# Evaluation of Cost of Plans in Multiple Dependent Queries Execution Using G.A. Techniques

Sambit Kumar Mishra and Srikanta Pattnaik

***Abstract*** **—Multiple processors are employed to improve the performance of database systems and the parallelism can be exploited at three levels in query processing: intra-operation, inter-operation, and inter-query parallelism. Intra-operation and inter-operation parallelism are also called intra-query parallelism which has been studied extensively recently. In contrast, inter-query parallelism has received little attention particularly for multiple dependent queries. As the cost of a given query execution plan is function of many parameters including database structures , the estimated cost of all possible execution plans will be evaluated and also average association degree coefficient between plans will be calculated.**

***Index Terms*** **—Query optimizer, inter query parallelism, plan, gene, chromosome, degree coefficient, fitness value.**

## I. Introduction

Modern database systems use a query optimizer to identify the most efficient strategy, called “plan”, to execute declarative SQL queries. Optimization is a mandatory exercise since the difference between the cost of the best plan and a random choice could be in orders of magnitude. The role of query optimizers is especially critical for the decision-support queries featured in data warehousing and data mining applications. Query optimization using this cost-based approach is computationally expensive with respect to the time and resources that need to be expended to find the best plan. Therefore, understanding and characterizing query optimizers with the ultimate objective of improving their performance is a fundamentally important issue in the database research literature.

The cost of a given query execution plan is a function of many parameters, including the database structure and contents, the engine settings, the system configuration, etc. For a query on a given database and system configuration, the optimizer’s plan choice is primarily a function of the selectivities of the base relations participating in the query that is, the estimated number of rows of each relation relevant to producing the final result. Varying the selectivities of one or more of the base relations produces the selectivity space with respect to these relations.

The key constituents of the query evaluation component of an SQL database system are the query optimizer and the query execution engine. The query optimizer is responsible for generating the input for the execution engine. It takes a parsed representation of an SQL query as input and is responsible for generating an efficient execution plan for the given SQL query from the space of possible execution plans. One aspect of optimization is where the system attempts to find an expression equivalent to the given expression, but more efficient to execute.

Another aspect is selecting a detailed strategy for processing the query. The task of an optimizer is computationally challenging since, for a given SQL query, there can be a large number of possible execution plans.

Query optimization is a difficult problem due to the large number of possible ways to execute a given query using different access methods, join orders, join operators, etc. while industrial strength query optimizers each have their own proprietary methods to identify the best plan.

In a multi-user environment, it is common for a system receiving multiple queries at the same time. As a result, several queries are running on different processors in parallel. Multiple queries execution can be classified into two categories based on query dependency, multiple dependent and independent queries. Currently, an active database research area is data mining, by which the extraction of information from large amounts of data accumulated and used for other purposes.

## II. Review of Literature

Stefan Berchtold et.al [1]have discussed in their paper that the problem of retrieving all objects satisfying a query which involves multiple attributes is a standard query processing problem prevalent in any database system. The problem especially occurs in the context of feature based retrieval in multi databases.

S.Babu et.al[4] have elaborated in their paper that multi database systems use a query optimizer to identify the most efficient strategy called plan to execute declarative queries. For a query on a given database and system configuration, the optimizer’s plan choice is primarily a function of the selectivities of the base relations participating in the query. Query optimizers often make poor decisions because their compile time cost models use inaccurate estimates of various parameters.

Falout C.Barber et.al [7] have evaluated the cost function in task allocation which is the sum of inter processor communication and processing cost and found that they are actually different in measurement unit.

Hong Chen et.al [5] have elaborated in their paper that the multi query processing takes several queries as input, optimizes them as a whole and generates a multi query execution strategy.

Cristina Lopez et.al [9] have defined in their paper that population of individuals known as chromosomes, represent the possible solutions to the problem. These are randomly

Manucript received December 15, 2010.

1 Associate Professor , Department of Computer Sc.&Engg.,Ajay Binay Institute of Technology , Cuttack , Orissa , India (e_mail : sambit_pr@rediffmail.com )

2 Ex-Professor , U.C.E. , Burla ,Director , InterScience Institute of Management & Technology , Bhubaneswar , Orissa , India (e_mail : srikantapatnaik@hotmail.com)

generated, although if there is some knowledge available concerning the said problem, it can be used to create part of the initial set of potential solutions.

Ahmed A.A. Radwan et.al [11] have suggested in their paper that in genetic algorithm ,the search space is composed of candidate solutions to the problem , each represented by a string is termed as a chromosome. Each chromosome has an objective function value, called fitness. A set of chromosomes together with their associated fitness is called the population.

