



## Bi-Objective Optimization Method for Horizontal Fragmentation Problem in Relational Data Warehouses as a Linear Programming Problem

Mohamed Barr, Kamel Boukhalifa & Karima Bouibede

To cite this article: Mohamed Barr, Kamel Boukhalifa & Karima Bouibede (2018) Bi-Objective Optimization Method for Horizontal Fragmentation Problem in Relational Data Warehouses as a Linear Programming Problem, Applied Artificial Intelligence, 32:9-10, 907-923, DOI: [10.1080/08839514.2018.1519096](https://doi.org/10.1080/08839514.2018.1519096)

To link to this article: <https://doi.org/10.1080/08839514.2018.1519096>



Published online: 05 Oct 2018.



Submit your article to this journal [↗](#)



Article views: 480



View related articles [↗](#)



View Crossmark data [↗](#)



Citing articles: 2 View citing articles [↗](#)



# Bi-Objective Optimization Method for Horizontal Fragmentation Problem in Relational Data Warehouses as a Linear Programming Problem

Mohamed Barr<sup>a</sup>, Kamel Boukhalifa , and Karima Bouibede<sup>c</sup>

<sup>a</sup>École nationale Supérieure d'Informatique (ESI), Alger, Algérie; <sup>b</sup>Laboratoire (LSI), Département informatique, Université des Sciences et de la Technologie Houari Boumediene (USTHB), Alger, Algérie;

<sup>c</sup>Faculté d'électronique et informatique, Université des Sciences et de la Technologie Houari Boumediene (USTHB), Alger, Algérie

## ABSTRACT

In this work, we relied on a particular exact method to solve NP-Hard problem of determining a horizontal fragmentation scheme in relational data warehouses. The method used is that of linear programming which is distinguished by other methods by the existence of practical methods that facilitate the resolution of problems that may be described in linear form. We quote the Simplex method and the interior points. To meet the linearity of the objective function and constraints, we used initially "De Morgan" theorem, which is based on properties of sets to transform and optimize decision queries, from any form to a linear one.



In addition to designing and solving the selection problem of horizontal fragmentation technique, we considered the problem in two simultaneous objectives, namely: the number of Inputs/Outputs needed to run the global workload, and number of fragments generated to identify the best solutions compared to the concept of Pareto dominance.

In addition, to carry out our work, we used the Benchmark APB1 invoked by a workload, to achieve satisfactory results.

## Introduction

Data warehouses as databases designed to contain decision-making information in relation to the themes of business professions do not cease to increase over time and take up volumetry. In addition, to extract the information from these data warehouses, two difficulties arise, namely: the gigantic number of information, plus the complexity of the decision-making queries used. Because they incorporate selection simple predicates, join operations between fact table and dimension tables, as well as aggregation and sorting operations.

The literature on optimization structures and techniques in relational data warehouses classifies the problem of selecting a horizontal fragmentation

**CONTACT** Mohamed Barr  [m\\_barr@esi.dz](mailto:m_barr@esi.dz)  École nationale Supérieure d'Informatique (ESI), B.P. 68M, 16270 Oued-Smar, El Harrach, Alger, Algérie

Color versions of one or more of the figures in the article can be found online at [www.tandfonline.com/uaai](http://www.tandfonline.com/uaai).

scheme (HFS) that is the subject of our research in the category of NP-Difficile problems. In Boukhalifa (2009), the author has proved, by reduction towards the problem 3-partitions, that the problem of FH is NP-Difficult in the strong sense.

Research on the problem of selecting an SF has evolved over time. The first works consider it a non-difficult problem and propose deterministic solutions (Özsu and Valduriez 2011; Ceri et al. 1982). In these works the number of final fragments is not taken into account and is considered as an output parameter. The second generation of work considers the number of fragments generated known as input and proposes greedy algorithms to find a better fragmentation scheme (Bellatreche et al. 2000).

The third generation considers the selection of an HFS as an optimization problem and generally takes into account the overall cost of executing a workload as an objective function (Boukhalifa 2009). This work considers the number of fragments generated as a constraint of the problem where an upper bound is to be fixed by the administrator.

This work proposes meta-heuristics to find a quasi-optimal HFS generating a number of fragments less than or equal to the upper bound. Nevertheless, this work does not give any indication on how to set this threshold or the most interesting values, hence the need to propose new approaches allowing, in addition to the performance of the solution, to generate an optimal fragment number. It is in this context that we consider the problem of SFH as a problem of multi-objective optimization which aims to minimize the response time as well as the number of fragments generated.

