Analyzing the range query performance of two partitioning methods in high-dimensional space

Tae-Wan Kim\textsuperscript{1}

Hak-Cheol Kim\textsuperscript{2}  Ki-Joune Li\textsuperscript{2}

\textsuperscript{1}Research Institute of Computer Information and Communication

\textsuperscript{2}Department of Computer Science

\{twkim\textsuperscript{1},hckim\textsuperscript{2},lik\textsuperscript{2}\}@quantos.cs.pusan.ac.kr

Pusan National University

\textbf{Abstract}

Given $d$-dimensional $N$ data items and a maximum branching factor $Bf_{\text{max}}$, \textit{packing} is partitioning $Bf_{\text{max}}$ amount of data and storing it in a disk page. The range query performance of packing is highly dependent on the methods by which data is partitioned. Thus, partitioning data is the main problem tackled in our work. We suggest two extreme cases of partitioning methods: grid-like balanced and onion-peeling-like unbalanced partitioning methods. We analyze them based on the Minkowski-sum cost model and present the expected numbers of intersecting pages given $d$-dimensional hypercubic range queries. Analysis clearly indicates the efficient method of partitioning. By experimentation, we validate our
analysis and show the superiority of the onion-peeling-like partitioning method when the dimension and the amount of data is quite large.

*Keywords*: high-dimensional data; cost model; bulking-loading; packing; access method

1 Introduction

In recent years, new database applications have been studied which are different in many aspects from conventional database applications. These are data mining, data warehousing, multimedia database systems, and others. The common characteristics of data in these new applications are its huge size and multi-attributes. Multi-attributed data objects are often transformed to multi-dimensional points called feature vectors. Similarity and range searches are one of the basic functionalities required in these applications. Subsequently, a high-dimensional index structure is required to process searches efficiently.

Packing or bulk-loading is one of the possible solutions to enhance query performance among many database techniques, and this is the area of our interest. Packing is a technique to partition $Bf_{\text{max}}$ amount of data and to store it on a fixed-size disk page. Each partitioned page corresponds to a rectangle for two-dimensional space, to a cube for three-dimensional space, and to a $d$-dimensional hypercube for $d$-dimensional space. A hypersphere could be an alternative representation, but we do not consider it here. Since
the performance of range queries on these partitioned data sets is mainly
dependent on the shape of the data sets, the method to partition data is
directly related to query performance.

For two-dimensional space, it is known that minimizing the perimeter of
bounding boxes or generating square-like bounding boxes[1, 3, 6] is query-
efficient for range queries. Thus a uniform grid-like partitioning method could
be a proper candidate. Note that NP-hardness of packing for $Bf_{max} \geq 3$ was
proved in [7]. But for high-dimensional data, metrics used for two-dimensional
space are no longer applicable [2]. In [2], they showed that a heavily unbal-
anced split strategy, such as a 9:1 proportion, results in better query perfor-
ance than a grid-like strategy. They showed the reason intuitively, but failed
to show it analytically.

In this paper, under a uniform data distribution assumption, we thor-
oughly analyze the query performance of two extreme cases of partitioning
methods: a uniform grid-like partitioning (called GRID) method and an onion-
peeling-like unbalanced partitioning (called CSP; Cyclic Sliced Partitioning)[4]
method. The main contribution of our paper is to suggest, based on analysis,
solid metrics on how to partition high-dimensional data to support efficient
hypercubic range queries.

In section 2, we analyze the Minkowski-sum cost model for high-dimensional
data suggested in[2, 5] and present observations on this model. In section 3,
we provide through analysis of both GRID and CSP. In section 4, we compare
our analysis to experiments, which show a remarkable analytical accuracy. In section 5, we conclude our paper.

