STUDIA UNIV. "BABEŞ-BOLYAI", INFORMATICA, Volume XLIV, Number 1, 1999

STOCHASTIC OPTIMIZATION FOR JOIN OF THREE RELATIONS IN DISTRIBUTED DATABASES II. GENERALIZATION AND MORE APPLICATIONS

VIORICA VARGA

ABSTRACT. Most of the literature devoted to the topic of query optimization in distributed databases addresses the problem of finding a deterministic strategy for assigning the component joins of a relational query to the processors of a network which can most efficiently execute the joins and can most economically perform the required interprocessor data transfers. The capacity of distributed systems for concurrent processing motivates the distribution of the database in a network. There is a different approach to query optimization if the system is viewed as one, which receives different types of queries at different times and process more than one query at the same time. The multiple-query problem is not deterministic; the multiple-query input stream constitutes a stochastic process. The strategy for executing the multiple-query is distributed over the sites of the network as a probability distribution. This kind of query optimization problem was solved for a single-join and a multiple-join of three relations, for relations stored at two sites. The problem of multiple-join leads to a special type of nonlinear programming problem, which we presented in [14]. In this article we give the general model for the join of three relations in sequential and parallel execution mode. These problems leads to the same type of nonlinear programming problem.

1. INTRODUCTION

The query optimization problem for a single query in a distributed database system was treated in great detail in the literature, see [3], [4], [10], [15].

For each new type of query that arrives at the system, a new optimal strategy is determined. A distributed system can receives different types of queries and processes them at the same time. In this case the determination of the optimal query processing strategy is a stochastic optimization problem. Query processing strategies may be distributed over the processors of a network as probability distributions. The "decision variables" of the stochastic query optimization problem are the probabilities that a component operator of the query is executed at a particular site of the network.

In [8] the authors extend the state-transition model proposed by Lafortune and Wong [9] and the original multiprocessing model of [7] and [6]. The main objective of the model is to give query-processing strategies, which are globally optimal. In [8] is presented the stochastic query optimization model for a single-join, the sequential and the parallel execution of a single-join and a multiple-join of three relations, where the component relations are stored in two site of the distributed database system.

In paper [14] we presented the stochastic model for the join of three relations, which are stored at three different sites. This stochastic query optimization problem leads to a nonlinear programming problem, which is specific one. In [14] we give an algorithm to solve this nonlinear programming problem. In this article we present general models, containing sequential and parallel operation of two queries of type join of three relations. These leads to the same type of nonlinear programming problem as the problem from [14], which we solve for different cases and the results are presented in tables. From the examples we can see the global optimality of the stochastic query optimization model.

One of the main objectives of distributed systems is the use of the database as shared resource. Different transactions can execute different types of queries against the same database at the same time. We consider two modes of multiple-query operation: sequential and parallel operation.

2. Sequential operation

Let be the following two queries:

$$Q_1 = A \bowtie B \bowtie C, Q_2 = D \bowtie E \bowtie F$$

where $A \cap B \neq \emptyset$; $B \cap C \neq \emptyset$ $D \cap E \neq \emptyset$; $E \cap F \neq \emptyset$ and the relations A and D are stored at site 1, the relations B and E at site 2 and the relations C and F at site 3. So the initial state of relations referenced by the query Q_3 in the three-site network is the next column vector:

$$x_0 = \begin{pmatrix} A, D \\ B, E \\ C, F \end{pmatrix}$$

where the i-th component of the vector x_0 is the set of relations stored at site $i (i = 1, 2, 3)$ at time $t = 0$. The initial state x_0 is given with time-invariant probability $p_0 = p(x_0)$.

We consider these queries arrive separately, one after the other, with an average interarrival time of length δ. We assume that no updates may be made to the relations referenced by the two query while they are being processed. We don't made any assumptions concerning the order of arrivals of Q_1 and Q_2 and their execution. The concurrent execution of these queries is treated in the next subsection. The state-transition graph for the stochastic query optimization of queries

Q_1 and Q_2 executed in sequential mode is shown in Figure 1. The notation for the nodes of the state-transition graph is: x_{ijk}, where i is for the state, j denotes the query type and k is for the stage. We will associate a transition probability to each transition arc of the state-transition model. Let p_{ij} denote the conditional, time-invariant probability that the system undergoes transition from state x_i to state x_j.

Let q_i be the probability that a query is of type $Q_i (i = 1, 2)$ and $q_1 + q_2 = 1$.

Proposition 2.1 *The general, sequential multiple-join stochastic query optimization model for 2 queries and 3 sites defines a nonlinear programming problem that me be decomposed into 2 independent nonlinear programming subproblems.*

Proof: We will associate the join-processing times with the nodes of the state-transition graph and communication times to the arcs of the graph. Let $T_i(X)$ denote the total processing time required for computing in state i.

