$\mathcal{MP2R}$: A Human-Centric Skyline Relaxation Approach

Djamal Belkasmi and Allel Hadjali

Abstract Skyline queries have gained much attention in the last decade and are proved to be valuable for multi-criteria decision making. They are based on the concept of Pareto dominance. When computing the skyline, two scenarios may occur: either (i) a huge number of skyline which is less informative for the user or (ii) a small number of returned objects which could be insufficient for the user needs. In this paper, we tackle the second problem and propose an approach to deal with it. The idea consists in making the skyline more permissive by adding points that strictly speaking do not belong to it, but are close to belonging to it. A new fuzzy variant of dominance relationship is then introduced. Furthermore, an efficient algorithm to compute the relaxed skyline is proposed. Extensive experiments are conducted to demonstrate the effectiveness of our approach and the performance of the proposed algorithm.

1 Introduction

Skyline queries [2] are specific and popular example of preference queries. They are based on Pareto dominance relationship. This means that, given a set D of d-dimensional points, a skyline query returns, the skyline S, set of points of D that are not dominated by any other point of D. A point p dominates another point q iff p is better than or equal to q in all dimensions and strictly better than q in at least one dimension. One can see that skyline points are incomparable. This kind of queries provide an adequate tool that can help users to make intelligent decisions in the presence of multidimensional data where different and often conflicting criteria must

D. Belkasmi
DIF-FS/UMBB, Boumerdes, Algeria
e-mail: belkasmi.djamel@gmail.com

A. Hadjali(✉)
LIAS/ENSMA, Poitiers, France
e-mail: allel.hadjali@ensma.fr

T. Andreasen et al. (eds.), *Flexible Query Answering Systems 2015*,
Advances in Intelligent Systems and Computing 400,
DOI: 10.1007/978-3-319-26154-6_18

be taken. Several research studies have been conducted to develop efficient algorithms and introduce multiple variants of skyline queries [9, 12, 16, 18]. However, querying a d-dimensional data sets using a skyline operator may lead to two possible cases: (i) a large number of skyline points returned, which could be less informative for users, (ii) a small number of skyline points returned, which could be insufficient for users. To solve the two above problems, various approaches have been proposed to refine the skyline, therefore reducing its size [1, 3, 4, 6, 10, 11, 14, 15], but only very few works exist to relax the skyline in order to increase the number of skyline results [8, 10]. Goncalves and Tineo [8] propose a flexible dominance relationship using fuzzy comparison operators. This increases the skyline with points that are only weakly dominated by any other point. In [10], some ideas of relaxing the skyline have also been proposed.

In this paper, and taking as starting point the study in [10], we address in deep the problem of skyline relaxation to return more interesting results to the user. In particular, we develop an efficient approach, called $\mathcal{MP2R}$ ($\mathcal{M}$uch $\mathcal{P}$referred $\mathcal{R}$elation for $\mathcal{R}$elaxation), for skyline relaxation. The approach relies on a novel fuzzy dominance relationship *Much Preferred (MP)* which makes more demanding the dominance between the points of D. In this context, a point still belonging to the skyline unless it is much dominated, in the spirit of MP relation, by another skyline point. By this way, much points would be considered as incomparable and then as elements of the new relaxed skyline, denoted S_{relax}. Note that such points are ruled out from the skyline when applying classical Pareto dominance. Furthermore, an algorithm for computing the skyline S_{relax} efficiently is provided. In summary, our main contributions cover the following points:

- We investigate a new variant of fuzzy dominance relation based on the MP relation and provide the semantic basis for a relaxed variant of skyline S_{relax}.
- We develop and implement an algorithm to compute S_{relax} efficiently.
- We conduct a set of experiments to study and analyze the relevance and effectiveness of S_{relax}.

The paper is structured as follows: Section 2 provides some necessary background on fuzzy set theory and skyline queries and a survey on existing approaches. In Section 3, we introduce a new approach for skyline relaxation based on MP dominance relationship. In Section 4, the algorithm to efficiently compute S_{relax} is presented, while Section 5 is devoted to the experimental study. Finally, Section 6 concludes the paper and draws some lines for future works.

2 Background

In this section, we recall some notions on fuzzy set theory and skyline queries. Then, we review some related works.

