Comput Syst Sci & Eng (2018) 5: 369–378

International Journal of
Computer Systems
Science & Engineering

Research on K maximum dominant skyline and ε-GA algorithm based on data stream environment

Wang Qi*

School of Electrical Engineering, Chengdu Technological University, Chengdu, Sichuan, China

With the continuous development of database technology, the data volume that can be stored and processed by the database is increasing. How to dig out information that people are interested in from the massive data is one of the important issues in the field of database research. This article starts from the user demand analysis, and makes an in-depth study of various query expansion problems of skylines. Then, according to different application scenarios, this paper proposes efficient and targeted solutions to effectively meet the actual needs of people. Based on k- representative skyline query problem in the data stream environment, a k-representative skyline selection standard k-LDS is presented which is applicable for data stream environment. k-LDS hopes to select the skyline subset with the largest dominant area (containing k skyline tuples only) as k- representative skyline set in data stream. And for the 3-dimensionalal and multidimensional k-LDS problems, this paper also proposes the approximation algorithm, namely GA algorithm. Finally, through the experiment, it is proved that k-LDS is more suitable for the data stream environment, and the algorithm proposed can effectively solve k-LD problems under the data stream environment.

Keywords: Skyline query, k-maximum dominant skylines, greedy algorithm, ε-GA algorithm.

1. INTRODUCTION

With the rapid development of information technology and market, especially after the 1990s, the database management technology has been developed greatly. The database not only can be used to organize, store and manage data, but also can carry out specific management according to user needs. With the development of database technology, the amount of data which can be stored and processed by it is increasing, so how to extract the information people are interested in from the mass data is one of the hot issues in the field of database research [1−3].

As a summary set of the whole data set, skyline query has been widely accepted by people. It can extract information people are concerned about from the whole massive information, and it plays an important role in data mining, multi-objective decision-making, and market decision and so on, which can help people to make effective decisions[4−8]. Back in the 1970, skyline query appeared in the form of the largest vector problem. Since 2001, Borzsonyi et al. first proposed the concept of skyline query, and then there were many research achievements in the application of skyline query. For example, Tao et al. proposed the skyline algorithm on data stream; Lazy, Eager, and Morse proposed a skyline query algorithm LookOut on two-level data stream; Tian et al. put forward a continuous skyline calculation method GICSC based on grid index; Wang et al. improved Lazy's algorithm and proposed the NNSC algorithm; Xin et al. used R*tree to manage uncertain data, so as to calculate the probabilistic skyline; and Cui et al. proposed the skyline calculation under distributed environments.

K-representative skyline is a research branch of skyline query problems, which is of practical significance. It plays a significant role in the market analysis, environmental monitoring, and product selection, etc. K -representative skyline refers to selecting K skyline points from the whole skyline set to replace the whole skyline. At present, there are some studies aiming to solve the k-representative skyline problem in static data. At the same time, they have put forward some criteria for the selection of K-representative skyline points. However, these selection criteria are not applicable to the data stream environment. This thesis focuses on k-representative skyline query problem on data stream. Firstly, it proposes a criterion suitable for the

*E-mail: chzhangqi_2017@163.com

selection of k-representative skyline on data stream, namely k-maximum dominate skyline (k-LDS). The k-LDS problem in 3-dimensional space and above has been proved to be very complicated, so it adopts the greedy algorithm to solve the problem. In order to further accelerate the speed of calculation, this paper puts forward the improved algorithm, ε-GA algorithm, to approximately calculate multidimensional k-LDS problems. Finally, through a series of experiments, it is proved that compared with other k-representative skyline selection standards, k-LDS is more suitable for the environment of data stream, and the proposed ε-GA algorithm can effectively solve the k-LDS problems.

2. K-REPRESENTATIVE SKYLINE RESEARCH UNDER DATA STREAM ENVIRONMENT

2.1 Symbols and definitions

This paper is a study of k-LDS on the data stream. To make it convenient for description, the following gives out the specific symbol definition[9−12].

D refers to the data set, DS for data stream, p_i, p_j for the data points on data streams, DomSpace(p_i) for dominating space of p_i, DomSize(p_i) for dominated area size of p_i, K for any set containing K skyline points, N for the size of the sliding window, DS_N for the data set of the current sliding window, and M for the number of data points of the overall skyline.

For a data set D in the d-dimensional space, for any data point in D, through a standardized method, each dimension value of pi can be conversed to the range of [0,1]. Next, we give the concept of dominant space and area.

Definition 1: the dominant space and area of points: given the data point $p_i = (p_i[1], p_i[2], \cdots, p_i[d])$, all data points falling into its dominant space can be dominated by p_i, and then the dominant space of p_i can be represented as $\text{DomSpace}(p_i) = ([p_i[1], 1], [p_i[2], 1], \cdots [p_i[d], 1])$. The dominant space of p_i is the size of the dominant area (or volume) it can dominate, namely $\text{DomSize}(p_i) = (1 - p_i[1]) \times (1 - p_i[2]) \times \cdots \times (1 - p_i[d])$.

Given a set $S = \{p_1, p_2, \cdots, p_i\}$, its dominant space is a union of all data points in its dominant space, and its dominant area is its dominant space area / volume, denoted as DomSize(S). Intersection space of S refers to the intersection of the dominant space of all data points in S, denoted as $\text{IntSpace}(S) = \Big(\left[\max_{\forall p_j \in S}\left\{p_j[1]\right\}, 1\right], \left[\max_{\forall p_j \in S}\left\{p_j[2]\right\}, 1\right], \cdots, \left[\max_{\forall p_j \in S}\left\{p_j[d]\right\}, 1\right]\Big)$. The intersection area of S refers to the area/ volume size of the intersection space of S, denoted as IntSize(S).

