A Region Splitting Strategy for Physical Database Design of Multidimensional File Organizations

Jong-Hak Lee[*] Young-Koo Lee, and Kyu-Young Whang
Computer Science Department
KAIST, Taejon, Korea
{jhlee,yklee,kywhang}@cs.kaist.ac.kr

Il-Yeol Song
College of Information Science and Technology
Drexel University, Philadelphia, U.S.A.
songiy@post.drexel.edu

Abstract

This paper presents a region splitting strategy for physical database design of multidimensional file organizations. Physical database design is the process of determining the optimal configuration of physical files for a given set of queries. Recently, many multidimensional file organizations for supporting multiattribute access have been proposed in the literature. However, there has been no effort for their physical database design. We first show that the performance of query processing is highly affected by the similarity between the shapes of query regions and page regions in the domain space, and then propose a new region splitting strategy that finds the optimal configuration of the multidimensional file by controlling the interval ratio of different axes to achieve the similarity. We also present the results of extensive experiments using the multilevel grid file (MLGF), a multidimensional file organization, and various types of queries and record distributions. The results indicate that our proposed strategy builds optimal MLGFs regardless of query types and record distributions. When the interval ratio of a two-dimensional query region is 1:1024, the performance of the proposed strategy is enhanced by as much as 7.5 times over that of the conventional cyclic splitting strategy. The performance is further enhanced for the query types having higher interval ratios. The result is significant since interval ratios can be far from 1:1 for many practical applications, especially when different axes have different domains.

1 Introduction

One of the most important criteria for evaluating the performance of a database management system(DBMS) is efficiency of query processing, i.e., how quickly the DBMS can respond to users' queries. Query processing is the process of retrieving the records that satisfy query conditions from the database. Physical database design[2, 3, 13] is concerned with determining the optimal configuration of physical files and access structures for supporting efficient query processing[2]. In this paper we present a region splitting strategy for physical database design of multidimensional file organizations supporting multidimensional clustering.

The record clustering method stores the records that are frequently accessed together on the same page so that the total number of pages accessed in response to a set of queries is minimized[17]. If the records are frequently accessed by one attribute, we can maintain the records in the sorted order of the attribute. In this

[*] Currently with Korea Telecom, Seoul, Korea

Proceedings of the 23rd VLDB Conference
Athens, Greece, 1997

case, the clustering property is exclusively owned by one attribute. On the other hand, if the records are frequently accessed by multiple attributes, we can maintain a multidimensional file that stores similar records having similar values for the accessed attributes in the same page. In this case, the clustering property is shared by multiple attributes. To maintain the clustering property, the multidimensional file organizations adapt to dynamic situations where record insertions or deletions could cause the domain space to be split or merged. After each split or merge, the records in a newly created region are stored on the same disk page.

The way we split the domain space into subregions to achieve clustering of records is called the *region splitting strategy*. One of the most widely used region splitting strategies is *cyclic splitting*, which alternately chooses a different axis to split the domain by a predetermined order. The cyclic splitting strategy makes the divided regions regular hyperrectangles in shape, and thus makes the clustering property shared by all of the attributes in the multidimensional file. Therefore, we are able to adjust the degree of clustering for each attribute involved in the clustering by varying the region splitting strategy. Cyclic splitting is based on the assumption that the interval sizes of all the attributes used in the query predicates are the same. In general, however, the sizes of intervals given by the query predicates are different, thus making the regions nonregular hyperrectangles in shape. For example, some attributes could be accessed by exact-match predicates, while others by range predicates. In these cases, the performance of query processing is degraded due to a discrepancy between the real and the assumed query types.

In this paper, we first show that the cost of processing a query is highly affected by the allocation of the clustering property among different attributes, and then propose a design algorithm that finds an optimal region splitting method based on precollected information about the queries.

We first prove analytically the optimality of the proposed method. We then perform extensive experiments using the multilevel grid file(MLGF)[14, 15, 16], a dynamic multidimensional file organization, with various query types and record distributions. Even though we use the MLGF for the experiments, our methodology is applicable to any other kind of multidimensional file organizations. To the best of our knowledge, there has been no study on physical database design of multidimensional file organizations, which is very important in practical applications. We believe that the contribution of this paper is new and provides a new insight in such problems.

This paper is organized as follows. Section 2 presents the problem definition and some terminology. Section 3 analyzes the optimality condition for minimizing the number of page accesses for query processing in two-dimensional file organizations. Section 4 presents a tunable splitting strategy that satisfies this optimality condition. Section 5 describes the physical database design algorithm for N-dimensional file organizations containing N organizing attributes. Section 6 presents the results of performance evaluation of our design method. Finally, Section 7 summarizes and concludes the paper.

2 Problem Definition

We first define some terminology for further discussions[14, 15]. A *file* is a collection of *records*, where a record consists of a list of *attributes*. A subset of these attributes that determines the placement of the records in the file is called the *organizing attributes*. A file has a *multidimensional organization* if it contains more than one organizing attribute[1]. We define the *domain space* as the Cartesian product of the domains of all the organizing attributes. We call any subset of the domain space a *region*.

Multidimensional file organizations can be classified into two categories by the way the boundary value for splitting the region is determined. One uses record-oriented splitting[1, 5, 8, 9, 11], and the other uses region-oriented splitting[4, 6, 10, 12, 15]. *Record-oriented splitting* divides the region into two subregions so that each subregion has the same number of records. Thus, the boundary value for splitting the region is dependent on the record distribution. *Region-oriented splitting* bisects the region regardless of the record distribution. Thus, the boundary value for splitting the region is predetermined independent of the record distribution.

The fact that the boundary value is predetermined gives two advantages. First, the file organization is not affected by the order of record insertions, which is not the case in record-oriented splitting. Second, the shape of the regions can be systematically controlled by choosing a splitting strategy. The physical database design method for multidimensional file organizations proposed in this paper takes advantage of these features of region-oriented splitting.

