Noname manuscript No.
(will be inserted by the editor)

Understanding the Meaning of a Shifted Sky

A General Framework on Extending Skyline Query

Zhenjie Zhang · Hua Lu · Beng Chin Ooi · Anthony K. H. Tung

Received: date / Accepted: date

Abstract Skyline queries are often used on data sets in multi-dimensional space for many decision-making applications. Traditionally, an object p is said to dominate another object q if, for all dimension, it is no worse than q and is better on at least one dimension. Therefore, the skyline of a data set consists of all objects not dominated by any other object.

To better cater to application requirements such as controlling the size of the skyline or handling data sets that are not well-structured, various works have been proposed to extend the definition of skyline based on variants of the dominance relationship. In view of the proliferation of variants, in this paper, a generalized framework is proposed to guide the extension of skyline query from conventional definition to different variants. Our framework explicitly and carefully examines the various properties that should be preserved in a variant of the dominance relationship so that: (1) maintaining original advantages, while extending adaptivity to application semantics, and (2) keeping computational complexity almost unaffected. We prove that traditional dominance is the only relationship satisfying all desirable properties, and present some new dominance relationships by relaxing some of the properties. These relationships are general enough for us to design new top-k skyline queries that return robust results of a controllable size. We analyze the existing skyline algorithms based on their minimum requirements on dominance properties. We also extend our analysis to data sets with missing values, and present extensive experimental results on the combinations of new dominance relationships and skyline algorithms.

1 Introduction

Given a set P of d-dimensional points, point p is said to *dominate* point q if p is no worse than q on any dimension and better than q on at least one. The subset of all points not dominated by others is called the *skyline* of P. A skyline query retrieves from P its skyline, which is interesting to users with multiple criteria, especially from economical perspective [15,24]. These are the traditional skyline concepts that have formed the basis for most previous works on skyline queries [4,22,13,18,10]. The essence of a skyline query is its **dominance** definition, which not only determines the relationship between any pair of points but also shapes the final query result. There are both benefits and negative effects in adopting the traditional dominance relationship for the definition of a skyline point.

In terms of benefits, the traditional dominance relationship guarantees the robustness of the query re-

Zhenjie Zhang
Department of Computer Science
School of Computing
National University of Singapore
E-mail: zhenjie@comp.nus.edu.sg

Hua Lu
Department of Computer Science
Faculties of Engineering, Science, and Medicine
Aalborg University
E-mail: luhua@cs.aau.dk

Beng Chin Ooi
Department of Computer Science
School of Computing
National University of Singapore
E-mail: ooibc@comp.nus.edu.sg

Anthony K. H. Tung
Department of Computer Science
School of Computing
National University of Singapore
E-mail: atung@comp.nus.edu.sg

sult. This is because scaling and shifting on any dimension do not impact the query result. For example, if the objects are associated with two attributes, including temperature and weight, the skyline points remain the same regardless of whether temperature is represented in Fahrenheit or Celsius, or whether weight is measured in kilograms or pounds. This property is the most important advantage of the skyline query, making it the only option for the user when dealing with incomparable dimensions.

As for negative effects, the rigorous definition of the traditional dominance relationship restricts the usefulness of the skyline query in real-life applications, which usually demand additional requirements of skyline points. One common requirement is the control over the result size of a skyline query, i.e., the number of skyline points being returned. This has in fact motivated the introduction of variances into traditional skyline queries [12,16,6]. Another requirement is to provide systematic flexibilities in tuning skyline points selection. Given a variant definition of skyline query, the result returned is expected to be both meaningful on semantics and controllable on cardinality. Yet another requirement is to apply skylining power to handle data sets that are not so well-structured, such as tables with missing values.

In view of these trends, in this paper, we formulate a generalized framework to serve as the basis for defining and examining variants. To formulate the framework, we carefully examine various properties that should be preserved in a variant of the dominance relationship so that: (1) the definition of skyline maintains its original advantages as much as possible while remaining adaptive to application semantics, and (2) the computational complexity of skyline based on a new variant is not too adversely affected. Through the introduction of this framework, we hope to remove the needs for researchers to re-examine these properties whenever there is a necessity to define a variant of the dominance relation or re-develop a new algorithm for computing skyline based on the new variant. This is done in the same spirit as previous work like [21] which provides theoretical answer to the question on when is nearest neighbors indexable.

Unlike previous studies on preference queries in relational databases [7,11] that focus on traditional properties for total and partial orders, we emphasize two important properties in the traditional dominance relationship: *scaling robustness* and *shifting robustness*. With these two properties, the skyline set remains robust even when the dimensions are totally incomparable [4]. Besides these two properties, we are also interested in the rationality property and the transitivity property, both of which are crucial to algorithm design for the skyline query. It is also interesting to see the traditional dominance relationship being the only binary relationship satisfying all the properties above.

Further, we consider relaxing one of the following properties: transitivity, scaling robustness and shifting robustness, as doing so allows us to design dominance relationships such that the size of the skyline can be controlled. We show that these relationships are likely to form an ordered class $\{D_1, D_2, \ldots, D_n\}$, with the property that object p can dominate object q under D_i for any $i \leq j$ if p can dominate q under D_j. Based on such an ordered class property, we propose a new type of top-k skyline query which attempts to find the smallest D_i such that the corresponding skyline is of a size smaller than a user specified parameter k.

On the efficiency issue of skyline query computation, we study some existing skyline algorithms, such as *BNL*[4], *SFS*[8], *TSA*[6] and *BBS*[19]. We analyze their applicable ranges by looking at the minimum requirement on the properties of the underlying dominance relationship. Additionally, we propose two algorithm frameworks for the top-k skyline query, namely *Binary Search* and *Progressive Search*, as well as their applicability conditions.

To illustrate the extensibility of our proposal, we apply our principles in two contexts. First, we apply our analysis to design a new dominance relationship called *cone dominance* which allows us to reduce the skyline size while sacrificing either scaling or shifting robustness. Second, we apply our analysis to handle data sets with missing values without losing any of the desirable properties. In both cases, we then select the appropriate algorithms for finding skyline and top-k skyline based on the exhibited properties.

The rest of the paper is organized as follows. Section 2 introduces some basic properties of the dominance relationship and reviews some related works. Section 3 studies dominance relationships that satisfy all or parts of the properties. Section 4 summarizes the current algorithms based on their basic requirements on dominance properties. Section 5 looks at the design of cone dominance in order to control the size of the skyline. Section 6 extends the analysis to data sets with missing values. Section 7 presents the experimental results and Section 8 concludes the paper.

2 Preliminaries

In this section, we first give the common definitions and notations used in rest of the paper. Then, we review some related work in the literature of skyline query processing.

2.1 Definitions and Notations

Given a d-dimensional numerical space $\mathcal{S}$, a point p in the space $\mathcal{S}$ is represented by a d-dimensional vector $(p[1], p[2], \ldots, p[d])$. In this space, we can define a binary relationship $D : \mathcal{S} \times \mathcal{S}$ called a dominance relationship. A point p is said to dominate another point q, if (p, q) is in D, denoted as $D(p, q)$. We will also use $\overline{D}(p, q)$ to denote that (p, q) is not in D, or p cannot dominate q. Based on the dominance relationship, we can define the skyline of a data set $P \subseteq \mathcal{S}$ as a subset $\mathbf{S}(P, D)$ of P, which contains all points not dominated by any point in P, i.e., $\mathbf{S}(P, D) = \{p \in P | \forall_{q \in D} \overline{D}(q, p)\}$. Since every dimension is simply numerical, the basic preference follows one of the two cases that, smaller value dominates larger value, or opposite. Without loss of generality, we simply assume that a point p is better than another point q on a dimension i, if $p[i] < q[i]$.

In most of the previous studies on skyline, a skyline query typically employs the traditional dominance relationship, where a point p dominates another point q, if p is not worse than q on all dimensions and p is better than q on at least one dimension. To distinguish the traditional dominance relationship from other dominance relationships, we shall call it TD.

Definition 1 Dominance Region
Based on the specified dominance relationship D, the dominance region of a point p is the largest region in $\mathcal{S}$ such that every point in the region must be dominated by p with respect to D.

The dominance region of a point p in the traditional dominance relationship TD, for example, is the hyper-rectangle in the space for points with no smaller values on all dimensions, except for the position of p itself.

In this paper, we focus on the study of dominance relationships for the skyline query. The basis of our study builds on the important properties of these relationships.

Definition 2 Rationality Property
A dominance relationship D satisfies the rationality property, if $\overline{D}(p, q)$ for any pair of p and q that $q[i] < p[i]$ for all $1 \leq i \leq d$.

The rationality property gives us the basic standard as to what is good and what is bad. A point cannot dominate another point if it is worse on all aspects.

Definition 3 Transitivity Property
A dominance relationship D satisfies the transitivity property, if $D(p, q)$ when there exists another point r such that $D(p, r)$ and $D(r, q)$.

This property is intuitive since preference usually embodies the transitivity property.

Given a d-dimensional vector $\alpha = (\alpha[1], \ldots, \alpha[d])$ and a point p in $\mathcal{S}$, we define the scaling operation as $\alpha p = (p[1]\alpha[1], \ldots, p[d]\alpha[d])$, where $\alpha[i] \geq 0$ for all i and $\alpha[j] > 0$ for some j. $\alpha[i]$ is the scaling factor of dimension i.

Definition 4 Property of Scaling Robustness
A dominance relationship D satisfies the property of scaling robustness if $D(\alpha p, \alpha q)$ when $D(p, q)$ for any valid α.

Similarly, given a d-dimensional constant vector $\beta = (\beta[1], \ldots, \beta[d])$ and a point p, we define the shifting operation as $p + \beta = (p[1] + \beta[1], \ldots, p[d] + \beta[d])$, where $\beta[i]$ is any real number for all i. $\beta[i]$ is said to be the shifting factor of dimension i.

Definition 5 Property of Shifting Robustness
A dominance relationship D satisfies the property of shifting robustness if $D(p + \beta, q + \beta)$ when $D(p, q)$ for any β.

The properties of scaling robustness and shifting robustness are important in real applications. This is because many real data sets contain incomparable dimensions. For example, in the case of hotel selection [4], the room price and distance to the beach are different in nature. The properties of scaling robustness and shifting robustness enable comparisons to be made among points with totally different dimensions, which is one of the most important advantages in the original skyline query.

With the concept of dominance relationship, we can provide a generic definition of skyline query as follows.

Problem 1 Skyline Query
Given a data set P and a dominance relationship D, locate the skyline $\mathbf{S}(P, D)$.

In the following, we define the concept of dominance class. Given a group of dominance relationships, we define the ordering property as follows.

Definition 6 Ordering Property
Given a fully sorted index set Θ, a set of dominance relationships indexed by Θ i.e., $\mathbf{D} = \{D_i | i \in \Theta\}$, $\mathbf{D}$ embodies the ordering property if for any $i \preceq j$, $D_i(p, q)$ must be valid if $D_j(p, q)$ is valid.