2 Analysis of partitioning methods based on the Minkowski–sum cost model

We assume that \( N \) \( d \)-dimensional data is uniformly distributed in the \( d \)-dimensional unit hyper space \([0, 1]^d\). Given a blocking factor \( B_{f_{\text{max}}} \), we use a number of partitions \( P \) of \( \lceil \frac{N}{B_{f_{\text{max}}}} \rceil \). A partitioning method partitions \( B_{f_{\text{max}}} \) amount of data and generates \( P \) number of minimum bounding hypercubes (MBHs). We denote \( \bar{q} \) as a \( d \)-dimensional hypercubic range query and \( q \) as its side length.

2.1 Minkowski-sum cost model

The Minkowski-sum cost model[2, 5] is a generalization of geometric probability by taking into account the boundary effect and it estimates the expected number of intersecting pages. More precisely, the expected number \( E(M, \bar{q}) \) of page accesses upon processing a \( \bar{q} \) on a set \( M \) of MBHs is:

\[
E(M, \bar{q}) = \sum_{i=1}^{m} \prod_{j=0}^{d-1} \frac{\min(high_{i,j}, 1 - q) - \max(low_{i,j} - q, 0)}{1 - q}, \quad (1)
\]
M : the set of MBHs where \(|M| = m\),

\( \text{high}_{i,j} \) : the right-most value of an interval of the \(i^{th}\) MBH at the \(j^{th}\) dimension,

\( \text{low}_{i,j} \) : the left-most value of an interval of the \(i^{th}\) MBH at the \(j^{th}\) dimension.

Now we look closer at equation (1) and examine its behavior. Equation (1) is expressed as the sum of the expected value of each MBH. And the expected value of each MBH is the product of the expected value of one-dimensional intervals of a MBH. And the expected value of each interval is different depending on where it is positioned in intervals partitioned by \(q\). The following example shows this more specifically.

**Example 2.1.** Assume that a one-dimensional data space is equally partitioned by \(P_1(=7)\) number of MBHs. Figure 1 shows seven MBHs which are one-dimensional intervals.

![Figure 1: Intervals partitioned by q](image)

\( \text{a. } q < \frac{1}{2} \quad \text{b. } q \geq \frac{1}{2} \)

When \(q < 0.5\), there are six intervals intersecting [0, \(q\)] and [1-\(q\), 1]. By equation (1), the sum of the expected values of these intervals is \(2 \frac{1+2+3}{P_1} / (1-q)\). The expected value of each interval in [\(q\), 1-\(q\)] is the same, \((P_1-6) \frac{1}{P_1} + q) / (1-\)
When \( q \geq 0.5 \), the number of intervals contained in \([0, 1-q]\) and \([q, 1]\) is four. The sum of the expected values of these intervals is \(2^{1+\frac{2}{P_1}}/(1-q)\). The expected value of each interval in \([1-q, q]\) is the same, \((P_1-4)(1-q)/(1-q) = P_1-4\). For example, when \( q \) equals 0.3 and 0.7, the expected value is 3.08 and 5.85, respectively.

From the example 2.1, we observe that (1) the expected values of the intervals in the middle of the data space are the same, \((1/P_1 + q)/(1-q)\) or 1, and (2) the expected value of the intervals around either end of the data space is directly related to the distance to an end. As a generalization of the above observations, we have the following proposition.

**Proposition 2.1.** Suppose that a MBH and a \( \bar{q} \) with \( q \geq 0.5 \) are given. Let the \( i^{th} \) dimensional interval of the MBH be \( I_i = [low_i, high_i] \). Then its expected value \( E(I_i, q) \) is

1. \( E(I_i, q) = \frac{high_i}{1-q} \), when \( high_i < 1-q \);
2. \( E(I_i, q) = \frac{1-low_i}{1-q} \), when \( low_i > q \);
3. \( E(I_i, q) = 1 \), when \( (high_i - low_i) \geq 1-q \);
4. \( E(I_i, q) = 1 \), when \( 1-q \leq high_i \leq q \) or \( 1-q \leq low_i \leq q \).