We define the *physical database design for multidimensional file organizations* as the problem of determining the region splitting strategy to construct an optimal multidimensional file organization, for which the cost of query processing is minimized for a given query pattern. Most conventional multidimensional file organizations are using cyclic splitting that alternates different axes to split the domain space by a predetermined cyclic order. In this case, the query pattern given by the users are not reflected in organizing the file. Alternatively, we can say that squares or regular hyperrectangles are always assumed to be given.

In general, query conditions rarely specify equal-sized intervals for each attribute referred to; rather, those having large range intervals are likely to be

[1] Since all of the attributes mentioned in the following sections are organizing attributes, we simply call them attributes unless otherwise mentioned.

imposed upon some specific attributes. The following query illustrates this aspect: "Find all employees whose ages are between 30 and 50 years old and whose salaries are between $50,000 and $100,000." In this query, the ratio of intervals for *age* and *salary* becomes 1:2500 when the same integer domain is used. In addition, since most of the multidimensional file organizations using region-oriented splitting construct the domain space by using hash functions, the sizes of the intervals for attributes given in queries become different in the hash domain after they are mapped by the hash functions. For example, suppose we map two attributes whose domain sizes are different from each other into the same hash domain. Even though the interval sizes of two attributes specified in a query are the same in the original domain, they would be quite different in the hash domain. For these reasons, the intervals for different attributes of a query are likely to be very different from one another in the hash domain.

We adopt the definition of query region from Robinson[11] as follows: "A query can be expressed by specifying a region, the *query region*. It is convenient to think of a region as a cross product of intervals." According to this definition, the query region of an *exact-match query* is represented by specifying a single point; that of a *partial-match query* by specifying full domains for some of the intervals; and finally, that of a *range query* by specifying a partial domain for at least one interval.

Accordingly, we can interpret query processing as an operation that retrieves the records included in a query region of the domain space. Therefore, the problem of physical database design for multidimensional file organizations can be redefined as the problem of minimizing the number of page regions that intersect with query regions. We define an *interval ratio* as the ratio of intervals for the attributes and use it to represent the shape of a region. Then, we are able to construct an optimal multidimensional file organization by determining the optimal interval ratio of page regions. To determine this ratio, we use a query pattern given by the user, and use the region splitting strategy that partitions the domain space in such a way that the interval ratios of page regions are close to those of query regions. The query pattern can be obtained by collecting the usage statistics of a database during a certain time interval[17] or by analyzing the application profiles provided by database administrators[3].

3 Optimal Condition for Query Processing

In this section, we propose the optimal condition for query processing using the interrelationship between the shapes of query regions and page regions. In order to simplify discussions, we use two-dimensional file organizations having two organizing attributes. We first present and then prove a basic principle that optimal query processing is equivalent to minimizing the average number of page regions intersecting with a single query region given at an arbitrary position of the domain space. We then present the method of determining the optimal interval ratio of page regions that minimizes the total number of page regions intersecting with a set of query regions given by the query pattern. We first approach this problem for uniform distributions of the records in the domain space, and then extend it to nonuniform distributions.

3.1 Optimal Interval Ratio of Page Regions for a Single Query

We show, by Theorem 1, that the number of page regions intersecting with a query region given at an arbitrary position is minimized when the interval ratio of each page region in the domain space is the same as that of the query region, regardless of the size of the page region. (Note that the size of a page region depends on the density of the record distribution in that region.)

Lemma 1 *If an arbitrary query region $q(x) \times q(y)$ (where $q(x)$ and $q(y)$ are intervals on axes x and y, respectively) is given in the two-dimensional domain space, the region where the center point of the query region intersecting with a page region of size B_i could be located is minimized when the interval ratio of the page region is the same as that of the query region.*

Proof: Figure 1 shows the shadowed region LQR where the center point of the query region QR, $q(x) \times q(y)$, intersecting with a page region PR_i of size $B_i(= p_i(x) \times p_i(y))$ can be located. Here, $p_i(x)$ and $p_i(y)$ are intervals of the page region on each axis.

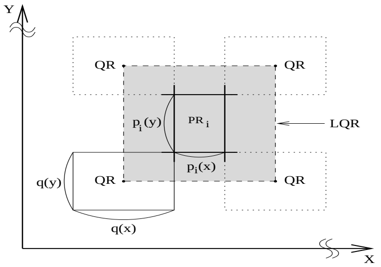


Figure 1: The region where the center point of the query region intersecting with a page region can be located.

The size of LQR is obtained as follows:

$$\begin{aligned} & SIZE\_LQR(p_i(x), p_i(y)) \\ & \quad = \quad (p_i(x) + q(x))(p_i(y) + q(y)) \qquad (1) \end{aligned}$$

Using $p_i(x) \times p_i(y) = B_i$, we replace $p_i(y)$ with $\frac{B_i}{p_i(x)}$ in Eq. (1):

$$SIZE\_LQR(p_i(x), \frac{B_i}{p_i(x)})$$

$$= (p_i(x) + q(x))(\frac{B_i}{p_i(x)} + q(y))$$

$$= B_i + \frac{q(x)\,B_i}{p_i(x)} + p_i(x)\,q(y) + q(x)\,q(y) \quad (2)$$

Thus, the value of $p_i(x)$ that minimizes Eq. (2) is obtained as $p_i(x) = \sqrt{\frac{q(x)}{q(y)} B_i}$. Using $p_i(x) \times p_i(y) = B_i$, $p_i(y) = \sqrt{\frac{q(y)}{q(x)} B_i}$. Hence, the interval ratio of page region PR_i that minimizes $SIZE\_LQR(p_i(x), p_i(y))$ is obtained as $p_i(x) : p_i(y) = q(x) : q(y)$. □

