

# Choosing the Right Thing: Cooperative Trade-Off Enhanced Skyline Queries

Christoph Lofi<sup>#1</sup>

<sup>#</sup>*Institute for Information Systems, Technische Universität Braunschweig  
Mühlenpfordtstraße 23, 38106, Braunschweig, Germany*

<sup>1</sup>lofi@ifis.cs.tu-bs.de

**Abstract**— Skyline queries are well-known for their intuitive query formalization and easy to understand semantics for selecting the most interesting data objects from large data sets. They naturally fill the gap between set-based queries using strict predicates and only few personalization options and rank-aware database retrieval, offering a high degree of personalization at the cost of very complex query formalization. Thus, skyline queries enjoyed popularity in the database personalization research community. Unfortunately, the simplicity and elegance of the query paradigm come at high costs: skyline queries often suffer from a problem usually known as “curse of dimensionality”. With the increasing number of query attributes, the size of skyline result sets grows exponentially and the results are thus hardly useful or manageable by users. This problem severely hinders the practical application of the skyline paradigm.

In this paper, the concept of trade-offs skylines is proposed as a natural extension to the skyline paradigm which is specifically designed as a remedy for the curse of dimensionality.

## I. INTRODUCTION

The ever growing amount of available information is one of the major problems of today’s information systems. Besides solving the resulting performance issues, it is imperative to provide personalized and tailored access to the vast amount of information available in information and data systems in order to avoid flooding the user with unmanageably large query results.

As a promising remedy to this problem, Skyline queries [1] have been proposed, filling the gap between set-based SQL queries and rank-aware database retrieval [2]. Due to the paradigms elegance and simplicity, it has stirred a lot of interest within the database community in recent years. Skyline queries rely on the notion of *Pareto dominance*, i.e. given the choice between two objects, with one object being better with respect to at least one attribute but at least equal with respect to all other attributes, users will always prefer the first object over the second one (the first object is said to *dominate* the second one). This simple concept can be used to implement an intuitive personalizable data filter as dominated objects can be safely excluded from the data collection, resulting in the *Skyline set* of the query. The semantic justification of this filter is easy to see using a simple example: if two car dealers in the neighborhood offer the same model (with same warranties, etc.) at different prices, why should one want to consider the more expensive car?

In order to compute the Skyline set in a personalized fashion, the user needs only to provide so-called *ceteris paribus* (“all other being equal”) preferences on each individual attribute (often called attribute preferences, e.g. “lower prices are better than higher prices given that all other attributes are equal”). This focus on individual attribute domains and the complete fairness of the Pareto paradigm are the major advantages of skyline queries: they are easy to specify and the algorithm will only remove definitely suboptimal objects. However, these characteristics also directly lead to the paradigms major shortcomings: skyline queries completely lack the ability to relate attribute domains to each other and thus prevent compensation, weighting or ranking *between* attribute domains. This often results in most objects being incomparable to each other and thus generally causes skyline sets to be rather large, especially in the quite common case of anti-correlated attribute dimensions. This effect is usually referred to as “curse of dimensionality”. It has been shown that the skyline size grows roughly exponential with the number of query attributes [3,4]. Experimentally, it has been validated that already for only 5 to 10 attributes, skylines can easily contain 30% or more of the entire database instance [1,5,6]. This is a size clearly unmanageable for most users, rendering the skyline paradigm inapplicable for many real-world problems.

In this paper, the concept of trade-off skylines is proposed which is directly targeted at abating the problem of overly large skyline sets. In order to motivate the extension of skyline queries with user trade-offs, the next section will present a brief survey on alternative approaches which also aim at countering the curse of dimensionality and discuss their advantages and shortcoming. In section III, trade-off skylines are presented and explained. Final discussions are part of section IV.

## II. RELATED WORKS AND DISCUSSIONS

Reducing the size of result sets by choosing the most interesting or most relevant objects from the skyline is a major and prominent problem. However, “interestingness” is usually an individual perception and is specific for each user and is thus hard to formalize. Nevertheless, for rendering the skyline paradigm useful in common real world scenarios, such techniques are mandatorily required. Accordingly, an impressive number of approaches have been developed in the recent years

introducing various heuristics for capturing the semantics of “interesting” in order to choose meaningful and manageable subsets from skylines in an efficient manner.

