A Formal-Concept-Lattice Driven Approach for Skyline Refinement

Mohamed Haddache[1(✉)], Allel Hadjali[2(✉)], and Hamid Azzoune[3(✉)]

[1] DIF-FS/UMBB, Boumerdes, Algeria
mohamed.haddache@ensma.fr
[2] LIAS/ENSMA, Poitiers, France
allel.hadjali@ensma.fr
[3] LRIA/USTHB, Algers, Algeria
azzoune@yahoo.fr

Abstract. Skyline queries constitute an appropriate tool that can help users to make intelligent decisions in the presence of multidimensional data when different, and often contradictory criteria are to be taken into account. Based on the concept of Pareto dominance, the skyline process extracts the most interesting (not dominated in sense of Pareto) objects from a set of data. However, this process often leads to a huge skyline, which is less informative for the end-users. In this paper, we propose an efficient approach to refine the skyline and reduce its size, using the principle of the formal concepts analysis. The basic idea is to build a formal concept lattice for skyline objects based on the minimal distance between each concept and the target concept. We show that the refined skyline is given by the concept that contains k objects (where k is a user-defined parameter) and has the minimal distance to the target concept. A set of experiments are conducted to demonstrate the effectiveness and efficiency of our approach.

Keywords: Skyline queries · Skyline refinement · Pareto dominance · Lattice of formal concepts

1 Introduction

The skyline queries are introduced by Borzsönyi in [4] to formulate multi-criteria searches. Recently, this concept, has gained much attention in the database community. It has been integrated in many database applications that require decision making and personalized services. Skyline process attempts to identify the most interesting (not dominated in sense of Pareto) objects from a set of data. Skyline queries are based on Pareto dominance relationship. This means that, given a set D of d-dimensional points (objects), a skyline query returns, the skyline S, set of points of D that are not dominated by any other point (object) 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.

F. Wotawa et al. (Eds.): IEA/AIE 2019, LNAI 11606, pp. 541–554, 2019.
https://doi.org/10.1007/978-3-030-22999-3_47

A great research effort has been devoted to develop efficient algorithms to skyline computation [12,14,17,20,24,29]. The skyline computation often leads to a huge number of skyline objects which is less informative for the user and does not bring any insight to decision making. In order, to solve this problem and reduce the size of skyline, several algorithms have been developed [2,6,7,9, 13,18,21,23,26]. In this paper, we consider this problem, but with another novel vision. In particular, the idea of the solution advocated is borrowed from the formal concept analysis field. This idea consists in building a formal concept lattice for skyline objects based on the minimal distance between each concept and the target concept (i.e., the ideal object w.r.t the user query). The refined skyline S_{ref} is given by the concept that has the minimal distance to the target concept and contains k objects (k is a parameter given by the user). Starting from this idea, we develop an algorithm to compute the refined skyline, called FLCMD. In summary, our main contributions cover the following points:

- We define an efficient approach to refine the skyline S based on the minimal distance between the concepts lattice and the target concept.
- We develop and implement an algorithm to compute S_{ref} efficiently.
- We conduct a set of thorough experiments to study, analyze and compare the relevance and effectiveness of proposed approach and the naive method.

This paper is structured as follows. In the Sect. 2, we define some necessary notions about the skyline queries, fuzzy set theory, the formal concept analysis and lattice then, we report some works related to the skyline refinement and at the end of this section, we explain the naive approach. In Sect. 3, we present our approach and we give the FLCMD algorithm that compute the refined skyline S_{ref}. Section 4 is dedicated to the experimental study and Sect. 5 concludes this paper and points out some future work.

2 Background and Related Work

2.1 Skyline Queries

Skyline queries [4] represent a very popular and powerful paradigm to extract objects from a multidimensional dataset. They are based 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 strictly better than u_j in at least one dimension. [25]*

Formally, we write:

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

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

In Eq. (1), without loss of generality, we assume that the minimal 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.*

$$u \in S \Leftrightarrow \nexists u' \in D, u' \succ u \tag{2}$$

Example 1. To illustrate the concept of the Skyline, let us consider a database containing information about apartments as shown in Table 1. The list of apartments includes the following information: code apartment, area of apartment (m^2), price in (€) and distance between work and home (apartment) (*dist_wh* in km). Ideally, a person is looking to rent an apartment with a minimal price and having a minimal distance to his/her work (price and *dist_wh*), ignoring the other pieces of information. Applying the traditional skyline on the apartments list of Table 1, returns the following apartments: $\{A_1, A_3, A_5, A_7\}$, see Fig. 1.