The simultaneous consideration of the two optimization objectives, which are: (i) the number of inputs/outputs and (ii) the number of fragments, makes the problem of selecting a fragmentation scheme more complex. To solve our bi-objective optimization problem, we need to use the multi-objective optimization methods known from the literature. Among these methods, we find that based on the concept of dominance used to classify the set of solutions of the two objectives according to the Pareto rank. According to this method, the dominant solutions belonging to the first ranks are the most favorable.

In this work, in addition to the problem of selecting a fragmentation scheme following the simultaneous optimization of two objective functions, we succeeded in formalizing the problem of selecting a HFS as an optimization problem following linear programming (LP) where the objective function and the constraints are expressed in a linear manner using the decision variables represented by the selection simple predicates contained in the workload. We recall that the solution of the problem of horizontal fragmentation in data warehouses based on an exact method such as LP that uses

practical methods well known in the literature has allowed us to infinitely reduce the complexity of the problem even for important instances.

Experiments carried out on the APB1 benchmark gave very satisfactory results.

## Work organization

In Part II, we briefly recalled some theoretical formalism on a linear program, via Part III, we presented our contribution beginning with the distinction between two forms of queries, a form which directly operates in linear form and a another that requires a transformation using the formula of “De Morgan” to make the entire workload in a linear program. In the fourth part, experiments were carried out at the base of Benchmark APB1 with bi-objective resolution of horizontal fragmentation problem. Comments on the results obtained have been some analyzes. Finally, in Part V, we have drawn some conclusions to reach certain perspectives.

## Mathematical formulation of a linear program

### Defining a linear program

By definition, a linear program is formalism whose decision variables are in real type.

Let  $\{x_1, x_2, \dots, X_n\}$  be a set of  $n$  real variables, and  $Z$  a linear objective function to be optimized (min or max):

$$z = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

Linear constraints (equalities and inequalities)

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &\leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &\leq b_2 \\ \dots & \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &= b_m \end{aligned}$$

### Solution of a problem in linear programming (LP)

Find a solution to a problem designed as a LP is to assign variables that meet the constraints. A solution is optimal if it maximizes (maximizes or minimizes) the objective function.

## Contribution

The basis of our contribution is to use a method of solving an NP-Complete problem using an exact method based on LP. The first work is the formalization of our problem in linear forms of the objective

function and constraints. In Collette and al Optimisation Multi objectif (2002), the authors recall that effectively treat of LP where the objective function and the constraints are linearly expressed in terms of decision variables. But in practice, the situations encountered often have several complications, distorting the use of these methods: as the non-linearity of the objective function.

In a second place, and after writing the objective function and the constraints in a linear form, we apply the Simplex method on the optimization problem of the horizontal fragmentation technique. Finally, we introduce the concept of dominance through bi-objective optimization which considers both the number of fragments generated and the number of inputs/outputs obtained in connection with each fragmentation scheme.

Our contribution is recorded in an optical support for horizontal fragmentation problem as a LP problem that addresses the gap that exists in the literature that deals with this subject and has a great rarity. In addition, to solve the problem by taking into account more than one objective at once (Özsu and Valduriez 2011; Barr and Bellatreche 2010; Boukhalfa 2009; Mahboubi 2008; Bellatreche Et and Boukhalfa 2005; Aouiche, et al., 2004) (Figure 1).

### **General form of a decisional query**

Q is a query that queries a data warehouse consists of a fact table  $F$  joined to several dimension tables  $D_i$ , following a star schema.

So the query Q can be written as follows:

$Q^1$ : Select  $C_1, C_2, \dots, C_m$ , statistical operator (\*)  
 from FactT FT, TDimension<sub>1</sub> DT<sub>1</sub>, TDimension<sub>2</sub> DT<sub>2</sub>, ..., TDimension<sub>d</sub> DT<sub>d</sub> Where ANPK<sub>1</sub> = Val<sub>1</sub> and ANPK<sub>2</sub> = Val<sub>2</sub> and ... ANPK<sub>d</sub> = Val<sub>d</sub> and FT.FK<sub>1</sub> = DT<sub>1</sub>. PK<sub>1</sub> and FT.FK<sub>2</sub> = DT<sub>2</sub>. PK<sub>2</sub> and ... and FT.FK<sub>d</sub> = DT<sub>d</sub>. PK<sub>d</sub> Group by  $C_1, C_2, \dots, C_m$   
 Order by  $C_1, C_2, \dots, C_m$