Lemma 1 *Given a dominance relationship class* $\mathbf{D}$ *indexed by* Θ *and a data set* P*, we have* $\mathbf{S}(P, D_i) \subseteq \mathbf{S}(P, D_j)$ *for* $i \preceq j$*.*

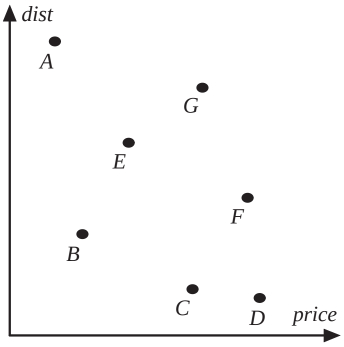


Fig. 1 Example of traditional dominance definition

Dominance	ε	Skyline	Dominated Set
D_1	0.5	$\{B\}$	$\{A, C, D, E, F, G\}$
D_2	0.2	$\{B, C\}$	$\{A, D, E, F, G\}$
D_3	0.1	$\{A, B, C\}$	$\{D, E, F, G\}$
D_4	0	$\{A, B, C, D\}$	$\{E, F, G\}$

Table 1 Example of Top-k Skyline Query

If Θ is a finite set, we say $\mathbf{D}$ is a finite ordered dominance class; otherwise, we say $\mathbf{D}$ is an infinite ordered dominance class. For example, Θ can be an integer set on $[1, n]$ or a real number interval $[a, b]$. In the rest of the section, we will assume $\mathbf{D}$ is finite and Θ contains all integers in $[1, n]$. The definitions can be easily extended to infinite cases. Given an ordered dominance class $\mathbf{D} = \{D_1, D_2, \ldots, D_n\}$, we can define the new problem in a way similar to the traditional top-k query in database systems.

Problem 2 Top-k Skyline Query
Given the specified parameter k and an ordered dominance class $\mathbf{D}$, find a dominance relationship that (1) $D_i \in \mathbf{D}$, that $|\mathbf{S}(P, D_i)| \geq k$ and $|\mathbf{S}(P, D_{i-1})| < k$, if $\mathbf{S}(P, D_n) \geq k$, or (2) D_n, if $\mathbf{S}(P, D_n) < k$.

In other words, the top-k skyline query tries to discover the dominance relationship in $\mathbf{D}$ with the minimal skyline cardinality but above k. If all of them in the dominance class lead to small skyline, the one with the maximal cardinality is returned instead. The top-k skyline query (Problem 1) is more attractive than the original skyline query (Problem 2) in many real applications since users can relate to results of manageable size more easily. Given the data shown in Figure 1, for example, if we construct an ordered dominance class based on ε-ADR [12], the corresponding skyline and dominated sets are as shown in Table 1. Therefore, the original skyline is $\{A, B, C, D\}$ while the top-2 skyline based on this ordered class will return $\{B, C\}$ as results.

For convenience and readability, we summarize the notations used in the rest of this paper in Table 2.

Notation	Description
$\mathcal{S}$	underlying numerical space
d	dimensionality of $\mathcal{S}$
p, q, r	points in $\mathcal{S}$
D	dominance relationship
$D(p, q)$	p dominates q by D
P	a data set in $\mathcal{S}$
$\mathbf{S}(P, D)$	skyline of P with D dominance
Θ	index set for dominance class
α, β	scaling and shifting vector
$\mathbf{D}$	a set of dominance relationships
TD	traditional dominance relationship
CD_γ	cone dominance with parameter γ
$E(p, q)$	Euclidean distance between p and q
f,g	mappings from a distribution to a point
MD_λ	mapping dominance with parameter λ

Table 2 Table of Notations

2.2 Related Work

We next review the existing dominance relationship definitions that are relevant to the skyline query. While the definition of skyline query was previously known as maximal vectors problem in algorithm community, the earlier studies only focuses on the computational complexity [3,2]. In the literature of database system, instead, I/O cost becomes the bottleneck as the data grows beyond the capacity of the memory. In this paper, we emphasize the skyline query processing algorithms capable on large database. Specifically, we consider five important aspects of interest. For each dominance relationship definition, we first consider whether its query result is *deterministic*, which indicates that re-executions (or different evaluations) of a given instance of a query type always produce the same subset as the query result. In addition, we look into whether each dominance relationship conforms to the four properties of: rationality, transitivity, scaling robustness and shifting robustness. All existing dominance relationship definitions and respective properties are listed in Table 3.

We include the top-k query as it can be regarded as a special case of the skyline query with multiple dimensions for comparison degenerating into only one. A top-k query yields deterministic results, and it conforms to the rationality and transitivity properties. Whether it exhibits the property of scaling robustness or shifting robustness is dependent on the concrete aggregation functions that are used for ranking purposes.

A traditional skyline query [1,4,22,13,19,10,23,14] yields deterministic results. It also conforms to all the four properties described in the previous subsection. The detailed proofs will be presented in Section 3.1. Another instance employing the traditional skyline is the recent k most representative skyline operator [16]. It

Dominance Definition	Determin.	Ration.	Transi.	Scaling Robust	Shifting Robust
Top-k	+	+	+	-	-
Traditional dominance [4,22,13,19,16]	+	+	+	+	+
Partially-ordered domain dominance [5]	+	+	+	N.A.	N.A.
ε-ADR dominance [12,9]	-	-	-	+	+
k-dominance [6]	+	+	-	+	+

Table 3 Revisiting existing dominance relationships

introduces a constraint on the number of skyline points to be returned, and selects from the traditional skyline those points that maximize the total number of dominated points.

The skyline query on partially ordered domains [5] and categorical domain [20,17] are special cases of the traditional skyline query, but the properties of scaling robustness or shifting robustness are not applicable because partially ordered domains do not support scaling or shifting operations.

In the approximate dominating representatives problem [12,9], each point is boosted (if larger values are preferred) by ε in all dimensions when being compared with other points. We call the underlying dominance relationship *ε-dominance*, and the corresponding skyline query *ε-ADR skyline query.* The ε-ADR skyline query does not have deterministic results, and *ε-dominance* violates both rationality and transitivity properties. However, *ε-dominance* conforms to the properties of scaling robustness and shifting robustness.

The k-dominant skyline [6] problem also alters the traditional dominance definition. Given a d-dimensional data set, a point p is said to k-dominate another point q if there exists a k-dimensional subspace ($k \leq d$) within which p traditionally (fully) dominates q. The k-dominant skyline query yields deterministic results, but it does not conform to the transitivity property [6].

3 Analysis on Relationship and Properties

In this section, we analyze the connections between dominance relationships and desired properties, and show how relaxations on some of the properties reshape the dominance relationships.

3.1 Traditional Dominance Relationship

We first consider the traditional dominance relationship (TD). In the following, we use $TD(p, q)$ to represent that p dominates q based on the definition of the traditional dominance relationship.

Theorem 1 *TD satisfies the properties of rationality, transitivity, scaling robustness and shifting robustness.*

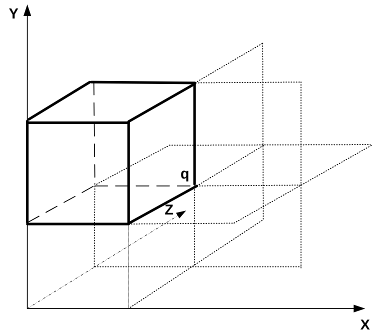


Fig. 2 Example of octants in three-dimensional space

Proof TD must satisfy the rationality property since a point p dominates another point q only when p is not worse than q on all dimensions.

As pointed out in [4], TD satisfies the transitivity property.

Given that $TD(p, q)$, we have $p[i] \leq q[i]$ for any $1 \leq i \leq d$, and there is at least one j that $p[j] < q[j]$. Consider the relationship between αp and αq. Since $\alpha[i] > 0$, $\alpha[i]p[i] \leq \alpha[i]q[i]$ for any $1 \leq i \leq d$ and $\alpha[j]p[j] < \alpha[j]q[j]$. This shows that $TD(\alpha p, \alpha q)$ satisfy the property of scaling robustness.

The proof for shifting robustness is similar to that of scaling robustness. Thus, we omit the detail here.

In the rest of the section, we will show that TD is the only binary relationship satisfying all the four properties. The proof begins with several lemmas.

Definition 7 Hyper-Octant
Given a point p in d-dimensional space $\mathcal{S}$, the point can divide the whole space into 2^d hyper-octants. For any two points q and r in the same octant, we have $(q[i] - p[i])(r[i] - p[i]) \geq 0$ for any $1 \leq i \leq d$.

By the definition, we know that for any two points in the same octant, both of them are of larger (smaller) values than p's value on any dimension i. In Figure 2, we present an example in three-dimensional space, where the cube with thick edges is the octant containing points of smaller values than q on the X and Z axes but of larger values than q on the Y axis.

Lemma 2 *Given a dominance relationship D that satisfies the properties of scaling robustness and shifting* *robustness, a point p and the octants induced by p, if $D(p,q)$ for some q in an octant X, $D(p,r)$ for any $r \in X - \{p\}$.*

Proof Consider any r in X, we construct a vector $\delta = \{\delta[1], \ldots, \delta[d]\}$ such that $\delta[i] = (r[i] - p[i])/(q[i] - p[i])$. By the property of octant, we are sure $\delta[i] \geq 0$ for all i. Then, we know that $\delta p + (1-\delta)p = p$ and $\delta q + (1-\delta)p = r$. By applying the property of scaling robustness, we have $D(\delta p, \delta q)$. By applying also the property of shifting robustness, we have $D(\delta p + (1-\delta)p, \delta q + (1-\delta)p)$, which directly leads to the conclusion that $D(p,r)$.

The last lemma implies that any dominance relationship D exhibiting scaling robustness and shifting robustness has some ability to expand from a single dominated point to the whole hyper-octant.

Given a dominance relationship D, if $D(p,q)$ and $D(p,r)$, we say D is convex if $D(p, \gamma q + (1-\gamma)r)$ for any constant real value $0 \leq \gamma \leq 1$.

Lemma 3 *Given a dominance relationship D satisfying the properties of scaling robustness, shifting robustness and transitivity, D must be convex.*

Proof Given p, q and r that $D(p,q)$ and $D(p,r)$, by the properties of scaling robustness and shifting robustness, we have $D(p, p+\gamma(q-p))$ and $D(p+\gamma(q-r), p+\gamma(q-p)+(1-\gamma)(r-p))$. By the transitivity property, we have $D(p, p+\gamma(q-p)+(1-\gamma)(r-p))$. Since $p+\gamma(q-p)+(1-\gamma)(r-p) = \gamma q + (1-\gamma)r$, we reach the convexity condition by $D(p, \gamma q + (1-\gamma)r)$.

Theorem 2 *If D is a dominance relationship satisfying all the properties proposed in the last section, D must be equal to TD in some subspace $\mathcal{S}' \subseteq \mathcal{S}$.*

Proof Given any point p in the space $\mathcal{S}$, by the rationality property, p dominates the hyper-octant that contains points worse than p on all dimensions. If this is the only hyper-octant dominated by p, it is TD in $\mathcal{S}$. If p dominates another hyper-octant with points better than p on some dimensions in $\mathcal{S}'' \subseteq \mathcal{S}$, by the convexity property, p dominates all points worse than p on dimensions in $\mathcal{S}' = \mathcal{S} - \mathcal{S}''$.

By Theorem 2, we have proved that the traditional dominance relationship is the only dominance relationship satisfying the properties of rationality, transitivity, scaling robustness and shifting robustness.

3.2 Relaxation of Properties

Although Theorem 2 shows that TD is the only option for a dominance relationship to satisfy all the four properties, we can obtain some other relationships if we are able to relax some of the properties. In this part of the section, we investigate along this direction by relaxing one of the following properties: transitivity, scaling robustness and shifting robustness. We will also discuss the scenarios in real applications when the relaxations are reasonable. Note that the rationality property cannot be relaxed since doing so can lead to unreasonable results.

