Universality of Serial Histograms

Yannis E. Ioannidis*
Computer Sciences Department
University of Wisconsin
Madison, WI 53706
yannis@cs.wisc.edu

Abstract

Many current relational database systems use some form of histograms to approximate the frequency distribution of values in the attributes of relations and based on them estimate query result sizes and access plan costs. The errors that exist in the histogram approximations directly or transitively affect many estimates derived by the database system. We identify the class of *serial* histograms and demonstrate that they are optimal for reducing the query result size error for several classes of queries when the actual query result size (and hence the value of that error) reaches some extreme. Specifically, serial histograms are shown to be optimal for arbitrary tree equality-join queries when the query result size is maximized, whether or not the attribute independence assumption holds, and when the query result size is minimized and the attribute independence assumption holds. We also show that the expected error for any such query is always zero under all histograms, and thus argue that histograms should be chosen based on the reduction of the extreme-cases error, since reduction of the expected error is meaningless.

*Partially supported by the National Science Foundation under Grants IRI-9113736 and IRI-9157368 (PYI Award) and by grants from DEC, IBM, HP, and AT&T.

Proceedings of the 19th VLDB Conference
Dublin, Ireland, 1993

1 Introduction

Query optimizers of relational database systems decide on the most efficient access plan for a given query based on a variety of statistics on the contents of the database relations that the system maintains. These are used to estimate the values of several parameters of interest that affect the decision of the optimizer [SAC+79]. Histograms are the most common type of maintained statistics containing the number of tuples in a relation for each of several subsets of values (buckets) in an attribute. Usually, the information contained in a histogram represents an inaccurate picture of the actual contents of the database, which affects the validity of the optimizers' decisions.

We investigate the optimality of histograms for limiting the errors in the estimates of query result sizes. Our focus is on histograms that accurately record the average frequency within each bucket. In an earlier effort, we identified the class of *serial* histograms and dealt with equality-join queries where each relation is joined on the same attribute for all joins in which it participates [IC92]. We showed that the optimal histogram for reducing the worst-case error in the result size of such a query is always serial. In this paper, we generalize these results by showing that serial histograms are optimal for arbitrary tree equality-join queries when the query result size is maximized, whether or not the attribute independence assumption holds, and when the query result size is minimized and the attribute independence assumption holds. We also show that the expected error for any such query is always zero under all histograms, and thus argue that histograms should be chosen based on the reduction of the extreme-cases error, since reduction of the expected error is meaningless.

Although used in many systems, the formal properties of histograms have not been studied extensively. The few pieces of work of which we are aware deal with histograms in the context of single operations, primarily selection. Specifically, Piatetsky-Shapiro and Con-

nell dealt with the effect of histograms on reducing the error for selection queries [PSC84]. They studied two classes of histograms: in an "equi-width" histogram, the number of attribute values associated with each bucket is the same; in an "equi-depth" (or "equi-height") histogram, the total number of tuples having the attribute values associated with each bucket is the same. Their main result showed that equi-width histograms have a much higher worst-case and average error for a variety of selection queries than equi-depth histograms. Muralikrishna and DeWitt [MD88] extended the above work for multidimensional histograms that are appropriate for multi-attribute selection queries. Several other researchers have dealt with "variable-width" histograms for selection queries, where the buckets are chosen based on various criteria [MO79b, MK85, KK85]. The survey by Mannino, Chu, and Sager [MCS88] contains various references to work in the area of statistics on choosing the appropriate number of buckets in a histogram for sufficient error reduction. That work deals primarily with selections as well. Histograms for single-join queries have been minimally studied and then again without emphasis on optimality [Chr83, Koo80, MK85]. Our work is different from all the above in that it deals with arbitrarily large join queries and identifies a single class of histograms that are universally optimal for reducing errors in the worst-case.

All results in this paper are given without proof due to lack of space. The full details can be found elsewhere [Ioa93]. This paper is organized as follows. Section 2 introduces some notation for queries and their result sizes and presents the basic definitions on histograms. It also includes some mathematical results from majorization theory that are used throughout the paper. Section 3 gives a summary of the results of our work on histogram optimality for a restricted class of queries. Section 4 demonstrates the optimality of serial histograms when the query result size is maximized for arbitrary tree queries, both when the attribute independence assumption does and does not hold. Section 5 repeats for the case where the query result size is minimized and the attribute independence assumption holds. Section 6 shows that all histograms estimate accurately the expected value of the query result size, and therefore, argues against choosing histograms based on the minimization of the expected error. Finally, Section 7 summarizes our results and gives directions for future work.

2 Mathematical Foundations and Problem Formulation

2.1 Majorization Theory

This subsection presents some important results from the mathematical theory of majorization [MO79a], which are used to study the effect of histograms on limiting the error in the estimates of query result sizes. In what follows, an $(M \times N)$-*matrix* $\underline{A}$ whose entries are $a_{kl}, 1 \leq k \leq M, 1 \leq l \leq N$, is denoted by $\underline{A} = (a_{kl})$. The results presented in this paper hold for all matrices with non-negative real entries. For database applications, all entries will be non-negative integers. The transpose of an $(M \times N)$-*matrix* $\underline{A}$ is denoted by $\underline{A}^T$, and is the $(N \times M)$-*matrix* constructed from $\underline{A}$ by switching its rows with its columns. We occasionally use the terms *horizontal vector* and *vertical vector* for matrices with $M = 1$ and $N = 1$, respectively. An M-vector $\underline{a}$ whose entries are $a_i, 1 \leq i \leq M$, is denoted by $\underline{a} = (a_1 \ \ldots \ a_M)$ or by $\underline{a} = (a_i)$. An M-vector $\underline{a}$ is *nonincreasing* when $\forall 1 \leq i < M$, the inequality $a_i \geq a_{i+1}$ holds. Similarly, $\underline{a}$ is *nondecreasing* when $\forall 1 \leq i < M$, the inequality $a_i \leq a_{i+1}$ holds. Generalizing the above, a matrix $\underline{A}$ is *nonincreasing* (resp. *nondecreasing*) when all its columns and all its rows are nonincreasing (resp. nondecreasing) vectors. (Note that the set of nonincreasing (resp. nondecreasing) matrices is closed under matrix multiplication.) Finally, $\underline{A}$ is *mixed-monotone* when either all its columns are nonincreasing or all of them are nondecreasing, and similarly for all its rows. The following definition and notation is from the work of Marshall and Olkin [MO79a].

Definition 2.1 For two M-vectors $\underline{a} = (a_i)$ and $\underline{b} = (b_i)$ with non-negative entries, $\underline{a}$ *weakly majorizes* $\underline{b}$, denoted by $\underline{a} \succ_w \underline{b}$ or $\underline{b} \prec_w \underline{a}$, if $\sum_{i=1}^{K} a_i \geq \sum_{i=1}^{K} b_i$, $\forall 1 \leq K \leq M$. If, in addition to the above, $\sum_{i=1}^{M} a_i = \sum_{i=1}^{M} b_i$, then $\underline{a}$ *majorizes* $\underline{b}$, denoted by $\underline{a} \succ \underline{b}$ or $\underline{b} \prec \underline{a}$.