Note that by proposition 2.1, we can rewrite the expected value of a MBH in terms of intervals and \( q \), such as \( \prod_{i=0}^{d-1} E(I_i, q) \).
2.2 GRID

A GRID partitions a data space into equal-sized MBHs, which are shown in Figure 2.a. A given query side length $q$ partitions an interval of each dimension into three parts as we explained in the previous subsection. Thus the expected values of the MBHs are different dependent on which parts of intervals the MBHs intersect. In the two-dimensional case, there are three different types of partitioned areas: those of which the side lengths are all $1 - q$, all $2q - 1$, and one $1 - q$ and one $2q - 1$ when $q \geq 0.5$. We can rewrite the volume of a $d$-dimensional hyper cube in terms of $q$, $\text{Vol}(\text{cube}, d)$, as follows:

$$\text{Vol}(\text{cube}, q) = [(2q - 1) + 2(1 - q)]^d = \sum_{i=0}^{d} C(d, i) \cdot (2q - 1)^i \cdot (2(1 - q))^{d-i}. \quad (2)$$

In equation (2), subscripts $i$ and $d - i$ denote the number of intervals of which the side lengths are $2q - 1$ and $1 - q$, respectively. Since multiplication of the two terms, $(2q - 1)^i$ and $(1 - q)^{d-i}$, constructs a hypercube whose shape is dependent on $i$ and $d - i$, its coefficient represents the number of such a
hypercube. For simplicity, we can formulate those numbers such as:

\[
COEF(d, i) = C(d, i) \cdot 2^{d-i}. \tag{3}
\]

In this subsection we assume that \( b \) is a side length of a MBH and the integer \( P_1 = \frac{1}{P} \) satisfying \( P = P_1^d \). For \( q \geq 0.5 \), we assume that \( c_l \) is \( \lfloor \frac{1-q}{P} \rfloor \) and \( s_l \) is \( P_1 - 2 \cdot c_l \). We first show the expected value of the intersections of MBHs generated by GRID upon processing a \( \bar{q} \).

**Lemma 2.1.** Among \( d+1 \) types of hypercubes induced by \( q \), we select a hypercube \( H \) which has \( i(0 \leq i \leq d) \) numbers of dimensions whose interval lengths are \( 2q - 1 \), and \( d-i \) numbers of dimensions whose interval lengths are \( 1-q \). Then, the expected value of the MBHs contained in the \( H \) with respect to \( \bar{q} \) is

\[
E(H, \bar{q}) = \begin{cases} 
(b^d \cdot \sum_{j=1}^{c_l} j)^d & \text{if } s_l = 0, c_l \neq 0, \\
(s_l^d) & \text{if } s_l \neq 0, c_l = 0, \\
(s_l^d \cdot (b^{d-i} \cdot \sum_{j=1}^{c_l} j)^{d-i}) & \text{if } s_l \neq 0, c_l \neq 0, \text{ and} \\
\text{undefined} & \text{otherwise}
\end{cases} \tag{4}
\]

**Proof.** We first show the expected value of the two-dimensional case and generalize it for the \( d \)-dimension. If \( s_l = 0, c_l \neq 0 \), then \( c_l \) is \( P_1/2 \) and the lengths of the intervals of \( H_{s_l=0,c_l\neq0} \) are all equal to \( 1 - q(= q) \), which is shown in Figure 3.a. By proposition 2.1.1 and 2.1.2, the expected value of the MBHs contained in \( H_{s_l=0,c_l\neq0} \) is:

\[
E(H, \bar{q}) = \frac{b}{1-q} \cdot \frac{b}{1-q} + \frac{b}{1-q} \cdot \frac{2b}{1-q} + \cdots + \frac{b}{1-q} \cdot \frac{c_l b}{1-q} +
\]

8
\[ 1 - q = q \]

\[ H_{s_l=0}, c_l \neq 0 \]

\[ 1 - q = q \]

\[ H_{c_l=0} \]

\[ q \]

\[ 1 - q \]