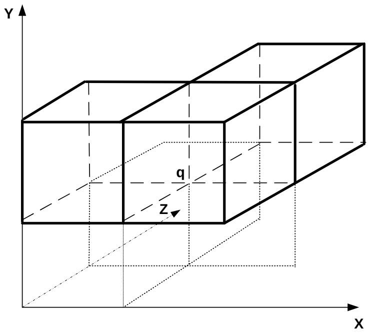


Fig. 3 Example of dominance region after relaxing transitivity property

3.2.1 Relaxing the Property of Transitivity

By relaxing the transitivity property, a point p may not be able to dominate another point r, even if it dominates some point q that dominates r. However, Lemma 2 still applies since we have not relaxed the properties of scaling robustness and shifting robustness. Therefore, the dominance region of a point p will occupy arbitrary hyper-octants divided by p in the space. By the rationality property, p cannot dominate the hyper-octant with points better than p on all dimensions. Therefore, the dominance region of p can be non-convex as shown in the example of Figure 3.

In this example, the dominance relationship definitely does not follow the transitivity property since the dominance region is not convex. We call this type of dominance relationship *Octant Dominance*, or *OD* in short.

We use Φ to denote the set of all OD relationships satisfying all the properties except transitivity. Since there are $2^d - 1$ hyper-octants to choose for the dominance relationship, there are 2^{2^d-1} dominance relationships in Φ. Unfortunately, there does not exist a total order on these 2^{2^d-1} relationships. Therefore, Φ cannot be an ordered class of dominance relationships, which prohibits the top-k skyline query directly over Φ. However, it is possible to find some ordered subset of Φ. It is not difficult to verify that the k-dominance proposed in [6] is a subset of Φ and is an ordered dominance class with d dominance relationships.

The property of transitivity is important in some applications, especially when consistent result is expected in recommendation system. However, with the increase of dimensionality, the property of transitivity incurs cost on the decreasing meaning of selective result, since the points become hard to dominate. Therefore, the relaxation of transitivity is a natural option for data in high dimensional space. More details on this can be referred to [6]

3.2.2 Relaxing the Property of Scaling Robustness

If we relax the property of scaling robustness, we can have many different variants of the traditional dominance relationship since it will be unnecessary to consider the scaling factor any more. The dominance region of a point p can even be discrete. For example, we can define a dominance relationship, D, which only allows a point p to dominate another point q if $q[i]-p[i] \in \mathbf{Z}^+$ for all i. It is not hard to verify that such dominance relationship must follow all properties except for scaling robustness.

Compared to the limited number of OD relationships introduced above, there are infinite dominance relationships satisfying all properties except scaling robustness. Moreover, in these relationships, we can find some ordered classes of infinite size. For example, an infinite ordered class can be defined as follows. Given the index set on all positive integers, $\Theta = \{1, 2, \ldots, i, \ldots\}$, a dominance relationship D_i is defined as $D_i(p, q)$ if $q[k]-p[k]$ is divisible by 2^{i-1} for all $1 \leq k \leq d$. Each D_i, as defined above, must be a valid dominance relationship, satisfying all properties except for scaling robustness. Each pair of D_i and D_j $(i < j)$ must also satisfy the requirement of Definition 6. Thus, these dominance relationships form an infinite ordered class and top-k skyline query can be issued on them.

The property of scaling robustness is helpful when the dimensions are in totally different domains. In the classic example of hotel selection [4], with two dimensions on price and distance to beach, scaling robustness plays an important role on deriving meaningful result. However, this property is no longer important if the dimensions are recorded with the same measurement. In movie rating data set with every user as a single dimension, such as Netflix[1], the score on each dimension can only be some integer between 1 and 5. In such cases, scaling robustness can be relaxed without affecting the meaningfulness of the skyline query result.

3.2.3 Relaxing the Property of Shifting Robustness

We can easily extend the analysis on relaxing scaling robustness to the new case where shifting robustness is relaxed instead. The key observation is that the property of scaling robustness is equivalent to the property of shifting robustness in the log-scale space $\log \mathcal{S}$, in which every point p is transformed to another point $p' = (\log p[1], \ldots, \log p[d])$ with the assumption that $p[i] > 0$ for all i.

Based on this observation, there is a mapping between the set of all relationships violating only scaling robustness and the set of relationships violating only shifting robustness. Given a relationship D satisfying all properties except for scaling robustness, we can construct a new relationship D' that $D'(p, q)$ if $D(\log p, \log q)$. D' must satisfy scaling robustness, since $D(\log p\alpha, \log q\alpha) = D(\log p + \log \alpha, \log q + \log \alpha)$ with any scaling vector α. Since the reverse mapping can be constructed similarly, we can prove that the number of relationships violating only scaling robustness must be equal to the number of relationships violating only shifting robustness.

Shifting robustness is an valuable property if some of the dimensions are error sensitive or subjective. Considering the example of movie rating data set, some of the raters are prone to give higher scores to all movies while some others always give low scores. Shifting robustness is able to remove the influence of these factors, rendering consistent results. In some applications with relatively stable and pointive values on all dimensions, such as climate data collected over sensor network, the necessity of shifting robustness does not exist any more, allowing relaxation on this property.

4 General Algorithm Design

In this section, we discuss how existing skyline algorithms can be modified to answer skyline query as well as top-k skyline query, based on the properties of the underlying dominance relationship used. Note that all the algorithm designs here assume the complete information of every point on all dimensions. The extensions to data with missing values will be discussed in later sections.

4.1 General Algorithms for Skyline Queries

There are four algorithm discussed below, including *Block Nested Loop*, *Sort Filter Skyline*, *Two Scan Algorithm* and *Branch-and-Bound Skyline.*

[1] http://www.netflix.com

Algorithm 1 General Block Nested Loop (data set P, dominance relationship D)

```
clear the skyline buffer $S$
for each point $p$ in $P$ do
    for each point $q$ in $S$ do
        $isSky$=TRUE
        if $D(q,p)$ then
            $isSky$=FALSE
        end if
        if $D(p,q)$ then
            remove $q$ from $S$
        end if
    end for
    if $isSky$ then
        move $p$ into $S$
    end if
end for
return $S$
```

4.1.1 Block Nested Loop

In Algorithm 1, we list the details of the *General Block Nested Loop* algorithm, which was first proposed in [4]. In this algorithm, a buffer S is maintained for the current skyline on the data seen so far. If a new point p dominates some current skyline point q, q is removed from S. If no point in S dominates p, p is moved into S. This simple algorithm is of the widest applicability range, as stated in the following theorem.

Theorem 3 *Algorithm 1 can be applied on any dominance relationship satisfying the transitivity property.*

Proof By Algorithm 1, S must contain all the true skyline points $\mathbf{S}$, since no other points dominates them by definition. On the other hand, assume S contains a false skyline point p, and p is dominated by another point q. If $q \in \mathbf{S}$, q must be eliminated when the algorithm visits p or q. If $q \notin \mathbf{S}$, q must be dominated by another point r. By the transitivity property r, r must dominate p. Following the logic, we can always find a skyline point dominating p, which contradicts the assumption.

If the underlying dominance relationship does not have the property of transitivity, BNL will fail to retrieve the correct skyline set, since some dominance pairs can be missed when some points are dropped from the temporary skyline buffer S.

4.1.2 Sort Filter Skyline

Next, we consider the *General Sort Filter Skyline* algorithm, which was first proposed in [8]. In this algorithm, a pre-sorting topologically on all dimensions is conducted before the block nested loop algorithm is applied. The benefit of pre-sorting is that any point moved into the skyline buffer must be a true skyline point at the end, which saves time spent on pruning false skyline points.

Algorithm 2 General Sort Filter Skyline (data set P, dominance relationship D)

```
1: sort the data set $P$ on the sum of all dimensions for every
   point.
2: clear the skyline buffer $S$
3: for each point $p$ in $P$ do
4:     for each point $q$ in $S$ do
5:         if $D(q,p)$ then
6:             go to line (3)
7:         end if
8:     end for
9:     move $p$ into $S$
10: end for
11: return $S$
```

The applicability range of Algorithm 2 is smaller than that of Algorithm 1.

Theorem 4 *Algorithm 2 can be applied on any dominance relationship satisfying the rationality and transitivity properties.*

Proof This algorithm must rely on the transitivity property since all points are pruned by only those skyline points. On the other hand, sorting can avoid a current skyline point being dominated only when the dominance relationship follows the rationality and transitivity properties.

If any of the two properties does not hold, SFS algorithm cannot output the correct result, because 1) the sorting of the points implicitly assumes the property of rationality when points with smaller values are preferred, and 2) the skyline buffer S may not contain enough points to dominate non-skyline points if transitivity is violated. Based on this observation, the requirements on SFS is tight.

4.1.3 Two Scan Algorithms

The third algorithm in our consideration is the *General Two Scan Algorithm*, which was first proposed in [6]. This algorithm consists of two scans. In the first scan, SFS is run on the data set to obtain a candidate set S. In the second scan, points in S are compared with all points in P to eliminate false skyline points. The details are summarized in Algorithm 3. This algorithm returns correct results even on a dominance relationship without the transitivity property.

Theorem 5 *Algorithm 3 can be applied on any dominance relationship satisfying the rationality property.*

Algorithm 3 General Two Scan Algorithm (data set P, dominance relationship D)

```
1: get candidate set $S$ by running SFS
2: for each point $p$ in $P$ do
3:    for each point $q$ in $S$ do
4:       if $D(p,q)$ then
5:          remove $q$ from $S$
6:       end if
7:    end for
8: end for
9: return $S$
```

Proof Compared against the SFS algorithm, Algorithm 3 employs a second scan. The second scan enables the algorithm to find skylines even when the dominance relationship does not follow the transitivity property.

The tightness of the requirement on TSA comes from the simple observation that the employment of SFS leads to the sorting of points depending on the property of rationality. As is discussed in SFS algorithm, rationality is underlying reason on the validity of the sorting process.

4.1.4 Branch-and-Bound Skyline

Finally, we look at the complicated *General BBS* algorithm, which was first proposed in [19]. In this algorithm, there exists an index structure, such as the R-Tree, where every point can be found efficiently. Each node in the index has an MBR (minimum bounding rectangle), which is the bounding range of all the points stored in its descendant nodes.

To facilitate the adoption of indexing tree structure, we propose a new concept, *Common Dominating Position*, over the Minimum Bounding Rectangles. Intuitively, common dominating position can be abstracted as some location in the space, which is able to dominate any possible point in the MBR. The following lemma implies the existence of common dominating position for any MBR.

Lemma 4 *If the dominance relationship satisfies the properties of rationality and transitivity, common dominating position always exists for any MBR.*

Proof We prove this by construction. Given two points p_1 and p_2, there is definitely at least one common dominating position for $\{p_1, p_2\}$ because of the property of rationality. Assuming there is a set $P = \{p_1, p_2, \ldots, p_n\}$ containing points stored in some MBR M, the common dominating position for M can be constructed in $n-1$ steps. In the first step, the common dominating position p'_2 is discovered for $\{p_1, p_2\}$. In step i $(1 < i \leq n-1)$, a new position p'_{i+1} is found as common dominating position for $\{p'_i, p_{i+1}\}$. Because p'_{i+1} dominates p'_i, it also dominates any p_j $(j \leq i)$ due to the property of transitivity. Therefore, the final position p'_n must be common dominating position for the whole set P, which completes the proof of the lemma.

Algorithm 4 General BBS Algorithm (data set P, dominance relationship D, index tree T)

```
1: clear a heap $H$ and skyline buffer $S$
2: put root node $N$ of $T$ into $H$
3: while $H$ is not empty do
4:    pick a node $n$ from $H$ with the minimum possible distance
      to the space origin, and remove $n$ from $H$.
5:    Set $M$ as the MBR on node $n$
6:    for each point $p$ in $S$ do
7:       if $DP(M)$ is dominated by $p$ then
8:          go to line (3)
9:       end if
10:   end for
11:   if $n$ is a single point then
12:      move $n$ into $S$
13:   else
14:      retrieve all children of $n$ in $T$, and insert them into $H$
15:   end if
16: end while
17: return $S$
```

In the rest of the paper, we use $DP(M)$ to denote the common dominating position for some MBR M. The computation of the positions will be covered when the specific dominance relationship is introduced. Generally speaking, the applicability of BBS algorithm can be summarized by the following theorem.