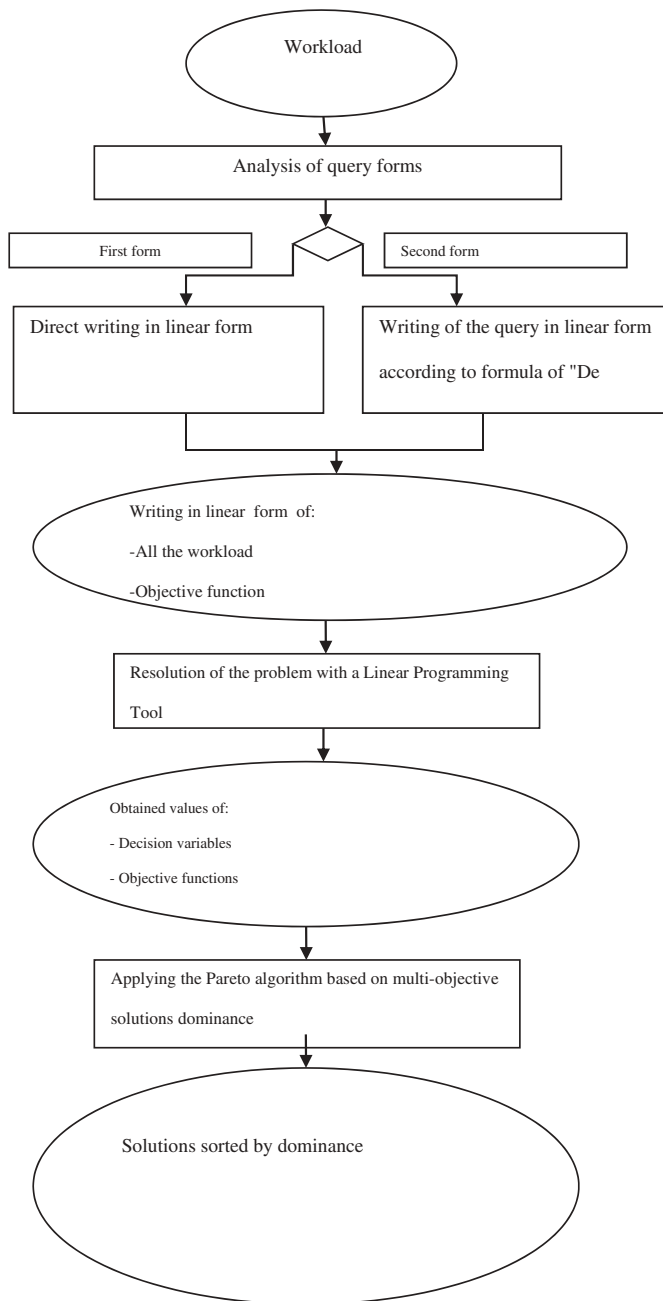
### **Different forms of queries**

In our context, to formalize the horizontal fragmentation problem as an optimization problem expressed using LP, we distinguish two types of queries.

In the result of this work, we will use the example of Benchmark APB1 as data warehouse.

#### **First form**

In this first form, a query Q contains restriction in part one or more predicates on a single attribute.



**Figure 1.** Overall scheme of our contribution.

### Example

```

Select Time_level, count (*)
ACTVARS from A, P PROLEVEL
Where

```

```

A.PRODUCT_LEVEL = P.CODE_LEVEL and
P.Class_LEVEL = A1DGFSPTJ473
Group by Time_level
Charging the query Q1
Cost (Q1) = Cost (Selection (Q1)) + Cost (Join (Q1));
Cost (Selection (Q1)) = (Selectivity (P1) * | F | * L) / PS
Cost (Join (Q1)) = 3 * (| Actvars | + | Prodlevel |)

```

To bring up the targeted linearity, the Q<sub>1</sub> query can be written based on the predicate P<sub>1</sub>, as follows:

$$Q_1 = a_1 * P_1 \text{ with } a_1 = (\text{Selectivity}(P_1) * |F| * L) / PS + 3 * (|Actvars| + |Prodlevel|)^2$$

### **Second form**

In the second form, the restriction part of the application we find a set of conjunctions of several attributes.

### **Example**

```

Select Customer_Level, Channel_level, Time_level,
sum (dollarcost)
From ACTVARS A PROLEVEL P, T TIMELEVEL
Where A.PRODUCT_LEVEL = P.CODE_LEVEL and
A.TIME_LEVEL = SUBSTR (T.TID, 1, 6) and
T.YEAR_LEVEL = 1996 and P.CLASS_LEVEL = ZYXHAYKT707N
Group by Customer_level, Channel_level, Time_level

```

The main idea in our work returns to separate the conjunction of two predicates T.YEAR\_LEVEL = 1996 and P.CLASS\_LEVEL = ZYXHAYKT707N, and the two joints A.PRODUCT\_LEVEL = P.CODE\_LEVEL and A.PRODUCT\_LEVEL = P.CODE\_LEVEL

In this second form, we used the “De Morgan” formula, and the relation algebra shown in Equation (4), then we estimated linearly the conjunction of predicates in accordance with Equation (12) below mentioned.

