Proc. of 11th Int'l Conf. on Database and Expert Applications (DEXA'00)
LNCS, Springer Verlag, Vol. 1873, pp 509-525, London, UK, Sept 2000

Cost Estimation for Large Queries via Fractional Analysis and Probabilistic Approach in Dynamic Multidatabase Environments*

Qiang Zhu, S. Motheramgari, and Yu Sun

Department of Computer and Information Science,
The University of Michigan, Dearborn, MI 48128, U.S.A.
{qzhu, motheram, yusun}@umich.edu

Abstract. Research on query cost estimation for local database systems in a multidatabase system (MDBS) has attracted many researchers in the database area recently. Obtaining good query cost estimates is crucial for performing effective global query optimization in an MDBS. However, most techniques suggested so far, including database calibrating and query sampling, consider only a static multidatabase environment. Recently, we proposed a qualitative approach to developing cost models for a dynamic multidatabase environment. It has been shown that this approach is promising in estimating the cost of a query run in any given contention state for a dynamic environment. However, a large (cost) query in practice may experience multiple contention states during its execution, which cannot be directly handled by the qualitative approach. In this paper, we propose two new techniques, i.e., fractional analysis and probabilistic approach, to solve the problem. The fractional analysis technique, which is suitable for a system environment that changes contention states gradually and smoothly, estimates a query cost by analyzing its fractions. The probabilistic approach, which is suitable for a system environment that changes contention states rapidly and randomly, estimates a query cost based on the theory of Markov chains. Cost estimation formulas for both techniques are derived, and their properties are studied. Our experimental results demonstrate that the suggested techniques are quite promising in estimating costs for large queries in a dynamic multidatabase environment.

1 Introduction

To meet users' increasing needs to access data from multiple pre-existing databases managed by heterogeneous database management systems (DBMS), multidatabase systems (MDBS) have been studied by many database researchers in recent years[7–9, 13]. An MDBS is a global system built on top of multiple local (component) DBMSs and provides users with a uniform interface to access local

* Research supported by the US National Science Foundation under Grant # IIS-9811980 and The University of Michigan under OVPR and UMD grants.

databases. A key feature of an MDBS is local autonomy that each local database system retains to serve existing local applications. The global system can only interact with local DBMSs at their external user interfaces.

A global query issued on an MDBS is decomposed into a set of local queries executed at local database systems during query processing. The results from local queries are integrated into the final query result returned to its user. However, the way to decompose a global query is not unique. Different decomposition strategies may yield significantly different performance in the distributed environment. Choosing a good decomposition and integration strategy for a given global query is the task of global query optimization. To perform global query optimization, cost information for local queries to be performed on local database systems is required. However, such information is unavailable to the global query optimizer since the internal implementation details of a local DBMS is unknown to the MDBS. Estimating the costs of local queries at the global level in an MDBS is a major challenge for global query optimization in the system.

To tackle this challenge, a number of techniques have been proposed in the literature. In [3], Du *et al.* proposed a calibration method that makes use of the observed costs of some special queries run against a special synthetic calibrating database to deduce necessary local cost parameters. In [6], Gardarin *et al.* extended Du *et al.*'s method so as to calibrate cost models for object-oriented local database systems in an MDBS. In [15–17], Zhu and Larson proposed a query sampling method that develops regression cost models for local query classes based on observed costs of sample queries run against actual user databases. In [14], Zhu and Larson introduced a fuzzy method based on fuzzy set theory to derive fuzzy cost models in an MDBS. In [10], Naacke *et al.* suggested an approach to combining a generic cost model with specific cost information exported by wrappers for local database systems. In [1], Adali *et al.* suggested to maintain a cost vector database to record cost information for every query issued to a local database system. In [12], Roth *et al.* introduced a framework for costing in the *Garlic* federated system.

All above techniques considered only a static environment, i.e., assuming that the environment does not change significantly over time. However, such an assumption may not be true in reality since many factors such as the number of concurrent processes in a multidatabase system environment may change significantly. The cost of a query can be dramatically different at different times in a dynamic system environment. For example, in one of our experiments, the cost of a sample query[1] performed on Oracle 8 in a dynamical environment varied from 2.58 sec. to 127.05 sec. (49 times!) when we had 1 to 30 concurrent user processes in the environment. Hence query cost estimates obtained for a static environment cannot be used in a dynamic environment.

To capture dynamic factors in a cost model, we recently proposed a qualitative approach[18]. This approach extends our previous query sampling method[15–17] and develops regression cost models using qualitative variables to indicate

[1] The query was *SELECT a1, a5, a7 FROM R WHERE a3 > 300 and a8 < 2000* on table $R(a1, a2, ..., a9)$ with 50,000 tuples of random data.

system contention states. Each contention state reflects a combined effect of dynamic factors on the system. Although such a cost model can be used to estimate the costs of queries for any contention state in the dynamic environment, each query is assumed to be run in a single contention state. The qualitative approach cannot directly solve the problem to estimate the cost of a large query run in multiple contention states.

To estimate the costs of large queries experiencing multiple contention states in a dynamic multidatabase environment, we develop two new techniques in this paper. The first technique, called fractional analysis approach, is to estimate query costs in a dynamic environment in which the system contention states change gradually and smoothly. The idea is to analyze and integrate the fractions of a query cost for multiple experienced contention states. The second technique, called probabilistic approach, is to estimate query costs in a dynamic environment in which the system contention states change rapidly and randomly. The idea is to make use of the theory of Markov chains to derive a cost formula to estimate the query costs in such an environment. These two techniques together with our qualitative approach provide a suite of techniques to estimate the costs of queries for different cases in a dynamic multidatabase environment.

The rest of this paper is organized as follows. Section 2 outlines our qualitative approach to developing cost models with qualitative variables for dynamic multidatabase environments. Section 3 presents the fractional analysis approach to estimating query costs. Section 4 discusses the probabilistic approach to estimating query costs. Section 5 shows some experimental results. The last section summarizes the conclusions.

2 Dynamic Cost Models with Qualitative Variables

To incorporate the dynamic factors in a multidatabase system into a cost model, we proposed an effective qualitative approach in [18]. In this approach, we consider the combined effect of all the factors on a query cost together rather than individually. Although the dynamic factors change differently in terms of changing frequency and level, they all contribute to the contention level of the underlying system environment, which represents their net effect. Notice that the cost of a query increases as the contention level. The system contention level can be divided into a number of discrete states (categories) such as "*High Contention*" (S_H), "*Medium Contention*" (S_M), "*Low Contention*" (S_L), and "*No Contention*" (S_N). A qualitative variable is used to indicate the contention states. This qualitative variable, therefore, reflects the combined effect of the dynamic environmental factors. A cost model including such a qualitative variable can capture the dynamic factors to certain degree.