Theorem 1 *If a query region QR, $q(x) \times q(y)$, is given in the two-dimensional domain space partitioned into page regions, the average number of page regions intersecting with the query region is minimized when the interval ratio of each page region is the same as that of the query region[2]. Here, a page region may have an arbitrary size.*

Proof: Let $DX \times DY$ be the size of the domain space consisting of n page regions PR_i, $p_i(x) \times p_i(y)(i = 1, ..., n)$, and $SIZE\_LQR(p_i(x), p_i(y))$ be the size of the region LQR where the center point of the query region QR, $q(x) \times q(y)$, intersecting with a page region PR_i can be located. Then, the probability $P(PR_i)$ that the query region QR intersects with the page region PR_i is as follows:

$$P(PR_i) = \frac{SIZE\_LQR(p_i(x), p_i(y))}{DX \times DY} \quad (3)$$

Thus, the expected number of page regions, NP, intersecting with the query region QR is as follows:

$$NP(QR) = \sum_{i=1}^{n} P(PR_i) = \frac{\sum_{i=1}^{n} SIZE\_LQR(p_i(x), p_i(y))}{DX \times DY} \quad (4)$$

Since, by Lemma 1, $SIZE\_LQR(p_i(x), p_i(y))$ is minimized when the interval ratio of a page region PR_i is the same as that of the query region, $NP(QR)$ is minimized when the interval ratio of each page region is the same as that of the query region. □

Theorem 1 presents the optimal condition for query processing for a single query(i.e., fixed interval ratio). However, the query pattern given by users consists of query regions having various interval ratios. We now discuss the optimal condition for more general case involving multiple queries.

3.2 Optimal Interval Ratio of Page Regions for Multiple Queries When Records are Uniformly Distributed

The following Theorem 2 forms the underlying principle that determines the optimal interval ratio of page regions for the query pattern including query regions of various interval ratios, where the records are uniformly distributed in the two-dimensional domain space. Under the uniform distribution assumption, every page region has the same size due to constant density regardless of the position of the domain space.

Lemma 2 *If a one-dimensional domain space is partitioned into page intervals(i.e. one-dimensional regions) of the same size $p(x)$, the average number of page intervals intersecting with an arbitrary query interval of the size $q(x)$ is $(\frac{q(x)}{p(x)} + 1)$.*

Proof: See the reference [7]. □

Lemma 3 *If a two-dimensional domain space is partitioned into page regions of the same size $p(x) \times p(y)$, the average number of the page regions that intersect with an arbitrary query region of the size $q(x) \times q(y)$ is $(\frac{q(x)}{p(x)} + 1)(\frac{q(y)}{p(y)} + 1)$.*

Proof: See the reference [7]. □

Theorem 2 *If a two-dimensional domain space is partitioned into page regions of the same size $p(x) \times p(y)$, the optimal interval ratio of a page region that minimizes the total number of page regions intersecting with n query regions of varying sizes $q_i(x) \times q_i(y)$(i = 1, ..., n) is obtained as $p(x) : p(y) = \sum_{i=1}^{n} q_i(x) : \sum_{i=1}^{n} q_i(y)$.*

Proof: Let the size of each page region in the domain space be

$$p(x) \times p(y) = B \quad (5)$$

Then, by Lemma 3, the total number of page regions $NP(p(x), p(y))$ intersecting with n query regions $q_i(x) \times q_i(y)$(i = 1, ..., n) is as follows:

$$NP(p(x), p(y)) = \sum_{i=1}^{n} (\frac{q_i(x)}{p(x)} + 1)(\frac{q_i(y)}{p(y)} + 1) \quad (6)$$

Substituting $p(y)$ with $\frac{B}{p(x)}$ in Eq. (6), we obtain

$$NP(p(x), \frac{B}{p(x)}) = \sum_{i=1}^{n} (\frac{q_i(x)}{p(x)} + 1)(\frac{q_i(y)\,p(x)}{B} + 1) = \frac{\sum_{i=1}^{n} q_i(x)\,q_i(y)}{B} + \frac{\sum_{i=1}^{n} q_i(x)}{p(x)} + \frac{\sum_{i=1}^{n} q_i(y)\,p(x)}{B} + n \quad (7)$$

[2] We assume that the total number of pages constituting a two-dimensional file is constant regardless of the interval ratio of page regions. Practically, the number of pages constituting a file depends on the number of records, the size of a page, and storage utilization. Since the interval ratio of page regions can affect storage utilization, the number of pages can vary slightly in principle. However, we ignore this variation assuming the storage utilization is constant. This assumption is well justified from the results of extensive experiments in Section 6.2.

The value of $p(x)$ that minimizes NP in Eq. (7) is obtained as $p(x) = \sqrt{\frac{\sum_{i=1}^{n} q_i(x)}{\sum_{i=1}^{n} q_i(y)}B}$. From Eq. (5), $p(y) = \sqrt{\frac{\sum_{i=1}^{n} q_i(y)}{\sum_{i=1}^{n} q_i(x)}B}$. Thus, the optimal interval ratio $p(x) : p(y)$ is $\sum_{i=1}^{n} q_i(x) : \sum_{i=1}^{n} q_i(y)$. □

3.3 Optimal Interval Ratio of Page Regions for Multiple Queries When Records are Nonuniformly Distributed

When records are nonuniformly distributed, the density of a region varies depending on the position in the domain space. So does the size of a page region since a page region with a higher density should be smaller than one with a lower density. For example, Figure 2 shows that the page region A having a higher record density is smaller in size than the page region B having a lower record density. Since records are not uniformly distributed in a real database environment, we need to extend Theorem 2 for the general case of nonuniform distributions.

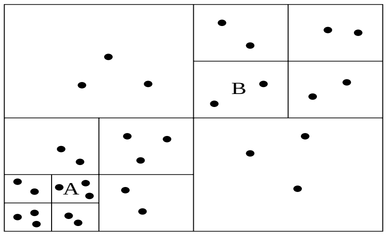


Figure 2: The size of page regions with a nonuniform distribution of records.