(a) \( s_l = 0 \) and \( c_l \neq 0 \)

(b) \( s_l \neq 0 \) and \( c_l = 0 \)

Figure 3: Two extreme cases of a hypercube \( H \)

\[
\begin{align*}
\frac{2b}{1-q} \cdot \frac{b}{1-q} + \frac{2b}{1-q} \cdot \frac{2b}{1-q} + \ldots + \frac{2b}{1-q} \cdot \frac{c_l b}{1-q} + \\
\ldots \\
\frac{c_l b}{1-q} \cdot \frac{b}{1-q} + \frac{c_l b}{1-q} \cdot \frac{2b}{1-q} + \ldots + \frac{c_l b}{1-q} \cdot \frac{c_l b}{1-q} \\
= \frac{b}{1-q} \cdot \frac{b}{1-q} \sum_{j=1}^{c_l} j + \frac{2b}{1-q} \cdot \frac{b}{1-q} \sum_{j=1}^{c_l} j + \ldots + \frac{c_l b}{1-q} \cdot \frac{b}{1-q} \sum_{j=1}^{c_l} j \\
= \frac{b}{1-q} \sum_{j=1}^{c_l} j \cdot \frac{b}{1-q} \sum_{j=1}^{c_l} j \\
= \left( \frac{b}{1-q} \sum_{j=1}^{c_l} j \right)^2.
\end{align*}
\]  

(5)

It is quite easy to show the \( d \)-dimensional case by induction, and thus the expected value is:

\[
E(H_{s_l=0, c_l \neq 0}, \bar{q}) = \left( \frac{b}{1-q} \sum_{j=1}^{c_l} j \right)^d.
\]  

(6)

If \( s_l \neq 0, c_l = 0 \), then the lengths of the intervals of \( H_{s_l \neq 0, c_l = 0} \) are all equal to \( 2q - 1 \). The expected value of MBHs intersecting \( H_{s_l \neq 0, c_l = 0} \) is 1 by proposition 2.1.4. So, the expected number of MBHs intersecting \( H_{s_l \neq 0, c_l = 0} \) is:

\[
E(H_{s_l \neq 0, c_l = 0}, \bar{q}) = s_l^d = P_1^d.
\]  

(7)
A case of $s_l \neq 0, c_l \neq 0$, there is a hypercube $H_{s_l, c_l \geq 1}$ which has $u$ numbers of dimensions whose interval lengths are all equal to $2q - 1$, and $d - u$ numbers of dimensions whose interval lengths are all equal to $q - 1$. If the interval length of the $i^{th}$ dimension of the $H_{s_l, c_l \geq 1}$ is $2q - 1$, the expected value of the $i^{th}$ interval of the MBHs in $H_{s_l, c_l \geq 1}$ is 1 by 2.1.4. If the interval length of $d - u$ numbers of dimensions of the $H_{s_l, c_l \geq 1}$ is $1 - q$, the expected value of those intervals can be derived by equation (6). Thus the expected value of the MBHs in this hypercube is:

$$E(H_{s_l, c_l \geq 1}, \bar{q}) = s_l^u \cdot \left(\frac{b}{1 - q} \sum_{j=1}^{c_l} j\right)^{d-u}. \quad (8)$$

Lemma 2.2. Given a set $S$ of $P^d_1$ uniform MBHs, the expected number $E_{grid}(S, \bar{q})$ of intersections for a query $\bar{q}$ with $q \geq 0.5$ is:

$$E_{grid}(S, \bar{q}) = \begin{cases} 
2^d \cdot \left(\frac{b}{1 - q} \sum_{j=1}^{c_l} j\right)^d & \text{if } s_l = 0, c_l \neq 0, \\
2^d \cdot \left(\frac{b}{1 - q} \sum_{j=1}^{c_l} j\right)^{d-i} & \text{if } s_l \neq 0, c_l \neq 0, \text{ and } i \leq u, \\
\sum_{i=0}^{d} COEF(d, i) \cdot s_l^i \cdot \left(\frac{b}{1 - q} \sum_{j=1}^{c_l} j\right)^{d-i} & \text{if } s_l \neq 0, c_l \neq 0, \text{ and } i > u, \\
\text{undefined} & \text{otherwise.} 
\end{cases} \quad (9)$$

