Efficient Skyline Queries under Weak Pareto Dominance

Wolf-Tilo Balke
L3S Research Center
University of Hannover
30539 Hannover, Germany
balke@l3s.de

Ulrich Güntzer
Department of Computer Science
University of Tübingen
72076 Tübingen, Germany
guentzer@informatik.uni-tuebingen.de

Abstract

Skylines with partial order preference semantics often result in huge answer sets and what is worse, they cannot be computed efficiently. In this paper we will explore the evaluation of so-called restricted skyline queries with partial order preferences under the paradigm of weak Pareto dominance. Weak Pareto dominance removes all objects from skylines, which are dominated by other objects in some query predicates, but in turn do not dominate these objects in any predicate. We will argue that this paradigm yields intuitive results, prove that it leads to lean sizes of the restricted skyline and show how it opens up the use of efficient algorithms for evaluation adopting the iteration of ranked result lists for each query predicate.

1 Introduction

Human preferences play an essential role in information systems, because exact match attribute-based querying without knowledge of the underlying database instance only too often produces empty or too big results. First approaches at cooperative databases like [Lacroix and Lavency, 1987; Motro, 1988] defined queries as wishes that could not always be satisfied, but would be automatically relaxed, if no perfect matches were found in the database instance.

Recently this paradigm has gained new attention: top-k queries [Güntzer et al., 2000; Fagin et al., 2001] have shifted retrieval models from simple exact matching of attribute values to the notion of *best matching* database objects. Top-k models rely on basic scorings of objects for each query predicate and a utility function to aggregate the objects' total scores. The paradigm has subsequently been extended under the name of skyline queries to cases where still score-based preferences exist for each query predicate, but no utility function exists to compromise between predicates, e.g. [Börzsönyi et al., 2001]. To get result sets in these cases skyline approaches adopted the principle of Pareto optimality, i.e. all objects are returned that have better or equal score values with respect to all query predicates and are strictly better in at least one. [Balke and Güntzer, 2004] then presented an algorithm that allowed evaluating interleaved skyline and top-k queries with optimal complexity.

While all these score-based approaches generally allowed for efficient query evaluation, their expressiveness in terms of user preferences remained rather limited, e.g. [Fishburn, 1999]. With the use of preferences in databases modeled as strict partial orders with an intuitive "I like A better than B" semantics [Kießling, 2002; Chomicki, 2002] this lack of expressiveness was remedied, however without providing an efficient evaluation of partial order preference queries. Also here the Pareto principle was the prime paradigm for evaluating queries involving several partial order preferences (if no ordering for the preferences themselves is provided). In [Kießling, 2002] a strong Pareto dominance principle called *'Pareto accumulation'* was presented, where an object had to be better or equal in all predicates and strictly better in at least one to dominate another object. In contrast [Chomicki, 2003] introduced a weak Pareto dominance principle called *'Pareto composition'*, where an object had to be better, equal or indifferent in all predicates and strictly better in at least one to dominate another object.

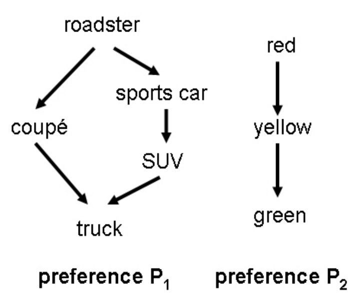
But for database retrieval such answer sets in the form of Pareto sets generally come at a price: Pareto sets grow exponentially in size with increasing numbers of preferences to combine [Bentley et al., 1978]. Hence, a further selection from Pareto sets is usually necessary to avoid the flooding of users with only more or less relevant results, c.f. [Koltun and Papadimitriou, 2005; Balke et al., 2005]. Due to the indifference property in partial order preferences the flooding effect for strong Pareto dominance even grows more dramatic. This is because unlike in score-based preferences, where all objects can be compared within each predicate, in partial order preferences an object can be the worst object with respect to almost all preferences, but will nevertheless be Pareto-optimal, if it is indifferent with respect to a single preference. Since users can not be expected to always completely model the preference relations of all possible attribute values within each query predicate carefully avoiding indifference, preference queries will usually involve some indifference and thus, inflate the size of result sets.