2.1 Fuzzy Set Theory

The concept of fuzzy sets has been developed by Zadeh [19] in 1965 to represent classes or sets whose limits are imprecise. They can describe gradual transitions between total belonging and rejection. Typical examples of these fuzzy classes are those described with adjectives or adverbs in natural language, as *not expensive*, *fast* and *very close*. Formally, a fuzzy set F on the universe X is described by a membership function $\mu_F : X \longrightarrow [0, 1]$, where $\mu_F(x)$ represents the **degree of membership** of x in F. By definition, if $\mu_F(x) = 0$ then the element x **does not belong to** F, if $\mu_F(x) = 1$ then x **completely belongs to** F, these elements form the **core** of F denoted by $Cor(F) = \{x \in F | \mu_F(x) = 1\}$. When $0 < \mu_F(x) < 1$, we talk about a **partial membership**, these elements form the **support** of F denoted by $Supp(F) = \{x \in F | \mu_F(x) > 0\}$. Moreover, more the value of $\mu_F(x)$ is close to 1, more x belongs to F. Let $x, y \in F$, we say that x is preferred to y iff $\mu_F(x) > \mu_F(y)$. If $\mu_F(x) = \mu_F(y)$, then x and y have the same preference. In practice, F is represented by a trapezoid membership function (t.m.f) $(\alpha, \beta, \varphi, \psi)$, where (α, ψ) is the support and (β, φ) is its core.

2.2 Skyline Queries

Skyline queries [2] are a specific, yet relevant, example of preference queries. They rely on Pareto dominance principle which can be defined as follows:

Definition 1. Let D be a set of d-dimensional data points and u_i and u_j two points of D. u_i is said to dominate in Pareto sense u_j (denoted $u_i \succ u_j$) iff u_i is better than or equal to u_j in all dimensions and better than u_j in at least one dimension. Formally, we write

$$u_i \succ u_j \Leftrightarrow (\forall k \in \{1, .., d\}, u_i[k] \geq u_j[k]) \wedge (\exists l \in \{1, .., d\}, u_i[l] > u_j[l]) \quad (1)$$

where each tuple $u_i = (u_i[1], u_i[2], u_i[3], ..., u_i[d])$ with $u_i[k]$ stands for the value of the tuple u_i for the attribute A_k.

In (1), without loss of generality, we assume that the largest value, the better.

Definition 2. The skyline of D, denoted by S, is the set of points which are not dominated by any other point.

$$u \in S \Leftrightarrow \nexists u' \in D, u' \succ u \quad (2)$$

Example 1. To illustrate the Skyline, let us consider a database containing information on candidates as shown in Table 1. The list of candidates includes the following information: Code, Age, Management experience (man_exp in years), Technical experience (tec_exp in years) and distance work to Home (dist_wh in Km).

Ideally, personnel manager is looking for a candidate with the largest management and technical experience (Max man_exp and Max tec_exp), ignoring the other pieces

Table 1 List of candidates.

code	age	man_exp	tec_exp	dist_wh
M1	32	5	10	35
M2	41	7	5	19
M3	37	5	12	45
M4	36	4	11	39
M5	40	8	10	18
M6	30	4	6	27
M7	31	3	4	56
M8	36	6	13	12
M9	33	6	6	95
M10	40	7	9	20

of information. Applying the traditional skyline on the candidate list shown in Table 1 returns the following candidates: M_5, M_8. As can be seen, such results are the most interesting candidates (see Fig. 1).

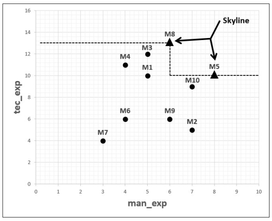

Fig. 1 Skyline of candidates

2.3 Related Work

Since its proposal, the skyline queries have been recognized as a useful and practical technique to capture user preferences and integrate them in the querying process. They have been widely used in various types of applications: (decision support, spatial data management, data mining, navigation systems, …). In the years that followed the emergence of the concept of skyline queries, computing the skyline was the major concern, most of the works were about designing efficient evaluation algorithms under different conditions and in different contexts. In general, the existing algorithms can be classified into two categories: sequential algorithms and index based algorithms. Sequential algorithms scan the entire dataset to compute the skyline and don't require a pre-computed structure (index, hash table, …). They include: the Block Nested Loops (BNL) [2], Divide and Conquer (D&C) [2], Sort First Skyline (SFS) [5] and Linear Elimination Sort for Skyline (LESS) [7]. Index-based algorithms compute the skyline by accessing just a part of the dataset through the use of indexes (R-tree, B-Tree, Bitmap Index, ...), let us mention: Bitmap algorithm [17], Index algorithm [17], Nearest Neighbor algorithm [13] and Branch and Bound algorithm [15].

Some research efforts have been made to develop efficient algorithms and to introduce different variants of skyline queries [3, 4, 11, 12, 14, 16]. But only few works have been proposed to address the skyline relaxation issue. In Goncalves and Tineo [8], the problem of skyline rigidity is addressed by introducing a weak dominance relationship based on fuzzy comparison operators. This relationship allows enlarging the skyline with points that are not much dominated by any other point (even if strictly speaking they are dominated).

