On the Propagation of Errors in the Size of Join Results

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

Stavros Christodoulakis†
Electronic and Computer Eng. Department
Technical University of Crete
73100 Chania, Crete, Greece
stavros%omalos@csi.forth.gr

Abstract

Query optimizers of current relational database systems use several statistics maintained by the system on the contents of the database to decide on the most efficient access plan for a given query. These statistics contain errors that transitively affect many estimates derived by the optimizer. We present a formal framework based on which the principles of this error propagation can be studied. Within this framework, we obtain several analytic results on how the error propagates in general, as well as in the extreme and average cases. We also provide results on guarantees that the database system can make based on the statistics that it maintains. Finally, we discuss some promising approaches to controlling the error propagation and derive several interesting properties of them.

1 Introduction

Query optimizers of relational database systems decide on the most efficient 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 [S+79]. In most cases, the statistics represent an inaccurate picture of the actual contents of the database. This is due to two reasons: first, only aggregate information is maintained by the system, e.g., maximum, minimum, and average value in an attribute, or a histogram with the number of tuples in a relation for each of several value ranges in an attribute; second, as the database is updated the information becomes obsolete. Hence, the query optimizer uses erroneous data to accomplish its task.

The above 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 can be devastating. Consequently, the decision of the query optimizer can 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.

In this paper, we present a formal framework based on which the principles of error propagation can be studied. Within this framework, we obtain analytic results on the problem under different models of the statistics that are kept by the database system. We also obtain results giving intuition on the methods that could be used to reduce the magnitude of the error propagation.

There are several parameters whose inaccurate estimation can lead a query optimizer in wrong decisions. Moreover, there are several operators that can 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 the join as the parameter and the operator of interest respectively. This choice is motivated by their importance in query optimization and their sensitivity to error propagation.

*Partially supported by the National Science Foundation under Grant IRI-8703592.

†On leave from the Computer Science Department, University of Waterloo, Waterloo, Ontario, Canada, N2L 3G1.

We are aware of no work in the area of error propagation in the context of database query optimization. There is extensive literature on deriving good estimates for the parameters of the result of database operations, which has been surveyed by Mannino, Chu, and Sager [MCS88]. This is not the case, however, with the effect of the unavoidable 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 been essentially ignored. The primary reason for that has been the low complexity of the queries that current systems have to face. As the query complexity increases in future database applications, this can no longer be the case. Formal techniques are necessary to increase our understanding of how much query complexity can be tolerated before the combined errors in the individual relations of the query become unacceptable. In hindsight, however, it becomes apparent that such an understanding is needed even for the currently common, low complexity queries, where errors can grow enough to cause erroneous decisions by the optimizers [Chr89, ML86a, ML86b, Sel89].

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. Section 3 derives precise formulas for the error in the result of a join query as a function of the errors in the query relations. Section 4 elaborates on the behavior of the formulas derived in the previous section as the interaction of the errors in the query relations changes. The focus of this section is on extreme and expected values of the result error. Section 5 addresses the case where the database system maintains some thresholds for the error in the query relations and derives some upper bounds on the error in the query result that can be guaranteed based on these thresholds. The results of all three of the last sections are rather pessimistic, showing that the error propagates exponentially with the number of joins. Section 6 discusses one form of controlling the error propagation by maintaining accurate statistics about certain interesting values in the join attributes of relations. An example is also shown where, with this form of correction, not only the error does not increase exponentially, but in fact beyond a certain point, it decreases with the number of joins. Finally, Section 7 summarizes our results and gives directions for future work.

2 Formulation

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*. The study of error propagation in its most general form is rather difficult. We make the following assumptions:

(A1) All joins are equality joins.

(A2) Only one attribute from each relation participates in joins (independent of the number of joins it does so).

(A3) The set of elements that appear in the join attribute of a relation is the same for all relations. This set is the *join domain* of the query.

An obvious implication of (A2) is that all join attributes are of the same type. Also, we assume some arbitrary ranking 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, i.e., the number of unique elements in the join attributes of R_j, $0 \leq j \leq N$.
- t_{ij} The number of tuples in R_j whose join attribute contains the i-th element of the join domain, $1 \leq i \leq M$, $0 \leq j \leq N$.
- S The size of the result relation of the query.

For each relation R_j, $0 \leq j \leq N$, the set $L_j = \{t_{ij} | 1 \leq i \leq M\}$ is called the *join element distribution* in R_j. Clearly, the above parameters are related with the following formula:

$$S = \sum_{i=1}^{M} \prod_{j=0}^{N} t_{ij}. \tag{1}$$

Most often database systems have inaccurate knowledge of the join element 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 2.1 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.

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 join element distribution is denoted by $L_j^e = \{t_{ij}^e\}$ 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 that of 'relative error'.

For each relation R_j, $0 \leq j \leq N$, the set $E_j = \{d_{ij} | d_{ij} = (t_{ij}/t^e_{ij}) - 1, 1 \leq i \leq M\}$ is called the *relative error distribution* of R_j. The maximum, average, and minimum values in E_j are called the *maximum, average*, and *minimum errors* in R_j respectively.

