ResearchGate

See discussions, stats, and author profiles for this publication at: http://www.researchgate.net/publication/220225465

Optimal Histograms for Limiting Worst-Case Error Propagation in the Size of Join Results.

ARTICLE *in* ACM TRANSACTIONS ON DATABASE SYSTEMS · DECEMBER 1993

Impact Factor: 0.68 · DOI: 10.1145/169725.169708 · Source: DBLP

CITATIONS
100

READS
26

2 AUTHORS:

Yannis Ioannidis
National and Kapodistrian University of Athens
235 PUBLICATIONS **6,769** CITATIONS

SEE PROFILE

Stavros Christodoulakis
Technical University of Crete
174 PUBLICATIONS **3,113** CITATIONS

SEE PROFILE

Available from: Stavros Christodoulakis
Retrieved on: 02 October 2015

Optimal Histograms for Limiting Worst-Case Error Propagation in the Size of Join Results

YANNIS E. IOANNIDIS
University of Wisconsin

and

STAVROS CHRISTODOULAKIS
Technical University of Crete

Many current relational database systems use some form of histograms to approximate the frequency distribution of values in the attributes of relations and on this basis 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 its subclass of *end-biased* histograms; the latter is of particular interest because such histograms are used in several database systems. We concentrate on equality join queries without function symbols where each relation is joined on the same attribute(s) for all joins in which it participates. Join queries of this restricted type are called *t-clique* queries. We show that the optimal histogram for reducing the worst-case error in the result size of such a query is always serial. For queries with one join and no function symbols (all of which are vacuously t-clique queries), we present results on finding the optimal serial histogram and the optimal end-biased histogram based on the query characteristics and the frequency distributions of values in the join attributes of the query relations. Finally, we prove that for t-clique queries with a very large number of joins, *high-biased histograms* (which form a subclass of end-biased histograms) are always optimal. To construct a histogram for the join attribute(s) of a relation, the values in the attribute(s) must first be sorted based on their frequency and then assigned into buckets according to the optimality results above.

Categories and Subject Descriptors: G.1.0 [**Numerical Analysis**]: General—*error analysis*; H.1.1 [**Models and Principles**]: Systems and Information Theory; H.2.4 [**Database Management**]: Systems—*query processing*

General Terms: Performance, Theory

Additional Key Words and Phrases: Histograms, join size estimation, query optimization, vector majorization

This work was partially supported by the National Science Foundation under grants IRI-9113736 and IRI-9157368 (PYI Award) and by grants from DEC, HP, and AT & T.

Authors' addresses: Y. E. Ioannidis, Computer Sciences Dept., University of Wisconsin, Madison, WI 53706; S. Christodoukalis, Electronic and Computer Eng. Dept., Technical University of Crete, 73100 Chania, Crete, Greece.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.
© 1993 ACM 0362-5915/93/1200-0709 $3.50

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. This is due to two reasons: first, for each subset of values in an attribute, only aggregate information is captured in the histogram; second, as the database is updated, the information becomes obsolete if it is not appropriately updated as well. Hence, the query optimizer uses erroneous data to accomplish its task.

This would not be much of a problem if the desired estimates were derived by applying some simple functions on the erroneous statistics only once. This is not the case, however, for many complex queries that are processed as a sequence of many simpler operations, e.g., multi-join queries processed as a sequence of 2-way joins. In that case, the query optimizer must estimate various parameters of the intermediate results of the operations, and then use the obtained values to estimate the corresponding parameters of the results of subsequent operations. Even if the original errors in the statistics maintained by the database system are small, their transitive effect on estimates derived for parameters of the complete query may be devastating. Consequently, the decision of the query optimizer may be wrong since it is based on data with large errors. This phenomenon where the errors in the original system statistics affect the error in the derived estimates is called *error propagation* and is one of the main issues that challenge current query optimizer technology.

There are several parameters whose inaccurate estimation can lead a query optimizer to wrong decisions. Moreover, there are several operators that may be present in a query, and each one is affected by errors in its operands differently. In this paper, we concentrate on the relation size and on the join as the parameter and the operator of interest, respectively. This choice is motivated by their importance in query optimization and by their sensitivity to error propagation.

We investigate the optimality of histograms for limiting the error propagation in the estimates of the result sizes of a restricted type of join queries. Specifically, we study equality join queries without function symbols, where each relation is joined on the same attribute(s) for all joins in which it participates. The focus of our work is on histograms that accurately record the average frequency within each bucket. We identify a class of such histograms and show that, for the specific type of join queries studied, the optimal histogram for reducing the worst-case error in the size of such a query is always in that class. For 2-way join queries with no function symbols (all of which are vacuously of the query type studied), we present several

results on finding the optimal histogram in that class based on the query characteristics and the frequency distributions of values in the join attributes of the query relations. In that class, we also identify a specific type of histogram that is always optimal for very large queries. Since all these results discuss histogram optimality with respect to a specific query, we conclude with some possible heuristics on choosing a histogram that is potentially effective on both large and small queries. In practice, constructing a histogram is straightforward. The values in the join attribute(s) of interest must first be sorted based on their frequency, and then assigned into buckets according to the criteria dictated by the optimality goals of the system as specified by the formal results or heuristics presented in this paper.

We are aware of no work in the area of error propagation in the context of database query optimization other than our own [4, 5]. There is an extensive literature on deriving good estimates for the parameters of the result of database operations, which has been surveyed by Mannino et al. [10]. This is not the case, however, with the effect of the errors in these estimates on the error of a sequence of such operations. The folklore has been that errors propagate exponentially, and therefore, beyond a certain point, computed estimates are unreliable—but the problem has essentially been ignored. The primary reason has been the low complexity of the queries that current systems have to face. But as the query complexity increases in future database applications, this can no longer be the case. In hindsight, however, it becomes apparent that a better understanding of error propagation is needed even for the currently common, low complexity queries, where errors can grow enough to cause erroneous decisions by the optimizers [3, 8, 9, 17].

Although used in many systems, the formal properties of histograms have not been studied extensively. In addition, 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 Connell [15] dealt with the effect of histograms on reducing the error for selection queries. 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 [13] extended this study to include multidimensional histograms that are appropriate for multiattribute selection queries. The details of those studies and the assumptions under which the above statement holds are very different from the foundations of our work. Several other researchers have dealt with "variable-width" histograms for selection queries, where the buckets are chosen based on various criteria [6, 12, 14]. The survey by Mannino et al. [10] contains various references to work on choosing the appropriate number of buckets in a histogram for sufficient error reduction in the area of statistics. That work deals primarily with selections as well. Histograms for single-join queries have been minimally studied, and then again without

emphasis on optimality [1, 7, 14]. Our work is different from all the above in that it deals with arbitrarily large join queries for the most part and discusses properties of histograms that are optimal for such queries.

This paper is organized as follows. Section 2 introduces some notation for the study of error propagation and states the assumptions made in this paper. It also gives some of the results derived from mathematics (majorization theory) that are used throughout the paper. Section 3 contains the basic definitions on histograms and some of their fundamental properties. Section 4 identifies a characteristic property of all histograms that are optimal for some query. The class of histograms that have this property is studied further in subsequent sections. In Section 5, criteria are provided for identifying the optimal histogram within that class for queries with one join and no function symbols. Similar results are also obtained for a special histogram subclass of interest. In Section 6, the properties of histograms for queries with very large number of joins are studied and the class of asymptotically optimal histograms (as the number of joins in the query tends to infinity) is identified. Section 7 makes some general recommendations on how to choose histograms that limit the worst-case error propagation based on the results of the previous sections. Finally, Section 8 summarizes our results and gives directions for future work.

2. MATHEMATICAL FOUNDATIONS AND PROBLEM FORMULATION

2.1 Majorization Theory

We present some important results from the mathematical theory of majorization, which will prove to be useful in studying the effect of histograms on limiting the error propagation. They are all taken from Marshall and Olkin [11]. In what follows, an M-vector $\underline{a}$ whose components are a_i, $1 \leq i \leq M$, is denoted by $\underline{a} = \langle a_1, \ldots, a_M \rangle$ or by $\underline{a} = \langle a_i \rangle$. The components of all vectors are nonnegative reals. A vector $\underline{a}$ is *nonincreasing* when $\forall 1 \leq i < M$, the inequality $a_i \geq a_{i+1}$ holds. Finally, for two vectors $\underline{a}$ and $\underline{b}$, their *inner product* is defined as

$$\underline{a} * \underline{b} = \sum_{i=1}^{M} a_i b_i.$$

This may also be generalized for an arbitrary number N of vectors $\underline{a}^{(j)}$, $1 \leq j \leq N$, whose *inner product* is defined as

$$*\left(\underline{a}^{(1)}, \ldots, \underline{a}^{(N)}\right) = \sum_{i=1}^{M} \prod_{j=1}^{N} a_i^{(j)}.$$

We have taken the liberty to use $*$ both in the infix notations as a binary operator on vectors and in the functional notation as an N-ary operator on vectors for arbitrary N. The relationship between the two uses is straightforward: $\underline{a} * \underline{b} = *(\underline{a}, \underline{b})$.

Definition 2.1. For two M-vectors $\underline{a} = \langle a_\iota \rangle$ and $\underline{b} = \langle b_\iota \rangle$ with nonnegative components, $\underline{a}$ *majorizes* $\underline{b}$ if

$$\sum_{\iota=1}^{K} a_\iota \geq \sum_{\iota=1}^{K} b_\iota, \qquad \forall 1 \leq K < M$$

$$\sum_{\iota=1}^{M} a_\iota = \sum_{\iota=1}^{M} b_\iota.$$

There are several important inequalities that can be derived based on the majorization property. The most significant one for this paper is expressed by the following theorem.

THEOREM 2.1. *If* $\underline{x}$ *is a nonincreasing vector and* $\underline{a}$ *majorizes* $\underline{b}$, *then* $\underline{a} * \underline{x} \geq \underline{b} * \underline{x}$.

Example 2.1. As an example of the above theorem, consider the vectors $\underline{x} = \langle 3, 2, 1 \rangle$, $\underline{a} = \langle 10, 5, 1 \rangle$, and $\underline{b} = \langle 1, 5, 10 \rangle$. One can easily verify that the premises of Theorem 2.1 are satisfied. The same holds for the conclusion of the theorem, since $\underline{a} * \underline{x} = 42$, whereas $\underline{b} * \underline{x} = 23$.

The above theorem can be extended to inner products of multiple vectors.

THEOREM 2.2. *If for all* $1 \leq j \leq N$, $\underline{a}^{(J)}$ *majorizes* $\underline{b}^{(J)}$, *then the inequality* $*(\underline{a}^{(1)}, \ldots, \underline{a}^{(N)}) \geq *(\underline{b}^{(1)}, \ldots, \underline{b}^{(N)})$ *holds.*

2.2 Problem Formulation

In this paper, we use the term *join* as an operator that combines tuples of two relations based on some condition satisfied by the tuples. In general, such a condition will be a conjunction of simple relationships, each relationship involving an attribute of the first relation and an attribute of the second relation. For example, if R_0, R_1, R_2 are relation names and a, b are attribute names of these relations, a query whose qualification is

$$(R_0.a = R_1.a \textbf{ and } R_0.b = R_1.b) \tag{1}$$

is a query with one join, whereas a query whose qualification is

$$(R_0.a = R_1.a) \textbf{ and } (R_0.b = R_2.b) \tag{2}$$

is a query with two joins. In both qualifications above, the condition inside each parenthesis is a join.

Consider a tree query of N joins in which relations $R_0, \ldots, R_N$ participate. To avoid potential confusion with the multiple use of the term "value," we refer to the values of the join attributes of these relations as the *join elements*. In this study, we make the following assumptions about the form of

the query:

All joins are equality joins with no function symbols. (A1)

Each relation is joined on exactly the same attribute(s) for all joins in which it participates. That is, even if a relation is joined with multiple other relations, each join is always on the same attribute(s). (A2)

Queries that satisfy the above assumptions are called *t-clique queries*, since their query graph would be a clique due to the *t*ransitivity of equality. For example, a query with qualification (1) is a t-clique query, but a query with qualification (2) is not, because R_0 participates in the join with R_1 with attribute a and in the join with R_2 with attribute b, i.e., (A2) is violated. Similarly, a query with qualification

$$(R_0.a = R_1.a \textbf{ and } R_0.b = R_1.b) \textbf{ and } (R_0.a = R_2.a \textbf{ and } R_0.b = R_2.b) \quad (3)$$

is a t-clique query, because R_0 is joined with both R_1 and R_2 based on both attributes a and b. Finally, a query with qualification

$$(R_0.a + R_0.b = R_1.a + R_1.b)$$

is not a t-clique query, because the join involves the arithmetic function +, i.e., (A1) is violated. Note that all queries with one join (often referred to as 2-way join queries because they involve two relations) satisfy (A2) by definition. Hence, for 2-way equality join queries with no function symbols, the results presented in this paper are completely general.

An obvious implication of (A2) is that all relations participate in joins with the same number of attributes of compatible types. We view this common set of attributes as a single attribute of potentially tuple form and refer to it as *the join attribute* of the query. Based on the above, the join elements, i.e., the values of the join attribute, may be of tuple form as well. For example, in both (1) and (3), the combination of attributes a, b is considered as a single attribute whose values are tuples (pairs) of atomic values. The *join domain* $\mathcal{D}$ of a t-clique query is the set of all join elements that could potentially appear in the join attribute of any relation in the query. All forthcoming results are independent of the number of attributes of each relation that appear in the query, i.e., independent of the form of the join attribute mentioned above.

We assume some arbitrary numbering of the elements in the join domain, so that referring to the i-th join element is meaningful. The following database parameters are of interest:

- M The size of the join domain.
- t_{ij} The number of tuples in R_j whose join attribute contains the i-th join element of the join domain, $1 \leq i \leq M$, $0 \leq j \leq N$. This is called the *frequency* of the i-th join element of the join domain in R_j.
- S The size of the result relation of the query.