### **“De Morgan formula”**

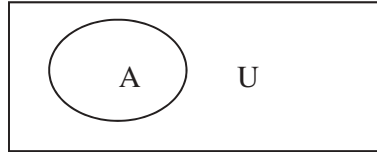
#### **Definition: additional sets**

#### **Definition**

U is the universal set. The complement of the set A, denoted  $\bar{A}$  is the set of U elements that do not belong to A. In other words, the complement of the set A is the difference U-A (Niefield and Rosenthal 1985).

$$\bar{A} = \{x|x\} \tag{1}$$

Given two subsets A and B of the universe U (Figure 2).



**Figure 2.** Scheme sets.

The law of “Morgan” has the following properties (Niefield and Rosenthal 1985)

The law of “De Morgan” checks the following properties (Niefield and Rosenthal 1985):

$$\overline{A \cap B} = \bar{A} \cup \bar{B} \quad (2)$$

$$\overline{A \cup B} = \bar{A} \cap \bar{B} \quad (3)$$

**Another property of relational algebra used**

$$A \cap B = (A \cup B) - ((A - B) \cup (B - A)) \quad (4)$$

**Objective function**

The objective function to be optimized in this case amounts to maximizing the difference between the overall<sup>3</sup> workload that interrogates the data warehouse before and after the fragmentation.

**Profit of a query**

The benefit of a query is equal to the difference of its charge (cost of input/output) before fragmentation of the data warehouse and that after fragmentation including both restriction and join costs.

**Profit of the global workload.** The overall benefit of the workload is equal to the sum of the profits of all weighted queries based on query execution frequencies respectively.

**Linear equations of the objective function**

First case: First query form

$$\begin{aligned} Cost(Q_i) &= (Cost(Selection(Q_i, P_j)) + Cost(Join(Q_i, P_j))) * P_j \\ Cost(Q_i) &= a_{ij} * P_j \end{aligned} \quad (5)$$

$a_{ij}$ : coefficients of the predicate  $P_j$  in the query  $Q_i$



### Transformation of the second form queries using the “De Morgan” law

Knowing that the selectivity of a conjunction of several simple predicates of selection is the product selectivities of these predicates, so to have a weighted sum of predicates based on profits related respecting the formalism above mentioned, it should be to go through a transformation to a weighted sum in linear form. To achieve this, we rely on the “De Morgan” formula described above, and we apply the famous formula expressed in Equation (4), to achieve the general form shown in Equation (6), we went through several steps that have given us the linear expression of Equation (12).

$$Q = a * P_1 + b * P_2 + c \text{ New linear general form of the query } Q \quad (6)$$

In Bellatreche and Boukhalfa (2005), the cost of restriction part contained in a decisional query can be expressed as follow:

$$CR\_BF\_F = (|F| * L)/PS \text{ Cost of Restriction part before Fragmentation} \quad (7)$$

In Aouiche (2005), the cost of using a hash join between two tables,  $F$  and  $D$ , is given according to the following equation.

$$CJ\_BF\_F = 3 * (|F| + |D|) \text{ Cost of joining part before Fragmentation} \quad (8)$$

$$CJ\_BF\_F = 3 * (|F| + |D|) \quad (9)$$

Cost of joining part before Fragmentation

$$Pr\ ofit(Q_i, Join) = 3 * ((|F| + |D|) - (|F_j| + |D_j|)) * P_j$$

$$\text{Cost of Joining part After Fragmentation depending on predicate } P_j \quad (10)$$

$$Pr\ ofit(Q_i, Restriction) + Pr\ ofit(Q_i, Join)$$

$$Pr\ ofit(Q_i) = f_i * \left( \sum_{j=1}^n (|F| * L)/PS - (Selectivity(P_j) * |F| * L)/PS \right) + 3 * ((|F| + |D|) - (|F_j| + |D_j|)) * P_j$$

where  $n$  is the number of selection predicates in conjunction and  $f_i$  is the frequency of the query  $Q_i$

$P_j$  : Simple selection predicate

The  $n$  selection predicates in this case belong to the same attribute.

$$(11)$$

Based on the “De Morgan” Equation (3) and other properties of the relational algebra, we successful in the linear expression of the second form of queries, as follows:

$$\begin{aligned} & \Pr ofit(Q_i, Restriction) + \Pr ofit(Q_i, Join) \\ \Pr ofit(Q_i) &= f_i * (2n - 1) * (\sum_{j=1}^n (Selectivity(P_j) * |F| * L) / PS) + 3 * \\ & ((|F| + |D|) - (|F_j| + |D_j|)) * P_j \end{aligned} \quad (12)$$

where  $n$  is the number of selection predicats in conjunction and  $f_i$  is the frequency of the query  $Q_i$

$P_j$  : Simple selection predicate

The  $n$  selection predicates in this case belong to different attributes.

