Incremental Trade-Off Management for Preference Based Queries[1]

BALKE Wolf-Tilo
LOFI Christoph
L3S Research Center
Leibniz University Hannover
Appelstr. 9a
30167 Hannover Germany
{balke, lofi}@l3s.de

GÜNTZER Ulrich
Institute of Computer Science
University of Tübingen
Sand 13
72076 Tübingen, Germany
ulrich.guentzer@informatik.uni-tuebingen.de

Abstract

Preference-based queries often referred to as skyline queries play an important role in cooperative query processing. However, their prohibitive result sizes pose a severe challenge to the paradigm's practical applicability. In this paper we discuss the incremental re-computation of skylines based on additional information elicited from the user. Extending the traditional case of totally ordered domains, we consider preferences in their most general form as strict partial orders of attribute values. After getting an initial skyline set our approach aims at incrementally increasing the system's information about the user's wishes. This additional knowledge then is incorporated into the preference information and constantly reduces skyline sizes. In particular, our approach also allows users to specify trade-offs between different query attributes, thus effectively decreasing the query dimensionality. We provide the required theoretical foundations for modeling preferences and equivalences, show how to compute incremented skylines, and proof the correctness of the algorithm. Moreover, we show that incremented skyline computation can take advantage of locality and database indices and thus the performance of the algorithm can be additionally increased.

Keywords: Personalized Queries, Skylines, Trade-Off Management, Preference Elicitation

1 Introduction

Preference-based queries, usually called skyline queries in database research [9], [4], [19], have become a prime paradigm for cooperative information systems. Their major appeal is the intuitiveness of use in contrast to other query paradigms like e.g. rigid set-based SQL queries, which only too often return an empty result set, or efficient, but hard to use top-k queries, where the success of a query depends on choosing the right scoring or utility functions.

[1] Part of this work was supported by a grant of the German Research Foundation (DFG) within the Emmy Noether Program of Excellence.

Skyline queries offer *user-centered* querying as the user just has to specify the basic attributes to be queried and in return retrieves the Pareto-optimal result set. In this set *all possible* best objects (where 'best' refers to being optimal with respect to any monotonic optimization function) are returned. Hence, a user cannot miss any important answer. However, the intuitiveness of querying comes at a price. Skyline sets are known to grow exponentially in size [8], [14] with the number of query attributes and may reach unreasonable result sets (of about half of the original database size, cf. [3], [7]) already for as little as six independent query predicates. The problem even becomes worse, if instead of totally ordered domains user preferences on arbitrary predicates over attribute-based domains are considered. In database retrieval, preferences are usually understood as partial orders [12], [15], [20] of domain values that allow for incomparability between attributes. This incomparability is reflected in the respective skyline sizes that are generally significantly bigger than in the totally ordered case. On the other hand such attribute-based domains like colors, book titles, or document formats play an important role in practical applications, e.g., digital libraries or e-commerce applications. As a general rule of thumb it can be stated that the more preference information (including its transitive implications) is given by the user with respect to each attribute, the smaller the average skyline set can be expected to be. In addition to prohibitive result set sizes, skyline queries are expensive to compute. Evaluation times in the range of several minutes or even hours over large databases are not unheard of.
One possible solution is based on the idea of refining skyline queries incrementally by taking advantage of user interaction. This approach is promising since it benefits skyline sizes as well as evaluation times. Recently, several approaches have been proposed for user-centered refinement:

- using an interactive, exploratory process steering the progressive computation of skyline objects [17]
- exploiting feedback on a representative sample of the original skyline result [8], [16]
- projecting the complete skyline on subsets of predicates using pre-computed skycubes [20], [23].

The benefit of offering intuitive querying and a cooperative system behavior to the user in all three approaches can be obtained with a minimum of user interaction to guide the further refinement of the skyline. However, when dealing with a massive amount of result tuples, the first approach needs a certain user expertise for steering the progressive computation effectively. The second approach faces the problem of deriving representative samples efficiently, i.e. avoiding a complete skyline computation for each sample. In the third approach the necessary pre-computations are expensive in the face of updates of the database instance.
Moreover, basic theoretical properties of incremented preferences in respect to possible preference collisions and induced query modification and query evaluation have been outlined in [13].

In this paper we will provide the theoretical foundations of modeling partial-ordered preferences and equivalences on attribute domains provide algorithms for incrementally and interactively computing skyline sets and prove the soundness and consistency of the algorithms (and thus giving a comprehensive view of [1], [2], [6]). Seeing preferences in their most general form as partial orders between domain values, this implicitly includes the case of totally ordered domains. After getting an (usually too big) initial skyline set our approach aims at interactively increasing the system's information about the user's wishes. The additional knowledge then is incorporated into the preference information and helps to reduce skyline sets. Our contribution thus is:

- Users are enabled to specify *additional preference information* (in the sense of domination), as well as *equivalences* (in the sense of indifference) between attributes leading to an incremental reduction of the skyline. Here our system will efficiently support the user by automatically taking care that newly specified preferences and equivalences will never violate the consistency of the previously stated preferences.
- Our skyline evaluation algorithm will allow specifying such additional information *within a certain attribute domain*. That means that more preference information about an attribute is elicited from the user. Thus the respective preference will be more complete and skylines will usually become smaller. This can reduce skylines to the (on average considerably smaller) sizes of total order skyline sizes by canceling out incomparability between attribute values.
- In addition, our evaluation algorithm will also allow specifying additional relationships between preferences on *different attributes*. This feature allows defining the qualitative importance or equivalence of attributes in different domains and thus forms a good tool to compare the respective utility or desirability of certain attribute values. The user can thus express trade-offs or compromises he/she is willing to take and also can adjust imbalances between fine-grained and coarse preference specifications.
- We show that the efficiency of incremented skyline computation can be considerably increased by employing *preference diagrams*. We derive an algorithm which takes advantage of the *locality* of incremented skyline set changes depending on the changes made by the user to the preference diagram. By that, the algorithm can operate on a considerable smaller dataset with an increased efficiency.

Spanning preferences across attributes (by specifying trade-offs) is the only way – short of dropping entire query predicates – to reduce the dimensionality of the skyline computation and thus severely reduce skyline sizes. Nevertheless the user stays in full control of the information specified and all information is only added in a qualitative way, and not by unintuitive weightings.