Normalization of Query Regions

If records are not uniformly distributed, we cannot obtain the optimal interval ratio of page regions by Theorem 2 since the number of page regions intersecting with a query region is affected not only by the size of the query region, but also by the density at the position of the query region. Thus, in order to properly compare the size of the query region with that of page regions, each query region must be weighted with record density. We define a *normalized query region* as the one weighted with the record density at the position where the query region is given.

A query region $q(x) \times q(y)$ containing nr records is transformed into the normalized query region $q(x)\sqrt{d} \times q(y)\sqrt{d}$, where the record density d is given by $\frac{nr}{q(x) \times q(y)}$. That is, the interval ratio of the normalized query region is maintained as the original interval ratio $q(x)\sqrt{d} : q(y)\sqrt{d} = q(x) : q(y)$, and only the size of the query region is multiplied by the record density d. Theorem 3 forms the underlying principle that determines the optimal interval ratio of page regions for query regions of various interval ratios, where records are nonuniformly distributed in the two-dimensional domain space.

Theorem 3 *Let n query regions of varying sizes $q_i(x) \times q_i(y) (i = 1, ..., n)$, where the record density of each query region is d_i, be given in the two-dimensional domain space partitioned into page regions of varying sizes. Then, the optimal interval ratio of page regions that minimize the total number of the page regions intersecting with the query regions is given by $p(x) : p(y) = \sum_{i=1}^{n} q_i(x)\sqrt{d_i} : \sum_{i=1}^{n} q_i(y)\sqrt{d_i}$. We assume that the interval ratio of all page regions is the same and includes the same number of records*[3].

Proof: For all page regions of varying sizes in the two-dimensional domain space, all the normalized page regions are of the same size, which is equivalent to the number of records K contained in a page region. Thus, we represent the normalized page region as

$$p'(x) \times p'(y) = K \qquad (8)$$

Since normalized page regions have the same size and interval ratio, we use Lemma 3 to derive the total number of normalized page regions $NP(p'(x), p'(y))$ intersecting with n normalized query regions $q_i(x)\sqrt{d_i} \times q_i(y)\sqrt{d_i}$(i = 1, ..., n) as follows:

$$\begin{aligned} & NP(p'(x), p'(y)) \\ = & \sum_{i=1}^{n} (\frac{q_i(x)\sqrt{d_i}}{p'(x)} + 1)(\frac{q_i(y)\sqrt{d_i}}{p'(y)} + 1) \qquad (9) \end{aligned}$$

As the same method in Theorem 2, the optimal interval ratio of normalized page regions $p'(x) : p'(y)$ becomes $\sum_{i=1}^{n} q_i(x)\sqrt{d_i} : \sum_{i=1}^{n} q_i(y)\sqrt{d_i}$. Since the interval ratio of the original page regions is the same as that of the normalized page regions, the optimal interval ratio of page regions becomes $p(x) : p(y) = p'(x) : p'(y) = \sum_{i=1}^{n} q_i(x)\sqrt{d_i} : \sum_{i=1}^{n} q_i(y)\sqrt{d_i}$. □

4 Optimal Region Splitting Strategy

In this section, we present a region splitting strategy that makes the interval ratio of the page regions in two-dimensional file organizations as close as possible to the optimal one determined by the method described in Theorem 3.

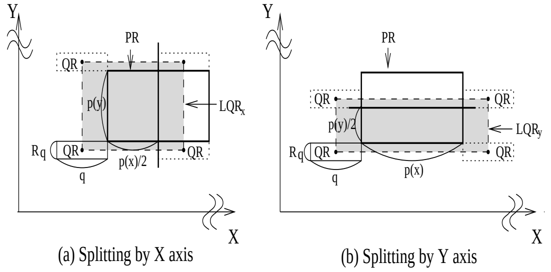
In the multidimensional file organizations using the region-oriented splitting strategy, when a page overflows due to insertion of a new record, the page region to which the record belongs splits into two equal-sized subregions and the records are distributed into two pages that are allocated for the new subregions. By choosing the splitting axis that makes the interval ratio of subregions closer to the optimal interval ratio, we can have the interval ratio of all page regions in the domain space approach the optimal ratio as regions split repeatedly.

Let the optimal interval ratio be $1 : R$. In Figure 3, LQR_x and LQR_y represent the shadowed region where

[3] This assumption is reasonable since the number of records in a page is related to the *blocking factor*[2] of disk pages.

the center point of a query region QR of $q \times Rq$ (having the optimal interval ratio $1 : R$) could intersect with one of the subregions split from a page region PR, $p(x) \times p(y)$. Figure 3(a) shows LQR_x for choosing the X axis, and Figure 3(b) LQR_y for choosing the Y axis. The sizes of LQR_x and LQR_y are obtained as follows:

$$SIZE(LQR_x) = (p(x)/2 + q)(p(y) + Rq) \quad (10)$$
$$SIZE(LQR_y) = (p(x) + q)(p(y)/2 + Rq) \quad (11)$$



(a) Splitting by X axis

(b) Splitting by Y axis

Figure 3: The region where the center point of the query region can intersect with one of subregions split from a page region.

We have shown, in Lemma 1 of Section 3, that the size of LQR is minimized when the interval ratio of a page region is the same as that of the query region. Thus, the interval ratio of the split subregion gets closer to the optimal interval ratio QR as the size of LQR gets smaller. Thus, we choose the axis that makes LQR smaller, to make the split subregion closer to the optimal interval ratio.

5 Design Algorithm

In this section, we present a general algorithm for the design of optimal configuration of multidimensional file organizations containing N organizing attributes. Algorithm 1 extends the concepts developed for the two-dimensional file organizations in Sections 3 and 4.