Since, for a given query, its cost increases as the system contention level, we can use the cost of a probing query to gauge the contention level and classify the contention states for the dynamic system environment. To obtain an appropriate classification of system contention states, we first partition the range of a probing query cost in the given dynamic environment into subranges (intervals) with an equal size. Each subrange represents a contention state. If some neighbor contention states are found to have a similar effect on the derived cost model, they are merged into one state. Such a uniform partition with merging adjustment procedure for a classification of contention states has been proven to be very effective in practice[18].

A qualitative variable X with M possible system contention states S_1, S_2, ..., S_M can be represented by a set of $M-1$ indicator (binary) variables Z_1, Z_2, ..., Z_{M-1}. That is, $X = S_i$ $(1 \le i \le M-1)$ is represented by $Z_i = 1$ and $Z_j = 0$ (for any $j \neq i$); and $X = S_M$ is represented by $Z_k = 0$ (for any $1 \le k \le M$). Including qualitative variable X in a cost model is equivalent to including indicator variables $Z_1, Z_2, ..., Z_M$ in the cost model.

To develop a cost model including the indicator variables, we extend our previous query sampling method in [15–17]. In other words, we use observed costs of sample queries to build a regression cost model with indicator variables as follows:

$$Y = \underbrace{(B_0^0 + \sum_{j=1}^{M-1} B_0^j Z_j)}_{intercepts} + \sum_{i=1}^{n} \underbrace{(B_i^0 + \sum_{j=1}^{M-1} B_i^j Z_j)}_{slopes} X_i, \qquad (1)$$

where Y is the query cost, X_i's are explanatory variables, Z_j's are indicator variables, and B_i^j's are the regression coefficients. The intercepts and slopes of equation (1) change from one contention state to another, indicated by the values of Z_i's. Since the above qualitative approach is obtained by introducing multiple contention states into our previous query sampling method, it is also called as the multi-states query sampling method. For more details of this method, please refer to [18].

3 Fractional Analysis Approach

One assumption made by the qualitative approach discussed in the last section is that the contention state does not change during the execution of a query although different executions of queries can be run in different contention states. This assumption is usually valid for small (cost) queries. For large (cost) queries, they may experience multiple contention states during their executions. How to estimate the cost of a query when it experiences multiple states during its execution is the issue to be discussed in this and the following sections.

There are two simple approaches to estimating the cost of a query experiencing multiple states. One is called the *single state analysis*. The idea is to ignore the changes in contention states during the execution of a query and use a dynamic cost model with a qualitative variable discussed in Section 2 together with one prevailing contention state to estimate the query cost. The prevailing contention state can be (1) the initial state in which the query is to start; (2) the median state among all states; or (3) a random state from all states. Unlike the initial state, the median and random states may not actually be experienced by the query at all. Hence the initial state may be superior in most cases for

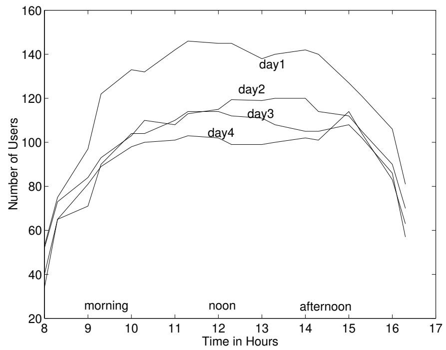

Fig. 1. System Loads in a Company

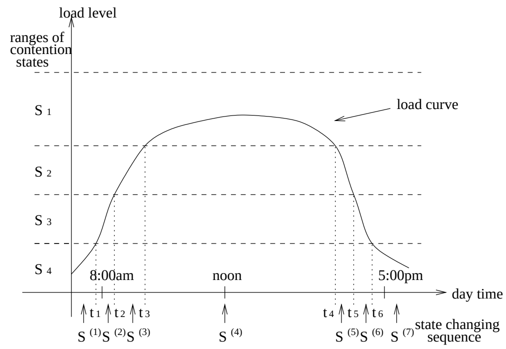

Fig. 2. A Typical Load Curve

the single state analysis approach. The advantage of this approach is that one step application of the dynamic cost model is sufficient to give a cost estimate. However, the resulting estimate may be inaccurate since not all experienced contention states are considered.

Another simple approach is called the *average cost analysis*. The idea is to take the average of costs for all the states in the environment as the cost estimate $C(Q)$ of query Q, that is, using $\bar{C}(Q) = \sum_{i=1}^{M} C(Q, S_i)/M$ to estimate $C(Q)$, where $C(Q, S_i)$ denotes the cost estimate for Q in state S_i and $S_1, S_2, ..., S_M$ are all possible states in a given environment. Although the average cost estimate is usually better than the single-state cost estimates, it may still be quite rough due to the fact that some contention states may never be experienced while other contention states may be experienced with various durations for the given query.

In this section, we are going to introduce a better cost estimation via a finer analysis, called *fractional analysis*. The key idea is to analyze a query cost by fractionalizing it according to the contention states to be experienced.

We notice that the system load in a particular application environment often demonstrates certain pattern. Fig. 1 shows the load for a system environment observed in a real-world company on different days. Clearly, the loads follow a similar pattern during every observed day in the company. The loads are minimum off working hours. The loads start to grow in the morning when the working hours begin and decline when the working hours are close to the end of the day. A curve depicting such a pattern in which the system load changes over time in an application environment[2] is called a *load curve*. Such a load curve can be obtained via calibrating the application environment under consideration. One assumption made in the following discussion is that the load curve for the given application environment is prior known.

As suggested in Section 2, the load (contention) level is divided into a number of discrete contention states (see Fig. 2), where the load level is measured by a probing query cost. Let $\Delta = \{ S_1, S_2, ..., S_M \}$ be the set of all possible contention

[2] In general, the time period for a load curve can be a day, a week, a year, or any other reasonable periodical durations. In this paper, we consider a day only.

states; $S^{(1)}, S^{(2)}, ..., S^{(N)}$ be the sequence of contention states occurred along the load curve in the given application environment, where $S^{(i)} \in \Delta$; $t^{(i-1)}$ and $t^{(i)}$ be the starting and ending times for state $S^{(i)}$ $(i = 1, 2, ..., N)$.

