31 May 1982

THE OPTIMAL SELECTION OF SECONDARY INDICES IS NP-COMPLETE

Gregory Piatetsky-Shapiro

Investment Division, Strategic Information
80 Blanchard Road, Burlington, Mass 01803
and
Department of Computer Science, Courant Institute, New York University
251 Mercer street, New York, NY 10012

ABSTRACT: The problem of selecting secondary indices for a file so as to minimize the expected transaction cost was frequently analyzed before. We prove that it is NP-complete by reducing the MINIMUM SET COVER problem to it.

INTRODUCTION

The selection of the secondary indices for a file so as to minimize the expected transaction cost is a very important problem in theory and practice of relational DBMS. Specifically, given

1. Relation (file) and list of its attributes
2. Frequencies of transactions of each type (2 Transactions have the same type if their cost estimates are the same for each possible set of indices)
3. Distribution of values of each attribute : i.e. selectivity and any other statistics considered by query optimizer

we want to select the set of indices which minimizes the average transaction cost.

This problem (or its special cases) was considered by many authors. Some of them have used formal analysis in an attempt to derive the exact solution - [Palermo 70], [King 74], [Stonebraker 74], [Schkolnik 75], while others have used a heuristic approach - [Lum 71], [Hammer 76]. In particular, Schkolnik [Schkolnik 75] gives an algorithm for the solution of this problem, whose running time is $O(2^{M^{.5}})$ logM), where M is the number of domains to consider; Hammer and Chan [Hammer 76] present a good heuristic algorithm, which has polynomial running time but is not guaranteed to find the optimal solution or to be within some percentage of the optimal solution. This paper, by showing that the problem is NP-complete, will probably skew the further effort in this area toward heuristic algorithms, rather than algoritms attempting to find the optimal solution.

TRANSFORMATION OF MINIMUM SET COVER INTO OPTIMAL INDEX SELECTION

Clearly, the problem of optimal selection of secondary indices for a file is in NP, since the transaction cost estimation function runs in time which is at most polynomial in the number of attributes. To show that the problem is NP-complete, we will reduce to it the following well-known NP-complete problem :

MINIMUM SET COVER (see [Garey 79], [Karp 72]).

INSTANCE: Collection C of N subsets of a finite set S, positive integer K <= |C|

QUESTION: Does C contain a cover for S of size K or less, i.e. a subset C' of C with K or less elements such that

$$\bigcup_{S_i \text{ in } C'} (S_i) = S$$

Given an instance of this problem, we transform it into the following instance of the optimal index selection:

INSTANCE:

1. A relation R, with N attributes $A_1, A_2, \ldots, A_N$.
2. Frequencies of each transaction type:
   The only transactions we have are queries and updates - there are no deletions or insertions. We specify the frequencies by mapping collection of subsets in MINIMUM COVER into a collection of transactions on R. The frequencies can be trivially abstracted from the actual counts of transactions of each type. The relational query language, loosely defined below, is based on relational calculus.
   a. Queries. The general form of a query is:

      FOR condition LIST all attributes

      The CONDITION is a conjunction of equality predicates of the type Attribute = *constant* . Subset Si corresponds to the queries having predicate

      A_1 = <value> .

      So, if a point in S belongs to S_1, S_2, S_3 then it corresponds to the query with condition:

      A_1 = <value1> AND A_2 = <value2> AND A_3 = <value3>

   b. Updates. The general form of an update is:

      FOR condition SET A_1 = <value1>, ..., A_k = <valueK>

      For each set S_1 we select a point and make it an update on A_1. If this point was already an update on some other attributes, then we just add an update on A_1 to other updates. In this way we have selected exactly ONE update for

each attribute.

3. Distribution of data values:
   All attributes have the same selectivity, which is equal to 1/NR , where NR = number of records = number of tuples. That means that on the average only one tuple will satisfy any equality predicate and thus, assuming appropriate ratio between cost of index access and data file access, at most one index will be used in resolving any query.