Theorem 6 *Algorithm 4 can be applied on any dominance relationship satisfying the rationality and transitivity properties.*

Proof Considering a node n in the indexing tree, if there is at least one skyline point q in node n, n can never been removed by BBS algorithm because of the properties of rationality and transitivity. Otherwise, some point p in the buffer S is able to dominate the MBR of n, contradicting to the fact that q is a skyline point. Therefore, every skyline point must be included in the buffer S. On the other hand, S can never contain any false positive skyline point q, because there is at least one dominating point in S for q based on the property of transitivity. As a summary, the output of BBS must be the correct skyline set.

To discuss the tightness of BBS algorithm, we first look at the property of rationality. If the dominance relationship violates rationality, there is no longer guarantee on the correctness of step (4), since the point selected may not be a skyline point. Secondly, when there is no property of transitivity, the skyline buffer S does not have full capacity to prune all non-skyline points, for similar reason for BNL and SFS. This leads to the conclusion that either property is removable from the requirements for BBS algorithm.

We summarize the necessary conditions of the four algorithms in this part in Table 4.

Algo Name	Rationality	Transitivity
BNL		√
SFS	√	√
TSA	√	
BBS	√	√

Table 4 Necessary Properties of Algorithms 1,2,3 and 4

4.2 General Algorithms for Top-k Skyline Queries

To support efficient computation of top-k skyline queries, we summarize two general methods which are based on the general algorithms proposed for skyline queries.

In this subsection, we also assume $\mathbf{D}$ is finite, where there are n dominance relationships $\mathbf{D} = \{D_1, D_2, \ldots, D_n\}$ with index set Θ on $[1, n]$. To handle the infinite dominance class with a dominance index Θ on real interval $[a, b]$ with the same algorithm, we can use a minimum gap ϵ to discretize the class, i.e., constructing $n = \lfloor \frac{a-b}{\epsilon} \rfloor$ dominance relationships where D_i $(1 \leq i \leq n)$ equal to the original relationship with index $a + \epsilon(i-1)$.

4.2.1 Binary Search

The first general algorithm in our consideration is a binary search on the index of dominance relationships $D_i \in \mathbf{D}$. It is so general that any algorithm proposed in Section 4.1 can be directly used to find the dominance relationship D_i, satisfying the condition in Problem 2, with the smallest index i in $\mathbf{D}$.

The correctness of Algorithm 5 is straightforward. Since the skyline monotonically shrinks with the decrease of the dominance relationship index i, binary search can definitely reach the smallest D_i with exactly k results (or smallest one above k). However, in some cases, even the weakest dominance relationship in the class cannot return a skyline of size no smaller than k. The simplest example is the construction of a dominance class with only one dominance relationship. In such cases, the algorithm can only return the maximal skyline calculated based on the weakest dominance relationship in the dominance class. Generally speaking, different D_is in different iterations in the binary search do not correlate. Therefore, Algorithm 5 computes each $\mathbf{S}(P, D_i)$ by calling the most appropriate algorithm as presented in Section 4.1.

Algorithm 5 General Binary Search Algorithm (data set P, dominance class $\mathbf{D}$, skyline algorithm A, skyline size k)

```
set $l = 1$ and $u = n$
compute $\mathbf{S}(P, D_u)$ by running a skyline query algorithm
if $|\mathbf{S}(P, D_u)| \leq k$ then
    return $\mathbf{S}(P, D_u)$
end if
$i = \lfloor (l+u)/2 \rfloor$
while $|\mathbf{S}(P, D_i)| \neq k$ and $l \neq u$ do
    if $|\mathbf{S}(P, D_i)| < k$ then
        $l = i$
    else
        $u = i$
    end if
    $i = \lfloor (l+u)/2 \rfloor$
end while
return $\mathbf{S}(P, D_i)$
```

4.2.2 Progressive Algorithm

The second general method is to modify any progressive skyline algorithm to return top-k skyline query results. A progressive skyline algorithm keeps all current skyline points in a buffer, which we use to answer top-k skyline queries. The general progressive algorithm is as shown in Algorithm 6. The algorithm starts with the input largest dominance index $i = n$ doing a progressive skyline search with dominance relationship D_i being decreased when a full buffer is encountered.

When the buffer contains $k + 1$ points, at least one point will be removed from the buffer. To guarantee the correctness of the current top-k result, we need to find the point easiest to dominate than any other points in S. This is implemented by Algorithm 7. Every pair of points in the buffer is examined, and the point p dominated by some q with the largest D_i will be exactly the one wanted. The efficiency of the algorithm can be further improved if we store the previous pair-wise computation result, since there is only one new point after Algorithm 7 is applied once.

The correctness of the algorithm depends on two conditions. First, the underlying skyline algorithm A must be progressive. Only with progressive algorithms, we can make sure points in the buffer are definitely top-k skyline query result after the algorithm completes its run. Second, it must be easy to find the largest $D_i \in \mathbf{D}$ for q to dominate p. Fortunately, all dominance relationships and their relaxed variants listed in this paper meet this condition. Therefore, we can focus on the underlying skyline algorithm in analyzing the applicability of the progressive scheme.

In Table 5, we list the analysis of the possible combinations of general skyline algorithms and the general top-k skyline algorithms. We can see that all the four

Algorithm 6 General Progressive Algorithm (data set P, dominance relationship class $\mathbf{D}$, progressive skyline algorithm A, expected skyline size k)

```
construct a skyline buffer $S$ of size $k+1$
set $i = n$ and use $D_i$ as the current dominance relationship
run $A$ on $P$ with $D_i$, run Skyline Pruning when the $S$ is full
return $S$
```

Algorithm 7 Skyline Pruning (Skyline Buffer S, current relationship D_i, dominance class $\mathbf{D}$)

```
set $\theta = 1$
for each point $p \in S$ do
  for each point $q \in S$ and $p \neq q$ do
    compute the largest $j$ that $D_j(q, p)$
    if $j \leq \theta$ then
      $\theta = j$ and mark $p$
    end if
  end for
end for
set $D_\theta$ as new relationship and remove the last marked point from $S$
```

algorithms can be directly used in binary search while only SFS and BBS can be used in the progressive algorithm.

Algo Name	Binary Search	Progressive
BNL	√	
SFS	√	√
TSA	√	
BBS	√	√

Table 5 Combinations of Algorithms and Top-k Skyline

5 Cone Dominance with Arbitrary Resolution

A common problem with skyline query is its uncontrollable result size. Although there are studies on skyline variants to reduce the result size when conventional skyline is too large, there does not exist any systematic method which can adaptively output result with specified size, no mater whether conventional skyline is over-sized or under-sized.

In this section, we apply our framework on the design of some generalized dominance relationships. We propose a new class of dominance relationships, namely *Cone Dominance*, with arbitrary selection resolution if the parameters are set appropriately. In the following, we use $E(p, q)$ to denote the Euclidean distance between p and q in the space $\mathcal{S}$.

Definition 8 Cone Dominance
In Cone Dominance, given the bias parameter γ, p dominates q if (1) $\sum p[i] < \sum q[i]$, and (2) $p[i] \leq q[i] + E(p, q)\gamma$ for all i, while there is at least one dimension j that $p[j] < q[j]) + E(p, q)\gamma$. We use $CD_\gamma(p, q)$ to denote such a relationship.

In Figure 4, an example is shown in two-dimensional space. Given a point p in the space, the dominance region can be represented by a cone region with the top point at p. In the figure, for example, there are three pairs of boundary lines of the dominance region, represented by dashed, normal and thick lines respectively. These dominance regions are achieved by assigning negative, zero and positive γs respectively. When $\gamma = 0$, as the pair of normal lines show, cone dominance degenerates to traditional dominance TD. The following lemma states the valid index set on γ.

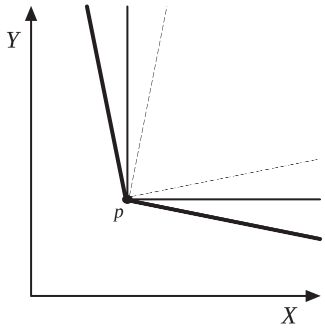


Fig. 4 Example of cone dominance relationships

Lemma 5 *CD_γ satisfies all properties except scaling robustness, if γ is chosen from index set*

$$\Theta = \left[-\sqrt{1/d}, \sqrt{(d-1)/d}\right]$$

Proof First, CD_γ is null if γ is smaller than $-\sqrt{1/d}$. Given any two point p and q in the space, if $\gamma < -\sqrt{1/d}$, $E^2(p, q) = \sum(p[i] - q[i])^2 < \sum E^2(p, q)/d = E^2(p, q)$. Thus, we cannot find any dominance pair in the space, making the dominance relationship useless.

Assuming there two points p, q holding the dominance relationship $CD_\gamma(p, q)$ and $\gamma > \sqrt{(d-1)/d}$, there is at least one dimension i satisfying that $p[i] > q[i] + E(p, q)\sqrt{(d-1)/d}$. Therefore, we have $\sum p[i] - \sum q[i] > E(p, q)\sqrt{(d-1)/d} - (d-1)E(p, q)\sqrt{1/d(d-1)} = 0$, contradicting to the first condition in the definition of cone dominance. Thus, CD with $\gamma > \sqrt{(d-1)/d}$ must be an empty relationship.

The shifting robustness property on cone dominance is straightforward since the distance between two points does not change after shifting operations. Rationality property is proved directly by the definition.

Considering transitivity property, given three points p, q, r, if $CD_\gamma(p,q)$ and $CD_\gamma(q,r)$, we have $\sum p[i] < \sum q[i]$ and $\sum q[i] < \sum r[i]$. So, $\sum p[i] < \sum r[i]$, which proves the first condition of cone dominance. On the other hand, since $p[i] \leq q[i] + E(p,q)\gamma$ and $q[i] \leq r[i] + E(q,r)\gamma$, $p[i] \leq r[i] + E(p,q)\gamma + E(q,r)\gamma$. By the triangle inequality of Euclidean distance, we have $p[i] \leq r[i] + E(p,r)\gamma$.

Based on the last lemma, the index set Θ can be defined as all real numbers in the interval

$$\left[-\sqrt{1/d}, \sqrt{(d-1)/d}\right]$$

Lemma 6 *If $CD_\gamma(p,q)$, $CD_{\gamma'}(p,q)$ for all $\gamma' \geq \gamma$.*

Proof If $CD_\gamma(p,q)$, $p[i] \leq q[i] + E(p,q)\gamma$. When $\gamma' \geq \gamma$, $p[i] \leq q[i] + E(p,q)\gamma'$. Since the first condition does not change when changing γ to γ', $CD_{\gamma'}(p,q)$ is valid in all cases.

Therefore, top-k skyline query can be issued on cone dominance class indexed by $\Theta = \left[-\sqrt{1/d}, \sqrt{(d-1)/d}\right]$. Note that the generic full order, $\preceq$, in Definition 6 is instantiated as $\gamma_1 \geq \gamma_2$ in cone dominance relationship.

Intuitively, we can interpret cone dominance as follows. If γ is negative, a point p dominates another point q only when every dimension contributes enough to the difference between p and q. In other words, thorough advantage to some extent is expected on all dimensions. On the other hand, if γ is positive, the definition allows p to dominate q even when p does not show too much disadvantage on all dimensions compared to the total difference between them. Based on the adjustment on the parameter γ, cone dominance can adaptively choose the resolution on the query, leading to controllable size of query output.