For simplicity, given the bijection between $\mathcal{D}$ and the set $\{1, 2, \ldots, M\}$, we treat the latter as the join domain itself, i.e., $\mathcal{D} = \{1, 2, \ldots, M\}$. Whenever we need to distinguish between the two, we use the term "actual join domain" to refer to the former. We should point out, however, that the bijection is arbitrary and thus not necessarily order preserving. For example, if the actual join domain is a finite subset of the integers, reals, or bounded-length character strings, the i-th join element based on the natural ordering of the join domain is not necessarily associated with i. For each relation R_j, $0 \leq j \leq N$, the vector $\underline{t}_j = \langle t_{1j}, \ldots, t_{Mj} \rangle$ is called the *frequency distribution* in R_j. Occasionally, it is also useful to treat all the frequencies in $\underline{t}_j$ as a collection, ignoring the join element with which each frequency is associated. That collection is in general a multiset (i.e., it may contain duplicates), is called the *frequency* set of R_j, and is denoted by M_j. For example, if $\underline{t}_0 = \langle 10, 3, 7, 3 \rangle$ and $\underline{t}_1 = \langle 3, 3, 10, 7 \rangle$, then $M_0 = M_1 = \{10, 7, 3, 3\}$.

Clearly, for t-clique queries, the above parameters are related with the following formula:

$$S = *(\underline{t}_0, \ldots, \underline{t}_N) = \sum_{i=1}^{M} \prod_{j=0}^{N} t_{ij}. \tag{4}$$

That is, the size of S of the result of a t-clique join query is equal to the inner product of the frequency distributions of the participating relations. Theorem 2.2 can be directly applied on (4) to derive the following:

THEOREM 2.2. *Consider a t-clique query Q with relations R_j, $0 \leq j \leq N$. Let the frequency sets M_j, $0 \leq j \leq N$, be given for all relations. The result size of Q is maximized when, for all $0 \leq j \leq N$, the corresponding frequency vector $\underline{t}_j$ is a nonincreasing vector.*

PROOF. Consider any frequency set M_j, $0 \leq j \leq N$, and let $\underline{t}_j, \underline{t}'_j$ be two potential frequency vectors for R_j (whose components are the elements of M_j). If t_j is a nonincreasing vector, then clearly, $\underline{t}_j$ majorizes $\underline{t}'_j$. Hence, by Theorem 2.2 and formula (4), the result size of Q is maximized when the frequency vectors of all relations are nonincreasing. □

Most often database systems have inaccurate knowledge of the frequency distributions in the query relations. Therefore, the estimate that they derive for the size S is inaccurate as well, and this affects the decisions of their query optimizers.

Definition 2.2. Suppose that a certain quantity has a definite value A whereas the database system approximates it by the value A^e. The difference $A - A^e$ is the *exact error* and the fraction $(A - A^e)/A^e$ is the *relative error* in the approximate value A^e.

One could have used the fraction $(A - A^e)/A$ in defining the relative error in A^e, instead of following Definition 2.2. Our choice was motivated by a desire to measure error based on the value that the database system knows, which is A^e. In addition, there is a very simple relationship between the two types of relative error, which can be used to derive the value of one given the

value of the other. Specifically, if $\epsilon_1 = (A - A^e)/A^e$ and $\epsilon_2 = (A - A^e)/A$, then the following holds:

$$1 + \epsilon_1 = \frac{A}{A^e} = \frac{1}{1 - \epsilon_2} \Rightarrow 1 + \frac{1}{\epsilon_1} = \frac{1}{\epsilon_2}.$$

Note that as ϵ_1 varies from -1 to ∞, ϵ_2 varies from $-\infty$ to 1. Also note that the values of the two types of relative error are very close to each other when they are very small (close to 0). Hence, when the database system achieves its goal of maintaining enough information so that it deals with small errors, the definition that is adopted for the relevant error plays no significant role. For all the above reasons, all the results on relative error in this paper are derived based on Definition 2.2.

For any quantity of interest, the potentially erroneous value used by the system is denoted by the same symbol as the correct value with an additional superscript "e." For example, the approximation of the frequency distribution is denoted by $\underline{t}_J^e = \langle t_{1J}^e, \ldots, t_{M_J}^e \rangle$ and the corresponding estimated result size is denoted by S^e. In the sequel, we concentrate on relative errors. If no confusion arises, we occasionally use the term "error" alone, the intended meaning being "relative error."

For a given collection of $\underline{t}_j^e$, $0 \leq j \leq N$, let $D = (S/S^e) - 1$ be the corresponding relative error in the estimated size of the query result. By (4), this implies that for t-clique queries

$$1 + D = \frac{\sum_{i=1}^{M} \prod_{j=0}^{N} t_{ij}}{\sum_{i=1}^{M} \prod_{j=0}^{N} t_{ij}^e}. \tag{5}$$

Note that for any fixed value of the estimated result size S^e, Theorem 2.3 and (5) imply that the error D is maximized when for all $0 \leq j \leq N$, $\underline{t}_j$ is a nonincreasing vector. Moreover, it has been shown that, under the uniform distribution assumption, the error grows exponentially with N [4, 5]. (This holds for many other approximations of the frequency distributions as well.) The focus of our attention is on reducing D for databases where such worst-case behavior is exhibited, i.e., where all frequency vectors are nonincreasing. The particular method that we study is maintaining appropriately chosen histograms on the frequency distributions.

2.3 An Example

Formula (5) holds for arbitrary frequency distributions. To obtain a better feeling for the magnitude of the error propagation, we apply (5) to a specific database instance, which will also be our running example for the entire paper. In particular, we examine the case where the actual frequency distributions are Zipf [2, 18]. The main characteristic of the Zipf distribution is that it assigns high frequencies to few join elements and low frequencies to most join elements. Thus, this example deals with a quite common special case, since the above is claimed to be a characteristic of the distribution in many databases.

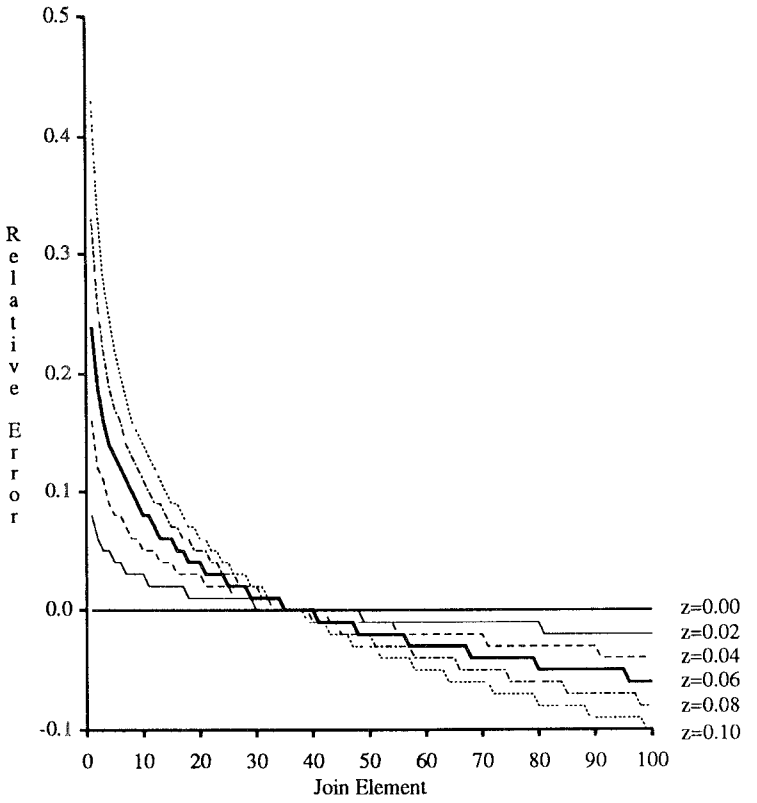

Fig. 1. Zipf frequency distribution.

Assume that all relations in the database are equal to each other and the frequency distribution is Zipf, i.e., for all j,

$$t_{ij} = T_j \frac{1/i^z}{\sum_{l=1}^{M} 1/i^z} \qquad \text{for all} \quad 1 \le i \le M. \tag{6}$$

In (6), T_j is the size of R_j in tuples, and we assume that it is equal to 10000 for all relations. Furthermore, we assume that the join domain contains $M = 100$ join elements. Figure 1 is a graphical representation of (6) for $z = 0.0, 0.02, \ldots, 0.1$. One can see that the deviation from the uniform distribution increases with z, but it is not very dramatic for the range depicted.

Suppose that the database system uses the Zipf distribution with $z = 0$ (uniform) as the approximation to the actual distribution. Figure 2 is a graphical representation of equation (5) for that case. Specifically, the relative error in the query result size is shown as a function of the number of joins for various values of z. The observed results are rather discouraging. Even small errors in the individual relations propagate in the query result growing at an exponential rate and generating a final error that very quickly

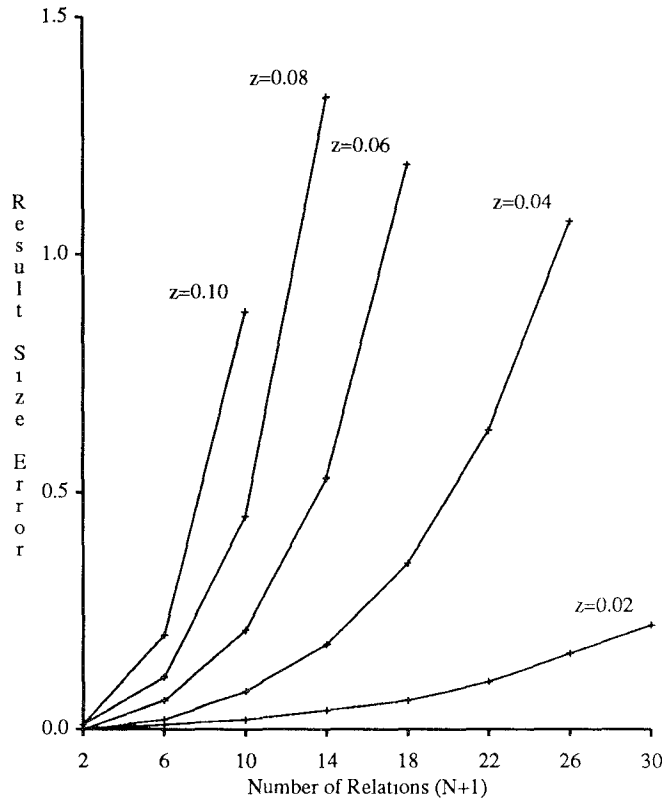

Fig. 2. Join result size error for Zipf distributions under uniform approximation

becomes intolerable. Note that by Theorem 2.3, this situation produces a worst-case error, since all frequency distributions are nonincreasing vectors.

3. BASIC DEFINITIONS AND PROPERTIES OF HISTOGRAMS

Among commercial systems today, maintaining *histograms* is a very common approach to approximating frequency distributions [17]. In histograms, the join domain is partitioned into *buckets*, and a uniform distribution is assumed within each bucket. Hence, based on our convention about the join domain, a bucket is a subset of $\mathcal{D} = \{1, 2, \ldots, M\}$. The approximate frequency distribution captured by a histogram is called the *histogram distribution*. We should emphasize the fact that buckets that do not represent a contiguous range in the join domain are perfectly valid, e.g., bucket $\{1, 3\}$. The join domain as defined provides no indication of how the join elements should be grouped in buckets. The numbering of the join elements based on the bijection between the actual join domain and $\{1, 2, \ldots, M\}$ (Section 2.2) has been arbitrary and does not reflect, for example, a natural value-based ordering of the join domain. Also note that maintaining the necessary information for the uniform distribution assumption over the entire join domain is equivalent to

maintaining a histogram with a single bucket. Such a histogram is called *trivial*.

In this study, we assume that the histograms maintained by the system have an accurate record of the average frequency within each bucket, i.e., for any bucket b in the histogram of R_j, $0 \le j \le N$, for all $1 \le i \le M$, $i \in b \Rightarrow t_{ij}^e = \sum_{k \in b} t_{kj}/|b|$. Thus, we concentrate on errors that arise from aggregating the frequency distributions and not on ones that arise from delaying the propagation of database updates to the histogram. The reason is that the former are the only types of error that can be controlled by an appropriate choice of histograms. The latter can only be controlled by an appropriate schedule of database update propagation to histograms, the investigation of which is beyond the scope of this paper.

The following lemma establishes a majorization relationship between a frequency distribution and any corresponding histogram distribution.

LEMMA 3.1. *If $\underline{t}$ is a nonincreasing frequency distribution and $\underline{h}$ any corresponding histogram distribution, then $\underline{t}$ majorizes $\underline{h}$.*

PROOF. Within each bucket of the histogram, t_i is nonincreasing, whereas t_i^e remains constant. Hence, within each bucket $\langle t_i \rangle$ majorizes $\langle t_i^e \rangle$. Combining the implications of this for all buckets yields the lemma. □

In general, each histogram reduces the join result size error differently. We attempt to identify the ones that are in some sense optimal. Unfortunately, we are aware of no useful result that is generally applicable. Hence, we concentrate on identifying optimal histograms for reducing the worst-case error, i.e., when for all $1 \le k \le M$, the k-th largest value in the frequency sets of all relations is associated with the same join element. This association represents the worst case because, for any given collection of frequency sets, it produces the maximum value for the join result size among all possible associations of frequencies to join elements (Theorem 2.3), which corresponds to the maximum error given any fixed value of the estimated result size. By Theorems 2.2 and 2.3 and Lemma 3.1, the approximate join result size is never larger than the maximum possible actual size. Therefore, the optimal histogram maximizes the approximate size.

For simplicity in the presentation and without loss of generality, we choose the bijection between the actual join domain and the set $\{1, 2, \ldots, M\}$ (Section 2.2) so that it preserves the frequency-based ordering of the join elements. Thus, the k-th most frequent join element is associated with k. Because of the decision to use $\{1, 2, \ldots, M\}$ as the join domain itself, the above can be restated simply as the following convention, which holds for the entire paper:

The frequency distributions of all relations are nonincreasing.

Recall that, in general, the frequency-based ordering captured by the distributions following the above convention is completely unrelated to any value-based ordering of the actual join domain.

Example 3.1. To illustrate the above definitions, conventions, and assumptions on histograms, consider the "canonical" EMP relation and focus on

Table I. Frequencies of Department Names in the EMP Relation

Department Name	Number of Employees	Frequency-based Rank
candy	10	4
jewelry	30	2
shoe	20	3
toy	40	1

the dept attribute, which contains the name of the department of each employee. For simplicity, assume that there are four different departments and that the frequency of each department in the EMP relation is given in the following table.