Consider query Q starting its execution at time $t_Q^{(s)}$ in state $S^{(k)}$. Let $C(Q, S^{(i)})$ $(i = k, k+1, ...)$ be the cost estimate for query Q if the query is executed entirely in state $S^{(i)}$.

If $C(Q, S^{(k)}) \leq (t^{(k)} - t_Q^{(s)})$, Q is expected to experience only one contention state $S^{(k)}$. Hence $C(Q, S^{(k)})$ is a good estimate of the cost for query Q, i.e., $C(Q) = C(Q, S^{(k)})$.

If $C(Q, S^{(k)}) > (t^{(k)} - t_Q^{(s)})$, query Q is expected to experience more than one contention state. Let $T^{(k)} = (t^{(k)} - t_Q^{(s)})$. Then $T^{(k)}/C(Q, S^{(k)})$ is the estimated fraction of work done for Q in state $S^{(k)}$. The remaining fraction $[1 - T^{(k)}/C(Q, S^{(k)})]$ of work for Q is to be done in the subsequent contention states. If $[1 - T^{(k)}/C(Q, S^{(k)})] * C(Q, S^{(k+1)}) \leq (t^{(k+1)} - t^{(k)})$, all remaining work of Q can be done in state $S^{(k+1)}$. The cost of Q can be estimated as: $C(Q) = T^{(k)} + [1 - T^{(k)}/C(Q, S^{(k)})] * C(Q, S^{(k+1)})$.

If $[1 - T^{(k)}/C(Q, S^{(k)})] * C(Q, S^{(k+1)}) > (t^{(k+1)} - t^{(k)})$, query Q is expected to experience more than two contention states. Let $T^{(k+1)} = (t^{(k+1)} - t^{(k)})$. Then $T^{(k+1)}/C(Q, S^{(k+1)})$ is the estimated fraction of work done for Q in state $S^{(k+1)}$, and $T^{(k)}/C(Q, S^{(k)}) + T^{(k+1)}/C(Q, S^{(k+1)})$ is the estimated fraction of work done so far (in both states $S^{(k)}$ and $S^{(k+1)}$). The remaining fraction $[1 - T^{(k)}/C(Q, S^{(k)}) - T^{(k+1)}/C(Q, S^{(k+1)})]$ of work for Q is to be done in the subsequent contention states. If $[1 - T^{(k)}/C(Q, S^{(k)}) - T^{(k+1)}/C(Q, S^{(k+1)})] * C(Q, S^{(k+2)}) \leq (t^{(k+2)} - t^{(k+1)})$, all remaining work of Q can be done in state $S^{(k+2)}$. The cost of Q can be estimated as: $C(Q) = T^{(k)} + T^{(k+1)} + [1 - T^{(k)}/C(Q, S^{(k)}) - T^{(k+1)}/C(Q, S^{(k+1)})] * C(Q, S^{(k+2)})$.

In general,

$$C(Q) = \sum_{i=k}^{m} T^{(i)} + [1 - \sum_{i=k}^{m} T^{(i)}/C(Q, S^{(i)})] * C(Q, S^{(m+1)}), \tag{2}$$

where $T^{(k)} = (t^{(k)} - t_Q^{(s)})$; $T^{(i)} = (t^{(i)} - t^{(i-1)})$ for $i \geq k+1$; m is the minimum integer such that $[1 - \sum_{i=k}^{m} T^{(i)}/C(Q, S^{(i)})] * C(Q, S^{(m+1)}) \leq T^{(m+1)}$.

Note that m cannot be determined in advance. It has to be determined during the fractional analysis. The following algorithm describes the fractional analysis procedure:

Algorithm 1 : Fractional Analysis

Input: The load curve including the contention states changing sequence $S^{(1)}$, $S^{(2)}$, ..., $S^{(N)}$ and the starting time $t^{(i-1)}$ and ending time $t^{(i)}$ for each state $S^{(i)}$ $(i = 1, 2, ..., N)$; the starting time $t_Q^{(s)}$ of query Q; the cost model $C(Q, S)$ for estimating the cost of query Q in any state S.
Output: Cost estimate $C(Q)$ for query Q.
Method:

begin
  Find the initial state $S^{(k)}$ for Q such that $t^{(k-1)} \leq t_Q^{(s)} < t^{(k)}$;
  Let $F := 0$; $C := 0$; $T := t^{(k)} - t_Q^{(s)}$; $m := k - 1$;
  while $(1 - F) * C(Q, S^{(m+1)}) \;>\; T$ **do**
    $C := C + T$;
    $F := F + T/C(Q, S^{(m+1)})$;
    $m := m + 1$;
    $T := t^{(m+1)} - t^{(m)}$;
  end;
  $C := C + (1 - F) * C(Q, S^{(m+1)})$;
  return C;
end.

Comparing the above fractional analysis with the single state analysis, we notice that, when query Q is expected to complete its execution entirely in its initial state (i.e., $C(Q, S^{(k)}) \leq (t^{(k)} - t_Q^{(s)})$), the cost estimates for Q from the fractional analysis and the initial single state analysis are identical. If such an initial state also happens to be the median (or randomly-selected) state, the query cost estimates from the fractional analysis and the median (or random) single state analysis are identical. However, in general, the execution of a large query may experience more than one contention state. Since the fractional analysis considers all the states that a query experiences, it usually gives better cost estimates than a single state analysis.

Comparing the fractional analysis with the average cost analysis, we have the following propositions:

Proposition 1. *Let $S^{(k)}, S^{(k+1)}, ..., S^{(m)}, S^{(m+1)}$ be the sequence of contention states experienced by query Q. Let I_j be the set of all indexes u's such that $S^{(u)}$ is in the sequence and $S^{(u)} = S_j \in \Delta$ for $j \in \{1, 2, ..., M\}$. Let $T_j = \sum_{u \in I_j} T^{(u)}$ (i.e., the accumulated duration[3] for Q in state S_j). If $T_1/C(Q, S_1) = T_2/C(Q, S_2) = ... = T_M/C(Q, S_M) = 1/M$ (i.e., $T_j = C(Q, S_j)/M$ for $1 \leq j \leq M$), then $C(Q) = \bar{C}(Q)$, where $C(Q)$ and $\bar{C}(Q)$ are the fractional cost estimate from (2) and the average cost estimate, respectively.*