Generally speaking there are three major groups of approaches to address the problem:

#### A. Relaxation of Pareto Semantics

Considering the definition of Pareto semantics, it is obvious that the manageability problems of skylines are heavily aggravated by incomparable attribute values, especially when working with natural preferences which are modeled as partial orders [7,8]. As soon as two database items are incomparable with respect to even a single attribute, both objects are incomparable and may end up in the skyline. One could say that the Pareto semantics generally is ‘too fair’. Accordingly, the first group of approaches uses weaker variants of the Pareto semantics which less likely lead to incomparability between database objects. Notable works in this spirit are *weak skylines* [9] which replace the Pareto definition for domination with: “one object is better than another one when it is better with respect to one attribute and not worse with respect to any other”, and *k-dominant skylines* [10] which require only a user-given number of  $k$  attributes to fulfill the Pareto condition. Skylines resulting from relaxed Pareto definitions can be of significant smaller size. However, their semantics are often hard to justify and the implied heuristics have a strong “ad-hoc” character. For example, such skylines may easily remove objects which are highly interesting to the user, and thus rendering the practical application semantically difficult.

#### B. Summarization approaches

Summarization approaches aim at finding a subset of objects which serves optimally as a summarization of the full skyline. The summarizing set should still maintain full the diversity and characteristics of the original skyline, but should be of a much more manageable size. The focus of these approaches is to enable the user to grasp a quick overview of the nature and contents of the skyline result set such that she is easily able to further refine her preferences and / or is directly able to perform subsequent queries for narrowing down the results even further (e.g. appending a top- $k$  query which ranks the skyline result, or provide some SQL constraints to remove unwanted data points). Notable approaches are *approximately dominating representatives* [11] which return a subset minimally covering all skyline objects with some  $\epsilon$ -balls and *statistical sampling approaches* [6] with subsequent top- $k$  ranking. Both approaches try to maintain the diversity of the original skyline. However, summarized skylines are only useful if they are intended to provide a quick overview and should be accompanied by additional succeeding queries which focus on the most interesting object from a user’s perspective.

#### C. Weighting approaches

Weighting approaches try to induce a ranking on skyline items based on some structural properties of the data set. The Pareto skyline operator treats all skyline objects as being equal, i.e. it does not impose any ranking on the result set. However, weighting approaches claim that there are more

interesting and less interesting skyline objects, and that “interesting” can be captured by properties like e.g. the data distribution, the structure of the subspace skylines, or other statistical means. Usually, they explicitly quantify the “interestingness” of a skyline object numerically and return the  $k$ -most interesting objects.

Especially subspace analysis [12] has gained a lot of attention which was encouraged by the development of efficient algorithms for materializing the possibly  $2^d - 1$  subspace skylines (see e.g. SkyCubes [13]). For example, subspace analysis can be used to define *top- $k$  frequent skylines* [14] which capture “interestingness” counting the occurrence of an object in each of the non-empty subspace skylines, i.e. claiming that objects which are more frequent in subspaces are also more interesting. A more elaborate subspace based ranking is provided by SkyRank [15], which uses subspace domination relationships of the full space skyline objects to construct a so called skyline graph which is used for a subsequent link-analysis and provides the interestingness scores with a variant of PageRank.

Other approaches use the number of dominated objects as a metric for interestingness, resulting in the *k Most Representative Skyline* [16], or elicit additional preferences expressing a precedence of the query attributes for constructing a ranked result set based on the subspace frequency of objects and the precedence of the attributes defining the subspace (e.g. Telescope [17]).

However, all of these presented approaches break the absolute fairness of Pareto semantics in some way and replace it with some heuristics for removing “unwanted” objects. While each of those approaches has benefits and advantages on their own right, the imposed heuristics all rely on some “ad-hoc” assumptions on what makes a skyline point more interesting than others. However, the “correctness” and usefulness of these assumptions with respect to the real information needs of a given, individual user is very subjective and thus hard to determine.