So we have:

$$
\begin{aligned}
T_{111}(B') &= t_1(A \bowtie B) + c_{21}(B);\\
T_{211}(B') &= t_2(A \bowtie B) + c_{12}(A);\\
T_{112}(C') &= t_3(B' \bowtie C) + c_{13}(B');\\
T_{212}(C') &= t_1(B' \bowtie C) + c_{31}(C);\\
T_{312}(C') &= t_3(B' \bowtie C) + c_{23}(B');\\
T_{412}(C') &= t_2(B' \bowtie C) + c_{32}(C);
\end{aligned}
$$

We use the decomposition principle of separable nonlinear programming. The stochastic query optimization subproblem for query Q_1 is:

$$
\begin{aligned}
&\tau_1 = T_{111}(B')p_{0,111} + T_{212}(C')p_{0,111}p_{111,212} \le \Delta\\
&\tau_2 = T_{211}(B')p_{0,211} + T_{412}(C')p_{0,211}p_{211,412} \le \Delta\\
&\tau_3 = T_{112}(C')p_{0,111}p_{111,112} + T_{312}(C')p_{0,211}p_{211,312} \le \Delta\\
&\quad p_{0,111} + p_{0,211} = 1\\
&\quad p_{111,112} + p_{111,212} = 1\\
&\quad p_{211,312} + p_{211,412} = 1\\
&\qquad \min \Delta
\end{aligned}
$$

The nonlinear programming problem can be solved with the algorithm from [14].

The stochastic query optimization subproblem for query Q_2 is:

$$
\begin{aligned}
&\tau_1 = T_{121}(E')p_{0,121} + T_{222}(F')p_{0,121}p_{121,222} \le \Delta\\
&\tau_2 = T_{221}(E')p_{0,221} + T_{422}(F')p_{0,221}p_{221,422} \le \Delta\\
&\tau_3 = T_{122}(F')p_{0,121}p_{121,122} + T_{322}(F')p_{0,221}p_{221,322} \le \Delta\\
&\quad p_{0,121} + p_{0,221} = 1\\
&\quad p_{121,122} + p_{121,222} = 1\\
&\quad p_{221,322} + p_{221,422} = 1\\
&\qquad \min \Delta
\end{aligned}
$$

where
$$
\begin{aligned}
T_{121}(E') &= t_1(D \bowtie E) + c_{21}(E);\\
T_{221}(E') &= t_2(D \bowtie E) + c_{12}(D);\\
T_{122}(F') &= t_3(E' \bowtie F) + c_{13}(E');
\end{aligned}
$$