Definition 2: the dominant area of sets

Given a set $S = \{p_1, p_2, \cdots p_i\}$, its dominant area can be calculated through the following formula.

$$\text{DomSize}(S) = \sum_{p_j \in S} \text{DomSize}\left(p_j\right) + (-1)^1 \sum_{S_j \in 2-Setv} \text{IntSize}\left(S_j\right) + \cdots + (-1)^{i-1} \sum_{S_j \in i-Setv(S)} IntSize\left(S_j\right) \tag{1}$$

$$\tag{2}$$

The formula (5.1) can be obtained by the inclusion-exclusion principle. The following gives out the concept of k-LDS.

Definition 3: k-LDS:

Given the overall skyline set $SKY = \{s_1, s_2, \cdots, s_M\}$ including M skyline points, if the parameter $k < M$, k-LDS is the collection of the sets containing k skyline points with the largest dominant area. Otherwise, k-LDS is the overall skyline, denoted as

$$k - LDS = \left\{K_i \,|\forall K_j \subseteq SKY, DomSize\left(K_j\right) \leqslant DomSize\left(K_i\right)\right\}.$$

2.2 Appearance of k-LDS under data stream environment

This paper uses a sliding window to achieve the modeling of data on data stream. The main consideration is the sliding window based on counting. The selection criteria of K-representative skyline selection on data stream should follow the following two conditions: first, as a representative skyline, being highly representative is unquestionable; second, in the data stream environment, frequent data change may lead to unacceptable computational overhead, so the computational efficiency is another standard of k-representative skyline on data stream[13−14].

We propose a new k-representative skyline selection standard, namely k-LDS, which is used for k-representative skyline query on data stream. The goal of K-LDS is to select a group of sets consisting k skyline points, and these sets have the largest dominant area. The dominant area of a set refers to the size of the union of the area that can be dominated by each point in the set. As shown in Figure 1, the dominant area of the set $\left\{p_3, p_5, p_8\right\}$ can be seen in blue regions, so its dominant area is 0.51.

3. K-LDS QUERY PROCESSING UNDER DATA STREAM ENVIRONMENT

Calculation of k-LDS in the sliding window contains the following two steps: (1) continuously maintain the overall skyline set SKY in the sliding window; (2) calculate k-LDS in the overall skyline set SKY. Since the skyline query technology in the sliding window has been very mature, the focus of this paper is how to calculate k-LDS in the whole skyline[15−17].

3.1 Overview of k-LDS problems

According to the distribution characteristics of the skyline points, k-LDS calculation can be divided into 2 parts.

In the 2-dimensional space, the distribution of the overall skyline SKY has the following features: when the points of $SKY = \{s_1, s_2, \cdots, s_M\}$ are in ascending order in one dimension, then they must be in descending order in other dimensions. Therefore, for all of the SKY and any of its subsets in

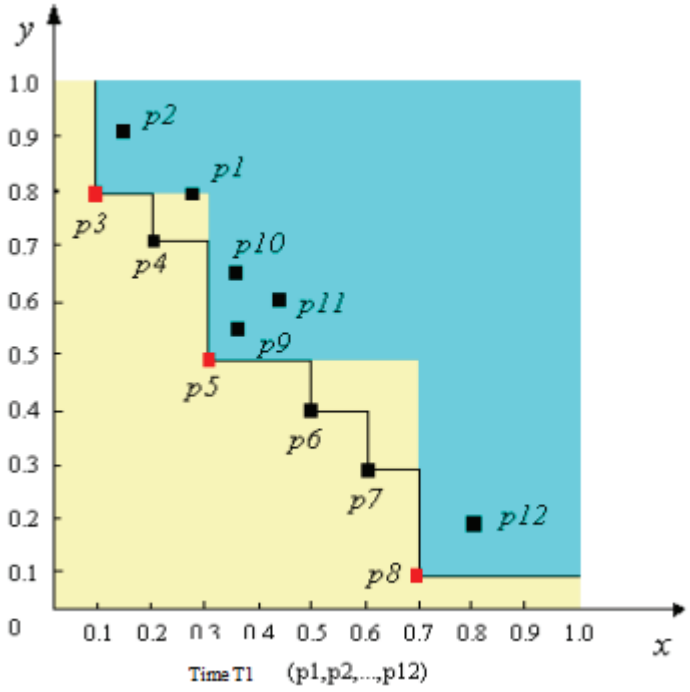

Figure 1 k-representative skyline (k=3) on a sliding window.

2-dimensional space, this paper will put them in ascending order according to their value in dimension 1. Therefore, for any set containing k skyline points $K = \{s'_1, s'_2, \cdots, s'_k\}$, its dominant area can be calculated by the formula (2):

$$\begin{aligned}\text{DomSize}(K) = &\sum_{i\in(1,k-1]} \left(1 - s'_i[2]\right)\left(s'_{i+1}[1] - s'_i[1]\right) \\ &+ \text{DomSize}\left(s'_k\right) \end{aligned} \tag{3}$$

In d-dimensional space, the distribution of the skyline points has no features, and dynamic programming thought will not guarantee the transitivity between local optimal solutions. Chen and others have pointed out that the time complexity of k-LDS in the d dimension is $O\left(\left(\frac{2\times e\times M}{k}\right)^k\right)$. Therefore, the price of accurately calculating k-LDS in the d dimension is unacceptable, so that we propose a heuristic algorithm to obtain the approximate k-LDS[18].