For a given set of relative error distributions for the relations R_j, $0 \leq j \leq N$, let $D = (S/S^e) - 1$ be the corresponding relative error in the estimated size of the query result. Also, let D_i be the value in the relative error distribution of the query result that is associated with the i-th element of the join domain. We focus our attention on two issues related to the problem of error propagation. First, we are interested in identifying the relationship between D and D_i on one hand and $\{d_{ij}\}$ on the other, which describes the behavior of error. Second, we are also interested in identifying the relationship between D and D_i and a variety of aggregations of $\{d_{ij}\}$. This is because most often database systems maintain only a handful of characteristic quantities that summarize the relative error distributions for all relations. Therefore, it is very important for database systems to be able to draw useful conclusions about the errors in the query result from this limited information. The first problem is primarily discussed in Sections 3 and 4, whereas the second one is discussed in Section 5. Due to the lack of space, all results in this paper are presented without proofs. For more details, the interested reader is refered to the full version of the paper [IC91].

3 Error Behavior

3.1 Arbitrary Join Element Error

We seek to identify the relationship between D_i and $\{d_{ij}\}$. Such a relationship essentially addresses the error propagation problem for a join query that is followed by an equality selection on one of the join attributes.

Theorem 3.1 Under assumptions (A1)-(A3), for all i, the following holds: $1 + D_i = \prod_{j=0}^{N}(1 + d_{ij})$.

3.2 Average Join Element Error

Let δ be the average error in the query result, i.e., $\delta = \text{avg}_{1 \leq i \leq M}\{D_i\}$. The following theorem provides a formula for δ.

Theorem 3.2 Under assumptions (A1)-(A3), the following holds:

$$1 + \delta = \frac{1}{M} \sum_{i=1}^{M} \prod_{j=0}^{N} (1 + d_{ij}). \tag{2}$$

3.3 Query Result Size Error

When dealing with the size of the full join result without a selection on the join attribute, it is difficult to extract a nice general formula as in Theorem (3.1) for the corresponding relative error D. There is a special case, however, in which a concise formula is derivable. This case is captured by the following assumption.

(A4) For all relations, the approximation of the join element distribution that the database system uses is uniform, i.e., for all i and j, $t^e_{ij} = t^e_j$, where t^e_j is a constant that depends on the relation R_j only.

Assumption (A4) is made by the query optimizers of several database systems, so the study of error propagation under uniform distribution is of special interest. The following theorem derives a formula for the error in the query result size for that case.

Theorem 3.3 Under assumptions (A1)-(A4), the following holds:

$$1 + D = \frac{1}{M} \sum_{i=1}^{M} \prod_{j=0}^{N} (1 + d_{ij}). \tag{3}$$

A comparison of equations (2) and (3) yields the following very interesting corollary for the case of uniform approximation.

Corollary 3.1 Under assumptions (A1)-(A4), the error in the query result size is equal to the average error in that result, i.e., $D = \delta$.

The primary implication of Corollary 3.1 is that all the forthcoming analysis and results for the error in the query size apply to the average error as well.

3.4 Discussion

Theorems 3.1, 3.2, and 3.3 do not allow for much optimism. All types of error in an N-way join grow exponentially with N. If there are both positive and negative values in $\{d_{ij}\}$, the situation may not be very bad, since their effect may be mutually canceled. It is very common, however, that the same join element appears many (few) times in most query relations, the number of times it does so is underestimated (overestimated) in most relations, and therefore, for the same i, most values in $\{d_{ij}\}$ are positive (negative). In these cases, the absolute value of the error continuously grows with the number of joins. This can severely affect the ability of query optimizers to make correct decisions.

4 Characteristics of the Error Behavior

As discussed above, the specific combination of positive and negative errors associated with the various join elements in the query relations affects differently the corresponding errors in the query result. In this section, we present results that provide some insight into the characteristics of the error behavior under different such combinations. Suppose that the distribution followed by the relative error in each relation R_j is given, without specific information on the specific error value associated with each join element. We consider all possible such associations and study the resulting differences in the error behavior. Being independent of the specific such association, our results provide relationships between the errors in the query result and the error in each query relation independent of all others.

For each relation R_j, let $V_j = \{d_j(k) | 1 \le k \le M$ and there exists a unique $1 \le i \le M$ s.t. $d_j(k) = d_{ij}\}$. From the preceding discussion, we assume no knowledge of the specific association of the i's to the k's. The following parameters are used in the coming subsections:

- $\mu_j^{(K)}$ The K-th moment about the origin of $V_j' = \{1 + d_j(k) | 1 \le k \le M\}$ for R_j, i.e., $\mu_j^{(K)} = \mathrm{avg}_{1 \le k \le M}\{(1 + d_j(k))^K\}$.
- δ_j The average relative error in R_j, i.e., $\delta_j = \mathrm{avg}_{1 \le k \le M}\{d_j(k)\}$ or $\delta_j = \mu_j^{(1)} - 1$.
- d_j^+ The maximum relative error[1] in R_j, i.e., $d_j^+ = \max_{1 \le k \le M}\{d_j(k)\}$.
- d_j^- The absolute value of the minimum relative error[2] in R_j, i.e., $d_j^- = -\min_{1 \le k \le M}\{d_j(k)\}$.