In [10], Hadjali et al. have introduced some ideas to define some novel variants of Skyline. First, one idea consists in refining the skyline by introducing some ordering between its points in order to single out the most interesting ones. The second idea aims at making the Skyline more flexible by adding some points that strictly speaking do not belong to it, but are not much dominated by any other point in all Skyline dimensions. The third one tries to simplify the skyline either by granulating the scales of the criteria which may enable us to cluster points that are somewhat similar. The last idea addresses the skyline semantics in the context of uncertain data.

Taking inspiration from that work, this papers tackles the problem of skyline relaxation. It develops a complete approach to make the Skyline more permissive where both the semantic basis of the relaxed skyline and its computation are addressed in a deep way.

3 $\mathcal{MP2R}$: An Approach for the Skyline Relaxation

Let a relation $R(A_1, A_2, ..., A_d)$ be defined in a d-dimensional space $\mathbb{D} = (\mathbb{D}_1, \mathbb{D}_2, ..., \mathbb{D}_d)$, where $\mathbb{D}_i$ is the domain attribute of A_i. We assume the existence of a total order relationship on each domain $\mathbb{D}_i$. U is a set of n tuples belonging to the relationship R, $U = (u_1, u_2, ..., u_n)$. Let S be the skyline of U and S_{relax} is the relaxed skyline of U returned by our approach $\mathcal{MP2R}$.

$\mathcal{MP2R}$ relies on a new dominance relationship that allows enlarging the skyline with the most interesting points among those ruled out when computing the initial skyline S. This new dominance relationship uses the relation *"Much Preferred (MP)"* to compare two tuples u and u'. So, u is an element of S_{relax} if there is no tuple $u' \in U$ such that u' is *much preferred* to u (denoted $MP(u', u)$) in all skyline attributes. Formally, we write:

$$u \in S_{relax} \Leftrightarrow \nexists u' \in U, \forall i \in \{1, ..., d\}, MP_i(u'_i, u_i) \tag{3}$$

where, MP_i is a fuzzy preference relation defined on the domain $\mathbb{D}_i$ of the attribute A_i and $MP_i(u'_i, u_i)$ expresses the extent to which the value u'_i is *much preferred* to the value u_i. Since *MP* relation is of a gradual nature, each element u of S_{relax} is associated with a degree ($\in [0, 1]$) expressing the extent to which u belongs to S_{relax}.

In fuzzy set terms, one can write:

$$\mu_{S_{relax}}(u) = 1 - \max_{u' \in U} \min_i \mu_{MP_i}(u'_i, u_i) = \min_{u' \in U} \max_i (1 - \mu_{MP_i}(u'_i, u_i)) \quad (4)$$

As for MP_i relation on $\mathbb{D}_i$, its semantics can be provided by the formulas (5) (see also Fig. 2). In terms of t.m.f., MP_i writes $(\gamma_{i1}, \gamma_{i2}, \infty, \infty)$, and denoted $MP_i^{(\gamma_{i1},\gamma_{i2})}$. It is easy to check that $MP_i^{(0,0)}$ corresponds to the regular preference relation expressed by means of the crisp relation *"greater than"*.

$$\mu_{MP_i^{(\gamma_{i1},\gamma_{i2})}}(u'_i, u_i) = \begin{cases} 0 & \text{if } u'_i - u_i \le \gamma_{i1} \\ 1 & \text{if } u'_i - u_i \ge \gamma_{i2} \\ \frac{(u'_i - u_i) - \gamma_{i1}}{\gamma_{i2} - \gamma_{i1}} & \text{else} \end{cases} \quad (5)$$

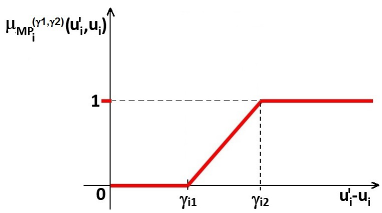

Fig. 2 The membership function $\mu_{MP_i^{(\gamma_{i1},\gamma_{i2})}}$

Let $\gamma = ((\gamma_{11}, \gamma_{12}), \cdots, (\gamma_{d1}, \gamma_{d2}))$ be a vector of pairs of parameters where $MP_i^{(\gamma_{i1},\gamma_{i2})}$ denotes the MP_i relation defined on the attribute A_i and $S_{relax}^{(\gamma)}$ denotes the relaxed skyline computed on the basis of the vector γ. One can easily check that the classical Skyline S is equal to $S_{relax}^{(\mathbf{0})}$, where $\mathbf{0} = ((0, 0), \cdots, (0, 0))$.

We say that $MP_i^{(\gamma_{i1},\gamma_{i2})}$ is more constrained than $MP_i^{(\gamma'_{i1},\gamma'_{i2})}$ if and only if $(\gamma_{i1}, \gamma_{i2}) \ge (\gamma'_{i1}, \gamma'_{i2})$ (i.e., $\gamma_{i1} \ge \gamma'_{i1} \wedge \gamma_{i2} \ge \gamma'_{i2}$).