The assumption that the frequency distributions of all relations are nonincreasing implies the following: first, the frequency-based ordering of departments is toy > jewelry > shoe > candy for all relations that have the dept attribute; second, the bijection between the actual join domain and $\{1, 2, 3, 4\}$, which is used as the join domain by convention, associates each department to the frequency-based rank mentioned in Table I. Note that this frequency-based ordering is completely unrelated to the alphabetical ordering of departments based on their names. Figure 3(a) is a graphical representation of the nonincreasing frequency distribution of department names in EMP. In the x-axis, both the actual and the conventional join domains are given. Based on the usual convention [9, 15], if one were to build a histogram on the dept attribute, buckets would be formed by departments that are close in the alphabetical ordering of their names. An example of such a histogram with two 2-element buckets is shown in Figures 3(b) and 3(c). In the former, we show the original distribution and which join elements (or equivalently frequencies) are placed in which bucket. In the latter, we show the actual histogram distribution that is the result of averaging the frequencies in each bucket. Note that the buckets $b_1 = \{1, 3\}$ and $b_2 = \{2, 4\}$ are not contiguous ranges within $\{1, 2, 3, 4\}$. As another example, consider the histogram with two 2-element buckets that is shown in Figures 3(d) and 3(e), again depicted in the two ways discussed for the first histogram. In this case, the buckets $b_1 = \{1, 2\}$ and $b_2 = \{3, 4\}$ are contiguous ranges within $\{1, 2, 3, 4\}$, but do not group departments based on their names. In principle, all possible histograms are equally valid for the purposes of this paper, including the two above.

Based on the above, histogram optimality is defined as follows.

Definition 3.1. Consider a query Q with relations R_j, $0 \leq j \leq N$, associated with nonincreasing frequency distributions. For each relation R_j, let $\mathscr{H}_j$ be a collection of histograms of interest. The $(N + 1)$-vector $\langle H_j \rangle$, where $H_j \in \mathscr{H}_j$, $0 \leq j \leq N$, is an *optimal* histogram vector for Q within $\langle \mathscr{H}_j \rangle$, if the approximate result size of Q that it generates is greater than or equal to the approximate result size of Q that any other such histogram vector generates.

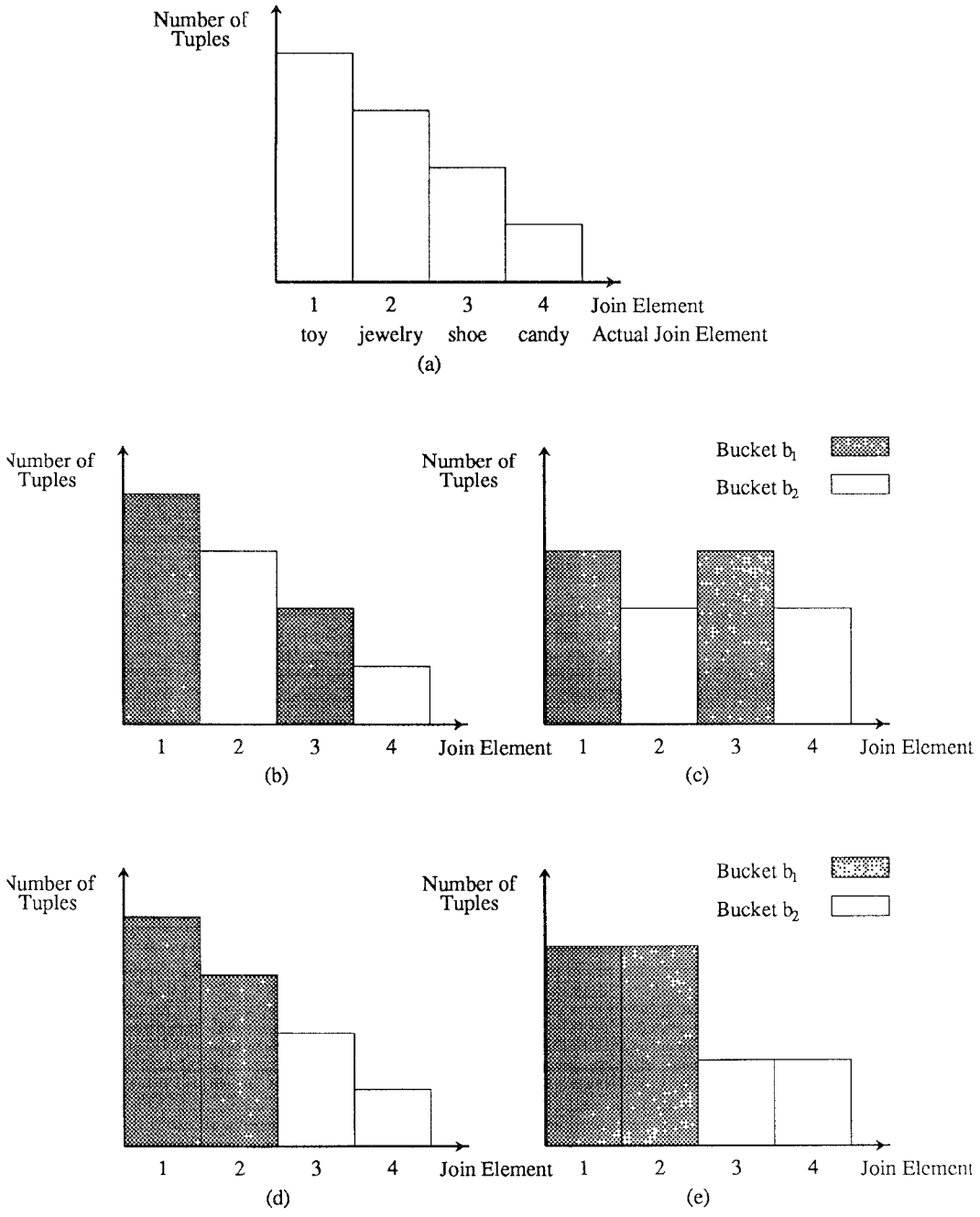

Fig. 3. Frequency distribution and two histograms of department names in EMP.

Note that optimality is defined per query and per collection of frequency distributions, and for the histograms of all relations together. The reason is that the optimal histograms differ for different queries and for different frequency distributions. In reality, one would like to identify one histogram per relation that is optimal for all queries and all frequency distributions of the remaining relations. Whenever possible, we present results of that form, but in general, we follow Definition 3.1 for optimality.

Before we proceed with the investigation of optimal histograms, we define several types of histograms and histogram buckets that play some significant roles in the rest of this paper.

Definition 3.2. A histogram bucket such that the frequencies associated with all the join elements in it are equal is called *univalued*. Any other bucket is called *multivalued*.

Note that for all join elements in a univalued bucket, the histogram captures their associated frequencies accurately.

Definition 3.3. A histogram with L univalued buckets and one multivalued bucket is called *biased*. If the L univaled buckets correspond to the join elements with the L_1 highest distinct frequencies and the L_2 lowest distinct frequencies for some L_1 and L_2 such that $L = L_1 + L_2$, then it is called *end-biased*. If $L_1 = L$ and $L_2 = 0$, then it is called *high-biased*. If $L_1 = 0$ and $L_2 = L$, then it called *low-biased*.

We should point out that high-biased histograms are being used by some current systems for approximating the frequency distributions of relations [Sel89].

Definition 3.4. A histogram is called *serial with respect* to its buckets b_1 and b_2, if either $\forall i \in b_1, k \in b_2$, the inequality $i > k$ holds, or $\forall i \in b_1, k \in b_2$, the inequality $i < k$ holds. It is called *serial* if it is serial with respect to all pairs of its buckets.

For example, the histogram of Figures 3(d)–(e) is serial, while that of Figures 3(b)–(c) is not. Note that because we are dealing with nonincreasing frequency distributions, the inequality $i > k$ above is equivalent to the reverse inequality between the corresponding frequencies, i.e., $t_i \leq t_k$. Also, note that end-biased histograms are serial.

4. CLASS OF OPTIMAL HISTOGRAMS

In searching for the optimal histograms, the following general result is useful.

LEMMA 4.1. *Consider a t-clique query with relations R_j, $0 \leq j \leq N$, and assume that the histogram distributions for R_j, $1 \leq j \leq N$, are nonincreasing. For two different histograms G and H for R_0, if the histogram distribution of G majorizes that of H, then the error under G is no higher than the error under H.*

PROOF. This is a straightforward consequence of Theorem 2.1, which implies that S^c obtains a larger (or equal) value under G than under H. Since we are dealing with the case that produces the highest possible actual size (nonincreasing frequency distributions), the above implies that the error under G must be less than or equal to the error under H. □

Clearly, the overall optimal histogram is always the one with singleton buckets, one for each join element. To avoid such vacuous arguments and to make fair comparisons, we define several interesting spaces of histograms, where each space contains histograms that are equivalent with respect to the amount of information that they maintain. Then we identify the optimal histograms within each such space.

Definition 4.1. The *bucket template* $\mathscr{B}$ of a histogram is a pair $\mathscr{B} = \langle \beta, B \rangle$, where B is a set of integers that denote the cardinalities of the buckets in the histogram and whose sum is equal to M, and $\beta = |B|$ is the *histogram size*, i.e., the number of buckets in the histogram.

As an example, the bucket template of both histograms of Example 3.1 is $\langle 2,\{2,2\} \rangle$, since they have two buckets, each one of which has two join elements.

Definition 4.2. For any bucket template $\mathscr{B}$, the space $\mathscr{H}_{\mathscr{B}}$ is the set of histograms that conform to $\mathscr{B}$ and the space $\mathscr{H}_\beta$ is the set of histograms that have β buckets. Also, the space $\mathscr{V}_\beta$ is the set of biased histograms that have β buckets.

Note that $\mathscr{H}_{\mathscr{B}}$ is closed under join element exchange between buckets. That is, given $H \in \mathscr{H}_{\mathscr{B}}$ and two of its buckets b_1 and b_2, consider any histogram H' that is identical to H in all remaining buckets, and has b_1 and b_2 replaced by b'_1 and b'_2 such that $|b'_1| = |b_1|$, $|b'_2| = |b_2|$, and $b'_1 \cup b'_2 = b_1 \cup b_2$. Then $H' \in \mathscr{H}_{\mathscr{B}}$. Also note that $\mathscr{H}_{\mathscr{B}} \subseteq \mathscr{H}_\beta$, since $\mathscr{H}_{\mathscr{B}}$ requires that not only the number but also the size of the buckets be the same. Given the above, the following lemma and the subsequent theorem shed some light on histogram majorization within $\mathscr{H}_\beta$. (Optimality results within $\mathscr{H}_\beta$ will be given in the next section.)

LEMMA 4.2. *Consider a bucket template $\mathscr{B}$ and a histogram $H \in \mathscr{H}_{\mathscr{B}}$ with two buckets b_s and b_t, and let s and t be the average of the corresponding join element frequencies, respectively. Consider a histogram $G \in \mathscr{H}_{\mathscr{B}}$ that is serial with respect to b_s and b_t and is constructed from H by exchanging elements of b_s and b_t and leaving all other buckets unchanged. Let the corresponding frequency averages for b_s and b_t be s_1 and t_1, respectively. Without loss of generality, assume that in G, $\forall i \in b_s$, $k \in b_t$, the inequality $i < k$ holds. If $s_1 \geq s \geq t_1$ and $s_1 \geq t \geq t_1$, then G majorizes H.*[1]

PROOF. Obviously, in what follows, we can ignore all other buckets and concentrate on b_s and b_t. Thus, without loss of generality, we assume that b_s

[1]For any histogram H, three of the four inequalities always hold.

and b_t contain the first $|b_s| + |b_t|$ elements of the join domain. Assume that the frequencies in H and G are denoted by h_i and g_i, respectively. Then the following three facts can be derived. (a) Since the set of elements remains unchanged in the two histograms, the following holds: $\sum_{i=1}^{|b_s|+|b_t|} h_i = \sum_{i=1}^{|b_s|+|b_t|} g_i$. (b) If $K \leq |b_s|$, then $\sum_{i=1}^{K} g_i \geq \sum_{i=1}^{K} h_i$, since for all $1 \leq i \leq K$, $g_i = s_1$ and h_i either takes the value s or the value t, both of which are no greater than s_1 by the premises of the lemma. (c) If $|b_s| < K < |b_s| + |b_t|$, then the following holds:

$$\sum_{i=1}^{K} g_i = \sum_{i=1}^{|b_s|+|b_t|} g_i - \sum_{i=K+1}^{|b_s|+|b_t|} g_i \geq \sum_{i=1}^{|b_s|+|b_t|} h_i - \sum_{i=K+1}^{|b_s|+|b_t|} h_i = \sum_{i=1}^{K} h_i.$$

In the above, the inequality is due to two observations: the first sums of the expressions on its two sides are equal (fact (a)); for the second sums, for all $K + 1 \leq i \leq |b_s| + |b_t|$, $g_i = t_1$ and h_i either takes the value s or the value t, both of which are no less than t_1 by the premises of the lemma.

By Definition 2.1, points (a)–(c) prove that G majorizes H. □

THEOREM 4.1. *For any bucket template $\mathcal{B}$ and any histogram $H \in \mathcal{H}_{\mathcal{B}}$, there is a serial histogram in $\mathcal{H}_{\mathcal{B}}$ that majorizes it.*

PROOF. Consider two arbitrary buckets b_s and b_t in H. We first show that there is a histogram in $\mathcal{H}_{\mathcal{B}}$ that is serial with respect to b_s and b_t that majorizes H. Specifically, consider histograms that can be constructed from H by exchanging elements of b_s and b_t and leaving all other buckets unchanged. Again, in what follows, we ignore all other buckets and concentrate on b_s and b_t. We distinguish two cases.

Case 1. $|b_s| \neq |b_t|$. In this case, there are only two histograms in $\mathcal{H}_{\mathcal{B}}$ that can be constructed as above and be serial with respect to b_s and b_t: one where $\forall i \in b_s$, $\forall k \in b_t$, $i < k$ (denoted by H_1) and one where $\forall i \in b_s$, $\forall k \in b_t$, $i > k$ (denoted by H_2). We prove that if one of them does not majorize H, then the other one does. Without loss of generality, assume that H_1 does not majorize H and that $|b_s| < |b_t|$. We use the following notation:

s	The average of the frequencies in bucket b_s in H.
t	The average of the frequencies in bucket b_t in H.
s_j	The average of the frequencies in bucket b_s in H_j, $j = 1, 2$.
t_j	The average of the frequencies in bucket b_t in H_j, $j = 1, 2$.

