

Selection of Important Sets by using K -Skyband Query for Sets

Md. Anisuzzaman Siddique, Asif Zaman, Yasuhiko Morimoto
Hiroshima University, Japan

Abstract—In this paper, we consider “sets” selection problem from a database. In conventional selection problem, which is “objects” selection problem, the skyline query has been utilized, since it can retrieve a set of important objects where each object isn’t dominated by another object in a database. However, it is not effective when we have to select important sets, each of which contains more than one objects. Thus, we consider a “set skyline query” that retrieves non-dominated sets of objects from a database, which we call “object sets.” The K -skyband query is a popular variant of the skyline query. It retrieves a set of objects, each of which is not dominated by K other objects. In this paper, we propose “ K -Skyband set query.” It retrieves important sets instead of objects. We investigated the properties of the query, as well as developing pruning strategies to avoid the unnecessary enumeration of objectsets and comparisons among them. Intensive experiments have been performed to examine the implemented algorithm. The results demonstrate the effectiveness and efficiency of the proposed algorithm.

Keywords—Set Selection; Skyline Query; Skyline Set Query; Skyband Query; Skyband-set Query

I. INTRODUCTION

To select important objects from a large-scale database is one of the most important processes to analyze the database. In the database literature, Borzsony et al. have been proposed a query that retrieves a set of objects where each is not dominated by another object in the database [1]. We call it a “skyline query.” It has attracted lots of researchers and practitioners due to broad applicability in decision making and analysis tasks [2].

Let us consider an example of a financial investment problem in the table in Fig. 1, which are seven stocks with their corresponding prices (a_1) and risks (a_2). In general, all investors want to invest in stocks with lower commission costs and lower predicted risks. Fig. 1 shows the skyline query result for the table where the result is $\{O_1, O_2, O_3\}$.

In some cases, we may need to select two or more objects in a selection or in a lookup. For example, a user might want to invest in a combination of stocks with low costs and risk. We often call this combination an investment portfolio. Skyline query cannot be utilized effectively in this case because a user may have to choose dominated objects that are not among the skyline objects during multiple object selection in order to obtain their optimal choices.

The investment portfolio selection problem can be analyzed by using sets of objects, which we call “objectsets.” We denote s as the number of objects in each objectsets. Let us consider an investor who wants to invest in two stocks, i.e., we assume that $s = 2$. Table I shows the different combinations of two

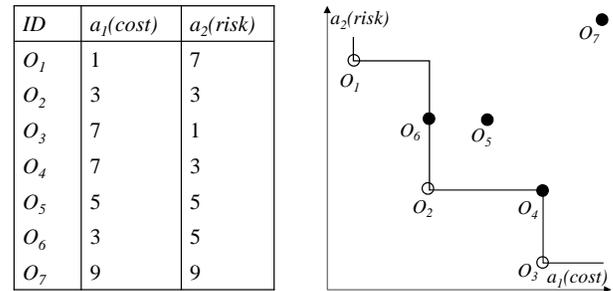


Fig. 1. Non-dominated objects and skyline.

objects from the running example where each record is a set of two stock objects. The attribute values for each objectset are the sum of their corresponding components values. Now, we can formalize the problem where the objectset skyline has lower attribute values a_1 (cost) as well as a_2 (risk). For $s = 2$, an objectset skyline query retrieves the result of $\{OS_{1,2}, OS_{2,3}, OS_{2,6}\}$ because no other objectset can dominate them (see Fig. 2). Notice that the conventional skyline result of $\{O_1, O_2, O_3\}$ does not provide sufficient insights into the selection problem. Investors always want to select non-dominated objectsets to ensure that their portfolio has the minimum cost with minimal risk.

Furthermore, we assume that an investor wants to invest in three stocks. After computing all combinations of three stocks, we need to check the dominance of these combinations to obtain the objectset skyline result. For combinations of three stocks, the objectset skyline query retrieves $\{OS_{1,2,3}, OS_{1,2,6}, OS_{2,3,4}, OS_{2,3,6}\}$ as the output result (see Fig. 3).

To address the issues related to set selection problem, we propose a “ K -skyband-set”. A K -skyband query, which is a popular variant of skyline query, returns objects that are not dominated by K other objects [6]. A K -skyband-set query retrieves objectsets, each individual objectset of which is not dominated by K other objectsets. In other word, an objectset in a K -skyband-set query’s results may be dominated by at most $K - 1$ other objectsets. For example, if we set the objectset size $s = 1$ and the skyband value $K = 1$, then the skyband set query retrieves objectsets $\{O_1, O_2, O_3\}$ for the seven stocks example as in Fig. 1. In addition, for $s = 1$ and $K = 2$, the skyband-set query retrieves $\{O_1, O_2, O_3, O_6\}$. This is because the objectsets comprising O_1, O_2, O_3 are not dominated by any objectset, and objectset O_6 is dominated by only one objectset O_2 . For $s = 1$ and $K = 3$, the skyband-set query retrieves $\{O_1, O_2, O_3, O_4, O_5, O_6\}$. Thus, the K -skyband query provides flexibility to increase and decrease the number of objectsets by varying the size of K .

TABLE I. SETS OF TWO STOCKS

ID	$a_1(cost)$	$a_2(risk)$	ID	$a_1(cost)$	$a_2(risk)$	ID	$a_1(cost)$	$a_2(risk)$
$OS_{1,2}$	4	10	$OS_{2,4}$	10	6	$OS_{3,7}$	16	10
$OS_{1,3}$	8	8	$OS_{2,5}$	8	8	$OS_{4,5}$	12	8
$OS_{1,4}$	8	10	$OS_{2,6}$	6	8	$OS_{4,6}$	10	8
$OS_{1,5}$	6	12	$OS_{2,7}$	12	12	$OS_{4,7}$	16	12
$OS_{1,6}$	4	12	$OS_{3,4}$	14	4	$OS_{5,6}$	8	10
$OS_{1,7}$	10	16	$OS_{3,5}$	12	6	$OS_{5,7}$	14	14
$OS_{2,3}$	10	4	$OS_{3,6}$	10	6	$OS_{6,7}$	12	14

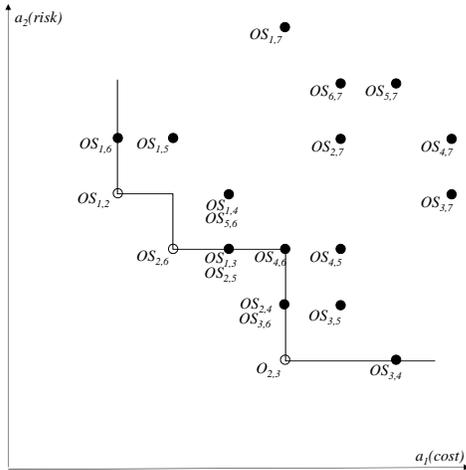


Fig. 2. Skyband-set (s=2, K=1).

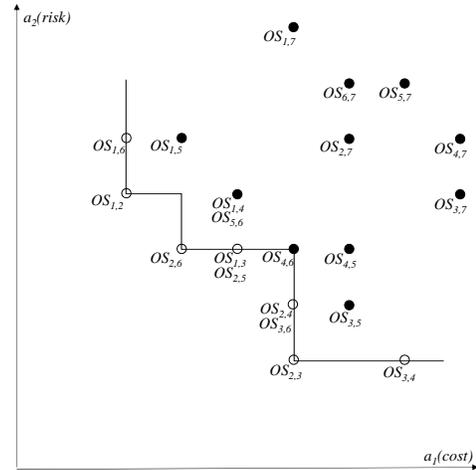


Fig. 4. Skyband-set (s=2, K=2).

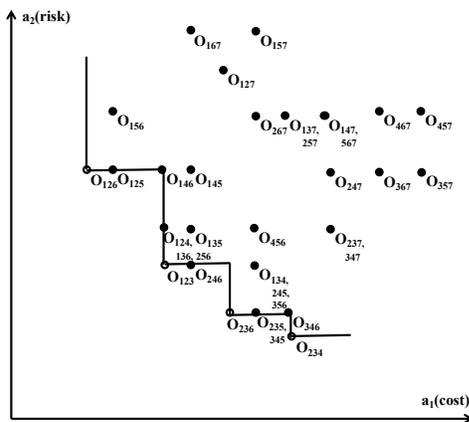


Fig. 3. Three-objects skyline.

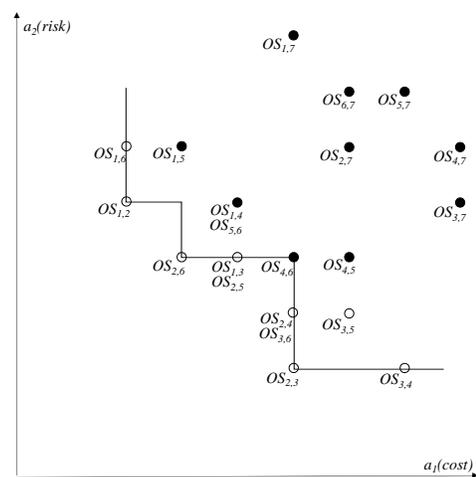


Fig. 5. Skyband-set (s=2, K=3).