4.1 Maximum Value of the Error

This subsection gives a tight upper bound on the error in the query result size when V_j is given. The bounds obtained for individual join elements are the same with those obtained in Section 5, so they are not presented here as well.

Based on known inequalities from mathematics, e.g., the Power Means inequality and the Hölder inequality [Kaz64], several interesting results can be obtained for the maximum value of D.

Theorem 4.1 Under assumptions (A1)-(A4), the following holds: $1 + D \le \left(\prod_{j=0}^{N} \mu_j^{(N+1)}\right)^{1/(N+1)}$.

[1]The maximum relative error is assumed to be positive.
[2]The minimum relative error is assumed to be negative.

Corollary 4.1 Under assumptions (A1)-(A4), if for all $0 \le j, l \le N$, $V_j = V_l$ and $\mu_j^{(N+1)} = \mu_l^{(N+1)} = \mu^{(N+1)}$, then the following holds: $1 + D \le \mu^{(N+1)}$.

The upper bound given in Theorem 4.1 or Corollary 4.1 is tight. D reaches that value when, for all $1 \le k \le M$, the k-th largest error is associated with the same join element in all relations and the relative magnitude of the error among the elements is the same. An interesting question that arises is how this worst case behaves as N grows. The following result offers some insight in that direction.

Proposition 4.1 Suppose that the average error in at least one relation is nonnegative, i.e., without loss of generality, $\sum_{k=1}^{M} d_N(k) \ge 0$. Then, $\left(\prod_{j=0}^{N} \mu_j^{(N+1)}\right)^{1/(N+1)} \ge \left(\prod_{j=0}^{N-1} \mu_j^{(N)}\right)^{1/N}$.

The result of Proposition 4.1 can be interpreted as follows. If for at least one relation, the approximation of its join element distribution used by the database system does not on the average overestimate the actual distribution, then the worst case error in the query result size monotonically increases with the number of joins. This captures as a special case the situation when an accurate average of the join element distribution is maintained, i.e., when the average error is zero.

A final comment on the upper bound of D is that it is always larger than a quantity that grows exponentially with N. More specifically, one can easily prove the following result. (Recall that $d_j^+ = \max_{1 \le k \le M}\{d_j(k)\}$ and that it is assumed positive.)

Proposition 4.2 The following inequality holds: $\left(\prod_{j=0}^{N} \mu_j^{(N+1)}\right)^{1/(N+1)} \ge \frac{1}{M} \prod_{j=0}^{N}(1 + d_j^+)$.

The main conclusion that can be drawn from the above results are again rather pessimistic. In the worst case, the error in the query result grows exponentially with the number of joins. Except for very small queries, the error in the query result size becomes too large for the query optimizer to trust it.

4.2 An Example

The above results on the error propagation problem hold for arbitrary join element distributions. To obtain a better feeling for their implications, we apply them to a specific instance of the problem, which will also be our running example for the entire paper. In particular, we examine the case where the assumed join element distribution is uniform whereas the actual join element distribution is Zipf [Chr84, Zip49]. The main characteristic of the Zipf distribution is that it assigns

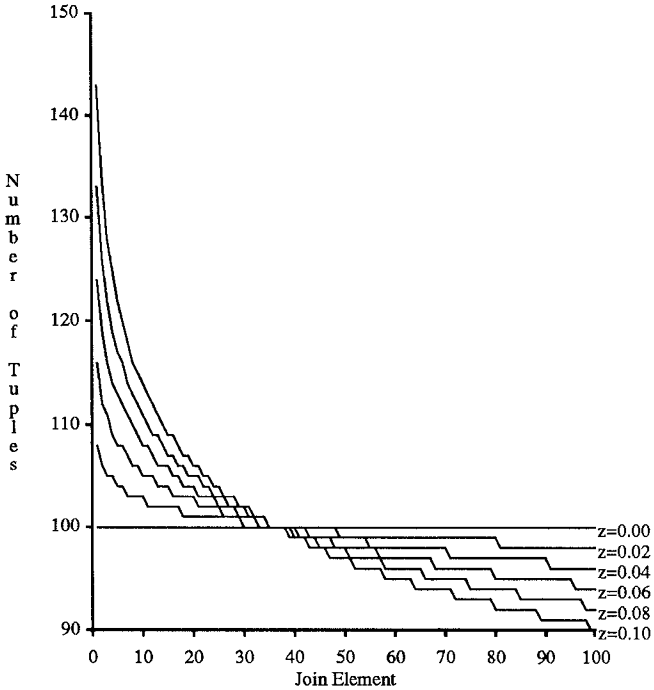

Figure 1: Zipf join element distribution.