By the mapping method derived in Section 3.2.3, *Log-Scale Cone Dominance*(LCD_γ) can be defined as $LCD_\gamma(p,q)$ if and only if $CD_\gamma(\log p, \log q)$. Log-Scale Cone Dominance is thus only violating shifting-robustness but satisfying all other properties.

Since cone dominance and log-scale cone dominance follow both rationality and transitivity property, all of the algorithms presented in Section 4 can be used to answer skyline query and top-k skyline query on them.

As presented in Section 4, the adoption of indexing tree structure depends on the existence of common dominating position for MBR M. Here, we discuss the details of the computation on the common dominating position with respect to cone dominance relationships. The extension to log-scale cone dominance is straightforward.

Algorithm 8 Find Common Dominating Position (MBR M, parameter γ)

```
1: if $\gamma \geq 0$ then
2:     Return $(M.l[1], M.l[2], \ldots, M.l[d])$
3: end if
4: Find $j$ with minimal $M.u[j] - M.l[j]$
5: Find the minimal $\alpha$ that $(M.u[j] - M.l[j] + \alpha)^2 \leq (\sum_i (M.u[i] - M.l[i] + \alpha)^2)\gamma^2$
6: Return $(M.l[1] - \alpha, \ldots, M.l[d] - \alpha)$
```

For a MBR M, the lower and upper boundaries of M on dimension i are represented by $M.l[i]$ and $M.u[i]$ respectively. In Algorithm 8, we present some method computing the common dominating position, with input MBR M and the cone dominance parameter γ.

In Algorithm 8, if γ is not negative, the method simply returns the left-bottom corner of the MBR, since the corner position is enough to dominate all points in it. It can be easily interpreted by the example in Figure 4. When γ is negative, the problem becomes more complicated because the corner is not capable enough. To discover a stronger position, the algorithm searches on the line crossing the the corner position, i.e. $p_\alpha = (M.l[1] - \alpha, \ldots, M.l[d] - \alpha)$ with α as a positive parameter. When the condition on line (4) of Algorithm 8 is satisfied, it is easy to verify that the whole MBR is covered by the dominance region of the position p_α. Since condition on line (4) is actually a quadratic inequality on variable α, a simple solution can be derived in constant time. Therefore, p_α can be easily calculated and returned as the common dominating position for the MBR M.

6 Dominance Relationships on Data with Missing Values

Data tuples with missing values are a common kind of data found in many databases. In a movie rating data set, for example, if we consider every rater as a dimension, every dimension has only very few entries in which a value is present, while others are filled with NULL values. This is because every user can possibly watch some small fraction of all movies. However, such uncertainties can lead to difficulty in using the skyline query on high-quality movie selection since it is difficult to compare a pair of movies when they have been watched and rated by totally different users.

In this section, we extend our analysis from certain data to data with missing values. We first propose a mapping-based dominance relationship definition for data with missing values. Then, we study the properties of rationality, scaling robustness, shifting robustness and transitivity for the mapping-based dominance relationship. We also discuss the combinations of mapping dominance with other relaxed dominance relationships presented in previous section, as well as the algorithm applicabilities on the computation of both skyline and top-k skyline queries with mapping dominance.

Note that mapping dominance is not the only option on possible dominance relationships defined on incomplete data tuples. The more important implication provided in this section is that well-defined dominance relationships can be easily incorporated into our general skyline framework, which facilitates easier analysis and querying algorithm design.

6.1 Mapping Dominance

Given a space $\mathcal{S}$ defined by a set of d dimensions $\{1, 2, \ldots, d\}$, without loss of generality, we suppose that missing values appear on the first k dimensions, i.e., from dimension 1 to dimension k. For a dimension i ($1 \leq i \leq k$), we use $\phi_i(x_i)$ to denote the probability distribution function (or pdf) of the value x_i on dimension i. Such a distribution can be retrieved by observing all the non-missing values on dimension i.

Now consider a point $p(x_1, x_2, \ldots, x_d)$ with missing values on some dimensions. A generic mapping h is defined as a function converting incomplete point p to a fixed point $p'(x'_1, x'_2, \ldots, x'_d)$ in $\mathcal{S}$, i.e., $h(p(x_1, x_2, \ldots, x_d)) = p'(x'_1, x'_2, \ldots, x'_d)$, where

$$x'_i = \begin{cases} F_i(\phi_i(x_i)), & \text{if } x_i \text{ is NULL} \\ x_i, & \text{otherwise} \end{cases}$$

The generic mapping h retains each concrete value of p, and estimates a concrete value for each missing value based on the known pdf on that relevant dimension. The estimation is accomplished by a function F_i, which takes a pdf as input and outputs a fixed value.

To facilitate comparison of an arbitrary pair of points p and q with possible missing values in a generic way, we define two mappings, f and g, as instances of the general mapping h aforementioned. Based on these two mappings, we define a special dominance relationship called *Mapping Dominance*, or MD, as follows:

$$MD = \{(p, q) \mid TD_{\mathcal{S}}(f(p), g(q))\},$$

where $TD_{\mathcal{S}}$ is the traditional dominance relationship over $\mathcal{S}$. We call the mapping f *dominating position mapping* and g *dominated position mapping*, in the sense that f tends to leave point p dominating others while g tends to leave point q being dominated by others.

Now we consider the properties of the mapping dominance (MD) relationship that we have proposed for data with missing values. The following theorems state the conditions that enable the MD relationship to satisfy the properties of scaling robustness, shifting robustness and transitivity.

Lemma 7 *Given any point p with possible missing values, the two mappings f and g that satisfy $\alpha f(p) = f(\alpha p)$ and $\alpha g(p) = g(\alpha p)$ for any scaling factors α, then MD satisfies the property of scaling robustness.*

Proof Assume two points p, q and $MD(p, q)$. If $\alpha f(p) = f(\alpha p)$ and $\alpha g(q) = g(\alpha q)$, $(\alpha p, \alpha q)$ must be in MD since $(f(p), g(q))$ is in TD and TD satisfies scaling robustness for any scaling factor α.

Lemma 8 *If for any point p, the mappings f and g satisfy that $f(p) + \beta = f(p + \beta)$ and $g(p) + \beta = g(p + \beta)$ for any shifting factors β, then MD satisfies the property of shifting robustness.*

Proof Assume two points p, q and $MD(p, q)$. If $f(p) + \beta = f(p + \beta)$ and $g(q) = g(q + \beta)$, $(p + \beta, q + \beta)$ must be in MD, since $f(p), g(q)$ is in TD and TD is shifting robust.

Lemma 9 *If for any point p with possible missing values, $(g(p), f(p)) \in TD$ or $g(p) = f(p)$, MD embodies the transitivity property.*

Proof Assume three point p, q, r that $MD(p, q)$ and $MD(q, r)$. Since $TD(f(p), g(q))$ and $TD(g(q), f(q))$, we have $TD(f(p), f(q))$ by the transitivity property of TD. Since $TD(f(q), g(r))$, $TD(f(p), g(r))$ and $MD(p, r)$ by using the transitivity property again.

To satisfy the conditions of the lemmas above, we propose a new type of dominance relationships called MD_λ. If O_i is the set of observed values on non-NULL entries on dimension i, we can construct the observation sets $O_1, O_2, \ldots, O_k$ by visiting the data set once. The dominating position mapping $f_{\lambda,i}$ for missing values on dimension i is thus defined as $f_{\lambda,i}(NULL) = x'_i$ that $(1 - \lambda)|O_i|$ values in O_i are smaller than x'_i, while dominated position mapping $g_{\lambda,i}$ is defined as $g_{\lambda,i}(NULL) = x''_i$ that $\lambda|O_i|$ values in O_i are smaller than x''_i.

Point	Dimension 1	Dimension 2
A	NULL	0.5
B	0.3	NULL
C	0.5	0.4
D	0.7	0.5
E	0.9	0.6

Table 6 Example of Mapping Dominance

In Table 6, we present a small example of mapping dominance over a data set with missing values. In this data set, the values on the first dimension of point A and on the second dimension of point B are missing. If $\lambda = 0.25$, $f_{\lambda,1}(A[1]) = 0.9$ and $g_{\lambda,1}(A[2]) = 0.3$, since one out of four existing values over the first dimension is no larger than 0.3 while only one value is above 0.9. If $\lambda = 0.5$, $f_{\lambda,2}(B[2]) = 0.5$ and $g_{\lambda,2}(B[2]) = 0.5$, since half of the existing values is no larger than 0.5 while the other half is no smaller than 0.5.

Theorem 7 *If $0 \leq \lambda \leq 1/2$, then the MD_λ relationship satisfies the properties of rationality, scaling robustness, shifting robustness and transitivity.*

Proof MD definitely follows the rationality property since it degenerates to TD when two points have no missing values. Given any scaling factors α and shifting factors β, the mappings $f_{\lambda,i}$ and $g_{\lambda,i}$ follow the condition of Lemma 7 and Lemma 8. Therefore, it must be scaling and shifting robust. Finally, when $0 \leq \lambda \leq 1/2$, $g_{\lambda,i}(x_i)$ must be smaller or equal to $f_{\lambda,i}(x_i)$ for any x_i; this property satisfies the condition of Lemma 9, making the transitivity property valid.

Even when λ is larger than 1/2, MD_λ only violates the transitivity property while keeping all the other properties valid. Another advantage of MD_λ is its natural extension to ordered dominance class based on the following lemma.

Lemma 10 *If $MD_\lambda(p, q)$, $MD_{\lambda'}(p, q)$ for all $\lambda' \geq \lambda$.*

Proof Since $f_{\lambda',i}(x_i) \leq f_{\lambda,i}(x_i)$ for any x_i when $\lambda' \geq \lambda$, and $g_{\lambda',i}(x_i) \geq g_{\lambda,i}(x_i)$ for any x_i when $\lambda' \geq \lambda$, p dominates q when λ increases to λ'.

Therefore, mapping dominance is an ordered dominance class with index set Θ based on all λ values on the real number interval $[0, 0.5]$.

We note here that our method for data with missing values should not be seen as a simple plug-in of constants for missing values. The reasons are twofold. First, we decide concrete values for missing values carefully based on the probability distribution of the corresponding dimension, rather than on an ad hoc basis. Second, our concrete value selection is enabled within the general framework we have established in this paper, which ensures solid semantics which simple plugins of constants apparently lack.

6.2 Combination with Other Dominance Relationships

In the original MD_λ relationship, the points are compared with respect to traditional dominance relationship after the mapping values are calculated. A natural question arises here on the possibility of combining mapping dominance with other dominance relationships on certain data, such as Cone Dominance and Log-Scale Cone Dominance introduced in the previous section. In this part of this section, we provide some positive answers to this question.

Here, let MD^D_λ denote a new relationship on missing value, with another certain dominance D replacing TD in the original definition of mapping dominance. The following lemmas imply that the properties are very likely to be inherited from D to MD^D_λ.

Lemma 11 *If the properties of scaling robustness or shifting robustness hold for D, they also hold with MD^D_λ*

The proof of the lemma above is similar to those for Lemma 7 and Lemma 8.