For the needs of this paper, we have extended the above standard definition to matrices, by viewing an $(M \times N)$-matrix as a vertical M-vector whose entries are horizontal N-vectors (the symmetric view would be equivalent).

Definition 2.2 Consider two $(M \times N)$-matrices $\underline{A} = (\underline{a}_i)$ and $\underline{B} = (\underline{b}_i)$ with non-negative entries, where for all $1 \leq i \leq M$, $\underline{a}_i$ and $\underline{b}_i$ are horizontal vectors. If $\sum_{i=1}^{K} \underline{a}_i \succ_w \sum_{i=1}^{K} \underline{b}_i$, $\forall 1 \leq K \leq M$, then $\underline{A}$ *weakly majorizes* $\underline{B}$, denoted by $\underline{A} \succ_w \underline{B}$ or $\underline{B} \prec_w \underline{A}$. If,

in addition to the above, $\sum_{i=1}^{N} \underline{a}_i = \sum_{i=1}^{N} \underline{b}_i$, then $\underline{A}$ *majorizes* $\underline{B}$, denoted by $\underline{A} \succ \underline{B}$ or $\underline{B} \prec \underline{A}$.

The following result states a well known implication of vector majorization.

Theorem 2.1 [MO79a] If $\underline{x}$ and $\underline{a}$ are nonincreasing horizontal vectors, $\underline{b}$ is a horizontal vector, and $\underline{a} \succ_w \underline{b}$, then $\underline{x}\,\underline{a}^T \geq \underline{x}\,\underline{b}^T$.

Example 2.1 As an example of the above theorem, consider the vectors $\underline{x} = (3\ 2\ 1)$, $\underline{a} = (10\ 5\ 1)$, and $\underline{b} = (1\ 5\ 10)$. The premises of Theorem 2.1 are satisfied. The same holds for the conclusion of the theorem, since $\underline{x}\,\underline{a}^T = 42 > \underline{x}\,\underline{b}^T = 23$. □

We have extended the above theorem to arbitrary products of matrices.

Theorem 2.2 If for all $1 \leq j \leq N$, $\underline{A}^{(j)}$ and $\underline{B}^{(j)}$ are nonincreasing matrices, and $\underline{A}^{(j)} \succ_w \underline{B}^{(j)}$, then the inequality $\underline{A}^{(1)}\,\underline{A}^{(2)} \cdots \underline{A}^{(N)} \succ_w \underline{B}^{(1)}\,\underline{B}^{(2)} \cdots \underline{B}^{(N)}$ holds.

2.2 Problem Formulation

The focus of this paper is tree equality-join queries. In what follows we often omit the 'tree' and/or 'equality-join' qualifications. Without loss of generality, we assume that joins between relations are on individual attributes. For example, if R_0, R_1, R_2 are relation names and a, b are attribute names of these relations, we do not deal with queries whose qualifications contain ($R_0.a = R_1.a$ **and** $R_0.b = R_1.b$). Also without loss of generality, we only deal with *chain* join queries, i.e., ones whose qualification is of the generic form

$$\begin{aligned} Q \;:=\; (R_0.a_1 = R_1.a_1 \quad &\textbf{and} \quad R_1.a_2 = R_2.a_2 \;\textbf{and} \\ \ldots \quad &\textbf{and} \quad R_{N-1}.a_N = R_N.a_N), \end{aligned}$$

where $R_0, \ldots, R_N$ are relations and $a_1, \ldots, a_N$ are appropriate attributes. Generalizing the results presented in this paper to arbitrary tree queries is straightforward. The required mathematical machinery becomes hairier but its essence remains unchanged.

Consider the above query Q and let $\mathcal{D}_j$ be the (finite) *domain* of attribute $a_j, 1 \leq j \leq N$. In principle, $\mathcal{D}_j$ contains all the potential attribute values that could appear in attribute a_j of either relation R_{j-1} or R_j. In practice, however, $\mathcal{D}_j$ may be assumed to contain only the a_j values that actually appear in the database at some point. The results presented below do not depend on the particular definition of $\mathcal{D}_j$. Let M_j be the size of $\mathcal{D}_j$ and $\mathcal{D}_j = \{d_{ij} | 1 \leq i \leq M_j\}$. Also, for convenience, define $M_0 = M_{N+1} = 1$. Note that $i < k$ does not imply $d_{ij} < d_{kj}$, i.e., the numbering of attribute values is arbitrary and does not reflect some natural ordering of them. The *frequency matrix* $\underline{T}_j = (t_{kl})$ of relation $R_j, 0 \leq j \leq N$, is defined as an $(M_j \times M_{j+1})$-matrix, whose t_{kl} entry is the number of tuples in R_j with $R_j.a_j = d_{kj}$, for $j > 0$, and $R_j.a_{j+1} = d_{l(j+1)}$, for $j < N$. Each entry t_{kl} is the *frequency* of the pair $< d_{kj}, d_{l(j+1)} >$ in the attributes a_j, a_{j+1} of R_j. Note that the frequency matrices of R_0 and R_N are a horizontal and a vertical vector, respectively. Occasionally, it is also useful to treat all the frequencies in $\underline{T}_j$ as a collection, ignoring the attribute values with which each frequency is associated. That collection is in general a bag (or multiset, i.e., it may contain duplicates), is called the *frequency set* of R_j, and is denoted by B_j.

The following theorem establishes that the size of the result of an equality-join query is equal to the product of the frequency matrices of the participating relations.

Theorem 2.3 The size S of the result relation of query Q is equal to

$$S = \underline{T}_0\,\underline{T}_1 \cdots \underline{T}_N. \tag{1}$$

Example 2.2 Consider three relations R_0, R_1, and R_2, whose frequency matrices are

$$\underline{T}_0 = (\,20\,15\,),\ \underline{T}_1 = \begin{pmatrix} 25 & 10 & 6 \\ 12 & 4 & 3 \end{pmatrix},\ \underline{T}_2 = \begin{pmatrix} 21 \\ 16 \\ 5 \end{pmatrix}.$$

One can easily verify that the size of the result of the corresponding query is equal to $S = \underline{T}_0\underline{T}_1\underline{T}_2 = 19,265$. □

As mentioned above, in this paper, we are primarily concerned with the cases when S reaches some extreme. The following results apply Theorem 2.2 on (1) to identify conditions for when this happens.

Theorem 2.4 [1] Consider an equality-join query Q on relations R_j, $0 \leq j \leq N$, and let B_j, $0 \leq j \leq N$, be their frequency sets. Also consider, for each $0 \leq j \leq N$, all possible arrangements of the elements of B_j in the frequency matrix $\underline{T}_j$. There is one such arrangement where, for all $0 \leq j \leq N$, $\underline{T}_j$ is nonincreasing and the result size of Q is maximized.

[1] This has been proved earlier for the special cases of a product of vectors by Marshall and Olkin [MO79a] and the product of a square matrix with itself by Schwarz [Sch64].