2 A Skyline Query Use-Case and Modeling

Before discussing the basic concepts of skyline processing, let us first take a closer look at a motivating scenario which illustrates our modeling approach with a practical example:

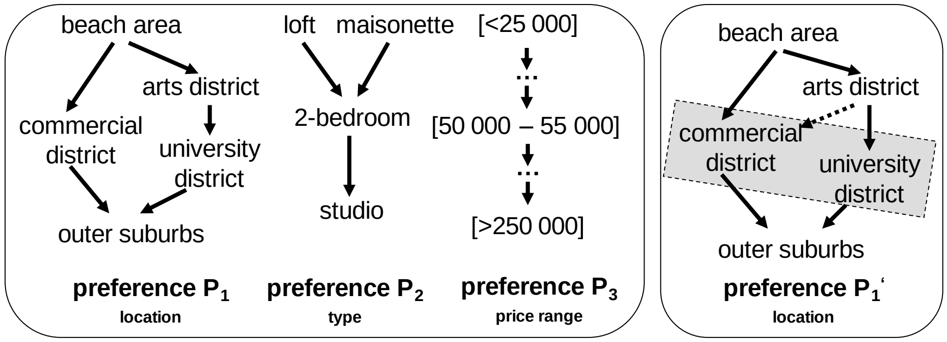

Figure 1. Three typical user preferences (left) and an enhanced preference (right)

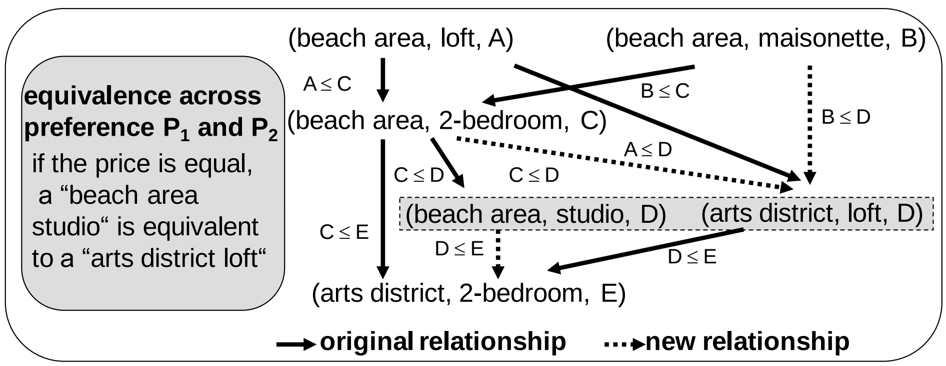

Figure 2. Original and induced preference relationships for trade offs

Example: Anna is currently looking for a new apartment. Naturally, she has some preferences how and where she wants to live. Figure 1 shows *preference diagrams* of Anna's *base preferences* modeled as a *strict partial order* on domain values of three attributes (cf. [15], [12]): location, apartment type and price. These preferences might either be stated explicitly by Anna together with the query or might be derived from Anna's user profile and/or activity history [11]. Some of these preferences may even be common domain knowledge (c.f. [5]) like for instance that in case of two equally desirable objects, the less costly alternative is generally preferred. Based on such preferences, Anna may now retrieve the skyline over a real-estate database. The result is the *Pareto-optimal set* of available apartments consisting of all apartments which are not dominated by others, e.g. a cheap beach area loft immediately dominates all more expensive 2-bedrooms or studios, but can, for instance, not dominate any maisonette. After the first retrieval Anna has to manually review a probably large skyline.

In the first few retrieval steps skylines usually contain a large portion of all database objects due to the incomparability between many objects. But the size of the Pareto-optimal set may be reduced incrementally by providing suitable additional information on top of the stated preferences, which will then result in new domination relationships on the level of the database objects and thus remove less preferred objects from the skyline. Naturally, existing preferences might be extended by adding some new preference relationships. But also explicit equivalences may be stated between certain attributes expressing actual indifference and thus resulting in new domination relationships, too.

Example (cont): Let's assume that the skyline still contains too many apartments. Thus, Anna interactively refines her originally stated preferences. For example, she might state that she actually *prefers* the arts district over the university district and the latter over the commercial district which would turn the preference P_1 into a totally ordered relation. This would for instance allow apartments located in the arts and university district to dominate those located in the commercial district with respect to P_1, resulting in a decrease of the size of the Pareto-optimal set of skyline objects. Alternatively, Anna might state that she actually does not care whether her flat is located in the university district or the commercial district – that these two attributes are *equally desirable* for her. This is illustrated in the right hand side part of Figure 1 as preference P_1'. In this case, it is reasonable to deduce that all arts district apartments will dominate commercial district apartments with respect to the location preference.

Preference relationships over attribute domains lead to domination relationships on database objects, when the skyline operator is applied to a given database. These resulting domination relationships are illustrated by the solid arrows in figure 2. However, users might also weigh some predicates as more important than others and hence might want to model trade-offs they are willing to consider. Our preference modeling approach introduced in [1] allows expressing such trade-offs by providing new preference relations or equivalence relations between different attributes. This means 'amalgamating' some attributes in the skyline query and subsequently reducing the dimensionality of the query.

Example (cont): While refining her preferences, Anna realizes that she actually would consider the area in which her new apartment is located as more important than the actual apartment type – in other words: for her, a relaxation in the apartment type is less severe than a relaxation in the area attribute. Thus she states that she would consider a beach area studio (the least desired apartment type in the best area) still as equally desirable to a loft in the arts district (the best apartment type in a less preferred area – by doing that, she stated a preference on an amalgamation of the attribute apartment type and location). This statement induces new domination relations on database objects (illustrated as the dotted arrows in Figure 2), allowing for example any cheaper beach area 2-bedroom to dominate all equally priced or more expensive arts district lofts (by using the *ceteris paribus* [18] assumption). In this way, the result set size of the skyline query can be decreased.

3 Theoretical Foundation and Formalization

In this section we formalize the semantics of adding incremental preference or equivalence information on top of already existing base preferences or base equivalences. First, we provide basic definitions required to model base and amalgamated preferences and equivalence relationships. Then, we provide basic theorems which allow for consistent incremented skyline computation (cf. [1]). Moreover, we show that it suffices to calculate incremental changes on transitively reduced preference diagrams. We show that local changes in the preference graph only result in locally restricted recomputations for the incremented skyline and thus leads to superior performance (cf. [2]).

3.1 Base Preferences and the Induced Pareto Aggregation

In this section we will provide the basic definitions which are prerequisites for section 3.1 and 3.1 . We will introduce the notion for base preferences, base equivalences, their amalgamated counterparts, a generalized Pareto composition and a generalized skyline.
The basic construct are so-called base preferences defining strict partial orders on attribute domains of database objects (based on [12], [15]):

Definition 1: (Base Preference)
Let D_1, D_2, …, D_m be a non-empty set of m domains (i.e. sets of attribute values) on the attributes $Attr_1$, $Attr_2$, … $Attr_m$ so that D_i is the domain of $Attr_i$. Furthermore let $O \subseteq D_1 \times D_2 \times \ldots \times D_m$ be a set of database objects and let $attr_i: O \rightarrow D_i$ be a function mapping each object in O to a value of the domain D_i.
Then a ***Base Preference*** $\boldsymbol{P_i} \subseteq D_i^2$ is a strict partial order on the domain D_i.