3.2 k-LDS calculation on the d-dimension

Chen and others have pointed out that the time complexity of k-LDS is $O\left(\left(\frac{2\times e\times M}{k}\right)^k\right)$. Therefore, to obtain accurate k-LDS, the brute force method should be adopted. Thus, to calculate the precise k-LDS in a sliding window is not possible, so we propose that greedy strategy should be used to calculate approximate k-LDS.

(1) k-LDS greedy calculation method

First, we introduce a basic concept.

Original $SKY = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7\}$, $K = \{s_2, s_5, s_6\}$/updated value, when s_8 inserts, $K_{new} = \{s_2, s_8, s_6\}$

Definition 4: incremental dominant area

Given a set with n skyline points $NS = \{s'_1, s'_2, \cdots, s'_N\}$, skyline points $s_j \in SKY$, and $s_j \notin NS$, the incremental dominant area of s_j in regard to NS is the difference value between DomSize $\left(NS \cup \{s_j\}\right)$ and DomSize (NS), denoted as IntSize $\left(NS, s_j\right)$, as shown in formula (3).

$$\begin{aligned}&\text{IncreSize}\left(NS, s_j\right) \\ &= \text{DomSize}\left(s_j\right) - \Bigg[\sum_{s_i\in NS} \text{IntSize}\left(\{s_j, s_i\}\right) + (-1)^{-1} \\ &\sum_{s_i\in 2-Setv(NS)} \text{IntSize}\left(\{s_j\}\cup S_i\right) + (-1)^{|N|-1} \\ &\sum_{s_i\in N-Setv(NS)} \text{IntSize}\left(\{s_j\}\cup S_i\right)\Bigg] \end{aligned} \tag{4}$$

Given the overall skyline $SKY = \{s_1, s_2, \cdots, s_M\}$ in the d dimensional space, for each $s_i \in SKY$, we calculate the dominant area DomSize (s_i) of s_i. first, we add the points with the largest dominant area into the set K. second, for each $s_i \in SKY - K$, we calculate the incremental dominant area of K at present, and add the points with the largest dominant area into K. Repeat this process until the set K has k skyline points[19–20].

The k-LDS problem can be easily converted into the maximum coverage problem of sets. Therefore, the approximation ratio of the greedy algorithm is $\left(1 - 1/\sqrt{e}\right)$, where e is the Euler constant. Specifically, OPT refers to the accurate k-LDS results, and the following corollary can be obtained:

$$\begin{aligned}&\text{DomSize}(K_i) - \text{DomSize}(K_{i-1}) \geqslant (1/k - i) \\ &\quad (\text{DomSize}(OPT) - \text{DomSize}(K_{i-1})) \end{aligned} \tag{5}$$

$$\text{DomSize}(K_i) \geqslant \left(1 - (1 - 1/k)^i\right)\text{DomSize}(OPT) \tag{6}$$

Thus, the approximate ratio of the greedy algorithm is $\left(1 - 1/\sqrt{e}\right)$.

Table 1 Calculation of k-LDS on the 3-dimensional.

s_i	Dom Size (s_i)	IntSize ($\{s_i, s_2\}$)	IncreSize ($\{s_2\}, s_i$)	IntSize ($\{s_i, s_2\}$) / ($\{s_i, s_8\}$)	IntSize ($\{s_i, s_2, s_5\}$) / ($\{s_i, s_2, s_8\}$)	IntSize ($\{s_2, s_5\}, s_i$) / ($\{s_2, s_8\}, s_i$)
s_1(0.3,0.4,0.9)	0.03	0.041	0.008	0.02/0.025	0.012/0.02	0.004/0.03
s_2(0.2,0.5,0.4)	0.232	-	-	-	-	-
s_3(0.3,0.4,0.5)	0.14	0.09	0.04	0.10/0.135	0.08/0.1	0.01/0.005
s_4(0.5,0.4,0.7)	0.078	0.052	0.044	0.052/0.052	0.048/0.048	0.012/0.012
s_5(0.6,0.3,0.4)	0.174	0.108	0.063	-/0.148	-/0.096	-/0.008
s_6(0.7,0.2,0.6)	0.112	0.07	0.052	0.08/0.08	0.06/0.06	0.015/0.015
s_7(0.7,0.4,0.4)	0.07	0.060	0.033	0.07/0.06	0.058/0.044	0.01/0.012
s_8(0.4,0.3,0.4)	0.15	0.12	0.05	-	-	-

Algorithm 1 greedy algorithm

Input: $SKY = \{s_1, s_2, \cdots, s_M\}$, parameter k.
Output: Approximate k-LDS results set.

01. Initialize $K = \emptyset$;
02. While ($|K| < k$)
03. For ($s_i \in SKY - K$)
04. Calculate the incremental dominant area of s_i in regard to K: IntSize (K, s_i).
05. EndFor
06. add the points with the largest incremental areas into K
07. EndWhile

By the formula (3), it can be obtained that the greedy algorithm's time complexity is $O\left((M-k) \times 2^k \times d \times k\right)$. When k is relatively large, the processing time of greedy algorithm is still too long, so we further propose the ε-GA algorithm which has made great improvements in speed only at the expense of small precision.

(2) ε-GA algorithm

At each iteration, we add a point with the largest incremental dominant area control area into the result set. Obviously, with the increase of the points added into the result set, the incremental dominant area of remaining points becomes smaller. Therefore, we accelerate the calculation of k-LDS through ε-GA algorithm.

In the ε-GA algorithm, we set a minimum valueε. When there are L data points added to the result set K, the remaining data will be put in order according to their current incremental dominant size. If the sum of the incremental dominant area of the first k-L data points is less than the value $\varepsilon \times$ DomSize (K), where DomSize (K) is the dominant area currently added into the result set, we can stop the calculation and directly add the first k-L data points into the result set. This is because the remaining data points contribute a little to the dominant area of the result set. The specific process of ε-GA algorithm is shown as algorithm 2.