Figure 4 can be used to illustrate the situation by showing on the original frequency distribution which join elements are placed in which bucket.

By the manner in which the b_s and b_t buckets have been constructed in the various histograms, the following inequalities can be derived:

$$s_1 \geq s \geq s_2 \tag{7}$$

$$s_1 \geq t_2 \geq t \geq t_1 \geq s_2. \tag{8}$$

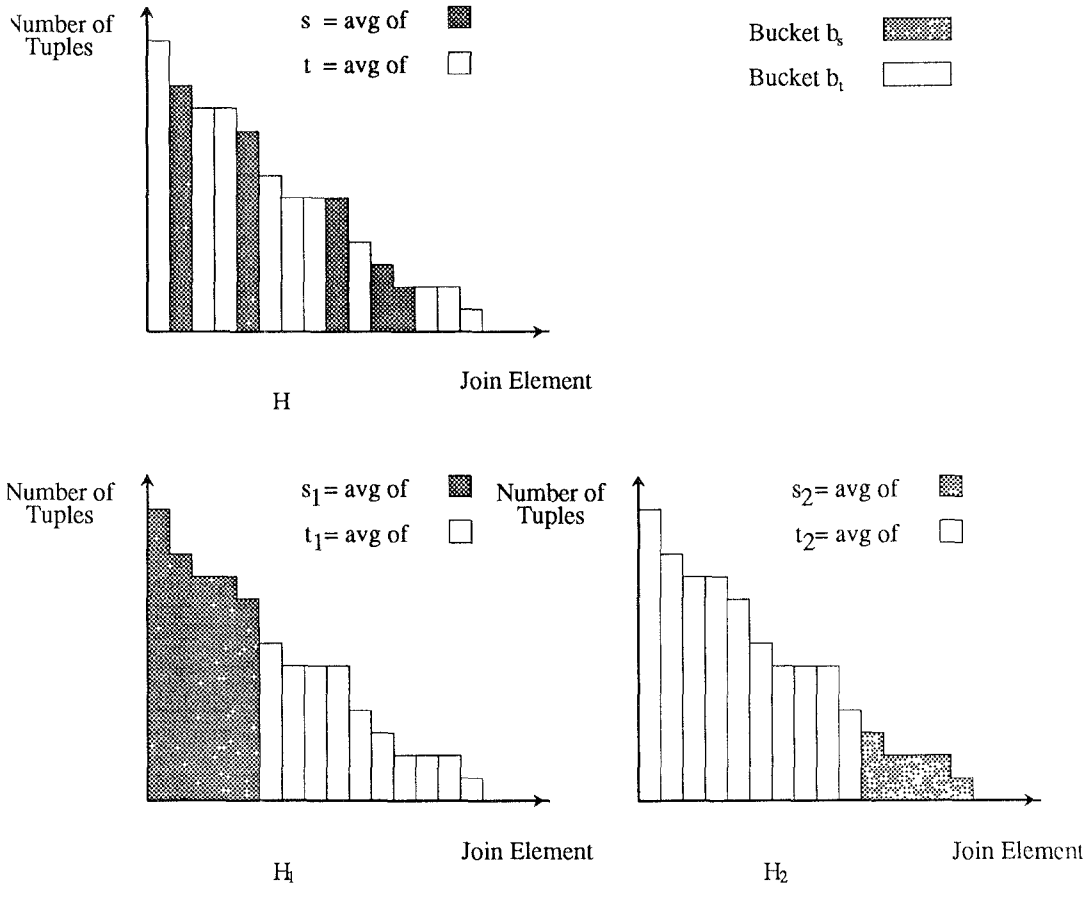

Fig. 4. Serializations of two buckets by element exchange.

The inequalities in (7) are derived from the fact that all three s's are averages of the same number $|b_s|$ of frequencies, with s_1 being the average of the $|b_s|$ largest frequencies among those in the two buckets, and s_2 being the average of the $|b_s|$ smallest such frequencies. The inequalities in (8) among the t's are derived similarly. Finally, the outermost inequalities $s_1 \geq t_2$ and $t_1 \geq s_2$ are due to the fact that b_s is smaller (has fewer join elements) than b_t.

The fact that H_1 does not majorize H implies that $t_1 > s$. Given the inequalities in (7) and (8), the above is derived as the contrapositive of Lemma 4.2. That inequality combined with (7) and (8) yields the following: $t_2 \geq t \geq s_2$ and $t_2 \geq s \geq s_2$. By Lemma 4.2, these imply that H_2 majorizes H.

Case 2. $|b_s| = |b_t|$. In this case, there is a unique histogram in $\mathscr{H}_{cs}$ (denoted by H_1) that can be constructed as above and be serial with respect to b_s and b_t. (Whether it is b_s that ends up with the high frequencies or b_t is immaterial, since both buckets contain the same number of elements.) We

prove that H_1 majorizes H, using the same notation as in case 1. Because $|b_s| = |b_t|$, both s and t can be no greater than s_1, since the latter is the average of the largest $|b_s|$ frequencies in the two buckets. Similarly, both s and t can be no less than t_1, since the latter is the average of the smallest $|b_s|$ frequencies in the two buckets. Therefore, Lemma 4.2 is directly applicable and implies that H_1 majorizes H.

The above concludes the proof that for any pair of buckets in H, we can obtain a histogram in $\mathscr{H}_{\mathscr{B}}$ that is serial with respect to the two buckets and majorizes H. Repeatedly applying the above construction for all pairs of buckets in H will result in a serial histogram in $\mathscr{H}_{\mathscr{B}}$ with the desired properties. Since there is a finite number of such pairs, this process is guaranteed to complete. □

The above theorem can now be used to derive results on the optimality of histograms with respect to the join result size error.

THEOREM 4.2. *Consider a t-clique query Q with relations R_j, $0 \le j \le N$, and an $(N+1)$-vector of bucket templates $\langle \mathscr{B}_j \rangle$. There exists an optimal histogram vector for Q within $\langle \mathscr{H}_{\mathscr{B}_j} \rangle$ where all histograms in it are serial.*

PROOF. By Theorem 4.1, for any histogram $H_j \in \mathscr{H}_{\mathscr{B}_j}$ there is a serial histogram $G_j \in \mathscr{H}_{\mathscr{B}_j}$ that majorizes it. Since G_j is serial, the corresponding histogram distribution is nonincreasing. By applying Lemma 4.1 the theorem is proved. □

COROLLARY 4.1. *Consider a t-clique query Q with relations R_j, $0 \le j \le N$, and an $(N+1)$-vector of histogram sizes $\langle \beta_j \rangle$. There exists an optimal histogram vector for Q within $\langle \mathscr{U}_{\beta_j} \rangle$ where all (biased) histograms in it are end-biased.*

PROOF. Similar to the previous set of results. □

An interesting observation concerning Theorem 4.2 is that usually histograms are constructed in a way that each bucket stores join elements that belong in a certain range in the natural total order of the actual join domain. What the above theorem implies is that this traditional approach may be far from optimal for t-clique queries. Moreover, it indicates that histograms should be constructed so that join elements are grouped in buckets based on closeness in their corresponding frequencies.

Example 4.1. As an example of the importance of serial histograms, consider the worst-case error when joining two identical relations of 10000 tuples whose join domain contains 100 elements and whose frequency distributions are Zipf with $z = 0.2$ (Section 2.3). Assume that the histograms maintained for the two relations are identical as well. We have calculated the error generated when the histograms are trivial (i.e., uniform approximation) and for three other interesting types of histograms that have five buckets: (a)

Table II. Error in a 2-Way Join Query for Various Histograms

Histogram	Error
Trivial	4.64%
Nonserial	4.60%
Serial	1.10%
High-biased	2.15%

a nonserial histogram where the i-th bucket b_i, $1 \leq i \leq 5$, is equal to $b_i = \{5x + i | 0 \leq x \leq 19\}$; (b) the unique serial histogram with five buckets with twenty elements each; and (c) the unique high-biased histogram with five buckets (four univalued buckets contain the join elements with the four highest frequencies and one that contains the remaining join elements). Note that (b) and (c) conform to the same bucket template, i.e., they belong to $\mathscr{H}_{\mathscr{B}}$ for $\mathscr{B} = \langle 5, \{20, 20, 20, 20, 20\} \rangle$, whereas (b), (c), and (d) have the same number of buckets, i.e., they belong to $\mathscr{H}_5$. The corresponding results on the error based on (5) are shown in Table II.

As expected the serial and high-biased histograms are better than the trivial and the nonserial ones. There are two more interesting points to note, however. First, the nonserial and the trivial histograms generate almost identical errors, although the former maintains five times more information than the latter. Second, the high-biased histogram generates a larger error than the serial one. Hence, in this case, the relatively common practice of accurately maintaining the highest frequencies is far from optimal. □

Theorem 4.2 is the most important result of this section and states that the optimal histograms are serial. It does not offer, however, any indication as to which of the possibly many serial histograms is the optimal one in each case. Unfortunately, we are aware of no general result in that direction. The optimal serial histogram depends on the specific frequency distributions of the relations but also on the query size. That is, even for the same frequency distribution for all relations, the number of relations joined in the query significantly affects the optimal serial histogram. The intuition behind these dependencies is the following:

(i) The frequencies that are rarer should be known more accurately, i.e., the associated join elements should be placed in univalued buckets or at least buckets with few join elements that have similar frequencies. The reason is that these frequencies offer more information about the overall distribution. The following artificial example will help drive the point home. Consider the frequency distribution $\langle 10, 9, 8, 1 \rangle$. The frequency of the fourth join element is much lower than the other three frequencies, which are very close to each other. Thus, for small join queries, histogram H_1 with buckets $\{1, 2, 3\}$ and $\{4\}$ is more preferable than, say, histogram H_2 with buckets $\{1\}$ and $\{2, 3, 4\}$. The histogram distribution of H_1 is very

close to the actual one, $\langle 9, 9, 9, 1\rangle$ instead of $\langle 10, 9, 8, 1\rangle$, whereas the histogram distribution of H_2 is quite different, $\langle 10, 6, 6, 6\rangle$ instead of $\langle 10, 9, 8, 1\rangle$.

(ii) As the number of joins in queries grows, large frequencies become more dominant in the computation of the query result sizes and therefore should be known more accurately. The reason is that the number of tuples in the results associated with the most frequent join elements becomes a larger fraction of the overall result sizes as the number of joins in queries grows. Consider a relation having the frequency distribution of the example in (i) above. For a 2-way join query of the relation with itself, the first join element contributes 100 tuples to the result out of a total of 246 tuples, that is approximately 40%. On the other hand, for a 5-way join query of the relation with itself, the corresponding percentage is approximately 52%. Thus, H_2 becomes more competitive as the query size grows, and beyond a certain number of joins it becomes more preferable than H_1.

The results presented in the next section quantify (i) for 2-way join queries. The results presented in the subsequent section quantify (ii) for very large queries (in the limit). A general result that captures the precise balance between (i) and (ii) for arbitrary size queries, i.e., arbitrary values of N, still escapes our efforts.

5. OPTIMAL SERIAL HISTOGRAMS FOR 2-WAY JOIN QUERIES

As mentioned earlier, all 2-way equality join queries with no function symbols are t-clique queries. Hence, assumption (A2) does not restrict the generality of the results in this section. For the rest of the paper, we concentrate on serial histograms. Let $\mathscr{S}_\beta$ be the subset of $\mathscr{H}_\beta$ that contains the serial histograms that have β buckets. We discuss histogram optimality within $\mathscr{S}_\beta$, i.e., for a given number of buckets β, we attempt to identify the optimal serial histogram among those with β buckets. The following notation needs to be introduced for histograms with β buckets:

p_i The maximum join element in bucket b_i, $1 \leq i \leq \beta$, of a serial histogram of size β. (Clearly, $p_\beta = M$, whereas we also define $p_0 = 0$.)

Note that the set $\{p_i | 0 \leq i \leq \beta\}$ completely specifies a serial histogram of size β.

As mentioned above, in this section, we concentrate on 2-way join queries. In two different subsections, we investigate optimal serial and optimal end-biased histograms, respectively. Before proceeding in that direction, however, we present the following general and rather unexpected result, which implies that for maximal error reduction, the same histograms should be used for both relations in a 2-way join query.

THEOREM 5.1. *Consider a function-free equality join query Q of two relations R_0 and R_1 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.*

PROOF. Let $H \in \mathcal{S}_\beta$ be the histogram used for R_0, characterized by the set $\{p_i | 0 \leq i \leq \beta\}$ of maximum join elements in its buckets. Consider an arbitrary histogram $G \in \bigcup_{\beta'=1}^{M} \mathcal{S}_{\beta'}$, characterized by the set $\{q_i | 0 \leq i \leq \beta'\}$ of maximum join elements in its buckets. Let T_{ij} (resp. U_{ij}), $j = 0, 1$, be equal to $T_{ij} = \sum_{k=1}^{p_i} t_{kj}$ (resp., $U_{ij} = \sum_{k=1}^{q_i} t_{kj}$). When j is omitted, T_i (resp. T'_i) denotes the sum of the frequencies of the first p_i join elements in the result relation of Q under the histogram pair $\langle H, H \rangle$ (resp. $\langle H, G \rangle$). We prove that the approximate size T_β under $\langle H, H \rangle$ is never less than the approximate size T'_β under $\langle H, G \rangle$.

Concentrate on two arbitrary consecutive members p_{l-1} and p_l, $1 \leq l \leq \beta$, of $\{p_i | 0 \leq i \leq \beta\}$. Assume that $q_i \leq p_{l-1} < q_{i+1}$ and $q_j \leq p_l < q_{j+1}$ for some $0 \leq i, j \leq \beta'$. Consider the join elements between p_{l-1} (exclusive) and p_l (inclusive) and their contribution $T_l - T_{l-1}$ (resp. $T'_l - T'_{l-1}$) to the approximate size of the result under the histogram pair $\langle H, H \rangle$ (resp. $\langle H, G \rangle$). Let Δ be defined as $\Delta = (T_l - T_{l-1}) - (T'_l - T'_{l-1})$. To find an equivalent expression for Δ, we distinguish two cases.