Algorithm 1 *A physical database design algorithm for multidimensional file organizations*

- **Input**
  1. Query regions: $q_i(1) \times q_i(2) \times \ldots \times q_i(N)$ with frequencies $f_i \;\; (i = 1, \ldots, n)$
  2. The number of records in each query region: $nr_i \;\; (i = 1, \ldots, n)$
- **Output**
  The optimal region splitting strategy that minimizes the query processing cost
- **Algorithm**
  1. Normalization of each query region $(i = 1, \ldots, n)$
     (1) Calculate record density d_i for each query region
     $$d_i = \frac{nr_i}{q_i(1) \times q_i(2) \times \ldots \times q_i(N)}$$
     (2) Calculate the interval size $q'_i(j)$ for each axis of the normalized query region $(j = 1, \ldots, N)$
     $$q'_i(j) = q_i(j) \times d_i^{\frac{1}{N}}$$
  2. Determination of the optimal interval ratio of page regions
     $$\begin{aligned} & a(1) : a(2) : \ldots : a(N) \\ = & \sum_{i=1}^{n} q'_i(1) f_i : \sum_{i=1}^{n} q'_i(2) f_i : \ldots : \sum_{i=1}^{n} q'_i(N) f_i \end{aligned}$$
  3. Construction of the optimal configuration for the multidimensional file
     (Suppose the page region that requires splitting is $p(1) \times p(2) \times \ldots \times p(j) \times \ldots \times p(N)$)

  Choose the axis j corresponding to
  $$\min_{1 \le j \le N} \left\{ \prod_{k=1}^{N} (p(k)\alpha(j) + a(k)) \right\},$$
  $$where \;\; \alpha(j) = \begin{cases} 1/2 & \text{if } j = k \\ 1 & \text{otherwise} \end{cases}$$
  □

Algorithm 1 consists of three steps. The first step is normalization of each query region in the given query pattern.

The next step is determination of the optimal interval ratio of page regions. This optimal interval ratio is obtained by summing up the normalized intervals of the query regions for each axis.

The last step is construction of the optimal configuration for the multidimensional file. We assume a query region $a(1) \times a(2) \times \ldots \times a(N)$ whose interval ratio is the same as the optimal interval ratio. Then, for the page region $p(1) \times p(2) \times \ldots \times p(N)$ to be split, we first calculate the size of the LQR_js, where LQR_j is the one when the j-th axis is chosen for the splitting axis. We choose the axis that makes LQR the smallest, making the interval ratio of the resulting subregion close to the optimal interval ratio. The size of LQR is $(p(1)/2 + a(1))(p(2) + a(2)) \ldots (p(N) + a(N))$ if we choose the first axis as the splitting axis, $(p(1) + a(1))(p(2)/2 + a(2)) \ldots (p(N) + a(N))$ if we choose the second one, and $(p(1)+a(1))(p(2)+a(2)) \ldots (p(N)/2+a(N))$ if we choose the Nth one. This can be easily understood by extending the two-dimensional case in Figure 3 of Section 4 to the N-dimensional case.

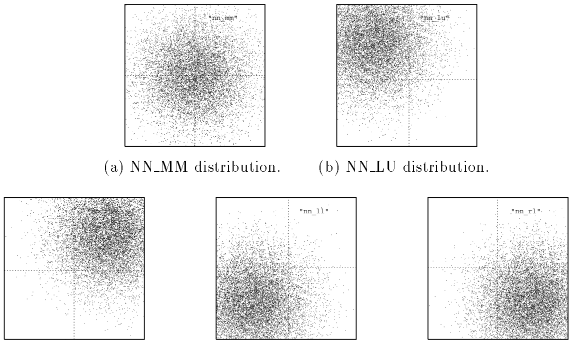
6 Performance Evaluation

In this section, we present the experimental results of the performance evaluation of query processing using the proposed region splitting strategy. We use the MLGF[14, 15, 16] as the multidimensional file organization. The purpose of the experiments is to prove the correctness and the usefulness of the proposed technique. In performing the experiments we vary factors such as the interval ratios and sizes of query regions, the interval ratios of page regions, and the distribution of records. In Section 6.1, we describe the environment for the experiments. In Section 6.2, we present and analyze the results.

6.1 Environment for Experiments

Two MLGFs containing 100,000 records were built: one with two organizing attributes and the other with three.

The record distributions used for the two-dimensional MLGF are (1) uniform distribution, (2) nonuniform distribution, and (3) real data distribution. Uniform distribution of data, denoted as UU, has two attributes and the values of each attribute are uniformly distributed over the range of $[-2^{31}, 2^{31}-1]$. For nonuniform distribution, five different data sets were generated to check the effect of record distribution on the optimal interval ratio of page regions. The values of each attribute take normal distribution $N(\mu, \sigma^2)$ with a standard deviation $\sigma = 2^{31} \times 2/5$ over the range of $[-2^{31}, 2^{31}-1]$. By controlling the mean values for the two attributes, we created five distributions, as shown in Figure 4, that have different centers of record concentrations: (a) NN_MM (meaning Normal_Normal_Middle_Middle) distribution, (b) NN_LU (meaning Normal_Normal_Left_Upper) distribution, (c) NN_RU (meaning Normal_Normal_Right_Upper) distribution, (d) NN_LL (meaning Normal_Normal_Left_Lower) distribution, and (e) NN_RL (meaning Normal_Normal_Right_Lower) distribution. For real data distribution, a topographical map data from around the Seoul area was used as shown in Figure 5. The data used for the three-dimensional MLGF have a nonuniform distribution denoted as NNN, where the values of each attribute take the normal distribution $N(0, \sigma^2)$ with $\sigma = 2^{31} \times 2/5$ over the range of $[-2^{31}, 2^{31}-1]$.



(a) NN_MM distribution. (b) NN_LU distribution.

(c) NN_RU distribution. (d) NN_LL distribution. (e) NN_RL distribution.

Figure 4: Five two-dimensional nonuniform distributions used in the experiments.