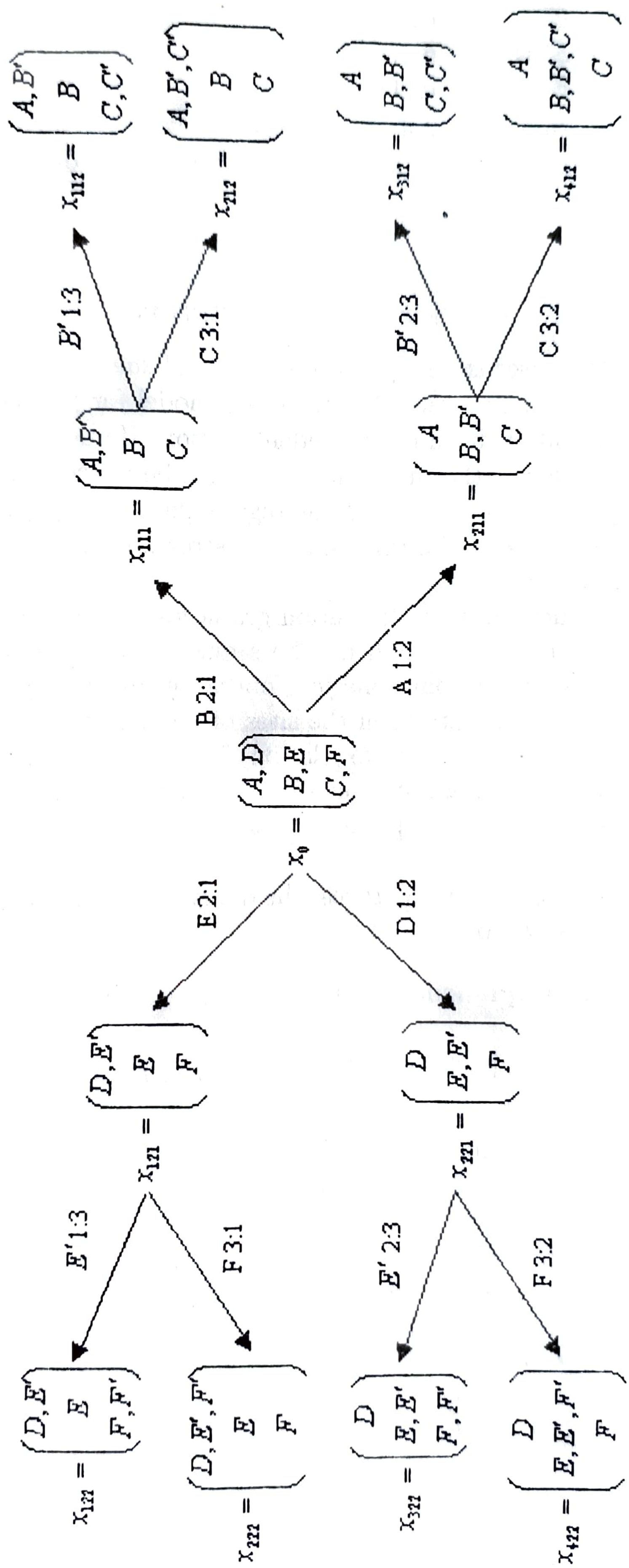

Fig. 1

$T_{222}(F') = t_1(E' \bowtie F) + c_{31}(F);$
$T_{322}(F') = t_3(E' \bowtie F) + c_{23}(E');$
$T_{422}(F') = t_2(E' \bowtie F) + c_{32}(F);$

The nonlinear programming problem can be solved with the algorithm from [14]. The result of the stochastic optimization problem for the sequential execution of queries Q_1 and Q_2 is:

$$\Delta = q_1\Delta_1 + q_2\Delta_2$$

3. The parallel operation

In distributed databases is a major interest to load-share through parallel processing. In this section we present the stochastic model for parallel processing of queries of types Q_1 and Q_2 from the precedent section. If queries of types Q_1 and Q_2 arrive approximately in the same time these will be processed parallel similar to section 3.2 of [8]. Let Q_3 denote the aggregate query type, which occurs with probability q_3, $(q_1 + q_2 + q_3 = 1)$, thus the input stream consists of mixed arrivals of types Q_1, Q_2, Q_3.

In order to construct the state-transition graph for the parallel operation, we start with the state-transition graph for the sequential operation. We will take into consideration to execute only one join operation in one site of a state and to execute different join operations in the sites of a state. In the case of parallel operation it is necessary to transfer two relations from one state of parallel machine to another state, which we mark on the edges of the state-transition graph. The notations for the states x_{ij} of the parallel machine are: i is for the strategy and j is for the stage.

There are different sequences to process the query of type Q_1, see section 2 from [14], for query of type Q_2 too.

3.1. One execution sequence. In this subsection we consider the following sequences:

$$Q_1 = (A \bowtie B) \bowtie C; Q_2 = (D \bowtie E) \bowtie F;$$

The state-transition graph for the parallel machine for Q_3 is shown in Figure 2. For the other execution sequences the state-transition graph can be constructed in the same manner and the stochastic optimization problem can be formulated and solved similar. We consider, that relation A and relation C has no commun attributes, so there are two correct sequences for the execution of the query of type Q_1. Similarry, we consider that relation D and relation F has no common attributes, so the query of type Q_2 can be executed in two different order of the join operations. Thus, for parallel execution of queries of types Q_1 and Q_2 there are four different possibilities to construct the state-transition graph combining the execution sequences for Q_1 and Q_2.