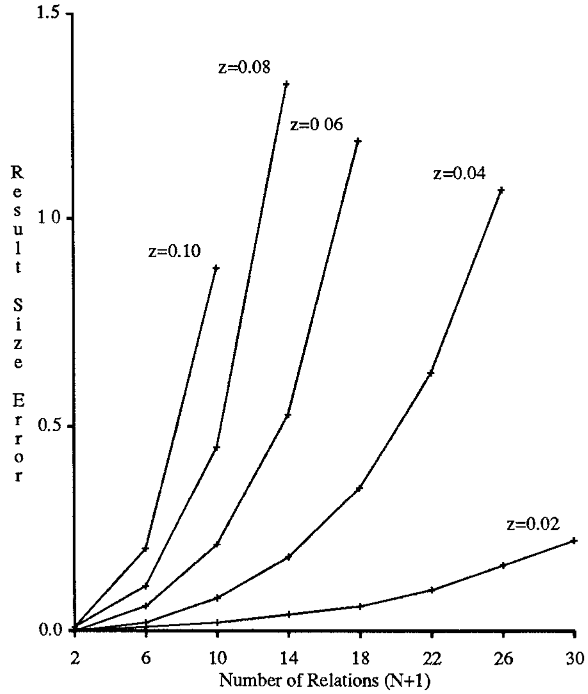

Figure 2: Join result size error for Zipf distributions under uniform approximation.

high values to few join elements and low values 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.

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

$$t_{ij} = T_j \frac{1/i^z}{\sum_{i=1}^{M} 1/i^z} \text{ for all } 1 \leq i \leq M. \tag{4}$$

In (4), 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 (4) 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 (3) 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. From the above discussion, the error in this case is equal to the upper bound given in Theorem 4.1 or Corollary 4.1, since the k-th largest error has the same value and is associated with the same join element in all relations. Hence, the error shown in Figure 2 is equal to the $(N+1)$-st moment of the sums of unity with each error in the individual relations. The speed with which small errors in the individual relations propagate in the result is rather discouraging.

4.3 Expected Value of the Error

The following parameters are used to represent the expected value of errors:

- a The expected value of the relative error associated with some element of the join domain in the query result.
- A The expected value of the relative error in the query result size.

The following result provides a relationship between a and $\{\delta_j\}$.

Theorem 4.2 Under assumptions (A1)-(A3), the following holds:[3] $1 + a = \prod_{j=0}^{N}(1 + \delta_j)$.

If the approximation of the join element distribution is uniform, the following result provides a relationship between A and $\{\delta_j\}$.

Theorem 4.3 Under assumptions (A1)-(A4), the following holds: $1 + A = \prod_{j=0}^{N}(1 + \delta_j)$.

[3] By the definition of δ_j, this can also be written as $1 + a = \prod_{j=0}^{N} \mu_j^{(1)}$.

Note that the above theorems imply that if for all j, δ_j=0, then $a = A = 0$ as well. That is, if the average error in the individual relations is zero, the same is true for the expected values of the error in the query result as well. These observations can be quite misleading. Errors can be both positive and negative. Hence, an expected error value of zero provides no information on the actual error in each specific instance, which can have an arbitrarily high absolute value.

Example 4.1 The Zipf distributions of Section 4.2 serve our purpose well in this case also. Assuming that the Zipf distribution with $z = 0$ is used as the database approximation, the average error in each relation is zero. According to Theorems 4.2 and 4.3, this implies that the expected value of the error among all associations of join elements to distribution values is zero. However, for every specific instance the error can be very significant. Such was the case presented in Section 4.2, where the error grew exponentially (Proposition 4.2) with the number of relations. □

5 Maintaining Thresholds on the Error

A reasonable mode of operation for database systems is to maintain a threshold on some aggregate error among all join elements of each relation, and based on that, predict a corresponding threshold for the error in the query result size. It has been proposed in the past that, for individual join element errors, the average error in each relation is the one on which a threshold should be placed. However, Theorems 3.1, 3.2, and 4.2 provide clear evidence for the inadequacy of that approach. Thresholds on the average error only bound the expected value of the error in the query result, but provide no guarantees for low errors in any specific case. Hence, we contend that, for individual join element errors, imposing thresholds on the maximum (and minimum) error is the correct approach [Chr89]. Similar comments can be made about the error in the query result as well.

Let t_j^+ (t_j^-) be the maximum (minimum) value in the join element distribution of R_j. We assume that, for each relation, the database system maintains both these extremes together with t_j^e (uniform approximation). Note that $d_j^+ = (t_j^+/t_j^e) - 1$ and $d_j^- = 1 - (t_j^-/t_j^e)$, so essentially the database system maintains the extreme relative errors for each relation as well. We demonstrate in the following subsections that, given the above, it is possible to obtain tight upper and lower bounds on D_i and D. These represent the values that the database system can guarantee not to be exceeded by the individual join element error in the query result and the query result size error respectively.

5.1 Join Element Error

Given d_j^+ and d_j^- for each relation R_j, the corresponding thresholds on the join element error in the query result are denoted by D^+ and D^- respectively. That is, $-D^- \leq D_i \leq D^+$. The following theorem derives formulas for these thresholds.