To generate the query patterns for the two-dimensional case, we used interval ratios 1:1, 1:4, 1:8, 1:16, 1:32, 1:64, 1:128, and 1:256. For each of these intervals, we generated query regions with three different sizes: (1) large (1/200 of the domain space) ones L1, L2, L4, L8, L16, L32, L64, L128, and L256; (2) medium (1/2,000 of the domain space) ones M1, M2, M4, M8, M16, M32, M64, M128, and M256; and (3) small (1/20,000 of the domain space) ones S1, S2, S4, S8, S16, S32, S64, S128, and S256. For the three-dimensional case, we used small (1/20,000 of the domain space) query regions S1_1_1, S1_2_4, S1_4_16, S1_8_64, and S1_16_256 that had interval ratios of 1:1:1, 1:2:4, 1:4:16, 1:8:64, and 1:16:256, respectively.

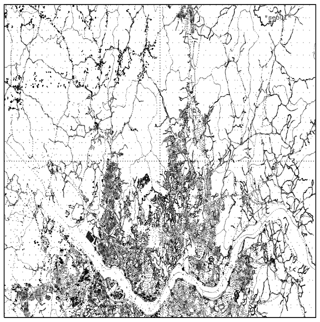


Figure 5: Topographical map data from around the Seoul area used for real data distribution.

6.2 Experimental Results

In the first experiment, we generated two-dimensional MLGFs with UU distribution with various target interval ratios of page regions. We then measured the average number of page accesses needed for processing queries with specific interval ratios. The purpose of this experiment was to confirm that the average number of pages accessed by a query region given at an arbitrary location in the domain space is minimized when the target interval ratio of page regions equals that of the query region (Theorem 1 in Section 3.1). Target interval ratios of page regions used for generating the MLGFs were 1:.25, 1:.5, 1:1, 1:2, 1:4, 1:8, 1:16, 1:32, 1:64, 1:128, 1:256, 1:512, and 1:1024. Query region types used were M1, M4, M16, M64, and M256 as defined in Section 6.1. We generated 1000 query regions for each type uniformly in the domain space and measured the average number of pages accessed for processing the queries. The results are plotted in Figure 6, where the horizontal axis represents the target interval ratios of page regions constituting the MLGF, and the vertical axis the average number of pages accessed for each type of query regions. As shown in Figure 6, in all of the cases, the best result occurs when the target interval ratio of page regions is equal to the interval ratio of the query regions. This result confirms the correctness of Theorem 1.

The second experiment was identical to the first one except that it used NN_MM distribution. The purpose of this experiment was to confirm that Theorem 1 is correct even when records are nonuniformly distributed in the domain space. As the same result

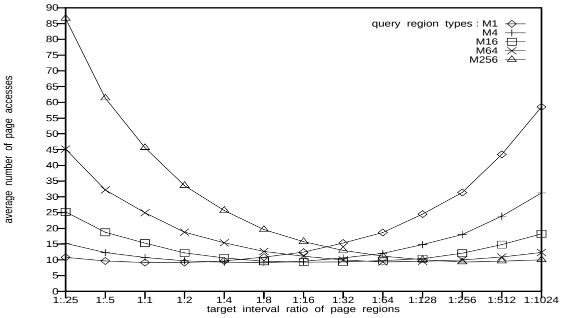


Figure 6: The cost of processing queries of specific types against data having the UU (uniform) distribution.

of the first experiment, the best result occurred when the target interval ratio of page regions was equal to the interval ratio of the query regions. Using other distributions NN_LU, NN_RU, NN_LL, NN_RL produced similar results. This result confirms the correctness of Theorem 1 for nonuniform distributions.

In the third experiment, we generated three-dimensional MLGFs with the NNN distribution with various target interval ratios of page regions and measured the average number of page accesses needed for processing queries with specific interval ratios. The purpose of this experiment was to confirm that Theorem 1 works for three-dimensional MLGFs. We used five target interval ratios of page regions for generating the MLGF, 1:1:1, 1:2:4, 1:4:16, 1:8:64, and 1:16:256. We used five types of query regions S1_1_1, S1_2_4, S1_4_16, S1_8_64, and S1_16_256. We generated 1000 query regions for each type uniformly distributed in the domain space. We then measured the average number of pages accessed for processing queries as in the second experiment. As shown in Figure 7, in all cases, the best result occurred when the target interval ratio of page regions constituting the MLGF was equal to the interval ratio of the query regions. This result shows that the three-dimensional MLGF has the same tendency as the two-dimensional one. Thus, from now on, we concentrate only on the results of two-dimensional cases.

In the fourth experiment, we generated two-dimensional MLGFs with the UU distribution with various target interval ratios of page regions as in the first experiment. We then measured the average number of page accesses needed for processing queries with mixed interval ratios. The purpose of this experiment was to confirm that Theorem 2, in Section 3.2, is correct when the records are uniformly distributed. The query pattern used included large query regions L1, L8, L64, medium ones M2, M16, M128, and small ones S4, S32, S256. We generated ten query regions for each type of large query regions, 100 for each type of medium ones, and 1000 for each type of small ones. All these query regions are uniformly distributed in the domain space. For this query pattern, using Theorem 2, we calculated the optimal interval ratio of page regions as 1:24.7. As shown in Figure 8, the best result in our experiment was obtained when the target interval ratio of page regions constituting the MLGF was 1:32, which is closest to the optimal interval ratio. This result justifies Theorem 2 for uniform distribution of records.

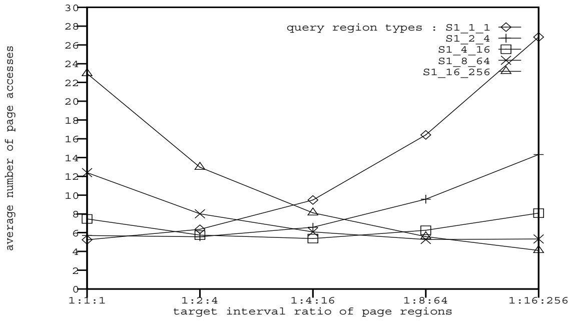


Figure 7: The cost of processing queries of specific types against data having the NNN (three-dimensional nonuniform) distribution.