Next, let us consider examples for $s = 2$. Table I represents the sets of two objects. According to Fig. 2, it is clear that the objectsets $\{OS_{1,2}, OS_{2,3}, OS_{2,6}\}$ are not dominated by any other objectset. Therefore, they are among the results obtained by the objectset skyband queries for $s = 2$ and $K = 1$. We can increase the number of the results set by increasing K . For $s = 2$ and $K = 2$, the skyband-set query retrieves $\{OS_{1,2}, OS_{1,3}, OS_{1,6}, OS_{2,3}, OS_{2,5}, OS_{2,6}, OS_{3,4}\}$ (see Fig. 4). Similarly, for $s = 2$ and $K = 3$, the skyband-set query retrieves $\{OS_{1,2}, OS_{1,3}, OS_{1,6}, OS_{2,3}, OS_{2,5}, OS_{2,6}, OS_{3,4}, OS_{3,5}\}$ (see Fig. 5).

The main challenge when developing an objectset skyband query is overcoming its large space complexity. For a data set with n records, the number of objectsets of size s is

up to nC_s . Thus, the time complexity is also high because we need to compute all of the objectsets to obtain the final result. The traditional skyline or skyband algorithm calculates all of the candidate objectsets progressively. Next, it updates the resulting objectset dynamically. Therefore, existing index structures, such as ZBtrees [5] and R-trees [6], are not suitable for objectset skyband computation. So far, there is no existing work that can compute K -skyband set efficiently.

We propose an efficient method that can select the K -skyband set in this paper. We utilize two filtering techniques to prevent computing large volumes of unnecessary objectsets. In addition, we examine and confirm that these pruning strategies

are also useful for skyband objectset computation. We empirically verified the efficiency of the proposed algorithm by conducting several experiments with various datasets including synthetic and real datasets.

Organization of the remainder of this paper is as follows. In Section II, we first review the most basic research into skyline queries, before explaining objectset skyline queries. In Section III, we present the problem definition for a skyband-set query and the related definitions. We also discuss the properties of objectsets in this section. We explain the implementation with detailed examples as well as analyzing the proposed method for computing a skyband-set query in Section IV. In addition, we discuss the objectset pruning strategies in this section. We demonstrate the effectiveness and experimentally enumerate our proposed algorithm in a variety of settings in Section V. In Section VI, we conclude the proposals and give some suggestions for future research.

II. RELATED WORK

There are two closely related works. One is “top- k combinatorial skyline queries” [9]. The other is “convex skyline objectsets” [8].

Su et al. [9] examined a method to obtain the optimal top- k combinations according to the preference order of attributes. They retrieve combinations, which are not dominated by another, incrementally according to the preference until the best k results are found. However, their method depends on the preference order of attributes and the required number of combinations (k value). The preference order and the limited number value k help to reduce the search space, which is exponential for combinations. In contrast, our proposed method does not require a attribute preference order nor the number of combinations. Therefore, their method is not suitable for solving our problem. Moreover, it is very difficult to select the appropriate preference order, which restricts the applicability of this method. Siddique et al. introduced the “convex skyline objectset” problem. In the problem, objectsets that are on the upper convex hull are all the skyline objectsets. However, the convex hull is a subset of the skyline. It means some of the skyline may not be on the convex hull. Our previously proposed method depends on the properties of a convex hull. Therefore, it cannot be applied to the skyband-set query retrieval problem.

In addition, “combination skyline queries” were introduced in [4], which described an index-based method called PBP for determining the skyline objectsets where they indexed each individual object. However, PBP also has some limitations. The key problem is that it is very difficult to specify the object selection pattern in advance because the end users are unfamiliar with the PBP algorithm. The pruning capability of the BPB algorithm depends on this pattern selection step. Thus, if a user selects an incorrect pattern, this can exponentially increase the search space for the computation of objectsets. Another limitation of the BPB algorithm is that it does not work well as the cardinality of the objectset size s changes. It only works well for a certain adequate size of s . Thus, it is necessary to start from scratch in order to set s adequately. By contrast, our solution does not require the construction of a pattern in advance. Therefore, there is no possibility of the search space

increasing exponentially. Moreover, it is possible to vary the objectset size s in our proposed method. Some previous studies also considered the combination selection problem, but they are not related to our proposed method. In particular, Roy et al. [7] discussed how to choose the “maximal combinations”. In the paper, a combination can be considered as “maximal”, if it exceeds the pre-specified constraint during incremental adding procedure of an object. Finally, they selected the k most representative maximal combinations, thereby outputting objects with high diversity [7]. Wan et al. [10] studied the issue of constructing k profitable products from a new product set that is non-dominated by other existing products on the market. They constructed non-dominant products by allocating prices to the new products, which differed from the existing products. However, no previous studies considered skyband-set querying. Thus, existing methods are not suitable for solving objectset queries.

III. PRELIMINARIES

Given a data set D with m -attributes $\{a_1, a_2, \dots, a_m\}$ and n objects $\{O_1, O_2, \dots, O_n\}$, we use $O_i.a_j$ to denote the j -th attribute’s value for object O_i . Without loss of generality, let us assume that the domain of each attribute has positive numerical values and a lower value is desirable for each attribute.

Definition Dominance:

An object $O_i \in D$ can dominate another object $O_j \in D$, denoted as $O_i \leq O_j$, if $O_i.a_r \leq O_j.a_r$ ($1 \leq r \leq m$) for all m attributes and $O_i.a_t < O_j.a_t$ ($1 \leq t \leq m$) for at least one attribute. We refer to O_i as the *dominant object* and O_j as the *dominated object* between O_i and O_j . If O_i dominates O_j , then O_i is more desirable than O_j .

In the table of Fig. 1, object O_1 dominates object O_7 , i.e., $O_1 \leq O_7$ because object O_1 has smaller values for both attributes than object O_7 .

Definition Skyline:

An object $O_i \in D$ is referred to as a *skyline object* of D if and only if no other object $O_j \in D$ ($j \neq i$) exists that can dominates O_i . The set of skyline objects in D is denoted by $Sky(D)$. Let us consider the seven stocks example, where object O_2 dominates $\{O_4, O_5, O_6, O_7\}$. No object can dominate objects $\{O_1, O_3\}$ in D . Therefore, a skyline query outputs the result of $Sky(D) = \{O_1, O_2, O_3\}$ (see Fig. 1).

Definition s -objectset:

An s -objectset, say OS , is a set of s objects chosen from D , i.e., $OS = \{O_1, \dots, O_s\}$. To ensure that the representation remains simple, we denote s objects as $OS = OS_{1, \dots, s}$, where each attribute value for OS is calculated using the following formula:

$$OS.a_j = f_j(O_1.a_j, \dots, O_s.a_j), (1 \leq j \leq m), \quad (1)$$

where f_j represent a monotonic aggregate function that receives s parameters as inputs and produces a single value. We can apply any monotonic aggregate function in our proposed method. However, for simplicity, we apply sum as the

aggregate function, which returns the aggregate values for s objects as follows:

$$OS.a_j = \sum_{i=1}^s O_i.a_j, (1 \leq j \leq m) \quad (2)$$

Notice that the total number of s -objectsets in D is ${}_nC_s = \frac{n!}{(n-s)!s!}$ and we denote the number by $|S|$.

Definition Dominance Relationship:

An s -objectset $OS \in D$ will dominate another s -objectset $OS' \in D$, denoted as $OS \leq OS'$, if $OS.a_r \leq OS'.a_r$ ($1 \leq r \leq m$) for all m attributes and $OS.a_t < OS'.a_t$ ($1 \leq t \leq m$) for at least one attribute. We refer to this OS as the dominant s -objectset and we also refer to OS' as the dominated s -objectset.

Definition Objectset Skyline:

Let OS be an s -objectset in D . If OS is not dominated by any other OS in D , we call it a skyline OS (s -objectset). The skyline of s -objectsets in D , represented by $Sky_s(D)$, is the set of skyline s -objectsets in D . If we consider $s = 2$, for the data set shown in Table I, 2-objectsets $OS_{1,2}, OS_{2,3}$, and $OS_{2,6}$ are not dominated by any other 2-objectsets in D . Therefore, the 2-objectset skyline query outputs $Sky_2(D) = \{OS_{1,2}, OS_{2,3}, OS_{2,6}\}$ (see Fig. 2).

Definition Skyband-set:

A skyband-set query retrieves a set of objectsets, where each individual objectset is not dominated by k other objectsets. It also means, an objectset in the skyband-set query can be dominated by at most $k - 1$ other objectsets. If we consider $s = 2$ and $k = 2$, a skyband-set query based on data set D outputs $\{OS_{1,2}, OS_{1,3}, OS_{1,6}, OS_{2,3}, OS_{2,5}, OS_{2,6}, OS_{3,4}\}$ as the query results (see Fig. 4).

Recall that any top- k query result (result of a top- k query based on an arbitrary linear function) is contained in the results of the skyband set. Therefore, we can use the skyband set results as a pre-processing step for skyline and top- k query computation.