## III. Problem Analysis

Multiple queries execution can be classified into two categories based on query dependency, multiple dependent and independent queries. Currently, an active database research area is data mining, by which the extraction of information from large amounts of data accumulated and used for other purposes.

A good example is the airline reservation system analysing the travellers pattern to keep planes fully booked. During the analysis, it is found that the result of one query is required by other queries; here is a situation where there are multiple dependent queries.

Alternative plans of a query, and other queries in the query set may contain the same task. Therefore in solving the multiple query execution with query dependency, the aim is to determine a set of tasks with minimal cost that contains all tasks of at least one plan of each query with the minimal cost.

## IV. Problem Formulation

Individual plan is represented as chromosome and individual task in a plan is represented as gene. Since a gene in a chromosome represents the plan selected for the query corresponding to the gene position, in the mutation operation the plan number is only replaced with randomly selected valid plan's number for that query. Therefore a mutation operation always generates valid solutions. Different crossover operations can be applied to chromosomes. In our representation scheme, one point and multipoint crossover techniques produce valid solutions for the multiple query processing problems. If two chromosomes are representing two valid solutions of the same multiple query processing problem, then any crossover operation on these two chromosomes produces new chromosomes representing valid solutions for the same multiple query processing problem. Since all chromosome segments that are going to be exchanged to produce a new chromosome represent valid plans for their corresponding queries , the new chromosome obtained by appending these segments represent a valid solution of the multiple query processing problem.

## V. Experimental Results and Analysis

Maximum generations=20
Number of relations=20
Number of queries=20
Planquery( Size of Chromosome )=7
Population=round( rand(number of queries, planquery))
Pc ( Probability for crossover operation)=0.07
Pm ( Probability for mutation operation)=0.001
Cp(crossover point)=round(1+rand*(planquery-1))

The genetic algorithm's chromosomes have a length of 07, which is the number of different terms with nonzero values. Hence the chromosomes that represent each plan and the query will be the following.

Chromosome C1= 1010110
Chromosome C2= 0001100
Chromosome C3= 1111111
Chromosome C4= 0001100
Chromosome C5= 1011001

With the method described, although the number of genes of the chromosomes are kept for the whole population, it will vary according to the query that is being processed and the plans supplied in the feedback.

Population: Genetic algorithm receives an initial population consisting of the chromosomes corresponding to the relevant plans, and to the query.

Selection: The genetic algorithm uses simple random sampling as a selection mechanism. This is implemented by assigning to each individual a selection probability equal to its fitness value divided by the sum of the fitness values of all the individuals.

If after generating the population, the best chromosome of the previous population is no longer present, the worst individual of the new population is withdrawn, and the missing best individual is put back.

## VI. Algorithm

```
summ=0;
summ1=0;
summ2=0;
 for pop=1:tempsz
        temp=0;
for i=1:planqry
temp=temp+2^(i-1)*temppop(pop,i);
         end
          x(pop)=temp;

planselect(pop)=x(pop)/(noqry*planqry);

real_cost(pop)=planselect(pop)/noqry
+t2 ;

est_cost(pop)=real_cost(pop)/noqry;

weight(pop)=(x(pop)*noqry)/(noqry-
x(pop));

fitness(pop)=1+((noqry*weight(pop))/((we
ight(pop)^2)+noqry^2));

summ2=summ2+real_cost(pop);
         min_est_cost=min(est_cost);
         %selection
         summ1=summ1+fitness(pop);
         s(pop)=fitness(pop)/summ1;
```

average association degree coefficient between plans= summ / ( norelartions*noqry)

x(i) represents number of chromosomes.
Crossover point, cp=2
Size of chromosomes=7

TABLE-I

| Sl.No. | x(plan) | est_cost | Fitness | s(pop) |
|---|---|---|---|---|
| 1 | 53 | 0.01892 | 0.5513 | 1 |
| 2 | 24 | 0.00857 | 0.8378 | 0.6031 |
| 3 | 127 | 0.04535 | 0.5072 | 0.2674 |
| 4 | 24 | 0.00857 | 0.8378 | 0.3064 |
| 5 | 77 | 0.0275 | 0.5217 | 0.1602 |
| 6 | 93 | 0.03321 | 0.5143 | 0.1364 |
| 7 | 100 | 0.03571 | 0.5122 | 0.1196 |
| 8 | 98 | 0.035 | 0.5127 | 0.1069 |
| 9 | 49 | 0.0175 | 0.5616 | 0.1048 |
| 10 | 56 | 0.02 | 0.5451 | 0.0923 |