Theorem 2.5 [MO79a] Consider an equality-join query Q on relations R_j, $0 \leq j \leq N$, such that, for all $0 < j < N$, the frequency matrix of R_j must be diagonal, and let B_j, $0 \leq j \leq N$, be their frequency sets. The result size of Q is maximized when, for all $0 \leq j \leq N$, the elements of B_j are arranged so that $\underline{T}_0$, $\underline{T}_N$, and the main diagonals of $\underline{T}_j$, $0 < j < N$, are nonincreasing.

Theorem 2.6 [MO79a] Consider a 2-way equality-join query Q on relations R_0 and R_1, and let B_0 and B_1 be their frequency sets. The result size of Q is minimized when the elements of B_0 and B_1 are arranged so that the frequency matrices (vectors) $\underline{T}_0$ and $\underline{T}_1$ are nonincreasing and nondecreasing, respectively.

Example 2.3 Consider a database with two relations R_0 and R_1 both of size T with a common attribute a_1, which follows the Zipf distribution in both relations [Chr84, Zip49]:

$$t_i = T\frac{1/i^z}{\sum_{i=1}^{M} 1/i^z} \text{ for all } 1 \leq i \leq M_1. \qquad (2)$$

The main characteristic of the Zipf distribution is that there are few attribute values with high frequencies and many with low frequencies. This has been claimed to be a characteristic of the distributions seen in many databases. Assume that $T = 10000$ (the size of the relations) and M_1=100. Figure 1 is a graphical representation of (2) for $z = 0, 0.02, \ldots, 0.1$, where the x-axis represents i, the rank of the attribute value with respect to its associated frequency in descending order. For $z = 0$, the Zipf and the uniform distributions coincide, but as z increases, their deviation from each other increases.

Note that, for all z, the frequency vector that corresponds to the Zipf distribution is nonincreasing. By Theorem 2.4, this maximizes the result size of the join between R_0 and R_1. If we reverse the Zipf frequencies in one of the vectors, that vector will become nondecreasing, and by Theorem 2.6, the result size of the join will be minimized. The following table shows these minimum and maximum values (in millions) for various values of z (same value for both relations).

z	0.0	0.2	0.4	0.6	0.8	1.0
Max Size	1.0	1.05	1.27	1.88	3.30	6.08
Min Size	1.0	0.97	0.89	0.75	0.57	0.38

□

2.3 Histograms

Among commercial systems, maintaining *histograms* is a very common approach to approximating frequency

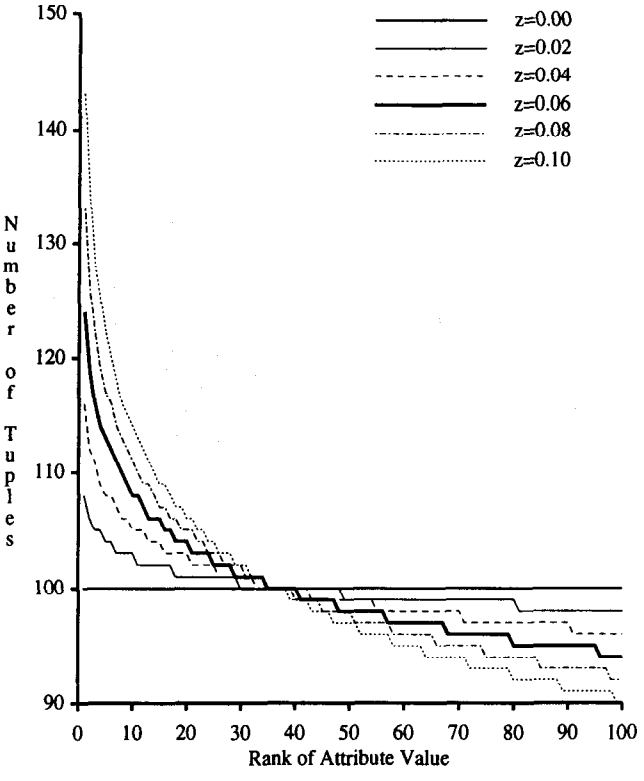

Figure 1: Zipf frequency distribution.

matrices. In what follows, we discuss histograms for two-dimensional matrices; histograms for matrices of any other dimension are defined similarly. In a histogram on attributes a_j, a_{j+1} of relation $R_j, 0 < j < N$, the set $\mathcal{D}_j \times \mathcal{D}_{j+1}$ is partitioned into *buckets*, and a uniform distribution is assumed within each bucket. That is, for any bucket b in the histogram, if $< d_{kj}, d_{l(j+1)} > \in b$ then t_{kl} is approximated by the closest integer to $\sum_{<d_{mj}, d_{n(j+1)}> \in b} t_{mn}/|b|$. The approximate frequency matrix captured by a histogram is called the *histogram matrix*. Note that any arbitrary subset of $\mathcal{D}_j \times \mathcal{D}_{j+1}$ may form a bucket, e.g., bucket $\{< d_{1j}, d_{4(j+1)} >, < d_{7j}, d_{2(j+1)} >\}$. Also note that the 'uniform distribution assumption' corresponds to maintaining a histogram with a single bucket. Such a histogram is called *trivial*. Whenever R_j is updated, the corresponding histogram matrix may need to be updated as well. The mechanism for this depends on how histograms are implemented. Both histogram implementation and histogram updates are outside the scope of this paper and do not affect the results presented, so they are not discussed any further.

Example 2.4 To illustrate the above definition of histograms, consider the following relation schema: WorksFor(*ename,dname*,year). The attributes in ital-

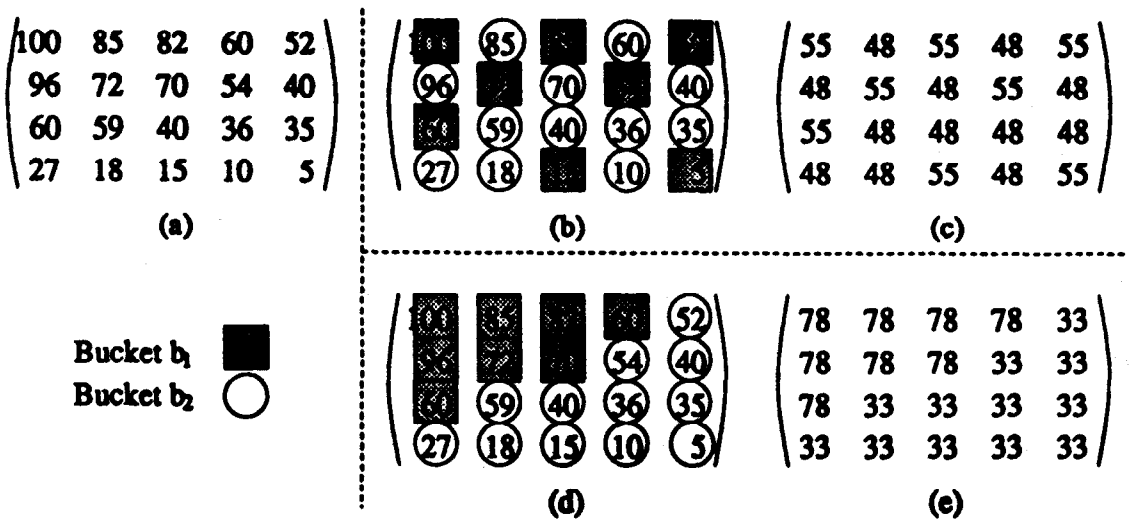

Figure 2: Frequency matrix and two histogram matrices on WorksFor.