In this work, trade-off skylines are presented which have been developed during my PhD research. Trade-off skylines are based on a completely different base idea: instead of relying on structural and semantically questionable heuristics for capturing interestingness, trade-off skylines interactively elicit additional user feedback to steer the selection of Skyline tuples from large result sets. This additional feedback is provided in the form of trade-offs which are especially designed to allow for a strictly *qualitative compensation* between individual attribute domains and closely resemble the concept of natural *compromises* which are part of each person’s every day’s decision processes. Especially, this empowers users to obtain a selection from the skyline set which is truly personalized and not computed by some user oblivious heuristics.

### III. TRADE-OFF SKYLINES

For motivating trade-off skylines, consider the following two database objects representing cars: let object A be a ‘blue metallic’ car for \$18,000 and object B be a ‘blue’ car for \$17,000, accompanied by a preference favoring cheaper cars and metallic colors. Looking at the ranking on attribute level, both cars are incomparable with respect to the Pareto order: one car is cheaper; the other car has the more preferred color. In this scenario, a natural question of a real-life car dealer would be, whether the customer is willing to compromise on those attributes, i.e. if she is willing to pay the additional \$1,000 for a metallic paint job for that particular car (such a compromise is called a trade-offs). If the answer is yes, then object A is the better choice for the user and should dominate object B with respect to a trade-off enhanced Pareto order. However, if some object C like a ‘blue’ car for \$15,000 exists, A and C would still be incomparable as the premium for the metallic color on that car C is larger than the \$1,000 the user is willing to pay. When adding some strong trade-offs, many skyline objects can now be removed and thus the skyline is focused consistently with the refined trade-off enhanced user preferences. Additionally, this kind of user interaction closely models the natural compromises of peoples every day’s decision processes. At the same time, the approach abstains from assuming arbitrary user agnostic heuristics for selecting objects from a too large skyline. Also, please note that the notion of trade-offs extends considerably beyond the expressiveness of Pareto skylines which can only rely on preferences on individual attribute domains.

Of course, eliciting additional feedback from users puts an additional burden on the interaction process compared to user agnostic approaches like those presented in the last section. However, by designing an intuitive interaction process, this burden can be alleviated. Trade-offs are designed to be elicited interactively, i.e. after computing a preliminary skyline, the user is guided through a trade-off elicitation process which suggests possible effective trade-offs (similar to a car dealer asking his customer additional questions). After the user decides for a trade-off, the trade-off skyline is recomputed and the user interaction continues until the user is satisfied. Be-

sides directly providing user trade-offs, two heuristic elicitation frameworks have been proposed: one based on suggesting promising trade-offs to the user [18], and another one relying on just simple object comparison feedback [19]. Based on a simple “I prefer this object over that object” statement, a set of trade-offs is heuristically deduced, thus allowing for a very intuitive user experience (e.g. see Fig. 1).

During our work on [18], we also evaluated the effects of simple trade-offs on the properties of resulting trade-off refined skyline sets. As an example, we will present an evaluation performed on a real-life data set containing 20,537 sale offers for notebook computers. After providing 7 base preferences, the resulting Pareto skyline was computed containing still 182 notebook offers, including all types of notebooks from lightweight netbooks to large and heavy desktop replacements. In this evaluation, we assume that the user is willing to sacrifice mobility in favor for performance and display size. Using a trade-off elicitation heuristic [18], this results in 13 trade-offs (which have been inferred from just two user interactions). Incorporating these trade-offs reduces the skyline set to just 59 notebooks (32% of the original skyline’s size). Furthermore, the *focus* and quality of the result is increased: instead of containing large numbers of notebooks from all different categories, the result focuses on 17”-screen notebooks (a cluster containing the majority of all desktop replacement machines) while many of the smaller notebooks have been removed as a result of the provide trade-offs (see Fig. 1). However, in contrast to simple filters, this refinement retained important characteristics of the Pareto semantics as all remaining notebooks, even those in the cluster of smaller netbooks, are especially interesting and non-dominated objects which are potentially still a good deal even after the user refined his intentions by proving trade-offs.