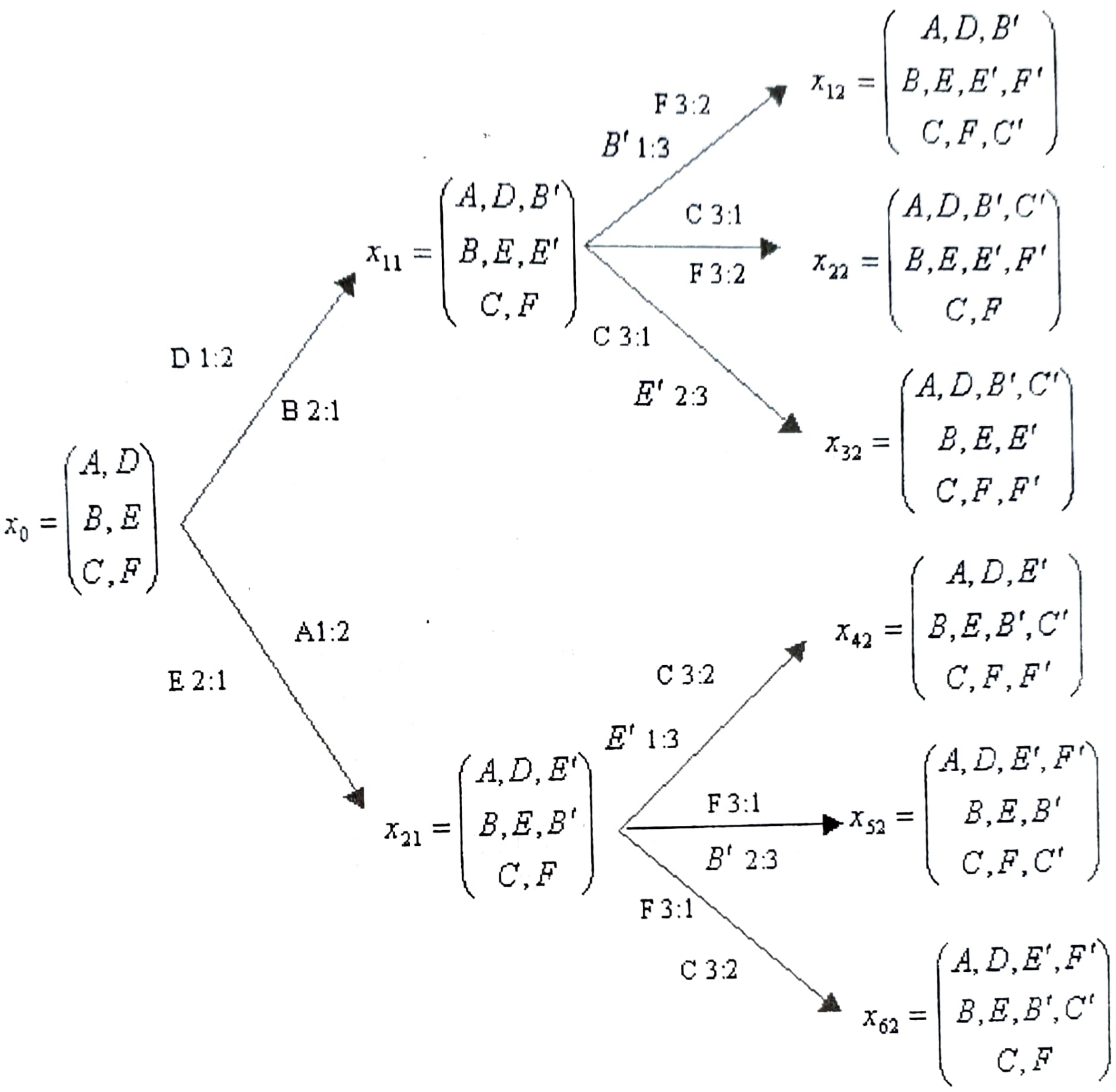

Fig. 2

Proposition 3.1 *The general, parallel multiple-join stochastic query optimization model for 2 queries and 3 sites defines a nonlinear programming problem.*
Proof: Let the notations be the following in the selected case:

$$B' = A \bowtie B; C' = B' \bowtie C; E' = D \bowtie E; F' = E' \bowtie F;$$

The stochastic query optimization problem for Q_3 is:

$$\tau_1 = T_{11}(B')p_{0,11} + T_{21}(E')p_{0,21} + T_{22}(C')p_{0,11}p_{11,22} + T_{32}(C')p_{0,11}p_{11,32} +$$
$$+ T_{52}(F')p_{0,21}p_{21,52} + T_{62}(F')p_{0,21}p_{21,62} \leq \Delta_3$$
$$\tau_2 = T_{11}(E')p_{0,11} + T_{21}(B')p_{0,21} + T_{12}(F')p_{0,11}p_{11,12} + T_{22}(F')p_{0,11}p_{11,22} +$$

$$+T_{42}(C')p_{0,21}p_{21,42} + T_{62}(C')p_{0,21}p_{21,62} \leq \Delta_3$$
$$\tau_3 = T_{12}(C')p_{0,11}p_{11,12} + T_{32}(F')p_{0,11}p_{11,32} + T_{42}(F')p_{0,21}p_{21,42}+$$
$$+T_{52}(C')p_{0,21}p_{21,52} \leq \Delta_3$$
$$p_{0,11} + p_{0,21} = 1$$
$$p_{11,12} + p_{11,22} + p_{11,32} = 1$$
$$p_{21,42} + p_{21,52} + p_{21,62} = 1$$
$$\min \Delta_3$$