Algorithm 2: ε-GA algorithm

Input: $SKY = \{s_1, s_2, \cdots, s_M\}$, parameter k.
Output:Approximate k-LDS result set.

01. Initialize K=the point with the largest dominant area;
02. While ($|K| < k$)
03. For ($s_i \in SKY - K$)
04. Calculate the incremental dominant area of s_i in regard to K: IntSize (K, s_i);
05. EndFor s
06. Put the remaining data points in descending order based on their incremental dominant area;
07. Transfer the point which ranks first into K;
08. Sum = the sum of the incremental area of the first $k - |K|$ tuples of the remaining tuples;
09. If ($Sum < \varepsilon \times$ IntSize(K, s_i))
10. Transfer the first $k - |K|$ tuples of the remaining tuples into K;
11. Break: // jump out of while circulation
12. EndIf
13. EndWhile
14. Back to K.

As shown in Figure 1, when $\varepsilon = 0.09$ and $k = 5$, first of all, s_2 is added to the result set. Then, s_5 and s_6 are added to the result set in succession. When $K = \{s_2, s_5, s_6\}$, its dominant area is 0.295. After the remaining data points are put in order according to the incremental dominant area, the two tuples s_4 and s_3 (or s_7) with the largest incremental dominant area in regard to $\{s_2, s_5\}$ are added to the result set, and the final set is get: $\{s_2, s_3, s_4, s_5, s_6\}$ ($\{s_2, s_4, s_5, s_6, s_7\}$).

Let L denote the number of data points included in the result set when $Sum < \varepsilon \times IntSize(K, s_i)$ in the ε-GA algorithm. As shown in Figure 1, when the condition $Sum <$

$\varepsilon \times \text{IntSize}(K, s_i)$ is triggered, the result set contains only 3 data points, so $L = 3$. Therefore, ε-GA algorithm's time complexity is $O\left((M - L) \times 2^L \times d\right)$, where L's value is smaller than K. the approximation ratio of ε-GA algorithm is $\frac{1}{1+\varepsilon}\left(1 - \frac{1}{\sqrt{e}}\right)$. With the increase of the value of ε, both the time consumption and approximation ratio of the algorithm are reduced. Therefore, the value of ε is a trade-off value. In the experiment, we analyze the suitable value of ε in different distributions.

3.3 Maintenance of ε-GA in sliding windows

At each iteration of ε-GA epsilon, some intermediate values are needed, including increment area and intersection area. Therefore, in the maintenance process of ε-GA, we store all intermediate results, and form the maintenance table. In the i-th iteration, $(M - (i - 1)) \times 2^{i-1}$ intermediate values are generated and stored. After the end of L iterations, the ε-GA algorithm is end. At the moment, the intermediate values of the first L iterations are accumulated, with a total of $(M - L) \times 2^L$ intermediate values being stored.

When a new skyline point s_{new} is added to SKY, we need to recalculate the maintenance table. If s_{new} does not belong to the final result set, only 2^{L-1} intermediate values need to be calculated. If s_{new} is the i-th one being inserted into the result set ($i < L$) and points in other result sets remain the same, there will be $(M - L) \times \left(2^L - 2^i\right)$ intermediate value needing to be calculated.

As shown in Figure 1, given $\varepsilon = 0.09$, $k = 5$, when a new data point $s_8 = (0.5, 0.4, 0.4)$ is inserted, we first calculate the dominant area of s_8: $\text{DomSize}(s_8) = 0.18 < \text{DomSize}(s_2)$, so $K = \{s_2\}$ has no change. Second, we calculate the incremental area of s_8 in regard to K: $\text{IncreSize}(K, s_8) = 0.06$. Since 0.06 is the largest incremental area currently, s_8 is inserted into K. Then, according to the same method, it is needed to re-calculate the incremental dominant area of the remaining data points. Finally, we can get a new result set: $\{s_2, s_8, s_6, s_4, s_7\}$.

When $Sold \in k - LDS$, all intermediate results related to $Sold$ need to be deleted. If $Sold$ is the i-th point inserted into the result set, then at most $(M - L) \times 2^L - (M - i) \times 2^i$ intermediate values need to be re-calculated. By setting the appropriate ε values, in all distributions, it can be ensured that the L value is smaller than 10. Therefore, the maintenance of ε-GA in the sliding window is very fast and convenient.

4. EXPERIMENTAL VERIFICATION

4.1 Experimental setup analysis

In this section, we use C++ software to analyze and validate the ε-GA algorithm of k-LDS problem on data stream.

The algorithm GA and ε-GA are validated and compared. GA and ε-GA algorithm are used to solve the calculation of approximate k-LDS in multidimensional space. GA is the basic greedy algorithm, as shown in algorithm 1. ε-GA is the improved greedy algorithm, as shown in algorithm 2, and it is also the default algorithm of multidimensional data processing.

This paper respectively uses real data and synthetic data to validate the algorithm's performance. The real data is stock data and forest fire monitoring data, and all data re-normalized to the range of [0, 1]. The synthetic data uses the skyline query standard test data set, including: independent and anti-correlated data sets.

To stabilize the test results, we randomly generate 100 data streams recording their average processing time.

4.2 Representativeness measurement of k-representative skyline under data stream environment

This section compares the performance of k-LDS and other K-representative skylines in a sliding window, including RSP, DRS, and k-regret.