Table 1. List of apartments

Code	Area (m^2)	Price (€)	*dist_wh* (km)
A_1	60	525	20
A_2	40	400	120
A_3	25	360	85
A_4	30	380	100
A_5	25	340	90
A_6	60	550	95
A_7	65	540	10

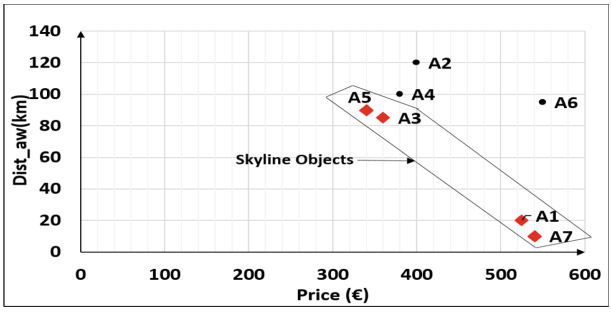

Fig. 1. Skyline of apartments

2.2 Fuzzy Set Theory

The concept of fuzzy sets has been developed by Zadeh [30] in 1965 to represent classes or sets whose limits are imprecise. They can describe gradual transitions

between total belonging and rejection. Formally, a fuzzy set F on the universe X is described by a membership function $\mu_F : X \rightarrow [0, 1]$, 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, $\mu_F(x) = 1$ then x completely belongs to F, these elements form the core of F denoted by $Cor(F) = \{x \in F \backslash \mu_F(x) = 1\}$. When $0 < \mu_F(x) < 1$ we talk about a partial membership, these elements form the support of F denoted by $supp(F) = \{x \in F \backslash \mu_F(x) > 0\}$. Moreover, $\mu_F(x)$ is closed 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 can be represented by a trapezoid membership function $(t.m.f)$ $(\alpha, \beta, \varphi, \psi)$ where $[\beta, \varphi]$ is the core and $]\alpha, \psi[$ is its support see Fig. 2.

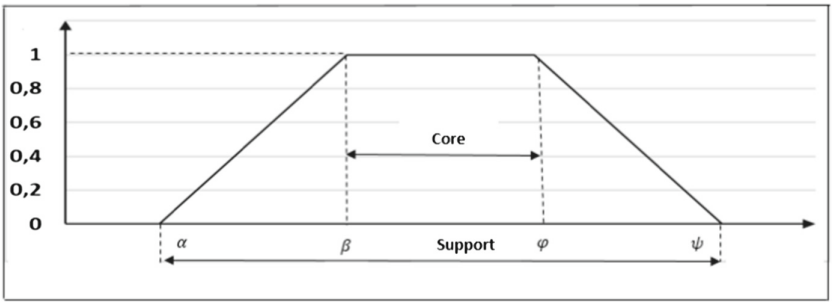

Fig. 2. Trapezoidal fuzzy set.

2.3 Formal Concept Analysis

The theory of formal concept analysis (FCA), proposed by Wille in 1982 [28]. It is based on a formal context $\mathcal{K} = (O, P, R)$, where O is a set of objects, P is a set of properties (attributes) and R a binary relation between O and P. Wille defined a correspondence between sets O and P. These correspondences are called a Galois derivation operator (or sufficiency operator) noted by $\triangle$. Given $A \subset O$, $B \subset P$, $A^{\triangle}$ express all the properties satisfied by all the objects of A and dually $B^{\triangle}$ express the set of objects satisfying all the properties of B (see [28]). The dual pair of operators $((.)^{\triangle},(.)^{\triangle})$ constitutes a Galois connection which allows to introduce formal concepts. A formal concept of a formal context $\mathcal{K}$ is a pair (A, B) with $A \subset O$, $B \subset P$, $A^{\triangle} = B$ and $B^{\triangle} = A$. A and B are respectively called extent and intent of the formal concept (A, B). The set of all formal concepts is equipped with a partial order denoted $\preceq$ defined by: $(A_1, B_1) \preceq (A_2, B_2)$ iff $A_1 \subseteq A_2$ or $B_2 \subseteq B_1$. Ganter and Wille have proved in [10] that, the set of all formal concepts ordered by $\preceq$ forms a complete lattice of the formal context $\mathcal{K}$ denoted by $\mathcal{L}(\mathcal{K})$. In most applications, like in our case the attributes are defined a fuzzy way. In order to take into account relations allowing a gradual satisfaction of a property by an object, a fuzzy FCA was proposed by Burusco and Fuentes-Gonzales [5] and belohlávek et al. in [3]. In

this case, the notion of satisfaction can be expressed by a degree $\in [0,1]$. A fuzzy context formal is a tuple (L, O, P, R), where a fuzzy relation $R \in L^{O\times P}$ is a function that is defined $O \times P \longrightarrow L$ which assigns to each object $o \in O$ and for each property $p \in P$ a degree $R(o,p)$ for which the object o has the property p. In general $L = [0,1]$. The generalization of the Galois derivation operator, to fuzzy settings is based on the fuzzy implication defined by belohlávek in [3]. It is defined for an subset $A \in L^O$ (and similarly defined for an subset $B \in L^P$) as follows:

$$A^{\Delta}(p) = \bigwedge_{o\in O} (A(o) \rightarrow R(o,p)) \tag{3}$$

$$B^{\Delta}(o) = \bigwedge_{p\in P} (B(p) \rightarrow R(o,p)) \tag{4}$$

$\rightarrow$: is a fuzzy implication that verify $(0 \rightarrow 0 = 0 \rightarrow 1 = 1 \rightarrow 1 = 1$ and $1 \rightarrow 0 = 0)$. We distinguish three type of fuzzy formal concepts. Concept with crisp extent and fuzzy intent, crisp extent and fuzzy intent the third type fuzzy extent and fuzzy intent.