Lemma 12 *If the property of transitivity holds for D and $TD(g(p), f(p)) \rightarrow D(g(p), f(p))$, transitivity is also valid with MD^D_λ.*

Proof If $MD^D_\lambda(p, q)$ and $MD^D_\lambda(q, r)$, it implies that $D(f(p), g(q))$ and $D(f(q), g(r))$, based on the definition of mapping dominance. Due to the condition of the lemma, we have $D(g(q), f(q))$ because $TD(g(q), f(q))$. With the property of transitivity on D, it is easy to derive that $D(f(p), g(r))$, leading to $MD^D_\lambda(p, r)$. This completes the proof on the validity of transitivity property on MD^D_λ

Last lemma implies that when D is stronger than TD, the property of transitivity in D can be passed to MD^D_λ. Considering the employment of CD and LCD in the generalized definition of mapping dominance, we can verify the properties of new mapping dominance relationship based on the lemmas above. To simplify the notation, we use $MCD_{\lambda,\gamma}$ and $MLCD_{\lambda,\gamma}$ to denote the new mapping dominance with these two dominance relationships respectively, with λ and γ being the parameters for them correspondingly. When $\gamma \geq 0$, both $MCD_{\lambda,\gamma}$ and $MLCD_{\lambda,\gamma}$ have property of transitivity, since CD and LCD are stronger than TD when $\gamma \geq 0$. Therefore, $MCD_{\lambda,\gamma}$ ($\gamma \geq 0$) is consistent with the properties of shifting robustness and transitivity, and $MLCD_{\lambda,\gamma}$ with the properties of scaling robustness and transitivity.

6.3 Algorithm Applicability

In Section 4, we present some general algorithms for skyline queries and discuss their applicabilities with respect to the properties of the underlying dominance relationship. However, all of the algorithms are designed for data sets with complete information on all dimensions. In this part of the section, we will present some extensions over the algorithms to handle mapping dominance over incomplete data.

For the BNL algorithm, no modification is necessary for the extension to mapping dominance. The only operation called in BNL is some verification between some pair of points on their dominance relationship, which can be simply implemented by an independent component. Since the mapping dominance relationship can be verified based on the definition, this component can be seamlessly integrated with BNL algorithm.

For SFS, TSA and BBS algorithms, the problem remains when they need to use some sorting function or indexing structure, which does not support points with missing values directly. To overcome the difficulties with these three algorithms, we hereby propose two simple schemes, enabling the system to employ traditional sorting and indexing component without too much modification.

6.3.1 Sorting for Mapping Dominance

When some sorting function is invoked over the points with missing values, a virtual position p' for the original point p is constructed by filling the missing value on dimension i with $f_{0.5,i}(p)$. Given the virtual positions for all data points, some conventional sorting algorithm will be called to order the points based on their virtual positions on the sum of all dimensions in non-descending order.

Lemma 13 *If p' is sorted before q' with the new sorting method, q cannot dominate p based on dominance relationship MD^D_λ, if 1) D is stronger than TD, i.e. $TD(p,q) \rightarrow D(p,q)$ and 2) D has the property of transitivity.*

Proof We prove this lemma by contradiction. If the lemma does not hold, meaning that we can find some p' sorted before q' but q dominates p by MD^D_λ. Then, the dominance must be valid on $D(f(q), g(p))$. By the condition 1) on the dominance relationship D, we have $D(g(p), p')$, $D(q', f(q))$. It leads to $D(q', p')$ if combining the previous result with the property of transitivity. This is contradicted to property of rationality, which implies that $\overline{D}(q', p')$ since p' is sorted before p'.

The correctness of last lemma directly implies that all of the mapping dominance relationships, proposed in this section, are consistent with the new sorting scheme, since the correctness of sorting in SFS and TSA algorithms depends on the result of the lemma.

6.3.2 Indexing for Mapping Dominance

When indexing the points with missing values, each point p is represented by some rectangle with two corners at $f(p)$ and $g(p)$. Thus, given a node in the indexing tree, such as R-Tree, the Minimum Bounding Box is the minimum rectangle in the space covering all the rectangles created for the points stored behind this node.

Lemma 14 *Given two MBRs M_1 and M_2, if one point $p \in M_1$ cannot be dominated by any other point $q \in M_2$ by the definition of mapping dominance, M_2 cannot dominate M_1.*

Proof Since no point q in M_2 dominates p in M_1, we know that $g(p)$ cannot be dominated by any $f(q)$. Considering the MBRs M_1 and M_2, there is at least one dimension that minimum boundary of M_1 cannot be bounded by the maximum boundary of M_2. Thus, M_2 cannot dominate M_1.

Based on the last lemma, the pruning strategies in any indexing tree must be valid, because no pruning will remove real skyline point from the candidate set. Therefore, it is safe to use the indexing scheme for mapping dominance relationship when computing skyline query or top-k skyline query.

7 Experiments

In this section, we evaluate the efficiencies of the algorithm on variants of skyline queries, and effectiveness of the new dominance definitions, on synthetic and real data sets.

7.1 Experimental Settings

In the experiments, both synthetic data sets and real data sets are used to evaluate the performance. There are three common types of synthetic data sets that have been used in previous studies of skyline queries [4], including correlated (C), independent (I) and anti-correlated (A) data sets. In correlated data sets, the dimensions of the points are positively correlated, meaning a point with a better value on one attribute is very likely to have better values on other attributes. In independent data sets, the dimensions are independent and uniformly distributed. In anti-correlated data sets, the dimensions are anti-correlated, which is implemented by keeping the sums on all dimensions for all points around the same value [4]. We also adopt two real data sets: the NBA data set[2] and the MovieLens data set[3], both of which have been used in the study of k-dominant skyline queries in [6]. The NBA data set contains more than 17,000 records of players' season

[2] http://www.databasebasketball.com

[3] http://movielens.umn.edu/

Parameter	Range
Dimensionality	5,10,**15**,20
Data Size (100K)	**1**,2,3,4,5,6,8,10
Distribution	**C,I,A**
Top-k Size	50,**100**,150,200,250
Parameter γ for CD (0.01)	5,**10**,15,20

Table 7 Parameters in Tests on Synthetic Data Sets

γ(0.01)	C	I	A
5	441	40999	81527
10	134	16214	58205
15	52	4908	32450
20	28	1165	8961

Table 8 Skyline Cardinalities on Synthetic Data Sets with Varying Dominance Parameter

statistics on 17 attributes from the first season of NBA in 1945 to the season in 2002. The MovieLens data set was collected by the movie-lens web site from September 1997 to April 1998. There are 100,000 ratings from 943 users on 1682 movies in the data set. All the users in the data set have rated at least 20 movies. However, this data set is still very sparse, with missing values in most of the entries.

We compare the performances of different algorithms and different dominance relationships. The skyline algorithms evaluated here include: General Block Nest Loop (BNL), General Sort Filter Skyline (SFS) and General Branch-and-Bound Skyline (BBS). The top-k skyline algorithms evaluated include: Binary SFS (B-SFS), Progressive SFS (P-SFS), Binary BBS (B-BBS) and Progressive BBS (P-BBS). The dominance relationships tested include the three new variant relationship proposed in this paper: Cone Dominance (CD), Log-scale Cone Dominance (LCD) and Mapping Dominance (MD).

All experiments are run on a PC with PIII 1.8GHz CPU, 1GB main memory and 20GB hard disk. The programs are compiled with GCC v3.4.3 in Linux Fedora 3 system.

7.2 Experiments on Synthetic Data Sets

In the experiments on synthetic data sets, we vary some parameters of the data sets, such as dimensionality and data size. Since LCD is not very different from CD, we only test the performances of CD on the data sets. For skyline queries and top-k skyline queries, we also vary the dominance parameter γ and specified result size k, respectively. The varying ranges of the parameters are summarized in Table 7, in which default values are marked in bold font.

7.2.1 On Skyline Algorithms for Cone Dominance

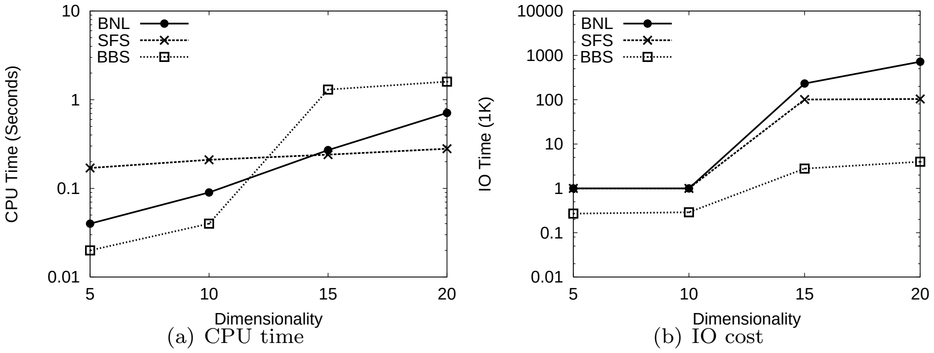
In Figures 5-7, we show the result on skyline queries with varying dimensionality. On correlated data, BBS is more CPU and IO efficient than the other two algorithms when the dimensionality is not very large. When the dimensionality grows, the CPU time performance of BBS deteriorates rapidly mainly because of the worse indexing quality in high dimensional space. This phenomenon happens even earlier on independent data sets when the dimensionality remains small. On independent data sets, SFS is usually the fastest method on high dimensional space. On anti-correlated data sets, the performances of the algorithms tend to converge when dimensionality increases since all of them have to compare every pair of points.

When γ increases from 0.05 to 0.2, the expansion of dominance ability leads to the quick decrease of skyline cardinalities (Table 8). All the three algorithms, BNL, SFS and BBS can be more CPU and IO efficient when given a larger γ (Figures 8-10). BNL is faster than SFS on high dimensional correlated data sets since the sorting in SFS takes too much time. SFS is more efficient on high dimensional independent data sets due to the advantage of sorting. The performances of SFS and BBS in CPU and IO are almost the same on anti-correlated data sets.

In Figures 11-13, we present the experiment results when the size of the synthetic data set is varied from 100K to 1M. In this group of tests, BNL is worse than SFS on CPU and IO on all data sets. BBS is much slower than SFS, but with almost the same IO cost since the IO cost of retrieving new points from the underlying indexing structure is much smaller than the IO cost spent on comparing a point with all current skyline points.

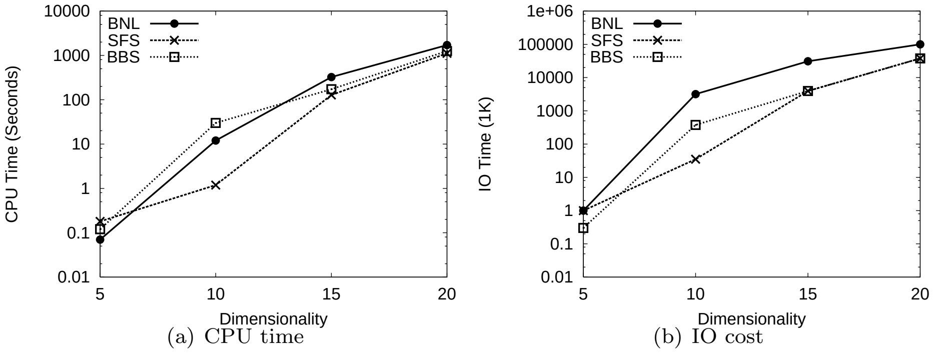
7.2.2 On Top-k Skyline Algorithms for Cone Dominance

The experimental results on varying dimensionality are presented in Figures 14-16. The figures show the overall optimality of progressive search over binary search. Binary search with SFS or BBS can only work in data sets with less than 10 dimensions while progressive search with SFS or BBS can be very scalable to high dimensional data sets. On all three types of data sets, P-SFS is quicker than P-BBS while the IO costs of P-SFS and P-BBS are about the same. Since B-SFS and B-BBS are much worse than P-SFS and P-BBS, we will only evaluate P-SFS and P-BBS in the rest of the section.