The intended interpretation of $(x, y) \in P_i$ with $x, y \in D_i$ (or alternatively written $x <_{P_i} y$) is the intuitive statement “I like attribute value y (for the domain D_i) better than attribute value x (of the same domain)”. This implies that for $o_1, o_2 \in O$ $(attr_i(o_1), attr_i(o_2)) \in P_i$ means “I like object o_2 better than object o_1 with respect to its i-th attribute value”.
In addition to specifying preferences on a domain D_i we also allow to define equivalences as given in Definition 2.

Definition 2: (Base Equivalence and Compatibility)
Let O a set of database objects and P_i a base preference on D_i as given in Definition 1. Then we define a ***Base Equivalence*** $\boldsymbol{Q_i} \subseteq D_i^2$ as an equivalence relation (i.e. Q_i is reflexive, symmetric and transitive) which is *compatible* with P_i and is defined as:
a) $Q_i \cap P_i = \emptyset$ (meaning no equivalence in Q_i contradicts any strict preference in P_i)
b) $P_i \circ Q_i = Q_i \circ P_i = P_i$ (the domination relationships expressed transitively using P_i and Q_i must always be contained in P_i)
In particular, as Q_i is an equivalence relation, Q_i trivially contains the pairs (x, x) for all $x \in D_i$.

The interpretation of base equivalences is similarly intuitive as for base preferences: $(x, y) \in Q_i$ with $x, y \in D_i$ (or alternatively written $x \sim_{Q_i} y$) means “I am indifferent between attribute values x and y of the domain D_i”.

As mentioned in Definition 2, a given base preference P_i and base equivalence Q_i have to be compatible to each other - this means that on one hand attribute value x can never be considered (transitively) equivalent and being (transitively) preferred to some attribute value y at the same time. On the other hand preference relationships between attribute values should always extend to all equivalent attribute values, too. Please note that generally there still may remain values $x, y \in D_i$ where neither $x <_{P_i} y$ nor $y <_{P_i} x$, nor $x \sim_{Q_i} y$ holds. We call these values *incomparable*.
The base preferences $P_1, \ldots, P_m$ together with the base equivalences $Q_1, \ldots, Q_m$ induce a partial order on the set of database objects according to the notion of Pareto optimality. This partial order is created by the Generalized Pareto Aggregation (cf. [6]) which is given in Definition 3.

Definition 3: (Generalized Pareto Aggregation for Base Preferences and Equivalences)
Let O be a set of database objects, $P_1, \ldots, P_m$ be a set of m base preferences as given in Definition 1 and $Q_1, \ldots, Q_m$ be a set of m compatible base equivalence relations as defined in Definition 2. Then we define the **Generalized Pareto Aggregation** for base preferences and equivalences as:
$Pareto(O, P_1, \ldots, P_m, Q_1, \ldots, Q_m) := \{(o_1, o_2) \in O^2 \mid \forall\ 1 \leq i \leq m: (attr_i(o_1), attr_i(o_2)) \in (P_i \cup Q_i) \wedge \exists\ 1 \leq j \leq m: (attr_j(o_1), attr_j(o_2)) \notin Q_j\}$

As stated before, the generalized Pareto aggregation induces an order on the *actual database objects.* This order can be extended to an *Object Preference* as given by Definition 4.

Definition 4: (Object Preference *P* and Object Equivalence *Q*)
An *Object Preference* $P \subseteq O^2$ is defined as a strict partial order on the set of database objects O containing the generalized Pareto aggregation of the underlying set of base preferences and base equivalences: $P \supseteq Pareto(O, P_1, \ldots, P_m, Q_1, \ldots, Q_m)$.
Furthermore, we define the *Object Equivalence* $Q \subseteq O^2$ as an equivalence relation on O (i.e. Q is reflexive, symmetric and transitive) which is *compatible* with P (cf. Definition 2) and respects $Q_1, \ldots, Q_m$ in the following way:
$\forall\ 1 \leq i \leq m\ (x_i, y_i) \in Q_i \ \Rightarrow\ ((x_1, \ldots, x_m), (y_1, \ldots, y_m)) \in Q$
In particular, Q contains at least the identity tuples (o, o) for each $o \in O$.

An object level preference P, as given by Definition 4, contains at least the order induced by the Pareto aggregation function on the base preferences and equivalences. Additionally, it can be enhanced and extended by other user-provided relationships (and thus leaving the strict Pareto domain). Often, users are willing to perform a *trade-off*, i.e. relax their preferences in one attribute in favor of an object's a better performance in another attribute. For modeling trade-offs, we therefore introduce the notion of *amalgamated preferences* and *amalgamated equivalences*.
Building on the example given in Section 2 , an *amalgamated equivalence* could be a statement like "I am indifferent between an Arts District Studio and an University District Loft" (as illustrated in Figure 3). Thus, this equivalence statement is modeled on an *amalgamation* of domains (location and type) and results in new equivalence and preference relationships on the database instance.

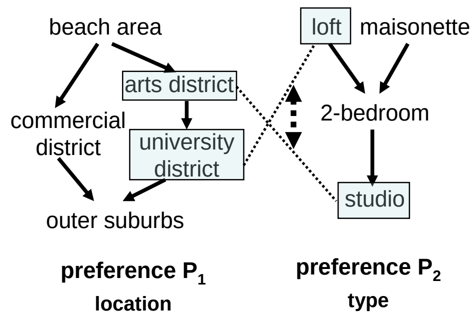

Figure 3. Modeling a trade-off using an Amalgamated Preference

Definition 5: (Amalgamated Preferences Functions)
Let $\mu \subseteq \{1, \ldots, m\}$ be a set with cardinality k. Using π as the projection in the sense of relational algebra we define the function
$AmalPref(x_\mu, y_\mu) : (\times_{i\in\mu} D_i)^2 \rightarrow O^2$
$(x_\mu, y_\mu) \longmapsto \{(o_1, o_2) \in O^2 \mid \forall\, i \in \mu\, : (\pi_{Attr_i}(o_1) = \pi_{Attr_i}(x_\mu) \wedge \pi_{Attr_i}(o_2) = \pi_{Attr_i}(y_\mu))$
$\wedge\, \forall\, i \in \{1, \ldots, m\} \setminus \mu.\, :\, (\pi_{Attr_i}(o_1) = \pi_{Attr_i}(o_2)))\, \}$

This means: Given two tuples x_μ, y_μ from the same amalgamated domains described by μ, the function $AmalPref(x_\mu, y_\mu)$ returns a set of relationships between database objects of the form (o_1, o_2) where the attributes of o_1 projected on the amalgamated domains equal those of x_μ, the attributes of o_2 projected on the amalgamated domains equal those of y_μ and furthermore all other attributes which are not within the amalgamated attributes are identical for o_1 and o_2. The last requirement denotes the well-known *ceteris paribus* [18] condition ("all other things being equal"). The relationships created by that function may be incorporated into P as long as they don't violate P's consistency. The conditions and detailed mechanics allowing this incorporation are the topic of section 3.1 .