IV. SKYBAND-SET ALGORITHM

In this section, we present the details of the proposed skyband-set method, which is a levelwise search algorithm. First, it calculates the 1-objectsets skyband, before computing all of the 2-objectsets skyband, and this procedure continues up to s -objectsets.

First, we assume that the skyband-set query is for objectset size $s = 1$ and skyband size $K = 1$. This query is similar to a conventional query and it produces similar non-dominant objects as the results set. Any traditional method is suitable for retrieving the results for this initial query. Therefore, we use the *SFS* algorithm developed in [3] to calculate the skyband-set query with $s = 1$ and $k = 1$. After completing all of the domination checks, we obtain the domination relation table shown in Table II, which we refer to as the *domRelationTable*.

Table II shows that objects O_1, O_2 , and O_3 are not dominated by another objects. Therefore, for $s = 1$ and $k = 1$,

TABLE II. DOMINANCE RELATIONSHIP AMONG 1-OBJECTSETS

Object	Dominant Object
O_1	\emptyset
O_2	\emptyset
O_3	\emptyset
O_4	O_2, O_3
O_5	O_2, O_6
O_6	O_2
O_7	$O_1, O_2, O_3, O_4, O_5, O_6$

the skyband-set query output is $\{O_1, O_2, O_3\}$. Next, if we retain the objectset size $s = 1$ and increase the skyband value to $k = 2$, then the skyband-set output becomes $\{O_1, O_2, O_3, O_6\}$. Similarly, a skyband-set query for $s = 1$ and $k = 3$ will retrieve $\{O_1, O_2, O_3, O_4, O_5, O_6\}$, and the query for $s = 1$ and $K = 4$ will retrieve all objects as outputs.

As discussed regarding the skyband-set problem, if we select the objectset size as equal to s , then a data set D with n objects can retrieve the total number objectsets as $|S| = {}_nC_s$. For a large value of n , this represents a severe algorithmic challenge compared with the traditional skyline computation problem. Fig. 2, 4, and 5 show that for $s = 2$, the total number of possible sets is $|S| = 21$ (${}_7C_2$). Therefore, in order to produce *domRelationTable* in a similar manner to Table II, we must conduct a domination check based on 420 ($21 * 20$) comparisons. Thus, even for a small data set containing thousands of objects, the total number of objectsets is remarkably large and vast numbers of comparisons are required. However, given Theorem 1 and Theorem 2, we are free to obtain the dominance relationships for the objectsets without composing them. This also avoids a large number of unnecessary comparisons in the domination check.

Theorem 1: Let OS_1, OS_2 , and OS_3 be three s -objectsets in D . If $OS_1 \leq OS_2$, then $OS_1OS_3 \leq OS_2OS_3$, where OS_1OS_3 is the $2s$ -objectset that includes the $2s$ objects that are included in OS_1 or OS_3 , and similarly, OS_2OS_3 is the $2s$ -objectset that includes the $2s$ objects that are included in OS_2 or OS_3 .

Proof: Let OS and OS' be two s -objectsets, and let O be an object that is not included in OS or OS' in an m -dimensional database D . Assume that $OS \leq OS'$. If we add O to OS and obtain the superset of OS , which is $O \cup OS$, then by the definition, $(O \cup OS).a_j = O.a_j + \sum_{i=1}^s O_i.a_j, (1 \leq j \leq m)$. Similarly, we can make $O \cup OS'$. We can say that $(O \cup OS).a_j \leq (O \cup OS').a_j$ for all j and $(O \cup OS).a_j < (O \cup OS').a_j$ at least one j ($1 \leq j \leq m$) by the assumption; therefore, $(O \cup OS) \leq (O \cup OS')$. \square

Theorem 2: If an objectset OS is dominated by at least k other objectsets, then all supersets that contain OS cannot be a member of the k -skyband-set. Therefore, further skyband-set calculations do not require the composition of the super-objectsets that contain OS .

Proof: Assume that a data set D has four objectsets: OS_1, OS_2, OS_3 , and OS_4 . If objectset OS_1 is dominated by OS_2 and OS_3 , then the proposed algorithm does not require that we compose super-objectset OS_1OS_4 for $K = 2$. By using Theorem 1, if $OS_2 \leq OS_1$, then $OS_2OS_4 \leq OS_1OS_4$. This is also true for $OS_3OS_4 \leq OS_1OS_4$. Thus, two objectsets dominate OS_1OS_4 . \square

TABLE III. DOMINANCE RELATIONSHIP AMONG 2-OBJECTSETS

Objectset	Dominant Objectset	Objectset	Dominant Objectset
$OS_{1,2}$	\emptyset	$OS_{3,4}$	$OS_{2,3}$
$OS_{1,3}$	\emptyset	$OS_{3,5}$	$OS_{2,3}, OS_{3,6}$
$OS_{1,4}$	$OS_{1,2}, OS_{1,3}$	$OS_{3,6}$	$OS_{2,3}$
$OS_{1,5}$	$OS_{1,2}, OS_{1,6}$	$OS_{3,7}$	$OS_{1,3}, OS_{2,3}, OS_{3,4}, OS_{3,5}, OS_{3,6}$
$OS_{1,6}$	$OS_{1,2}$	$OS_{4,5}$	$OS_{2,5}, OS_{3,5}, OS_{2,4}, OS_{4,6}$
$OS_{1,7}$	$OS_{1,2}, OS_{1,3}, OS_{1,4}, OS_{1,5}, OS_{1,6}$	$OS_{4,6}$	$OS_{2,6}, OS_{3,6}, OS_{2,4}$
$OS_{2,3}$	\emptyset	$OS_{4,7}$	$OS_{1,4}, OS_{2,4}, OS_{3,4}, OS_{4,5}, OS_{4,6}, OS_{2,7}, OS_{3,7}$
$OS_{2,4}$	$OS_{2,3}$	$OS_{5,6}$	$OS_{2,6}, OS_{2,5}$
$OS_{2,5}$	$OS_{2,6}$	$OS_{5,7}$	$OS_{1,5}, OS_{2,5}, OS_{3,5}, OS_{4,5}, OS_{5,6}, OS_{2,7}, OS_{6,7}$
$OS_{2,6}$	\emptyset	$OS_{6,7}$	$OS_{1,6}, OS_{2,6}, OS_{3,6}, OS_{4,6}, OS_{5,6}, OS_{2,7}$
$OS_{2,7}$	$OS_{1,2}, OS_{2,3}, OS_{2,4}, OS_{2,5}, OS_{2,6}$		

By utilizing Theorem 1 and 2, we can prune a large number of the unnecessary dominance checks. By checking Table II from top to bottom, we first find that object O_4 is dominated by O_2 and O_3 , from which we derive $O_2 \leq O_4$ and $O_3 \leq O_4$. Each of the relationships with $\{O_1, O_5, O_6, O_7\}$ has not yet been examined. By applying Theorem 2 and without performing any comparisons, we can easily compute the following dominance relationship for 2-objectsets.

$$\begin{aligned} OS_{1,2} &\leq OS_{1,4} & OS_{1,3} &\leq OS_{1,4} & OS_{2,5} &\leq OS_{4,5} \\ OS_{3,5} &\leq OS_{4,5} & OS_{2,6} &\leq OS_{4,6} & OS_{3,6} &\leq OS_{4,6} \\ OS_{2,7} &\leq OS_{4,7} & OS_{3,7} &\leq OS_{4,7} & & \end{aligned}$$

Moreover, we also obtain two more dominance relationship for O_4 , as follows:

$$OS_{2,3} \leq OS_{2,4} \quad OS_{2,3} \leq OS_{3,4}$$

Similarly, object O_5 is dominated by $\{O_2, O_6\}$, from which we derive $OS_2 \leq O_5$ and $O_6 \leq O_5$. These relationships can also be used to derive the following relationships for other objects $\{O_1, O_3, O_4, O_7\}$.

$$\begin{aligned} OS_{1,2} &\leq OS_{1,5} & OS_{1,6} &\leq OS_{1,5} & OS_{2,3} &\leq OS_{3,5} \\ OS_{3,6} &\leq OS_{3,5} & OS_{2,4} &\leq OS_{4,5} & OS_{4,6} &\leq OS_{4,5} \\ OS_{2,7} &\leq OS_{5,7} & OS_{6,7} &\leq OS_{5,7} & & \end{aligned}$$

In addition, we can derive

$$OS_{2,6} \leq OS_{2,5} \quad OS_{2,6} \leq OS_{5,6}.$$

For the sixth relationship $O_2 \leq O_6$ in Table II, the relationships with others are derived as follows:

$$\begin{aligned} OS_{1,2} &\leq OS_{1,6} & OS_{2,3} &\leq OS_{3,6} & OS_{2,4} &\leq OS_{4,6} \\ OS_{2,5} &\leq OS_{5,6} & OS_{2,7} &\leq OS_{6,7} & & \end{aligned}$$

Similarly, for the last relationship in Table II, the following relationships are derived.