Theorem 5.1 Under assumptions (A1)-(A3), for all i, the following holds:

$$1 + D^+ = \prod_{j=0}^{N}(1 + d_j^+), \qquad (5)$$

$$1 - D^- = \prod_{j=0}^{N}(1 - d_j^-). \qquad (6)$$

Clearly, D_i can become equal to D^+ ($-D^-$) when the same join element is associated with the maximum (minimum) relative error in all relations. Thus, Theorem 5.1 shows that the upper bound that can be guaranteed for the maximum error in the query result grows exponentially with the query size. The database system should enforce very strict thresholds on the error in the base relations to achieve reasonable errors in multi-relation join queries.

Example 5.1 Consider again the example introduced in Section 4.2. Clearly, for this case, the maximum (minimum) error is associated with the most (least) common element in the join domain. Applying Theorem 5.1 to this specific case yields the relative error for this element, which is graphically shown in Figure 3 for the Zipf parameter z=0.2, 0.4, and 0.8. Clearly, if not accurate enough information is kept about the individual relations, the maximum error in the result becomes untrustworthy after very few joins. □

5.2 Query Result Size Error

Given d_j^+ and d_j^- for each relation R_j, the corresponding thresholds on the query result size error are denoted by Γ^+ and Γ^- respectively. That is, $-\Gamma^- \leq D \leq \Gamma^+$.

All previous results on D are based on assumption (A4), which states that the database system uses a uniform distribution as an approximation of the join element distribution. There is no restriction in (A4), however, on the characteristics of that uniform distribution, i.e., all these results hold for arbitrary values of $\{t_j^e\}$. In many systems, the value of t_j^e is equal to the average number of tuples per join element in R_j at some point in time. Hence, the previous results hold even in the case where the assumed average is inaccurate because updates have been performed on the relation since that average was obtained.

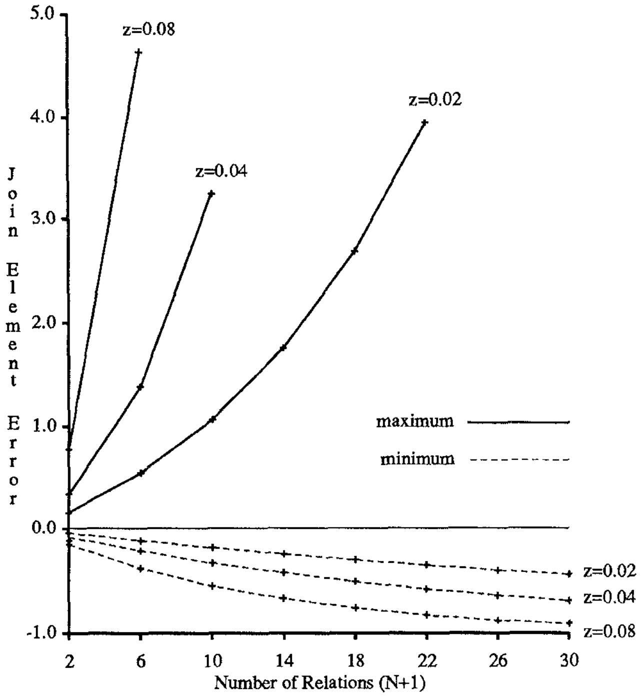

Figure 3: Maximum and minimum errors in the query result.

Assumption (A4) is not adequate to obtain an accurate threshold for the query result size error when the database system operates as described in the beginning of this section. Specifically, we need to make the following assumption:

(A5) For each relation R_j, $t_j^e = (\sum_{i=1}^{M} t_{ij})/M$.

In other words, we study the problem of error propagation when the join element distribution assumed by the database system is uniform and its value is the average value of the real distribution. In that case, we say that the approximation of the join element distribution used by the system is *accurate uniform*.

For each relation R_j, let

$$M_j = M\frac{t_j^e - t_j^-}{t_j^+ - t_j^-} = M\frac{d_j^-}{d_j^+ + d_j^-}. \tag{7}$$

Without loss of generality, we assume that the M_j values are ordered based on the subscripts of the corresponding relation names, i.e., $j < l$ implies that $M_j \leq M_l$. Based on that, we define $D_{k,l}^+$ and $D_{k,l}^-$ as follows:

$$1 + D_{k,l}^+ = \begin{cases} \prod_{i=k}^{l}(1 + d_i^+) & \text{if } k \leq l \\ 1 & \text{otherwise} \end{cases},$$

$$1 - D_{k,l}^- = \begin{cases} \prod_{i=k}^{l}(1 - d_i^-) & \text{if } k \leq l \\ 1 & \text{otherwise} \end{cases}.$$

The following theorem provides the formula for Γ^+. Its proof is based on extensions of known results on majorization [MO79].