Case 1. $q_i < q_j$ (or equivalently $q_{i+1} \leq q_j$). The relative ordering of the p's and q's for an arbitrary instance of this case are shown in Figure 5. From Figure 5, the following holds:

$$
\begin{aligned}
\Delta &= (T_l - T_{l-1}) - (T'_l - T'_{l-1}) \\
&= (p_l - p_{l-1}) \frac{T_{l0} - T_{(l-1)0}}{p_l - p_{l-1}} \frac{T_{l1} - T_{(l-1)1}}{p_l - p_{l-1}} - \frac{T_{l0} - T_{(l-1)0}}{p_l - p_{l-1}} \\
&\quad \times \left((q_{i+1} - p_{l-1}) \frac{U_{(i+1)1} - U_{i1}}{q_{i+1} - q_i} + \sum_{k=i+2}^{J} (q_k - q_{k-1}) \frac{U_{k1} - U_{(k-1)1}}{q_k - q_{k-1}} \right. \\
&\qquad \left. + (p_l - q_j) \frac{U_{(j+1)1} - U_{j1}}{q_{j+1} - q_j} \right) \\
&= \frac{T_{l0} - T_{(l-1)0}}{p_l - p_{l-1}} \left(T_{l1} - T_{(l-1)1} - (q_{i+1} - p_{l-1}) \frac{U_{(i+1)1} - U_{i1}}{q_{i+1} - q_i} \right. \\
&\qquad \left. - U_{j1} + U_{(i+1)1} - (p_l - q_j) \frac{U_{(j+1)1} - U_{j1}}{q_{j+1} - q_j} \right) \\
&= \frac{T_{l0} - T_{(l-1)0}}{p_l - p_{l-1}} \left(T_{l1} - T_{(l-1)1} - (q_i - p_{l-1}) \frac{U_{(i+1)1} - U_{i1}}{q_{i+1} - q_i} \right. \\
&\qquad \left. - U_{j1} + U_{i1} - (p_l - q_j) \frac{U_{(j+1)1} - U_{j1}}{q_{j+1} - q_j} \right)
\end{aligned}
$$

Fig. 5. Relative placement of maximum join elements of buckets.

$$= \frac{T_{l0} - T_{(l-1)0}}{p_l - p_{l-1}} \left(\left(T_{l1} - U_{j1} - (p_l - q_j) \frac{U_{(j+1)1} - U_{j1}}{q_{j+1} - q_j} \right) - \left(T_{(l-1)1} - U_{i1} - (p_{l-1} - q_i) \frac{U_{(i+1)1} - U_{i1}}{q_{i+1} - q_i} \right) \right). \quad (9)$$

Case 2. $q_i = q_j$ (or equivalently $q_{i+1} > p_l$). Similarly to case 1, the following holds:

$$\Delta = (p_l - p_{l-1}) \frac{T_{l0} - T_{(l-1)0}}{p_l - p_{l-1}} \frac{T_{l1} - T_{(l-1)1}}{p_l - p_{l-1}} - \frac{T_{l0} - T_{(l-1)0}}{p_l - p_{l-1}} \left((p_l - p_{l-1}) \frac{U_{(i+1)1} - U_{i1}}{q_{i+1} - q_i} \right)$$

$$= \frac{T_{l0} - T_{(l-1)0}}{p_l - p_{l-1}} \left(\left(T_{l1} - U_{i1} - (p_l - q_i) \frac{U_{(i+1)1} - U_{i1}}{q_{i+1} - q_i} \right) - \left(T_{(l-1)1} - U_{i1} - (p_{l-1} - q_i) \frac{U_{(i+1)1} - U_{i1}}{q_{i+1} - q_i} \right) \right). \quad (10)$$

Formulas (9) and (10) can be captured uniformly by a single formula. Specifically, assume that q_j is the largest among the $q's$ that is no larger than p_l, i.e., $q_j \le p_l < q_{j+1}$. In addition, let x_l, y_l, and z_l, $0 \le l \le \beta$, be defined as follows:

$$x_l = \frac{T_{l0} - T_{(l-1)0}}{p_l - p_{l-1}} \quad (11)$$

$$y_l = T_{l1} - U_{j1} \quad (12)$$

$$z_l = (p_l - q_j) \frac{U_{(j+1)1} - U_{j1}}{q_{j+1} - q_j}. \quad (13)$$

Then both (9) and (10) are equivalent to

$$(T_l - T_{l-1}) - (T'_l - T'_{l-1}) = x_l((y_l - z_l) - (y_{l-1} - z_{l-1})). \quad (14)$$

Formula (14) can now be used to capture the difference between T_β and T'_β. Clearly, $T_\beta = \Sigma_{l=1}^{\beta}(T_l - T_{l-1})$ and $T'_\beta = \Sigma_{l=1}^{\beta}(T'_l - T_{l-1})$. By (14) we obtain the

following:

$$\begin{aligned} T_\beta - T'_\beta &= \sum_{l=1}^{\beta} ((T_l - T_{l-1}) - (T'_l - T'_{l-1})) \\ &= \sum_{l=1}^{\beta} x_l((y_l - z_l) - (y_{l-1} - z_{l-1})) \\ &= -x_1(y_0 - z_0) + \sum_{l=1}^{\beta-1} (x_l - x_{l+1})(y_l - z_l) + x_\beta(y_\beta - z_\beta) \\ &= \sum_{l=1}^{\beta-1} (x_l - x_{l+1})(y_l - z_l). \end{aligned} \tag{15}$$

The last equality is due to the fact that all four of y_0, z_0, y_β, and z_β are equal to 0. This can be easily verified by using $p_0 = q_0 = T_{01} = U_{01} = 0$ and $p_\beta = q_{\beta'} = M$ and $T_{\beta 1} = U_{\beta' 1} = S_1$ (the size of R_1) in (12) and (13).

For each term of (15) we make the following two observations. First, the inequality $x_1 \geq x_{l+1}$ holds, since x_l is the average of some frequencies that are all no less than the frequencies whose average is equal to x_{l+1} (the frequency distribution of R_0 is nonincreasing). Second, for similar reasons, the inequality $y_l \geq z_l$ holds. Specifically, (12) and (13) yield

$$\begin{aligned} y_l - z_l &= T_{l1} - U_{j1} - (p_l - q_j)\frac{U_{(j+1)1} - U_{j1}}{q_{j+1} - q_j} \\ &= (p_l - q_j)\left(\frac{T_{l1} - U_{j1}}{p_l - q_j} - \frac{U_{(j+1)1} - U_{j1}}{q_{j+1} - q_j}\right). \end{aligned}$$

The first term inside the rightmost parenthesis is the average frequency in R_1 of the join elements between q_j and p_l while the second term is the corresponding average of the join elements between q_j and q_{j+1}. Since $q_{j+1} > p_l$, the parenthesis is non-negative. In conjunction with the fact that $p_l \geq q_j$ (by definition), this implies that $y_l \geq z_l$. Applying the above two observations to (15) yields the theorem. □

Example 5.1. Consider a relation R_0 that follows the Zipf frequency distribution with $z = 0.2$. Assume that a 2-bucket serial histogram is used for R_0 such that the maximum join element in the high-frequency bucket is 10. Figure 6 shows the error as a function of the corresponding join element 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. In all cases, $p = 10$ generates the least error. It is interesting to note that the error grows quite fast 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.

As mentioned right before Theorem 5.1, its most important implication is that for maximal error reduction, the same histograms should be used for

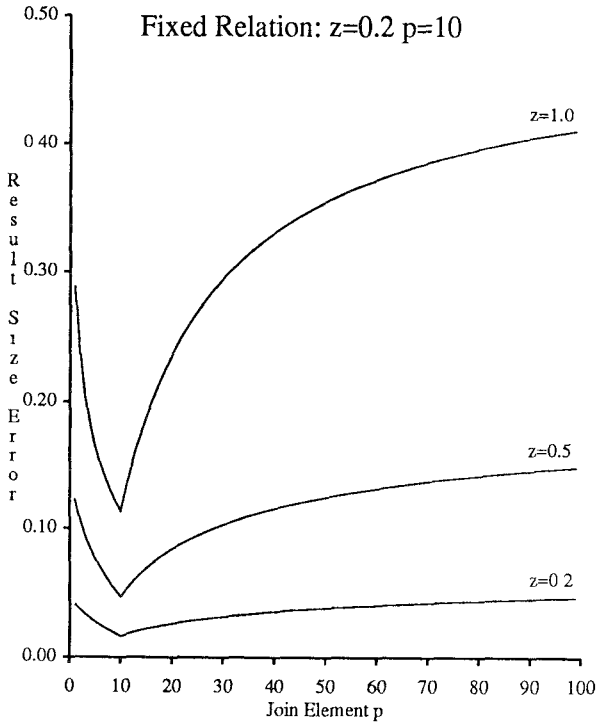

Fig. 6. Choosing a histogram for one relation given the histogram of the other.

both relations in a 2-way join query. There is no point in having more buckets in the histogram of one relation than in the other or having different buckets. Hence, we use this result in the following subsections and only search for a single histogram for both relations that would be optimal for such a query.

5.1 Optimality Among ALL Serial Histograms

In this section, we identify the optimal histogram within $\mathcal{S}_\beta$ for a histogram size $\beta \geq 2$. We assume that for both R_0 and R_1, not all frequencies are equal in each one of them, because otherwise all histograms are optimal and generate zero error. The following notation needs to be introduced for relation R_j, $j = 0, 1$:

S_j	The size of R_j measured in tuples.
T_{ij}	The sum of the highest i frequencies in the frequency distribution of R_j, i.e., $T_{ij} = \Sigma_{k=1}^{i} t_{kj}$. By definition, $S_j = T_{M_j}$.
$T_j(x)$	A monotonically increasing differentiable function on the reals that agrees with T_{ij} on the integers. (Its derivative on x is denoted by $\dot{T}_j(x)$.)

(Note the difference in the definition of T_{ij} compared to the proof of Theorem 5.1.) In the following result, T_{i1} is approximated by $T_1(x)$ so that the latter's

derivatives can be obtained. Although based on such an approximation, the outcome works well in practice. We first present the special case of the theorem for $\beta = 2$ so that intuition is developed.

THEOREM 5.2. *Consider a function-free equality join query Q of two relations R_0 and R_1 such that not all frequencies in R_0 (resp., R_1) are equal. For Q, the optimal serial histogram with 2 buckets for R_0 and R_1 satisfies the following*:

$$\frac{2\dot{T}_0(p_1)}{\gamma_0^-(p_1)} + \frac{2\dot{T}_1(p_1)}{\gamma_1^-(p_1)} = \frac{\gamma_0^+(p_1)}{\gamma_0^-(p_1)} + \frac{\gamma_1^+(p_1)}{\gamma_1^-(p_1)}, \tag{16}$$

where for $j = 0, 1$,

$$\gamma_j^+(p_1) = \frac{T_j(p_1)}{p_1} + \frac{S_j - T_j(p_1)}{M - p_1} \quad \text{and} \quad \gamma_j^-(p_1) = \frac{T_j(p_1)}{p_1} - \frac{S_j - T_j(p_1)}{M - p_1}.$$

PROOF. The size for the join result generated by a 2-bucket serial histogram as a function of p_1 is equal to

$$S^e(p_1) = \frac{T_0(p_1)T_1(p_1)}{p_1} + \frac{(S_0 - T_0(p_1))(S_1 - T_1(p_1))}{M - p_1}.$$

To find the optimal value for p_1, we differentiate $S^e(p_1)$ and equate with 0.

$$\begin{aligned}
\dot{S}^e(p_1) = 0 \Leftrightarrow & \frac{T_0(p_1)\dot{T}_1(p_1) + \dot{T}_0(p_1)T_1(p_1)}{p_1} - \frac{T_0(p_1)T_1(p_1)}{p_1^2} \\
& - \frac{(S_0 - T_0(p_1))\dot{T}_1(p_1) + \dot{T}_0(p_1)(S_1 - T_1(p_1))}{M - p_1} \\
& + \frac{(S_0 - T_0(p_1))(S_1 - T_1(p_1))}{(M - p_1)^2} = 0 \\
\Leftrightarrow \dot{T}_0(p_1)&\left(\frac{T_1(p_1)}{p_1} - \frac{S_1 - T_1(p_1)}{M - p_1}\right) \\
& + \left(\frac{T_0(p_1)}{p_1} - \frac{S_0 - T_0(p_1)}{M - p_1}\right)\dot{T}_1(p_1) \\
& = \frac{T_0(p_1)T_1(p_1)}{p_1^2} - \frac{(S_0 - T_0(p_1))(S_1 - T_1(p_1))}{(M - p_1)^2} \\
\Leftrightarrow & \dot{T}_0(p_1)\gamma_1^-(p_1) + \gamma_0^-(p_1)\dot{T}_1(p_1) \\
& = (\gamma_0^-(p_1)\gamma_1^+(p_1) + \gamma_0^+(p_1)\gamma_1^-(p_1))/2.
\end{aligned}$$

The last equality is derived by the definition of $\gamma_j^-(p_1)$ and $\gamma_j^+(p_1)$. Dividing both sides of it by $\gamma_0^-(p_1)\gamma_1^-(p_1)$ yields (16). The division is indeed allowed since both $\gamma_0^-(p_1)$ and $\gamma_1^-(p_1)$ are nonzero. This is due to the fact that not all

frequencies in each relation are equal, which forces the average of the frequencies for elements up to p_1 to be greater than the average of those after p_1.