$$\begin{aligned} OS_{1,2} &\leq OS_{1,7}, OS_{1,3} \leq OS_{1,7}, OS_{1,4} \leq OS_{1,7}, \\ &OS_{1,5} \leq OS_{1,7}, OS_{1,6} \leq OS_{1,7} \\ OS_{1,2} &\leq OS_{2,7}, OS_{2,3} \leq OS_{2,7}, OS_{2,4} \leq OS_{2,7}, \\ &OS_{2,5} \leq OS_{2,7}, OS_{2,6} \leq OS_{2,7} \\ OS_{1,3} &\leq OS_{3,7}, OS_{2,3} \leq OS_{3,7}, OS_{3,4} \leq OS_{3,7}, \\ &OS_{3,5} \leq OS_{3,7}, OS_{3,6} \leq OS_{3,7} \\ OS_{1,4} &\leq OS_{4,7}, OS_{2,4} \leq OS_{4,7}, OS_{3,4} \leq OS_{4,7}, \\ &OS_{4,5} \leq OS_{4,7}, OS_{4,6} \leq OS_{4,7} \\ OS_{1,5} &\leq OS_{5,7}, OS_{2,5} \leq OS_{5,7}, OS_{3,5} \leq OS_{5,7}, \\ &OS_{4,5} \leq OS_{5,7}, OS_{5,6} \leq OS_{5,7} \end{aligned}$$

$$\begin{aligned} OS_{1,6} &\leq OS_{6,7}, OS_{2,6} \leq OS_{6,7}, OS_{3,6} \leq OS_{6,7}, \\ OS_{4,6} &\leq OS_{6,7}, OS_{5,6} \leq OS_{6,7} \end{aligned}$$

Thus, we can easily construct another ‘‘Dominance Relationship Table’’ for objectset with $s = 2$ using Table II without comparing objectsets according to Theorem 1 and Theorem 2. The new ‘‘Dominance Relationship Table’’ with $s = 2$ is shown in Table III.

Table III is used to retrieve candidates for the objectset skyband queries with $s = 2$ (the objectset size is two). For example, if a user specifies $s = 2$ and $k = 1$ for a skyband-set query, then the proposed algorithm will select the candidate objectsets comprising $\{OS_{1,2}, OS_{1,3}, OS_{2,3}, OS_{2,6}\}$ from Table III. Next, it will compose the four objectsets and perform the domination checks among them. After the domination checks, we find that $OS_{1,3}$ is dominated by objectset $OS_{2,6}$. Thus, the proposed algorithm will output $\{OS_{1,2}, OS_{2,3}, OS_{2,6}\}$ as the skyband-set.

Next, if the user specifies $s = 2$ and $k = 2$ for a skyband-set query, then the proposed algorithm will first select the candidate objectsets $\{OS_{1,2}, OS_{1,3}, OS_{1,6}, OS_{2,3}, OS_{2,4}, OS_{2,5}, OS_{2,6}, OS_{3,4}, OS_{3,6}\}$, before performing domination checks among these candidate objectsets. Finally, it retrieves $\{OS_{1,2}, OS_{1,3}, OS_{1,6}, OS_{2,3}, OS_{2,5}, OS_{2,6}, OS_{3,4}\}$ as the skyband-set query result.

Similar to the example above, we can retrieve candidate objectsets for any skyband-set query with $s = 2$ from the dominance relations in Table III.

Now, if the user wants to select the top-3 objectsets, our proposed algorithm will examine the μ score of each objectset in the skyband set results and select $OS_{1,2}, OS_{2,3}$, and $OS_{2,6}$ with μ scores of 10, 12, and 11, respectively, which are the top-3 scores in the database. In general, the proposed algorithm can retrieve the top- k query from the k -skyband-set results, which can be computed efficiently by the algorithm.

Next, to construct ‘‘Dominance Relationship Table’’ for $s = 3$, we follow the same procedure again and by utilizing Theorems 1 and 2, we can prune a large number of unnecessary dominance checks. After checking Table III, we first find that objectset $OS_{1,4}$ is dominated by $OS_{1,2}$ and $OS_{1,3}$, from which we can derive $OS_{1,2} \leq OS_{1,4}$ and $OS_{1,3} \leq OS_{1,4}$. Each of the relationships with $\{O_5, O_6, O_7\}$ has not yet been examined. By using Theorem 2, we can find the following dominance relationship for 3-objectsets without making any comparisons.

$$OS_{1,2,5} \leq OS_{1,4,5} \quad OS_{1,3,5} \leq OS_{1,4,5}$$

TABLE IV. DOMINANCE RELATIONSHIP AMONG 3-OBJECTSETS

<i>Obj.set</i>	<i>Dominant Objectset</i>	<i>Obj.set</i>	<i>Dominant Objectset</i>
$OS_{1,2,3}$	\emptyset	$OS_{2,3,7}$	$OS_{1,2,3}, OS_{2,3,4}, OS_{2,3,5}, OS_{2,3,6}$
$OS_{1,2,4}$	$OS_{1,2,3}$	$OS_{2,4,5}$	$OS_{2,3,5}, OS_{2,4,6}$
$OS_{1,2,5}$	$OS_{1,2,6}$	$OS_{2,4,6}$	$OS_{2,3,6}$
$OS_{1,2,6}$	\emptyset	$OS_{2,4,7}$	$OS_{1,2,4}, OS_{2,3,4}, OS_{2,3,7}, OS_{2,4,5}, OS_{2,4,6}$
$OS_{1,2,7}$	$OS_{1,2,3}, OS_{1,2,4}, OS_{1,2,5}, OS_{1,2,6}$	$OS_{2,5,6}$	\emptyset
$OS_{1,3,4}$	$OS_{1,2,3}$	$OS_{2,5,7}$	$OS_{1,2,5}, OS_{2,3,5}, OS_{2,4,5}, OS_{2,5,6}, OS_{2,6,7}$
$OS_{1,3,5}$	$OS_{1,2,3}, OS_{1,3,6}$	$OS_{2,6,7}$	$OS_{1,2,6}, OS_{2,3,6}, OS_{2,4,6}, OS_{2,5,6}$
$OS_{1,3,6}$	$OS_{1,2,3}$	$OS_{3,4,5}$	$OS_{2,3,4}, OS_{2,3,5}, OS_{3,4,6}$
$OS_{1,3,7}$	$OS_{1,2,3}, OS_{1,3,4}, OS_{1,3,5}, OS_{1,3,6}$	$OS_{3,4,6}$	$OS_{2,3,4}, OS_{2,3,6}$
$OS_{1,4,5}$	$OS_{1,2,4}, OS_{1,2,5}, OS_{1,3,5}, OS_{1,4,6}$	$OS_{3,4,7}$	$OS_{1,3,4}, OS_{2,3,4}, OS_{2,3,7}, OS_{3,4,5}, OS_{3,4,6}$
$OS_{1,4,6}$	$OS_{1,2,4}, OS_{1,2,5}, OS_{1,2,6}, OS_{1,3,6}$	$OS_{3,5,6}$	$OS_{2,3,5}, OS_{2,3,6}$
$OS_{1,4,7}$	$OS_{1,2,4}, OS_{1,2,7}, OS_{1,3,4}, OS_{1,3,7}, OS_{1,4,5}, OS_{1,4,6}$	$OS_{3,5,7}$	$OS_{1,3,5}, OS_{2,3,5}, OS_{2,3,7}, OS_{3,4,5}, OS_{3,5,6}, OS_{3,6,7}$
$OS_{1,5,6}$	$OS_{1,2,5}, OS_{1,2,6}$	$OS_{3,6,7}$	$OS_{1,3,6}, OS_{2,3,6}, OS_{2,3,7}, OS_{3,4,6}, OS_{3,5,6}$
$OS_{1,5,7}$	$OS_{1,2,5}, OS_{1,2,7}, OS_{1,3,5}, OS_{1,4,5}, OS_{1,5,6}, OS_{1,6,7}$	$OS_{4,5,6}$	$OS_{2,4,5}, OS_{2,4,6}, OS_{2,5,6}, OS_{3,5,6}$
$OS_{1,6,7}$	$OS_{1,2,6}, OS_{1,2,7}, OS_{1,3,6}, OS_{1,4,6}, OS_{1,5,6}$	$OS_{4,5,7}$	$OS_{1,4,5}, OS_{2,4,5}, OS_{2,4,7}, OS_{2,5,7}, OS_{3,4,5}, OS_{3,5,7}, OS_{4,5,6}, OS_{4,6,7}$
$OS_{2,3,4}$	\emptyset	$OS_{4,6,7}$	$OS_{1,4,6}, OS_{2,4,6}, OS_{2,4,7}, OS_{2,6,7}, OS_{3,4,6}, OS_{3,6,7}, OS_{4,5,6}$
$OS_{2,3,5}$	$OS_{2,3,6}$	$OS_{5,6,7}$	$OS_{1,5,6}, OS_{2,5,6}, OS_{2,5,7}, OS_{2,6,7}, OS_{3,5,6}, OS_{4,5,6}$
$OS_{2,3,6}$	\emptyset		

$$OS_{1,2,6} \leq OS_{1,4,6} \quad OS_{1,3,6} \leq OS_{1,4,6}$$

$$OS_{1,2,7} \leq OS_{1,4,7} \quad OS_{1,3,7} \leq OS_{1,4,7}$$

Moreover, we also obtain two more dominance relationship for $OS_{1,4}$, as follows.