Proof. Without loss of generality, we assume $S^{(m+1)} = S_M$. From (2), we have

$$\begin{aligned} C(Q) &= \sum_{j=1}^{M-1} \sum_{u \in I_j} T^{(u)} + \sum_{u \in I_M \;\wedge\; u \neq m+1} T^{(u)} + [1 - \sum_{j=1}^{M-1} \sum_{u \in I_j} T^{(u)}/C(Q, S_j) \\ &\quad - \sum_{u \in I_M \;\wedge\; u \neq m+1} T^{(u)}/C(Q, S_M)] * C(Q, S_M) \\ &= \sum_{j=1}^{M-1} T_j + T_M - T^{(m+1)} + [1 - \sum_{j=1}^{M-1} T_j/C(Q, S_j) - T_M/C(Q, S_M) \qquad (3) \\ &\quad + T^{(m+1)}/C(Q, S_M)] * C(Q, S_M) \end{aligned}$$

[3] Assume that $T_j = 0$ if $I_j = \emptyset$ (empty set).

$$= \sum_{j=1}^{M-1} C(Q,S_j)/M + C(Q,S_M)/M - T^{(m+1)} + [1 - (M-1)/M - 1/M$$

$$+T^{(m+1)}/C(Q,S_M)] * C(Q,S_M) = \sum_{j=1}^{M} C(Q,S_j)/M = \bar{C}(Q).$$

Therefore, the cost estimate from (2) and the average cost estimate are identical in such a case. □

Note that $T_j/C(Q,S_j)$ is the fraction of work done for Q in state S_j (during its total stay in the state if there are several visits). Proposition 1 actually states that the cost estimate from the fractional analysis is identical to the average cost of all states when query Q experiences every possible contention state in Δ at least once and completes an equal fraction $(1/M)$ of work in each of the M states. In such a case, both cost estimates are quite accurate. However, in general, a query may finish more work in one state than others, which implies that the above condition does not hold. In this case, the fractional analysis is expected to give better estimates since it considers the actual fraction of work done in each state for the query.

Proposition 2. *Assume that $C(Q,S_j) \neq C(Q,S_i)$ for some $j \neq i$. Using the same notation as in Proposition 1, if $T_1 = T_2 = ... = T_M$, then $C(Q) < \bar{C}(Q)$.*

Proof. Let $T = T_j$ $(1 \leq j \leq M)$. Note that $T_j/C(Q,S_j)$ $(1 \leq j \leq M)$ is the fraction of work done for Q in state S_j. The fractions in all states should add to 1, that is: $\sum_{j=1}^{M} T_j/C(Q,S_j) = 1$. Hence, $T = 1/[\sum_{j=1}^{M} 1/C(Q,S_j)]$. From (3), we have

$$C(Q) = M/[\sum_{j=1}^{M} 1/C(Q,S_j)]. \tag{4}$$

On the other hand, we have

$$[\sum_{j=1}^{M} 1/C(Q,S_j)] * [\sum_{i=1}^{M} C(Q,S_i)] = \sum_{j=1}^{M}\sum_{i=1}^{M} C(Q,S_i)/C(Q,S_j)$$

$$= M + \sum_{j=1}^{M-1}\sum_{i=j+1}^{M} [C(Q,S_i)/C(Q,S_j) + C(Q,S_j)/C(Q,S_i)]. \tag{5}$$

Notice that $\frac{a}{b} + \frac{b}{a} \geq 2$ for $a,b > 0$ and the equality is true only if $a = b$. Since $C(Q,S_j) \neq C(Q,S_i)$ for some $j \neq i$ is assumed, i.e., $C(Q,S_i)/C(Q,S_j) + C(Q,S_j)/C(Q,S_i) > 2$ for some $j \neq i$. Hence, from (5)

$$\sum_{j=1}^{M} 1/C(Q,S_j) * \sum_{i=1}^{M} C(Q,S_i) > M + \sum_{j=1}^{M-1}\sum_{i=j+1}^{M} 2 = M^2. \tag{6}$$

From (4), (6) and the average cost estimate formula, we have

$$\frac{\bar{C}(Q)}{C(Q)} = [\sum_{j=1}^{M} 1/C(Q,S_j)] * [\sum_{i=1}^{M} C(Q,S_i)]/M^2 > \frac{M^2}{M^2} = 1.$$

Therefore, $C(Q) < \bar{C}(Q)$. □

People might think that a query cost would be equal to the average cost for all states if the query spends the same amount of time in every state. However, Proposition 2 states that the cost estimate for a query from the average cost analysis is larger than the cost estimate for the query from the fractional cost analysis when query Q spends an equal amount of time in every state. The reason for this phenomenon is that Q runs in different states with different working rates. The higher the contention level is, the slower the working rate. Therefore, with the same amount of time, the query will complete less work in a state with a higher contention level. If all states spend the same amount of time on Q, most work of Q will be done in the states with lower contention levels. The actual cost of Q will be smaller than the average cost in such a case.

4 Probabilistic Approach

Although the fractional analysis approach in the last section can estimate costs for queries experiencing multiple contention states during their executions, one assumption made is that the load curve in the given dynamic environment is prior known and the load changes gradually. To deal with the cases with rapidly and randomly changing loads in a dynamic environment, we develop a probabilistic approach in this section.

Note that a rapidly and randomly changing load in a dynamic environment causes frequent changes in its contention states. The occurrence of a contention state is a random phenomenon and governed by laws of probability.

Let $\Delta = \{ S_1, S_2, ..., S_M \}$ be the set of all possible contention states in a dynamic environment. We consider a sequence of occurrences of contention states $\{X^{(n)}, n = 0, 1, 2, ...\}$ in the given environment as a stochastic process, where $X^{(n)}$ is a random variable taking values from Δ. $X^{(n)} = S_i$ indicates that the environment is in contention state S_i at time $t_n = t_0 + n * \delta$, where δ is the observing time interval. We notice that the probability for the next contention state $X^{(n+1)}$ taking a particular value usually depends only on the value of the present contention state $X^{(n)}$ and is independent of values of past contention states $X^{(0)}, X^{(1)}, ..., X^{(n-1)}$. For example, if the present contention state is "very busy", the next contention state is most likely to be "quite busy" or "extremely busy" regardless of past contention states. In other words, the conditional distribution of any future state $X^{(n+1)}$ satisfies:

$$\begin{aligned} & P(X^{(n+1)} = S_j \ \mid \ X^{(n)} = S_i, X^{(n-1)} = S_{k_{n-1}}, ..., X^{(0)} = S_{k_0}) \\ = \quad & P(X^{(n+1)} = S_j \mid X^{(n)} = S_i) \quad = \quad P_{ij}, \\ & \qquad\qquad \textit{for any } n \geq 0, \textit{ and } S_j,\ S_i,\ S_{k_{n-1}},\ ...,\ S_{k_0} \in \Delta. \end{aligned}$$

P_{ij} $(1 \leq i, j \leq M)$ denotes the (one-step) transition probability for the system contention state changing from S_i to S_j in the next time interval. Clearly, $P_{ij} \geq 0$ and $\sum_{j=1}^{M} P_{ij} = 1$ since the system has to be in one of the states in Δ in the next time interval. Such a stochastic process is known as a (finite) Markov chain[11].