In this paper, we use concept with crisp extent and fuzzy intent, i.e., the set of objects is crisp and the set of properties is fuzzy.

Example 2. To illustrate the computation of formal concepts in our case, let us consider a database containing information about hotels as shown in Table 2. The set of objects O is composed by different hotels $\{h_1, h_2, h_3\}$, the set of properties P contains the properties *cheap* (denoted Ch) and *Near the beach* (denoted Nb), i.e., $P = \{Ch, Nb\}$. $R(o_i, p_j)$ represents the degree for witch the object o_i satisfies the property p_j, for example $R(h_2, ch) = 0.5$ means that the hotel h_2 satisfies the property *cheap* with degree 0.5. Let us consider the sets of objects $A_1 = \{h_2, h_3\}$, $A_2 = \{h_2\}$ and the set of properties $B_1 = \{Ch^{0.5}, Nb^{0.5}\}$. Now, let us describe how to compute $(A_1)^{\Delta}$, $(A_2)^{\Delta}$ and $(B_1)^{\Delta}$. For $(A_1)^{\Delta}$ and $(A_2)^{\Delta}$, we use Eq. (3) and the implication of Gödel defined by

$$p \longrightarrow q = \begin{cases} 1 & \text{if } p \leq q \\ q & \text{else} \end{cases} \tag{5}$$

Table 2. List of hotels

Hotel	Cheap (Ch)	Near the beach (Nb)
h_1	0.0	0.8
h_2	0.5	0.5
h_3	0.5	0.6

$A_1 = \{h_2, h_3\} = \{h_1^0, h_2^1, h_3^1\}$
$(A_1)^{\triangle}(Ch) = \wedge(0 \to 0,\ 1 \to 0.5,\ 1 \to 0.5) = \wedge(1,\ 0.5,\ 0.5) = 0.5$
$(A_1)^{\triangle}(Nb) = \wedge(0 \to 0.8,\ 1 \to 0.5,\ 1 \to 0.6) = \wedge(1,\ 0.5,\ 0.6) = 0.5$
$(A_1)^{\triangle} = \{Ch^{0.5}, Nb^{0.5}\} = B_1$
Similarly, we obtain $(A_2)^{\triangle} = \{Ch^{0.5}, Nb^{0.5}\} = B_1$

To compute $(B_1)^{\triangle}$, we use Eq. (4) and the implication of Rescher Gaines defined by

$$p \longrightarrow q = \begin{cases} 1 & \text{if} p \leq q \\ 0 & \text{else} \end{cases} \tag{6}$$

$(B_1)^{\triangle}(h1) = \wedge(0.5 \to 0,\ 0.5 \to 0.8) = \wedge(0,\ 0.5) = 0$
$(B_1)^{\triangle}(h2) = \wedge(0.5 \to 0.5,\ 0.5 \to 0.5) = \wedge(1,\ 1) = 1$
$(B_1)^{\triangle}(h3) = \wedge(0.5 \to 0.5,\ 0.5 \to 0.6) = \wedge(1,\ 1) = 1$
$(B_1)^{\triangle} = \{h_1^0, h_2^1, h_3^1\} = \{h_2, h_3\} = A_1$.

- $(A_1)^{\triangle} = B_1$ and $(B_1)^{\triangle} = A_1$, this means that (A_1, B_1) forms a fuzzy formal concept, A_1 is its extent and B_1 its intent.
- $(A_2)^{\triangle} = B_1$ but $(B_1)^{\triangle} = \{h_2, h_3\} \neq A_2$ then, (A_2, B_1) is not a fuzzy formal concept.

2.4 Related Work

The work proposed by Börzsönyi and al. in [4] is the first work that addresses the issue of skyline queries in the database field. They have proposed two different algorithms to process skyline queries in complete database, namely, Block Nested Loop (BNL) and Divide and Conquer (D& C). Later, many algorithms have been developed which are inspired from BNL and D&C [4,8,19,23,27,27]. Several authors have been interested in the problem of huge skyline and have proposed additional mechanisms to refine the skyline and reduce its size.