Definition 3. Let γ and γ' be two vectors of parameters. We say that $\gamma \ge \gamma'$ if and only if $\forall i \in \{1, \cdots, d\}$, $(\gamma_{i1}, \gamma_{i2}) \ge (\gamma'_{i1}, \gamma'_{i2})$.

Proposition 1. *Let γ and γ' be two vectors of parameters. The following property holds: $\gamma' \le \gamma \Rightarrow S_{relax}^{(\gamma')} \subseteq S_{relax}^{(\gamma)}$.*

Proof. Let $\gamma' \le \gamma$ and let $u \in S_{relax}^{(\gamma')} \Rightarrow \nexists u' \in U, \forall i \in \{1, \cdots, d\}, MP_i^{(\gamma'_{i1},\gamma'_{i2})}(u'_i, u_i)$
$\Rightarrow \nexists u' \in U, \forall i \in \{1, \cdots, d\}, \mu_{MP_i^{(\gamma'_{i1},\gamma'_{i2})}}(u'_i, u_i) > 0$
$\Rightarrow \nexists u' \in U, \forall i \in \{1, \cdots, d\}, u'_i - u_i > \gamma'_{i1} \Rightarrow \forall u' \in U, \forall i \in \{1, \cdots, d\}, u'_i - u_i \le \gamma'_{i1}$
$\Rightarrow \forall u' \in U, \forall i \in \{1, \cdots, d\}, u'_i - u_i \le \gamma_{i1} \Rightarrow \nexists u' \in U, \forall i \in \{1, \cdots, d\}, u'_i - u_i > \gamma_{i1}$

$\Rightarrow \nexists u' \in U, \forall i \in \{1, \cdots, d\}, \mu_{MP_i^{(\gamma_{i1},\gamma_{i2})}}(u'_i, u_i) > 0$

$\Rightarrow \nexists u' \in U, \forall i \in \{1, \cdots, d\}, MP_i^{(\gamma_{i1},\gamma_{i2})}(u'_i, u_i) \Rightarrow u \in S_{relax}^{(\gamma)} \Rightarrow S_{relax}^{(\gamma')} \subseteq S_{relax}^{(\gamma)}$

□

Lemma 1. *Let* $\gamma = ((0, \gamma_{12}), \cdots, (0, \gamma_{d2}))$ *and* $\gamma' = ((\gamma'_{11}, \gamma'_{12}), \cdots, (\gamma'_{d1}, \gamma'_{d2}))$, *the following holds:* $S_{relax}^{(0)} \subseteq S_{relax}^{(\gamma)} \subseteq S_{relax}^{(\gamma')}$

Example 2. Let us come back to the skyline calculated in Example 1. Assume that the *"much preferred"* relations corresponding to the skyline attributes (man_exp and tec_exp) are respectively given by:

$$\mu_{MP_{man\_exp}^{(1/2,2)}}(u', u) = \begin{cases} 0 & \text{if } u' - u \leq 1/2 \\ 1 & \text{if } u' - u \geq 2 \\ 2/3(u' - u) - 1/3 & \text{else} \end{cases} \quad (6)$$

$$\mu_{MP_{tec\_exp}^{(1/2,4)}}(u', u) = \begin{cases} 0 & \text{if } u' - u \leq 1/2 \\ 1 & \text{if } u' - u \geq 4 \\ 2/7(u' - u) - 1/8 & \text{else} \end{cases} \quad (7)$$

Now, applying the $\mathcal{MP2R}$ approach, to relax the skyline $S = \{M_5, M_8\}$ found in example 1, leads to the following relaxed skyline $S_{relax} = \{(M_5, 1), (M_8, 1), (M_3, 0.85), (M_{10}, 0.85), (M_1, 0.66), (M_2, 0.66), (M_4, 0.57)\}$, see Table 2.

Table 2 Degrees of the elements of S_{relax}

Mat	M5	M8	M3	M10	M1	M2	M4	M6	M7	M9
$\mu_{S_{relax}}$	1	1	0.85	0.85	0.66	0.66	0.57	0	0	0

One can note that some candidates that were not in S are now elements of S_{relax} (such M_{10} and M_4) see Fig. 3 . As can be seen, S_{relax} is larger than S. Let us now take a glance at the content of S_{relax}, one can observe that (i) the skyline elements of S are still elements of S_{relax} with a degree equal to 1 ; (ii) Appearance of new elements recovered by our approach whose degrees are less than 1 (such as M_3). Interestingly, the user can select from S_{relax}:

- the Top-k elements (k is a user-defined parameter) : elements of S_{relax} with highest degrees, or
- the subset of elements , denoted $(S_{relax})_\sigma$, with a degrees higher than a threshold σ provided by the user.