where $T_{11}(B') = t_1(A \bowtie B) + c_{21}(B); T_{11}(E') = t_2(D \bowtie E) + c_{12}(D);$
$T_{21}(B') = t_2(A \bowtie B) + c_{12}(A); T_{21}(E') = t_1(D \bowtie E) + c_{21}(E);$
$T_{12}(C') = t_3(B' \bowtie C) + c_{13}(B'); T_{12}(F') = t_2(E' \bowtie F) + c_{32}(F);$
$T_{22}(C') = t_1(B' \bowtie C) + c_{31}(C); T_{22}(F') = t_2(E' \bowtie F) + c_{32}(F);$
$T_{32}(C') = t_1(B' \bowtie C) + c_{31}(C); T_{32}(F') = t_3(E' \bowtie F) + c_{23}(E');$
$T_{42}(C') = t_2(B' \bowtie C) + c_{32}(C); T_{42}(F') = t_3(E' \bowtie F) + c_{13}(E');$
$T_{52}(C') = t_3(B' \bowtie C) + c_{23}(B'); T_{52}(F') = t_1(E' \bowtie F) + c_{31}(F);$
$T_{62}(C') = t_2(B' \bowtie C) + c_{32}(C); T_{62}(F') = t_1(E' \bowtie F) + c_{31}(F);$

In order to solve this nonlinear programming problem we can apply the algorithm from [14], but it has to be modified. The continuous functions are:

$$f_1(x_1, x_2, \ldots, x_8) = c_1x_1 + c_2x_2 + c_3x_1x_4 + c_4x_1x_5 + c_5x_2x_7 + c_6x_2x_8;$$
$$f_2(x_1, x_2, \ldots, x_8) = c_7x_1 + c_8x_2 + c_9x_1x_3 + c_{10}x_1x_4 + c_{11}x_2x_6 + c_{12}x_2x_8;$$
$$f_3(x_1, x_2, \ldots, x_8) = c_{13}x_1x_3 + c_{14}x_1x_5 + c_{15}x_2x_6 + c_{16}x_2x_7;$$

where

$x_1 = p_{0,11}; x_2 = p_{0,21}; x_3 = p_{11,12}; x_4 = p_{11,22};$
$x_5 = p_{11,32}; x_6 = p_{21,42}; x_7 = p_{21,52}; x_8 = p_{21,62};$
$c_1 = T_{11}(B'); c_2 = T_{21}(E'); c_3 = T_{22}(C'); c_4 = T_{32}(C'); c_5 = T_{52}(F');$
$c_6 = T_{62}(F'); c_7 = T_{11}(E'); c_8 = T_{21}(B'); c_9 = T_{12}(F'); c_{10} = T_{22}(F');$
$c_{11} = T_{42}(C'); c_{12} = T_{62}(C'); c_{13} = T_{12}(C'); c_{14} = T_{32}(F');$
$c_{15} = T_{42}(F'); c_{16} = T_{52}(C');$

The results obtained applying the algorithm are in the table from Figure 3.

Case	a)	b)	c)
Nr. of bits for A	8.000.000	8.000.000	8.000.000
Nr. of bits for B	4.000.000	1.000.000	1.000.000
Nr. of bits for C	10.000000	5.000.000	1.000.000
Nr. of bits for D	10.000000	8.000.000	5.000.000
Nr. of bits for E	8.000.000	1.000.000	2.000.000
Nr. of bits for F	4.000.000	5.000.000	3.000.000
Δ_3	165,82	146,35	91,33
$p_{0,11}$	0,325	0,025	1
$p_{0,21}$	0,675	0,975	0
$p_{11,12}$	0,85	0	0
$p_{11,22}$	0,15	0,4	0,16
$p_{11,32}$	0	0,6	0,84
$p_{21,42}$	0	0	0
$p_{21,52}$	1	0,85	0
$p_{21,62}$	0	0,15	1

Fig. 3

3.2. **Another execution sequence for the parallel operation.** If we consider another execution sequence for the query of type Q_2 and for the query of type Q_1 we maintain the sequence from the previous section. So the execution sequences are the following:

$$Q_1 = (A \bowtie B) \bowtie C; Q_2 = D \bowtie (E \bowtie F);$$

We obtain the state-transition graph from Figure 4.