This behavior obviously is not sensible in practical applications and recent research in [Kießling, 2005] has started to combat indifference in partial order preferences by means of so called 'substitute values'. The substitute values semantics slightly changes the preference semantics such that some indifferent values become comparable and are assigned equal usefulness, if the indifferent values dominate and are dominated by exactly the same predicate values. First experiments in [Kießling, 2005] show that skyline sizes can already be considerably reduced using this semantics. Still, this semantics only remedies a small number of cases and does not provide efficient evaluation schemes.

In this paper we argue that using the weak Pareto dominance does not only allows the removal of less interesting objects from Pareto result sets, but also an efficient query evaluation. The usefulness of our approach is thus twofold: first restricted skylines help to effectively combat the explosion of result set sizes due to indifference for partial order preferences. On the other hand our approach allows deriving these restricted skylines without having to compute the entire Pareto set first. In the following we introduce an evaluation framework relying on ranked result lists for each query predicate and give a pruning condition, which allows us to derive an efficient algorithm to evaluate restricted skyline queries with partial order preferences.

2 The Weak Pareto Dominance Paradigm

In the following we will discuss our basic approach and present some motivating examples. Consider for instance the following two user preferences on car types and colors:



Example 1:
Consider the following database instance: a green roadster, a red coupé, a yellow SUV and a black truck. Due to the indifference between coupé and SUV and between red and black, yellow and black, or green and black, the skyline contains all four elements.

Using the strong definition of Pareto sets, in example 1 the whole data set would have to be delivered. Since a user usually is interested in only a few most interesting objects, a sophisticated selection from the skyline supports users.

2.1 Weak Pareto Dominance

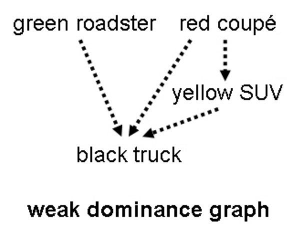
We will now define the weak Pareto dominance semantics (which is identical to the concept of Pareto composition in [Chomicki, 2003]) and show some first characteristics.

Definition 1: (weak Pareto dominance)
Let O be a set of database objects and $x, y \in O$. An object x is said to *weakly dominate* object y with respect to partial order preferences $P_1, \ldots, P_n$, if and only if there is an index i $(1 \leq i \leq n)$ such that x dominates y with respect to P_i and there is no index j $(1 \leq j \leq n)$ such that y dominates x with respect to P_j, i.e.
$x \blacktriangleright y \;:\Longleftrightarrow\; \exists\, i\ (1 \leq i \leq n): x >_{P_i} y \wedge \neg\exists\, j\ (1 \leq j \leq n): y >_{P_j} x$
where $>_P$ denotes the normal domination with respect to partial order P.

We will call the set of all non-weakly-dominated objects in the following the *restricted skyline*. Let us now consider some characteristics of weak Pareto dominance. First we have to note that weak Pareto dominance is not an order relation, because it is not necessarily transitive. Consider for instance three (incomplete) preference graphs P_1, P_2 and P_3 for three objects a, b and c. If $a >_{P1} b$ and $b >_{P2} c$ and $c >_{P3} a$, then we can derive $a \blacktriangleright b \blacktriangleright c \blacktriangleright a$ according to definition 1. Thus in some cases like in the (not very realistic) example from above, the restricted skyline can get empty due to intransitivity. But here also the normal skyline would yield the unconvincing result of simply returning all database objects and thus is not particularly helpful either.

Since we only use the weak Pareto dominance to limit down a result set, transitivity is not really needed. If there is only a cycle of weakly dominated objects on top level, there simply are no 'better' objects and this is reflected in our approach by the restricted skyline being empty. Because cooperative systems usually want to avoid the 'empty result effect', an adequate reaction of the system would consist in either returning all these objects from the cycle (like in the Pareto set) or even better in asking the user to reconsider some of her/his preferences involved in the cycle.

Example 1 (cont.): Consider the objects from above again under the notion of weak Pareto dominance. There is still no weak dominance relation between the green roadster, and the red coupé. But both of them weakly dominate the black truck and it can be removed in the restricted skyline.