In [2,7,13,18,21,23,26] ranking functions are used to refine the skyline. The idea of these approaches is to combine the skyline operator with the top-K operator. For each tuple in the skyline, one joins a related score, which is computed by the means of ranking function F. We note that F must be monotonic on all its arguments. Skyline tuples are ordered according to their scores, and the top-K tuples will be returned.

In [11] authors, propose the notion of fuzzy skyline queries, which replaces the standard comparison operators $(=, <, >, \leq, \geq)$ with fuzzy comparison operators defined by user. While in [15], Hadjali and al. have proposed some ideas to introduce an order between the skyline points in order to single out the most interesting ones. In [1], a new definition of dominance relationship based on the fuzzy quantifier "almost all" is introduced to refine the skyline, while in [16] authors, introduce a strong dominance relationship that relies on the relation called "much preferred". This leads to a new extension of skyline, called MPS (Must Preferred Skyline), to find the most interesting skyline tuples. In [22] authors, propose a flexible approach called "$\theta - skyline$" to categorize and

refine the skyline set by applying successive relaxations of the dominance conditions with respect to the user's preferences. This approach is based on the ranking method which deals with decision-making in the presence of conflicting choices. Furthermore, they define a global ranking method over the skyline set. In [13], Haddache et al. have proposed an approach based on ELECTRE method borrowed from the outranking domain to refine the skyline.

Furthermore, several researchers have worked on skyline's refinement for the evidential data. In [9] authors, have developed efficient algorithms to retrieve the best evidential skyline objects over uncertain data.

2.5 Naive Method

This approach [6] is based on two steps: (i) first compute for each skyline point p, the number of points dominated by p denoted by $num(p)$. (ii) The skyline points are sorted according to $num(p)$ in order to choose the $Top-k$.

3 Our Approach

In this section, we will present the main steps of our approach. First, we assume that, we have

- A database formed by a set of m objects (tuples), $O = \{o_1, o_2, \cdots, o_m\}$.
- A set P of n properties (or dimensions or attributes), $P = \{p_1, p_2, \cdots, p_n\}$.
- Each object o_j from the set O is evaluated for every property p_i.
- S the skyline of O, $S = \{o_1, o_2, \cdots, o_t\}$, $t <= m$, t is the size of skyline.
- S_{ref} the refined skyline returned by our approach.
- In our approach we use the implication($\longrightarrow$) of Rescher Gaines defined by Eq. (6).

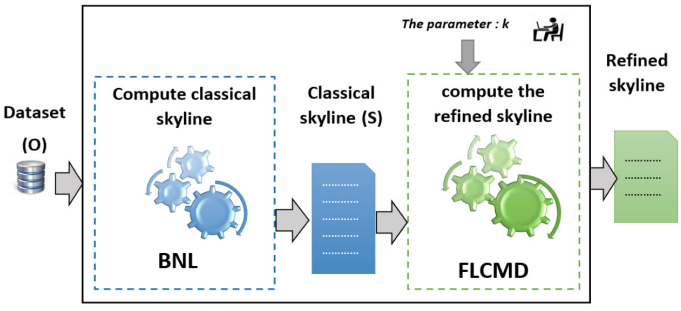

Fig. 3. Steps of our approach

The principle of our approach is to build the fuzzy concept lattice of the skyline points based on the minimal distance between each new concept and the target concept. In summary, our approach is based on the following steps (see Fig. 3).

Algorithm 1. FLCMD

Input: A Skyline S, K: the number of objects chosen by the user
Output: A refined skyline S_{ref}

$S_{ref} \leftarrow \emptyset$; $dist \leftarrow 100$; $stop \leftarrow false$;
$Compute\_degree(S)$; /*compute the degrees of objects*/;
$Intent\_target \leftarrow Compute\_target\_intent()$;
for $i := 1$ ***to*** n **do**
  $Intent\_min(i) \leftarrow S(1, i)$;
  for $j := 2$ ***to*** m **do**
    if $S(j, i) < Intent\_min(i)$ **then**
      $Intent\_min(i) \leftarrow S(j, i)$;
    end
  end
end
while $stop = false$ **do**
  for $i := 1$ ***to*** nb_d **do**
    $New\_Intent \leftarrow Next\_intent(Intent\_min, i)$;;
    $d \leftarrow Compute\_distance(New\_Intent, Intent\_target)$;
    if $(d < dist)$ **then**
      $dist = d$; $save\_Intent \leftarrow New\_Intent$;
      $extent \leftarrow Compute\_Extent(New\_Intent)$;
      if $(size(extent) = k)$ **then**
        $S_{ref} \leftarrow extent$; $stop \leftarrow true$;
      end
    end
  end
  $Intent\_min \leftarrow save\_Intent$;
end
return S_{ref};