Efficiently computing trade-off skylines is quite hard. This is mainly due to trade-offs directly modifying the object order resulting from the Pareto aggregation which in turn loses its *separability characteristic* [20]. Separability describes the possibility of decomposing the object order losslessly into its respective base preferences (this why most skyline algorithms can avoid operating on the object order at all). In contrast, in our early works [21,22], it was shown that trade-offs will induce additional relationships into the object order and thus breaking the separability. Materializing at least some parts of



Fig. 1 Qualitative Object Comparison Interface for heuristically deducing trade-offs. Only simple interaction is required.

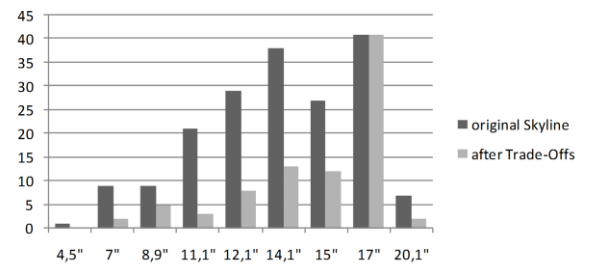


Fig. 2 Skyline Result Focus  
Notebook dataset; user is looking for a desktop replacement  
y-axis: number of result objects, x-axis: display size of notebook

the object order is mandatory if the full semantics of trade-offs shall be used. This effect can be explained by the definition of trade-offs: trade-offs can be considered as a user decision between two sample objects focusing on only a subspace of the available attributes, while treating all other attributes with *ceteris paribus* semantics. Furthermore, trade-off relationships are transitive and thus may form complex domination relationships structures spanning several trade-offs and dimensions which are called *trade-off chains*. Especially, the problem of trade-off inconsistencies poses severe challenges as inconsistencies are difficult to detect as they are basically circles in the materialized object order. This problem has been solved in a subsequent work [23].

The algorithms available for computing trade-off skylines have evolved from early algorithms relying on the full materialization of the object order [22], trade-off skylines with severely reduced expressiveness but not requiring the object order at all [18], to computing trade-off skylines allowing the specification of any consistent trade-off without restrictions while still providing acceptable performance by relying on a compressed datastructure minimally representing those parts of the object order which are crucial for computing the respective trade-off skyline [24].

#### IV. CONCLUSION

Trade-Off skylines are a novel cooperative approach for mastering the curse of dimensionality for skyline queries in the area of database personalization. Instead of relying on some user oblivious heuristics, additional user feedback in the form of trade-offs is interactively elicited. Trade-off skylines extend the semantics of Pareto skylines by allowing the qualitative compensation between different attribute domains. This compensation allows focusing a skyline result set further, and thus represents a truly personalized approach for choosing interesting objects from skylines.

During the course of my PhD research, the theoretical foundations for trade-off skyline have been provided as well as different algorithms for actually computing them. Also, the problem of trade-off inconsistency has been addressed and first approaches for trade-off enabled user interfaces and elicitation heuristics have been designed. Future works will delve deeper into the challenge of intuitive user interaction, especially focusing on ease of use and the semantic implications of trade-off skylines.

#### ACKNOWLEDGMENT

I especially thank Prof. Dr. Wolf-Tilo Balke who is supervising my doctoral thesis.