Proof. There is only one possible type of a hypercube induced by $q$ when $s_l = 0, c_l \neq 0$ and $s_l \neq 0, c_l = 0$. When $s_l = 0, c_l \neq 0$, there are $2^d$ number of such hypercubes and thus $COEF(d, i)$ is $2^d$. When $s_l \neq 0, c_l = 0$, there is one such hypercube and thus the value of $COEF(d, i)$ is 1. When $s_l \neq 0$
and $c_l \neq 0$, there are $COEF(d, i)$ numbers of hypercubes of the same type. Since subscript $i$ denotes the number of dimensions whose interval lengths are $2q - 1$, the equation holds. 

So far we have shown the expected value when $q \geq 0.5$. Now we show it when $q < 0.5$. For $q < 0.5$ we assume that $c_l$ is $\lceil \frac{q}{2} \rceil$, $c_u$ is $\lfloor \frac{q}{2} \rfloor$, $s_l$ is $P_1 - 2 \cdot c_l$, and $s_u$ is $P_1 - 2 \cdot c_u$. Figure 4 is shown to enhance understanding each case of lemma 2.2.

**Lemma 2.3.** Given a set $S$ of $P_1^d$ uniform MBHs, the expected number $E_{grid}(S, q)$ of intersections for a query $\bar{q}$ whose side length $q < 0.5$ is:

$$E_{grid}(S, q) = \begin{cases} 
\sum_{i=0}^{d} COEF(d, i) \cdot 1 \cdot \left( \frac{b}{1-q} \sum_{j=1}^{c_l} j \right)^{d-i} & \text{if } s_u < 0, \\
2^d \cdot \left( \frac{b}{1-q} \sum_{j=1}^{c_u} j \right)^d & \text{if } s_u = 0, \\
\sum_{i=0}^{d} COEF(d, i) \cdot \left( \frac{b + q}{1-q} s_u \right)^i \cdot \left( \frac{b}{1-q} \sum_{j=1}^{c_u} j \right)^{d-i} & \text{if } s_u > 0, \text{ and} \\
\text{undefined} & \text{otherwise.}
\end{cases}$$

(10)
Proof. The proof of lemma 2.3 is similar to those of lemma 2.1 and lemma 2.2.

2.3 CSP

In this subsection, we analyze an unbalanced partitioning method called CSP (Cyclic Sliced Partition), which partitions the data space as we peel an onion. Two-dimensional cases are shown in Figure 2.b. At each partition, CSP partitions $d$-dimensional data by a chosen one-dimensional value and the other $(d-1)$ dimensions are not considered to be split axes. The CSP method is described more precisely below.

- CSP: For each dimension $d_i$, retrieve the smallest and the largest $B_{f_{\text{max}}}$ data from an input data set $D$. Let the interval of the retrieved set of smaller values be $[\text{low}_{i,\text{left}}, \text{high}_{i,\text{left}}]$ and that of the larger values be $[\text{low}_{i,\text{right}}, \text{high}_{i,\text{right}}]$. Remember $\text{high}_{i,\text{left}}$ and $1-\text{low}_{i,\text{right}}$. Among these distance values and associated dimensions, we select the dimension whose distance is the minimum and split a set $D$ into two subsets $D_1$ and $D_2$ where $|D_1| = B_{f_{\text{max}}}$ and $D_1 = \{e_1 \mid (e_1 \leq e_2, \forall e_2 \in D_2) \lor (e_1 \geq e_2, \forall e_2 \in D_2)\}$ by a chosen dimension and a direction. Generate a node encompassing a set $D_1$. Let $D_2$ be $D$ and partition it recursively until we have $P$ MBHs.
In the following lemmas, we present the assigned length of CSP at each partition and present its expected value in terms of the assigned length.