We have shown that $S^e(p_1)$ takes an extreme value when p_1 satisfies (16). To see that it is indeed a maximum, it is enough to observe the following. (We avoid an argument based on the formula for $\dot{S}^e(p_1)$ due to its complexity.) As p_1 grows, $T_j(p_1)$ increases (and correspondingly $\dot{T}_j(p_1)$ decreases) at a rate determined by the frequencies added to T_j. The larger the value of p_1, the smaller the frequency that is added to T_j, so as p_1 grows, $\dot{T}_j(p_1)$ decreases at that rate. At the same time, $\gamma_j^+(p_1) = T_j(p_1)/p_1 + (S_j - T_j(p_1))/(M - p_1)$ also decreases (both terms decrease), but at a slower rate due to averaging. Hence, considering the value of p_1 that satisfies (16), to the left of it $\dot{S}^e(p_1) > 0$ and to the right of it $\dot{S}^e(p_1) < 0$, or equivalently $\ddot{S}^e(p_1) < 0$. Hence, for the value of p_1 that satisfies (16), S^e has a maximum. □

To use Theorem 5.2 in database histograms, $\dot{T}(p_1)$ must be approximated by a discrete quantity. We adopt one of the canonical methods that uses the average of the value differences of $T_j(x)$ between $p_1 - 1$ and p_1 and between p_1 and $p_1 + 1$, i.e., $2\dot{T}_j(p_1) = (T_j(p_1) - T_j(p_1 - 1)) + (T_j(p_1 + 1) - T_j(p_1)) = T_j(p_1 + 1) - T_j(p_1 - 1) = t_{p_1 j} + t_{(p_1+1)j}$. Hence, the following corollary of Theorem 5.2 can be obtained.

COROLLARY 5.1. *Consider a function-free equality join query Q of two relations R_0 and R_1 such that not all frequencies in R_0 (resp., R_1) are equal. For Q, the optimal serial histogram with 2 buckets for R_0 and R_1 satisfies the following*:

$$\frac{t_{p_1 0} + t_{(p_1+1)0}}{\gamma_{10}^-} + \frac{t_{p_1 1} + t_{(p_1+1)1}}{\gamma_{11}^-} \approx \frac{\gamma_{10}^+}{\gamma_{10}^-} + \frac{\gamma_{11}^+}{\gamma_{11}^-}, \tag{17}$$

where γ_{1j}^- and γ_{1j}^+ are derived from $\gamma_j^-(p_1)$ and $\gamma_j^+(p_1)$, respectively, by replacing $T_j(p_1)$ with $T_{p_1 j}$.

Applying the corollary to identify the optimal 2-bucket serial histogram can be done in time linear in the number of join elements. Roughly, this would involve scanning the join elements from 1 to M, and for each one calculating the expressions in the two sides of (17) and comparing them, until (17) is satisfied. The following example illustrates how Corollary 5.1 can be applied on specific frequency distributions; the results have been obtained by using such an algorithm.

Example 5.2. Consider again the Zipf distribution of Section 2.3 with $z = 0.1$ and assume that both relations follow it. In this case, (17) is satisfied for $p_1 = 19$. One can verify that this choice generates the largest approximation for the 2-way join result size. Note that the break point between the two buckets is far from the median (which would be 50 in this case). This is due to the fact that the distribution is skewed, with its high frequencies being rarer than the lower ones (Figure 1). Hence, this example illustrates how (17) captures point (i) of the intuition described in the end of Section 4 on the dependency of optimal serial histograms on the rarity of frequencies.

Theorem 5.2 can be generalized to deal with an arbitrary number of buckets β. This generalized version is given below without its proof, which follows the steps of the one of Theorem 5.2.

THEOREM 5.3. *Consider a function-free equality join query Q of two relations R_0 and R_1 such that not all frequencies in R_0 (resp., R_1) are equal. For Q, the optimal serial histogram with β buckets for R_0 and R_1 satisfies the following*:

$$\forall 1 \le l \le \beta - 1, \qquad \frac{2\dot{T}_0(p_l)}{\gamma_0^-(p_l)} + \frac{2\dot{T}_1(p_l)}{\gamma_1^-(p_l)} = \frac{\gamma_0^+(p_l)}{\gamma_0^-(p_l)} + \frac{\gamma_1^+(p_l)}{\gamma_1^-(p_l)},$$

where for $j = 0, 1$,

$$\gamma_j^+(p_l) = \frac{T_j(p_l) - T_j(p_{l-1})}{p_l - p_{l-1}} + \frac{T_j(p_{l+1}) - T_j(p_l)}{p_{l+1} - p_l} \quad \text{and}$$

$$\gamma_j^-(p_l) = \frac{T_j(p_l) - T_j(p_{l-1})}{p_l - p_{l-1}} - \frac{T_j(p_{l+1}) - T_j(p_l)}{p_{l+1} - p_l}.$$

Applying the above theorem to identify the optimal serial histogram for R_0 and R_1 still requires polynomial time in the number of join elements, where the degree of the polynomial is the number of desired buckets minus 1. The reason for the complication compared to the 2-bucket case is that the optimal break points cannot be identified independently of each other, so roughly all possible combinations of points must be examined. If the number of desired buckets is considered as a parameter of the size of the input, then the algorithm requires exponential time in that parameter.

5.2 Optimality Among End-Biased Histograms

The above results identify the optimal histograms among all serial ones. Because of the complexity involved in applying these results, a reasonable alternative that several database systems have adopted is to only support end-biased histograms [17]. In that case, the candidate break points for buckets are much fewer. Unfortunately, differentiation has not been as effective in identifying the optimal end-biased histograms as before. Although it results in a theorem similar to Theorem 5.2 or 5.3, the formulas involved are complex and unintuitive. As an alternative, we explore an inductive approach in which the univalued buckets of the end-biased histogram are chosen one at a time. Given a fixed set of $\beta - 1$ univalued buckets, choosing the optimal next univalued bucket results in a histogram that may not always be the optimal end-biased histogram with β buckets, but it often provides a satisfactory approximation. According to the above approach, the first k, $1 \le k < \beta - 1$, univalued buckets that are chosen can be ignored when choosing the $(k + 1)$-th one. This is done by concentrating on the unique multivalued bucket that has been formed at that point and identifying which of the two (sets of) join elements that are associated with its highest and lowest frequency should be placed in a univalued bucket. Hence, without

loss of generality and to simplify presentation, the following series of results only addresses the first step in the histogram formation process, i.e., the case of finding the first frequency to be placed in a univalued bucket.

Based on Theorem 5.1, for optimal results the two relations must have the same histogram. Therefore, there are only two alternatives for the optimal end-biased histogram with two buckets for a 2-way join query. The first such histogram places in a univalued bucket the maximum possible number of join elements that are all associated with the highest frequency in both R_0 and R_1. The second such histogram does the same for elements associated with the lowest frequency in both R_0 and R_1. Let k and k' be the number of join elements placed in a univalued bucket in the above two histograms, respectively. In an effort to obtain succinct and usable criteria, we assume for simplicity that $k = k'$. The following theorem establishes a relationship between the optimal histogram and properties of the specific distributions involved. As in the previous subsection, S_j, $j = 0, 1$, denotes the size of R_j.

THEOREM 5.4. *Consider a function-free equality join query Q of two relations R_0 and R_1 such that not all frequencies in R_0 (resp., R_1) are equal. Consider the end-biased histogram with 2 buckets that is optimal for Q. The frequency that is accurately maintained in its univalued bucket is chosen as follows*:

$$\textit{Frequency in univalued bucket of } R_j = \begin{cases} t_{1j} & \text{if } \alpha_0 + \alpha_1 \geq 0 \\ t_{Mj} & \text{if } \alpha_0 + \alpha_1 \leq 0 \end{cases}$$

where for $j = 0, 1$,

$$\alpha_j = \frac{(t_{1j} + t_{Mj})/2 - S_j/M}{t_{1j} - t_{Mj}}.$$

PROOF. Let S_1^e and S_2^e denote the result size approximations when t_{1j} and t_{Mj} are chosen as the frequencies to be maintained in the univalued buckets, respectively. Based on (4), and assuming k join elements in the univalued buckets of each of the candidate histograms, a comparison of the two sizes yields the following:

$$\begin{aligned}
S_1^e \geq S_2^e &\Leftrightarrow kt_{10}t_{11} + (S_0 - kt_{10})(S_1 - kt_{11})/(M - k) \\
&\quad \geq kt_{M0}t_{M1} + (S_0 - kt_{M0})(S_1 - kt_{M1})/(M - k) \\
&\Leftrightarrow (M - k)kt_{10}t_{11} - (M - k)kt_{M0}t_{M1} \\
&\qquad - S_0k(t_{11} - t_{M1}) - S_1k(t_{10} - t_{M0}) + k^2t_{10}t_{11} - k^2t_{M0}t_{M1} \geq 0 \\
&\Leftrightarrow t_{10}t_{11} - t_{M0}t_{M1} - \frac{S_0}{M}(t_{11} - t_{M1}) - \frac{S_1}{M}(t_{10} - t_{M0}) \geq 0 \\
&\Leftrightarrow \left(\frac{t_{10} + t_{M0}}{2} - \frac{S_0}{M}\right)(t_{11} - t_{M1}) + (t_{10} - t_{M0})\left(\frac{t_{11} + t_{M1}}{2} - \frac{S_1}{M}\right) \geq 0 \\
&\Leftrightarrow (t_{11} - t_{M1})(t_{10} - t_{M0})(\alpha_0 + \alpha_1) \geq 0.
\end{aligned}$$

Based on the premises of the theorem, $t_{1j} > t_{Mj}$ for $j = 0, 1$. Hence, dividing the above formula by the nonzero $(t_{10} - t_{M0})(t_{11} - t_{M1})$ completes the proof. □

If R_0 and R_1 are the same relation R with frequencies t_i, $1 \leq i \leq M$, and size S_R, the following simple corollary can be obtained.

COROLLARY 5.2. *Consider a function-free equality join query Q of relation R with itself and suppose that not all frequencies in R are equal. Consider the end-biased histogram with 2 buckets that is optimal for Q. The frequency that is accurately maintained in its univalued bucket is chosen as follows*:

$$\textit{Frequency in univalued bucket of } R = \begin{cases} t_1 & \textit{if } \dfrac{t_1 + t_M}{2} \geq \dfrac{S_R}{M} \\ t_M & \textit{if } \dfrac{t_1 + t_M}{2} \leq \dfrac{S_R}{M} \end{cases}.$$

PROOF. If $\alpha_0 = \alpha_1 \equiv \alpha$, Theorem 5.4 implies that the optimal choice depends on the sign of α. Since t_1 is always greater than t_M, the denominator of α is always positive. Therefore, the optimal choice depends on the sign of its nominator alone. □

There is a rather intuitive explanation of Corollary 5.2, which captures point (i) of Section 4. When $\alpha \geq 0$, the frequency distribution is more skewed towards lower values (the average of the distribution is lower than the average of the two extremes). In that case, the highest frequency of the distribution is, in some sense, more distinguished (rarer) than the lowest one, and is therefore more valuable to maintain in a univalued bucket. A similar argument holds for the opposite case as well.

The criterion provided by Theorem 5.4 is rather simple to apply for any specific 2-way join between two relations, since α_j can be easily computed. Unfortunately, it does not provide a general answer when considering all possible 2-way joins that can be formed between relations from a large set. Depending on the particular join, the answer for any relation may be different. Below we offer a general heuristic that will work well in many cases. For that we need to first prove the following simple proposition.

PROPOSITION 5.1. *If not all frequencies in a relation R are equal, then* $|\alpha| \leq 1/2$.

PROOF. We prove that $\alpha \leq 1/2$; the other inequality can be proved similarly.

$$\alpha \leq 1/2 \Leftrightarrow \frac{(t_1 + t_M)/2 - S_R/M}{t_1 - t_M} \leq 1/2 \Leftrightarrow Mt_M \leq S_R.$$

Since there is at least one frequency (t_1) that is higher than t_M, the last inequality always holds in the strict sense. Equality can be obtained at the limit $M \to \infty$. □

Table III. Values of α for the Zipf Distribution for Various Values of z

z	$0.0+\epsilon$	0.02	0.04	0.06	0.08	0.10
α	0.290	0.296	0.301	0.307	0.312	0.318

The above proposition can be used in applying the following heuristic. For each relation R_j of interest, α_j should be calculated. Based on the values that are obtained, the following decision can be made: (a) if α_j is positive (resp. negative) for most relations, then t_{1j} (resp. t_{Mj}) should be chosen for all histograms; (b) if α_j has a large absolute value (e.g., above 0.3) whenever it is positive (resp. negative) and a small absolute value (e.g., below 0.1) whenever it is negative (resp. positive), then t_{1j} (resp. t_{Mj}) should be chosen for all histograms. If none of the above holds and there is a varied mix of positive and negative values of α_j, then additional considerations should be taken into account, e.g., frequency or importance of queries, which are beyond the scope of this paper.

Example 5.3. Consider again the Zipf distribution of Section 2.3 for various values of z. The corresponding values of α are given in Table III.

In the above, ϵ is an arbitrarily small number. There are three points that we want to emphasize for this example. First, the value of α does not change very dramatically as z increases. Hence one should expect to see large differences in the values of this parameter only when there are dramatic differences in the skew of distributions. Second, for the Zipf distribution, which is claimed to be quite common in "natural" data, α is always positive. Third, even for extremely small values of z, the value of α is relatively high. The combination of the three points above implies that, for data that follows Zipf distributions, the univalued buckets of end-biased histograms should contain the high frequencies in the distributions.

6. ASYMPTOTICALLY OPTIMAL SERIAL HISTOGRAMS

The techniques used in the previous section to characterize optimal histograms for 2-way join queries cannot be generally applied to obtain similar characterizations for larger queries. Moreover, even in the cases where they can, the resulting criteria are rather cumbersome and are not easily applicable in practice. Nevertheless, they have shown a clear trend of the optimal histograms when the number of relations in a t-clique query increases. Specifically, the optimal serial histograms tend to have many buckets each one of which contains few elements associated with high frequencies and one large one with the remaining elements of low frequency. Similarly, the optimal end-biased histograms tend to have increasingly more of their univalued buckets containing elements with high frequencies. The following theorem formalizes the above observation in the limit. We first introduce some

notation about a family of relations $\{R_j | j \in \mathcal{N} \cup \{0\}\}$ (where $\mathcal{N}$ is the set of positive natural numbers):

- θ_j The minimum join element, i.e., value of i, in the multivalued bucket of a high-biased histogram $H_j \in \mathcal{V}_{\beta_j}$ for R_j, $j \in \mathcal{N} \cup \{0\}$. That is, $\theta_j - 1$ denotes the number of join elements associated with the $\beta_j - 1$ highest frequencies in R_j.
- θ The minimum of all values of θ_j, i.e., $\theta = \min_{j \in \mathcal{N} \cup \{0\}}\{\theta_j\}$.