### Decisions variables

The principle of a linear program is determining the values of the decision variables that optimize the objective function. Where does the interest of the definition of decision variables in our case the different selection simple predicates contained in the workload. A predicate is equal one (1) if it participates in the fragmentation scheme, and it is equal zero (0) otherwise.

### Constraints

The first constraint we can admit hypothetically is the participation of selection simple predicates or not in different fragmentation schemes.

$$P_j \in \{0, 1\} \quad (13)$$

$$\sum_{c=1}^k P_c \leq NP \quad (14)$$

$$N = \prod_{i=1}^k M_i \quad (15)$$

$M_i$  is the number of fragments of the dimension table  $D_i$ , and  $K$  is the number of dimension tables fragmented, then the total number of fragments of the fact table is  $N$ .

We note here that we have considered for each simple predicate of selection a correspondent sub-domain of the candidate attribute.

### Bi-objective method used

In this work, we have limited to the use of a popular method which belongs to the so-called scalar methods. This method is called Epsilon-constraints.

### **Epsilon-constraints method**

The method allows transforming a multi-objective optimization problem into a mono-objective optimization problem with some additional constraints.

The approach is as follows:

- We choose an objective function to optimize as a priority;
- An initial constraint vector is selected;
- The problem is transformed by maintaining the priority objective and by transforming the other objectives in terms of inequality constraints (Collette and al Optimisation Multi objectif 2002).

### **Presentation of the method**

We start with the problem  $P$ .

It is assumed that the priority objective function is the rank 1 function.

We choose a constraint vector.

Mathematically the method is formalized as follows:

$$\left\{ \begin{array}{l} \text{Minimize } f_1(\vec{x}) \\ f_i(\vec{x}) \leq \xi_i, i = 2, \dots, k \text{ and } \xi_i \leq 0 \\ \text{S.t } g(\vec{x}) \leq 0 \\ \text{and } h(\vec{x}) = 0 \end{array} \right.$$

as  $\xi_i$  is an upper bound for the  $i$ th objective (Collette and al Optimisation Multi objectif 2002).

In our study, we have taken the number of input/output between memory and disk during the execution of decisional queries, as the proprietary function, and the number of fragments as constraints.

## **Experiments**

The objective function defined above is to be maximized because it represents the difference between the cost of a query invoking a non-fragmented data warehouse and the fragmented one, and less the input/output cost relative to the fragmented data warehouse more the difference is maximum.

### **Benchmark used**

To carry out our work, it is necessary to place the method on a relational data warehouse (the Benchmark APB1 in our case) and collect the results to verify the effectiveness of the method and discuss the results.

Benchmark APB1 used is modeled after a star schema as in [Table 1](#).

**Table 1.** Characteristics of the APB1 benchmark.

Table	Type	Primary key	Foreign key	Tail
Acvars	Facts			24786000
Prodlevel	Dimension	Code_level	Product_level	9000
Custlevel	Dimension	Customer_level	Store_level	900
Chanlevel	Dimension	Chanal_level	Base_level	9
Timelevel	Dimension	Tid	Time_level	24

### **Workload used**

We used a workload of 275 queries that covers a set of 150 simple predicates of selection belonging to all nine attributes other than the primary keys of dimension tables.

We generated 150 different fragmentation schemes, and assessed the performance of each solution. The results are prepared in Table 3. We note that the rate of reduction in number of input/output is proportional to the number of fragments. Moreover, that more the scheme covers a large number of predicates, the better is the result. The best scheme obtained in terms of the number of inputs/outputs is minimal one that matches the torque number of fragments and the number of inputs/outputs respectively (78939800, 516499200).

We note that the results reported in column (Number Input/Output) reflect maximizing the difference of workload execution on the data warehouse before fragmentation and after the fragmentation.

### **Operating environment**

The experiments were performed on a brand MSI Laptop, Intel® Core processor (™) i7 CPU 2.20 GHz, 8G RAM, OS Under Windows 7, 64 bit.

The solver used is lpsolve 5.5.2.2 IDE, based on the Simplex method.

First, we involved all selection predicates contained in the workload. Then, to prune one or more predicates according to their participation in the maximization of the objective function defined in Equation 5 following different equations mentioned above, we proceeded with the change in the terminal number of predicates participants decrementing each time the number and view the cost of workload in number of inputs/outputs (Figure 3).