$$OS_{1,2,3} \leq OS_{1,2,4} \quad OS_{1,2,3} \leq OS_{1,3,4}$$

Similarly, objectset $OS_{1,5}$ is dominated by $\{OS_{1,2}, OS_{1,6}\}$, from which we derive $OS_{1,2} \leq OS_{1,5}$ and $OS_{1,6} \leq OS_{1,5}$. These relationships are used to derive the following relationships with other objects $\{O_3, O_4, O_7\}$.

$$OS_{1,2,3} \leq OS_{1,3,5} \quad OS_{1,3,6} \leq OS_{1,3,5}$$

$$OS_{1,2,4} \leq OS_{1,4,5} \quad OS_{1,4,6} \leq OS_{1,4,5}$$

$$OS_{1,2,7} \leq OS_{1,5,7} \quad OS_{1,6,7} \leq OS_{1,5,7}$$

In addition, we can derive:

$$OS_{1,2,6} \leq OS_{1,2,5} \quad OS_{1,2,6} \leq OS_{1,5,6}$$

For $OS_{1,2} \leq OS_{1,6}$ and the relationships with $\{O_3, O_4, O_5, O_7\}$, we can derive the following:

$$OS_{1,2,3} \leq OS_{1,3,6} \quad OS_{1,2,4} \leq OS_{1,4,6}$$

$$OS_{1,2,5} \leq OS_{1,5,6} \quad OS_{1,2,7} \leq OS_{1,6,7}$$

Next, for $\{OS_{1,2}, OS_{1,3}, OS_{1,4}, OS_{1,5}, OS_{1,6}\} \leq OS_{1,7}$, we derive the following.

$$OS_{1,2,3} \leq OS_{1,3,7} \quad OS_{1,2,4} \leq OS_{1,4,7}$$

$$OS_{1,2,5} \leq OS_{1,5,7} \quad OS_{1,2,6} \leq OS_{1,6,7}$$

$$OS_{1,2,3} \leq OS_{1,2,7} \quad OS_{1,3,4} \leq OS_{1,4,7}$$

$$OS_{1,3,5} \leq OS_{1,5,7} \quad OS_{1,3,6} \leq OS_{1,6,7}$$

$$OS_{1,2,4} \leq OS_{1,2,7} \quad OS_{1,3,4} \leq OS_{1,3,7}$$

$$OS_{1,4,5} \leq OS_{1,5,7} \quad OS_{1,4,6} \leq OS_{1,6,7}$$

$$OS_{1,2,5} \leq OS_{1,2,7} \quad OS_{1,3,5} \leq OS_{1,3,7}$$

$$OS_{1,4,5} \leq OS_{1,4,7} \quad OS_{1,5,6} \leq OS_{1,6,7}$$

$$OS_{1,2,6} \leq OS_{1,2,7} \quad OS_{1,3,6} \leq OS_{1,3,7}$$

$$OS_{1,4,6} \leq OS_{1,4,7} \quad OS_{1,5,6} \leq OS_{1,5,7}$$

For $OS_{2,3} \leq OS_{2,4}$ and the relationships with $\{O_1, O_5, O_6, O_7\}$, we can derive the following.

$$OS_{1,2,3} \leq OS_{1,2,4} \quad OS_{2,3,5} \leq OS_{2,4,5}$$

$$OS_{2,3,6} \leq OS_{2,4,6} \quad OS_{2,3,7} \leq OS_{2,4,7}$$

For $OS_{2,6} \leq OS_{2,5}$ and the relationships with $\{O_1, O_3, O_4, O_7\}$, we can derive the following.

$$OS_{1,2,6} \leq OS_{1,2,5} \quad OS_{2,3,6} \leq OS_{2,3,5}$$

$$OS_{2,4,6} \leq OS_{2,4,5} \quad OS_{2,6,7} \leq OS_{2,5,7}$$

Next, for $\{OS_{1,2}, OS_{2,3}, OS_{2,4}, OS_{2,5}, OS_{2,6}\} \leq OS_{2,7}$, we can derive the following relationships:

$$OS_{1,2,3} \leq OS_{2,3,7} \quad OS_{1,2,3} \leq OS_{1,2,7}$$

$$OS_{1,2,4} \leq OS_{2,4,7} \quad OS_{2,3,4} \leq OS_{2,4,7}$$

$$OS_{1,2,5} \leq OS_{2,5,7} \quad OS_{2,3,5} \leq OS_{2,5,7}$$

$$OS_{1,2,6} \leq OS_{2,6,7} \quad OS_{2,3,6} \leq OS_{2,6,7}$$

$$OS_{1,2,4} \leq OS_{1,2,7} \quad OS_{1,2,5} \leq OS_{1,2,7}$$

$$OS_{2,3,4} \leq OS_{2,3,7} \quad OS_{2,3,5} \leq OS_{2,3,7}$$

$$OS_{2,4,5} \leq OS_{2,5,7} \quad OS_{2,4,5} \leq OS_{2,4,7}$$

$$OS_{2,4,6} \leq OS_{2,6,7} \quad OS_{2,5,6} \leq OS_{2,6,7}$$

$$OS_{1,2,6} \leq OS_{1,2,7} \quad OS_{2,3,6} \leq OS_{2,3,7}$$

$$OS_{2,4,6} \leq OS_{2,4,7} \quad OS_{2,5,6} \leq OS_{2,5,7}$$

Similarly, for $OS_{2,3} \leq OS_{3,4}$ and the relationships with $\{O_1, O_5, O_6, O_7\}$, we can derive the following.

$$OS_{1,2,3} \leq OS_{1,3,4} \quad OS_{2,3,5} \leq OS_{3,4,5}$$

$$OS_{2,3,6} \leq OS_{3,4,6} \quad OS_{2,3,7} \leq OS_{3,4,7}$$

Objectset $OS_{3,5}$ is dominated by $\{OS_{2,3}, OS_{3,6}\}$, from which we derive $OS_{2,3} \leq OS_{3,5}$ and $OS_{3,6} \leq OS_{3,5}$. These relationships can be used to derive the following relationships with other objects comprising $\{O_1, O_4, O_7\}$.

$$OS_{1,2,3} \leq OS_{1,3,5} \quad OS_{1,3,6} \leq OS_{1,3,5}$$

$$OS_{2,3,4} \leq OS_{3,4,5} \quad OS_{3,4,6} \leq OS_{3,4,5}$$

$$OS_{2,3,7} \leq OS_{3,5,7} \quad OS_{3,6,7} \leq OS_{3,5,7}$$

In addition, we can derive:

$$OS_{2,3,6} \leq OS_{3,5,6} \quad OS_{2,3,6} \leq OS_{2,3,5}$$

Again, for $OS_{2,3} \leq OS_{3,6}$ and the relationships with $\{O_1, O_4, O_5, O_7\}$, we can derive the following:

$$OS_{1,2,3} \leq OS_{1,3,6} \quad OS_{2,3,4} \leq OS_{3,4,5}$$

$$OS_{2,3,5} \leq OS_{3,5,6} \quad OS_{2,3,7} \leq OS_{3,6,7}$$

Then, for $\{OS_{1,3}, OS_{2,3}, OS_{3,4}, OS_{3,5}, OS_{3,6}\} \leq OS_{3,7}$, we can derive the following relationships:

$$OS_{1,2,3} \leq OS_{2,3,7} \quad OS_{1,2,3} \leq OS_{1,3,7}$$

$$OS_{1,3,4} \leq OS_{3,4,7} \quad OS_{2,3,4} \leq OS_{3,4,7}$$

$$OS_{1,3,5} \leq OS_{3,5,7} \quad OS_{2,3,5} \leq OS_{3,5,7}$$

$$\begin{aligned}
 OS_{1,3,6} &\leq OS_{3,6,7} & OS_{2,3,6} &\leq OS_{3,6,7} \\
 OS_{1,3,4} &\leq OS_{1,3,7} & OS_{1,3,5} &\leq OS_{1,3,7} \\
 OS_{2,3,4} &\leq OS_{2,3,7} & OS_{2,3,5} &\leq OS_{2,3,7} \\
 OS_{3,4,5} &\leq OS_{3,5,7} & OS_{3,4,5} &\leq OS_{3,4,7} \\
 OS_{3,4,6} &\leq OS_{3,6,7} & OS_{3,5,6} &\leq OS_{3,6,7} \\
 OS_{1,3,6} &\leq OS_{1,3,7} & OS_{2,3,6} &\leq OS_{2,3,7} \\
 OS_{3,4,6} &\leq OS_{3,4,7} & OS_{3,5,6} &\leq OS_{3,5,7}
 \end{aligned}$$

For $OS_{2,4}, OS_{2,5}, OS_{3,5}, OS_{4,6} \leq OS_{4,5}$ and the relationships with $\{O_1, O_7\}$, we can derive the following:

$$\begin{aligned}
 OS_{1,2,4} &\leq OS_{1,4,5} & OS_{2,4,7} &\leq OS_{4,5,7} \\
 OS_{1,2,5} &\leq OS_{1,4,5} & OS_{2,5,7} &\leq OS_{4,5,7} \\
 OS_{1,3,5} &\leq OS_{1,4,5} & OS_{3,5,7} &\leq OS_{4,5,7} \\
 OS_{1,4,6} &\leq OS_{1,4,5} & OS_{4,6,7} &\leq OS_{4,5,7}
 \end{aligned}$$