#### REFERENCES

- [1] S. Borzsonyi, D. Kossmann, K. Stocker, and U. Passau, "The Skyline Operator," *Int. Conf. on Data Engineering (ICDE)*, Heidelberg, Germany: 2001, pp. 421-430.
- [2] R. Fagin, A. Lotem, and M. Naor, "Optimal aggregation algorithms for middleware," *Symposium on Principles of Database Systems (PODS)*, Sanata-Barbara, California, USA: 2001, p. 102.
- [3] J.L. Bentley, H.T. Kung, M. Schkolnick, and C.D. Thompson, "On the Average Number of Maxima in a Set of Vectors and Applications," *Journal of the ACM (JACM)*, vol. 25, 1978.
- [4] S. Chaudhuri, N. Dalvi, and R. Kaushik, "Robust Cardinality and Cost Estimation for Skyline Operator," *22nd Int. Conf. on Data Engineering (ICDE)*, Atlanta, Georgia, USA: 2006.
- [5] P. Godfrey, "Skyline cardinality for relational processing. How many vectors are maximal?," *Symp. on Foundations of Information and Knowledge Systems (FoIKS)*, Vienna, Austria: 2004.
- [6] W.-T. Balke, J.X. Zheng, and U. Guntzer, "Approaching the Efficient Frontier: Cooperative Database Retrieval Using High-Dimensional Skylines," *Int. Conf. on Database Systems for Advanced Applications (DASFAA)*, Beijing, China: 2005.
- [7] M. Lacroix and P. Lavency, "Preferences: Putting More Knowledge into Queries," *Int. Conf. on Very Large Data Bases (VLDB)*, Brighton, UK: 1987.
- [8] C.-Y. Chan, P.-K. Eng, and K.-L. Tan, "Stratified computation of skylines with partially-ordered domains," *International Conference on Management of Data (SIGMOD)*, Baltimore, USA: 2005.
- [9] W.T. Balke, U. Guntzer, and W. Siberski, "Restricting skyline sizes using weak Pareto dominance," *Informatik - Forschung und Entwicklung*, vol. 21, May. 2007, pp. 165-178.
- [10] Chee-Yong Chan, "Finding k-dominant skylines in high dimensional space," *ACM SIGMOD Int. Conf. on Management of Data (SIGMOD 2006)*, Chicago, Illinois, USA: 2006.
- [11] V. Koltun and C. Papadimitriou, "Approximately dominating representatives," *10th International Conference on Database Theory (ICDT 2005)*, Edinburgh, Scotland: 2005.
- [12] Jian Pei, "Catching the best views of skyline: a semantic approach based on decisive subspaces," *31st Int. Conf. on Very Large Databases (VLDB '05)*, Trondheim, Norway: 2005.
- [13] Yidong Yuan, "Efficient computation of the skyline cube," *31st Int. Conf. on Very Large Databases (VLDB '05)*, Trondheim, Norway: 2005.
- [14] C.-Y. Chan, H.V. Jagadish, K.-L. Tan, A.K.H. Tung, and Z. Zhang, "On High Dimensional Skylines," *Advances in Database Technology (EDBT 2006)*, Munich, Germany: 2006.
- [15] A. Vlachou and M. Vazirgiannis, "Ranking the sky: Discovering the importance of skyline points through subspace dominance relationships," *Data & Knowledge Engineering*, vol. 69, Sep. 2010, pp. 943-964.
- [16] X. Lin, Y. Yuan, Q. Zhang, and Y. Zhang, "Selecting Stars: The k Most Representative Skyline Operator," *23rd IEEE International Conference on Data Engineering*, Istanbul, Turkey: IEEE, 2007, pp. 86-95.
- [17] J. Lee, G.-won You, and S.-won Hwang, "Personalized top-k skyline queries in high-dimensional space," *Information Systems*, vol. 34, Mar. 2009, pp. 45-61.
- [18] C. Lofi, W.-T. Balke, and U. Guntzer, "Efficient skyline refinement using trade-offs respecting don't-care attributes," *International Journal of Computer Science and Applications*, vol. 6, 2009.
- [19] C. Lofi, W.-T. Balke, and U. Guntzer, "Eliciting Customer Wishes using Example-Based Heuristics in E-Commerce Applications," *IJIS Technical Report*, 2010.
- [20] Sven Ove Hansson, "Preference Logic," *Handbook of Philosophical Logic*, vol. 4, 2002, pp. 319-393.
- [21] W.-T. Balke, C. Lofi, and U. Guntzer, "Incremental Trade-Off Management for Preference Based Queries," *International Journal of Computer Science & Applications (IJCSA)*, vol. 4, 2007.
- [22] W.-T. Balke, U. Guntzer, and C. Lofi, "Eliciting Matters - Controlling Skyline Sizes by Incremental Integration of User Preferences," *Int. Conf. on Database Systems for Advanced Applications (DASFAA)*, 2007.
- [23] C. Lofi, W.-T. Balke, and U. Guntzer, "Consistency check algorithms for multi-dimensional preference trade-offs," *International Journal of Computer Science & Applications (IJCSA)*, vol. 5, 2008, pp. 165-185.
- [24] C. Lofi, U. Guntzer, and W.-T. Balke, "Efficient Computation of Trade-Off Skylines," *13th International Conference on Extending Database Technology (EDBT)*, Lausanne, Switzerland: 2010.