In the context of example 2, it is easy to check that $Top - 6 = \{(M_5, 1), (M_8, 1), (M_3, 0.85), (M_{10}, 0.85), (M_1, 0.66), (M_2, 0.66)\}$ and $(S_{relax})_{0.7} = \{(M_5, 1), (M_8, 1), (M_3, 0.85), (M_{10}, 0.85)\}$.

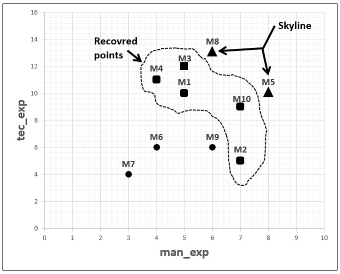

Fig. 3 Skyline relaxation

4 S_{relax} Computation

According to the definition of S_{relax}, a tuple u belongs to S_{relax} if: it does not exist a tuple u' which is *much preferred* to u w.r.t. all skyline attributes. So, to compute S_{relax}, we proceed in two steps (see Fig. 4):

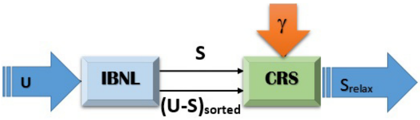

Fig. 4 The process of skyline relaxation

First, the classical skyline (S) is computed using an Improved Basic Nested Loop algorithm ($IBNL$), see algorithm 1. The dataset U is sorted in ascending order by using a monotonic function Mf (e.g., the sum of attributes skyline multiplying by -1 attribute values whose criterion is MAX). Sorting allows the following property:

$$\forall u, v \in U \mid Mf(u) \leq Mf(v) \Longrightarrow \neg(v \succ u) \tag{8}$$

SkylineCompare function evaluates the dominance , in the sense of Pareto, between u_i and u_j on all skyline dimensions and returns the result in status. It may be equal to: 0 if $u_i = u_j$, 1 if $u_i \succ u_j$, 2 if $u_i \prec u_j$, 3 if they are incomparable.
Secondly, we introduce an efficient algorithm called CRS (Computing Relaxed Skyline) to relax S using a vector of parameters γ (see algorithm 2).

Algorithm 1. IBNL

Input: A set of tuples U
Output: A skyline S

```
Sort(U);
for i := 1 to n − 1 do
    if ¬u_i.dominated then
        for j := i + 1 to n do
            status = 0;
            if ¬u_j.dominated then
                evaluate SkylineCompare(u_i,u_j,status);
                switch status do
                    case 1
                        u_i.dominated= true;
                    case 2
                        u_j.dominated= true;
        if ¬u_i.dominated then
            S = S ∪ {u_i};
return S;
```

Algorithm 2. CRS

Input: A set of tuples U; Skyline S; γ a vector of parameters;
Output: A relaxed skyline S_{relax};

```
begin
    for i = 1 to n do
        if u_i ∉ S then
            for j = 1 to n do
                for k = 1 to d do
                    evaluate μ_{MP_k}(u_i, u_j);
                compute min_k(μ_{MP_k});
            compute max_j(min_k(μ_{MP_k})); μ_{S_relax}(u_i) = 1 − max_j(min_k(μ_{MP_k}));
        if μ_{S_relax}(u_i) > 0 then
            S_relax = S_relax ∪ {u_i};
    rank u_i in decreasing order w.r.t. μ_{S_relax}(u_i);
    return S_relax;
```

5 Experimental Study

In this section, we present the experimental study that we have conducted. The goal of this study is to prove the effectiveness of $\mathcal{MP2R}$ and its ability to relax small skylines with the most interesting tuples.

5.1 Experimental Environment

All experiments were performed under Linux OS, on a machine with an Intel core i7 2,90 GHz processor, a main memory of 8 GB and a 250 GB of disk. All algorithms were implemented with Java. Dataset benchmark is generated using method described in [2] following three conventional distribution schema (correlated, anti-correlated and independent). For each dataset, we consider three different sizes (10K, 50K and 100K). Each tuple contains an integer identifier (4 bytes), 8 decimal fields (8 bytes) with values belonging to the interval [0,1], and a string field with length of 10 characters. Therefore, the size of one tuple is 78 bytes.

5.2 Experimental Results

We vary a collection of parameters that impact the result. This collection includes the dataset size [D] (1OK, 50K, 100K), dataset distribution schema [DIS] (independent, correlated, anti-correlated), the number of dimensions [d] (2, 4, 6, 8) and the relaxation thresholds [$\gamma = (\gamma_{i1}, \gamma_{i2})$ for i$\in \{1, \ldots, d\}$] where (γ_{i1}, γ_{i2} $\in$[0,1] and $\gamma_{i1} \le \gamma_{i2}$). These parameters are set as follows: D=10K; DIS="Correlated"; d=2; γ=((0.25,0.5),(0.25,0.5)). In our experiment, we consider that the less the value, the better is. Thus, this study addresses the following points:

– Impact of [DIS] on the size of relaxed skyline and on the computation time,
– Impact of [d] on the size of relaxed skyline and on the computation time,
– Impact of [D] on the size of relaxed skyline and on the computation time,
– Impact of $(MP_i^{(\gamma_{i1},\gamma_{i2})})$ on the size of relaxed skyline.

