Bi-Objective Optimization Based on Compromise Method for Horizontal Fragmentation in Relational Data Warehouses

Mohamed Barr

***Abstract*—Generally, research that dealt with the selection problems for optimization techniques or structures in relational data ware houses supports these problems by considering only a single criterion of optimization. The optimization criteria may be the response time of query execution, the number of inputs/outputs between the main memory and the disk, the space allocated to store the index or materialized views, or the number fragments required by the administrator of the data warehouse when using the fragmentation technique. The present work deals with the problem of selecting the horizontal fragmentation technique while considering both the number of I/O between memory and disk during decisional queries and the number of fragments, as two objective functions to minimize. To reduce the scope of choice solutions, we are based on a scalar method, called compromise method. The method is complemented by the principle of Pareto front to infer the best solutions. The study has been experimented on APB1 benchmark of data warehouse.**

***Index Terms*—Data warehouse, optimization, multiobjective, pareto.**

I. Introduction

In [1], the administrator of the data warehouse must find a compromise between the number of fragments generated following the fragmentation scheme that he has chosen and the number of I/O performed during the execution of decisional query.

In some work, namely that based on Genetic Algorithm, the administrator of the Data Warehouse introduced the number of fragments as a constraint optimization [2]. Another work based on Ant colony algorithm which treats the problem of selecting horizontal fragmentation as a knapsack problem and assumes that the number of fragments reflects the number of items to put in the sack to do not be exceeded [3]. Work based on the K-means method limits the explosion of the number of fragments by exploiting the requirement of the method [4].

But few studies have taken the optimization of techniques and structures in datawarehouses as multiobjective problems, we cite as an example the work that uses the MOGA algorithm (Multi Objective Genetic Algorithm) which interested by resolving the selection of materialized views as multiobjective problem [5].

The aim of this work is to consider both optimization criteria of horizontal fragmentation which are the number of I/O between memory and disk during decisional queries, and the number of fragments.

The paper is organized as follows. Section II recalls the art state of horizontal fragmentation in relational data warehouses. The key concepts used in the field of multiobjective optimization and an overview of the multiobjective methods used, is the subject of Section III. The details of our contribution and algorithms used are explained in Section IV. Discussion of experimental results is given in Section V. The last section offers concluding remarks and future perspectives.

Manuscript received March 9, 2013; revised May 28, 2013.

Mohamed Barr is with ESI (Ex. INI), Algiers, Alegria (e-mail: m_barr@esi.dz).

II. Horizontal Fragmentation in Relational Data Warehouses

A. Definition

In [4], the technique of horizontal fragmentation was defined as follows: The horizontal fragmentation is to divide a data set into several partitions, called fragments, so that the combination of fragments covers all data sources without addition or loss of information. Horizontal fragmentation can be classified in two versions: primary and derived [2].

The primary horizontal fragmentation of a relationship is done with simple predicates defined on a data set of the same relationship.

The derived horizontal fragmentation from a horizontal table is to partition the table according predicates defined on another data table [1].

B. The Complexity of the Process of Fragmentation

In [2], it was demonstrated that the management and maintenance of derived horizontal fragments is almost impossible if all possible combinations are retained. Indeed, if *Mi* is the number of fragments of the dimension table *Di*, and *K* is the number of dimension tables fragmented, then the total number of fragments of the fact table is:

$$N = \prod_{i=1}^{K} Mi \ .$$

In this respect, we recall that the technical based on construction predicates and on affinities do not support the explosion phenomenon of the fragments number. [4]

Against, approaches based on data mining that use k-means can limit the number of fragments [4].

In [2], the authors used the genetic algorithm and they took into account in their work the problem of explosion of the fragments number.

C. Approaches for Horizontal Fragmentation

In the literature, we find several approaches used to select a

horizontal fragmentation scheme: 1) approaches based on the construction of predicates, 2) approaches based on affinities, and 3) approaches based on data mining. 4) and 5) approaches based on metaheuristics.

D. Approaches Based on Predicates

The principle of these approaches comes to identify subsets of the predicates contained in the dimension tables provided that these predicates guarantee minimality (disjunction of pairs fragments obtained) and completeness (possibility of rebuilding a relationship using the union of all its constituent fragments). These subsets are then used for the fragmentation derived [4].

E. Approaches Based on Affinity

Further selection of simple predicates; these approaches are based on the use of frequencies of the workload including the same predicates to construct fragments.

F. Approaches Based on Data Mining

These approaches exploiting data mining algorithms and intelligent data analysis to select a fragmentation pattern [4]. In [4], the author shows that his approach has been proven for the selection of data structures helping to improve the performance of a management system databases. In the context of horizontal fragmentation technique, several studies have been developed to support this technique basing on data mining [4].