Removing the black truck seems indeed a very intuitive thing to do anyway, because P_1 tells us that everything is better than a truck and the user, although giving explicit color preferences, did not voice his/her opinions on black cars. Moreover, we have to take a closer look at the relation between the red coupé and the yellow SUV. Since the user is indifferent between both car types, the red coupé fits his/her color wishes to a higher degree, hence is probably more desirable. The weak dominance relation reflects this semantics: the red coupé weakly dominates the yellow SUV and the yellow SUV can be removed in the restricted skyline. Thus, the result size in our small example is already halved.

Moreover, we can also easily see that restricted skylines really always are part of the normal skyline, i.e. Pareto set.

Lemma 1: (restricted skylines are part of Pareto skylines)
Let R be the restricted skyline set with respect to partial order preferences $P_1, \ldots, P_n$. Then R will never contain dominated objects under the notion of Pareto optimality with respect to $P_1, \ldots, P_n$.

Proof: Let o be any Pareto dominated object, but $o \in R$. Thus there must exist an object w in the Pareto skyline which dominates o, i.e. $\forall\, j\ (1 \leq j \leq n)\ (w >_{Pj} o \vee w =_{Pj} o) \wedge \exists\, i\ (1 \leq i \leq n)$: $w >_{Pi} o$. But the first part also implies $\neg\exists\, j\ (1 \leq j \leq n)$: $o >_{Pj} w$, thus w must also weakly dominate o in contradiction to $o \in R$. ■

If we focus on numerical preferences only, weak Pareto dominance and strong Pareto dominance are actually exactly the same. This is because numerical preferences impose a total ordering with respect to all predicates. Thus every two objects can be compared with respect to all predicates and if object *x* weakly dominates object *y*, then *x* dominates *y* also in the usual Pareto sense.

Moreover, we can state that the substitute values semantics in [Kießling, 2005] is a special case of the weak Pareto dominance semantics. This is because if any object *o* can be removed from the skyline under the substitute values semantics, there has to be an object *w* that for at least one preference P_i dominates *o* and with respect to all other preferences *w* either dominates *o*, or has equal or substitutable (i.e. indifferent values) values. Thus there cannot be any preference, where *o* dominates *w* and also our weak Pareto dominance semantics would remove *o* from the skyline set.

3 Evaluation of Restricted Skylines

In this section we focus on efficient evaluation schemes to derive restricted skyline sets from a (possibly large) number of database objects. Of course a naïve way of computing the set is to first derive the Pareto skyline, then test all pairs of objects for weak Pareto dominance and subsequently remove all weakly dominated objects. However, this is very inefficient way since for partial order preference skyline computation usually all database objects have to be accessed. Adopting an approach where each preference is processed by an (independent) subsystem, we will now focus on pruning large parts of the database, however, still deriving the correct restricted skyline.

3.1 Evaluation Scenario

In both top k and skyline retrieval the most often used scenario for evaluating complex queries was a middleware scenario, e.g. [Güntzer et al., 2000; Fagin et al., 2001; Balke and Güntzer, 2004]. Here (possibly independent) subsystems evaluated different query predicates by scoring a common set of database objects and delivering them in sorted order. Usually two kinds of accesses on subsystems are enabled:

- A *sorted access* iterated over the subsystem's objects rank by rank, retrieving (oid, score) pairs
- A *random access* asked for the specific score value for a certain object

To get the final top k or skyline result a central instance basically either iterates over the subsystems' sorted lists or requests scores for specific objects, until it can guarantee that all objects relevant to the result set have been accessed and hence all database objects still unseen can be ignored, i.e. pruned. Such retrieval schemes not only allow for a high degree of distribution, but also have been proven to perform very efficiently accessing only an instance-optimal number of database objects, cf. [Fagin et al., 2001; Balke and Güntzer, 2004]. However, all these schemes were exclusively designed for score-based retrieval, i.e. only consider numerical preferences imposing total orders. We will build our evaluation scheme for the same practical scenario, but enable the use of arbitrary partial order preferences.

3.2 Sorting under Partial Order Preferences

Our evaluation approach aims at pruning large parts of irrelevant database objects. Thus each subsystem has to sort objects in a way that possibly relevant objects are returned on smaller ranks than definitely irrelevant objects. We will use a simple breadth first topological ordering based on the partial order preference given for each query predicate.