Impact of DIS. Fig. 5 shows that for anti-correlated and independent Distribution, our approach provides more tuples than the correlated data context. This is due to the type of data. We observe that the efficiency of CRS to relax the skyline is very high for all types of data. We note that the execution time of the CRS, for the three distributions, is almost similar.

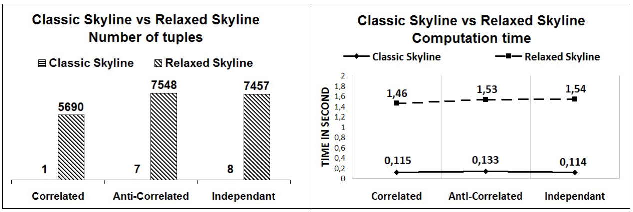

Fig. 5 Impact of [DIS].

Impact of the Number of Dimension. It is well-known that the number of dimensions increases the classic skyline; this phenomenon is known as "the problem of dimensionality". The CRS leads to the same behavior where data are highly correlated (correlation coefficient = 0.9952) see Fig. 6. From 2 to 8 dimensions, the relaxed skyline size changes from 5690 to 7519 tuples and the execution time increases from 1.46 to 2.45 second.

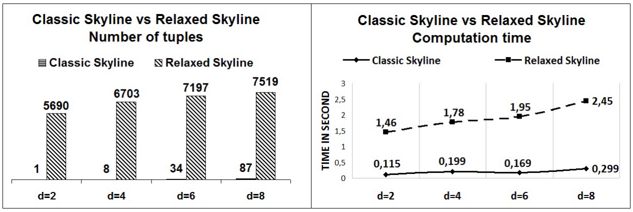

Fig. 6 Impact of [d]

Impact of the Dataset Size. Fig. 7 shows that the size of relaxed skyline is proportional to the data size, which confirms CRS's ability to relax the skyline with respect to the size of Dataset but it is time consuming.

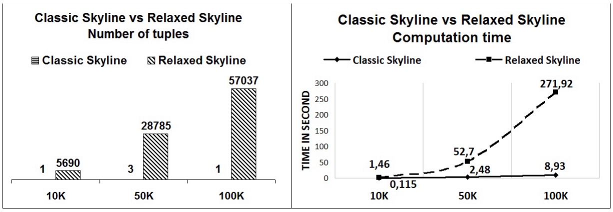

Fig. 7 Impact of [D]

Impact of $(MP_i^{(\gamma_1,\gamma_2)})$ Dominance Relationship. Now, we show the influence of the "Much Preferred" dominance relationship $(MP_i^{(\gamma_1,\gamma_2)})$ on the size of relaxed skyline.Tthe idea is to vary both thresholds (γ_{i1} and γ_{i2}) of the relationship. For the sake of simplicity, and since the data are normalized, we will apply the same function $MP_i^{(\gamma_1,\gamma_2)}$ for all skyline dimensions. Note that the size of the skyline is equal to 1 and we will analyze the variation of the number of tuples whose degree $\mu_{S_{relax}}(u) > 0$. The following scenarios are worth to be discussed :

Scenario 1: In this scenario, we fix γ_{i1} and vary γ_{i2} to increase the relaxation zone. We observe the following cases:

Case 1: $\gamma_{i1} = 0$ and $\gamma_{i2} \in \{0.25; 0.5; 0.75; 1\}$. Fig. 8 shows the results obtained. The analysis of these curves shows that the size of relaxed skyline increases when the value of γ_{i2} increases. We also note that there are no tuples whose degrees of relaxation is equal to 1 (this is due to the value of $\gamma_{i1} = 0$)

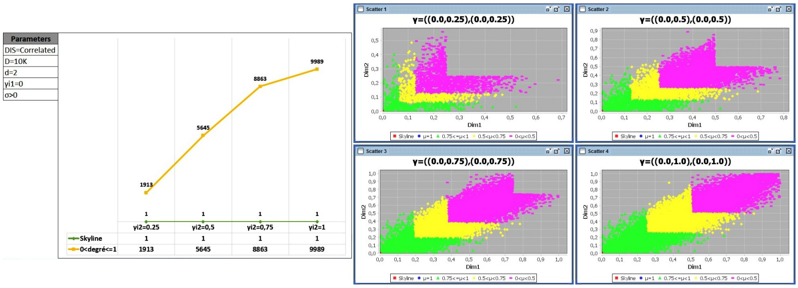

Fig. 8 Fix γ_{i1} and vary γ_{i2} **(case 1)**

Case 2: $\gamma_{i1} = 0.25$ and $\gamma_{i2} \in \{0.25; 0.5; 0.75; 1\}$. In this case, the size of relaxed skyline increases also when the value of γ_{i2} increases. We note the appearance of tuples whose degree is equal to 1 (see Fig. 9).