1. First, we calculate the skyline using the Basic Nested Loop algorithm (BNL) for more details see [4].
2. Second, we compute the refined skyline using Algorithm 1 (FLCMD). This algorithm, starts by computing for each object o_i the degree $R(o_i, p_j)$ for witch the o_i minimizes the property p_j chosen by the user. Then, it computes the formal concept whose intent minimizes the properties chosen by the user, i.e., maximizes the degrees $R(o_i, p_j)$ for these properties (this concept is called target concept).
3. FLCMD builds the fuzzy lattice for skyline objects. It starts by computing the formal concept whose intent minimizes the degrees $R(o_i, p_j)$ for the properties chosen by user,
4. The algorithm FLCMD computes all the following concepts of this concept.
5. For each new concept, FLCMD, computes the size of its extent and the distance between its intent and the intent of target concept.
6. If the size of the extent equals k (where k is a user-defined parameter), the process stopped. The refined skyline is given by the objects of this extent

(when the number of extents having a size equals k is greater than 1, FLCMD chooses the extent whose intent has the minimal distance).

7. If the size of the extent is greater than k, FLCMD selects the intent that has the minimal distance and it starts from step 4.

FLCMD algorithm uses the following functions:

- $Next\_intent(Intent\_min, i)$: gives the following intent of $Intent\_min$ on the dimension i.
- $Compute\_Extent(New\_Intent)$: computes the extent of New_Intent, using the equation (4) and the implication given by the Eq. (6).
- $Compute\_distance(New\_Intent, Intent\_target)$: computes the Euclidean distance between New_Intent and $Intent\_target$.

Example 3. To illustrate our approach, let us come back to the skyline calculated in Example 1 presented in Sect. 2.1. As a reminder, we use two properties, namely price and *dist_wh*. Furthermore, we assume that the minimal value, the better. **BNL** algorithm returns as skyline the following apartments: $\{A_1, A_3, A_5, A_7\}$, see Table 3.

Remark *In the following, we note the intent (price$^\alpha$, dist_wh$^\beta$)* by (α, β).

Table 3. Classic skyline and objects degrees

Classic skyline				Object degrees $R(A_i, P_i)$	
Code	Area (m2)	Price (€)	*dist_wh*(*km*)	Price	*dist_wh*
A_1	60	525	20	0.075	0.875
A_3	25	360	85	0.9	0.0625
A_5	25	340	90	1	0
A_7	65	540	10	0	1

First, we compute for each object skyline A_i the degree $R(A_i, P_i)$ for which A_i minimizes the property P_i. These degrees are given by (see Fig. 4).

- $R(A_i, price) = 1 - (x_1 - 340)/200$, x_1 is the value of A_i w.r.t property *price*.
- $R(A_i, dis\_wh) = 1 - (x_2 - 10)/80$, x_2 is the value of A_i w.r.t property *dis_wh*.

Second, we compute the target intent and the intent that minimizes the degree $R(A_i, P_i)$ w.r.t cheap price and short distance. Using data from Table 3, and the Algorithm 1, one cane observe that $Intent\_target = (1, 1)$ $Intent\_min = (0, 0)$. Then, we compute the following intents of the intent_min.

For $i = 1$, $New\_Intent = (0.075, 0)$. The distance between this intent and the target intent $d = \sqrt{(1 - 0.075)^2 + (1 - 0)^2} = 1.36$, extent $= (A_1, A_3, A_5)$.

For $i = 2$, $New\_Intent = (0, 0.0625)$. The distance between this intent and the target intent $d = \sqrt{(1 - 0)^2 + (1 - 0.0625)^2} = 1.37$, extent$=(A_1, A_3, A_7)$.

If $k = 3$, the process stopped and $S_{ref} = \{A_1, A_3, A_5\}$.

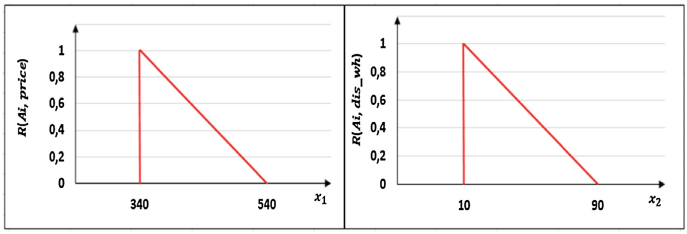

Fig. 4. Objects degrees

If $k < 3$, we select the intent $(0.075, 0)$ (because it has the minimal distance (1.36) with the target intent) then, we compute its following intents and the process continues as shown in Fig. 5. From Fig. 5, we can see that, if $k = 2$, the refined skyline equals $\{A_3, A_5\}$, when $k = 1$ $S_{ref} = \{A_3\}$.