First of all, through the use of stock data, we test the representativeness of four k-representative skylines. Using the dominance number to measure the representativeness of the set selected has been accepted by people. Therefore, this chapter uses the same test standard. Figure 3 contains 800 stock records and 11 skyline points. When $k = 4$, it can be found that k-LDS and RSP can almost dominate all non-skyline points, while the dominance number of DRS and k-regret is much smaller than that of k-LDS and RSP. Therefore, k-LDS is highly representative.

Figure 2 shows the efficiency of 4 concepts in the sliding window. The figure records the continuous 10^5window sliding and the average processing time of each sliding. It is shown that the treatment time of k-LDS is far lower than that of other 3 kinds of definition. At the same time, with the increase of k,the growth rate of other defined time is higher than that of k-LDS. Therefore, the processing speed of k-LDS in the sliding window is very efficient.

Here, for more experimental environments, we give the test results for 4 concepts, as shown in Figure 4

From Figure 4, it can be found that compared with DRS and k-regret, k-LDS performs better in the processing efficiency and the dominance number, so that k-LDS is more applicable in data stream environment. Compared with RSP, although the dominance number of k-LDS is slightly lower, its processing efficiency is much higher than that of RSP. Moreover, in multi-dimensional anti-correlated data, the treatment efficiency of RSP is not acceptable. So, overall, k-LDS is the most suitable for k-representative skyline in the data stream environment.

4.3 Precision comparison of greedy algorithm

The MaxDis algorithm is proposed in the literature. Although it is not the concept proposed for the k-representative skyline, its goal is in accordance with the algorithm in this chapter. For this, we compare the MaxDis algorithm with the greedy algorithm GA and ε-GA in this chapter. Since our PBA algorithm in 2-dimension is accurate algorithm, we only give the precise k-LDS in 2-dimension. With the same setting, there are 10^5 sliding altogether in the sliding window.

Figure 3 tests the effect of ε on the accuracy and time of the algorithm. In the independent data, when $\varepsilon < 0.1$, the accuracy of the output results of GA and ε-GA is basically the same.

Table 2 Experimental parameters

Parameters	Default value	Range of variation
Size of the sliding window(M)	1	0.5,1,1.5,2,2.5
Data dimension	4	3,4,5,6
k	10	5,10,15,20,25

Table 3 Representativeness of k-LDS in the stock data.

	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	Dominance number
LDS					*	*			*	*		784
RSP			*		*	*				*		778
DRS			*					*		*	*	675
k-regret		*			*	*		*				654

Table 4 The performance of the algorithm in different parameters.

Parameters				Concepts	Test values	
Dimension	Size of the sliding window	k	Distribution		Dominance number	Time consumption(ms)
2	10^6	10	Independent	k-LDS	999892	0.0016
				RSP	999986	0.2846
				DRS	878484	0.0156
				k-regret	818366	0.0578
2	2×10^6	20	Anti-correlated	k-LDS	1999565	0.0038
				RSP	1999730	1.567
				DRS	1734857	874657
				k-regret	1678977	816757
4	10^6	10	Independent	k-LDS	991643	0.0210
				RSP	996887	198.65
				DRS	786865	0.5644
				k-regret	712747	2.856
4	2×10^6	20	Anti-correlated	k-LDS	1443432	0.2127
				RSP	1578698	19445.84
				DRS	1358555	10.7454
				k-regret	1285356	98.856

When $\varepsilon > 0.05$, the processing time ε-GA is reduced greatly, so in the independent data set, we set $\varepsilon = 0.05$. In anti-correlated data, when $\varepsilon < 0.15$, the accuracy of the output results of GA and ε-GA is basically the same. When $\varepsilon > 0.15$, the processing time ε-GA is reduced greatly, so in the independent data set, we set $\varepsilon = 0.15$. For all data sets with the unknown distribution, they are processed in accordance with the anti-correlated εvalue because the anti-correlated distribution is the worst distribution of skyline query.

4.4 Evaluation of algorithm efficiency

In this section, we compare the ε-GA algorithm with the MaxDis algorithm, and evaluate the performance of the proposed algorithm in general.

First, we test the performance of ε-GA algorithm with the MaxDis algorithm under the multidimensional environment. As shown in Figure 4, with the increase of the size of the sliding window, the skyline number increases slightly, so does the algorithm. In the independent data set, the time consumption of the MaxDis algorithm is slightly smaller than that of the ε-GA algorithm in this paper. In the anti-correlated data set, time consumption of ε-GA algorithm is slightly less than MaxDis, which is because in the anti-correlated data, there are more skyline points, and the ε-GA algorithm in this paper is more insensitive to the skyline number.

As shown in Figure 5, with the increase of k value, there is a sharp increase in processing time of GA algorithm, while that of ε-GA algorithm has no change, along with the linear increase of MaxDis. In the independent data set, time efficiency of MaxDis algorithm and ε-GA algorithm is almost the same. In the anti-correlated data set, time consumption of ε-GA algorithm is slightly less than MaxDis.