We propose this sequence in order to share the work in the first step of the parallel query execution between the sites. In the sequence of the previous section there was nothing to do in site 3, with the actual sequence, in the first step every three sites participate in the parallel execution of queries Q_1 and Q_2. Thus the number of states of the state-transition graph will increase, there are much more possibilities of execution, therefore we expect a smaller execution time than in the previous section. Let be the next notations:

$$B' = A \bowtie B; C' = B' \bowtie C; E' = E \bowtie F; D' = D \bowtie E';$$

The stochastic query optimization problem is the following:

$$\tau_1 = T_{11}(B')p_{0,11} + T_{21}(B')p_{0,21} + T_{12}(D')p_{0,11}p_{11,12} + T_{22}(C')p_{0,11}p_{11,22} + \\ +T_{42}(C')p_{0,21}p_{21,42} + T_{52}(D')p_{0,21}p_{21,52} + T_{62}(D')p_{0,31}p_{31,62} + \\ +T_{82}(D')p_{0,31}p_{31,82} \leq \Delta_4$$

$$\tau_2 = T_{21}(E')p_{0,21} + T_{31}(B')p_{0,31} + T_{32}(D')p_{0,21}p_{21,32} + T_{42}(D')p_{0,21}p_{21,42} + \\ +T_{72}(C')p_{0,31}p_{31,72} + T_{82}(C')p_{0,31}p_{31,82} \leq \Delta_4$$

$$\tau_3 = T_{11}(E')p_{0,11} + T_{31}(E')p_{0,31} + T_{12}(C')p_{0,11}p_{11,12} + T_{22}(D')p_{0,11}p_{11,22} + \\ +T_{32}(C')p_{0,21}p_{21,32} + T_{52}(C')p_{0,21}p_{21,52} + T_{62}(C')p_{0,31}p_{31,62} +$$

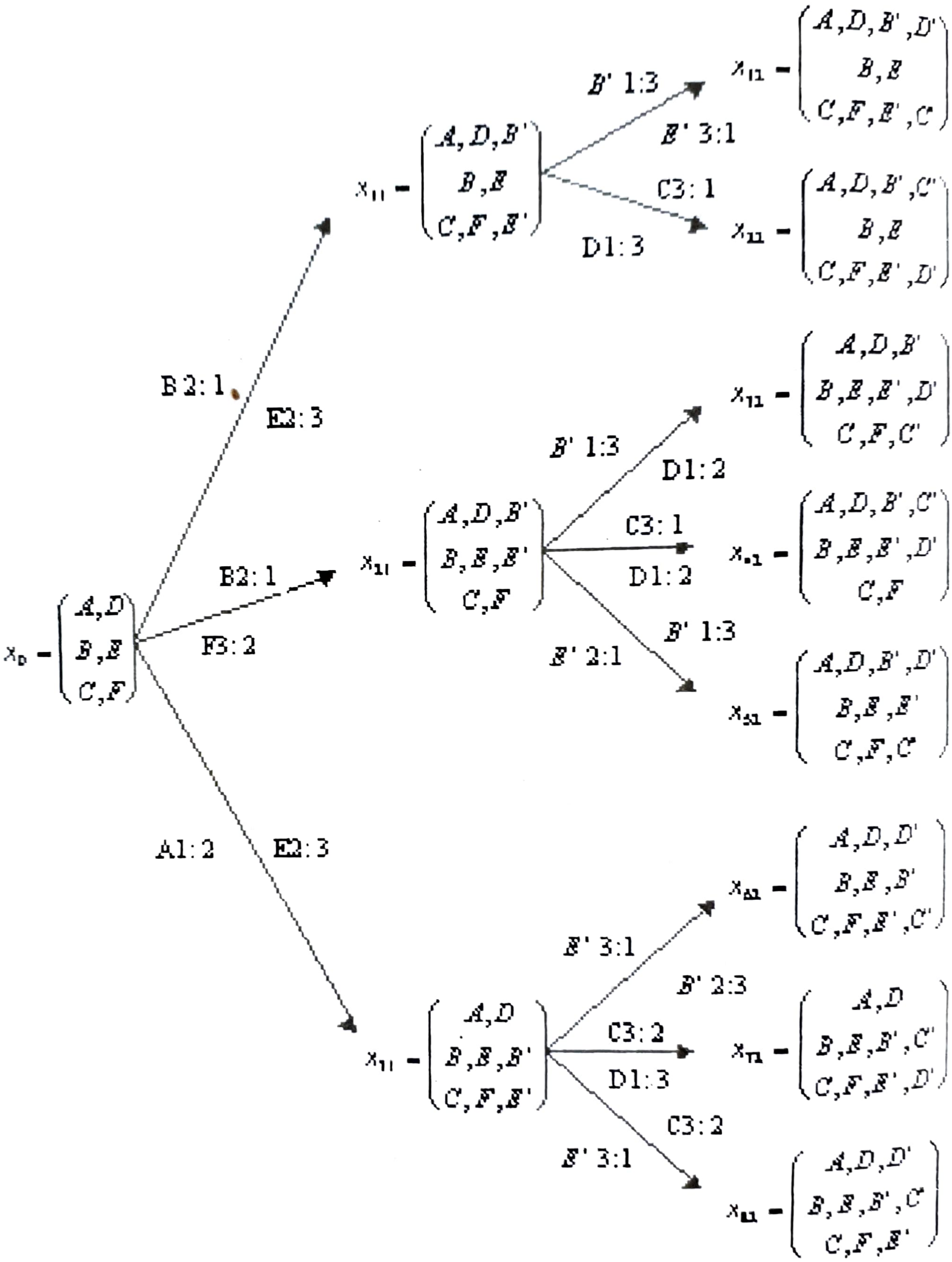
B' 1:3
E' 3:1
C3: 1
D1: 3
B2: 1
E2: 3
B' 1:3
D1: 2
C3: 1
D1: 2
B2: 1
F3: 2
B' 1:3
E' 2:1
A1: 2
E2: 3
E' 3:1
B' 2:3
C3: 2
D1: 3
C3: 2
E' 3:1
A,D,B',D'
B,E
C,F,E',C'
A,D,B'
B,E
C,F,E'
A,D,B',C'
B,E
C,F,E',D'
A,D,B'
B,E,E',D'
C,F,C'
A,D,B'
B,E,E'
C,F
A,D,B',C'
B,E,E',D'
C,F
A,D
B,E
C,F
A,D,B',D'
B,E,E'
C,F,C'
A,D,D'
B,E,B'
C,F,E',C'
A,D
B,E,B'
C,F,E'
A,D
B,E,B',C'
C,F,E',D'
A,D,D'
B,E,B',C'
C,F,E'