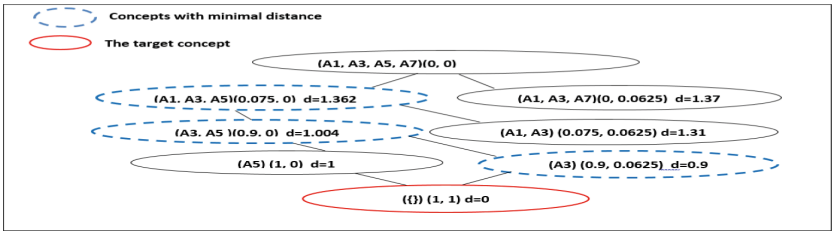

Fig. 5. Lattice of skyline points based on minimal distance

4 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 our algorithm and its ability to refine huge skyline and compare its relevance to the naive method. All experiments were performed under Windows OS, on a machine with an Intel core i7 2,90 GHz processor, a main memory of 8 GB and 250 GB of disk. All algorithms were implemented with Java. Dataset benchmark is generated using the method described in [4]. The test parameters used are distribution dataset [DIS] (correlated, anti-correlated and independent), the dataset size [D] (100K, 250K, 500K, 1000K, 2000K, 4000K) and the number of dimensions [d] (2, 4, 6, 10, 15). To interpret the results we define the following refinement rate (ref_rate):

$$ref\_rate = \frac{(ntcs - ntrs)}{(ntcs)} \tag{7}$$

where $ntcs$ is the number of tuples of the regular skyline and $ntrs$ is the number of tuples for the refined skyline.

Impact of [DIS]. In this case, we use a dataset with $|D| = 100K$, $d = 6$. Figure 6 shows that the execution time of the two algorithms for anti-correlated data is high compared to the correlated or independent data. This is due to the important number of tuples to refine (14758 tuples for anti-correlated data, 2184 and 89 tuples for independent and correlated data). Figure 6 shows also that our algorithm has the best execution time compared to the naive algorithm (0.004 s for FLCMD, 0.85 s for naive algorithm in the case of correlated data, 10.41 s for FLCMD and 72.32 s for the naive algorithm in the case of anti-correlated data, 0.38 s for FLCMD and 18.2 s for the naive algorithm in the case of independent data). The refinement rate for the two algorithms is very high (for correlated data $= (89-10)/89 = 0.88$, for anti-correlated data $= (14758-10)/14758 = 0.99$ and for independent data $= (2184-10)/2184 = 0.995$).

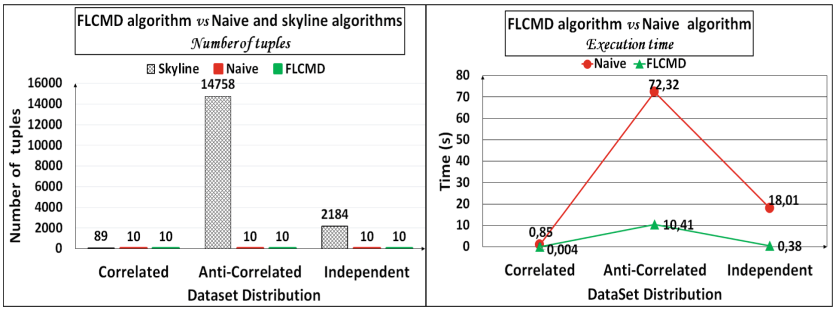

Fig. 6. Impact of [DIS]

Impact of the Size of the Dataset [D]. In this case, we study the impact of the size of the database on the execution time of the refined skyline and the refinement rate for the two algorithms. To do this, we use an anti-correlated database with $d = 4$. Figure 7, shows that, the execution time increases with the increase of the database size. But the execution time of our algorithm remains the best compared to the naive algorithm (the execution time increases from 0.3 s if $|D| = 100K$ to 10.52 s when $|D| = 4000K$ for FLCMD and from 13.48 s if

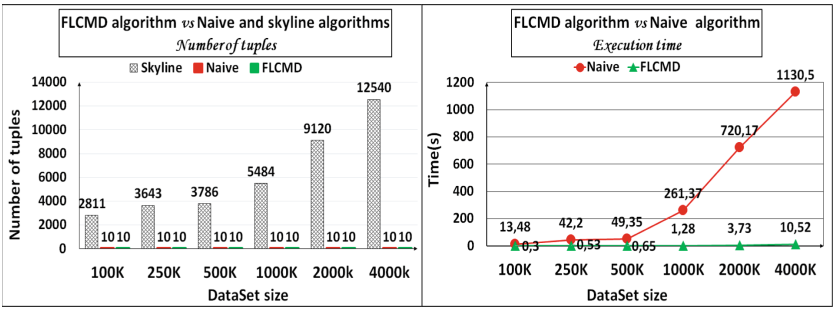

Fig. 7. Impact of [D]

$|D| = 100K$ to 1130.5 s if $|D| = 4000K$ for naive algorithm). The refinement rate for the two algorithms is very high varied from 0.996 $((2811-10)/2811 = 0.996)$ when $|D| = 100K$ to 0.999 $((12540-10)/12540 = 0.999)$ when $|D| = 4000K$.