As shown in Figure 6, with the increase of dimension, skyline points increase exponentially, so does the processing time

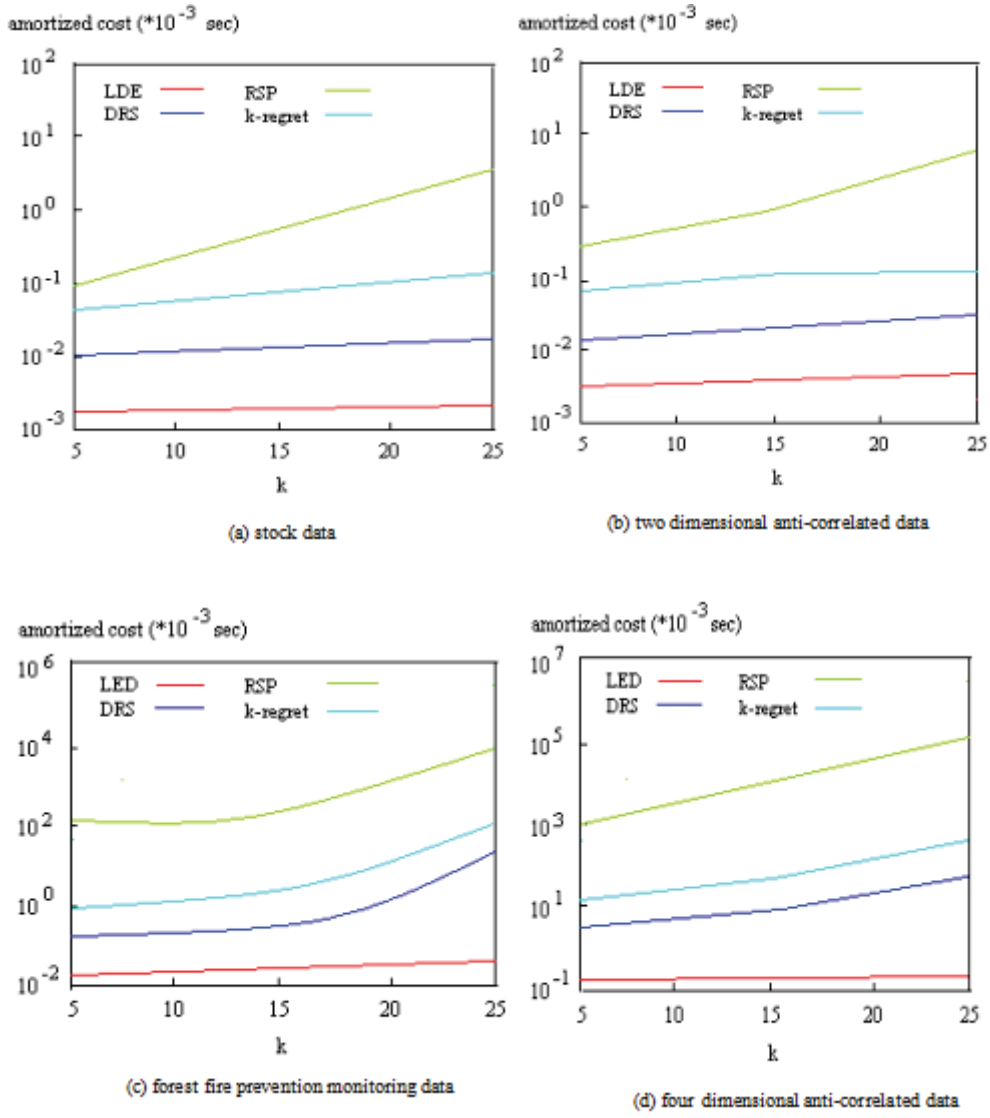

Figure 2 Time consumption of k-LDS.

of algorithms. Similarly, in the independent data set, time efficiency of MaxDis algorithm and ε-GA is almost the same. In the anti-correlated data set, time consumption of ε-GA algorithm is slightly less than MaxDis.

Based on all of the above experimental results, it can be found that compared with other k-representative skyline concepts, k-LDS performs well in processing efficiency and representativeness. According to comprehensive consideration, k-LDS is more suitable for the data stream environment. In dealing with the issue of k-LDS, ε-GA algorithm has almost the same performance as MaxDis in processing efficiency, but in accuracy it is far better than MaxDis. Therefore, ε-GA algorithm proposed in this paper performs better in resolving k-LDS problem.

5. CONCLUSION

This paper studies k-maximum dominant skyline and ε-GA algorithm under the data flow environment. Firstly, we propose the selection standard k-LDS suitable for k-representative skyline on data stream, and prove that k-LDS has two characteristics: high representativeness and efficiency. These two features make it very suitable for data flow environment. Then, in order to quickly calculate k-LDS in the sliding window, based on the 2-dimensional data environment, we propose an accurate algorithm PBA, which improves the time complexity of the algorithm to $O((M-k)\times k)$ based on dynamic programming method, thus enhancing the processing efficiency.

Then, we put forward the maintenance strategy of algorithm PBA in the sliding window. After this, in the 3-dimensional space and above, the k-LDS problem has been proved to be very complicated, so we use greedy algorithm to solve the problem. In order to further accelerate the computation speed, we propose ε-GA algorithm to approximately compute the k-LDS problem in multidimensional space. Though ε-GA algorithm sacrifices a low degree of accuracy, its calculation speed is greatly improved. Finally, through a series of experiments, it is proved that k-LDS is more suitable for the data stream environment than other K-representative skyline concepts, and the proposed ε-GA algorithm can effectively solve the k-LDS problem.