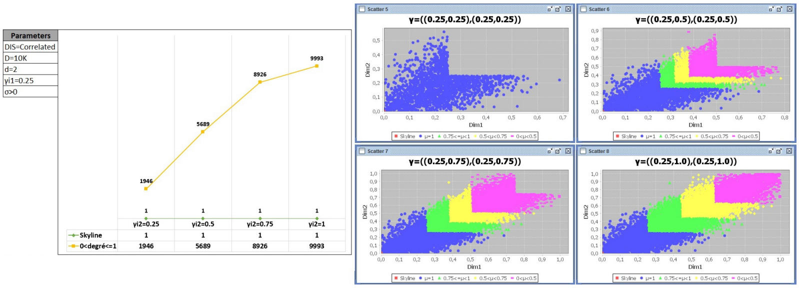

Fig. 9 Fix γ_{i1} and vary γ_{i2} **(case 2)**

Case 3: $\gamma_{i1} = 0.5$ and $\gamma_{i2} \in \{0.5; 0.75; 1\}$.. The same results are obtained in this case but the number of tuples whose degrees are equal to 1 is more important than in the case 2 (see Fig. 10).

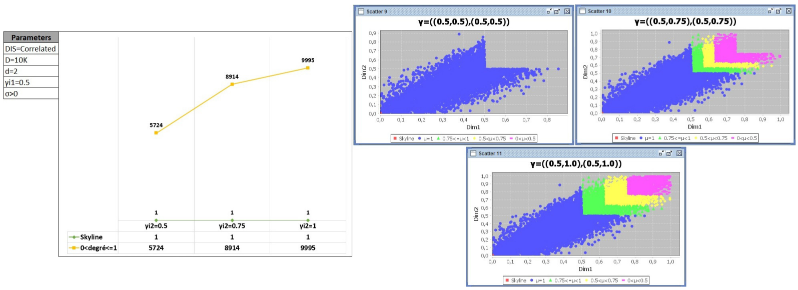

Fig. 10 Fix γ_{i1} and vary γ_{i2} **(case 3)**

Case 4: $\gamma_{i1} = 0.75$ and $\gamma_{i2} \in \{0.75; 1\}$. The same results are obtained in this case but the number of tuples whose degrees are equals to 1 is more important than in the cases 2 and 3 (see Fig. 11). This means that more the values of γ_{i1} and γ_{i2} are close to 1, the larger the number of tuples whose degrees are equal to 1.

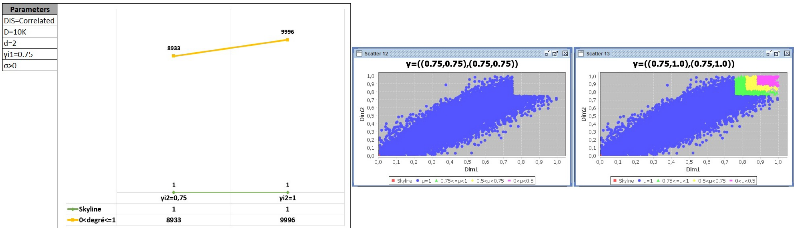

Fig. 11 Fix γ_{i1} and vary γ_{i2} **(case 4)**

Scenario 2: In this scenario, we vary both thresholds. The obtained results are shown in Fig. 12. The analysis of these curves shows that the relaxation function becomes more permissive when thresholds move away from the origin.

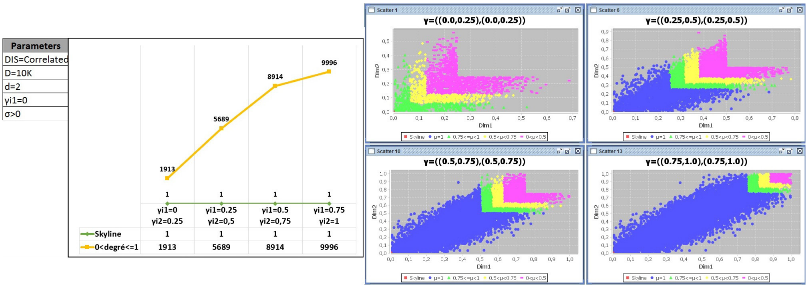

Fig. 12 Varying γ_{i1} and γ_{i2}

Finally, the choice of values for $\gamma = (\gamma_{i1}, \gamma_{i2})$ is extremely important in the relaxation process. This choice also depends on the domains values of the different dimensions used in the skyline.