**Lemma 2.4.** Suppose that a $d$-dimensional unit hypercube $[0, 1]^d$ is partitioned into $P$ MBHs by CSP. Let $S_{CSP}(u, v)$ be an assigned length of an edge of the $v^{th}$ dimension at the $u^{th}$ partition of a MBH, and let $L_{CSP}(u, v)$ be the length of the data space remaining after the $v^{th}$ dimension at the $u^{th}$ partition. Then

\[
S_{CSP}(1, v) = \frac{1}{P - v + 1} \quad (11)
\]

\[
S_{CSP}(u, v) = L_{CSP}(u - 1, v) \cdot \frac{1}{k} \quad (12)
\]

\[
L_{CSP}(u, v) = L_{CSP}(u - 1, v) \cdot \frac{k - 1}{k} \quad (13)
\]

where $k = P - d \cdot (u - 1) - (v - 1), 1 \leq v \leq d$, and $2 \leq u \leq \lfloor \frac{P}{d} \rfloor$.

**Proof.** When $u = 1$, it is trivial. Suppose that a $d$-dimensional hypercube whose length of each edge is $(w_1, w_2, \ldots, w_d)$ at the end of the $t^{th}$ partition. That is, $L_{CSP}(t, i) = w_i$. Since the volume of a data space is decreased by $1/P$ at each partition, it is

\[
\prod_{i=1}^{d} w_i = \frac{k_1 - 1}{P} \quad (a)
\]

where $k_1$ is $P - d \cdot (t - 1) - (d - 1)$. At the $(k_1 + 1)^{th}$ partition, the volume of a partitioned MBH is $1/P$, and thus the following equality holds.

\[
S_{CSP}(t + 1, 1) \cdot w_2 \ldots w_d = \frac{1}{P} \quad (b)
\]
From equations (a) and (b), we have

\[ S_{CSP}(t + 1, 1) = \frac{w_1}{k_1 - 1}. \]  \hspace{1cm} (c)

Since \( k_1 - 1 \) is \( P - d \cdot t \) and \( k \) for \( (t + 1, 1) \) is \( P - d \cdot t \), we can rewrite equation (c) as

\[ S_{CSP}(t + 1, 1) = \frac{w_1}{k} = \frac{L_{CSP}(t, 1)}{k}. \]  \hspace{1cm} (d)

Thus after the \((t + 1)\text{th}\) partition at the 1\text{st} dimension, the length remaining is

\[
L_{CSP}(t + 1, 1) = L_{CSP}(t, 1) - S_{CSP}(t + 1, 1) = L_{CSP}(t, 1) \frac{k - 1}{k}. \]  \hspace{1cm} (e)

We can prove an arbitrary dimension at the \((t + 1)\text{th}\) partition similarly.  \( \square \)

To introduce the expected value of CSP, we define a cumulative length or a boundary distance function \( CL \) as follows.

**Definition 1 (Cumulative length or boundary distance, CL).**

\[
CL_{left}(u, v) = \sum_{i=1}^{u} S_{CSP}(2i - 1, v). \]  \hspace{1cm} (14)

\[
CL_{right}(u, v) = \sum_{i=1}^{u} S_{CSP}(2i, v). \]  \hspace{1cm} (15)

By definition 1, \( CL_{left} \) and \( CL_{right} \) denote a distance from the lowest boundary (that is, 0) and the highest boundary (that is, 1) of the unit data space, respectively, as shown in Figure 5.