Fig. 4

$$+T_{72}(D')p_{0,31}p_{31,72} \leq \Delta_4$$
$$p_{0,11} + p_{0,21} + p_{0,31} = 1$$
$$p_{11,12} + p_{11,22} = 1$$
$$p_{21,32} + p_{21,42} + p_{21,52} = 1$$
$$p_{31,62} + p_{31,72} + p_{31,82} = 1$$
$$\min \Delta_4$$

where $T_{11}(B') = t_1(A \bowtie B) + c_{21}(B); T_{11}(E') = t_3(E \bowtie F) + c_{23}(E);$
$T_{21}(B') = t_1(A \bowtie B) + c_{21}(B); T_{21}(E') = t_2(E \bowtie F) + c_{32}(F);$
$T_{31}(B') = t_2(A \bowtie B) + c_{12}(A); T_{31}(E') = t_3(E \bowtie F) + c_{23}(E);$
$T_{12}(C') = t_3(B' \bowtie C) + c_{13}(B'); T_{12}(D') = t_1(D \bowtie E') + c_{31}(E');$
$T_{22}(C') = t_1(B' \bowtie C) + c_{31}(C); T_{22}(D') = t_2(D \bowtie E') + c_{13}(D);$
$T_{32}(C') = t_3(B' \bowtie C) + c_{13}(B'); T_{32}(D') = t_2(D \bowtie E') + c_{12}(D);$
$T_{42}(C') = t_1(B' \bowtie C) + c_{31}(C); T_{42}(D') = t_2(D \bowtie E') + c_{12}(D);$
$T_{52}(C') = t_3(B' \bowtie C) + c_{13}(B'); T_{52}(D') = t_1(D \bowtie E') + c_{21}(E');$
$T_{62}(C') = t_3(B' \bowtie C) + c_{23}(B'); T_{62}(D') = t_1(D \bowtie E') + c_{31}(E');$
$T_{72}(C') = t_2(B' \bowtie C) + c_{32}(C); T_{72}(D') = t_3(D \bowtie E') + c_{13}(D);$
$T_{82}(C') = t_2(B' \bowtie C) + c_{32}(C); T_{82}(D') = t_1(D \bowtie E') + c_{31}(E');$

The nonlinear programming problem can be solved by the algorithm from [14] with the necessary modifications and with the following continuous functions:

$$f_1(x_1, x_2, \ldots, x_{11}) = c_1x_1 + c_2x_2 + c_3x_1x_4 + c_4x_1x_5 + c_5x_2x_7 + c_6x_2x_8 + \\ +c_7x_3x_9 + c_8x_3x_{11};$$
$$f_2(x_1, x_2, \ldots, x_{11}) = c_9x_2 + c_{10}x_3 + c_{11}x_2x_6 + +c_{12}x_2x_7 + c_{13}x_3x_{10} + \\ +c_{14}x_3x_{11};$$
$$f_3(x_1, x_2, \ldots, x_{11}) = c_{15}x_1 + c_{16}x_3 + c_{17}x_1x_4 + c_{18}x_1x_5 + c_{19}x_2x_6 + \\ +c_{20}x_2x_8 + c_{21}x_3x_9 + c_{22}x_3x_{10};$$

where $x_1 = p_{0,11}; x_2 = p_{0,21}; x_3 = p_{0,31}; x_4 = p_{11,12}; x_5 = p_{11,22}; x_6 = p_{21,32};$
$x_7 = p_{21,42}; x_8 = p_{21,52}; x_9 = p_{31,62}; x_{10} = p_{31,72}; x_{11} = p_{31,82};$

The results obtained applying the algorithm for two cases of the previous section are in the table from Figure 5.