In addition, we can derive the following:

$$\begin{aligned}
 OS_{2,3,4} &\leq OS_{3,4,5} & OS_{2,3,5} &\leq OS_{3,4,5} \\
 OS_{2,4,6} &\leq OS_{4,5,6} & OS_{2,5,6} &\leq OS_{4,5,6} \\
 OS_{2,3,5} &\leq OS_{2,4,5} & OS_{2,4,6} &\leq OS_{2,4,5} \\
 OS_{3,5,6} &\leq OS_{4,5,6} & OS_{3,4,6} &\leq OS_{3,4,5}
 \end{aligned}$$

For $OS_{2,4}, OS_{2,6}, OS_{3,6} \leq OS_{4,6}$ and the relationships with $\{O_1, O_5, O_7\}$, we can derive the following:

$$\begin{aligned}
 OS_{1,2,4} &\leq OS_{1,4,6} & OS_{2,4,5} &\leq OS_{4,5,6} \\
 OS_{1,2,6} &\leq OS_{1,4,6} & OS_{2,5,6} &\leq OS_{4,5,6} \\
 OS_{1,3,6} &\leq OS_{1,4,6} & OS_{3,5,6} &\leq OS_{4,5,6}
 \end{aligned}$$

In addition, we can derive the following:

$$\begin{aligned}
 OS_{2,4,7} &\leq OS_{4,6,7} & OS_{2,3,4} &\leq OS_{3,4,6} \\
 OS_{2,6,7} &\leq OS_{4,6,7} & OS_{2,3,6} &\leq OS_{3,4,6} \\
 OS_{3,6,7} &\leq OS_{4,6,7} & OS_{2,3,6} &\leq OS_{2,4,6}
 \end{aligned}$$

For $OS_{1,4}, OS_{2,4}, OS_{2,7}, OS_{3,4}, OS_{3,7}, OS_{4,5}, OS_{4,6} \leq OS_{4,7}$, the following relationships can be derived.

$$\begin{aligned}
 OS_{1,2,4} &\leq OS_{2,4,7} & OS_{1,3,4} &\leq OS_{3,4,7} \\
 OS_{1,4,5} &\leq OS_{4,5,7} & OS_{1,4,6} &\leq OS_{4,6,7} \\
 OS_{1,2,4} &\leq OS_{1,4,7} & OS_{2,3,4} &\leq OS_{3,4,7} \\
 OS_{2,4,5} &\leq OS_{4,5,7} & OS_{2,4,6} &\leq OS_{4,6,7} \\
 OS_{1,2,7} &\leq OS_{1,4,7} & OS_{2,3,7} &\leq OS_{3,4,7} \\
 OS_{2,5,7} &\leq OS_{4,5,7} & OS_{2,6,7} &\leq OS_{4,6,7} \\
 OS_{1,3,4} &\leq OS_{1,4,7} & OS_{2,3,4} &\leq OS_{2,4,7} \\
 OS_{3,4,5} &\leq OS_{4,5,7} & OS_{3,4,6} &\leq OS_{4,6,7} \\
 OS_{1,3,7} &\leq OS_{1,4,7} & OS_{2,3,7} &\leq OS_{2,4,7} \\
 OS_{3,5,7} &\leq OS_{4,5,7} & OS_{3,6,7} &\leq OS_{4,6,7} \\
 OS_{1,4,5} &\leq OS_{1,4,7} & OS_{2,4,5} &\leq OS_{2,4,7} \\
 OS_{3,4,5} &\leq OS_{3,4,7} & OS_{4,5,6} &\leq OS_{4,6,7} \\
 OS_{1,4,6} &\leq OS_{1,4,7} & OS_{2,4,6} &\leq OS_{2,4,7} \\
 OS_{3,4,5} &\leq OS_{3,4,7} & OS_{4,5,6} &\leq OS_{4,5,7}
 \end{aligned}$$

For $OS_{2,5}, OS_{2,6} \leq OS_{5,6}$ and the relationships with $\{O_1, O_3, O_4, O_7\}$, we can derive the following:

$$\begin{aligned}
 OS_{1,2,5} &\leq OS_{1,5,6} & OS_{1,2,6} &\leq OS_{1,5,6} \\
 OS_{2,3,5} &\leq OS_{3,5,6} & OS_{2,3,6} &\leq OS_{3,5,6} \\
 OS_{2,4,5} &\leq OS_{4,5,6} & OS_{2,4,6} &\leq OS_{4,5,6} \\
 OS_{2,5,7} &\leq OS_{5,6,7} & OS_{2,6,7} &\leq OS_{5,6,7}
 \end{aligned}$$

For $OS_{1,5}, OS_{2,5}, OS_{2,7}, OS_{3,5}, OS_{4,5}, OS_{5,6}, OS_{6,7} \leq OS_{5,7}$, the following relationships can be derived:

$$\begin{aligned}
 OS_{1,2,5} &\leq OS_{2,5,7} & OS_{1,2,5} &\leq OS_{1,5,7} \\
 OS_{1,2,7} &\leq OS_{1,5,7} & OS_{1,3,5} &\leq OS_{1,5,7} \\
 OS_{1,4,5} &\leq OS_{1,5,7} & OS_{1,5,6} &\leq OS_{1,5,7} \\
 OS_{1,6,7} &\leq OS_{1,5,7} & OS_{1,3,5} &\leq OS_{3,5,7} \\
 OS_{2,3,5} &\leq OS_{3,5,7} & OS_{2,3,7} &\leq OS_{3,5,7} \\
 OS_{2,3,5} &\leq OS_{2,5,7} & OS_{2,4,5} &\leq OS_{2,5,7} \\
 OS_{2,5,6} &\leq OS_{2,5,7} & OS_{2,6,7} &\leq OS_{2,5,7} \\
 OS_{1,4,5} &\leq OS_{4,5,7} & OS_{2,4,5} &\leq OS_{4,5,7} \\
 OS_{2,4,7} &\leq OS_{4,5,7} & OS_{3,4,5} &\leq OS_{4,5,7} \\
 OS_{3,4,5} &\leq OS_{3,5,7} & OS_{3,5,6} &\leq OS_{3,5,7} \\
 OS_{3,6,7} &\leq OS_{3,5,7} & OS_{1,5,6} &\leq OS_{5,6,7} \\
 OS_{2,5,6} &\leq OS_{5,6,7} & OS_{2,6,7} &\leq OS_{5,6,7} \\
 OS_{3,5,6} &\leq OS_{5,6,7} & OS_{4,5,6} &\leq OS_{5,6,7} \\
 OS_{4,5,6} &\leq OS_{4,5,7} & OS_{4,6,7} &\leq OS_{4,5,7}
 \end{aligned}$$

Finally, for $OS_{1,6}, OS_{2,6}, OS_{2,7}, OS_{3,6}, OS_{4,6}, OS_{5,6} \leq OS_{6,7}$, the following relationships can be derived:

$$\begin{aligned}
 OS_{1,2,6} &\leq OS_{2,6,7} & OS_{1,2,6} &\leq OS_{1,6,7} \\
 OS_{1,2,7} &\leq OS_{1,6,7} & OS_{1,3,6} &\leq OS_{1,6,7} \\
 OS_{1,4,6} &\leq OS_{1,6,7} & OS_{1,5,6} &\leq OS_{1,6,7} \\
 OS_{1,3,6} &\leq OS_{3,6,7} & OS_{2,3,6} &\leq OS_{3,6,7} \\
 OS_{2,3,7} &\leq OS_{3,6,7} & OS_{2,3,6} &\leq OS_{2,6,7} \\
 OS_{2,4,6} &\leq OS_{2,6,7} & OS_{2,5,6} &\leq OS_{2,6,7} \\
 OS_{1,4,6} &\leq OS_{4,6,7} & OS_{2,4,6} &\leq OS_{4,6,7} \\
 OS_{2,4,7} &\leq OS_{4,6,7} & OS_{3,4,6} &\leq OS_{4,6,7} \\
 OS_{3,4,6} &\leq OS_{3,6,7} & OS_{3,5,6} &\leq OS_{3,6,7} \\
 OS_{1,5,6} &\leq OS_{5,6,7} & OS_{2,5,6} &\leq OS_{5,6,7} \\
 OS_{2,5,7} &\leq OS_{5,6,7} & OS_{3,5,6} &\leq OS_{5,6,7} \\
 OS_{4,5,6} &\leq OS_{5,6,7} & OS_{4,5,6} &\leq OS_{4,6,7}
 \end{aligned}$$

Now, by using all of the relationships given above, we can compute ‘‘Dominance Relationship Table’’ for $s = 3$ from ‘‘Dominance Relationship Table’’ for $s = 2$, as shown in Table IV.