Definition 2: (level of database objects)
Let *O* be a set of database objects and $x \in O$. An object *x* is said to *belong to level i* or *level(o) = i* with respect to partial order preference *P*, if and only if the longest path from any maximum object in P to *x* consists of (i - 1) edges.

Definition 2 implies that all objects not explicitly mentioned by *P* (this may be quite a large number), are considered to belong to level 1. This is necessary, if objects whose attribute values are all not explicitly mentioned in any preference (e.g. a white limousine in example 1) have to be in the restricted skyline. If a user really wants to see these objects, e.g. to allow for serendipity, performing an evaluation with definition 1 and 2 will get the correct result.

The intuitive notion of our levels is that of imposing a sensible order: all maximum (i.e. non-dominated) objects of *P* are on level 1, all objects that are only dominated in *P* by maximum objects are on level 2, and so on. In the special case of numerical or total order preferences the level corresponds to each object's rank, if objects with identical scores/attribute values are considered to have same rank. But this level order has another nice property:

Lemma 2: (level order domination)
Let *O* be a set of database objects and $x, y \in O$. Then object *x* can only dominate object *y* with respect to partial order preference *P*, if $level(x) < level(y)$ with respect to *P*.

Proof: If *x* dominates *y* there is a path of length $q > 1$ from *x* to *y* in *P*. Thus it directly follows from the definition of levels by longest paths in definition 2, that $level(x) < level(x) + q \leq level(y)$. ■

Please note, though objects can only be dominated by objects in higher levels, due to the partial order semantics they do not have to be dominated by all objects on higher levels. In the following we will assume all subsystems to return database objects for sorted access in *level order*. Consider for example preference P_1 in Example 1. Assuming that there are roadsters, coupés, sports cars, SUVs and trucks in the database, the level of all database objects that are roadsters is 1, the level of all coupés and sports cars is 2, the level of all SUVs is 3 and the level of trucks is 4. So our subsystem first has to return all roadsters, then coupés and sports cars can be returned in arbitrary order, followed by the SUVs and finally all trucks. In contrast, another possible topological ordering returning first roadsters, then sports cars, then SUVs and then coupés would violate the output in level order.

3.3 Pruning Database Objects

Given the basic scenario we will now introduce the concept of *l*-cuts, whose consideration is necessary to check whether all relevant, i.e. possibly not weakly dominated, objects have been accessed already.

Definition 3: (*l*-cut of preference orders)
Let O be a set of database objects and S be a sorting of O in level order with respect to partial order preference P. Then a subset $C \subseteq O$ is called a *l-cut* with respect to P, if
(a) $\forall\, w \in C : level(w) \leq l$
(b) $\forall\, (o \in S \wedge level(o) > l)\ \exists\, w \in C : w >_P o$

The intuitive meaning of *l*-cuts is to form sets of objects that dominate all objects below the *l*-th level. Every complete level of objects forms a trivial *l*-cut. But generally *l*-cuts will be much smaller and in the following we only need to consider minimum *l*-cuts. Consider for instance preference P_1 in example 1. Every single roadster forms a 1-cut with respect to P_1 (trivially the set of all roadsters also forms a 1-cut, but is not minimal). A 2-cut is formed by any pair of a coupé and a sports car. If there are no coupés in the database every single sports car will form a 2-cut.

In the special case of numerical or total order preferences every object forms a trivial cut: All objects on lower levels are automatically dominated by all objects belonging to higher levels. For instance in example 1 every red car will form a 1-cut with respect to P_2, every yellow car will form a 2-cut with respect to P_2, and so on.

Having defined the basic concept of *l*-cuts with respect to a single partial order preference, we will now present a way to guarantee during a preference query evaluation that all relevant objects for the restricted skyline computation have been accessed in at least some of the level sorted subsystems. This is the major component needed to build an efficient evaluation algorithm for partial order preference queries under the weak Pareto dominance paradigm. The following theorem will show a sufficient condition.