ics form the key to the relation, and represent employee and department names such that the employee is working in the department. The year attribute represents the year the given employee started working at the given department. We focus on the combination of dept and year attributes, and assume for simplicity that there are four different departments (toy, jewelry, shoe, and candy) and five different years (1989 through 1993). The frequency matrix of the relation is shown in Figure 2(a), where dname is used for the rows and year is used for the columns of the matrix, and the corresponding values are arranged in the order specified above. Note that the matrix is nonincreasing. An example histogram matrix with two buckets is shown in Figures 2(b) and 2(c). In the former, we show the original matrix with an indication of which attribute value pairs (or equivalently, frequencies) are placed in which bucket. In the latter, we show the actual histogram matrix that is the result of averaging the frequencies in each bucket. Another example histogram matrix with two buckets is shown in Figures 2(d) and 2(e), again depicted in the two ways discussed for the first histogram. □

Optimal histograms are defined as follows.

Definition 2.3 Consider a query Q on relations $R_j, 0 \leq j \leq N$, whose result size is S, as determined by the frequency matrices of the relations. For each relation R_j, let $\mathcal{H}_j$ be a collection of histograms of interest. The $(N+1)$-tuple $< H_j >$, where $H_j \in \mathcal{H}_j, 0 \leq j \leq N$, is an *optimal* histogram tuple for Q within $< \mathcal{H}_j >$, if it minimizes $|S - S'|$, where S' is the approximate query result size determined by any such histogram tuple.

Note that optimality is defined per query and per collection of frequency matrices, and for the histograms of all relations together. The reason is that the optimal histograms differ for different queries and for different frequency matrices. The following defines a very important class of histograms for relations with nonincreasing matrices.

Definition 2.4 Consider relation $R_j, 0 \leq j \leq N$, with a nonincreasing frequency matrix $\underline{T}_j$. A histogram for relation R_j is *serial with respect to* its buckets b_1 and b_2, if either $\forall < d_{kj}, d_{l(j+1)} > \in b_1, < d_{mj}, d_{n(j+1)} > \in b_2$, the inequality $t_{kl} \geq t_{mn}$ holds, or $\forall < d_{kj}, d_{l(j+1)} > \in b_1, < d_{mj}, d_{n(j+1)} > \in b_2$, the inequality $t_{kl} \leq t_{mn}$ holds. It is called *serial* if it is serial with respect to all pairs of its buckets.

Note that the buckets of a serial histogram group frequencies that are close to each other with no interleaving. For example, the histogram of Figures 2(d)-(e) is serial, while that of Figures 2(b)-(c) is not.

3 Previous Results

In this section, we summarize the main results that we have obtained earlier [IC92], which form the basis for the results presented in this paper. Specifically, in our previous work, we dealt with a quite restricted type of equality-join queries, called $t - clique$ queries, where each relation is joined on exactly the same attribute for all joins in which it participates. The formulation and the underlying mathematical foundations required for the problem were much more restrictive than what is required for this work. It involved only vectors and

majorization on them. For uniformity with the rest of the paper, however, we cast the specialized problem into our formulation of Section 2.2 and use that to present the results of our earlier work. Specifically, dealing with a t-clique query is equivalent to dealing with a query Q of the general form discussed in this paper with the additional restriction on the data that, for each relation $R_j, 0 < j < N$, the functional dependencies $a_j \rightarrow a_{j+1}$ and $a_{j+1} \rightarrow a_j$ hold. Without loss of generality, assume that the attribute value mappings under the above functional dependencies satisfy $d_{ij} \rightarrow d_{i(j+1)}$ and $d_{i(j+1)} \rightarrow d_{ij}$, for all $1 \leq i \leq M_j$. Then, one can easily verify that each frequency matrix $\underline{T}_j, 0 < j < N$, is square and diagonal. Hence, the query result size is equal to the sum of the componentwise product of their diagonals and the vectors $\underline{T}_0$ and $\underline{T}_N$.

We have concentrated on the case where the query result size is maximized. Since we are only concerned with diagonal matrices, by Theorem 2.5, maximization occurs when the diagonals of all frequency matrices $\underline{T}_j, 0 < j < N$, and the vectors $\underline{T}_0$ and $\underline{T}_N$ are nonincreasing. This maximum is never exceeded by the result size approximated by any histogram, hence, the optimal histogram must maximize that approximation.

We have investigated histogram optimality within the class of histograms $\mathcal{B}$ that only partition the diagonal entries and ignore the rest (or equivalently put all the rest in a separate exclusive bucket). In what follows, when referring to a bucket of a histogram in $\mathcal{B}$, we always mean a bucket formed by some of the diagonal entries. Let $\mathcal{B}_\beta$ be the subset of $\mathcal{B}$ that only contains histograms of size β, i.e., with β buckets. Note that, for any given β, all histograms in $\mathcal{B}_\beta$ are equivalent with respect to the amount of information that they maintain. The following pair of theorems establish the importance of serial histograms.

Theorem 3.1 [IC92] For any histogram size β and any histogram $H \in \mathcal{B}_\beta$, there is a serial histogram in $\mathcal{B}_\beta$ that majorizes it.

Theorem 3.2 [IC92] Consider a t-clique query Q on relations R_j, $0 \leq j \leq N$, with nonincreasing diagonals in their frequency matrices and an $(N+1)$-tuple of histogram sizes $< \beta_j >$. There exists an optimal histogram tuple for Q within $< \mathcal{B}_{\beta_j} >$ where all histograms in it are serial.

Histograms are usually constructed in a way that each bucket stores attribute values that belong in a certain range in the natural total order of the attribute domain. The important implication of Theorem 3.2 is that this traditional approach may be far from optimal for t-clique queries. Histograms should be constructed so that attribute values are grouped in buckets based on closeness in their corresponding frequencies and not in their actual values. This is a significant difference, since the two orderings may be completely unrelated.

Identifying which of the many serial histograms is the optimal one in each case is not straightforward. In our previous work, we have solved the problem for the two extreme cases with respect to query size. First, for 2-way equality-join queries (all of which are t-clique) on relations with nonincreasing frequency matrices (vectors), we have obtained a closed-form formula identifying the buckets that should be formed in the optimal histograms. Second, for t-clique queries with N joins, we have proved that, as $N \rightarrow \infty$, the optimal histograms of all relations tend to become identical favoring the placement of the highest frequencies in individual buckets. We do not attempt to formally present the above results due to lack of space. Similarly we do not discuss several other results on optimal histograms within other interesting classes of histograms (details can be found elsewhere [IC92]). However, there is one important result that we want to discuss, since we make use of it later. In what follows, we refer to the class of serial histograms $\mathcal{S}$ and its subclasses $\mathcal{S}_\beta$ of serial histograms with β buckets.