The next issue is how to establish the transition probabilities in the Markov chain for a dynamic environment. Note that the contention state in a dynamic environment after each time interval can either remain in the same state or change to other states. However, the probability for the contention state changing to a far-away state is less than the one for it changing to a neighbor state.

Let the system contention state at time t_0 be S_i, and the system contention state at next time t_1 be S_j. Recall that a contention state reflects a set of close contention levels which are measured by probing query costs. Let L_k $(1 \leq k \leq M)$ be the center of gravity of the contention levels for contention state $S_k \in \Delta$. Let d_{ij} be the distance between L_i and L_j.

If the probability for the system contention state remaining in the same state S_i is q_i, the probability for the system contention state changing to other states from S_i will be $(1 - q_i)$. Among other states, a reasonable assumption is that the probabilities are inversely proportional to their distances to S_i. Hence,

$$P_{ii} = q_i; \quad P_{ij} = [(1 - q_i)/d_{ij}]/[\sum_{j=1}^{M} 1/d_{ij}], \qquad for\ 1 \leq i, j \leq M.$$

Parameter q_i can be calibrated via experiments. Matrix $(P_{ij})_{M \times M}$ lists all one-step transition probabilities for the Markov chain.

The probability $P_{ij}(n)$ for a contention state S_i changing to another contention state S_j after n time intervals is called an n-step transition probability for the Markov chain. For a finite Markov chain, the limit $\pi_j = \lim_{n \to \infty} P_{ij}(n)$ exists[11] and is called the limit probability of state S_j. Two interesting properties[11] of a limit probability are: (1) it is independent of the initial state (i.e., S_i) and (2) it not only represents the probability of a contention state in a Markov chain after a sufficiently large number of transitions but also represents the long-run portion of time for the Markov chain being in the state.

The limit probabilities for a finite Markov chain satisfy the following system of linear equations[11]:

$$\pi_j = \sum_{i=1}^{M} \pi_i P_{ij}, \qquad for\ j = 1, 2, ..., M,\ subject\ to\ \sum_{j=1}^{M} \pi_j = 1,$$

which can be used to determine π_j $(1 \leq j \leq M)$. All limit probabilities $\{\pi_j \mid j = 1, 2, ...M\}$ comprise a so-called long-run distribution for the Markov chain.

Since we consider, in this section, the situation in which Q is a large (cost) query and the contention states in the dynamic environment change frequently, it is expected that there are many transitions during the execution of Q. Since π_i represents the long-run portion of time for the Markov chain being in S_i, $\pi_i * C(Q)$ is the amount of cost incurred in state S_i. Hence $(\pi_i * C(Q))/C(Q, S_i)$

is the portion of work done for Q in S_i. Clearly, the portions of work done for Q in all states should add to 1, i.e., $\sum_{i=1}^{M} \frac{\pi_i * C(Q)}{C(Q,S_i)} = 1$. Solving this equation, we have

$$C(Q) = 1/[\sum_{i=1}^{M} \frac{\pi_i}{C(Q,S_i)}]. \tag{7}$$

Since $C(Q,S_i)$ can be estimated by using the dynamic cost models with qualitative variables, discussed in Section 2, formula (7) can be used to estimate the cost of query Q in the dynamic environment in which the contention states change rapidly.

Comparing the above probabilistic approach with the single state analysis approach, we notice that the cost estimate of a query given by the single state analysis approach is identical to the one given by the probabilistic approach when the limit probability for the selected single state is 1 (i.e., the limit probabilities for other states are 0). However, in general, more than one state has a non-zero limit probability. The single state analysis is, therefore, not an appropriate approach for such a rapidly changing environment.

Comparing the probabilistic approach with the average cost analysis approach, we have the following propositions:

Proposition 3. *If $\pi_1/C(Q,S_1) = \pi_2/C(Q,S_2) = ... = \pi_M/C(Q,S_M)$, then $C(Q) = \bar{C}(Q)$, where $C(Q)$ and $\bar{C}(Q)$ are the Markov cost estimate from (7) and the average cost estimate, respectively.*

Proof. Since $\pi_i/C(Q,S_i) = \pi_j/C(Q,S_j)$, we have $\pi_i = \pi_j * C(Q,S_i)/C(Q,S_j)$. From (7) and the average cost estimate formula, we have

$$\frac{\bar{C}(Q)}{C(Q)} = \frac{1}{M}\sum_{i=1}^{M} C(Q,S_i) \sum_{j=1}^{M} \pi_j/C(Q,S_j) = \frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{M} C(Q,S_i) * \pi_j/C(Q,S_j)$$
$$= \frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{M} \pi_i = \frac{1}{M}\sum_{i=1}^{M} M * \pi_i = \sum_{i=1}^{M} \pi_i = 1$$

Therefore, $C(Q) = \bar{C}(Q)$. □

Although $\pi_i/C(Q,S_i)$ does not have any physical meaning, $\pi_i * C(Q)/C(Q,S_i)$ is the portion of work done for Q in state S_i. Therefore, the assumption in the proposition implies that each state completes an equal portion of work for Q. In such a case, the cost estimates from the probabilistic approach and average cost analysis are the same. However, such an assumption is usually not met. Since the probabilistic approach can cope with different limit probability distributions, it is expected to give better cost estimates.

Proposition 4. *If $\pi_1 = \pi_2 = ... = \pi_M = 1/M$, then $C(Q) < \bar{C}(Q)$.*

Proof. Similar to the proof for Proposition 2, omitted. □

Proposition 4 states that the cost estimate for a query by the average cost analysis is larger than the cost estimate for the query by the probabilistic approach when the limit probabilities for all states are the same. This phenomenon can also be explained by the different working rates in the contention states, like the explanation given for Proposition 2.