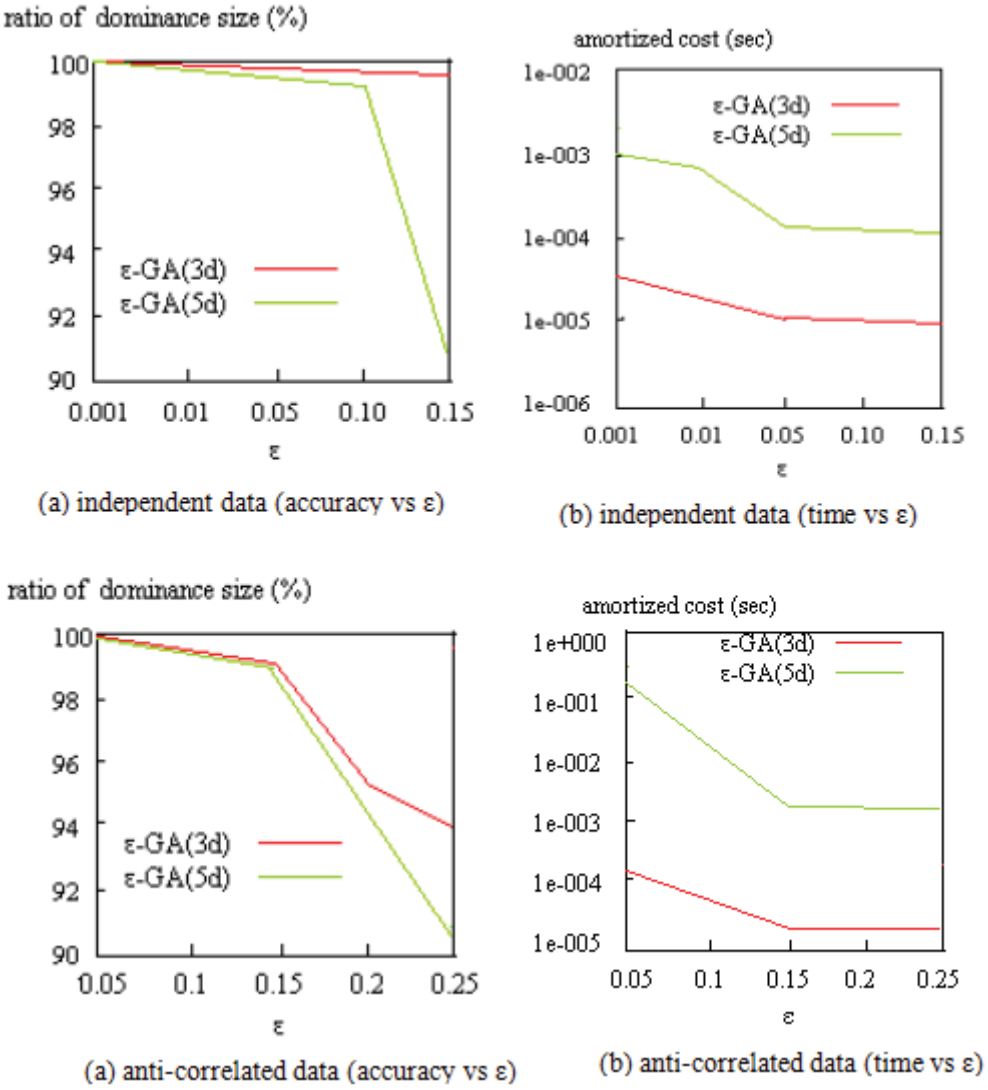

Figure 3 Influence of the value of ε on ε-GA algorithm.

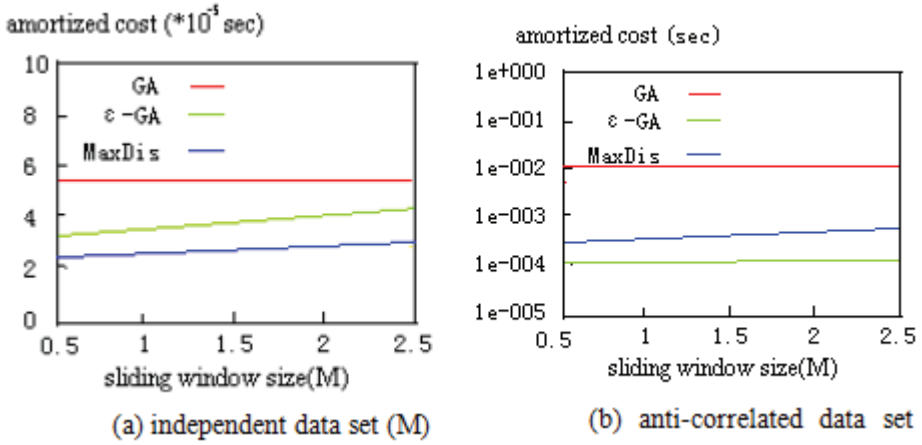

Figure 4 Influence of the size of the sliding window on algorithms.

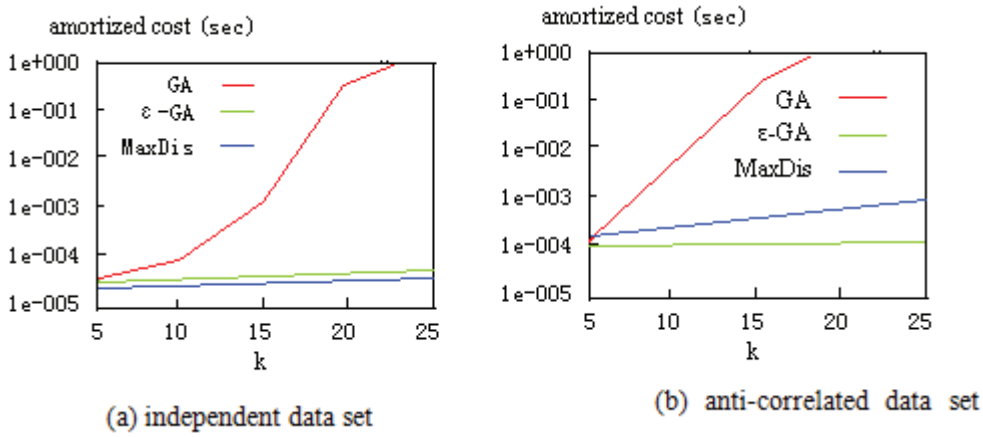

Figure 5 Influence of processing time on algorithms.