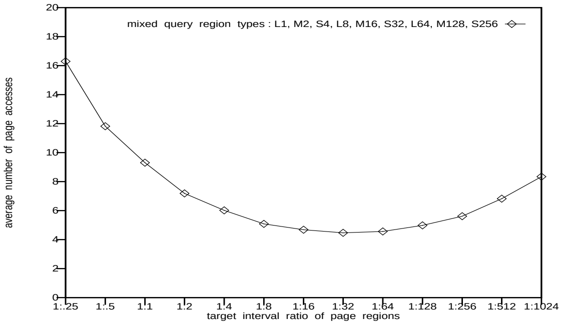


Figure 8: The cost of processing a query pattern consisting of mixed query types against data having the UU (uniform) distribution.

In the fifth experiment, we generated two-dimensional MLGFs with five nonuniform distributions NN_LU, NN_RU, NN_MM, NN_LL, and NN_RL with various target interval ratios of page regions. We then measured the average number of page accesses needed for processing queries with mixed interval ratios. The purpose of this experiment was to confirm the correctness of Theorem 3, in Section 3.3, which involves normalization of query regions for nonuniform distribution of records. Figure 9 shows the types and distribution of query regions constituting the query pattern used in this experiment. The query pattern included small query regions S2, S8, S32, S128, and S512. We generated 200 query regions for each type, where query regions for each type were allocated different portions of the domain space as shown in Figure 9. From this query pattern, using Theorem 3, we calculated the optimal interval ratios of page regions through normalization of query regions for each of distributions NN_LU, NN_RU, NN_MM, NN_LL, and NN_RL. The results are 1:3.01, 1:9.64, 1:35.03, 1:107.44, and 1:360.23, respectively. As shown in Figure 10, the best results were obtained when the target interval ratios of the MLGF were closest to the optimal ones calculated. This result justifies Theorem 3.

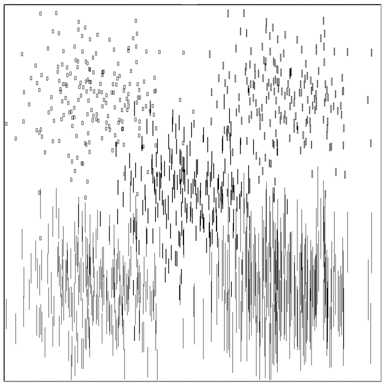


Figure 9: A query pattern consisting of query region types of different interval ratios where each type is allocated a different portion of the domain space.

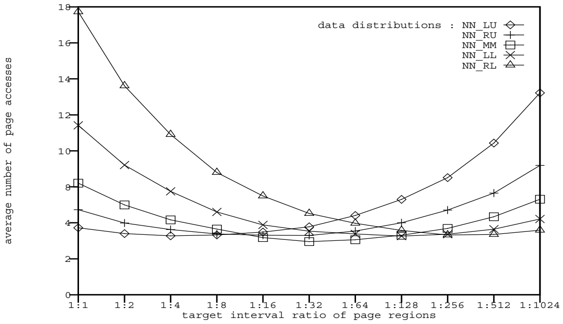


Figure 10: The cost of processing a query pattern in Figure 9 consisting of mixed query types against data having nonuniform distributions of data.

Finally, in the sixth experiment, we compared the performance of the MLGFs constructed using the proposed region splitting strategy with that of the one (which we define the *canonical MLGF*) constructed using the cyclic splitting strategy (i.e., target interval ratio = 1:1). First, we generated eleven two-dimensional MLGFs with real data shown in Figure 5 with target interval ratios of page regions 1:1, 1:2, 1:4, 1:8, 1:16, 1:32, 1:64, 1:128, 1:216, 1:512, and 1:1024. For each MLGF, we processed a query pattern consisting of 1000 medium query regions having the same interval ratio as the target interval ratio. The results were compared with those of processing the same query patterns in the canonical MLGF. Figure 11 shows the results, where the horizontal axis represents the interval ratios of page and query regions, and the vertical axis the *efficiency ratios*, which we define as the ratio of the costs of processing the same query pattern using the MLGF over those using the canonical MLGF.

As shown in Figure 11, the more the interval ratio of query regions deviates from 1:1, the higher the efficiency ratio we obtain. We observe that the efficiency ratio is as much as 7.5 for the interval ratio of 1:1024. The performance should be further enhanced for the query regions having higher interval ratios. This result proves the effectiveness of the proposed region splitting strategy.

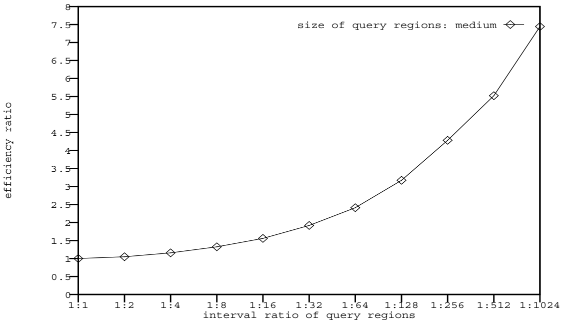


Figure 11: The efficiency ratio of the MLGF constructed by using the proposed physical database design technique for various interval ratios of query regions.