### **Performance of the method based on linear programming**

#### ***The optimum of a linear program, if it exists, is formed at least at a vertex of the polyhedron***

Knowing that in the simplex method, the optimal solution passes through the vertices of the polyhedron, it will be interesting to enumerate the number of possible vertices. Theoretically for  $m$  constraints and  $n$  variables the number

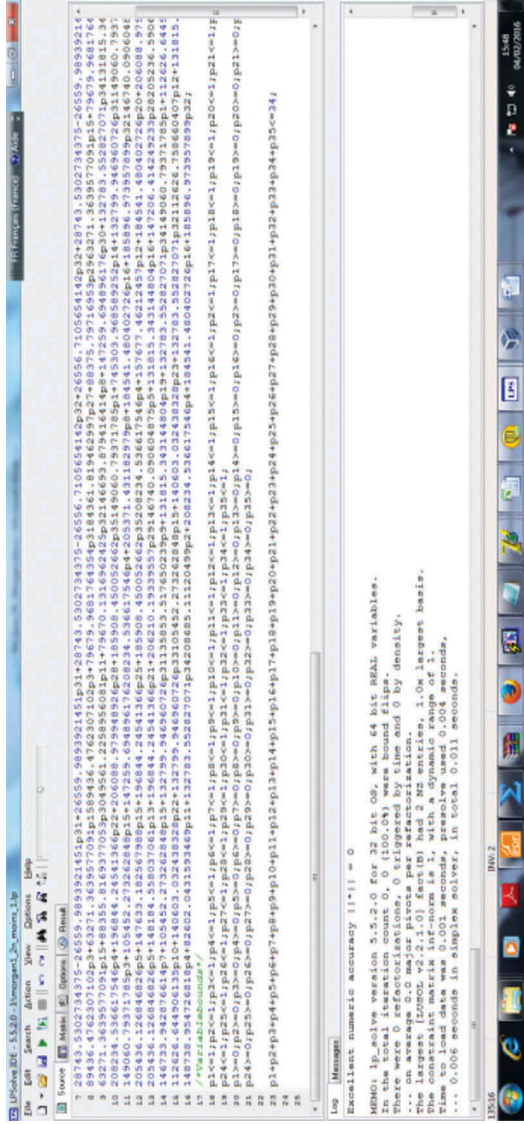


Figure 3. Interface of linear programming tool used (IDE Ipsolve 5.5.2.0).

of vertices rises to  $C_{m+n}^n = \frac{(m+n)!}{n!m!}$  which represents a digit which grows with  $m$  and  $n$  in a very coarse way. If we consider, for example, a problem under Linear Program of  $m = 30$  constraints and  $n = 50$  variables, we will have  $8.871412534840452e + 21$  possible vertices. In addition, if we have a machine that processes 10 million vertices per second, it will take 282083.478 centuries to treat us these vertices! (Saaty 1955)

### ***Applying the simplex method on our case***

To highlight the performance of LP to solve a linear problem in a quick way, we made a comparison between the three methods used for pruning or remember simple predicates selection of participating or not in the HFS. The first method is based on ants' algorithm that uses both essential elements to choose a beneficial predicates that are pheromone and visibility. The second method is a deterministic method which proceeds by scanning the fact table to identify collections of predicates which are grouped together to store them finally in the same fragments (Barr, Boukhalfa, and Bouibede 2016) (Table 2).

### ***Achievements***

According to data collected on APB1Benchmark and under cost model previously defined. We weighted the decision variables (predicates in our formalization) by profits related to the objective function with respect to all 275 queries of the Workload. Moreover, each time, depending predicate selected by the solver, we deducted the number of fragments using Equation (14). The results in Table 3 show that the objective function is optimal in function of the increasing number of predicates in the fragmentation schemes.

### ***Dominant fragmentation schemes according to Pareto sense***

After obtaining the set of solutions by considering the objective of Input/Output as a priority function and the number of fragments as constraint, we have introduced the set of pairs (number of inputs/outputs, number of fragments) according to the dominance concept using an Algorithm which classifies the solutions according to the ranks (Collette and al Optimisation Multi objectif 2002) The "Pareto Front" is the solution that dominates all solutions.