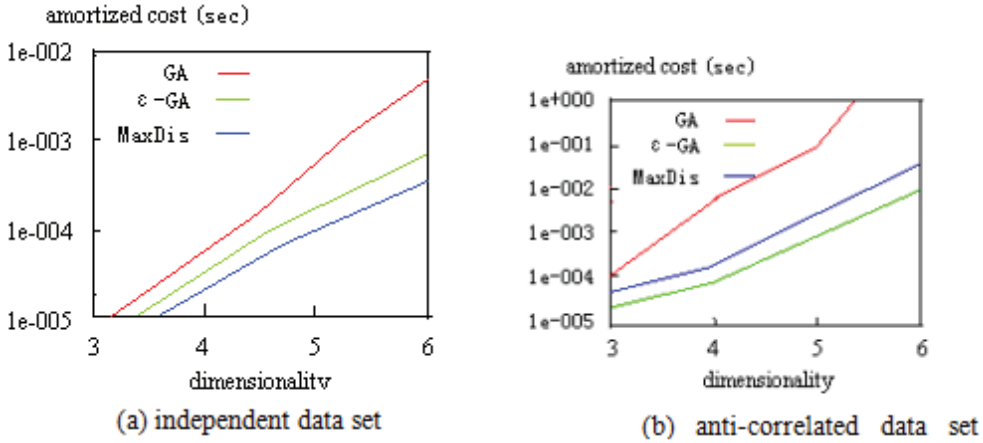

Figure 6 Influence of dimension on algorithms.

REFERENCES

1. Ma Z, Zhang K, Wang S, et al. A double-index-based k-dominant skyline algorithm for incomplete data stream[C]// IEEE International Conference on Software Engineering and Service Science. IEEE, 2013:750-753.
2. Dong L G, Cui X W, Wang Z F, et al. Finding k-dominant Skyline cube based on sharing-strategy[C]// International Conference on Fuzzy Systems & Knowledge Discovery. IEEE, 2010:1694-1698.
3. Cui X W, Dong L G, Zou H, et al. Notice of RetractionFinding k-dominant skyline in dynamic data set[C]// International Conference on Natural Computation. IEEE, 2011:1247-1250.
4. Siddique M A, Morimoto Y. Efficient Selection of Various k-Objects for a Keyword Query Based on MapReduce Skyline Algorithm[J]. Lecture Notes in Computer Science, 2014, 8381:40-52.
5. Jian Y. An Index Based Efficient k-Dominant Skyline Algorithm[J]. Chinese Journal of Computers, 2010, 33(7):1236-1245.
6. Ruan P Q, Xu C W, Huang J T, et al. A Distributed Algorithm for Skyline Query Based on Pre-Clustering[J]. Advanced Materials Research, 2013, 756-759:3982-3986.
7. Gemsa A, Haunert J H. Multirow Boundary-Labeling Algorithms for Panorama Images[J]. Acm Transactions on Spatial Algorithms & Systems, 2015, 1(1):1-30.
8. Bing T U, Pan J, Zhang G, et al. Research on skyline detection algorithm based on LBP and sparse representation[J]. Computer Engineering & Applications, 2016.
9. Jing Y U, Liu P P. k-dominant skyline query algorithm based on Map Reduce framework[J]. Journal of Yanshan University, 2014, 46:470-470.
10. Yuan W, Liu S, Yu G, et al. Global estimates of evapotranspiration and gross primary production based on MODIS and global meteorology data[J]. Remote Sensing of Environment, 2010, 114(7):1416-1431.
11. Estrada-Carmona J, Weber B. Petrogenesis of Ordovician magmatic rocks in the southern Chiapas Massif Complex: Relations with the early Palaeozoic magmatic belts of northwestern Gondwana[J]. International Geology Review, 2012, 54(16):1918-1943.
12. Vimercati S D C D, Foresti S, Samarati P. Protecting Information Privacy in the Electronic Society[M]// e-Business and Telecommunications. Springer Berlin Heidelberg, 2011:20-36.
13. Jasna S, J Pillai M. An Algorithm for Retrieving Skyline Points based on User Specified Constraints using the Skyline Ordering[J]. International Journal of Computer Applications, 2014, 104(11):24-29.
14. Miao X, Gao Y, Chen G, et al. k-dominant skyline queries on incomplete data[J]. Information Sciences An International Journal, 2016, 367(C):990-1011.
15. Example F, Quality C, Environment A, et al. Alternative Tuples Based Probabilistic Skyline Query Processing in Wireless Sensor Networks[J]. Mathematical Problems in Engineering,2015,(2015-12-30), 2015, 2015:1-10.
16. Siddique M A, Tian H, Morimoto Y. k-Dominant Skyline Query Computation in MapReduce Environment[J]. Ieice Transactions on Information & Systems, 2015, E98.D(5):1027-1034.

17. Radovitzky R, Seagraves A, Tupek M, et al. A scalable 3D fracture and fragmentation algorithm based on a hybrid, discontinuous Galerkin, cohesive element method[J]. Computer Methods in Applied Mechanics & Engineering, 2011, 200(1–4):326-344.
18. Siddique M A, Zaman A, Islam M M, et al. Multicore Based Spatialk-dominant Skyline Computation[C]// Third International Conference on NETWORKING and Computing. IEEE, 2013:188-194.
19. Lalithadevi B, Leelambika K V, Sageengrana S. Finest price prophecy for finding top-K popular and profitable products based on skyline analysis[J]. International Journal of Applied Engineering Research, 2014, 9(22):13253-13264.
20. Siddique M A, Morimoto Y. Extended k-dominant Skyline in High Dimensional Space[C]// International Conference on Information Science and Applications. IEEE, 2010:1-8.