Impact of the Number of Dimensions [d]. In this case, we study the impact of varying the number of dimensions skyline in the process of computing S_{ref}. We use an anti-correlated distribution data with $|D| = 50K$. Figure 8 shows that the execution time increases with the number of dimensions (from 0.008 s for $d = 2$ to 120.3 s when $d = 15$ for the FLCMD algorithm) and (between 0.5 s and 420 s when d varied from 2 to 15 for naive algorithm). This indicates that our algorithm gives the best execution time compared to naive algorithm. The refinement rate increases from 0.94 $((187-10)/187 = 0.94)$ when $d = 2$ to 0.99 $((48103-10)/48103 = 0.99)$ for $d = 15$.

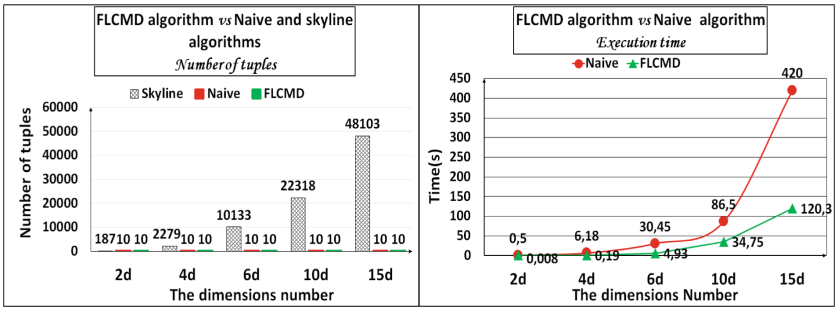

Fig. 8. Impact of [d]

5 Conclusion and Perspectives

In this paper, we addressed the problem of the skyline, especially a huge skyline and we proposed a new approach to reduce its size. The basic idea of this approach is to build a fuzzy concept lattice for skyline objects based on the minimal distance between each concept and the target concept. The process of refinement stopped when we compute the concept that contains k objects (where k is a user-defined parameter) and has the minimal distance with the target concept. The refined skyline is given by the objects of this concept. An algorithm called FLCMD to calculate the refined skyline is proposed. In addition, we implemented the naive algorithm to compare its performance to that of our algorithm. The experimental study we have done showed that, our approach is a good alternative to reduce the size of the classic skyline (the refinement rate reached 99%) and has a reasonable time computation also, the execution time of our algorithm is the best compared to the naive algorithm. As for future work, we will explore, on the one hand the use of semantic distance between concepts to build the refinement lattice and on the other hand, we will use the lattice construction algorithms that gives fuzzy extensions in order to sort the objects of the same concept.

References

1. Abbaci, K., Hadjali, A., Lietard, L., Rocacher, D.: A linguistic quantifier-based approach for skyline refinement. In: Joint IFSA World Congress and NAFIPS Annual Meeting, IFSA/NAFIPS, 24–28 June, Edmonton, Alberta, Canada, pp. 321–326 (2013)
2. Balke, W., Guntzer, U., Lofi, C.: User interaction support for incremental refinement of preference-based queries. In: Proceedings of the First Inter. Conference on Research Challenges in Information Science (RCIS), 23–26 April, Ouarzazate, Morocco, pp. 209–220 (2007)
3. Belohlávek, R.: Fuzzy galois connections. Math. Log. Q. **45**, 497–504 (1999)
4. Börzsönyi, S., Kossmann, D., Stocker, K.: The skyline operator. In: Proceedings of the 17th International Conference on Data Engineering, 2–6 April, Heidelberg, Germany, pp. 421–430 (2001)
5. Burusco, A., Fuentes-González, R.: The study of the l-fuzzy concept lattice. Mathware Soft Comput. **3**(1), 208–209 (1994)
6. Chan, C.Y., Jagadish, H.V., Tan, K., Tung, A.K.H., Zhang, Z.: Finding k-dominant skylines in high dimensional space. In: Proceedings of the International Conference on Management of Data (ACM SIGMOD), 27–29 June, Chicago, Illinois, USA, pp. 503–514 (2006)
7. Chan, C.-Y., Jagadish, H.V., Tan, K.-L., Tung, A.K.H., Zhang, Z.: On high dimensional skylines. In: Ioannidis, Y., et al. (eds.) EDBT 2006. LNCS, vol. 3896, pp. 478–495. Springer, Heidelberg (2006). https://doi.org/10.1007/11687238_30
8. Chomicki, J., Ciaccia, P., Meneghetti, N.: Skyline queries, front and back. SIGMOD Rec. **42**(3), 6–18 (2013)
9. Elmi, S., Tobji, M.A.B., Hadjali, A., Yaghlane, B.B.: Selecting skyline stars over uncertain databases: semantics and refining methods in the evidence theory setting. Appl. Soft Comput. **57**, 88–101 (2017)
10. Ganter, B., Wille, R.: Formal Concept Analysis. Mathematical Foundations. Springer, Heidelberg (1999). https://doi.org/10.1007/978-3-642-59830-2
11. Goncalves, M., Tineo, L.: Fuzzy dominance skyline queries. In: Wagner, R., Revell, N., Pernul, G. (eds.) DEXA 2007. LNCS, vol. 4653, pp. 469–478. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74469-6_46
12. Gulzar, Y., Alwan, A.A., Salleh, N., Shaikhli, I.F.A.: Processing skyline queries in incomplete database: issues, challenges and future trends. JCS **13**(11), 647–658 (2017)
13. Haddache, M., Belkasmi, D., Hadjali, A., Azzoune, H.: An outranking-based approach for skyline refinement. In: 8th IEEE International Conference on Intelligent Systems, IS 2016, 4–6 September 2016, Sofia, Bulgaria, pp. 333–344 (2016)
14. Hadjali, A., Pivert, O., Prade, H.: Possibilistic contextual skylines with incomplete preferences. In: Second International Conference of Soft Computing and Pattern Recognition, (SoCPaR), 7–10 December, Cergy Pontoise/Paris, France, pp. 57–62 (2010)
15. 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 (LNAI), vol. 6804, pp. 581–591. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21916-0_62
16. Hamiche, M., Hadjali, A., Drias, H.: A strong-dominance-based approach for refining the skyline. In: Proceedings of the 12th International Symposium on Programming and Systems (ISPS), 28–30 April, Algiers, Algeria, pp. 1–8 (2015)