Please note the typical cross-shape introduced by trade-offs: A relaxation in one attribute is compared to a relaxation in the second attribute. Though in the Pareto sense the two respective objects are not comparable (in Figure 3 the arts district studio has a better value with respect to location, whereas the university district loft has a better value with respect to type), amalgamation adds respective preference and equivalence relationships between those objects.

Definition 6: (Amalgamated Equivalence Functions)
Let $\mu \subseteq \{1, \ldots, m\}$ be a set with cardinality k. Using π as the projection in the sense of relational algebra we define the function
$AmalEq(x_\mu, y_\mu) : (\times_{i\in\mu} D_i)^2 \rightarrow O^2$
$(x_\mu, y_\mu) \longmapsto \{(o_1, o_2) \in O^2 \mid \forall\, i \in \mu\, : [\, (\pi_{Attr_i}(o_1) = \pi_{Attr_i}(x_\mu) \wedge \pi_{Attr_i}(o_2) = \pi_{Attr_i}(y_\mu)) \vee$
$(\pi_{Attr_i}(o_2) = \pi_{Attr_i}(x_\mu) \wedge \pi_{Attr_i}(o_1) = \pi_{Attr_i}(y_\mu))\,] \wedge \forall\, i \in \{1, \ldots, m\} \setminus \mu.\, :\, (\pi_{Attr_i}(o_1) = \pi_{Attr_i}$
$(o_2)))\, \}$

The function differs from amalgamated preferences in that as it returns symmetric relationships, i.e. if $(o_1, o_2) \in Q$, also (o_2, o_1) has to be in Q. Furthermore, these relationships have to be incorporated into Q instead of P as long as they don't violate consistency. But due to the compatibility characteristic also P can be affected by new relationships in Q.
Based on an *object preference* P we can finally derive the Skyline set (given by Definition 7) which is returned to the user. This set contains *all possible* best objects with respect to the underlying user preferences. This means that the set contains only those objects that are not dominated by any other object in the object level preference P. Note that we call this set *generalized* Skyline as it is derived from P which is initially the Pareto order but may also be extended with additional relationships (e.g. trade-offs). Note that the generalized skyline set is solely derived using P, still it respects Q by introducing new relationships into P based on recent additions to Q (cf. Definition 8).

Definition 7: (Generalized Skyline)
The *Generalized Skyline* $S \subseteq O$ is the set containing all optimal database objects in respect to a given object preference P and is defined as

$$S := \{ o \in O \mid \neg \exists\, o' \in O : (o, o') \in P \}$$

3.1 Incremental Preference and Equivalence Sets

The last section provided the basic definitions required for dealing with preference and equivalence sets. In the following sections we provide a method for *incremental* specification and enhancements of object preference sets P and equivalence sets Q. Also, we show under which conditions the addition of new object preferences / equivalences is safe and how compatibility and soundness can be ensured.
The basic approach for dealing with incremental preference and equivalent sets is illustrated in Figure 4. First, the base preferences P_1 to P_m (Definition 1) and their according base equivalences Q_1 to Q_m (Definition 2) are elicited. Based on these, the initial object preference P (Definition 4) is created by using the generalized Pareto aggregation (Definition 3). The initial object equivalence Q starts as a minimal relation as defined in Definition 4. The generalized Pareto skyline (Definition 7) of P is then displayed to the user and the iterative phase of the process starts. Users now have the opportunity to specify additional base preferences or equivalences or amalgamated relationships (Definition 5, Definition 6) as described in the previous section. The set of new object relationships resulting from the *Ceteris Paribus* functions of the newly stated user information then is checked for compatibility using the constrains given in this section. If the new object relationships are compatible with P and Q they are inserted and thus incremented sets P^* and Q^* are formed. If the relationships were not consistent with the previously stated information, then the last addition is discarded and the user is notified. The user thus can state more and more information until the generalized skyline is lean enough for manual inspection.

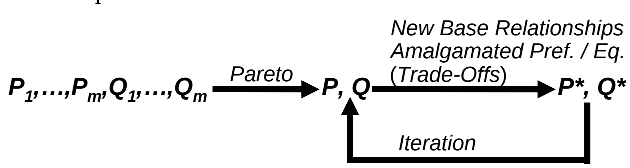

Figure 4. General Approach for Iterated Preference / Equivalence Sets

Definition 8: (Incremented Preference and Equivalence Set)
Let O be a set of database objects, $P \subseteq O^2$ be a strict preference relation, $P^{conv} \subseteq O^2$ be the set of converse preferences with respect to P, and $Q \subseteq O^2$ be an equivalence relation that is compatible with P. Let further $S \subseteq O^2$ be a set of object pairs (called incremental preferences) such that

$\forall\, x, y \in O$: $(x, y) \in S \Rightarrow (y, x) \notin S$ and $S \cap (P \cup P^{conv} \cup Q) = \varnothing$

and let $E \subseteq O^2$ be a set of object pairs (called incremental equivalences) such that

$\forall\, (x, y) \in O$: $(x, y) \in E \Rightarrow (y, x) \in E$ and $E \cap (P \cup P^{conv} \cup Q \cup S) = \varnothing$.

Then we will define T as the transitive closure $T := (P \cup Q \cup S \cup E)^+$ and the incremented preference relation P^* and the incremented equivalence relation Q^* as

$P^* := \{ (x, y) \in T \mid (y, x) \notin T\}$ and
$Q^* := \{ (x, y) \in T \mid (y, x) \in T\}$

The basic intuition is that S and E contain the new preference and equivalence relationships that have been elicited from the user additionally to those given in P and Q. For example, S and E can result from the user specifying a trade-off and, in this case, are induced using the *ceteris paribus* semantics (cf. Definition 5 and Definition 6). The only conditions on S and E are that they can neither directly contradict each other, nor are they allowed to contradict already known information. The sets P^* and Q^* then are the new preference/equivalence sets that incorporate all the information from S and E and that will be used to calculate the new generalized and probably smaller skyline set. Definition 8 indeed results in the desired incremental skyline set as we will prove in Theorem 1:

Theorem 1: (Correct Incremental Skyline Evaluation with *P and *Q**)**
Let P^* and Q^* be defined like in Definition 8. Then the following statements hold:

1) P^* defines a strict partial order (specifically: P^* does not contain cycles)
2) Q^* is a compatible equivalence relation with preference relation P^*
3) $Q \cup E \subseteq Q^*$
4) The following statements are equivalent
   a) $P \cup S \subseteq P^*$
   b) $P^* \cap (P \cup S)^{conv} = \varnothing$ and $Q^* \cap (P \cup S)^{conv} = \varnothing$
   c) No cycle in $(P \cup Q \cup S \cup E)$ contains an element from $(P \cup S)$

   and from either one of these statements follows: $Q^* = (Q \cup E)^+$

Proof:
Let us first show two short lemmas:

Lemma 1: $T \circ P^* \subseteq P^*$
Proof: Due to T's transitivity $T \circ P^* \subseteq T \circ T \subseteq T$ holds. If there would exist objects $x, y, z \in O$ with $(x, y) \in T$, $(y, z) \in P^*$, but $(x, z) \notin P^*$, then follows $(x, z) \in Q^*$ because T is transitive and the disjoint union of P^* and Q^*. Due to Q^*'s symmetry we also get $(z, x) \in Q^*$ and thus $(z, y) = (z, x) \circ (x, y) \in T \circ T \subseteq T$. Hence we have $(y, z), (z, y) \in T \Rightarrow (y, z) \in Q^*$ in contradiction to $(y, z) \in P^*$. ■

Lemma 2: $P^* \circ T \subseteq P^*$

Proof: analogous to Lemma 1 ■

ad 1) From Lemma 1 directly follows $P^* \circ P^* \subseteq P^*$ and thus P^* is transitive. Since by Definition 8 P^* is also anti-symmetric and irreflexive, P^* defines a strict partial order. ■

ad 2) We have to show the three conditions for compatibility:

a) Q^* is an equivalence relation. This can be shown as follows: Q^* is symmetric by definition, is transitive because T is transitive, and is reflexive because $Q \subseteq T$ and trivially all pairs $(q, q) \in Q$.

b) $Q^* \cap P^* = \emptyset$ is true by Definition 8

c) From Lemma 1 we get $Q^* \circ P^* \subseteq P^*$ and due to Q^* being reflexive also $P^* \subseteq Q^* \circ P^*$. Thus $P^* = Q^* \circ P^*$. Analogously we get $P^* \circ Q^* = P^*$ from Lemma 2.

Since a), b) and c) hold, equivalence relation Q^* is compatible to P^*. ■

ad 3) Since $Q \subseteq T$ and Q is symmetric, $Q \subseteq Q^*$. Analogously $E \subseteq T$ and E is symmetric, $E \subseteq Q^*$. Thus, $Q \cup E \subseteq Q^*$. ■

ad 4) We have to show three implications for the equivalence of a), b) and c):

a) $\Rightarrow$ c): Assume there would exist a cycle $(x_0, x_1) \circ \ldots \circ (x_{n-1}, x_n)$ with $x_0 = x_n$ and edges from $(P \cup Q \cup S \cup E)$ where at least one edge is from $P \cup S$, further assume without loss of generality $(x_0, x_1) \in P \cup S$. We know $(x_2, x_n) \in T$ and $(x_1, x_0) \in T$, therefore $(x_0, x_1) \in Q^*$ and $(x_0, x_1) \notin P^*$. Thus, the statement $P \cup S \subseteq P^*$ cannot hold in contradiction to a).

c) $\Rightarrow$ b): We have to show $T \cap (P \cup S)^{conv} = \emptyset$. Assume there would exist $(x_0, x_1) \circ \ldots \circ (x_{n-1}, x_n) \in (P \cup S)^{conv}$ with $(x_{i-1}, x_i) \in (P \cup Q \cup S \cup E)$ for $1 \leq i \leq n$. Because of $(x_0, x_n) \in (P \cup S)^{conv}$ follows $(x_n, x_0) \in P \cup S$ and thus $(x_0, x_1) \circ \ldots \circ (x_{n-1}, x_n)$ would have been a cycle in $(P \cup Q \cup S \cup E)$ with at least one edge from P or S, which is a contradiction to c).

b) $\Rightarrow$ a): If the statement $P \cup S \subseteq P^*$ would not hold, there would be x and y with $(x, y) \in P \cup S$, but $(x, y) \notin P^*$. Since $(x, y) \in T$, it would follow $(x, y) \in Q^*$. But then also $(y, x) \in Q^* \cap (P \cup S)^{conv}$ would hold, which is a contradiction to b).

This completes the equivalence of the three conditions now we have to show that from any of we can deduce $Q^* = (Q \cup E)^+$. Let us assume condition c) holds.

First we show $Q^* \subseteq (Q \cup E)^+$. Let $(x, y) \in Q^*$, then also $(y, x) \in Q^*$. Thus we have two representations $(x, y) = (x_0, x_1) \circ \ldots \circ (x_{n-1}, x_n)$ and $(y, x) = (y_0, y_1) \circ \ldots \circ (y_{m-1}, y_m)$, where all edge are in $(P \cup Q \cup S \cup E)$ and $x_n = y = y_0$ and $x_0 = x = y_m$. If both representations are concatenated, a cycle is formed with edges from $(P \cup Q \cup S \cup E)$. Using condition c) we know that none of these edges can be in $P \cup S$. Thus, $(x, y) \in (Q \cup E)^+$.

The inclusion $Q^* \supseteq (Q \cup E)^+$ holds trivially due to $(Q \cup E)^+ \subseteq T$ and $(Q \cup E)^+$ is symmetric, since both Q and E are symmetric. ■

The evaluation of skylines thus comes down to calculating P^* and Q^* as given by Definition 8 after we have checked their consistency as described in Theorem 1, i.e. verified that no inconsistent information has been added. It is a nice advantage of our system that at any point we can incrementally check the applicability and then accept or reject a statement elicited from the user or a different source like e.g. profile information. Therefore, skyline computation and preference elicitation are interleaved in a transparent process.

3.1 Efficient Incremental Skyline Computation

In the last sections, we provided the basic theoretical foundations for incremented preference and equivalence sets. In addition, we showed how to use the generalized Pareto aggregation for incremented Skyline computation based on the previous skyline objects. In this section, we will improve this algorithm's efficiency by exploiting the local nature of incrementally added preference / equivalence information.

While in the last section we facilitated skyline computation by modeling the full object preference P and object equivalence E, we will now enhance the algorithm to be based on transitively reduced (and thus considerably smaller) *Preference Diagrams*. Preferences diagrams are based on the concept of Hasse diagrams but, in contrast, do not require a full intransitive reduction (e.g. some transitive information may remain in the diagram). Basically, a preference diagram is a simple graph representation of attribute values and preference edges as following:

Definition 9: (Preference Diagrams)
Let P be a preference in the form of a finite strict partial order. A *preference diagram* $PD(P)$ for preference P denotes a (not necessarily minimal) graph such that the transitive closure $PD(P)^+ = P$.

Please note that there may be several preference diagrams provided (or incrementally completed) by the user to express the same preference information (which is given by the transitive closure of the graph). Thus the preference diagram may contain redundant transitive information if it was explicitly stated by the user during the elicitation process. This is particularly useful when the diagram is used for user interface purposes [2].

In the remainder of this section, we want to avoid the handling of the bulky and complex to manage incremented preference P^* and rather only incorporate increments of new preference information as well as new equivalence information into the preference diagram instead. The following two theorems show how to do this.