**Lemma 2.5.** Suppose that a \( d \)-dimensional MBH set \( M \) partitioned by CSP and a \( \bar{q} \) of side length \( q \geq 0.5 \) are given. Let \( m_1, m_2, v_1 \) and \( v_2 \) be the largest...
Figure 5: $\text{CL}_{\text{left}}(m_1, v_1)$ and $\text{SCSP}(2 \cdot m_1 - 1, v_1)$ of CSP

Integers satisfying $\text{CL}_{\text{left}}(m_1, v_1) \leq 1 - q$ and $\text{CL}_{\text{right}}(m_2, v_2) \leq 1 - q$. Then, the expected value $E_{\text{CSP}}(M, \bar{q})$ of CSP when $q \geq 0.5$ is

$$E_{\text{CSP}}(M, \bar{q}) = \frac{\sum_{i=1}^{m_1-1} \sum_{j=1}^{d} \text{CL}_{\text{left}}(i, j)}{1 - q} + \frac{\sum_{j=1}^{v_1} \text{CL}_{\text{left}}(m_1, j)}{1 - q} + \frac{\sum_{i=1}^{m_2-1} \sum_{j=1}^{d} \text{CL}_{\text{right}}(i, j)}{1 - q} + \frac{\sum_{j=1}^{v_2} \text{CL}_{\text{right}}(m_2, j)}{1 - q} + (P - x)$$  \hspace{1cm} (16)

where $x = (m_1 + m_2) \cdot d + v_1 + v_2$ and $P = |M|$.

Proof. Instead of a formal proof, we briefly sketch a proof using Figure 5. Suppose that the $v_1$th dimensional edge of $\text{MBH}_i$ is one of $\text{SCSP}(2m - 1, v_1)(m \leq m_1)$ as in Figure 5. Since it is contained in an interval $[0, 1-q]$, its expected value is $\text{C}_{\text{left}}(m, v_1)/(1 - q)$ by proposition 2.1. Since $\text{SCSP}(u, v)$ is a monotonically increasing function, $\text{C}_{\text{left}}(m, v_1)$ is less than $1 - q$ and the longest distance
from 0 among all $C_{left}(m, v)$ where $mt < m$ and among all $C_{left}(m, vt)$ where $vt \leq v_1$. It also implies $C_{right}(mt, v_1) < C_{left}(m, v_1)$ for all $mt < m$. Thus, the length of the edges of $MBH_i$ except the $v_1^{th}$ dimension until the $m^{th}$ partition, is larger than $2q - 1$. The expected value of these edges is 1 by proposition 2.1. After the $(m_1, v_1)^{th}$ and the $(m_2, v_2)^{th}$ partition, the edges generated by CSP intersect or are contained in an interval $[1 - q, q]$. The expected value of these edges is 1 by proposition 2.1. Thus, the expected value of all $(P - x)$ number of MBHs to be generated is 1.

Note that the expected value of CSP is the distance to one of the boundaries and the number of MBHs remaining. When $q < 0.5$, we approximate the expected value by replacing $(P - x)$ in equation (16) with $\sum_{i=x+1}^{P} (f(i) + q) / (1 - q)$ where $f(x) = S_{CSP}(u, v)$, $u = (x \text{ div } d) + 1$, and $v = (x \text{ mod } d) + 1$.

## 3 Validation of our analysis and observations

To validate our analysis of GRID and CSP, we generate $P$ MBHs which tile $[0, 1]^d$ data space and perform range queries on them to compare analyses. Figure 6 shows the results of the comparisons. Letters -E and -A mean experiment and analysis, respectively. The analysis of GRID shows a less than 1% relative error for all ranges of queries while that of CSP shows a less than 1% error when a query side length is larger than 0.3 due to approximation.

For practical purposes, two assumptions must be taken into account. First,
the number of partitions $P$ is dependent on the amount of data $N$ and the
size of data page $|B|$ rather than dimensionality. Thus, in our experiments, we
assume that $N$ is $10^6$ and $10^9$ and $|B|$ is 4 KB. Based on this assumption we de-
determine that $P = (\lceil (N/B_{f_{max}})^{1/d} \rceil)^d$ where $B_{f_{max}} = \lceil |B|/(\text{sizeof(float)}\cdot d) \rceil$.