5 Experimental Results

To validate the cost estimation techniques proposed in the previous sections, experiments were conducted using a multidatabase system prototype, named *CORDS-MDBS*[2]. Two commercial DBMSs, i.e., Oracle 8.0 and DB2 5.0, were used as component database systems running under Solaris 5.1 on two SUN UltraSparc 2 workstations. Fig. 3 shows the experimental environment. To test the techniques in various dynamic environments, we developed a load builder which can generate dynamic system loads to simulate various dynamic environments following different load curves (for the fractional analysis approach) or retention probability distributions (for the probabilistic approach) for contention states.

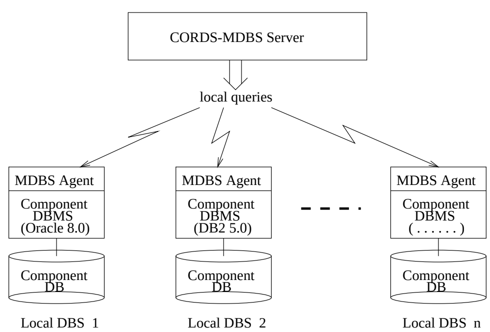

Fig. 3. Experimental Environment

The table schemas in the component databases used in the experiments were the same as those[4] in [15, 17]. More specifically, each component database contains 12 tables $R_i(a_1, a_2, ..., a_n)$ $(1 \leq i \leq 12; 1 \leq n \leq 13)$ with all integer columns. The data in the tables are randomly generated using different ranges for different columns to achieve various selectivities. The table cardinalities range from 3,000 to 250,000. Each table has some indexed columns. For more details of the test database, please refer to [15, 17].

Since the costs of unary queries are usually not large and the techniques in this paper are for large cost queries, we chose a join query class for our experiments. Following the multi-states query sampling method in Section 2, we drew a sample of queries from the query class and executed them in a dynamic

[4] The size of each table is ten times larger than that in [15, 17].

multidatabase environment. Based on the observed costs of sample queries, our cost model building tool automatically selects significant variables and uses them together with a qualitative variable (represented by a set of indicator variables) indicating system contention states to develop a cost model for the query class. Our tool also applies some statistical measures to validate the significance of the cost model. Table 1 shows the cost models developed for the query class

Table 1. Cost Models for a Join Query Class in a Dynamic MDBS Environment (RN_J — result table size; TN_{J1}, TN_{J2} — 1st and 2nd intermediate table sizes; TN_{J12} — $TN_{J1} * TN_{J2}$; RL_J — result table tuple length; L_{J1}, L_{J2} — 1st and 2nd operand table tuple lengths; TZ_{J1} — $N_{J1} * L_{J1}$; TZ_{J2} — $N_{J2} * L_{J2}$; N_{J1}, N_{J2} — 1st and 2nd operand table sizes.)

	(Dynamic) Cost Model with Qualitative Variable			
Component DBMS 1 (Oracle)	$(0.7419e{+}1 - 0.1169e{+}3 * Z_4 + 0.2748e{+}2 * Z_3 - 0.3963e{+}2 * Z_2 + 0.3626e{+}2 * Z_1) \ + \ (0.1131e\text{-}2 + 0.2212e\text{-}2 * Z_4 + 0.4383e\text{-}2 * Z_3 + 0.5517e\text{-}2 * Z_2 + 0.6588e\text{-}2 * Z_1) * RN_J \ + \ (0.4952e\text{-}3 - 0.5211e\text{-}3 * Z_4 - 0.2522e\text{-}3 * Z_3 + 0.3308e\text{-}3 * Z_2 + 0.2617e\text{-}2 * Z_1) * TN_{J1} \ + (-0.4691e\text{-}3 + 0.6974e\text{-}3 * Z_4 + 0.9641e\text{-}3 * Z_3 + 0.1900e\text{-}2 * Z_2 + 0.3194e\text{-}2 * Z_1) * TN_{J2} \ + \ (-0.1121e{+}1 + 0.5489e{+}1 * Z_4 + 0.1419e{+}2 * Z_3 + 0.7953e{+}1 * Z_2 + 0.1452e{+}2 * Z_1) * RL_J \ + (-0.1995e{+}2 + 0.3754e{+}2 * Z_4 + 0.2510e{+}2 * Z_3 + 0.3139e{+}2 * Z_2 + 0.8524e{+}1 * Z_1) * L_{J1} \ + \ (0.1647e{+}2 - 0.2275e{+}2 * Z_4 - 0.3715e{+}2 * Z_3 - 0.2881e{+}2 * Z_2 - 0.2503e{+}2 * Z_1) * L_{J2}$			
	coef. of multi. determination	std. error of estimation	avg. sample cost (sec.)	F-statistics (critical value at $\alpha = 0.01$)
	0.9992	0.1708e+3	0.1178e+4	16166.13 (> 1.25)
	(Dynamic) Cost Model with Qualitative Variable			
Component DBMS 2 (DB2)	$(-0.1529e{+}2 - 0.1923e{+}1 * Z_3 + 0.1178e{+}2 * Z_2 + 0.3510e{+}1 * Z_1) + \ (0.4171e\text{-}7 - 0.1697e\text{-}7 * Z_3 + 0.6500e\text{-}7 * Z_2 + 0.9597e\text{-}6 * Z_1) * TN_{J12} + \ (0.8934e\text{-}3 + 0.1697e\text{-}2 * Z_3 + 0.2478e\text{-}2 * Z_2 + 0.3554e\text{-}2 * Z_1) * RN_J + \ (0.2763e\text{-}3 - 0.1106e\text{-}2 * Z_3 - 0.4020e\text{-}3 * Z_2 + 0.7719e\text{-}2 * Z_1) * TN_{J1} + \ (0.1234e\text{-}4 + 0.4639e\text{-}4 * Z_3 + 0.2241e\text{-}4 * Z_2 + 0.3693e\text{-}3 * Z_1) * TZ_{J1} + \ (0.3684e\text{-}4 + 0.6292e\text{-}4 * Z_3 + 0.1345e\text{-}4 * Z_2 - 0.1768e\text{-}3 * Z_1) * TZ_{J2}$			
	coef. of multi. determination	std. error of estimation	avg. sample cost (sec.)	F-statistics (critical value at $\alpha = 0.01$)
	0.9963	0.2436e+3	0.7860e+3	5108.04 (> 1.41)

on the two component DBMSs, i.e., Oracle and DB2. The coefficient of total determination indicates that both cost models can capture over 99% variations in the query costs. The F-test also shows that both cost models are useful. Between the two models, the one for Oracle is even better.