To compute the skyband for $s = 3$ and $k = 1$, it is sufficient to compare $\{OS_{1,2,3}, OS_{1,2,6}, OS_{2,3,4}, OS_{2,3,6}, OS_{2,5,6}\}$. We find that $OS_{1,2,6} \leq OS_{2,5,6}$ and the skyband set query retrieves the results of $\{OS_{1,2,3}, OS_{1,2,6}, OS_{2,3,4}, OS_{2,3,6}\}$. Similarly, for $s = 3$ and $k = 2$, the proposed method will retrieve $\{OS_{1,2,3}, OS_{1,2,5}, OS_{1,2,6}, OS_{2,3,4}, OS_{2,3,5}, OS_{2,3,6}\}$. Again, for $k = 3$, the proposed method retrieves $\{OS_{1,2,3}, OS_{1,2,5}, OS_{1,2,6}, OS_{2,3,4}, OS_{2,3,5}, OS_{2,3,6}, OS_{2,4,6}\}$ as the objectset skyband results.

Now, if a user wants to select the top-3 objectsets for $s = 3$, then our proposed algorithm will examine the μ score of each objectset in the skyband set results and select $OS_{1,2,3}, OS_{2,3,6}$, and $OS_{1,2,6}$ with μ scores of 8, 7, and 6, respectively, which are the top-3 scores in the database. By applying the same procedure, the proposed method constructs the dominance relation table for higher objectsets, i.e., for higher values of s , in order to answer skyband-set queries and to provide the top-k objectset for any value of s .

V. PERFORMANCE EVALUATION

We have examined the proposed method through intensive experiments in a simulated environment on a computer with the Windows 64-bit operating system. The configuration of this computer comprised a core i7 CPU 3.4 GHz, 4 GB RAM, and 250 GB SATA disk. No existing approach can discover the skyband set, so we adapted the SFS method to calculate

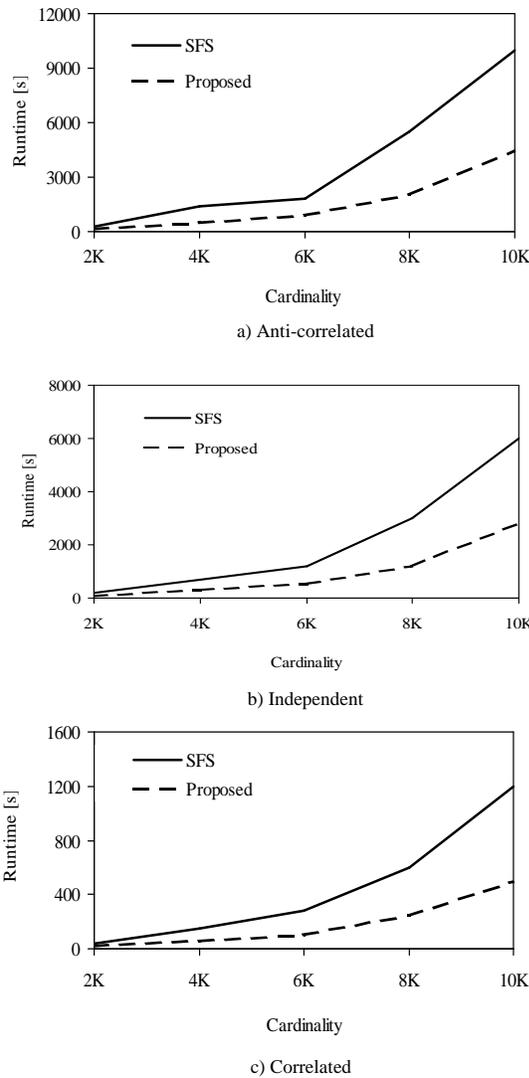


Fig. 6. Performance with different cardinalities.

the skyband set and compared the performance of our method with the skyband-set algorithm proposed in [3].

To ensure that the comparison was fair, we ignored the objectset computation cost for the SFS algorithm. We performed the experiments with different data cardinalities (n), objectset sizes (s), and dimensionalities (m) in order to compare the efficiency and performance of our proposed method. Each experiment was conducted five times. We determined the average output to assess the performance.

A. Performance Based on Synthetic Data Sets

We used benchmark synthetic data sets with three different data distributions, i.e., anti-correlated, correlated, and independent, as proposed in [1].

Correlated: This type of phenomenon occurs in an environment where an objectset is good in one dimension but it also has a better value in other dimensions. In this relationship, a small number of objectsets can dominate a large volume of other objectsets.

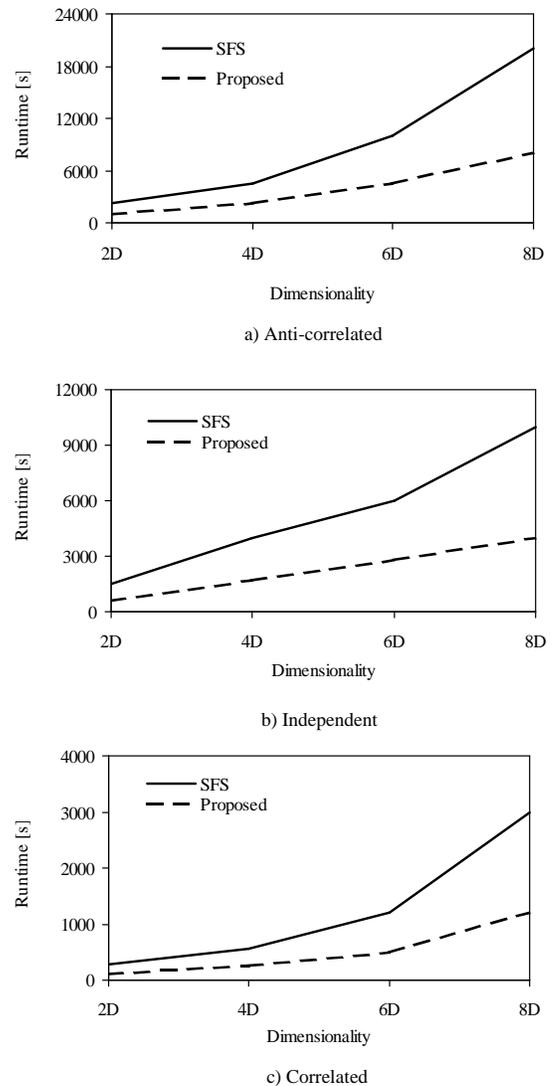


Fig. 7. Performance with different data dimensions.

Anti-Correlated: In this relationship, an objectset with small coordinate values on some dimensions is expected to have higher coordinate values on other dimensions or at least in another dimension.

Independent: In this data distribution, the values of all dimensions are generated independently. In this relationship, the total non-dominant objectsets lie between the resulting objectsets of the correlated and anti-correlated data sets. All of the experimental results obtained based on the synthetic data set are shown in Fig. 6, 7, and 8. The total number of objectsets for synthetic data sets ranged from 2 k to 10 k depending on the data cardinality (n).

1) *Cardinality Effect:* We studied the impact of the cardinality by keeping the data dimensionality m as 6 and the objectset size s as 3, but we changed the data set cardinality n from 2 k to 10 k. Fig. 6 (a), (b), and (c) shows the result obtained with the anti-correlated, independent, and correlated data sets, respectively.

The x-axis and y-axis represent the data cardinality and

execution time, respectively. The graphs show clearly that both techniques were affected by the data cardinality n . As the data cardinality value increased, the execution time increased considerably with both methods. However, our method performed better compared with SFS. For the anti-correlated data distribution, the performance of SFS declined dramatically as the data set size increased. Thus, the proposed method did not require the composition of all the objectsets and it also avoided many unnecessary comparisons.

2) *Dimensionality Effect*: We also assessed the impact of the dimensionality on our proposed method. The dataset dimensionality m was varied from 2 to 8, but we kept the data cardinality n as 10 k and the objectset size s as 3. Fig. 7 (a), (b), and (c) show the runtime requirements for the anti-correlated, independent, and correlated data sets, respectively. The performance of both methods became slower as the dimension size increased because the non-dominant objectset increased with higher dimensions. The graphs show that the proposed method performed better than SFS in all cases, where it was 15 and 10 times faster than SFS with the anti-correlated and independent data sets, respectively.

3) *Objectset Size Effect*: We examined the performance of the proposed method with different objectset size of s in this experiments. We fixed the data set dimensionality m to 6 and the data cardinality n to 10 k. The results are shown in Fig. 8 (a), (b), and (c), where the x-axis represents the objectset size s and the y-axis represents the runtime. The performance of both methods decreased as the objectset size became higher. The results indicated that when the objectset size became $s > 1$, the performance of the SFS method was much worse than our method. This is because when $s = 1$, the proposed method needed to construct *domRelationTable* as well as performing domination checks for all of the objects. However, it was not necessary to calculate all of the objectsets subsequently, thereby avoiding numerous dimension checks.

The experimental results suggested that for synthetic data sets, our proposed method was efficient in terms of the runtime as well as being highly scalable. The results indicated that our method was 4-10 times faster than SFS. This is because the skyband set number for anti-correlated data sets was greater than that for independent data sets, which was also true for the correlated data sets. Therefore, both techniques required more computational time for the anti-correlated distribution than other two distributions.