Theorem 3.3 Consider a 2-way equality-join query Q on two relations R_0 and R_1 with nonincreasing frequency matrices (vectors), and an integer $\beta \geq 1$. If $H \in \mathcal{S}_\beta$ is the histogram used for R_0, then for Q, H is optimal within $\bigcup_{\beta'=1}^{M} \mathcal{S}_{\beta'}$ for R_1 as well.

An implication of the above is that, for 2-way join queries, for optimal approximations, the same histogram should be used for both relations involved.

In the rest of the paper, we generalize the above results for arbitrary tree queries and data distributions. Particular consideration is given to some important special cases.

4 Maximum Value of the Query Result Size

4.1 The Attribute Independence Assumption

To the best of our knowledge, most database systems employ the *attribute independence assumption* when estimating the sizes of query results. Expressed in terms of frequency matrices, the assumption states

that, in the frequency matrix $\underline{T}_j$ of relation R_j, for all $1 \leq k, m \leq M_j$, $1 \leq l, n \leq M_{j+1}$, the equality $t_{kl}/t_{kn} = t_{ml}/t_{mn}$ holds. That is, if the tuples of R_j are grouped based on their value in the a_j attribute, the frequency distribution of the values in attribute a_{j+1} within each group is identical up to a constant factor. Similarly if the grouping is based on the values in the a_{j+1} attribute. One can easily verify that, when the above holds, $\underline{T}_j$ is equal to the product of a vertical vector with a horizontal vector.

Example 4.1 The following frequency matrix is equal to any of the given products of vectors:

$$\begin{pmatrix} 50 & 20 & 10 \\ 30 & 12 & 6 \\ 15 & 6 & 3 \end{pmatrix} = \begin{pmatrix} 10 \\ 6 \\ 3 \end{pmatrix} (5 \ \ 2 \ \ 1)$$

$$= \begin{pmatrix} 80 \\ 48 \\ 24 \end{pmatrix} (0.625 \ \ 0.25 \ \ 0.125).$$

We show two different products to bring up the point that, for frequency matrices that satisfy the attribute independence assumption, there is an infinite number of pairs of vectors whose product is equal to the matrix. Moreover, choosing a single entry in any one of the two multiplied vectors uniquely determines all the other entries. The first product above represents an arbitrary such choice. The second product uses the sums of the entries of each row for the vertical vector. Each entry of the horizontal vector ends up being the percentage of the tuples of any row attribute value associated with the column attribute value corresponding to that entry. □

When the attribute independence assumption holds, the frequency matrix $\underline{T}_j$ should be approximated with two histograms on the individual attributes a_j and a_{j+1} instead of one on the combination of the attributes. The reason is dual: first, most systems only support single-dimensional histograms, possibly because they are useful more often than higher-dimension ones; second, a two-dimensional histogram may destroy the independence of attributes, i.e., may result in a histogram matrix that is not a product of two vectors, and this important piece of information on the data will be lost. Therefore, the question of histogram optimality for this case is reduced to identifying the optimal single-dimensional histogram for each attribute a_j separately.

Consider query Q on relations $R_j, 0 \leq j \leq N$, whose frequency matrices are nonincreasing (so, by Theorem 2.4, S is maximized) and satisfy the attribute independence assumption. For all $0 < j < N$, let $\underline{T}_j = \underline{v}_j \underline{h}_{j+1}$, where $\underline{v}_j$ and $\underline{h}_{j+1}$ are vertical and horizontal vectors, respectively. The specific choice of $\underline{h}_j$ and $\underline{v}_j$ is not important, since they are all identical up to a constant factor, which does not affect the optimality of histograms. For convenience, define $\underline{T}_0 = \underline{h}_1$ and $\underline{T}_N = \underline{v}_N$. Then, by (1), the size of the result of Q is equal to

$$S = \underline{T}_0 \underline{T}_1 \cdots \underline{T}_N = (\underline{h}_1 \underline{v}_1)(\underline{h}_2 \underline{v}_2) \cdots (\underline{h}_N \underline{v}_N). \quad (3)$$

Consider the parenthesization shown in the last formula. Each parenthesis is a product of a horizontal with a vertical vector, i.e., the result of each parenthesis is equal to a scalar number. As mentioned above, each histogram will be associated to an individual vector, independent of all others. Therefore, maximizing the approximation to S is equivalent to maximizing each parenthesis separately. This is equivalent to treating the query Q as multiple independent 2-way join queries, and maximizing the approximation to the result of each one. Hence, the results presented in Section 3 can be used to identify the optimal histogram for this case as well. Specifically, let $\mathcal{H}$ be the class of all histograms for vectors and $\mathcal{H}_\beta$ its subclass of histograms with β buckets. Recall that, by Theorem 3.3, for all $1 \leq j \leq N$, the optimal histograms for $\underline{h}_j$ and $\underline{v}_j$ are the same. Then, Theorem 3.2 implies the following.

Theorem 4.1 Consider an equality-join query Q on relations $R_j, 0 \leq j \leq N$, with nonincreasing frequency matrices that satisfy the attribute independence assumption, and an N-tuple of histogram sizes $< \beta_j >$, where β_j is associated with both $\underline{h}_j$ and $\underline{v}_j$, $1 \leq j \leq N$. There exists an optimal histogram tuple for Q within $< \mathcal{H}_{\beta_j} >$ where all histograms in it are serial.

Example 4.2 Consider a query Q on many relations that satisfy the premises of Theorem 4.1. Assume that the columns and rows of the frequency matrices of all relations are identically distributed based on the Zipf distribution introduced in Example 2.3 with $z = 0.2$. Recall that the domain size for all attributes is 100. Assume that all individual attribute histograms maintained are identical as well. We have calculated the error generated when the histograms are trivial (i.e., they capture a uniform distribution) and for three other interesting types of histograms that have five buckets: (a) a nonserial histogram whose i-th bucket, $1 \leq i \leq 5$, includes the $(5x+1)$-st highest frequencies, $0 \leq x \leq 19$; (b) the unique serial histogram with five buckets with twenty elements each; and (c) the unique serial histogram with four buckets containing the four highest frequencies and one containing the remaining

frequencies. The values of the relative error in the estimate of the query result sizes for various sizes of queries is shown below.

Histogram	Number of Joins			
	1 join	2 joins	5 joins	10 joins
Trivial	4.64%	9.50%	25.46%	57.39%
Nonserial	4.60%	9.41%	25.22%	56.79%
Serial-(b)	1.10%	2.21%	5.62%	11.56%
Serial-(c)	2.15%	4.34%	11.22%	23.70%

As expected, the serial histograms do better than the trivial and the nonserial ones. Note that the nonserial and the trivial histograms generate almost identical errors, although the former maintains five times more information than the latter. Also, although not clear due to their small values, the errors grow exponentially with the query size. □

As mentioned in Section 3, for 2-way equality-join queries, not only have we established the optimality of serial histograms, but we also have obtained a closed-form formula identifying the buckets that should be formed in the optimal histograms. These results may be carried over to arbitrary queries under the attribute independence assumption to identify the specific serial histograms that are optimal for each join attribute separately.