Theorem 1: (correctly pruning database objects)
Let O be a set of database objects, $S_1, \ldots, S_n$ be sortings of O in level order with respect to partial order preferences $P_1, \ldots, P_n$. Given $o_1, \ldots, o_k \in O$ and let $\{o_1, \ldots, o_k\}$ form an l_i-cut with respect to every sorting $S_1, \ldots, S_n$ for some natural numbers $l_1, \ldots, l_n$, then no object that for all i occurs on a higher level than l_i in S_i can be part of the restricted skyline under the notion of weak Pareto dominance with respect to $P_1, \ldots, P_n$.

Proof: Let $\{o_1, \ldots, o_k\}$ be as defined above and $o \in O$ be an object with $level(o) > l_i$ with respect to P_i ($1 \leq i \leq n$). For the sake of contradiction we will assume that object o belongs to the restricted skyline set thus it cannot be weakly dominated by any other object. Without loss of generality consider the first partial order preference P_1. Since $level(o) > l_1$ with respect to P_1 and $\{o_1, \ldots, o_k\}$ form an l_1-cut there has to be an object o_j ($1 \leq j \leq k$) that dominates o with respect to P_1, i.e. $o_j >_{P1} o$. For o not to be weakly dominated by object o_j, there has to be a preference P_q ($1 \leq q \leq n$) in which $o >_{Pq} o_j$. But since o_j is part of an l_q-cut with respect to P_q we get $level(o_j) \geq l_q > level(o)$ and thus an object with higher level would be dominated by an object with smaller level, which is impossible according to lemma 2. Hence, o is weakly dominated by o_j, i.e. not part of the restricted skyline in contradiction to our assumption. ■

Theorem 1 states a sufficient condition for the pruning of objects in partial order preference query evaluation and thus a basic evaluation algorithm can be derived. We will define a sorted access with respect to a query predicate by returning a pair consisting of an object's oid and score (if the predicate is given by a numerical preference) or an object's oid and attribute value (if the predicate is given by an attribute-based preference). Accordingly random accesses either return a score or an attribute value.

Basic Algorithm for Restricted Skyline Computation

1. Perform sorted accesses on all subsystems (e.g. in round robin fashion)
2. Consider all minimum *l*-cuts among the objects accessed (for all *l* smaller than the current levels)
3. Once all objects of some cut have been accessed in all subsystems, prune all objects on lower levels in all subsystems
4. For the remaining objects perform random accesses and check objects pairwise for weak Pareto domination
5. Remove all weakly dominated objects and return the restricted skyline

Again considering the special case of numerical preferences, we observed earlier that strong and weak Pareto dominance coincide. For skyline computation an instance-optimal condition is given in [Balke et al., 2004], where basically some object had to be accessed in every single subsystem by sorted access, before the unnecessary objects for the skyline computation could be pruned. Since for numerical preferences every single object forms a *l*-cut once it is accessed, both conditions are indeed equivalent and our algorithm for restricted skyline evaluation is also equivalent to the respective instance-optimal evaluation algorithm for numerical preferences given in [Balke et al., 2004].

4 Dealing with Implicit Isolated Maxima

Let us now consider an efficiency improvement for the case that users do not require objects, whose attribute values are never mentioned in any preference, to be returned.

We can easily see that these objects only can be returned, if we assign the level 1 to objects with values that do not explicitly occur in a preference, so-called implicit isolated maxima. This is because we might be able to prune all objects below level 1, which would then include all those objects not explicitly mentioned in any preference, i.e. global implicit isolated maxima. On the other hand our subsystems for the evaluation of single query predicates are independent. Thus, objects not mentioned in the respective preference can be arbitrary many, because users tend to state only incomplete preferences with just a few preferred attribute values instead of exhaustively modelling a domain. But accessing all objects that are implicit isolated maxima with respect to only some predicates is an unnecessary overhead.

Example 1 (cont.): Assume P_1 and P_2 are handled by two independent subsystems in ranked lists S_1 and S_2.

Imagine an object like a black limousine. Since neither P_1 nor P_2 mention its attribute values, it is a global implicit isolated maximum and not weakly Pareto dominated by any object, i.e. part of the restricted skyline. This is accommodated by definition 2 by assigning level 1 with respect to all preferences; thus *every l*-cut respects this object.