B. Performance Based on a Real Data Set

We also observed the impact of the proposed algorithm using a real data set, where we selected the FUEL data set, which can be downloaded from “www.fueleconomy.gov”. The volume of the FUEL data set was 24 k and it contained six-dimensional objects, where each object represented the performance attribute of a vehicle, e.g., mileage/gallon gasoline in a city and on a highway. The domain of each attribute varied from 8–89.

We conducted experiments similar to those using the synthetic data set based on the FUEL data set. Initially, we observed the effect of the dimension. We set the data cardinality n to 10 k, the objectset size s to 3, and varied the data set dimensionality m from 2–8. Fig. 9(a) shows the results

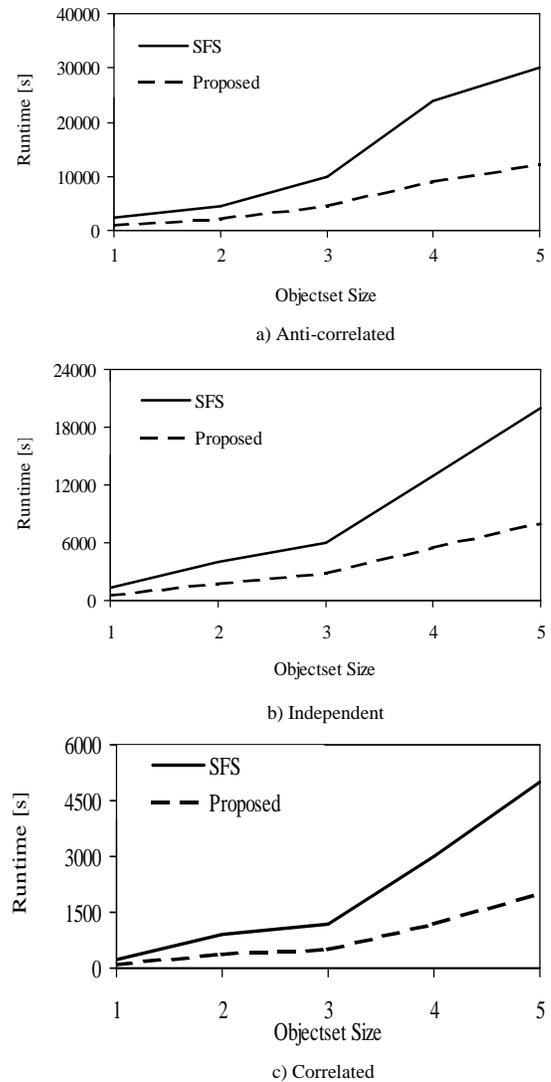


Fig. 8. Performance with different objectset sizes.

obtained for the skyband set queries with various dimension sizes. There was a positive correlation between the dimension size and runtime. However, the proposed method had a shorter runtime than the SFS technique.

Next, we conducted an experiment where we varied the data cardinality n . We fixed the dimensionality m to 4, the objectset size s to 3, and varied the data set cardinality n from 2 k to 10 k. Fig. 9(b) shows the results obtained. The computational time required by both algorithms increased rapidly with the data cardinality. The results in Fig. 9(b) show that our method performed better than the SFS method.

In the final experiment, we studied the performance by varying the objectset size s . We set the dimensionality m to 4 and the data cardinality n to 10 k. The outputs are shown in Fig. 9(c). For both methods, the execution time increased with the objectset size s . The computational time required by the proposed method was much shorter than that by SFS. We obtained similar standardized outputs such as an independent data distribution, thereby confirming the scalability of our technique with the FUEL data set, which was also the case

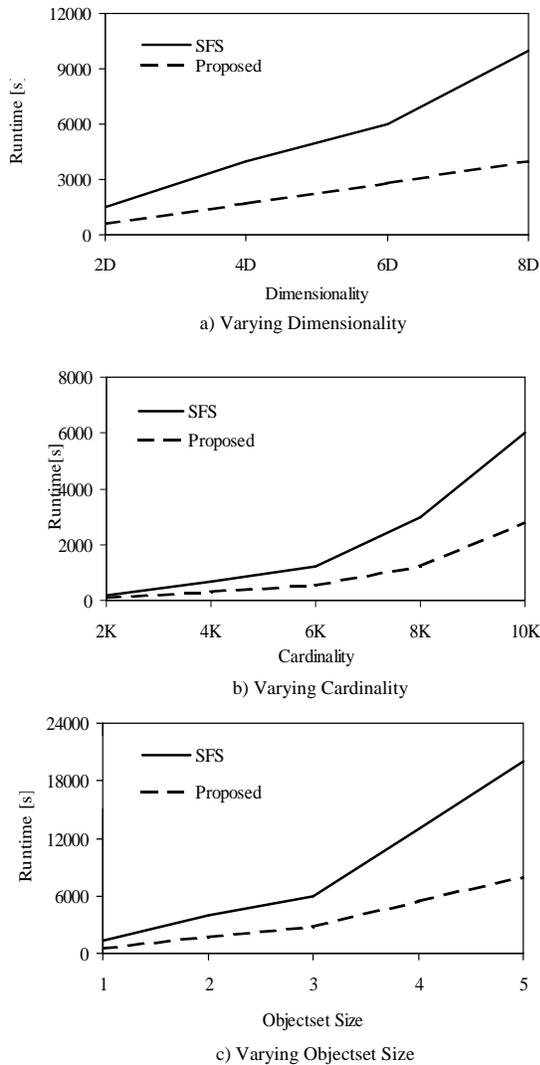


Fig. 9. Experiments based on the FUEL data set.

in all of the experiments.

All of the experiments confirmed that the proposed method was much more efficient than the SFS technique for synthetic and real data sets. Thus, the experiments confirmed the effectiveness and scalability of the proposed method.

VI. CONCLUSION

In this study, we proposed a query method for sets of objects called a “skyband-set.” We proposed an efficient method for calculating the skyband-set output. Our proposed method allows the objectset size to be varied from 1 to s and the skyband size from 1 to k . Furthermore, in order to filter out a large search space and to enhance efficiency, we also developed

domRelationTable, which allows the proposed method to prune out large portions of the search space. We validated our method using synthetic and real data sets. The results in Fig. 6, 7, 8, and 9 confirm the usefulness and superior performance of the proposed method.

In addition to the contributions above, the proposed method has preferable property in privacy-aware environment. In privacy-aware environment, we are not allowed to disclose individual record’s values. This is a significant restriction to analyze a database. Our set-based database analysis does not have to disclose individual record’s values, which implies that the proposed method can be an effective database analysis measure in privacy-aware environment.

In the future, we aim to enhance our proposed algorithm in various areas. In particular, more effective calculations of the objectsets can be performed with a parallel distributed architecture such as MapReduce.

ACKNOWLEDGMENTS

Md. Anisuzzaman Siddique was supported by “JSPS postdoctoral fellowship for foreign researchers program (25.03040)” Japan. Asif Zaman was supported by Japan’s MEXT scholarship program. This work is supported by KAKENHI (16K00155) Japan.

REFERENCES

- [1] S. Borzsony, D. Kossmann, and K. Stocker. The skyline operator. In *Proceedings of IEEE International Conference on Data Engineering (ICDE)*, pages 421–430, 2001.
- [2] S. Bothe, P. Karras, A. Vlachou. eSkyline: Progressing Skyline Queries over Encrypted Data. In *Proceedings of International Conference on Very Large Data Bases (VLDB)*, pages 1338–1341, 2013.
- [3] J. Chomicki, P. Godfrey, J. Gryz, and D. Liang. Skyline with presorting. In *Proceedings of IEEE International Conference on Data Engineering (ICDE)*, pages 717–719, 2003.
- [4] X. Guo, C. Xiao, Y. Ishikawa. Combination Skyline Queries. *Transactions on Large-Scale Data- and Knowledge-Centered Systems VI*, Vol. 7600, pages 1–30, 2012.
- [5] K. C. K. Lee, B. Zheng, H. Li, W.-C. Lee. Approaching the skyline in Z order. In *Proceedings of International Conference on Very Large Data Bases (VLDB)*, pages 279–290, 2007.
- [6] D. Papadias, Y. Tao, G. Fu, B. Seeger. Progressive skyline computation in database systems. *ACM Transactions on Database Systems*, Vol. 30, No. 1, pages 41–82, 2005.
- [7] S. B. Roy, S. Amer-Yahia, A. Chawla, G. Das, C. Yu. Constructing and exploring composite items. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 843–854, 2010.
- [8] M. A. Siddique, Y. Morimoto. Algorithm for computing convex skyline objectsets on numerical databases. *IEICE TRANSACTIONS on Information and Systems*, Vol. E93-D, No. 10, pages 2709–2716, 2010.
- [9] I.-F. Su, Y.-C. Chung, C. Lee. Top-k combinatorial skyline queries. *Proceedings of DASFAA*, pages 79–93, 2010.
- [10] Q. Wan, R. C.-W. Wong, Y. Peng. Finding top-k profitable products. In *Proceedings of IEEE International Conference on Data Engineering (ICDE)*, pages 1055–1066, 2011.