7 Conclusions

We have proposed a region splitting strategy for optimal construction of multidimensional file organizations. We have first proved analytically that the number of page accesses is minimized regardless of record distributions when the interval ratio of page regions is equal to the interval ratio of a single given query region. And then, we have proved, for multiple N-dimensional query regions $q_i(1) \times q_i(2) \times \cdots \times q_i(j) \times \cdots \times q_i(N) (i = 1, \cdots, n)$, we can obtain the optimal interval ratio of page regions as follows. When the records are uniformly distributed in the domain space, the optimal interval ratio of the page regions is the ratio of the summations of the interval sizes of the query regions for each axis (i.e., $\sum_{i=1}^{n}(q_i(1)) : \sum_{i=1}^{n}(q_i(2)) : \cdots : \sum_{i=1}^{n}(q_i(j)) : \cdots : \sum_{i=1}^{n}(q_i(N))$, where $q_i(j)$ is the interval size for the j-th axis of the query q_i). When the records are nonuniformly distributed, the optimal interval ratio of the page regions is the ratio of the summations of the interval sizes of the normalized query regions for each axis (i.e., $\sum_{i=1}^{n}(q'_i(1)) : \sum_{i=1}^{n}(q'_i(2)) : \cdots : \sum_{i=1}^{n}(q'_i(j)) : \cdots : \sum_{i=1}^{n}(q'_i(N))$, where $q'_i(j)$ is the interval size for the j-th axis of the normalized query q'_i).

We also have proposed a region splitting strategy that makes the interval ratio of the page regions close to the optimal interval ratio theoretically obtained. We have performed extensive experiments to validate the analysis using the MLGF. The experimental results indicate that the proposed method builds an optimal MLGF for any query pattern regardless of record distributions. As shown in Figure 11, the performance is enhanced by as much as 7.5 times over the conventional method using the cyclic splitting strategy for the query regions having the interval ratio of 1:1024. The performance should be further enhanced for the query regions having higher interval ratios. This proves that the proposed method is very useful for practical applications.

The performance enhancement provided by the proposed region splitting strategy results from the fact that query conditions typically impose different intervals for different attributes. The more the shapes of the query regions deviate from regular hyperrectangles, the more useful the proposed method becomes.

Acknowledgment

This work was partially supported by the Development of a Spatial Object Storage System project sponsored by the Korean Ministry of Science and Technology (MOST) and the Content-based Multimedia Information Retrieval project sponsored by the Korea Science and Engineering Foundation (KOSEF) through the Center for Artificial Intelligence Research (CAIR).

References

[1] Bentley, J. L., "Multidimensional Binary Search Trees Used for Associative Searching," *Comm. of the ACM*, Vol. 18, No. 9, pp. 509–517, 1975.

[2] Elmasri, R. and Navathe, S. B., *Fundamentals of Database Systems,* Benjamin/Cummings Publishing Co., Redwood City, California, Second Ed., 1994.

[3] Finkelstein, S. et al., "Physical Database Design for Relational Databases," *ACM Trans. on Database Systems*, Vol. 13, No. 1, pp. 91–128, Mar. 1988.

[4] Freeston, M., "The BANG File: A New Kind of Grid File," In *Proc. Intl. Conf. on Management of Data*, ACM SIGMOD, pp. 260–269, San Francisco, 1987.

[5] Henrich, A. et al., "The LSD tree: Spatial Access to Multidimensional Point and Non Point Objects," In *Proc. Intl. Conf. on Very Large Data Bases*, pp. 45–53, Aug. 1989.

[6] Kumar, A., "G-tree: A New Data Structure for Organizing Multidimensional Data," *IEEE Trans. on Knowledge and Data Engineering*, Vol. 6, No. 2, pp. 341–347, Apr. 1994.

[7] Lee, J. H. et al., A Physical Database Design Method for Multidimensional File Organizations, Technical Report CS/TR-96-104, Computer Science Department, KAIST, Nov. 1996.

[8] Lomet, D. and Salzberg, B., "The hB-tree: A Multiattribute Indexing Method with Good Guaranteed Performance," *ACM Trans. on Database Systems*, Vol. 15, No. 4, pp. 625–658, Dec. 1990.

[9] Nakamura, Y. et al., "A Balanced Hierarchical Data Structure for Multidimensional Data with Highly Efficient Dynamic Characteristics," *IEEE Trans. on Knowledge and Data Engineering*, Vol. 5, No. 4, pp. 682–694, Aug. 1993.

[10] Nievergelt, J. et al., "The Grid File: An Adaptable, Symmetric Multikey File Structure," *ACM Trans. on Database Systems*, Vol. 9, No. 1, pp. 38–71, Mar. 1984.

[11] Robinson, J. T., "The K-D-B-Tree: A Search Structure for Large Multidimensional Dynamic Indexes," In *Proc. Intl. Conf. on Management of Data*, ACM SIGMOD, pp. 10–18, Ann Arbor, Michigan, Apr. 1981.

[12] Seeger, B. and Krieger, H. P., "The Buddy-Tree: An Efficient and Robust Access Method for Spatial Data Base Systems," In *Proc. Intl. Conf. on Very Large Data Bases*, pp. 590–601, Brisbane, Australia, Aug. 1990.

[13] Whang, K. Y. et al., "Separability–An Approach to Physical Database Design," *IEEE Trans. on Computers*, Vol. c-33, No. 3, pp. 209–222, Mar. 1984.

[14] Whang, K. Y. and Krishnamurthy, R., Multilevel Grid Files, IBM Research Report RC 11516, IBM Thomas J. Watson Research Center, Nov. 1985.

[15] Whang, K. Y. and Krishnamurthy, R., "The Multilevel Grid File – A Dynamic Hierarchical Multidimensional File Structure," In *Proc. Intl. Conf. on Database Systems for Advanced Applications*, pp. 449–459, Tokyo, Apr. 1991.

[16] Whang, K.Y., Kim, S. W., and Wiederhold, G., "Dynamic Maintenance of Data Distribution for Selectivity Estimation," *The VLDB Journal*, Vol. 3, No. 1, pp. 29–51, Jan. 1994.

[17] Yu, C. T. et al., "Adaptive Record Clustering," *ACM Trans. on Database Systems*, Vol. 10, No. 2, pp. 180–204, June 1985.