uniformed distributed. The columns are pair-wise statistically independent. We performed 30 randomly generated multicriteria queries. Multicriteria varied between 1, 5 or 9 selections.

Skyline, SQLf and Hybrid were written in PL/SQL and Swi Prolog. Skyline was implemented as the basic SFS algorithm without optimizations [24]. SQLf was evaluated using the Derivation Principle algorithm [12] and implemented on the top of the SQL query engine, and the Hybrid operator was a combination of these two previous algorithms.

The experiments were executed on an Intel 866-MHz PC with 512-MB main memory and an 18-GB disk running Red Hat Linux 8.0.

We report on quality of the answers and query processing time. First, we can observe that the Skyline can return irrelevant objects, while SQLf may either produce irrelevant objects or miss relevant ones. An object is considered irrelevant if it does not belong to the top k skyline objects identified by the Hybrid operator. Between 10% and 30% of the Skyline returned objects were irrelevant. On the other hand, more than 90% of the SQLf objects were irrelevant and less than 10% of the relevant objects were not produced. These initial results motivate the definition of a unified approach that does not produce irrelevant objects or miss relevant ones.

Second, we report on the time taken by Skyline, SQLf and the Hybrid operator to compute the answer. We can observe that the Skyline algorithm time was 3 orders of magnitude greater than the SQLf time, and the Hybrid algorithm time was 2 orders of magnitude less than the Skyline time. These differences may occur because the Skyline algorithm scans the whole data set, while the Hybrid operator scans only the subset of objects produced by SQLf. It has been shown that the best algorithm to compute the full skyline has a (worst-case) complexity of $O(n(\log n)^{d-2})$, where n is the number of points in the data set and d is the number of dimensions [36]. In contrast, the Derivation Principle used to implement the SQLf algorithm just reads a subset of the whole data. The running time of this algorithm depends on the database access time and the score function processing time. The (worst-case) complexity is O(m) where m is the number of points in the answer [12]. Finally, the Hybrid algorithm has a (worst-case) complexity of $O(m(\log m)^{d-2})$. So this task can be very expensive and it is very important to define efficient physical operators.

5 Conclusions and Future Work

In this paper we have described the limitations of the existing approaches to express and evaluate top k multicriteria queries. We have defined our problem and presented a naive solution. We have implemented a first approximation of a unified algorithm as a combination of SQLf and Skyline. To study the performance and the quality of a naive hybrid algorithm, we have conducted an initial experimental study. Our initial results show the quality of the answers identified by the Hybrid operator and the necessity of defining physical operators to efficiently evaluate top k multicriteria queries.

In the future, we plan to define physical Top-k Skyline operators and integrate them into a relational DBMS. Finally, we will extend these operators to access Web data sources.

References

[1] Agrawal, R., and Wimmers, E. L. A framework for expressing and combining preferences. In Proc. of SIGMOD (May 2000), pp. 297-306.

[2] Agrawal, R., Chaudhuri, S., Das, G., and Gionis, A. Automated ranking of database query results. In Proc. of CIDR (Jan 2003).

[3] Aref, W. G., Elmagarmid, A. K., Ilyas, I. F. Supporting Top-k join queries in relational databases. In VLDB Journal (Sep 2004), pp. 207-221.

[4] Balke, W-T., Güntzer, U., and Kiebling, W. Optimizing multi-feature queries for image databases. In VLDB, (Sep 2000), pp. 10-14.

[5] Balke, W-T., Güntzer, U., and Kiebling, W. Towards efficient multi-feature queries in heterogeneous environments. In ITCC (2001), pp. 622-628

[6] Balke, W-T., Güntzer, U., and Xin, J. Efficient Distributed Skylining for Web Information Systems. In EDBT (2004), pp. 256-273.

[7] Balke, W-T. and Güntzer, U. Multi-objetive Query Processing for Database Systems. In Proceedings of VLDB (Sep 2004), pp. 936-947.

[8] Bentley, J. L., Kung, H. T., Schkolnick, M., and Thompson, C. D. On the average number of maxima in a set of vectors and applications. JACM, 25, 4 (1978), pp. 536-543.

[9] Börzönyi, S., Kossman, D., and Stocker, K. The skyline operator. In Proc. of ICDE (Apr. 2001), pp. 421-430.

[10] Bosc, P., and Brisson, A. On the evaluation of some SQLf nested queries. Proceeding International Workshop on Fuzzy Databases and Information Retrieval, 1995.

[11] Bosc, P., and Pivert, O. SQLf: A Relational Database Language for Fuzzy Querying. IEEE Transactions on Fuzzy Systems 3, 1 (Feb 1995).

[12] Bosc, P., and Pivert, O. On the efficiency of the alpha-cut distribution method to evaluate simple fuzzy relational queries. Advances in Fuzzy Systems-Applications and Theory, (1995), 251-260.

[13] Bosc, P., and Pivert, O. SQLf Query Functionality on Top of a Regular Relational Database Management System. Knowledge Management in Fuzzy Databases (2000), 171-190.

