Heuristic Algorithms for Designing a Data Warehouse with SPJ Views *

Spyros Ligoudistianos  Timos Sellis  Dimitri Theodoratos  Yannis Vassiliou

Department of Electrical and Computer Engineering
Computer Science Division
National Technical University of Athens
Zographou 157 73, Athens, Greece
{spyros,timos,dth,yv}@dblab.ece.ntua.gr

Abstract. A Data Warehouse (DW) can be abstractly seen as a set of materialized views defined over relations that are stored in distributed heterogeneous databases. The selection of views for materialization in a DW is thus an important decision problem. The objective is the minimization of the combination of the query evaluation and view maintenance costs. In this paper we expand on our previous work by proposing new heuristic algorithms for the DW design problem. These algorithms are described in terms of a state space search problem, and are guaranteed to deliver an optimal solution by expanding only a small fraction of the states produced by the (original) exhaustive algorithm.

1 Introduction

A Data Warehouse (DW) can be seen as a set of materialized views defined over distributed heterogeneous databases. All the queries posed to the DW are evaluated locally using exclusively the data that are stored in the views. The materialized views have also to be refreshed when changes occur to the data of the sources. The operational cost of a Data Warehouse depends on the cost of these two basic operations: query answering and refreshing. The careful selection of the views to be maintained in the DW may reduce this cost dramatically. For a given set of different source databases and a given set of queries that the DW has to service, there is a number of alternative sets of materialized views that the administrator can choose to maintain. Each of these sets has different refreshment and query answering cost while some of them may require more disk space than the available in the DW. The Data Warehouse design problem is the selection of the set of materialized views with the minimum overall cost that fits into the available space.

Earlier work [8] studies the DW design and provides methods that generate the view selections from the input queries. It models the problem as a state space search problem, and designs algorithms for solving the problem in the case of SPJ relational queries and views.

* Research supported by the European Commission under the ESPRIT Program LTR project "DWQ: Foundations of Data Warehouse Quality"

Mukesh Mohania and A Min Tjoa (Eds.): DaWaK'99, LNCS 1676, pp. 96-105, 1999
© Springer-Verlag Berlin Heidelberg 1999
1.1 Related Work

Many authors in different contexts have addressed the view selection problem. H. Gupta and I.S. Mumick in [2] use an A* algorithm to select the set of views that minimizes the total query-response time and also keeps the total maintenance time less than a certain value. A greedy heuristic is also presented in this work. Both algorithms are based on the theoretical framework developed in [1] using AND/OR view directed acyclic graphs. In [3] a similar problem is considered for selection-join views with indexes. An A* algorithm is also provided as well as rules of thumb, under a number of simplifying assumptions. In [10], Yang, Karlapalem and Li propose heuristic approaches that provide a feasible solution based on merging individual optimal query plans. In a context where views are sets of pointer arrays, Roussopoulos also provides in [7] an A* algorithm that optimizes the query evaluation and view maintenance cost.

1.2 Contribution and Paper Outline

In this paper we study heuristic algorithms for the DW design problem. Based on the model introduced in [8, 9] we introduce a new A* algorithm that delivers the optimal design. This algorithm prunes the state space and provides the optimal solution by expanding only a small fraction of the whole state space. We also present two variations of the heuristic function used in A*, a ‘static’ and a ‘dynamic’ heuristic function. The dynamic heuristic function is able to do further pruning of the state space. To demonstrate the superiority of the A* algorithm, we compare it analytically and experimentally with the algorithms introduced in [8].

The rest of the paper is organized as follows. In Section 2 we formally define the DW design problem as a state space search problem providing also the cost formulas. In Section 3 we propose a new A* algorithm that delivers an optimal solution for the DW design problem. Improvements to the A* algorithm are proposed in Section 4. Section 5 presents experimental results. We summarize in Section 6.

2 The DW design problem

We consider a nonempty set of queries $Q$, defined over a set of source relations $R$. The DW contains a set of materialized views $V$ over $R$ such that every query in $Q$ can be rewritten completely over $V$ [4]. Thus, all the queries in $Q$ can be answered locally at the DW, without accessing the source relations in $R$. By $Q^V$, we denote a complete rewriting of the query $Q$ in $Q$ over $V$.

Consider a DW configuration $C = < V, Q^V >$ [8, 9]. We define:
- $E(Q^V)$ : The sum of the evaluation cost of each query rewriting $Q^V_i$ in $Q^V$ multiplied by the frequency of the associate input query $Q_i$,
- $M(V)$ : The sum of the view maintenance cost of each view in $V$,
- $S(V)$ : The sum of the space needed for all views in $V$,