**Table 2.** Comparison of performance between the three methods.

Average time pruning		
Ants algorithm	Deterministic method	Linear programming
0.46153846 s	01 s	0.013 s

**Table 3.** Achievements.

Output/ input #	Fragments #	Output/ input #	Fragments #	Output/ input Fragments	Output/ input #	Fragments #	
10975800	2	53275900	92400	67130400	25945920	75624000	77474880
13326500	3	53850700	184800	67360200	26943840	75739000	83931120
15677100	4	54411800	203280	67589900	27941760	75854000	90387360
18027500	5	54873400	304920	67819600	28939680	75969000	96843600
20378000	5	55333400	609840	68049400	29937600	76084000	103299840
21757700	10	55793400	914760	68279100	30935520	76199000	118056960
23137400	15	56253300	1143450	68508800	31933440	76314000	132814080
24496000	20	56711200	1334025	68738500	32931360	76429000	147571200
25794600	40	57169100	1524600	68968300	33929280	76544000	163968000
27093100	60	57627000	1715175	69198000	34927200	76659000	180364800
28391600	80	58084900	1905750	69427700	35925120	76773900	196761600
29609000	100	58542700	2096325	69657200	36923040	76888900	213158400
30826400	120	58996100	2515590	69886700	37920960	77003900	229555200
32043700	140	59449500	2934855	70116200	38918880	77118900	245952000
33261100	160	59902900	3354120	70345700	39916800	77233800	262348800
34478500	180	60355900	3773385	70575200	40914720	77348800	278745600
35625800	360	60796600	5031180	70804700	41912640	77463500	295142400
36599700	400	61229500	5488560	71034100	42910560	77578000	313588800
37573600	440	61662300	5488560	71263600	43908480	77692400	332035200
38547500	440	62080300	7318080	71493100	44906400	77806900	332035200
39465500	660	62498100	9147600	71722600	45904320	77920200	350481600
40370800	880	62915200	10977120	71952100	46902240	78033600	368928000
41276200	1100	63321000	10977120	72181600	47900160	78146900	387374400
42181400	1320	63683100	10 977120	72411000	48898080	78260200	405820800
42986200	1540	63913100	11975040	72640500	49896000	78373500	424267200
43790900	3080	64143100	12972960	72870000	50893920	78486800	442713600
44594100	3520	64373100	13970880	73099400	51891840	78600100	461160000
45397200	3960	64603100	14968800	73328900	52889760	78713400	479606400
46200300	4400	64832900	15966720	73558400	53887680	78826600	498052800
46992600	6600	65062700	16964640	73787800	54885600	78939800	516499200
47784800	8800	65292400	17962560	74017300	55883520	79052900	516499200
48576500	11000	65522200	18960480	74246800	56881440	79165700	516499200
49266500	13200	65751900	19958400	74476200	57879360	79270300	590284800
49953500	26400	65981700	20956320	74705700	58877280	79374600	664070400
50640600	39600	66211400	21954240	74935100	59875200	79478900	737856000
51327400	52800	66441200	22952160	75164600	60873120	79581900	737856000
52014200	66000	66670900	23950080	75394000	60873120		
52700900	79200	66900700	24948000	75509000	71018640		

For each sub-set of predicates that participates in a fragmentation scheme, we computed the number of fragments generated using Equation (15) under APB1 benchmark. Table 3 illustrates the couples (number of input/output in system pages, number of fragments).

The course of the classification algorithm of the two objective functions: the number of inputs/outputs and number of fragments allowed us to identify three ranks:

- The first rank which forms the “Pareto Front” in the sense of dominance and contains two solutions: (62915200, 10977120) and (79165700,

516499200). These solutions reduce the overall cost of the load to 87% and 89%, respectively.

- The second contains eight solutions that are (20378000,5), (38547500,440), (61662300, 5488560), (75394000,60873120), (77806900,332035200), (79581900, 737856000), (63321000, 10977120), and (79052900, 516499200).
- The last rank contains the rest of the solutions (Figure 4).

## Conclusions and perspectives

In conclusion, we can say that the most important point for the use of an accurate method such as LP is the success to design the problem treated in linear form.

In our case regarding the optimization of a workload based on the relational algebra, using the properties of sets allowed us to transform decisional queries in linear forms depending on the decision variables expressing the simple predicates of selection.

Compared to methods based on meta-heuristics such as that based on ants, or the one designated in a deterministic manner, the LP method shows an interesting opportunity in the minimal time for pruning bad candidates predicates involved in determining good HFS. The practical results show that the time taken by the ant algorithm respectively and the one set by the deterministic method is 36 and 77 times slower compared to that used by the linear program.

The bi-objectives view of the optimization problem of data warehouses using horizontal fragmentation technique allowed us to limit the solutions of the first rank Pareto to automatically assist the administrator of the data warehouse in two solutions preferred for a set of 150 solutions.