We can see from the results, that in case a), which is identical with the case a) from the previous section, the result is better, 161,65 seconds, regard to 165,82 seconds. The case b) is the same with the case c) from the previous section and the result is 64,163 seconds, less than 91,33 seconds obtained in the previous sections. So if the number of states increase, the mean processing time decrease.

Case	a)	b)
Nr. of bits for A	8.000.000	8.000.000
Nr. of bits for B	4.000.000	1.000.000
Nr. of bits for C	10.000000	1.000.000
Nr. of bits for D	10.000000	5.000.000
Nr. of bits for E	8.000.000	2.000.000
Nr. of bits for F	4.000.000	3.000.000
Δ_4	161,65	64,163
$p_{0,11}$	0,05	0,55
$p_{0,21}$	0,9	0,45
$p_{0,31}$	0,05	0
$p_{11,12}$	0	0
$p_{11,22}$	1	1
$p_{21,32}$	0,45	0
$p_{21,42}$	0,15	1
$p_{21,52}$	0,4	0
$p_{31,62}$	0,4	0,65
$p_{31,72}$	0,45	0,35
$p_{31,82}$	0,15	0

Fig. 5

References

[1] M. S. Chen, P. S. Yu: *Interleaving a Join Sequence with Semijoins in Distributed Query Processing*, in IEEE Trans. on Parall. and Distr. Syst. Vol.3 No.5, 1992, pp. 611-621.

[2] M. S. Chen, P. S. Yu: *Combining Join and Semi-Join Operations for Distributed Query Processing*, IEEE Trans.on Knowledge and Data Engin. Vol.5 No.3. 1993, pp. 534-542.

[3] M. S. Chen, P. S. Yu: *A Graph Theoretical Approach to Determine a Join Reducer Sequence in Distributed Query Processing*, IEEE Trans.on Knowledge and Data Engin. Vol.6 No.1. 1994, pp. 152-165.

[4] M. S. Chen, M. Lo, P. S. Yu, H. C. Young: *Applying Segmented Right-Deep Trees to Pipelining Multiple Hash Joins*, IEEE Trans.on Knowledge and Data Engin. Vol.7 No.4. 1995. pp. 656-667.

[5] C. J. Date: *An Introduction to Database Systems*, Addison- Wesley Publishing Company, 1995.

[6] R. F. Drenick: *A Mathematical Organization Theory*, Elsevier, New York, 1986.

[7] P. E. Drenick, R. F. Drenick: *A Design Theory for Multi-processing Computing Systems*, in Large Scale Syst. Vol. 12, 1987, pp. 155-172.

[8] P. E. Drenick, E. J. Smith: *Stochastic Query Optimization in Distributed Databases*, in ACM Trans. on Database Systems, Vol. 18, No. 2, June 1993, pp. 262-288.

[9] S. LaFortune, E. Wong: *A State Transition Model for Distributed Query Processing*, in ACM Trans. on Database Systems, Vol. 11, No. 3, Sept. 1986, pp. 294-322.

[10] E. Omiecinski, E. T. Lin, *The Adaptive-Hash Join Algorithm for a Hypercube Multicomputer*, IEEE Trans. on Parall. and Distr. Syst. Vol.3 No.3. 1992, pp. 334-349.

[11] M. T. Özsu, P. Valduriez: *Principles of Distributed Database Systems*, Prentice-Hall, 1991.

[12] R. Ramakrishnan: *Database Management Systems*, WCB McGraw-Hill, 1998.

[13] J. D. Ullman: *Principles of Database and Knowledge-Base Systems*, Vol. I-II, Computer Science Press, 1988.

[14] V. Varga: *Stochastic Optimization for the Join of three Relations in Distributed Databases I. The Theory and one Application*, Studia ....1999.

[15] J. L. Wolf, P. S. Yu, J. Turek, D. Dias: *A Parallel Hash Join Algorithm for Managing Data Skew*, IEEE Trans. on Parall. and Distr. Syst. Vol.4 No.12. 1993, pp. 1355-1371.

Babeş-Bolyai University, Faculty of Mathematics and Informatics, RO 3400 Cluj-Napoca, str. Kogălniceanu 1, Romania

E-mail address: ivarga@cs.ubbcluj.ro