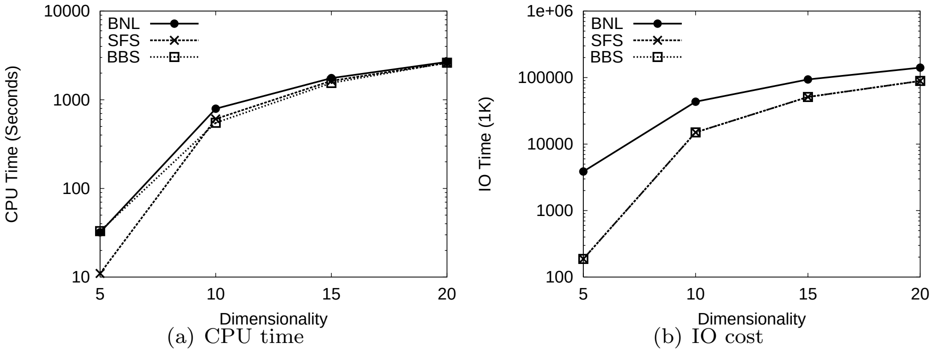
(a) CPU time (b) IO cost

Fig. 5 Skyline Queries with Varying Dimensionality on Correlated Data Sets



(a) CPU time (b) IO cost

Fig. 6 Skyline Queries with Varying Dimensionality on Independent Data Sets

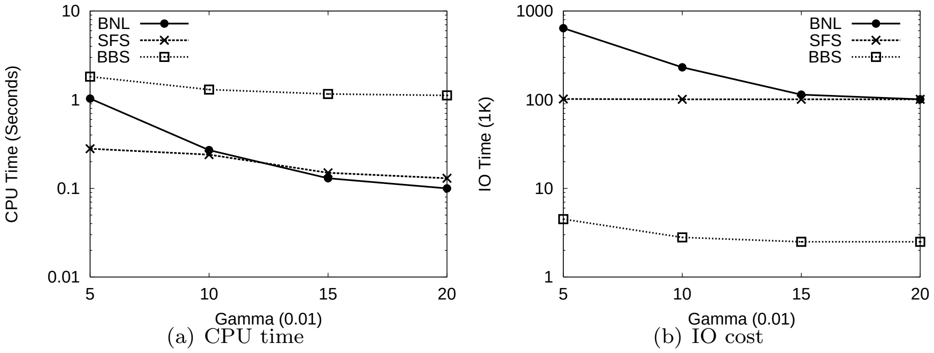


(a) CPU time (b) IO cost

Fig. 7 Skyline Queries with Varying Dimensionality on Anti-Correlated Data Sets

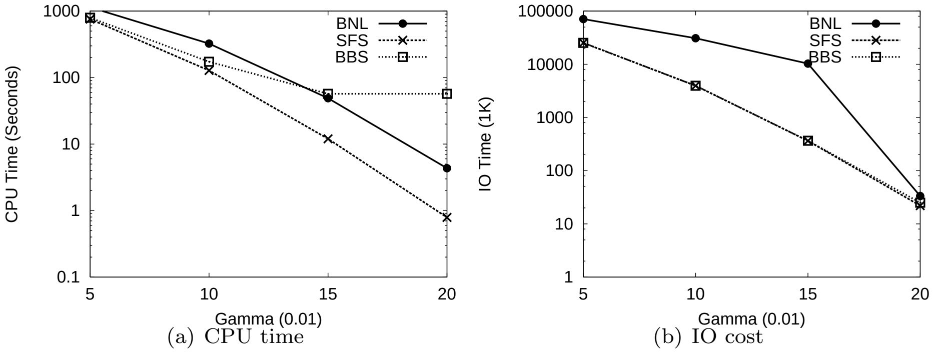
In the experiments of varying the specified skyline size k on correlated datasets as shown in Figure 17, both the P-SFS and P-BBS algorithms scale well with k. P-BBS has the advantage of IO efficiency while P-SFS is better on CPU time. This is due to the fact that simple sorting is more effective than indexing in high dimensional space, but indexing is better on reducing IO. We omit similar results on independent and anti-correlated data sets.

In the results shown in Figure 18, we can conclude that both computation costs and IO costs of P-SFS and P-BBS are linear to the data size on correlated data sets. Similar results on independent and anti-correlated data sets are also omitted here.



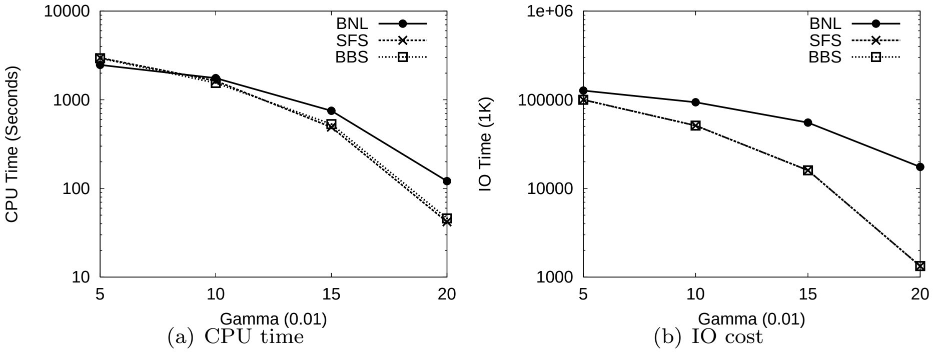
(a) CPU time (b) IO cost

Fig. 8 Skyline Queries with Varying γ on Correlated Data Set



(a) CPU time (b) IO cost

Fig. 9 Skyline Queries with Varying γ on Independent Data Set



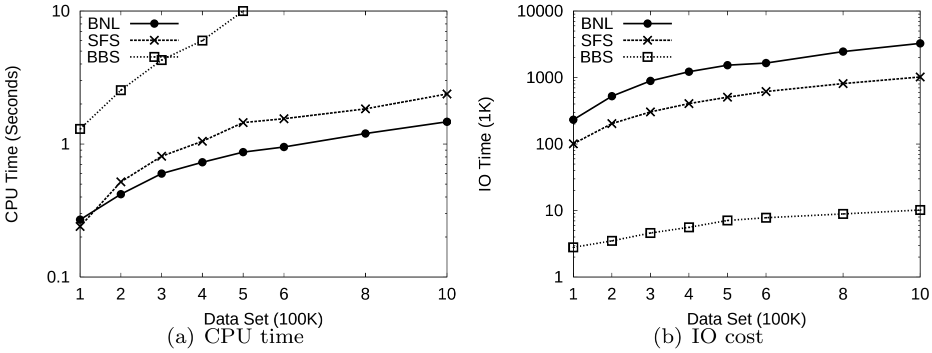
(a) CPU time (b) IO cost

Fig. 10 Skyline Queries with Varying γ on Anti-Correlated Data Set

7.3 Experiments on Real Data Sets

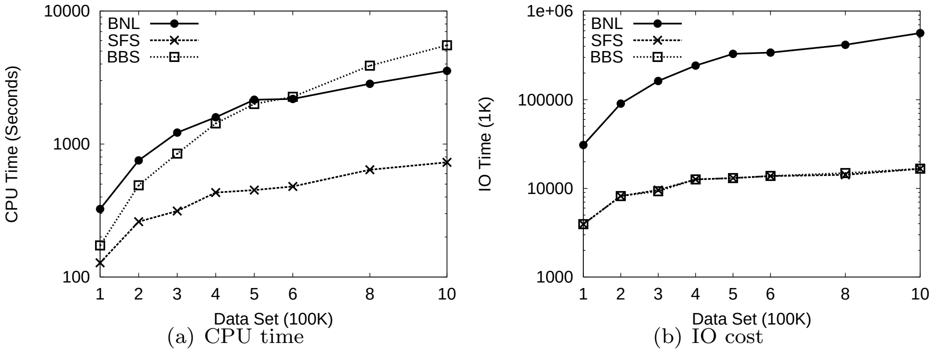
7.3.1 CD and LCD on NBA data set

In Table 9, we show the two skylines retrieved by the top-k skyline query with cone dominance and log-scale cone dominance respectively. By simple observations, we can see that the skyline with cone dominance (on the left) prefers center players while the skyline with log-scale cone dominance (on the right) prefers guard players. The difference stems from the Euclidean distances used in cone dominance and log-scale cone dominance. In cone dominance, the Euclidean distance between two players are dominated by those "large" attributes, such as points, rebounds and assists, which leads to bias to centers with high scoring and rebounds. In log-scale cone dominance, the Euclidean distance is a more average aggregation of all attributes, preferring who are more average on different attributes.



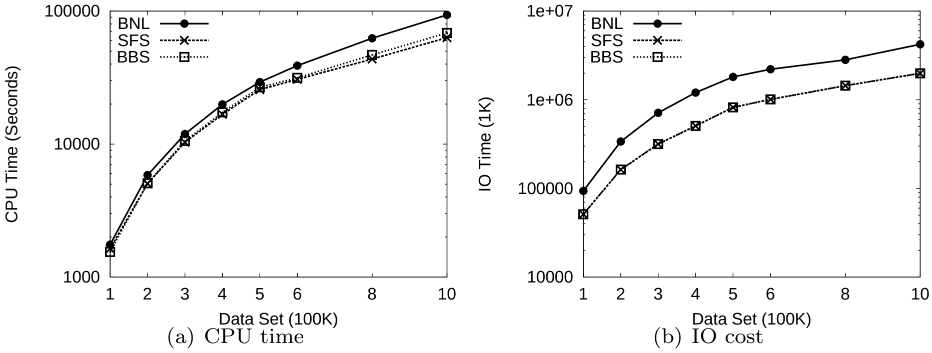
(a) CPU time (b) IO cost

Fig. 11 Skyline Queries with Varying Data Size on Correlated Data Sets



(a) CPU time (b) IO cost

Fig. 12 Skyline Queries with Varying Data Size on Independent Data Sets

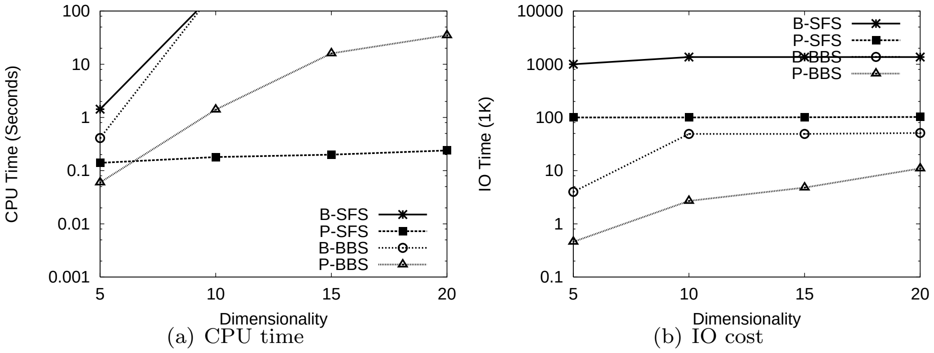


(a) CPU time (b) IO cost

Fig. 13 Skyline Queries with Varying Data Size on Anti-Correlated Data Sets

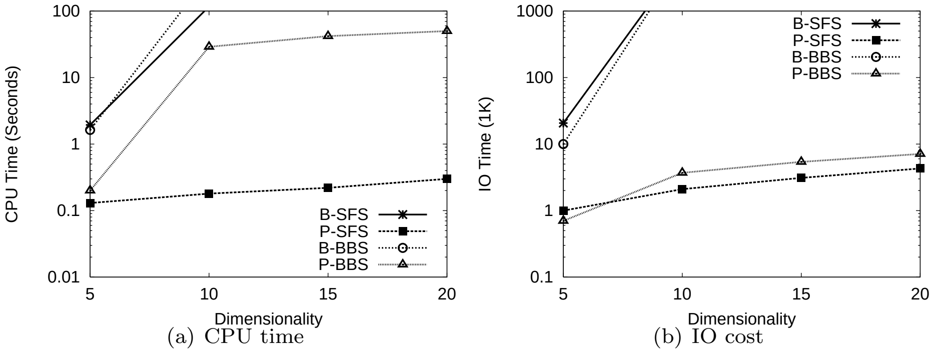
7.3.2 Mapping Cone Dominance on Movie Data Set

Although we can construct an ordered dominance class $\{MD_\lambda\}$ by gradually increasing λ from 0 to 1/2, the skyline size can still be much larger than expected. Experiments of MD on the MovieLens data set shows that even if we set λ to 0.5, there are still more than 1000 skyline points returned, which makes the result meaningless. This indicates that mapping dominance itself may not be plausible in reducing skyline size to user's expectations. A straightforward alternative is employing other mapping dominance relationships, combined with other dominance on certain data, such as MCD and MLCD, introduced in Section 6.2. Since MCD and MLCD have two parameters, with λ controlling the mapping procedure and γ controlling the degree of cone, the index set on the ordered dominance class for top-k skyline query becomes hard to define. To simplify the problem, we fix λ at 0.5 and give freedom on γ when computing the top-k skyline query.