We can also mention the approach of classification by k-means algorithm which aims to obtain fragments answering queries according to common characteristics [4].

The general conclusion after all the studies and methods used to support horizontal fragmentation technique in relational data warehouses, shows the absence of methods that consider both the two objectives which are to minimize the number of fragments generated and to reduce number of I/O.

III. Multiobjective Optimization

In the daily life of individuals and societies, many problems must be seen along several points of view. We cite for example the means of transport the least polluting and which must ensure a minimum time and cheaper cost of transport as possible. Taking into account the traveler in these conditions, we are in situation of an embarrassment of choices between the worst to the ideal. Hence, walking is the possible solution without pollution and little cost, as; we can take the plane to get faster but with significant pollution. In this case, the train is the most suitable [6].

Based on the previous example, we can say that the multi-objective optimization is concerned with problems in which we want to model a problem in order to satisfy several objectives at once.

A. Mathematical Modeling

The modeling of a multiobjective optimization problem can be written in the following form:

Optimize (maximize or minimize) $f(\overrightarrow{x})$ (function to be optimized) with $\overrightarrow{g(x)} \leq 0$ (m inequality constraints) and $\overrightarrow{h(x)} = 0$ (p equality constraints). We have $(\overrightarrow{x}) \in R^n$, $(\overrightarrow{g})(\overrightarrow{x}) \in R^m$, and $\overrightarrow{h}(\overrightarrow{x}) \in R^p$.

In a multiobjective optimization problem, we have no longer a single goal, but we have several goals at once. Multiobjective optimization problem is to optimize the "best" overall objectives.

Often we encounter conflicting objectives in other words: the objective of a reduction causes an increase in the other objective [7].

B. Methods of Multiobjective Optimization

In the literature, several methods have been developed to support multi-objective optimization problems, namely:

- Scalar methods,
- Interactive methods,
- The fuzzy methods,
- Operating a metaheuristic methods
- Methods for decision making [7].

IV. Contribution

In our contribution, we will use a scalar method called compromise method responsible to optimize an objective function while considering the second function as a constraint. The principle of the method is to transform a multiobjective problem into single objective one under additional constraints.

The approach of the compromise method is structured as follows:

-Choosing an objective to optimize in priority;

-We keep the objective taken as priority and transform other objectives in inequality constraints [7].

In our case study, we are interested in minimizing the number of I/O between memory and disk during execution of decisional queries using the technique of horizontal fragmentation while setting each time the number of fragments needed. We are based on the Genetic Algorithm [2].

A. Used Algorithm of Compromise

Inputs:
*N*Max = Maximum Number of fragments
For *i* from *N* Min to *N* Max (Minimal Number)
Do
Find solutions for all objective functions that minimize the number I/O between memory and disk
*N*Max = *N*Max-1
End
Apply the algorithm on the entire Pareto solutions (NBF, NBIO) where NBF is the number of fragments and the number
NBIO = Number Of Inputs / Outputs

Pseudo Algorithm of Constraints ξ

V. Experiments

A. Used Benchmark

To implement our method, we used the APB1 benchmark of data warehouse shown in Fig. 1, whose the conceptual in star model is following:

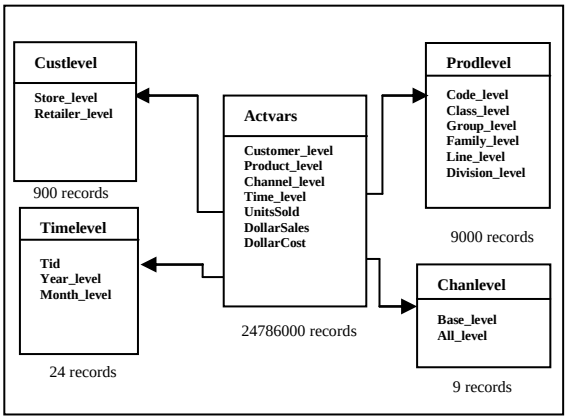

Fig. 1. Conceptual model used benchmark APB1.

B. Results Obtained for the Two Objective Functions

To realize the meaning of compromise of our multiobjective optimization method, we have collected the results obtained using Genetic Algorithm that gives whenever the number of inputs/outputs according to the number of fragments introduced [2].