Second, a query side length should be determined by a function of selectivity $s$
which is (the amount of data retrieved by a query)/$N$, since a fixed query side
length has different meanings at different dimensions. That is, a query side
length 0.1 has different meanings when $d = 2$ and $d = 32$. In our experiment,
we set $q$ as $s^{1/d}$ where $10^{-12} \leq s \leq 10^{-1}$. We summarize the values of the
parameters used in the experiments in Table 1.

Figure 7.a and Figure 7.b show the analytical results of GRID and CSP
when $d$ varies from 4 to 32, respectively. These figures show that GRID is very
sensitive to the increase of dimension and selectivity, while CSP shows a steady
increase of access ratio irrespective of their increase. This is due to the steady
Table 1: The values of the parameters used in experiments

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$N = 10^6$</td>
<td>$N = 10^9$</td>
<td>$s = 10^{-12}$</td>
</tr>
<tr>
<td>4</td>
<td>$8^4$</td>
<td>$45^4$</td>
<td>0.001</td>
</tr>
<tr>
<td>8</td>
<td>$4^8$</td>
<td>$8^8$</td>
<td>0.032</td>
</tr>
<tr>
<td>16</td>
<td>$2^{16}$</td>
<td>$3^{16}$</td>
<td>0.178</td>
</tr>
<tr>
<td>32</td>
<td>$2^{32}$</td>
<td>$2^{32}$</td>
<td>0.422</td>
</tr>
</tbody>
</table>

increase of $S_{CSP}$ and thus $CL$. For low-dimensional space, GRID shows a better performance than CSP since a smaller query has a higher probability to be included in a tiled MBH. As a dimension grows, GRID shows a 100% access ratio, even a very low selectivity; for example, $s$ is $10^{-9}$ when $d = 32$. This can be explained by the previously mentioned two practical assumptions. First, when a dimension is very high, the number of partitions at each dimension cannot exceed two. Second, when a dimension is very high, a query side length is larger than 0.5, even selectivity is very low, which results in almost $2^d (= P)$ accesses on MBHs. But CSP shows only a 21.7% access ratio, and even the selectivity and dimension are very high, that is, a 21.7% access ratio when $s = 10^{-1}$ and $d = 32$. The results are shown in Figure 7.c and Figure 7.d.

CSP is apparently a better partitioning strategy for range queries when a dimension is high ($d \geq 32$). Insensitivity of CSP to the access ratio also indicates the applicability of CSP to recent database applications which deal
A. GRID when $N = 10^6$

B. CSP when $N = 10^6$

c. GRID vs. CSP when $N = 10^6$

d. GRID vs. CSP when $N = 10^9$

Figure 7: Analytical comparisons of GRID and CSP methods

with huge-sized and high-dimensional data. Such applications could be data mining, bioinformatics, data warehousing, multimedia databases, and others.

4 Conclusions

Since the geometric shape of partitioned data is directly related to the expected value of hypercubic range queries, we studied two extreme cases of partitioning methods: GRID, which partitions data or space equally at all
dimensions, and CSP, which partitions data or space as we peel an onion. Analysis of these methods based on observations of the Minkowski-sum cost model, in comparison with experiments, showed a remarkable accuracy. Analysis of GRID showed that under realistic assumptions most MBHs generated by GRID are accessed, even the selectivity is very low. Analysis of CSP showed two important metrics which are directly related to the expected value: the distance between a MBH of partitioned data and one of two one-dimensional boundaries of data space, and the number of MBHs which intersect a hypercube \([q, 1 - q]^d\) or \([1 - q, q]^d\). Further study on CSP showed that only about 21% of MBHs generated by CSP are accessed even if the selectivity is very high. And even though the dimension and the amount of data increase, the fluctuation of the ratio of the number of intersecting MBHs to \(P\) is almost insensitive to their increase at all ranges of selectivities. These facts imply that CSP is a quite suitable packing method for application areas which deal with huge-sized and high-dimensional data.

**References**