In closing this discussion, we would like to comment on the case where the attribute independence assumption is made by a database system although the data does not satisfy it. That is, each attribute is dealt with separately by the system, and the frequency matrix of a relation for a query Q is approximated by a product of a vertical with a horizontal vector. It is easily verifiable that there is a unique matrix that satisfies the above and preserves the row-sums and column-sums of the original matrix. Consider a query Q and one of the relations $R_j, 0 < j < N$, and let $\underline{T}_j$ be its actual frequency matrix and $\underline{T}'_j$ be its corresponding unique approximation that satisfies the attribute independence assumption and preserves the original row-sums and column sums. Depending on characteristics of $\underline{T}_j$, replacing $\underline{T}_j$ with $\underline{T}'_j$ in (1) may increase or decrease the computed result size. In the former case, the histograms that would be optimal for approximating $\underline{T}'_j$ may result in an approximation of the query result size S that is greater than S. This implies that serial histograms for individual attributes may not be optimal when the attribute independence assumption does not hold.

Example 4.3 Consider the following two matrices for which the attribute independence assumption does not hold:

$$\underline{T}_1 = \begin{pmatrix} 4 & 3 \\ 2 & 1 \end{pmatrix}, \quad \underline{T}_2 = \begin{pmatrix} 10 & 3 \\ 2 & 1 \end{pmatrix}.$$

The corresponding approximations that do satisfy the assumption and preserve the original row-sums and column-sums are

$$\underline{T}'_1 = \begin{pmatrix} 4.2 & 2.8 \\ 1.8 & 1.2 \end{pmatrix}, \quad \underline{T}'_2 = \begin{pmatrix} 9.75 & 3.25 \\ 2.25 & 0.75 \end{pmatrix}.$$

It is easy to verify that $\underline{T}_1 \prec \underline{T}'_1$ and $\underline{T}'_2 \prec \underline{T}_2$. Thus, for any horizontal 2-vector $\underline{h}$ and vertical 2-vector $\underline{v}$, $\underline{h}\,\underline{T}_1\,\underline{v} \leq \underline{h}\,\underline{T}'_1\,\underline{v}$ while $\underline{h}\,\underline{T}_2\,\underline{v} \geq \underline{h}\,\underline{T}'_2\,\underline{v}$, i.e., all other things being equal, $\underline{T}'_1$ would overestimate query result sizes, while $\underline{T}'_2$ would underestimate them. □

4.2 The General Case

In this section, we address histogram optimality when no assumptions are being made about the frequency matrices of relations. We operate within the class of general histograms $\mathcal{H}$ and refer to its subclass $\mathcal{H}_\beta$, which contains only histograms with β buckets.

Consider query Q on relations $R_j, 0 \leq j \leq N$, whose frequency matrices are nonincreasing (hence, S is maximized) and focus on one of its relations $R_j, 0 < j < N$. Let $\underline{h} = \underline{T}_0\underline{T}_1 \cdots \underline{T}_{j-1}$ and $\underline{v} = \underline{T}_{j+1}\underline{T}_{j+2} \cdots \underline{T}_N$, where $\underline{h} = (h_1\ h_2\ \ldots\ h_{M_j})$ and $\underline{v}^T = (v_1\ v_2\ \ldots\ v_{M_{j+1}})$. Then, clearly $S = \underline{h}\underline{T}_j\underline{v}$, and more precisely $S = \sum_{k=1}^{M_j} \sum_{l=1}^{M_{j+1}} h_k t_{kl} v_l$. Based on the premise that the actual query result size is maximized, and the fact that the product of two nonincreasing matrices is nonincreasing, Theorem 2.4 implies that for all $1 \leq k, m \leq M_j, 1 \leq l, n \leq M_{j+1}$, $h_k v_l \leq h_m v_n$ if and only if $t_{kl} \leq t_{mn}$. Hence, independent of the specific entries of $\underline{h}$ and $\underline{v}$, Theorem 3.2 can be applied to yield the following:

Theorem 4.2 Consider a query Q on relations R_j, $0 \leq j \leq N$, with nonincreasing frequency matrices, and an $(N+1)$-tuple of histogram sizes $< \beta_j >$. There exists an optimal histogram tuple for Q within $< \mathcal{H}_{\beta_j} >$ where all histograms in it are serial.

The above theorem shows that even in the most general case of equality-join queries, serial histograms are optimal when the actual query result size is maximized.

5 Minimum Value of the Query Result Size

All the results presented in Sections 3 and 4 deal with the case where the actual query result size is maxi-

mized. The other extreme, when the actual query result size is minimized, is equally interesting but harder to deal with in general. The reason for the difficulty is that there are no general results on arrangements of the entries of arbitrary numbers of matrices that guarantee the minimization of their product, i.e., there is no counterpart to Theorem 2.4. Theorem 2.6 is the only result that we are aware of, dealing with the special case of 2-way equality-join queries. In the following subsection, we study this case, and show that serial histograms are again optimal. We then use that result to show optimality of serial histograms for arbitrary queries when the attribute independence assumption holds.

5.1 2-Way Equality Join Queries

The basis for the results in this section is the following theorem, which is similar to Theorem 2.1.

Theorem 5.1 [MO79a] If $\underline{a}^{(1)}$ and $\underline{b}^{(1)}$ are nonincreasing horizontal vectors, $\underline{a}^{(2)}$ and $\underline{b}^{(2)}$ are nondecreasing vertical vectors, and $\underline{a}^{(1)} \succ \underline{b}^{(1)}$ and $\underline{a}^{(2)} \prec \underline{b}^{(2)}$, then $\underline{a}^{(1)}\underline{a}^{(2)} \leq \underline{b}^{(1)}\underline{b}^{(2)}$.

Note that $\underline{a} \succ \underline{b}$ for nondecreasing vectors $\underline{a}$ and $\underline{b}$ is equivalent to $\underline{b}' \succ \underline{a}'$, where $\underline{a}'$ is constructed from $\underline{a}$ by reversing the order of its entries, and similarly for $\underline{b}'$.

Consider a query with two relations whose corresponding frequency vectors are nonincreasing and nondecreasing, respectively. By Theorem 2.6, S is minimized. This minimum is never exceeded by the result size approximated by any histogram, hence, the optimal histogram must minimize that approximation. Theorem 5.1 implies that histograms should be compared again in terms of majorization. For nonincreasing vectors, the desirability of serial histograms has been shown by Theorem 3.1. For nondecreasing vectors, a similar result can be obtained, essentially as a corollary of Theorem 3.1.

Corollary 5.1 Consider nondecreasing vectors. For any histogram size β and any histogram $H \in \mathcal{H}_\beta$, there is a serial histogram in $\mathcal{H}_\beta$ majorized by H.

The main result of this section is a consequence of Corollary 5.1 and Theorem 5.1.

Theorem 5.2 Consider a 2-way equality-join query Q on relations R_0 and R_1 and a pair of histogram sizes $< \beta_0, \beta_1 >$. Assume that the frequency vectors of R_0 and R_1 are nonincreasing and nondecreasing, respectively. There exists an optimal histogram pair for Q within $< \mathcal{H}_{\beta_0}, \mathcal{H}_{\beta_1} >$ where both histograms in it are serial.