Indeed, we have ξ initialized to 128, and we got all pairs of solutions (NBF, NBIO). We note that for a ξi we can find many solutions as we find no solution. The results are prepared in the following Table I:

TABLE I: NUMBER OF I/O DEPENDING ON THE NUMBER OF FRAGMENTS

Number of fragments	Number of I/O	Number of fragments	Number of I/O
128	28076276	64	28816454
128	26435075	60	30429886
128	26129505	60	30094147
128	26119638	60	29000600
120	29687000	60	28397907
120	27963561	56	32642337
112	27110018	56	29816180
112	26976264	56	27649530
108	26089942	48	31335328
96	28338210	48	29255171
96	26327400	24	32827583
84	30396994	24	31960943
84	27247547	20	33949678
80	29066588	16	33500005
80	28202075	14	37610517
72	29860054	14	36662852
72	29069423	12	34435604
72	28520612	10	36633668
72	27573685	8	40920737
72	27301473	6	42879738
72	26058587	4	45010562
64	28971765	2	50962061

C. Discussion of the Results

According to the results shown in the table above, we can say that if we consider only the number of fragments to be minimized, we go directly to the fragmentation scheme which generates two fragments because it contains the minimum number of fragments and provides the most degraded in the number of I/O which is equal to 50962061. But if we focus only on the number of inputs / outputs between memory and disk, we choose probably the minimum number 26058587 I/O that goes with 72 fragments.

This observation justifies that in multiobjective optimization, we have to accept the concept of compromise if objectives are often contradictory.

But there is a solution that meets both objectives at once? Which is this solution ?

To achieve this, we have to go through another concept which is dominance.

D. Dominance

Solving a multiobjective optimization problem leads to a large number of solutions that are not all optimal. The way to guide us to choose one or several solutions is based on the concept of compromise solutions obtained on optimizing certain objectives and degrading performance for others. A solution is interesting if it satisfies a relation of domination between it and the rest of the obtained solutions [7].

Definition of the dominance relation

We say the vector $\overrightarrow{x_1}$ dominates the vector $\overrightarrow{x_2}$ if:

- $\overrightarrow{x_1}$ is at least as good as $\overrightarrow{x_2}$ in all the objectives, and
- $\overrightarrow{x_1}$ is strictly better than $\overrightarrow{x_1}$ in at least one objective.

The points that dominate all other and do not dominate between themselves represent the optimal solutions in the Pareto sense. These solutions belong to rank 1 of domination [7].

E. Algorithm Used

CurrentRank $= 1$, $m = N$
(N is the number of points of the set on which comparisons are made.)
While
$N \neq 0$
Do
For i from 1to m
Do
If Xi is not dominated
then
Rank(Xi, t) = CurrentRank
End If
End For
For i from 1 to m
Do
If
Rank(Xi, t) = CurrentRank
then
Ranger Xi in a temporary population,
$N = N - 1$
End If
End For
CurrentRank = CurrentRank $+ 1$, $m = N$
End While

Pareto rank assignment algorithm

In [7], the following algorithm is used to assign the Pareto rank.

For an optimal Pareto, we have programmed the previous algorithm, and we came to results prepared in the following Table II.

TABLE II: CLASSIFICATION OF SOLUTIONS IN THE SENSE OF PARETO RANKING

Rank of the solution	Number of fragments	Number of I/O
1	72	26058587
	56	27649530
2	60	28397907
	72	27301473
3	72	27573685
	108	26089942
	48	29255171
	96	26327400
	60	29000600
	64	28816454
	84	27247547
4	64	28971765
	56	29816180
	72	28520612
	80	28202075
	112	26976264
5	112	27110018
	72	29069423
	80	29066588
	60	30094147
	128	26119638
	96	28338210
	48	31335328
6	120	27963561
	128	26129505
	72	29860054
	60	30429886
7	128	26435075
	56	32642337
	120	29687000
	84	30396994
8	128	28076276

Theoretically, the optimal solution for both objectives simultaneously is represented by the two points of rank 1.

That is to say, the two pairs (72, 26058587) and (56, 27649530) which respectively represent the two objectives at a time, which are: the number of fragments and the number of I/O (inputs/outputs) between main memory and disk.

F. The surface of compromise or Pareto Front

The surface of compromise (or Pareto front) is formed from the solutions obtained and classified as Tier 1 based on the definition of dominance.

To illustrate the concept of compromise surface, schematically the following example, which uses the minimization of two objective functions f_1 and f_2 under the constraints $\vec{g}(\vec{x}) \leq 0$ and $\vec{h}(\vec{x}) = 0$

-*S* is the set of values of couples ($f_1(\vec{x})$, $f_2(\vec{x})$) when $(\vec{x})$ respects constraints $\vec{g}(\vec{x})$ and $\vec{h}(\vec{x})$.

-*P* is the compromise surface.

The surface of compromise is shown in the following Fig. 2.

In [8], the Pareto front is defined as "the border between the space of solutions", feasible and infeasible and compromise is the set of non-dominated solutions called Pareto-optimal, characterized by the following principle: it is impossible to a better solution on a criterion without being worse on at least one other criterion.

In our case, the Pareto front consists of two points circled in green in Fig. 3. These points are those that belong to the rank 1of respective coordinates for f_1 and f_2 (72, 26058587) and (56, 27649530).