[14] Bosc, P., Pivert, O., and Farquhar, K. Integrating Fuzzy Queries into an Existing Database Management System: An Example. International Journal of Intelligent Systems 9, (1994), 475-492.

[15] Bruno, N., Chaudhuri, SL., and Gravano, L Top-k selection queries over relational databases: Mapping strategies and performance evaluation. In TODS 27,2 (2002), pp 153-187.

[16] Bruno, N., Gravano, L, and Marian, A. Evaluating top-k queries over web-accessible databases. In ICDE (2002)

[17] Chang, C. L. Deduce : A deductive query language for relational databases. In Pattern Rec. and Art. Int., C. H. Chen, Ed. Academic Press, New York, 1976, pp. 108-134.

[18] Chang, K. and Hwang, S-W. “Minimal Probing: Supporting Expensive Predicates for Top-k Queries”. Proceedings of the ACM SIGMOD Conference, (Jun 2002).

[19] Chang, K. and Hwang, S-W. "Optimizing access cost for top-k queries over Web sources: A unified cost-based approach". Technical Report UIUCDS-R-2003-2324, University of Illinois at Urbana-Champaign, (Mar 2004).
[20] Chomicky, J. Querying with intrinsic preferences. In Proc. of EDBT (2002), Springer (LNCS 2287), pp. 34-51.
[21] Chomicky, J. Preference formulas in relational queries. ACM TODS 28, 4 (Dec. 2003), 427-466.
[22] Chomicky, J. Semantic optimization of preference queries. In 1st Int. Sym. On Appl. Of Constraint Databases (2004), Springer (LNCS 3074).
[23] Chomicky, J. Godfrey, P., Gryz, J., and Liang, D. Skyline with presorting. In Proc. of ICDE (Mar 2003)., pp. 717-719.
[24] Chomicky, J. Godfrey, P., Gryz, J., and Liang, D. On skyline Computation. (Jun. 2002)
[25] Eng, P. K,. Ooi , B. C., and Tan, K. L. Efficient progressive skyline computation. Proc. Of 27th VLDB (2001), 301-310.
[26] Eng, P. K., Ooi, B. C., and Tan, K. L. Indexing for progressive skyline computation. Data and Knowl. Eng. 46, 2 (2003), pp. 169-201.
[27] Fagin, R. Combining fuzzy information from multiple systems. Journal of Computer and System Sciences (JCSS), 58, 1 (Feb 1996), pp. 216-226.
[28] Fagin, R. Lotem, A., and Naor, M. Optimal aggregation algorithms for middleware. In PODS, Santa Barbara, California (May 2001), pp. 102-113.
[29] Godfrey, P. Skyline cardinality for relational processing. In Proc. of FolKS (Feb 2004), Springer, pp. 78-97.
[30] Goncalves, M and Vidal, M.E. "Preferred Skyline: A hybrid approach between SQLf and Skyline". In Proceedings of DEXA (Ago 2005).
[31] Hristidis, V., Koudas, N., and Papakonstantinou, Y. PREFER: A system for the efficient execution of multi-parametric ranked queries. In Proc. of SIGMOD (May 2001), pp. 259-270.
[32] Ilyas, I.F., Shah, R., Aref, W.G., Vitter, J.S. and Elmagarmid, A.K. Rank-aware Query Optimization. In Proceedings of the 2004 ACM SIGMOD Conf. on Mgmt. of Data, pp. 203 - 214, Paris, France, June 2004.
[33] Kiessling, W. Foundations of preferences in database systems. In Proc. of VLDB (Aug 2002), pp. 311-322.
[34] Kiessling, W., and Köstler, G. Preference SQL: Design, implementation, experiences. In Proc. of VLDB (Aug 2002), pp. 900-1001.
[35] Kossman, D., Ransak, F., and Rost, S. Shooting stars in the sky: An online algorithm for skyline queries. In Proc. of VLDB (Aug 2002), pp. 275-286.
[36] Kung, H. T., Luccio, F., Preparata, F. P. On finding the maxima of a set of vectors. JACM 22, 4 (1975), pp. 469-476.
[37] Lacroix, M., and Lavency, P. Preferences: Putting more knowledge into queries. In Proc. of VLDB (Sept. 1987), pp 217-225.
[38] Motro, A. Supporting goal queries in relational databases. In Proc. of the 1st Int. Conf. on Expert Database Sys. (Apr 1986), pp. 85-96.
[39] Papadimitriou, C. H., and Yannakakis, M. Multiobjective Query Optimization. Proc. ACM SIGMOD/SIGACT Conf. Princ. Of Database Syst. (PODS), Santa Barbara, CA, USA, May 2001.
[40] Preparata, F. P. and Shamos, M. I. Computational Geometry: An Introduction. Springer-Verlag, 1985.
[41] Yalamanchi, A., Srinivasan, J., and Gawlick, D. Managing expressions as data in relational, database systems. In Proc. of CIDR (Jan 2003).