THEOREM 6.1. *Consider a family of t-clique queries* $\{Q_N | N \in \mathcal{N}\}$, *where* Q_N *is on relations* R_j, $0 \le j \le N$. *Let* $\langle H_j^{(N)} \rangle$ *be the* $(N+1)$*-vector of serial histograms that is optimal for* Q_N *within* $\langle \mathcal{S}_{\beta_j} \rangle$, *where all histogram sizes satisfy* $\beta_j > 1$ *and the approximate frequencies are denoted by* $t_{ij}^{e(N)}$. *Let* $\langle H_j \rangle$ *be the corresponding high-biased histogram vector in* $\langle \mathcal{V}_{\beta_j} \rangle$. *If for an infinite number of values of* $j \in \mathcal{N} \cup \{0\}$, $t_{\theta_j}^{e(N)} < t_{1j}$, *then at the limit, the following holds*:

$$\lim_{N \to \infty} \langle H_j^{(N)} \rangle = \langle H_j \rangle.$$

PROOF. Let S^e and $S^{e(N)}$ denote the result size approximations for Q_N under the high-biased histograms and under arbitrary serial histograms $G_j^{(N)}$ with β_j buckets for R_j, $0 \le j \le N$, respectively. The theorem is proved by showing that $\lim_{N \to \infty}(S^e - S^{e(N)}) \ge 0$. Observe that $t_{1j}^e = t_{1j}$, since H_j is high-biased. Formula (4) yields the following:

$$\begin{aligned} S^e - S^{e(N)} &= \sum_{i=1}^{M} \prod_{j=0}^{N} t_{ij}^e - \sum_{i=1}^{M} \prod_{j=0}^{N} t_{ij}^{e(N)} \\ &= \prod_{j=0}^{N} t_{1j} \left(\sum_{i=1}^{\theta-1} \prod_{j=0}^{N} \frac{t_{ij}^e}{t_{1j}} - \sum_{i=1}^{\theta-1} \prod_{j=0}^{N} \frac{t_{ij}^{e(N)}}{t_{1j}} \right. \\ &\qquad \left. + \sum_{i=\theta}^{M} \prod_{j=0}^{N} \frac{t_{ij}^e}{t_{1j}} - \sum_{i=\theta}^{M} \prod_{j=0}^{N} \frac{t_{ij}^{e(N)}}{t_{1j}} \right). \end{aligned}$$

By the premises of the theorem, for each product in the rightmost sum of the above formula, an arbitrary number of its fractions are less than 1. Hence, as $N \to \infty$, that sum tends to 0. In addition, the limit of the sum immediately to its left is clearly non-negative. Combining these two facts yields

$$\lim_{N \to \infty} (S^e - S^{e(N)}) \ge \lim_{N \to \infty} \prod_{j=0}^{N} t_{1j} \left(\sum_{i=1}^{\theta-1} \prod_{j=0}^{N} \frac{t_{ij}^e}{t_{1j}} - \sum_{i=1}^{\theta-1} \prod_{j=0}^{N} \frac{t_{ij}^{e(N)}}{t_{1j}} \right) \Leftrightarrow$$

$$\lim_{N \to \infty} (S^e - S^{e(N)}) \ge \lim_{N \to \infty} \left(\sum_{i=1}^{\theta-1} \prod_{j=0}^{N} t_{ij}^e - \sum_{i=1}^{\theta-1} \prod_{j=0}^{N} t_{ij}^{e(N)} \right).$$

We claim that the quantity in the parenthesis is always non-negative and therefore the same holds for its limit. Assume the worst case that maximizes

the rightmost sum in the parenthesis (with the negative sign), i.e., assume that $\theta - 1$ is the highest join element in some bucket of $G_j^{(N)}$, for all $j \in \mathcal{N} \cup \{0\}$. (This represents a worst case because then $t_{(\theta-1)j}^{e(N)}$ is the average of frequencies that are no lower than $t_{(\theta-1)j}$.) Also, by the definition of θ, for all $1 \leq i \leq \theta - 1$ and $j \in \mathcal{N} \cup \{0\}$, $t_{ij}^e = t_{ij}$. Hence, the claim is equivalent to

$$\sum_{i=1}^{\theta-1} \prod_{j=0}^{N} t_{ij} \geq \sum_{i=1}^{\theta-1} \prod_{j=0}^{N} t_{ij}^{e(N)}, \tag{18}$$

with $t_{ij}^{e(N)}$ representing the worst case discussed above. By Lemma 3.1, for all $j \in \mathcal{N} \cup \{0\}$, the M-vector $\langle t_{ij} \rangle$ majorizes the M-vector $\langle t_{ij}^{e(N)} \rangle$. In conjunction with Theorem 2.2, this fact proves (18). Therefore, $\lim_{N \to \infty}(S^e - S^{e(N)}) \geq 0$. The above was shown for an arbitrary histogram vector, so it holds for the optimal vector $\langle H_j^{(N)} \rangle$ as well. In that case, it can be equivalently expressed as $lim_{N \to \infty} \langle H_j^{(N)} \rangle = \langle H_j \rangle$. □

A reasonable criticism of Theorem 6.1 is that it holds under a rather technical and unnatural condition that θ must satisfy. In reality, this condition is necessary to ensure that there is enough variability in the histograms that the behavior in the limit is indeed asymptotic. For example, consider the case where the condition is violated because all but two frequency distributions are uniform. Then essentially only the two non-uniform distributions determine the optimal histogram and Theorem 5.4 should be used for that purpose. We expect that in most cases the condition of the theorem would be satisfied. Therefore, the above result captures the essence of the asymptotic behavior of optimal histograms. Characteristic examples where Theorem 6.1 is applicable include the case where all relations are the same, the case where for each relation the θ highest frequencies are distinct, and the case where for all $j \in \mathcal{N} \cup \{0\}$, $\theta_j = \theta$.

Example 6.1. To illustrate the importance of high-biased histograms in large join queries, we continue Example 4.1, where instead of dealing with a 2-way join query, we deal with a 5-way join query. Recall that the relations are identical, contain 10000 tuples whose join domain contains 100 elements, and their frequency distributions are Zipf with $z = 0.2$ (Section 2.3). Assume that the histograms maintained for the five relations are identical as well. We have calculated the error generated by the same types of histograms that were used in Example 4.1. The results for the 5-way join query are shown in Table IV.

Again the serial and high-biased histograms are better than the trivial and the nonserial ones. There are two additional interesting points to note. First, the errors are significantly larger than those of the 2-way join query. The exponential growth is very evident once again [4, 5]. Second, contrary to what was observed in Example 4.1, the high-biased histogram generates a smaller error than the serial one. This was to be expected at the limit due to Theorem 6.1. In this case, however, the distributions are skewed enough that the cross-over point comes early, at relatively small queries.

Table IV. Error in a 5-Way Join Query for Various Histograms

Histogram	Error
Trivial	79.42%
Nonserial	78.79%
Serial	25.00%
High-biased	16.43%

Example 6.2. As another example, we show the effect of the error of using the high-biased histogram with $L + 1$ buckets in all relations of the example introduced in Section 2.3 (for various values of L). That is, we assume that the join elements of the relations follow a Zipf distribution with $z = 0.02$ and $z = 0.1$, and show the effect on the error when there are $L = 1$, 5, and 10 univalued buckets in the histogram. Note that $L = 0$ corresponds to the trivial histogram (uniform distribution). Figure 7 shows a graphical representation of (5) for these cases.

The results are rather impressive. We observe that in both cases, even maintaining a single element has tremendous impact in reducing the total error. An even more surprising result is that, in all cases with $L > 0$, the error as a function of N has a maximum. That is, beyond a certain point, as the query size grows, the error decreases. The reason is that with more relations, the value of the frequency distribution for the most common elements becomes an increasingly larger fraction of the total size of the query result, thus reducing the error (point (ii) of the intuition described in the end of Section 4). As expected, this is more dramatic for the more skewed distribution ($z = 0.1$). We must emphasize that, by Theorem 2.3, the case presented deals with the largest possible result size (worst-case error given a fixed value for the estimated result size). If the frequencies given by the Zipf distributions were associated with the join elements of a different way, then the original error (for $L = 0$) would be less than what is shown in Figure 7, but the error under the high-biased histograms for each value of L could be larger. Nevertheless, the improvement for the worst-case error gives much hope for being able to optimize very large queries in some cases, without being overwhelmed by the errors in the query relations.

The phenomenon of the worst-case error having a maximum as a function of the number of joins N under high-biased histograms is not unique to the above example. The next theorem provides general conditions for when this happens, and also extends the conditions to capture the case where the error tends to a finite number other than 0. Thus, it complements Theorem 6.1, which showed that for large join queries, high-biased histograms are optimal: not only are they optimal, but often the corresponding error tends to zero as well.

THEOREM 6.2. *Consider a family of t-clique queries* $\{Q_N | N \in \mathcal{N}\}$, *where* Q_N *is on relations* R_j, $0 \leq j \leq N$. *Let* $\langle H_j \rangle$ *be an* $(N + 1)$*-vector of arbitrary*

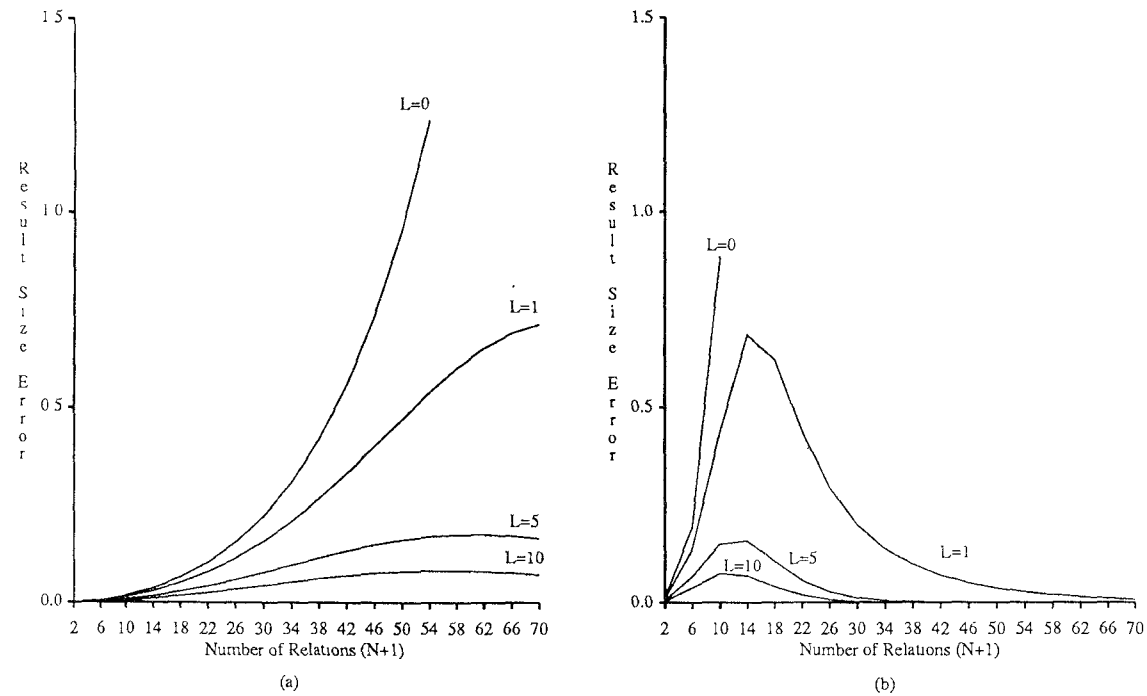

Fig. 7. Query result size error under high-biased histograms for all relations: (a) $z = 0.02$ and (b) $z = 0.1$.

histograms in $\langle \mathcal{S}_{\beta_j} \rangle$ *with approximate frequencies denoted by* t^e_{ij}. *Suppose that* Z (*resp.* Z') *denotes the minimum join element, i.e., value of* i, *such that for an infinite number of values of* $j \in \mathcal{N} \cup \{0\}$, $t_{ij} < t_{1j}$ (*resp.*, $t^e_{ij} < t_{1j}$), *and let* K'_i *denote* $K'_i = \{j | t^e_{ij} < t_{1j}$ *and* $j \in \mathcal{N} \cup \{0\}\}$. *Then the following holds*:

$$\lim_{N \to \infty} D = \begin{cases} c & \text{if } Z' > 1 \\ \infty & \text{if } Z' = 1, \end{cases}$$

where c *is a constant and* $0 \leq c < \infty$. *Moreover, if* $Z = Z' > 1$ *and* $\forall 1 \leq i \leq Z - 1, \forall j \in K'_{Z-1}$, *the equality* $t_{ij} = t^e_{ij}$ *holds, then* $c = 0$.

PROOF. Using (5), we obtain the following for the error D:

$$1 + D = \frac{\sum_{i=1}^{M} \left(\prod_{j=0}^{N} t_{ij} \right)}{\sum_{i=1}^{M} \left(\prod_{j=0}^{N} t^e_{ij} \right)} = \frac{\sum_{i=1}^{Z-1} \prod_{j=0}^{N} \frac{t_{ij}}{t_{1j}} + \sum_{i=Z}^{M} \prod_{j=0}^{N} \frac{t_{ij}}{t_{1j}}}{\sum_{i=1}^{Z'-1} \prod_{j=0}^{N} \frac{t^e_{ij}}{t_{1j}} + \sum_{i=Z'}^{M} \prod_{j=0}^{N} \frac{t^e_{ij}}{t_{1j}}}. \tag{19}$$

By the definition of Z and Z', the limit of the rightmost sums in both the nominator and the denominator is 0. Similarly to K'_i, let K_i denote $K_i = \{j | t_{ij} < t_{1j}$ and $j \in \mathcal{N} \cup \{0\}\}$. Then (19) implies that

$$1 + \lim_{N \to \infty} D = \frac{\sum_{i=1}^{Z-1} lim_{N \to \infty} \prod_{j=0}^{N} \frac{t_{ij}}{t_{1j}}}{\sum_{i=1}^{Z'-1} lim_{N \to \infty} \prod_{j=0}^{N} \frac{t^e_{ij}}{t_{1j}}} = \lim_{N \to \infty} \frac{\sum_{i=1}^{Z-1} \prod_{j \in K_i} \frac{t_{ij}}{t_{1j}}}{\sum_{i=1}^{Z'-1} \prod_{j \in K'_i} \frac{t^e_{ij}}{t_{1jj}}}, \tag{20}$$

where for all $1 \leq i \leq Z - 1$ (resp. $1 \leq i \leq Z' - 1$), K_i (resp. K'_i) is finite. The nominator in (20) is always strictly positive, since for $i = 1$, $\prod_{j=0}^{N} (t_{ij}/t_{1j}) = 1$. If $Z' = 1$, then there are no terms in the sum of the denominator, which therefore tends to 0, implying that $lim_{N \to \infty} D = \infty$. If $Z' > 1$, then due to the finiteness of K_i and K'_i, the fraction in (20) is independent of N, and therefore equal to itself in the limit. This implies that $lim_{N \to \infty} D = c$, where $c \geq 0$ is a constant that is equal to that fraction minus 1.