To further validate the cost models, we ran some randomly-generated test queries[5] in the dynamic environment under the restriction that each query only experiences a (random) single contention state. We applied the dynamic cost

[5] The test queries used in this paper are the same as those in [15].

models in Table 1 to estimate the costs of the test queries and compared the estimated costs with their observed costs as well as the estimated costs using a static cost model[6]. The comparison results are shown in Fig.'s 4 and 5. From the figures we can see that the estimated costs given by the dynamic cost model are much better than the ones given by the static cost model.

However, assuming a query to experience one state may be only valid for small (cost) queries. For large (cost) queries, the execution of a query may experience more than one contention state. The techniques presented in Sections 3 and 4 should be applied to estimate the cost of such a query.

In the experiments to evaluate the effectiveness of the fractional analysis technique, the "shape" of load curve in Fig. 1 is assumed. However, for simplicity, (1) the "noon" contention state S_1 is split into two state occurrences, e.g., the contention states sequence $S_4, S_3, S_2, S_1, S_1, S_2, S_3, S_4$ was used for experiments on DB2; and (2) the load curve repeats its pattern (the contention states sequence) periodically so that the queries that cannot be finished within the current cycle can be completed within the following cycle(s). The initial starting state for each test query was randomly selected from the contention states sequence. Different changing patterns ("increase", "decrease", "equal", and "random") of time durations for the contention state occurrences along the curve were tested in our experiments. Fig.'s 6 and 7 show the experimental results for the "random" case, which well represents all other cases. The following observations can be obtained from our experiments:

- The fractional analysis technique can give good cost estimates for the test queries in a gradually and smoothly changing environment. Most cost estimates in the experiments have relative errors within 30%.
- The cost estimates given by the fractional analysis are clearly better than the ones given by the average cost analysis or the initial single state analysis for most cases.
- There are some cases (i.e., queries completed entirely within one state) in which the cost estimates from the fractional analysis and the initial single state analysis are the same, which is consistent with our theoretical analysis.
- The average costs are larger than the estimated costs given by the fractional analysis when a query spent an equal (accumulated) amount of time in every state, which is consistent with Proposition 2 we showed in Section 3.
- The accuracy of the cost model with a qualitative variable used to estimate $C(Q, S_i)$ of query Q for state S_i is important to the fractional analysis. Since we obtained a better cost model for Oracle, its fractional analysis results are overall more accurate than the ones for DB2.

In the experiments to evaluate the effectiveness of the probabilistic approach, we tested various dynamic environments with different retention probability changing patterns ("increase", "decrease", "equal", and "random"). Since an environment with an equal retention probability for all contention states may not yield an equal limit probability for every contention state, we also tested

[6] The static cost model was developed by using our static sampling query method in [15], i.e., assuming that the environment has only one state (static).

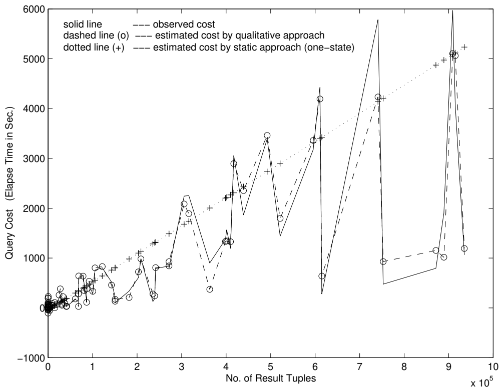

Fig. 4. Estimated Costs for Test Queries via Qualitative Approach on Oracle 8.0

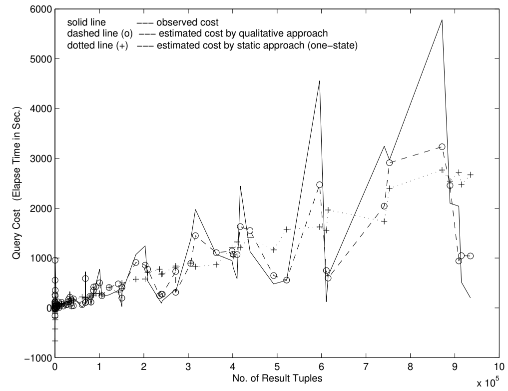

Fig. 5. Estimated Costs for Test Queries via Qualitative Approach on DB2 5.0

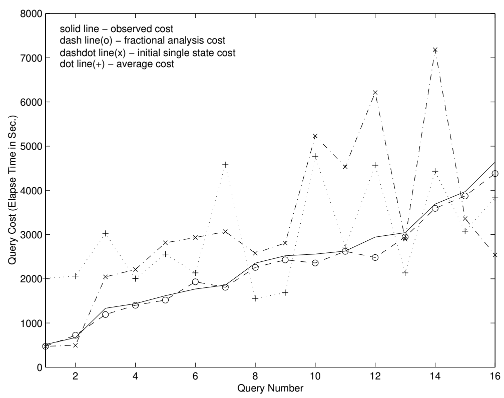

Fig. 6. Estimated Costs for Test Queries via Fractional Analysis on Oracle 8.0

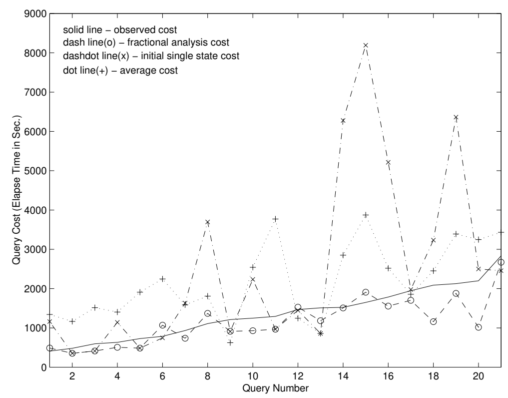

Fig. 7. Estimated Costs for Test Queries via Fractional Analysis on DB2 5.0

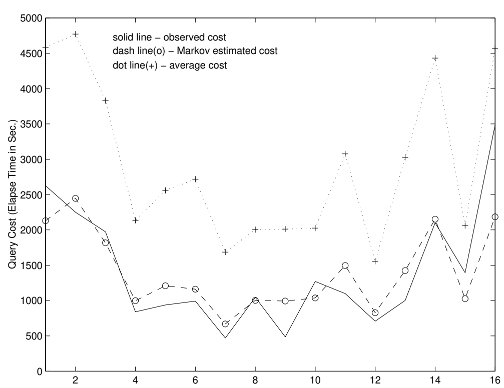

Fig. 8. Estimated Costs for Test Queries via Probabilistic Approach on Oracle 8.0