Theorem 2: (Calculation of *P**)**
Let O be a set of database objects and P, P^{conv}, and Q as in Definition 8 and $E := \{(x, y), (y, x)\}$ new equivalence information such that $(x, y), (y, x) \notin (P \cup P^{conv} \cup Q)$. Then P^* can be calculated as

$$P^* = (P \cup (P \circ E \circ P) \cup (Q \circ E \circ P) \cup (P \circ E \circ Q)).$$

Proof: Assume $(a, b) \in T$ as defined in Definition 8. The edge can be represented by a chain $(a_0, a_1) \circ \ldots \circ (a_{n-1}, a_n)$, where each edge $(a_{i-1}, a_i) \in (P \cup Q \cup E)$ and $a_0 := a$, $a_n := b$. This chain can even be transcribed into a representation with edges from $(P \cup Q \cup E)$, where at most one single edge is from E. This is because, if there would be two (or more) edges from E namely (a_{i-1}, a_i) and (a_{j-1}, a_j) (with $i < j$) then there are four possibilities:

a) both edges are (x, y) or both edges are (y, x), in both of which cases the sequence $(a_i, a_{i+1}) \circ \ldots \circ (a_{j-1}, a_j)$ forms a cycle and can be omitted
b) the first edge is (x, y) and the second edge is (y, x), or vice versa, in both of which cases $(a_{i-1}, a_i) \circ \ldots \circ (a_{j-1}. a_j)$ forms a cycle and can be omitted, leaving no edge from E at all.
Since we have defined Q as compatible with P in Definition 8, we know that $(P \cup Q)^+ = (P \cup Q)$ and since elements of T can be represented with at most one edge from E, we get $T = P \cup Q \cup ((P \cup Q) \circ E \circ (P \cup Q))$.
In this case both edges in E are consistent with the already known information, because there are no cyclic paths in T containing edges of P (c.f. condition 1.4.c) in [1]): This is because if there would be a cycle with edges in $(P \cup Q \cup E)$ and at least one egde from P (i.e. the new equivalence information would create an inconsistency in P^*), the cycle could be represented as $(a_0, a_1) \in P$ and $(a_1, a_2) \circ \ldots \circ (a_{n-1}, a_0)$ would at most contain one edge from E and thus the cycle is either of the form $P \circ (P \cup Q)$, or of the form $P \circ (P \cup Q) \circ E \circ (P \cup Q)$. In the first case there can be no cycle, because otherwise P and Q would already have been inconsistent, and if there would be cycle in the second case, there would exist objects $a, b \in O$ such that $(a, x) \in P$, $(x, y) \in E$ and $(y, b) \in (P \cup Q)$ and $(y, x) = (y, a) \circ (a, x) \in (P \cup Q) \circ P \subseteq P$ contradicting $(x, y) \notin P^{conv}$.
Because of $T = (P \cup Q \cup (P \circ E \circ P) \cup (Q \circ E \circ P) \cup (P \circ E \circ Q) \cup (Q \circ E \circ Q))$ and $P^* = T \setminus Q^*$ and since $(P \cup (P \circ E \circ P) \cup (Q \circ E \circ P) \cup (P \circ E \circ Q)) \cap Q^* = \emptyset$ (if the intersection would not be empty then due to Q^* being symmetric there would be a cycle in P^* with edges from $(P \cup Q \cup E)$ and at least one edge from P contradicting the condition 1.4 above), we finally get $P^* = (P \cup (P \circ E \circ P) \cup (Q \circ E \circ P) \cup (P \circ E \circ Q))$.
■

We have now found a way to derive P^* in the case of a new incremental equivalence relationship, but still P^* is a large relation containing all transitive information. We will now show that we can also get P^* by just manipulating a respective preference diagram in a very local fashion. Locality here results to only having to deal with edges that are directly adjacent in the preference diagram to the additional edges in E. Let us define an abbreviated form of writing such edges:

Definition 10: (Set Shorthand Notations)
Let R be a binary relation over a set of database objects O and let $x \in O$. We write:
$(\_\ R\ x) := \{ y \in O \mid (y, x) \in R\}$ and
$(x\ R\ \_) := \{ y \in O \mid (x, y) \in R)\}$
If R is an equivalence relation we write the objects in the equivalence class of x in R as:
$R[x] := \{ y \in O \mid (x, y), (y, x) \in R\}$

With these abbreviations we will show what objects sets have to be considered for actually calculating P^* via a given preference diagram:

Theorem 3: (Calculation of *PD(P))**
Let O be a set of database objects and P, P^{conv}, and Q as in Definition 8 and $E := \{(x, y), (y, x)\}$ new equivalence information such that
$(x, y), (y, x) \notin (P \cup P^{conv} \cup Q)$.
If $PD(P) \subseteq P$ is some preference diagram of P, and with

$PD(P)^* := (PD(P) \cup (PD(P) \circ E \circ Q) \cup (Q \circ E \circ PD(P)))$, holds: $(PD(P)^*)^+ = P^*$
i.e. $PD(P)^*$ is a preference diagram for P^*, which can be calculated as:
$PD(P)^* = PD(P) \cup ((\_\ PD(P)\ x) \times Q[y]) \cup ((\_\ PD(P)\ y) \times Q[x]) \cup (Q[x] \times (y\ PD(P)\_)) \cup (Q[y] \times (x\ PD(P)\_)\ )$.

Proof: We know from Theorem 2 that $P^* = (P \cup (P \circ E \circ P) \cup (Q \circ E \circ P) \cup (P \circ E \circ Q))$ and for preference diagrams $PD(P)$ of P holds:

a) $P = PD(P)^+ \subseteq (PD(P)^*)^+$

b) $(P \circ E \circ P) = (P \circ E) \circ P \subseteq (P \circ E \circ Q) \circ P = (PD(P)^+ \circ E \circ Q) \circ PD(P)^+ \subseteq (PD(P)^*)^+$, because $(PD(P)^+ \circ E \circ Q) \subseteq (PD(P)^*)^+$ and $PD(P)^+ \subseteq (PD(P)^*)^+$.

c) Furthermore $(P \circ E \circ Q) = PD(P)^+ \circ E \circ Q \subseteq (PD(P) \circ E \circ Q)^+ \subseteq (PD(P)^*)^+$

d) And similarly $(Q \circ E \circ P) = Q \circ E \circ PD(P)^+ \subseteq (Q \circ E \circ PD(P))^+ \subseteq (PD(P)^*)^+$

Using a) – d) we get $P^* \subseteq (PD(P)^*)^+$ and since $PD(P)^* \subseteq P^*$, we get $(PD(P)^*)^+ \subseteq (P^*)^+ = P^*$ and thus $(PD(P)^*)^+ = P^*$.
To calculate $PD(P)^*$ we have to consider the terms in $PD(P) \cup (PD(P) \circ E \circ Q) \cup (Q \circ E \circ PD(P)))$: The first term is just the old preference diagram. Since the second and third terms both contain a single edge from E (i.e. either (x, y) or (y, x)), the terms can be written as
$(PD(P) \circ E \circ Q) = ((\_\ PD(P)\ x) \times Q[y]) \cup ((\_\ PD(P)\ y) \times Q[x])$ and
$(Q \circ E \circ PD(P)) = (Q[x] \times (y\ PD\ (P)\_)) \cup (Q[y] \times (x\ PD(P)\_))$ ■