17. Khalefa, M.E., Mokbel, M.F., Levandoski, J.J.: Skyline query processing for incomplete data. In: Proceedings of the 24th International Conference on Data Engineering, ICDE 2008, 7–12 April 2008, Cancún, México, pp. 555–565 (2008)
18. Koltun, V., Papadimitriou, C.H.: Approximately dominating representatives. In: Eiter, T., Libkin, L. (eds.) ICDT 2005. LNCS, vol. 3363, pp. 204–214. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30570-5_14
19. Kossmann, D., Ramsak, F., Rost, S.: Shooting stars in the sky: an online algorithm for skyline queries. In: Proceedings of the 28th International Conference on Very Large Data Bases (VLDB), 20–23 August, Hong Kong, China, pp. 275–286 (2002)
20. Lee, J., Hwang, S.: Scalable skyline computation using a balanced pivot selection technique. Inf. Syst. **39**, 1–21 (2014)
21. Lee, J., You, G., Hwang, S.: Telescope: zooming to interesting skylines. In: Kotagiri, R., Krishna, P.R., Mohania, M., Nantajeewarawat, E. (eds.) DASFAA 2007. LNCS, vol. 4443, pp. 539–550. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-71703-4_46
22. Loyer, Y., Sadoun, I., Zeitouni, K.: Personalized progressive filtering of skyline queries in high dimensional spaces. In: Proceedings of the 17th International Conference on Database Engineering Applications Symposium (IDEAS), 09–11 October, Barcelona, Spain, pp. 186–191 (2013)
23. Papadias, D., Tao, Y., Fu, G., Seeger, B.: An optimal and progressive algorithm for skyline queries. In: Proceedings of the International Conference on Management of Data (ACM SIGMOD), 9–12 June, San Diego, California, USA, pp. 467–478 (2003)
24. Pei, J., Jiang, B., Lin, X., Yuan, Y.: Probabilistic skylines on uncertain data. In: Proceedings of the 33rd International Conference on Very Large Data Bases, University of Vienna, 23–27 September, Austria, pp. 15–6 (2007)
25. Santiago, A., et al.: A survey of decomposition methods for multi-objective optimization. In: Castillo, O., Melin, P., Pedrycz, W., Kacprzyk, J. (eds.) Recent Advances on Hybrid Approaches for Designing Intelligent Systems. SCI, vol. 547, pp. 453–465. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-05170-3_31
26. Sarma, A.D., Lall, A., Nanongkai, D., Lipton, R.J., Xu, J.J.: Representative skylines using threshold-based preference distributions. In: Proceedings of the 27th International Conference on Data Engineering, (ICDE), 11–16 April, Hannover, Germany, pp. 387–398 (2011)
27. Tan, K., Eng, P., Ooi, B.C.: Efficient progressive skyline computation. In: Proceedings of 27th International Conference on Very Large Data Bases (VLDB), 11–14 September, Roma, Italy (2001)
28. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival, I. (ed.) Ordered Sets, pp. 445–470. Springer, Heidelberg (1982). https://doi.org/10.1007/978-94-009-7798-3_15
29. Yiu, M.L., Mamoulis, N.: Efficient processing of top-k dominating queries on multi-dimensional data. In: Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB), 23–27 September, University of Vienna, Austria, pp. 483–494 (2007)
30. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**(3), 338–353 (1965)