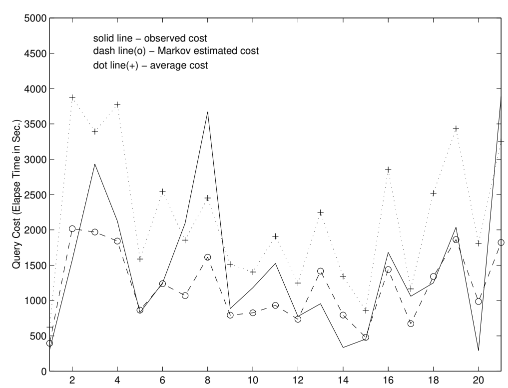

Fig. 9. Estimated Costs for Test Queries via Probabilistic Approach on DB2 5.0

the “equal” limit probability case. The experimental results for the “random” case, which well represents other cases, are shown in Fig.’s 8 and 9. From the experiments, we can obtain similar observations that we had (see above) for the fractional analysis (except that the single state analysis was not considered).

In summary, our experimental results demonstrate that the qualitative approach, fractional analysis approach and probabilistic approach comprise a promising suite of techniques in estimating local query costs for a dynamic multidatabase environment.

6 Conclusion

A major challenge for performing global query optimization in an MDBS is that some local cost information may be unavailable at the global level. Most techniques suggested so far in the literature considered only static system environments, namely, assuming the environment never changes. However, the cost of a query changes dramatically in a realistic dynamic environment.

To solve the problem, we have studied several new techniques. In our recent work[18], we employed a multi-states query sampling method to develop cost models with qualitative variables indicating the system contention states. For queries experiencing a single state, such developed cost models can be used to directly estimate their costs. To estimate the costs of (large) queries experiencing multiple states, we propose two new techniques, i.e., fractional analysis and probabilistic approach, to estimate their costs in this paper. The first one is suitable for a gradually and smoothly changing environment, while the latter is suitable for a rapidly and randomly changing environment. The cost formulas in both cases are derived. Their properties are analyzed. Note that although the two new techniques make use of the cost models developed by the multi-states query sampling method, it is not required; in other words, any method that can estimate the cost of a query in each contention state can be used together with the two techniques proposed in this paper. To validate the effectiveness of the new techniques, we conducted extensive experiments. Our experimental results demonstrate that the presented techniques are quite promising in estimating query costs in a dynamic multidatabase environment. Their produced cost estimates for most queries have relative errors within 30%.

Dynamic environmental factors were ignored in existing cost models for database systems due to lack of appropriate techniques. The work reported in this paper has shown some promise in solving the problem. However, our work is just the beginning of work needed to be done in order to completely solve all related issues.

Acknowledgements

We would like to thank Thanikachalam Ponnusamy for some study on the dynamic environments for real-world application systems. We are grateful to Amira Rahal-Arabi for her help in running some experiments. We would also like to thank Per-Åke (Paul) Larson and Frank Olken for their valuable suggestions and comments for the work reported in this paper.

References

1. S. Adali, K. S. Candan, Y. Papakonstantinou, and V. S. Subrahmanian: Query caching and optimization in distributed mediator systems. In *Proc. of ACM SIGMOD Inf. Conf. on Management of Data*, pp 137–48 (1996)
2. G. Attaluri, D. Bradshaw, N. Coburn, P.-A. Larson, P. Martin, A. Silberschatz, J. Slonim, and Q. Zhu: The CORDS multidatabase project. *IBM Systems Journal*, **34** (1995) 39–62
3. W. Du, R. Krishnamurthy, and M. C. Shan: Query optimization in heterogeneous DBMS. In *Proceedings of the 18th VLDB Conference*, pp 277–91 (1992)
4. W. Du, M. C. Shan, and U. Dayal: Reducing multidatabase query response time by tree balancin. In *Proc. of SIGMOD'95*, pp 293 – 303 (1995)
5. C. Evrendilek, A. Dogac, S. Nural, and F. Ozcan: Multidatabase query optimization. *Distributed and Parallel Databases*, **5** (1997) 77–114
6. G. Gardarin, F. Sha, and Z.-H. Tang: Calibrating the query optimizer cost model of IRO-DB, an object-oriented federated database system. In *Proceedings of the 22nd VLDB Conference*, pp 378–89 (1996)
7. C. Lee and C.-J. Chen: Query optimization in multidatabase systems considering schema conflicts. *IEEE Trans. on Knowledge and Data Eng.*, **9** (1997) 941–55
8. W. Litwin, L. Mark, and N. Roussopoulos: Interoperability of multiple autonomous databases. *ACM Computing Surveys*, **22** (1990) 267–293
9. H. Lu, B.-C. Ooi, and C.-H. Goh: On global multidatabase query optimization. *SIGMOD Record*, **21** (1992) 6–11
10. H. Naacke, *et al.*: Leveraging mediator cost models with heterogeneous data sources. In *Proc. of 14th Int'l Conf. on Data Eng.*, pp 351–60 (1998)
11. E. Parzen: *Stochastic Processes*. Holden-Day, Inc. (1962)
12. M. T. Roth, F. Ozcan, and L. M. Haas: Cost models DO matter: providing cost information for diverse data sources in a federated system. In *Proc. of the 25th VLDB Conf.*, pp 599 – 610 (1999)
13. A. P. Sheth and J. A. Larson: Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, **22** (1990) 183–236
14. Qiang Zhu and P.-A. Larson: A fuzzy query optimization approach for multidatabase systems. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, **5** (1997) 701 – 22
15. Qiang Zhu and P.-A. Larson: Solving local cost estimation problem for global query optimization in multidatabase systems. *Distributed and Parallel Databases*, **6** (1998) 373–420
16. Qiang Zhu and P.-A. Larson: Building regression cost models for multidatabase systems. In *Proceedings of the 4th IEEE International Conference on Parallel and Distributed Information Systems*, pp 220–31 (1996)
17. Qiang Zhu and P.-A. Larson: A query sampling method for estimating local cost parameters in a multidatabase system. In *Proceedings of the 10th IEEE International Conference on Data Engineering*, pp 144–53 (1994)
18. Qiang Zhu, Y. Sun, and S. Motheramgari: Developing cost models with qualitative variables for dynamic multidatabase environments. In *Proceedings of the 16th IEEE International Conference on Data Engineering (ICDE'2000)*, pp 413-424 (2000)