In general these sets will be rather small because first they are only derived from the preference diagram which is usually considerably smaller than preference P and second in these small sets there usually will be only few edges originating or ending in x or y. Furthermore, these sets can be computed easily using an index on the first and second entry of the binary relation $PD(P)$ and Q. Getting a set like e.g., $\_PD(P)\ x$ then is just an inexpensive index lookup of the type 'select all first entries from $PD(P)$ where second entry is x'.
Therefore we can calculate the incremented preference P^* by simple manipulations on $PD(P)$ and the computation of a transitive closure like shown in the commutating diagram in Figure 5.

$$\begin{tikzcd} P \arrow[r, "\text{extend by } E"'] & P^* = (PD(P)^*)^+ \\ PD(P) \arrow[u, "\text{trans. closure}"] \arrow[r, "\text{extend by } E"'] & PD(P)^* \arrow[u, "\text{trans. closure}"] \end{tikzcd}$$

Figure 5. Diagram for deriving incremented skylines using preference diagrams

Having completed incremental changes introduced by new equivalence information, we will now consider incremental changes by new preference information.

Theorem 4: (Incremental calculation of P^*)
Let O be a set of database objects and P, P^{conv}, and Q as in Definition 8 and $S := \{(x, y)\}$ new preference information such that $(x, y) \notin (P \cup P^{conv} \cup Q)$. Then P^* can be calculated as

$$P^* = (P \cup (P \circ S \circ P) \cup (P \circ S \circ Q) \cup (Q \circ S \circ P) \cup (Q \circ S \circ Q)).$$

Proof: The proof is similar to the proof of Theorem 2. Assume $(a, b) \in T$ as defined in Definition 8. The edge can be represented by a chain $(a_0, a_1) \circ \ldots \circ (a_{n-1}, a_n)$, where each edge $(a_{i-1}, a_i) \in (P \cup Q \cup S)$ and $a_0 := a$, $a_n := b$. This chain can even be transcribed into a representation with edges from $(P \cup Q \cup S)$, where edge (x, y) occurs at most once. This is because, if (x, y) occurs twice the two edges would enclose a cycle that can be removed.

Since we have assumed Q to be compatible with P in Definition 8, we know $T = P \cup Q \cup ((P \cup Q) \circ S \circ (P \cup Q))$ and (like in Theorem 2) edge (x, y) is consistent with the already known information, because there are no cyclic paths in T containing edges of P (c.f. 4.c in Theorem 1): This is because if there would be a cycle with edges in $(P \cup Q \cup S)$ and at least one egde from P (i.e. the new preference information would create an inconsistency in P^*), the cycle could be represented as $(a_0, a_1) \in P$ and $(a_1, a_2) \circ \ldots \circ (a_{n-1}, a_0)$ would at most contain one edge from S and thus the cycle is either of the form P, or of the form $P \circ (P \cup Q) \circ S \circ (P \cup Q)$. In the first case there can be no cycle, because otherwise P would already have been inconsistent, and if there would be cycle in the second case, there would exist objects $a, b \in O$ such that $(a, x) \in P$ and $(y, b) \in (P \cup Q)$ and $(y, x) = (y, a) \circ (a, x) \in (P \cup Q) \circ P \subseteq P$ contradicting $(x, y) \notin P^{conv}$.

Similarly, there is no cycle with edges in $(Q \cup S)$ and at least one egde from S, either: if there would be such a cycle, it could be transformed into the form $S \circ Q$, i.e. $(x, y) \circ (a, b)$ would be a cycle with $(a, b) \in Q$, forcing $(a, b) = (y, x) \in Q$ and thus due to Q's symmetry a contradiction to $(x, y) \notin Q$.

Because of $T = (P \cup Q \cup (P \circ S \circ P) \cup (Q \circ S \circ P) \cup (P \circ S \circ Q) \cup (Q \circ S \circ Q))$ and $P^* = T \setminus Q^*$ and since $(P \cup (P \circ S \circ P) \cup (Q \circ S \circ P) \cup (P \circ S \circ Q) \cup (Q \circ S \circ Q)) \cap Q^* = \emptyset$ (if the intersection would not be empty then due to Q^* being symmetric there would be a cycle in P^* with edges from $(P \cup Q \cup S)$ and at least one edge from P contradicting the condition 1.4 above), we finally get $P^* = (P \cup (P \circ S \circ P) \cup (Q \circ S \circ P) \cup (P \circ S \circ Q) \cup (Q \circ S \circ Q))$. ■

Analogously to Theorem 2 and Theorem 3 in the case of a new incremental preference relationship, we can also derive P^* very efficiently by just working on the small preference diagram instead on the large preference relation P.

Theorem 5: (Incremental calculation of *PD(P*)*)

Let O be a set of database objects and P, P^{conv}, and Q as in Definition 8 and $S := \{(x, y)\}$ new preference information such that $(x, y) \notin (P \cup P^{conv} \cup Q)$. If $PD(P) \subseteq P$ is some preference diagram of P, and with

$PD(P)^* := (PD(P) \cup (Q \circ S \circ Q))$, holds: $(PD(P)^*)^+ = P^*$

i.e. $PD(P)^*$ is a preference diagram for P^*, which can be calculated as:

$PD(P)^* = PD(P) \cup (Q[x] \times Q[y])$ with $PD(P) \cap (Q[x] \times Q[y]) = \emptyset$.

Proof: We know from Theorem 3 that $P^* = (P \cup (P \circ S \circ P) \cup (P \circ S \circ Q) \cup (Q \circ S \circ P) \cup (Q \circ S \circ Q))$ and for preference diagrams $PD(P)$ of P holds:

a) $P = PD(P)^+ \subseteq (PD(P)^*)^+$
b) since $S \subseteq Q \circ S \circ Q \subseteq PD(P)^*$,
   it follows $(P \circ S \circ P) \subseteq (PD(P)^*)^+ \circ PD(P)^* \circ (PD(P)^*)^+ \subseteq (PD(P)^*)^+$
c) since $Q \circ S \subseteq (Q \circ S) \circ Q \subseteq PD(P)^*$,
   it follows $((Q \circ S) \circ P) \subseteq PD(P)^* \circ (PD(P)^*)^+ \subseteq (PD(P)^*)^+$
d) analogously $(P \circ (S \circ Q)) \subseteq (PD(P)^*)^+ \circ PD(P)^* \subseteq (PD(P)^*)^+$
e) finally by definition $(Q \circ S \circ Q) \subseteq PD(P)^* \subseteq (PD(P)^*)^+$

Using a) – e) we get $P^* \subseteq (PD(P)^*)^+$ and since $PD(P)^* \subseteq P^*$, we get $(PD(P)^*)^+ \subseteq (P^*)^+ = P^*$ and thus $(PD(P)^*)^+ = P^*$.
To calculate $PD(P)^*$ analogously to Theorem 3 we have to consider the terms in $PD(P) \cup (Q \circ S \circ Q)$: The first term again is just the old preference diagram. Since the second term contains (x, y), it can be written as $(Q \circ S \circ Q) = (Q[x] \times Q[y])$. Moreover, if there would exist $(a, b) \in PD(P) \cap (Q[x] \times Q[y])$ then $(a, b) \in PD(P)$ and there would also exist $(a, x), (y, b) \in Q$. But then $(x, y) = (x, a) \circ (a, b) \circ (b, y) \in P$, because Q is compatible with P, which is a contradiction. ■

Thus, we can also calculate the incremented preference P^* by simple manipulations on $PD(P)$ in the case of incremental preference information. Again the necessary set can efficiently be indexed for fast retrieval. In summary, we have shown that the incremental refinement of skylines is possible efficiently by manipulating only the preference diagrams.