Identifying the particular serial histograms that are optimal for the case where the query result size is minimized depends on the frequency vectors corresponding to the query relations. Unlike for the maximizing case (Section 3), we have not been able to obtain any formal results that settle the question. However, there is a counterpart to Theorem 3.3 for this case as well, which reduces the number of potentially optimal histogram pairs from quadratic to linear in the size of domain $\mathcal{D}_1$. Recall that $\mathcal{S}$ is the class of serial histograms and $\mathcal{S}_\beta$ is its subclass of such histograms with β buckets.

Theorem 5.3 Consider a 2-way equality-join query Q on relations R_0 and R_1 and an integer $\beta \geq 1$. Assume that the frequency vectors of R_0 and R_1 are nonincreasing and nondecreasing, respectively. If $H \in \mathcal{S}_\beta$ is the histogram used for R_0, then for Q, H is optimal within $\bigcup_{\beta'=1}^{M} \mathcal{S}_{\beta'}$ for R_1 as well.

The above implies that $\beta_0 = \beta_1$ should hold in the statement of Theorem 5.2.

Example 5.1 Consider a relation R_0 whose a_1 attribute follows the Zipf distribution with z=0.2 introduced in Example 2.3. Assume that a 2-bucket serial histogram is used for R_0 such that the $p = 10$ highest frequencies form a bucket, and the remaining 90 frequencies form another. Figure 3 shows the absolute value of the error in approximating the query result size as a function of the corresponding break-point p' of a 2-bucket histogram for R_1. Three different Zipf frequency distributions for R_1 are shown with z=0.2, 0.5, and 1.0, respectively. For R_1 these are increasing Zipf distributions, i.e., the first attribute value is associated with the lowest frequency of the Zipf distribution, while the last attribute value is associated with its highest frequency. In all cases, $p' = 10$ generates the least error, which grows on the two sides of the optimal p' value, indicating the importance of choosing the appropriate histogram. As expected, more skewed distributions (e.g., Zipf with $z = 1.0$) are affected more severely. □

5.2 Arbitrary Queries under the Attribute Independence Assumption

Based on Section 4.1, the result size of a query Q on relations $R_j, 0 \leq j \leq N$, that satisfy the attribute independence assumption is given by (3), i.e., $S = (\underline{h}_1\underline{v}_1)(\underline{h}_2\underline{v}_2)\cdots(\underline{h}_N\underline{v}_N)$, where for all $0 < j < N$,

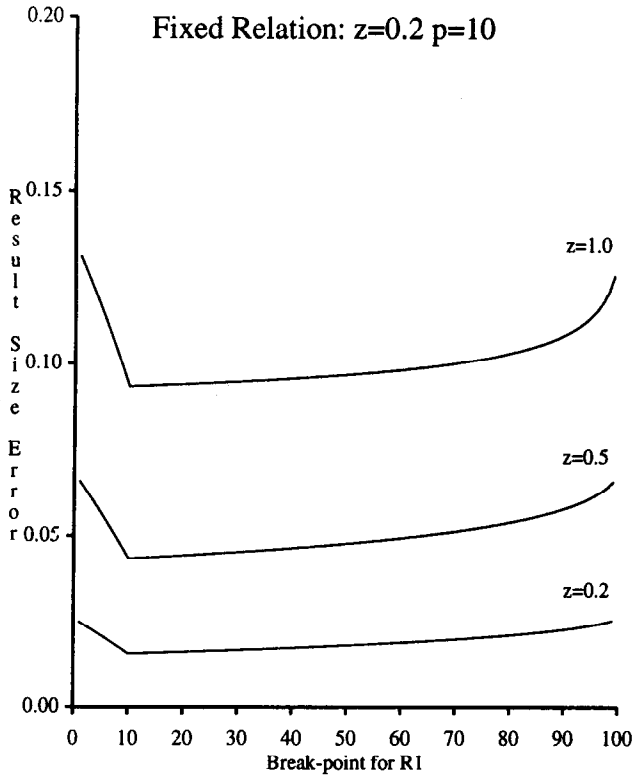

Figure 3: Choosing a histogram for one relation given the histogram of the other.

$\underline{v}_j$ and $\underline{h}_{(j+1)}$ are vertical and horizontal vectors, respectively. By Theorem 2.6, S is minimized when, for all $1 \leq i \leq N$, either $\underline{h}_j$ is nonincreasing and $\underline{v}_j$ is nondecreasing, or vice-versa. Concentrating on histograms on individual attributes, the approximation to S is minimized when the approximation to each individual product $\underline{h}_j\underline{v}_j$ is minimized. Hence, by Theorem 5.2 the following can be derived:

Theorem 5.4 Consider an equality-join query Q on relations R_j, $0 \leq j \leq N$, with (mixed-monotone) frequency matrices that satisfy the attribute independence assumption and minimize the result size of Q. Also consider an N-tuple of histogram sizes $< \beta_j >$, where β_j is associated with both $\underline{h}_j$ and $\underline{v}_j$, $1 \leq j \leq N$. There exists an optimal histogram tuple for Q within $< \mathcal{H}_{\beta_j} >$ where all histograms in it are serial.

6 Expected Value of the Query Result Size

The previous results provide answers to the histogram optimality question for the extreme cases, when the query result size is maximized or minimized. Formal results that are applicable to arbitrary cases are unlikely to exist, due to the complexity of the operations involved. In fact, one can construct examples where many serial and nonserial histograms with a large number of buckets produce higher errors than the trivial histogram. Hence, we argue that a database system should use serial histograms to limit the error in the extreme cases, so that the worst may be avoided.

Another approach that could potentially be useful would be to identify the optimal histogram for the average case and use that in database systems. Specifically, consider the expected value of the result size of a query Q over all possible associations of frequencies in the frequency sets of the relations of Q to attribute values in the corresponding domains. Define optimal histograms as those that minimize the difference between the above and the expected value of the approximate result size that they generate. The results presented below show that all histograms are equivalent in that respect. Therefore, this approach is not useful.

We first provide a formal definition for the expected value of the query result size. We assume that for each relation R_j, $0 \leq j \leq N$, in query Q, its frequency set B_j together with a partitioning of B_j into buckets are given. Let B'_j be the approximate frequency set generated from B_j by replacing each frequency by the average of the frequencies that belong to the same bucket in the given partitioning. For each frequency matrix $\underline{T}_j$, consider all possible arrangements of the elements of B_j in the matrix that respect any functional dependencies that may hold in R_j, $0 < j < N$. Each combination of arrangements in all matrices corresponds to a (possibly unique) value for the query result size. The average of all these values is the expected value for the query result size and is denoted by $E[S]$. Similarly, consider all possible combinations of arrangements of the elements of B'_j in the frequency matrices that respect any functional dependencies that may hold in R_j, $0 < j < N$. Then, generate the expected approximate value for the query result size, which is denoted by $E[S']$. The following theorem deals with the value of $E[S] - E[S']$.