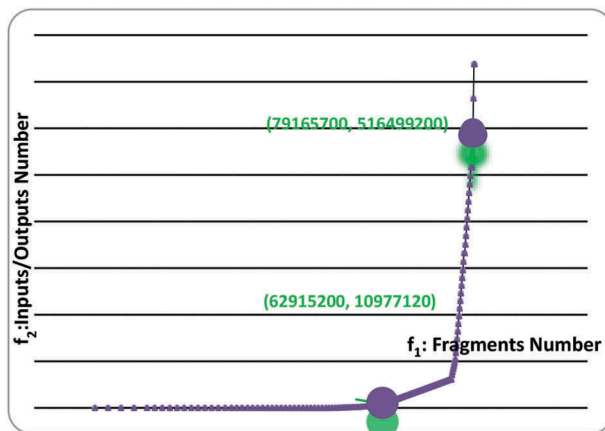


Figure 4. Pareto front (the two shaded points).



The classification of the different pairs of solutions according to rank, offers the data warehouse administrator a view on the choices in a more conscious way. To clarify this conclusion, we can favor the solution of the second rank represented by the couple (38547500, 440) which reduces the overall cost of the workload to 78%, and requires only 440 fragments.

From the same point of view, we note that the solutions of the first rank and which dominate all the other solutions, present very high fragments numbers. A situation that forces data warehouse administrators and researchers to push further reflection on a distributed solution of the scheme belonging to the “Pareto Front”. That is the first perspective.

As a second perspective, we can combine horizontal fragmentation with other structures or techniques such as Binary Join Index, to improve more solutions and helps to reduce the number of fragments.

## Notes

1. ANPK: Attribut Not Primary Key, FK: Foreign Key, PK: Primary Key.
2.  $|F|$ ,  $L$ , and  $PS$  respectively represent the size of the fact table, the record length of fact table, and the size of a system page.
3.  $A - B$  means, elements belong to the set  $A$  and don't belong to the set  $B$ .

## ORCID

Kamel Boukhalfa  <http://orcid.org/0000-0002-9746-579X>

## References

- Aouiche, K. 2005. *Techniques de fouille de données pour l'optimisation automatique des performances des entrepôts de données*. Université Lumière Lyon 2, École Doctorale de Sciences Cognitives, France: thèse de doctorat.
- Aouiche, K., J. Darmon, and O. Boussaid. 2004. *Sélection automatique d'index dans les entrepôts de données*. Laboratoire ERIC Université Lumière Lyon 2 École Doctorale de Sciences Cognitives: rapport de recherche.
- Barr, M., K. Boukhalfa, and K. Bouibede. 2016. Méthode Déterministe pour la Fragmentation Horizontale des Entrepôts de Données. In *10ième édition de la Conférence ASD'2016 Avancées des Systèmes Décisionnels*. 14–16 May Annaba University: Algeria.
- Barr, M., and L. Bellatreche. 2010. *A new approach based on ants for the resolution of Horizontal fragmentation in Relational Data warehouses*. Algiers: ICMW12010. Oct.
- Bellatreche, L. and K. Boukhalfa. An Evolutionary Approach to Schema Partitioning Selection in a Data Warehouse Environment, 7th International Conference on Data Warehousing and Knowledge Discovery (DAWAK'05)(3589), edited by Lecture Notes in Computer Science (LNCS), August, Springer-Verlag, 2005.
- Bellatreche, L., K. Karlapalem, M. K. Mohania, and M. Schneider, What can partitioning do for your data warehouses and data marts?”, IDEAS '00 Proceedings of the 2000

- International Symposium on Database Engineering & Applications, Pages 437–46, September 2000.
- Boukhalfa. 2009. *De la Conception Physique aux Outils d. thèse de doctorat.* ENSM: Administration et de Tuning des Entrepôts de Données.
- Ceri, S., M. Negri, and G. Pelagatti. Horizontal data partitioning in database design.;1982 ACM SIGMOD International Conference on Management of Data, pages 128–36,1982.
- Collette and al Optimisation Multi objectif. 2002. *Clamart et l' Electricité de France:* Université de Paris XII Val-de-Marne. doi:10.1044/1059-0889(2002/er01).
- Özsu and Valduriez. 2011. *Principles of Distributed Database Systems.* New York: Springer. Third Edition.
- Mahboubi, H. 2008. Optimisation de la performance des entrepôts de données XML par fragmentation et répartition. PhD in Computer Science: University of Lyon 2. Dec.
- Niefeld, S. B., and K. I. Rosenthal. 1985. 'Strong de Morgan's Law and the Spectrum of a Commutative Ring', Department of Mathematics, Union College, Schenectady, New York 12308. Communicated by Saunders MaeLane, Received July 25, 1985, Journal of Algebra 93, 169-181. doi:10.1016/0021-8693(85)90181-4
- Saaty, T. L.. 1955. The Number of Vertices of a Polyhedron. *The American Mathematical Monthly* 62 (5):326–331. doi:10.1080/00029890.1955.11988636. Taylor & Francis, Ltd. on behalf of the Mathematical Association of America.