Theorem 5.2 Under assumptions (A1)-(A5), the following holds:

$$\begin{aligned} \Gamma^+ &= \sum_{j=0}^{N} D_{0,j-1}^- d_j^+ (1 + D_{j+1,N}^+) \\ &= \sum_{j=0}^{N} (1 - D_{0,j-1}^-) d_j^- D_{j+1,N}^+. \end{aligned}$$

Note that, if $D_{0,j-1}^-$ is replaced by its maximum possible value, i.e., $D_{0,j-1}^- = 1$, then $\Gamma^+ = D_{1,N}^+$. Assumption (A3) actually prohibits $D_{0,j-1}^-$ from becoming equal to 1: all join elements must appear at least once in every relation. If there are join elements, however, that appear very few times in each relation, then $D_{0,j-1}^-$ can be very close to 1, and therefore Γ^+ can be very close to $D_{1,N}^+$. For real databases, this is a rather important observation, since experience shows that quite often data follow distributions where few elements appear many times in an attribute and the remaining elements appear very few times, thus resulting in minimum errors whose absolute value is very close to 1.

Example 5.2 Given join element distributions that have the same maximum and minimum values as the Zipf distributions of Figure 1, we compare the value of Γ^+, as given by Theorem 5.2, with the actual error when the Zipf distributions are used, as given in Section 4.2. The latter was shown in Figure 3 as a function of the number of relations in the query. The corresponding curves are drawn in Figure 4 for comparison. It is clear from the above figure that although the error in the Zipf join element distribution case was growing very fast, there can be much worse situations for other distributions that result in much higher errors. The main point is that when the database system maintains the maximum, average, and minimum values of the join element distribution of relations, the range of the error in the size of the join of the relations is extremely large even when relatively few relations are involved. □

For Γ^-, similar formulas can be obtained as the ones given by Theorem 5.2, although their derivation is a bit trickier. For two relations R_0 and R_1, however, the value of Γ^- is given by the following theorem.

Theorem 5.3 Under assumptions (A1)-(A5), the following holds:

$$\Gamma^- = \begin{cases} d_0^- d_1^- & \text{if } M_0 + M_1 \leq M \\ d_0^+ d_1^+ & \text{if } M_0 + M_1 \geq M \end{cases} = \min\{d_0^- d_1^-, d_0^+ d_1^+\}.$$

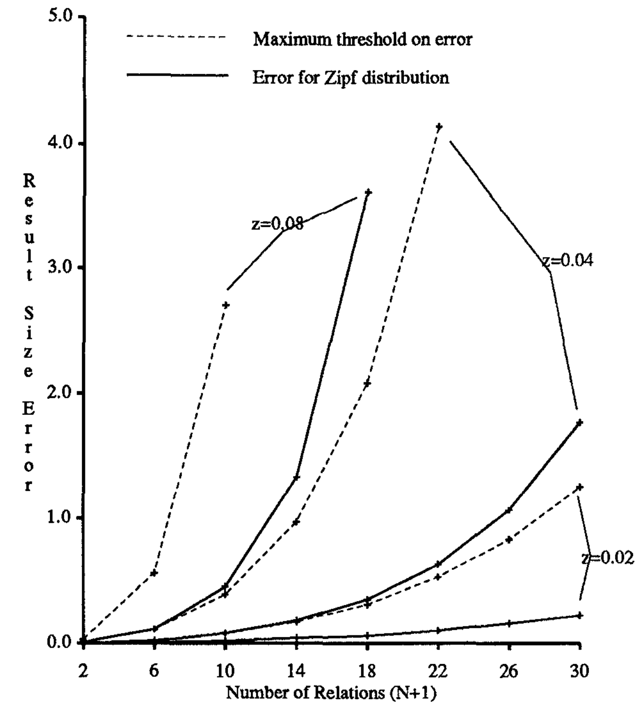

Figure 4: Maximum threshold on the query result size error.

5.3 Discussion

Assume that for all j, $d_j^- = d^-$ and $d_j^+ = d^+$, for some constants d^- and d^+, i.e., the maximum and minimum errors are the same in all relations. Then, from (6) we have that

$$D^+ = (1 + d^+)^{N+1} - 1. \tag{8}$$

The above relates the following three parameters: the maximum error in the query relations d^+, the number of joins N, and the maximum error in the query result D^+. Given desirable thresholds for any two of the above three parameters, we can find a threshold for the third one. Thus, (8) provides answers to three abstract problems. Given N and d^+, finding D^+ is the "error propagation" problem, for which we have that

$$D^+ \leq (1 + d^+)^{N+1} - 1.$$

One can see immediately that the error is exponential in the number of joins. Given N and D^+, finding d^+ is the "required accuracy" problem, for which we have that

$$d^+ \leq \sqrt[N+1]{1 + D^+} - 1.$$

In other words, the maximum join element error in each relation must be kept below the $(N+1)$-st root of the maximum allowed join element error in the result. Finally, given d^+ and D^+, finding N is the "tolerable query complexity" problem, for which we have that

$$N \leq \frac{\log(1 + D^+)}{\log(1 + d^+)} - 1.$$

That is, given some threshold for the join element errors in the relations, the maximum number of joins that can be performed that would still guarantee that no join element error in the result exceeds some other threshold is roughly the quotient of the logarithms of the thresholds.