Next we present a sufficient condition for when $c = 0$. Observe that the following always hold: $Z' \leq Z$, $K_i \subseteq K'_i$, $K_i \subseteq K_{i+1}$, and $K'_i \subseteq K'_{i+1}$. From (20) we obtain the following:

$$1 + c = \frac{\sum_{i=1}^{Z'-1} \prod_{j \in K_i} \frac{t_{ij}}{t_{1j}}}{\sum_{i=1}^{Z'-1} \prod_{j \in K'_i} \frac{t^e_{ij}}{t_{1j}}} + \frac{\sum_{i=Z'}^{Z-1} \prod_{j \in K_i} \frac{t_{ij}}{t_{1j}}}{\sum_{i=1}^{Z'-1} \prod_{j \in K'_i} \frac{t^e_{ij}}{t_{1j}}} \equiv E + F. \tag{21}$$

Clearly, $F \geq 0$. Also, for any given collection $\{K'_i | 1 \leq i \leq Z' - 1\}$, the lowest value of E is obtained when $\forall 1 \leq i \leq Z' - 1$, (i) $K_i = K'_i$, and (ii) $Z' - 1$ is

the highest join element in some bucket of H_j for all $j \in K'_t$ (so that $t^e_{(Z'-1)j}$ is computed without taking into account any frequencies that are lower than $t_{(Z'-1)j}$). In that case, by Lemma 3.1, for all $j \in K'_{Z'-1}$, the $(Z'-1)$-vector $\langle t_{ij}/t_{1j} \rangle$ majorizes the $(Z'-1)$-vector $\langle t^e_{ij}/t_{1j} \rangle$. Therefore, by Theorem 2.2, $E \geq 1$. By (21), these lower bounds on E and F imply that $c = 0$ is equivalent to $E = 1$ and $F = 0$. These in turn satisfy the following.

$$F = 0 \Leftrightarrow Z = Z'$$

$$E = 1 \Leftarrow \forall 1 \leq i \leq Z' - 1, \quad \forall j \in K_{Z'-1}, \quad t^e_{ij} = t_{ij}.$$

The first equivalence is immediately derived from the definition of F in (21). (Note that $Z = Z'$ is also necessary for $c = 0$.) The second reverse implication is again straightforward since it equates the nominator and the denominator of the fraction defining E.[2] (Note that this condition also implies that $\forall 1 \leq i \leq Z' - 1$, $K_i = K'_i$.) The two together provide a sufficient condition for $c = 0$. □

Example 6.3. The above theorem shows that the phenomenon observed in Example 6.2 where the corrected error presented a maximum and tended to 0 for a large number of relations is more general. We present one additional example, where we compare the behavior of the error under high-biased versus under low-biased histograms. With the Zipf distribution, even for a 2-way join, high-biased histograms are to be preferred over low-biased ones. (This can be verified by applying Corollary 5.2.) We therefore use a different distribution to expose the fact that, even if for a small number of joins low-biased histograms are more preferable, there is some value of N beyond which high-biased ones are the right choice.

Assume that all relations in the database are equal to each other and the frequency distribution for all $0 \leq j \leq N$ is as follows:

$$t_{ij} = \begin{cases} 143 - \lfloor (i+1)/2 \rfloor & \text{if} \quad 1 \leq i \leq 80 \\ 101 - i & \text{if} \quad 81 \leq i \leq 100 \end{cases}. \tag{22}$$

The numbers were chosen so that there are some common characteristics with the Zipf distributions discussed in Section 2.3, i.e., the size of the relations is very close to 10000, the join domain contains 100 elements, and the maximum value is almost the same as that of the most skewed Zipf distribution ($z = 0.1$) that we examined (143 vs. 142). Figure 8(a) is a graphical representation of (22), where one can easily see that the distribu-

[2] Some necessary and sufficient conditions for when vector majorization implies strict inequality (or equality) of vector functions do exist. They could be used to derive a necessary and sufficient condition for $c = 0$. They are very complex, however, and would be rather impossible to use in practice. In addition, they would require us to introduce much additional notation together with several other results from mathematics. Hence, we decided to present a much simpler sufficient condition that covers many common cases.

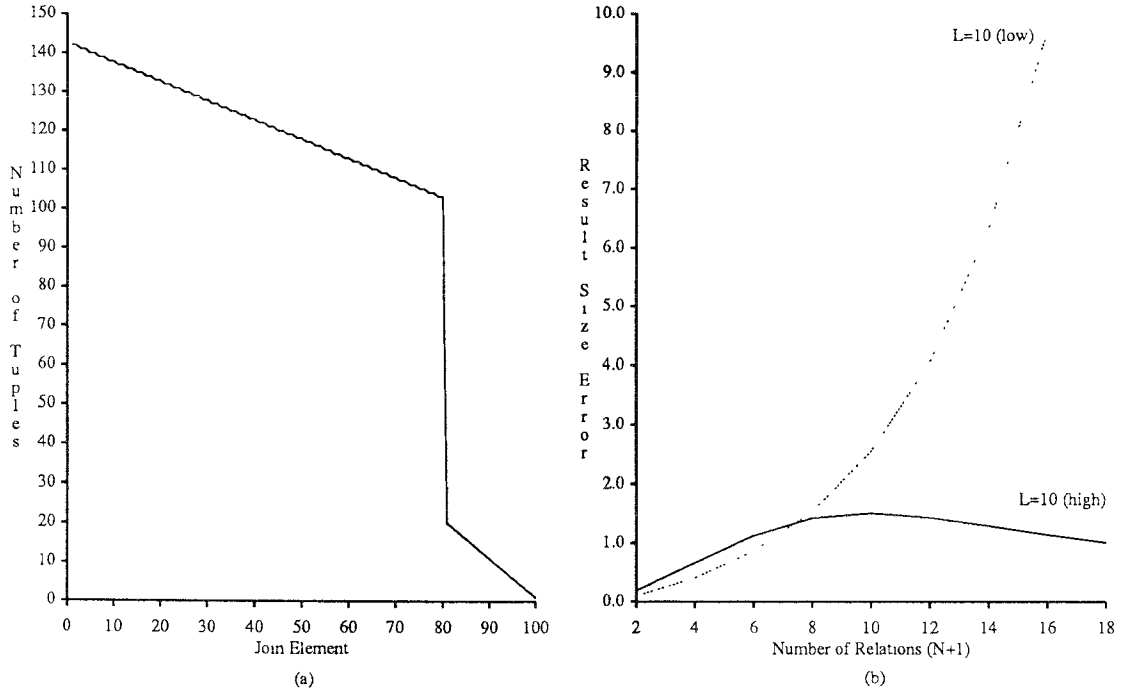

Fig. 8. (a) Frequency distribution; (b) Corresponding join result size error when maintaining the highest or the lowest frequencies.

tion is skewed towards high frequencies, i.e., high frequencies are more common.

Consider the two cases where the database system uses a high-biased histogram with $L + 1$ buckets for $L = 10$ or a low-biased histogram with $L + 1$ buckets for $L = 10$. Figure 8(b) is a graphical representation of (5) for the two cases. The relative error in the query result size is shown as a function of the number of relations in the query. As expected from Corollary 5.2, for a small number of relations the error is smaller under the low-biased than under the high-biased histogram. Nevertheless, the result of Theorem 6.2 can also be observed, since the error under the high-biased histogram has a maximum beyond which it tends to zero, whereas the error under the low-biased histogram tends to infinity.

7. HISTOGRAM RECOMMENDATION

In this section, we reflect on the entire set of results presented in this paper and, based on them, make some concrete recommendations on what types of histograms should be maintained for database relations and how these should be chosen. To put the recommendations that follow in the right perspective, however, we should emphasize the fact that the results in this paper have not addressed the entire issue of histogram optimality with respect to limiting worst-case error propagation. They are based on certain assumptions, the most restrictive of which is the form of the queries (t-clique queries). Nevertheless, we believe that they shed enough light on the problem and that the directions to which they point with respect to how histograms should be chosen may be useful in general.

To reduce the worst-case error, the optimal histograms are always serial. For any specific join query, the specific serial histogram that is optimal depends on the number of participating relations in the query and their specific frequency distributions. In reality, the database administrator decides on the maintained histogram having a collection of queries in mind, with an overall goal of obtaining good estimates for all of them, large and small. The results presented in this paper showed that high-biased histograms are the optimal choice for large queries. On the contrary, for small queries, the correct decision depends on the specific frequency distributions of the participating relations in ways that one cannot deal with each individual relation independently. Clearly, the database administrator would like to be able to decide on the optimal histogram for each relation by looking at the characteristics of that relation alone.

Given the above, for an overall effective estimation, we suggest the following heuristic approach for choosing a histogram for a relation. Independent of the frequency distribution of the relation, its histogram has some univalued buckets with the join elements associated with the largest frequencies. Further, if the distribution is very skewed towards large frequencies (which is not very common), the remaining elements should be divided among another set of buckets based on the specific distribution of the individual relations, so that small queries, e.g., 2-way join queries, do not suffer as well. The

effectiveness of the above heuristic approach on real databases is an issue that requires further investigation.

8. SUMMARY

Error propagation in the context of query optimization is one of the most significant challenges facing the efforts to effectively optimize queries of much higher complexity than those with which conventional technology can deal. Maintaining histograms to approximate frequency distributions in relations is a common technique used by database systems to limit the propagation of errors. In this paper, we have studied histograms and how they reduce errors that represent the worst case. Specifically, we have introduced serial and end-biased histograms and showed that for the restricted class of t-clique queries, the optimal histogram for errors in the query result size is always serial. For 2-way equality join queries with no function symbols, we have presented results on finding the optimal serial histogram and the optimal end-biased histogram based on the query characteristics and the frequency distributions of values in the join attributes of the query relations. We have also examined histogram optimality for very large t-clique queries (in the limit) and showed that high-biased histograms are always optimal.

This work has raised several interesting questions and issues. What is the precise characterization of optimal histograms for queries that have more than one join? 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? What are the characteristics of optimal histograms for non-t-clique queries, primarily arbitrary equality join queries with more than one join? How do the results of this paper change when considering completely different types of queries (e.g., nonequality 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.

REFERENCES

1. CHRISTODOULAKIS, S. Estimating block transfers and join sizes. In *Proceedings of the 1983 ACM-SIGMOD Conference on the Management of Data*, (San Jose, Calif., May 1983), 40–54.
2. CHRISTODOULAKIS, S. Implications of certain assumptions in database performance evaluation. *ACM Trans. Database Syst. 9*, 2 (June 1984), 163–186.
3. CHRISTODOULAKIS, S. On the estimation and use of selectivities in database performance evaluation. Res. Rep. CS-89-24, Dept. of Computer Science, Univ. of Waterloo, June 1989.
4. IOANNIDIS, Y., AND CHRISTODOULAKIS, S. On the propagation of errors in the size of join results. In *Proceedings of the 1991 ACM-SIGMOD Conference on the Management of Data* (Denver, Colo., May 1991), 268–277.
5. IOANNIDIS, Y., AND CHRISTODOULAKIS, S. Error propagation under the uniform distribution assumption. In preparation, 1993.

6. KAMEL, N., AND KING, R. A model of data distribution based on texture analysis. In *Proceedings of the 1985 ACM-SIGMOD Conference on the Management of Data*, (Austin, Tex., May 1985), 319–325.
7. KOOI, R. P The optimization of queries in relation databases. Ph.D. dissertation, Case Western Reserve Univ., Sep. 1980.
8 MACKERT, L. F., AND LOHMAN, G. M. R^* validation and performance evaluation for distributed queries In *Proceedings of the 12th International VLDB Conference* (Kyoto, Aug. 1986), 149–159.
9. MACKERT, L. F., AND LOHMAN, G. M. R^* validation and performance evaluation for distributed queries. In *Proceedings of the 1986 ACM-SIGMOD Conference on the Management of Data* (Washington, D.C., May 1986), 84–95.
10 MANNINO, M. V., CHU, P., AND SAGER, T. Statistical profile estimation in database systems. *ACM Comput. Serv., 20,* 3 (Sep. 1988), 192–221.
11. MARSHALL, A. W., AND OLKIN, I. *Inequalities: Theory of Majorization and Its Applications.* Academic Press, New York, 1979.
12. MERRETT, T. H., AND OTOO, E. Distribution models of relations. In *Proceedings of the 5th International VLDB Conference* (Rio de Janiero, Oct. 1979), 418–425.
13. MURALIKRISHNA, M., AND DEWITT, D. J. Equi-depth histograms for estimating selectivity factors for multi-dimensional queries. In *Proceedings of the 1988 ACM-SIGMOD Conference on the Management of Data* (Chicago, Ill., June 1988), 28–36.
14. MUTHUSWAMY, B., AND KERSCHBERG, L. A ddsm for relational query optimization. In *Proceedings of the ACM Annual Conference* (Denver, Colo., Oct. 1985).
15 PIATETSKY-SHAPIRO, G., AND CONNELL, C. Accurate estimation of the number of tuples satisfying a condition. In *Proceedings of the 1984 ACM-SIGMOD Conference on the Management of Data* (Boston, Mass., June 1984), 256–276.
16. SELINGER, P. G., ASTRAHAN, M. M., CHAMBERLIN, D. D., LORIE, R. A., AND PRICE, T. G. Access path selection in a relational database management system. In *Proceedings of the ACM SIGMOD International Symposium on Management of Data* (Boston, Mass., June 1979), 23–34
17. SELINGER, P. G. June 1989. Personal communication.
18. ZIPF, G. K. *Human Behavior and the Principle of Least Effort.* Addison-Wesley, Reading, Mass., 1949.

Received September 1991; revised August 1992; accepted December 1992