6 Conclusion

In this paper, we addressed the problem of skyline relaxation, especially less skylines. An approach for relaxing the skyline, called $MP2R$, is discussed. The key concept of this approach is a particular relation named *much preferred* whose semantics is user-defined. In addition, a new algorithm called CRS to compute the relaxed skyline is proposed. The experimental study we done has shown that, on the one hand, the $MP2R$ approach is a good alternative to tackle the relaxation issue of classic skyline and, on the other hand, the computation cost of S_{relax} is quite reasonable. $MP2R$ involves various parameters, which can be used to control the size of the relaxed skyline.

As for future works, we will explore the possibility of using multidimensional index (R-Tree and its variants) to accelerate the computation of S_{relax}.

References

1. Abbaci, K., Hadjali, A., Lietard, L., Rocacher, D.: A linguistic quantifier-based approach for skyline refinement. In: IFSA/NAFIPS, pp. 321–326 (2013)
2. Börzsönyi, S., Kossmann, D., Stocker., K.: The skyline operator. In: ICDE, pp. 421–430 (2001)
3. Chan, C.Y., Jagadish, H.V., Tan, K., Tung, A.K.H., Zhang, Z.: Finding k-dominant skylines in high dimensional space. In: ACM SIGMOD, pp. 503–514 (2006)
4. Chan, C.-Y., Jagadish, H.V., Tan, K.-L., Tung, A.K.H., Zhang, Z.: On high dimensional skylines. In: Ioannidis, Y., Scholl, M.H., Schmidt, J.W., Matthes, F., Hatzopoulos, M., Böhm, K., Kemper, A., Grust, T., Böhm, C. (eds.) EDBT 2006. LNCS, vol. 3896, pp. 478–495. Springer, Heidelberg (2006)
5. Chomicki, J., Ciaccia, P., Meneghetti, N.: Skyline queries, front and back. SIGMOD Record (2013)
6. Endres, M., Kießling, W.: Skyline snippets. In: Christiansen, H., De Tré, G., Yazici, A., Zadrozny, S., Andreasen, T., Larsen, H.L. (eds.) FQAS 2011. LNCS, vol. 7022, pp. 246–257. Springer, Heidelberg (2011)
7. Godfrey, P., Shipley, R., Gryz, J.: Maximal vector computation in large data sets. In: VLDB (2005)
8. Goncalves, M., Tineo, L.J.: Fuzzy dominance skyline queries. In: Wagner, R., Revell, N., Pernul, G. (eds.) DEXA 2007. LNCS, vol. 4653, pp. 469–478. Springer, Heidelberg (2007)
9. HadjAli, A., Pivert, O., Prade, H.: Possibilistic contextual skylineswith incomplete preferences. In: SoCPaR, pp. 57–62
10. Hadjali, A., Pivert, O., Prade, H.: On different types of fuzzy skylines. In: Kryszkiewicz, M., Rybinski, H., Skowron, A., Raś, Z.W. (eds.) ISMIS 2011. LNCS, vol. 6804, pp. 581–591. Springer, Heidelberg (2011)
11. Hüllermeier, E., Vladimirskiy, I., Prados Suárez, B., Stauch, E.: Supporting case-based retrieval by similarity skylines: basic concepts and extensions. In: Althoff, K.-D., Bergmann, R., Minor, M., Hanft, A. (eds.) ECCBR 2008. LNCS (LNAI), vol. 5239, pp. 240–254. Springer, Heidelberg (2008)
12. Khalefa, M.E., Mokbel, M.F., Levandoski, J.J.: Skyline query processing for incomplete data. In: IEEE ICDE, pp. 556–565 (2008)
13. Kossmann, D., Ramsak, F., Rost, S.: Shooting stars in the sky: an online algorithm for skyline queries. In: VLDB, pp. 275–286 (2002)
14. Lin, X., Yuan, Y., Zhang, Q., Zhang, Y.: Selecting stars: the k most representative skyline operator. In: ICDE (2007)

15. Papadias, D., Tao, Y., Fu, G., Seeger, B.: An optimal and progressive algorithm for skyline queries. In: ACM SIGMOD (2003)
16. Pei, J., Jiang, B., Lin, X., Yuan, Y.: Probabilistic skylines on uncertain data. In: VLDB, pp. 15–26 (2007)
17. Tan, K., Eng, P., Ooi, B.C.: Efficient progressive skyline computation. In: VLDB, pp. 301–310 (2001)
18. Yiu, M.L., Mamoulis, N.: Efficient processing of top-k dominating queries on multi-dimensional data. In: VLDB, pp. 483–494 (2007)
19. Zadeh, L.A.: Fuzzy sets. Information and Control, 338–353 (1965)