But let us now focus on the black truck, which according to definition 2 would occur on level 4 in S_1 and on level 1 in S_2. We have seen the black truck to be weakly dominated by *any* roadster, coupé, sports car or SUV. Nevertheless we would definitely access it in S_2, because 'black' is an implicit isolated maximum with respect to S_2. This is an undesirable effect that can only be avoided if we also assign a high level to black cars (in fact cars of any color except for red, yellow or green). Now the black truck can be pruned.

However, then also our black limousine would have to be assigned high levels in all preferences and would be pruned, though not being weakly dominated. Thus, this optimization only works, if a user is also willing to do without global implicit isolated maxima in the restricted skyline.

For improved efficiency in the following we will use a slightly different semantics:

- **Definition of skylines:** Amend the definition to be the set of all not weakly dominated objects without those objects from the skyline that *only* have attribute values, which are not explicitly mentioned in any preference.
- **Definition 2':** Alter definition 2 such that all objects not explicitly mentioned by P are considered to belong to the highest (i.e. worst) level, i.e. (maximum path-length in P) + 1.
- **Definition 3':** Alter condition (b) in definition 3 such that for all objects o in S belonging to levels higher than l either o is dominated by w or o is indifferent to all objects on level 1, i.e. incomparable to all maximum objects (and therefore also to all other objects).

Given these changes we can show that the pruning condition still holds, and we will only lose those objects in the restricted skyline, whose attribute values are not explicitly mentioned in any partial order preference, i.e. all global implicit isolated maxima.

Theorem 2: (pruning also implicit isolated maxima) Let O be a set of database objects, $S_1,\ldots, S_n$ be sortings of O in level order with respect to partial order preferences $P_1,\ldots, P_n$. Given $o_1,\ldots, o_k \in O$ and let $\{o_1,\ldots, o_k\}$ form an l_i-cut according to definition 3' with respect to every sorting $S_1,\ldots, S_n$ for some natural numbers $l_1,\ldots, l_n$, then no object that for all i occurs on a higher level than l_i in S_i can be part of the restricted skyline under the notion of weak Pareto dominance according to definition 1' with respect to $P_1,\ldots, P_n$.

Proof: Let us divide the objects not yet accessed into three possible classes:

(a) objects with attribute values mentioned in all preferences. For these objects theorem 1 still holds, because they have not been shifted in level, are not indifferent to at least one object of the respective cut in each preference and have not been removed from the restricted skyline.

(b) objects with at least one attribute value mentioned in some preference and with at least one attribute value not mentioned in some other preference. These objects have been shifted into the highest level for all predicate where their attribute values are not mentioned in the respective preference and are incomparable to all the maximum objects in these predicates. Thus, if a l-cut according to definition 3' dominates all mentioned attribute values, the objects are still weakly dominated with respect to definition 1 and thus correctly not part of the restricted skyline. If at least with respect to one preference they belong to a lower level than the l-cut they have been accessed by sorted access and will be correctly checked for weak domination.

(c) objects with attribute values mentioned in no preference. These objects reside due to definition 2' at the highest possible levels. Thus if a l-cut exists on a lower level in all preferences, they are pruned which is accommodated by definition 1'. ■

Theorem 2 thus allows us to improve our evaluation procedure even in the presence of partially modelled domains. Now only objects that stand a chance of being not weakly dominated are accessed in step 1 of the evaluation algorithm (except for global implicit isolated maxima). Please note that *explicit isolated maxima*, i.e. objects mentioned in some partial order preference, however having neither fathers nor descendants (e.g. given by a simple 'I like object A' semantics), are still considered for the restricted skyline, which is intuitive, since the user explicitly modelled them.

5 Summary and Outlook

Preference query evaluation plays an essential role in modern human-centered databases and information systems. In this paper we discussed the evaluation of database queries under the notion of *weak Pareto dominance*. We have shown this paradigm to decrease result set sizes, since it presents a stronger notion of optimality than given by strong

Pareto optimality even together with the notion of substitute values. Our improvement was gained by addressing the problem of indifference in partial order preferences (which usually increases result sizes) and presenting an intuitive way to choose only 'better matching' objects from the skyline. An object can be removed from our result set, if it is dominated by an object in some predicate, but it does not dominate this object with respect to other predicates. For total order preferences (that do not allow indifference) this notion coincides with usual Pareto optimality, but is a stronger paradigm for partial order preferences.