Similar statements can be made for D^- and for Γ^+ as well. For the latter, the following formula can be obtained:

$$\Gamma^+ = \frac{d^-}{d^- + d^+}(1 + d^+)^{N+1} + \frac{d^+}{d^- + d^+}(1 - d^-)^{N+1} - 1. \tag{9}$$

Comparing (8) and (9) yields that Γ^+ increases exponentially with N, only at a slightly lower rate than D^+. This is captured by the following statement:

Proposition 5.1 Under assumptions (A1)-(A5), if for all j, $d_j^- = d^-$ and $d_j^+ = d^+$, then the following holds:

$$\frac{1 + \Gamma^+}{1 + D^+} \geq \frac{d^-}{d^- + d^+}.$$

In the above proposition, equality is attained only at the limit, i.e., when $N \to \infty$.

6 Partial Corrections

Given a set of relative errors $\{d_{ij}\}$ and a corresponding query result size error D, an interesting question is how D is affected when some members of the relative error distribution are corrected. Some current systems maintain accurate values for a small number of t_{ij}'s for each relation (usually the largest ones) [Sel89]. In this section, we investigate how this particular partial correction affects D. In its general form, this approach is captured by the following assumption.

(A6) The approximation of the join element distribution that the database system uses for R_j is accurate for L elements in the join domain and accurate uniform for the remaining $M - L$ elements.

If without loss of generality we assume that $\{t_{1j}, t_{2j}, \ldots, t_{Lj}\}$ is the set of values that are maintained accurately for R_j by the database system, then assumption (A6) implies that, for $1 \leq i \leq L$, $t_{ij}^e = t_{ij}$, and for $L + 1 \leq i \leq M$, $t_{ij}^e = (\sum_{i=L+1}^{M} t_{ij})/(M - L)$.

We first want to study the case where (A6) is applied to exactly one relation. Without loss of generality, suppose that R_0 is that relation. For all relations except R_0, assumption (A4) holds, i.e., the system assumes uniform join element distribution. The following result shows the inadequacy of this approach to correcting errors when applied to a single relation.

Theorem 6.1 Under assumptions (A1)-(A3) and (A6) for R_0 and (A1)-(A4) for R_j, $1 \leq j \leq N$, D has the same value independent of the value of L.

The above result can be interpreted as follows. When for all relations R_j, $1 \leq j \leq N$, the join element distribution assumed by the database system is uniform, there is no advantage in maintaining more accurate information for relation R_0. Simply maintaining the average of the distribution accurately (or equivalently the size of R_0) results in the same error as maintaining the full distribution.

Theorem 6.1 does not hold in general: if assumption (A6) is extended to more relations, simply maintaining an accurate average for these relations is not equivalent to maintaining more information about them.

The next result that we want to present is for the case where the discussed style of correction is applied to all relations. In particular we want to investigate whether the highest values in the join element distribution are the most beneficial to maintaining or not. It is rather difficult to answer this question for the error D in general. The following theorem addresses the case where for all $1 \leq k \leq M$, the k-th largest value in the join element distribution is associated with the same join element in all relations. As discussed in Section 4.1, under assumptions (A1)-(A4), this is a necessary condition for D to reach the upper bound given in Theorem 4.1. In that case, it can be shown that the error is given by the equation

$$1 + D = \frac{\sum_{i=1}^{M}(\prod_{j=0}^{N} t_{ij})}{\sum_{i=1}^{L}(\prod_{j=0}^{N} t_{ij}) + \frac{1}{(M-L)^N}\prod_{j=0}^{N}\sum_{i=L+1}^{M} t_{ij}}. \tag{10}$$

Theorem 6.2 Under assumptions (A1)-(A3) and (A6) for R_j, $0 \leq j \leq N$, D in equation (10) is minimized when the L values of the join element distribution maintained by the system are the L highest such values.

Example 6.1 We show the effect of correcting L values in all relations of the example introduced in Section 4.2. That is, we assume that the join elements of the relations follow a Zipf distribution (Figure 1). We present the cases for $z = 0.02$ and $z = 0.1$, and we show the effect on the error when L=1, 5, and 10 elements are maintained per relation. Figure 5 shows a graphical representation of equation (10).