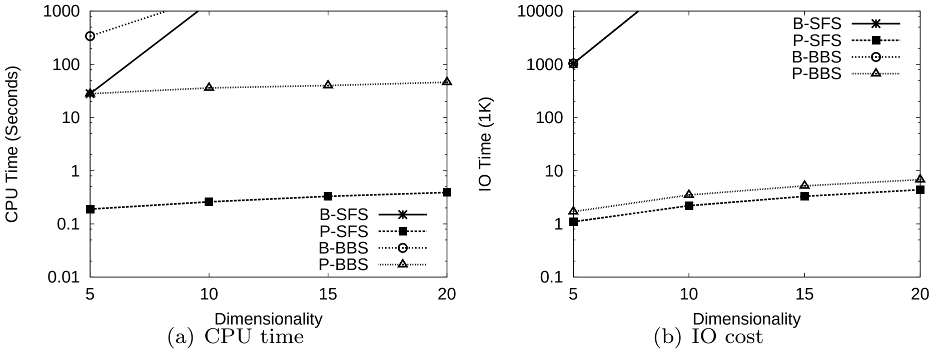
(a) CPU time (b) IO cost

Fig. 14 Top-k Skyline Query with Varying Dimensionality on Correlated Data Sets



(a) CPU time (b) IO cost

Fig. 15 Top-k Skyline Query with Varying Dimensionality on Independent Data Sets

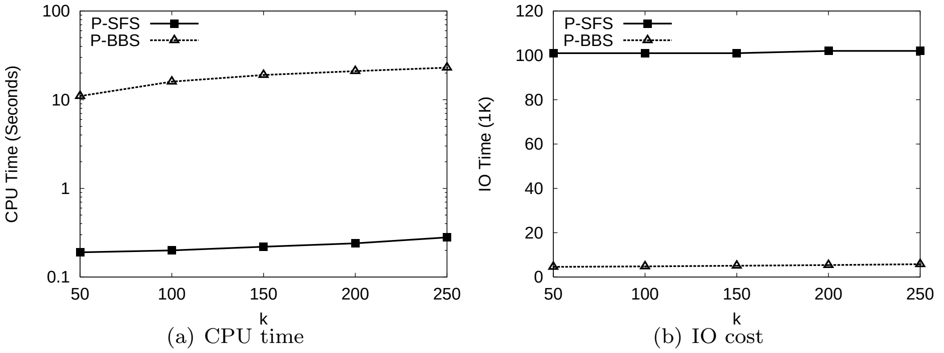


(a) CPU time (b) IO cost

Fig. 16 Top-k Skyline Query with Varying Dimensionality on Anti-Correlated Data Sets

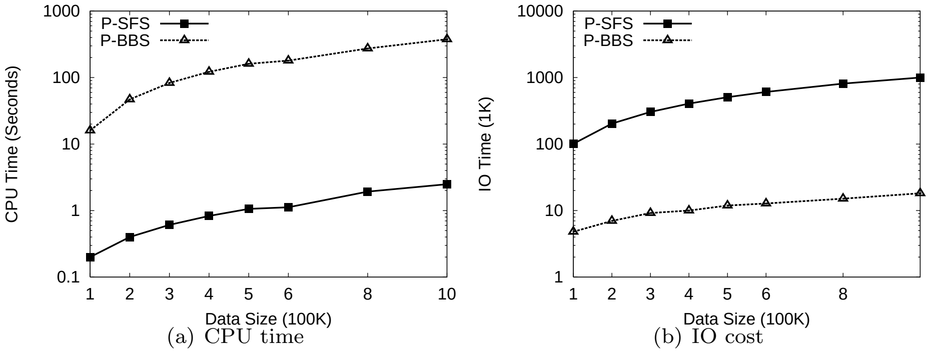
In Table 10, we present the top-k skyline set with Mapping Cone Dominance (left) and Mapping Log-Scale Cone Dominance (right) as underlying dominance relationships. In terms of their IMDB scores[4], our method does discover the popular movies. The advantage of the top-k skyline query is that we do not need to manually adjust the aggregation function as IMDB does. By comparing the two results of MCD and MLCD, we find five movies shared by both skylines, all of which are well recognized classic movies. The left ones indicate the difference on the preference of the two skylines. MCD prefers artistic movies, which are liked by a small fraction of reviewers, while MLCD biases to mass entertainment movies, such as two animations "wrong trousers" and "close shave". This is still due to the distance used by these two dominance relationships, as is discussed in relation with the results of the NBA data set.

[4] www.imdb.com



(a) CPU time (b) IO cost

Fig. 17 Top-*k* Skyline Query with Varying *k* on Correlated Data Set



(a) CPU time (b) IO cost

Fig. 18 Top-*k* Skyline Query with Varying Data Size on Correlated Data Sets

Movie & Year	**IMDB score**	**Movie & Year**	**IMDB score**
Star Wars 1977	8.8	Living in Oblivion 1995	7.3
Forbidden Planet 1956	7.7	Godfather 1972	9.1
Manchurian Candidate 1962	8.4	Wrong Trousers 1993	8.5
Big Sleep 1946	8.3	As Good As It Gets 1997	7.7
Killing Fields 1984	8.0	Schindler's List 1993	8.8
As Good As It Gets 1997	7.7	One Flew Over the Cuckoo's Nest 1975	8.8
Godfather 1972	9.1	Close Shave 1995	8.3
8 1/2 1963	8.2	Star Wars 1977	8.8
Wings of Desire 1987	8.0	Wild Bunch 1969	8.1
One Flew Over the Cuckoo's Nest 1975	8.8	Manchurian Candidate 1962	8.4

Table 10 Top-10 Skylines with MCD and MLCD on Movie Data Set

Name & Year	**Position**	**Name & Year**	**Position**
G. Mcginnis 1974	Forward	C. Barkeley 1987	Forward
M. Malone 1978	Center	C. Barkeley 1988	Forward
M. Malone 1981	Center	M. Johnson 1989	Guard
C. Barkley 1987	Forward	M. Jordan 1989	Guard
H. Olajuwon 1989	Center	S. Pippen 1994	Forward
J. Stockton 1990	Guard	A. Walker 1997	Forward
J. Stockton 1991	Guard	V. Carter 2000	Guard
G. Payton 1999	Guard	A. Walker 2000	Forward
A. Walker 2000	Forward	K. Bryant 2002	Guard
P. Pierce 2001	Guard	T. McGrady 2002	Guard

Table 9 Top-10 Skylines on NBA data set

8 Conclusion

In this paper, we have investigated the possibility of using dominance relationships other than the traditional one in skyline queries. Among the extensive studies on skyline queries recently, we are the first to present a general framework on the robustness of skyline query results. While traditional dominance is the only binary relationship satisfying all desired properties, as we have proved, we have proposed some new dominance relationships with relaxed properties to improve the flexibility of skyline queries. Our study has also identified the basic requirements for use as a guide in designing specific skyline queries with expected results.

References

1. I. Bartolini, P. Ciaccia, and M. Patella. Efficient sort-based skyline evaluation. *ACM Trans. Database Syst.*, 33(4), 2008.
2. J. L. Bentley, K. L. Clarkson, and D. B. Levine. Fast linear expected-time algorithms for computing maxima and convex hulls. In *SODA*, pages 179–187, 1990.
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. *J. ACM*, 25(4):536–543, 1978.
4. S. Börzsönyi, D. Kossmann, and K. Stocker. The skyline operator. In *ICDE*, pages 421–430, 2001.
5. C. Y. Chan, P.-K. Eng, and K.-L. Tan. Stratified computation of skylines with partially-ordered domains. In *SIGMOD*, pages 203–214, 2005.
6. C.-Y. Chan, H. V. Jagadish, K.-L. Tan, A. K. H. Tung, and Z. Zhang. Finding k-dominant skylines in high dimensional space. In *SIGMOD*, pages 503–514, 2006.
7. J. Chomicki. Preference formulas in relational queries. *ACM TODS*, 24(4):427–466, 2003.
8. J. Chomicki, P. Godfrey, J. Gryz, and D. Liang. Skyline with presorting. In *ICDE*, pages 717–719, 2003.
9. I. Diakonikolas and M. Yannakakis. Succinct approximate convex pareto curves. In *SODA*, pages 74–83, 2008.
10. P. Godfrey, R. Shipley, and J. Gryz. Maximal vector computation in large data sets. In *VLDB*, pages 229–240, 2005.
11. W. Kießling. Foundations of preferences in database systems. In *VLDB*, pages 311–322, 2002.
12. V. Koltun and C. H. Papadimitriou. Approximately dominating representatives. In *ICDT*, pages 204–214, 2005.
13. D. Kossmann, F. Ramsak, and S. Rost. Shooting stars in the sky: an online algorithm for skyline queries. In *VLDB*, pages 275–286, 2002.
14. K. C. K. Lee, B. Zheng, H. Li, and W.-C. Lee. Approaching the skyline in z order. In *VLDB*, pages 279–290, 2007.
15. C. Li, B. C. Ooi, A. K. H. Tung, and S. Wang. DADA: A data cube for dominant relationship analysis. In *SIGMOD*, pages 659–670, 2006.
16. X. Lin, Y. Yuan, Q. Zhang, and Y. Zhang. Selecting stars: The k most representative skyline operator. In *ICDE*, pages 86–95, 2007.
17. M. D. Morse, J. M. Patel, and H. V. Jagadish. Efficient skyline computation over low-cardinality domains. In *VLDB*, pages 267–278, 2007.
18. D. Papadias, Y. Tao, G. Fu, and B. Seeger. An optimal and progressive algorithm for skyline queries. In *SIGMOD*, pages 467–478, 2003.
19. D. Papadias, Y. Tao, G. Fu, and B. Seeger. Progressive skyline computation in database systems. *TODS*, 30(1):41–82, 2005.
20. N. Sarkas, G. Das, N. Koudas, and A. K. H. Tung. Categorical skylines for streaming data. In *SIGMOD Conference*, pages 239–250, 2008.
21. U. Shaft and R. Ramakrishnan. When is nearest neighbors indexable? In *ICDT*, pages 158–172, 2005.
22. K. L. Tan, P. K. Eng, and B. C. Ooi. Efficient progressive skyline computation. In *VLDB*, pages 301–310, 2001.
23. P. Wu, D. Agrawal, Ö. Egecioglu, and A. E. Abbadi. Deltasky: Optimal maintenance of skyline deletions without exclusive dominance region generation. In *ICDE*, pages 486–495, 2007.
24. Z. Zhang, L. V. S. Lakshmanan, and A. K. H. Tung. On domination game analysis for microeconomic data mining. *TKDD*, 2(4), 2009.