The resulting *restricted skyline* even has a more important advantage, since we have also presented a way to efficiently evaluate such queries without having to compute the real skyline before selecting the more relevant objects. Our evaluation scheme is based on the evaluation scenarios in top k database retrieval, where each query predicate is evaluated by (possibly independent) subsystems that offer individual rankings for the evaluation process. Those approaches, however, were restricted to total order preferences (i.e. numerical preferences), whereas our approach caters for arbitrary partial order preferences, thus avoiding the limited expressiveness of numerical preferences. We showed the algorithm's considerable optimization potential, which is gained by pruning definitely irrelevant database objects already at an early stage and then checking weak Pareto dominance relations only for a smaller subset of the database instance. We have also shown the instance optimal algorithm for numerical preference skyline evaluation to be a special case of our evaluation algorithm.

Our future work will focus on the implementation of our evaluation strategy in practical middleware scenarios and providing a detailed analysis and empirical quantification of the reduction in size of the restricted skyline set as opposed to normal skylines. Moreover, though we already know the number of object accesses (and thus execution times) to be instance-optimal for numerical preferences, we still need to investigate the speed-up of our evaluation scheme over skyline computations with partial order preferences. We will also focus on designing an efficient interleaved evaluation scheme for numerical and partial order preferences generalizing our work on multi-objective optimization in [Balke and Güntzer, 2004]. Finally, we want to experiment with a related approach for restricting skylines that would replace all given partial orders by the corresponding total orders induced by our level order semantics.

Acknowledgments

Part of this work was funded by the German Research Foundation (DFG) within the Emmy Noether excellence program.

References

[Balke and Güntzer, 2004] W.-T. Balke, U. Güntzer. Multi-objective Query Processing for Database Systems. In *Proc. of the Int. Conf. on Very Large Databases (VLDB)*, Toronto, Canada, 2004.

[Balke et al., 2004] W.-T. Balke, U. Güntzer, J. Zheng. Efficient Distributed Skylining for Web Information Systems. In *Proc. of the Int. Conf. on Extending Database Technology (EDBT)*, LNCS 2992, Heraklion, Crete, Greece, 2004.

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

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

[Börzsönyi et al., 2001] S. Börzsönyi, D. Kossmann, K. Stocker. The Skyline Operator. In *Proc. of the Int. Conf. on Data Engineering (ICDE)*, Heidelberg, Germany, 2001.

[Chomicki, 2002] J. Chomicki. Querying with Intrinsic Preferences. In *Proc. of the Int. Conf. on Extending Database Technology (EDBT)*, LNCS 2287, Prague, Czech Republic, 2002.

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

[Fagin et al., 2001] R. Fagin, A. Lotem, M. Naor. Optimal Aggregation Algorithms for Middleware. In *ACM Symp. on Principles of Database Systems (PODS'01)*, Santa Barbara, USA, 2001.

[Fishburn, 1999] P. Fishburn. Preference Structures and their Numerical Representations. *Theoretical Computer-Science*, vol. 217, 1999.

[Güntzer et al., 2000] U. Güntzer, W.-T. Balke, W. Kießling. Optimizing Multi-Feature Queries for Image Databases. In *Proc. of the Int. Conf. on Very Large Databases (VLDB)*, Cairo, Egypt, 2000.

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

[Kießling, 2005] W. Kießling. Preference Queries with SV-Semantics. In *Proc. of the Int. Conf. on Management of Data (COMAD)*, Goa, India, 2005.

[Koltun and Papadimitriou, 2005] V. Koltun, C. Papadimitriou. Approximately Dominating Representatives. In *Proc. of the Int. Conf. on Database Theory (ICDT)*, Edinburgh, UK, 2005.

[Lacroix and Lavency, 1987] M. Lacroix, P. Lavency. Preferences: Putting more Knowledge into Queries. In *Proc. of the Int. Conf. on Very Large Databases (VLDB)*, Brighton, UK, 1987.

[Motro, 1988] A. Motro. VAGUE: A User Interface to Relational Databases that Permits Vague Queries. In *ACM Transactions on Office Information Systems (TOIS)*, vol. 6(3), 1988.