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,

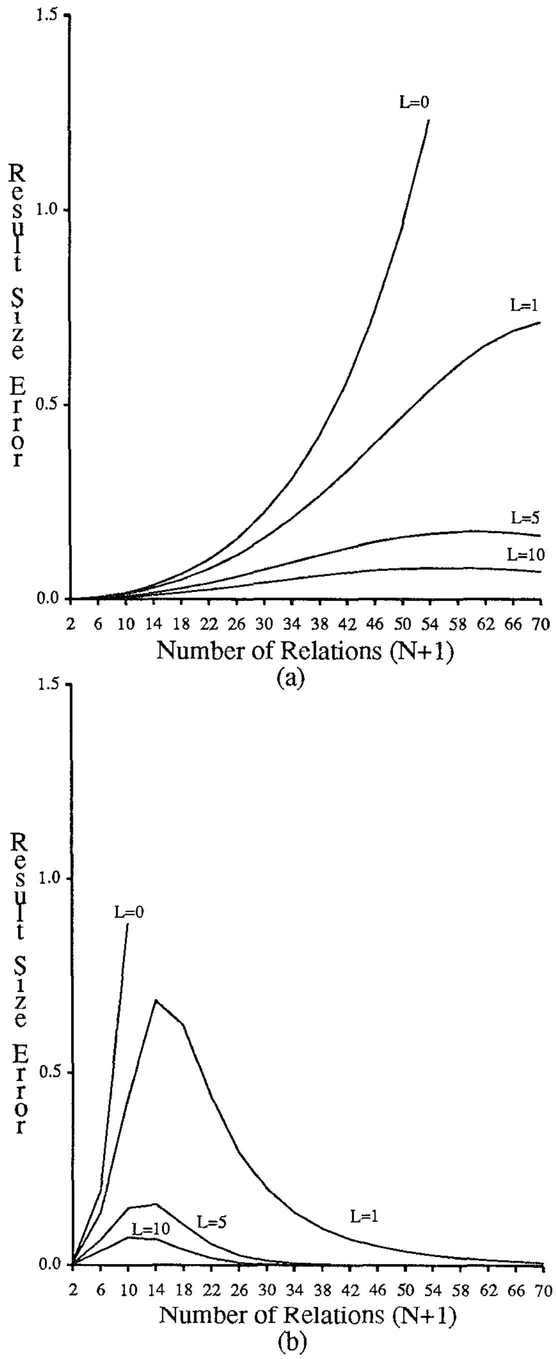

Figure 5: Query result size error under assumption (A6) for all relations: (a) $z = 0.02$ and (b) $z = 0.1$.

beyond a certain point, as the query size grows, the error decreases. This is because with more relations, the value of the join element distribution for the most common elements becomes an increasingly larger fraction of the total size of the query result, thus reducing the error. As expected, this is more dramatic for the more skewed distribution ($z = 0.1$). We must emphasize that, by Theorem 4.1, the case presented corresponds to an upper bound on the query result size error. If the Zipf distributions were associated with the join elements in a different way, then the error would be even less than what is shown in Figure 5 for each value of L. Hence, this example gives much hope for being able to optimize very large queries in some cases, without being overwhelmed by the errors in the query relations. □

7 Summary

An understanding of the error propagation problem in the context of query optimization is essential in complex database environments. Nevertheless, to the best of our knowledge, no previous work exists on the subject. In this paper, we have presented a formal framework based on which the principles of error propagation can be studied. Within this framework, we have obtained precise formulas for the error in the result of a join query as a function of the errors in the query relations. The behavior of these formulas has been studied with respect to the extreme and expected values of the error. Analytic results have also been derived on the maximum error under various statistics maintained by the database system. All these results have shown that in general the error increases exponentially with the number of joins. Finally, we have studied some promising approaches to decreasing the effect of the error propagation and have derived several interesting characteristics of them.

We believe that the results in this paper are only a first step towards understanding the effects of error propagation and the appropriate methods to control it. They can be extended in several directions so that the restrictions imposed by our model are removed, e.g., assumptions (A1)-(A3), and the usefulness of other types of maintained statistics is explored, e.g., histograms approximating join element distributions. In addition, further work is necessary to understand how errors affect the values of other interesting parameters besides size, e.g., operator cost, as well as how they affect the ranking of alternative access plans, which determines the final decision of the optimizer. We hope that the results in this paper will be helpful in these directions as well.

References

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

[Chr89] S. Christodoulakis. On the estimation and use of selectivities in database performance evaluation. Research Report CS-89-24, Dept. of Computer Science, University of Waterloo, June 1989.

[IC91] Y. E. Ioannidis and S. Christodoulakis. On the propagation of errros in the size of join results. Technical Report 1018, Computer Science Dept., University of Wisconsin, Madison, March 1991.

[Kaz64] N. D. Kazarinoff. *Analytic Inequalities.* Holt, Rinehart and Winston, New York, NY, 1964.

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

[ML86a] L. F. Mackert and G. M. Lohman. R^* validation and performance evaluation for distributed queries. In *Proc. 12th International VLDB Conference*, pages 149–159, Kyoto, Japan, August 1986.

[ML86b] L. F. Mackert and G. M. Lohman. R^* validation and performance evaluation for local queries. In *Proc. 1986 ACM-SIGMOD Conference on the Management of Data*, pages 84–95, Washington, DC, May 1986.

[MO79] A. W. Marshall and I. Olkin. *Inequalities: Theory of Majorization and Its Applications.* Academic Press, New York, NY, 1979.

[S+79] P. G. Selinger et al. Access path selection in a relational data base system. In *Proc. of the 1979 ACM-SIGMOD Conference on the Management of Data*, pages 23–34, Boston, MA, June 1979.

[Sel89] P. G. Selinger, June 1989. Personal communication.

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