4 Conclusion

In this paper we laid the foundation to efficiently compute incremented skylines driven by user interaction. Building on and extending the often used notion of Pareto optimality, our approach allows users to interactively model their preferences and explore the resulting generalized skyline sets. New domination relationships can be specified by incrementally providing additional information like new preferences, equivalence relations, or acceptable trade-offs. Moreover, we investigated the efficient evaluation of incremented generalized skylines by considering only those relations that are directly affected by a user's changes in preference information. The actual computation takes advantage of the local nature of incremental changes in preference information leading to far superior performance over the baseline algorithms.
Although this work is an advance for the application of the skyline paradigm in real world applications, still several challenges remain largely unresolved. For instance, the time necessary for computing initial skylines is still too high hampering the paradigm's applicability in large scale scenarios. Here, introducing suitable index structures, heuristics, and statistics might prove beneficial.

References

[1] W.-T. Balke, U. Güntzer, C. Lofi. Eliciting Matters – Controlling Skyline Sizes by Incremental Integration of User Preferences. *Int. Conf. on Database Systems for Advanced Applications (DASFAA), Bangkok, Thailand, 2007*

[2] W.-T. Balke, U. Güntzer, C. Lofi. User Interaction Support for Incremental Refinement of Preference-Based Queries. *1st IEEE International Conference on Research Challenges in Information Science (RCIS),* Ouarzazate, Morocco, 2007.

[3] W.-T. Balke, U. Güntzer, W. Siberski. Getting Prime Cuts from Skylines over Partially Ordered Domains. *Datenbanksysteme in Business, Technologie und Web (BTW 2007)*, Aachen, Germany, 2007

[4] W.-T. Balke, U. Güntzer. Multi-objective Query Processing for Database Systems. *Int. Conf. on Very Large Data Bases (VLDB),* Toronto, Canada, 2004.

[5] W.-T. Balke, M. Wagner. Through Different Eyes - Assessing Multiple Conceptual Views for Querying Web Services. *Int. World Wide Web Conference (WWW)*, New York, USA, 2004.

[6] W.-T. Balke, U. Güntzer, W. Siberski. Exploiting Indifference for Customization of Partial Order Skylines. *Int. Database Engineering and Applications Symp. (IDEAS)*, Delhi, India, 2006.

[7] W.-T. Balke, J. Zheng, U. Güntzer. Efficient Distributed Skylining for Web Information Systems. *Int. Conf. on Extending Database Technology (EDBT)*, Heraklion, Greece, 2004.

[8] W.-T. Balke, J. Zheng, U. Güntzer. Approaching the Efficient Frontier: Cooperative Database Retrieval Using High-Dimensional Skylines. *Int. Conf. on Database Systems for Advanced Applications (DASFAA)*, Beijing, China, 2005.

[8] J. Bentley, H. Kung, M. Schkolnick, C. Thompson. On the Average Number of Maxima in a Set of Vectors and Applications. *Journal of the ACM (JACM),* vol. 25(4) ACM, 1978.

[9] S. Börzsönyi, D. Kossmann, K. Stocker. The Skyline Operator. *Int. Conf. on Data Engineering (ICDE)*, Heidelberg, Germany, 2001.

[10] C. Boutilier, R. Brafman, C. Geib, D. Poole. A Constraint-Based Approach to Preference Elicitation and Decision Making. *AAAI Spring Symposium on Qualitative Decision Theory*, Stanford, USA, 1997.

[11] L. Chen, P. Pu. Survey of Preference Elicitation Methods. *EPFL Technical Report IC/2004/67*, Lausanne, Swiss, 2004.

[12] J. Chomicki. Preference Formulas in Relational Queries. *ACM Transactions on Database Systems (TODS)*, Vol. 28(4), 2003.

[13] J. Chomicki. Iterative Modification and Incremental Evaluation of Preference Queries. *Int. Symp. on Found. of Inf. and Knowledge Systems (FoIKS)*, Budapest, Hungary, 2006.

[14] P. Godfrey. Skyline Cardinality for Relational Processing. *Int Symp. on Foundations of Information and Knowledge Systems (FoIKS)*, Wilhelminenburg Castle, Austria, 2004.

[15] W. Kießling. Foundations of Preferences in Database Systems. *Int. Conf. on Very Large Databases (VLDB)*, Hong Kong, China, 2002.

[16] V. Koltun, C. Papadimitriou. Approximately Dominating Representatives. *Int. Conf. on Database Theory (ICDT)*, Edinburgh, UK, 2005.

[17] D. Kossmann, F. Ramsak, S. Rost. Shooting Stars in the Sky: An Online Algorithm for Skyline Queries. *Int. Conf. on Very Large Data Bases (VLDB)*, Hong Kong, China, 2002.

[18] M. McGeachie, J. Doyle. Efficient Utility Functions for Ceteris Paribus Preferences. In Proc. of Conf. on Artificial In-telligence and Conf. on Innovative Applications of Artificial Intelligence (AAAI/IAAI'02), Edmonton, Canada, 2002.

[19] D. Papadias, Y. Tao, G. Fu, B. Seeger. An Optimal and Progressive Algorithm for Skyline Queries. *Int. Conf. on Management of Data (SIGMOD)*, San Diego, USA, 2003.

[20] J. Pei, W. Jin, M. Ester, Y. Tao. Catching the Best Views of Skyline: A Semantic Approach Based on Decisive Subspaces. *Int. Conf. on Very Large Databases (VLDB)*, Trondheim, Norway, 2005.

[21] T. Satty. A Scaling Method for Priorities in Hierarchical Structures. *Journal of Mathematical Psychology*, 1977

[22] T. Xia, D. Zhang. Refreshing the sky: the compressed skycube with efficient support for frequent updates. *Int. Conf. on Management of Data (SIGMOD),* Chicago, USA, 2006.

[23] Y. Yuan, X. Lin, Q. Liu, W. Wang, J. Yu, Q. Zhang. Efficient Computation of the Skyline Cube. *Int. Conf. on Very Large Databases (VLDB)*, Trondheim, Norway, 2005.