Theorem 6.1 Consider a tree equality-join query Q on relations R_j, $0 \leq j \leq N$, and an $(N+1)$-vector $< B_j >$ of frequency sets together with partitionings of them. If $E[S]$ and $E[S']$ are defined as above, then $E[S] - E[S'] = 0$.

Note that Theorem 6.1 deals with arbitrary histograms, not only serial ones. The above, somewhat surprising, result implies that all histograms are accurate in their approximation of the expected value of the query result size. Hence, this quantity cannot be used for optimizing the histogram choice.

Example 6.1 As a simple example of the equality between $E[S]$ and $E[S']$, consider a query Q on two relations, R_0 and R_1, such that their frequency sets are $B_0 = \{a, b, c\}$ and $B_1 = \{d, e, f\}$. Let these sets be partitioned as $\{\{a\}, \{b, c\}\}$ and $\{\{d\}, \{e, f\}\}$, respectively, resulting in the approximate frequency sets $B_0' = \{a, (b+c)/2, (b+c)/2\}$ and $B_1' = \{d, (e+f)/2, (e+f)/2\}$. To compute the expected value of the query result size, it is enough to consider a fixed frequency vector for R_0 and examine all arrangements of the elements of B_1 for the frequency vector of R_1. There are six such arrangements, resulting in the following formula for $E[S]$:

$$\begin{aligned} 6\ E[S] &= (ad+be+cf)+(ad+bf+ce) \\ &+ (ae+bd+cf)+(ae+bf+cd) \quad (4) \\ &+ (af+bd+ce)+(af+be+cd). \end{aligned}$$

Similarly, the following formula is obtained for $E[S']$:

$$\begin{aligned} 6\ E[S'] &= (ad+\frac{b+c}{2}\frac{e+f}{2}+\frac{b+c}{2}\frac{e+f}{2}) \\ &+ (ad+\frac{b+c}{2}\frac{e+f}{2}+\frac{b+c}{2}\frac{e+f}{2}) \\ &+ (a\frac{e+f}{2}+\frac{b+c}{2}d+\frac{b+c}{2}\frac{e+f}{2}) \\ &+ (a\frac{e+f}{2}+\frac{b+c}{2}\frac{e+f}{2}+\frac{b+c}{2}d) \quad (5) \\ &+ (a\frac{e+f}{2}+\frac{b+c}{2}d+\frac{b+c}{2}\frac{e+f}{2}) \\ &+ (a\frac{e+f}{2}+\frac{b+c}{2}\frac{e+f}{2}+\frac{b+c}{2}d). \end{aligned}$$

Note that the first two parentheses of (5) are equal. Simple algebraic manipulations show that their sum is equal to the sum of the first two parentheses in (4). Similarly, the last four parentheses of (5) are equal and their sum is equal to the sum of the corresponding parentheses in (4). Hence, $E[S] - E[S'] = 0$. □

7 Summary

Maintaining histograms to approximate frequency distributions in relations is a common technique used by database systems to limit the errors in the estimates of query optimizers. In this paper, we have studied histograms and how they reduce errors when the result size of a tree join query reaches some extreme. We have focused on the class of serial histograms, which was previously shown to be optimal for a restricted type of tree join queries, and have generalized these results to include arbitrary such queries. Specifically, we have demonstrated that serial histograms are optimal for arbitrary tree equality-join queries when the query result size is maximized, whether or not the attribute independence assumption holds, and when the query result size is minimized and the attribute independence assumption holds. We have also shown that the expected error for any such query is always zero under all histograms, and thus argue that histograms should be chosen based on the reduction of the extreme-cases error, since reduction of the expected error is meaningless.

Several interesting and important questions on histogram optimality remain open. How many buckets should an optimal histogram have in order for the error to be within certain prespecified bounds? How is histogram optimality defined with respect to multiple queries and which histograms are to be preferred for a variety of queries? Is it reasonable to use histograms that are optimal in reducing the variance of the error instead of the worst-case error and what are the characteristics of such histograms? How do the results of this paper change when considering completely different types of queries (e.g., cyclic joins, non-equality joins, or selections) and different parameters of interest (e.g., operator cost or ranking of alternative access plans, which determines the final decision of the optimizer)? Many of these questions are part of our current and future work.

Acknowledgements: We are indebted to Y. C. Tay for several useful comments that improved many aspects of the paper.

References

[Chr83] S. Christodoulakis. Estimating block transfers and join sizes. In *Proc. of the 1983 ACM-SIGMOD Conference on the Management of Data*, pages 40–54, San Jose, CA, May 1983.

[Chr84] S. Christodoulakis. Implications of certain assumptions in database performance evaluation. *ACM TODS*, 9(2):163–186, June 1984.

[IC92] Y. Ioannidis and S. Christodoulakis. Optimal histograms for limiting worst-case error propagation in the estimates of query optimizers, 1992. To appear in ACM-TODS.

[Ioa93] Y. Ioannidis. Universality of serial histograms. Unpublished manuscript, June 1993.

[KK85] N. Kamel and R. King. A model of data distribution based on texture analysis. In

Proc. of the 1985 ACM-SIGMOD Conference on the Management of Data, pages 319–325, Austin, TX, May 1985.

[Koo80] R. P. Kooi. *The Optimization of Queries in Relational Databases.* PhD thesis, Case Western Reserve University, September 1980.

[MCS88] M. V. Mannino, P. Chu, and T. Sager. Statistical profile estimation in database systems. *ACM Computing Surveys*, 20(3):192–221, September 1988.

[MD88] M. Muralikrishna and D. J. DeWitt. Equi-depth histograms for estimating selectivity factors for multi-dimensional queries. In *Proc. of the 1988 ACM-SIGMOD Conference on the Management of Data*, pages 28–36, Chicago, IL, June 1988.

[MK85] B. Muthuswamy and L. Kerschberg. A ddsm for relational query optimization. In *Proc. ACM Annual Conference*, Denver, CO, October 1985.

[MO79a] A. W. Marshall and I. Olkin. *Inequalities: Theory of Majorization and Its Applications.* Academic Press, New York, NY, 1979.

[MO79b] T. H. Merrett and E. Otoo. Distribution models of relations. In *Proc. 5th Int. VLDB Conference*, pages 418–425, Rio de Janeiro, Brazil, October 1979.

[PSC84] G. Piatetsky-Shapiro and C. Connell. Accurate estimation of the number of tuples satisfying a condition. In *Proc. 1984 ACM-SIGMOD Conference on the Management of Data*, pages 256–276, Boston, MA, June 1984.

[SAC+79] P. G. Selinger, M. M. Astrahan, D. D. Chamberlin, R. A. Lorie, and T. G. Price. Access path selection in a relational database management system. In *Proceedings of the ACM SIGMOD Int. Symposium on Management of Data*, pages 23–34, Boston, MA, June 1979.

[Sch64] B. Schwarz. Rearrangements of square matrices with nonnegative elements. *Duke Mathematical Journal*, 31:45–62, 1964.

[Zip49] G. K. Zipf. *Human Behavior and the Principle of Least Effort.* Addison-Wesley, Reading, MA, 1949.