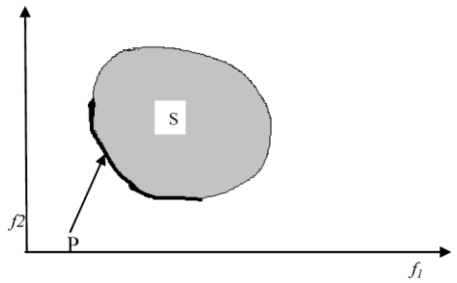

Fig. 2. Representation of the compromise surface [7].

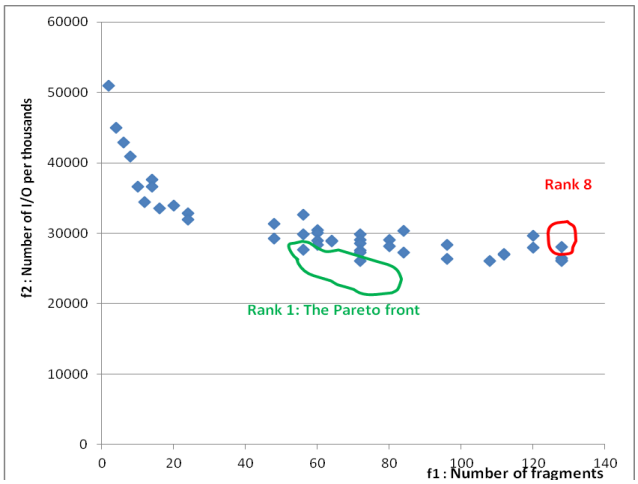

Fig. 3. Graph representing the Pareto front or solutions of rank 1.

VI. CONCLUSION

Our study had as objective the use of an optimization method for multiobjective problem to help the administrator of the data warehouse to find a compromise between two different objectives, namely the number of fragments reasonable and the number of inputs/outputs required which reduces the response time when handling decision queries. The practical method of compromise scalar in our case and use the concept of Pareto dominance has limited the scope of the administrator's choice in a solution of two parts, something which helps reduce query execution time and at the same time having a reduced number of fragments which facilitates the administration task.

This work can be extended to other methods of multiobjective optimization that support other preferences of the administrator.

REFERENCES

[1] L. Bellatreche, " Utilisation des vues materialisées, des index et de la fragmentation pour la conception logique et physique d'un entrepot de données ," Doctoral thesis, Clermont-Ferrand 2 University, December 2000.

[2] L. Bellatreche and K. Boukhalfa "Sélection de schéma de fragmentation horizontale dans les entrepôts de données : formalisation et algorithmes," research report, LESI/ENSMA and University of Laghouat Algeria, 2006.

[3] M. Barr and L. Bellatreche, "A new approach based on ants for the resolution of horizontal fragmentation in relational data warehouses," *ICMWI 2010*, Algiers, October 2010.

[4] H. Mahboubi, "Optimisation de la performance des entrepôts de données XML par fragmentation et répartition ," Doctoral thesis, Lumière University Lyon 2, December 2008.

[5] M. Lawrence, "Multiobjective genetic algorithms for materialized view selection in olap data warehouses," Faculty of Computer Science Dalhousie University, Halifax, NS, Canada, 2006.

[6] T. Grabener, “Calcul d’itinéraire multimodal et multiobjectif en milieu urbain,” Doctoral thesis, Institute for Computer Research in Toulouse, France, 2010.
[7] Y. Collette and P. Siarry, “ Optimisation multiobjectif, ” Electricité de France, Clamart et l’Université de Paris XII Val-de-Marne, 2002.
[8] C. Vella and R. Rabaylade, “ Compromis entre efficacité énergétique et confort: une méthode d’optimisation multicritère des paramètres d’un bâtiment prenant en compte les incertitudes,” Lyon University, Department of Civil and Building Engineering, March 2011.

Mohamed Barr was born on August 22, 1963 in Oum-El-Thiour, El-Oued, Algeria. He received his Bachelor degree in Mathematics in 1983 and Magister in Computer Science-Information and Knowledge Systems in 2010. Now he is a Ph.D. student at ESI, El-Harrach, Algiers, Algeria. He is also a state engineer in Computer science, option: Information Systems. He got his Post-Graduation Specializing (PGS) diploma in management of information systems, organized by the Algerian Oil, petroleum Company (Sonatrach) and Perpignan University–France, in 2004. He presented some research papers at international conferences held in Algeria, China, Korea, and Denmark and published in journals (IEEE, JCIT – AICIT, IACSIT). The main purpose of his research is the optimization of Data Warehouses administration. Since 1995, He has worked as an executive in the Algerian Oil Company (Sonatrach), where he studied and developed several information systems.