4. Estimating transaction cost:
   We are considering only single - attribute indices (with such a low selectivity multiattribute indices have no advantages).

   a. The cost of Query consists of the cost of evaluating the condition and the cost of outputting all attributes. The latter cost we can disregard, it being the same regardless of indices available. The former cost, measured in number of records accessed, is NR, if sequential scan is used. If index access is used then, since for any index we expect only one data record to satisfy the condition, the cost is 1.

   b. The cost of update is sum of 3 components:

      (Cost of evaluating the condition) +
      (Cost of updating data records) +
      (Cost of updating indices if there are any)

      The first component is the same as for query. The second component is the same, regardless of the choice of the indices, so we can ignore it in the process of index selection. Thus cost estimate of an update can be considered to be equal to

      (cost estimate of the query) +
      COST_INDEX_UPDATE * (Number of updated attributes) .

      We can safely assume that the latter cost is the same for all indices and is less than the difference between the cost of sequential scan and the cost of the indexed access. This is a reasonable assumption, considering the enormous cost of the sequential scan.

QUESTION: Find the set of indices which minimizes the average cost of database use (hereafter *The COST*) under the given Transaction frequency distribution.

PROOF OF NP-COMPLETENESS

Without writing exact cost formulas we observe that adding new index increases *The COST* by COST_INDEX_UPDATE . However, if there is at least one query which can be resolved by index access on the new index and which has to be resolved by the sequential scan without it, then *The COST* is decreased by at least COST_SEQ_ACCESS - COST_IND_ACCESS which is more than COST_INDEX_UPDATE . Hence *THE COST* will decrease overall.

Therefore it is clear that the collection of attributes (A_{I_1}, A_{I_2}, ..., A_{I_m}) indices on which

minimize *The COST* is is the collection with the minimal number of attributes such that the union of corresponding subsets of S is equal to S.

Thus if we want to solve the MINIMUM COVER problem, then we transform it to the corresponding OPTIMAL INDEX SELECTION problem and solve the latter. If the number of indices in the optimal index set is more than K, then there is no cover for S of K subsets or less. If the number of indices is less than K then the sets S_{1_1}, S_{1_2}, etc.. corresponding to the indexed attributes A_{1_1}, A_{1_2}, etc.. form the desired cover. Therefore the problem of optimal index selection is NP-Complete, Q.E.D.

A related problem is the optimal index selection when updates are made off-line. Then the cost of database use should be minimized under the condition that the cost of updates does not exceed certain constant; however the latter is not included in the cost of database use . This problem is also NP-complete . The proof is similar and also uses reduction of MINIMUM COVER to the index selection.

ACKNOWLEDGEMENTS

I wish to thank Dr. Boris Khazanov for his constant support and prof. Malcolm Harrison for his warm encouragement.

REFERENCES

[Garey 79] Garey, M. R., and Johnson, D. S.
Computers and Intractability.
W. H. Freeman and Co. , 1979.

[Hammer 76] Hammer, M. M., and Chan, A. Y.
Index selection in a self-adaptive database management system.
Proceedings ACM-SIGMOD , 1976.

[Karp 72] Karp, R. M.
Reducibility among combinatorial problems.
Complexity of Computer Computations, R. E. Miller and J. W. Thatcher (eds.), Plenum Press , 1972.

[King 74] King, W. F.
On the selection of indices for a file.
IBM Research RJ 1341, San Jose , January, 1974.

[Lum 71] Lum, V. Y., and Hong, H.
An optimization problem on the selection of secondary keys.
ACM Proc. National Annual Conference , 1971.

[Palermo 70] Palermo, F.
A Quantative approach to the selection of secondary indexes.
IBM Research RJ 730, San Jose , July, 1970.

[Schkolnik 75] Schkolnik, M.
The optimal selection of secondary indices for files.
Information Systems 1, 1975.

[Stonebraker 74]
Stonebraker, M.
The choice of partial inversions and combined indices.
Int. J. of Computer and Information Sciences 3(2), 1974.