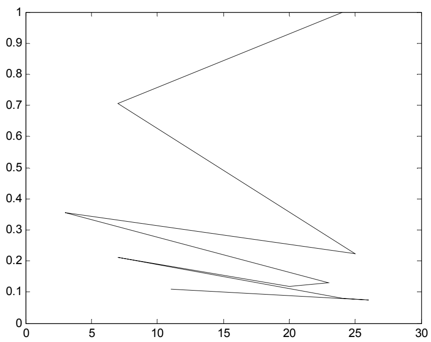

Figure-I

( Plan VS selection parameter, s(pop))

**Selection parameter,s(pop)**

**Plan**

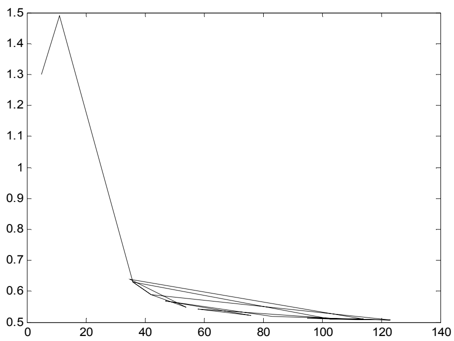

Figure-II

(Plan VS fitness value)

Fitness

**Plan**

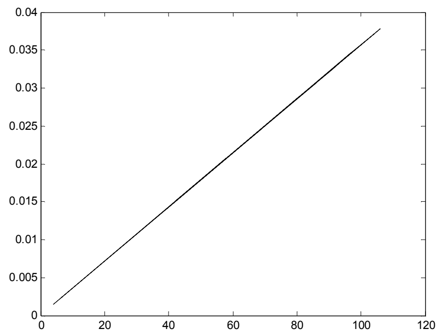

Figure-III

( Plan VS estimated cost)

**Estimated Cost**

**Plan**

## VII. Comparative Study

T.Sellis et.al [1] have implemented a heuristic algorithm which performs a search over some state space defined over access plans. The search space is constructed by defining over state for each possible combination of plans among the queries.

On the given state S=<p1k1,p2k2,----pnkn>,

heuristic function h(S)=∑ est_cost(pik1) + ∑ min ji( est_cost(pij)- Scost(S)

where p1,p2-----pn are possible plans , and 1<=i<=n , 1<=j<=n.

The estimated cost of tasks t is defined as est_cost(t)= cost(t) / nq

where nq is the number of queries and cost(t) represents the cost of task t.

The estimated cost for plan pij is defined as est_cost(pij)=∑ est_cost(t)

where t ε pij.

Maximum generations=100

Probability for crossover operation=0.06

Probability for mutation operation=0.001

TABLE –II (QUERY SET USED IN THE EXPERIMENT)

| Query Set | No. of Plan | Range of tasks | Est_cost of Plan |
|---|---|---|---|
| Qset0 | 3-6 | 3-6 | 0.48 |
| Qset1 | 3-6 | 3-5 | 0.38 |
| Qset2 | 3-6 | 3-6 | 0.47 |
| Qset3 | 2-6 | 3-5 | 0.35 |
| Qset4 | 2-6 | 3-5 | 0.39 |
| Qset5 | 2-6 | 3-5 | 0.38 |
| Qset6 | 2-6 | 2-4 | 0.32 |
| Qset7 | 2-6 | 2-4 | 0.32 |

Murat Ali et.al [15] have considered n number of queries q1 to qn and optimized them together. Each query qi has number of possible solution plans , and each plan of a query contains a set of tasks, which when executed in a certain order and produce the answer for the query. Each task also has an associated cost, and for convenience the cost value is represented by a positive integer number. Alternative plans of a query, and other queries in the query set, may contain the same task. Therefore the aim is to determine a set of tasks, with minimal total cost, that contains all the tasks of at least one plan of each query.

Assume that mi be the number of queries among the remaining set of queries without an assigned plan, with an alternative plan containing the task ti.

The estimated cost of task ti is defined as

Est_cost(ti)= Real_cost(ti)/ mi .

The estimated cost of plan pij is defined as

Est_cost(pij)= ∑ Est_cost(ti) .

Parameter values used in the simulation of genetic algorithm are defined as follows.

Population size=100

Number of generations=100

Maximum number of genes to transfer=2

Probability for mutation=0.001

Probability for crossover operation=0.06

TABLE-III

| No. of Plans | Estimated Cost | Action |
|---|---|---|
| P11, p12 | 8 | In expansion list |
| P12 | 7 | Expanded |
| P12, p21 | 7 | Expanded |
| P12,p22 | 10 | Expanded |
| P12,p21,p31 | 8 | Not Solution |
| P12,p21,p32 | 7 | Solution |
| P12,p23,p31 | 7 | Solution |

To reduce the estimation error in the heuristic the plan cost estimation function defines and experimentally evaluates alternative query ordering heuristics for determining the best order of alternative plan assignment for each query in the query set.

The initial estimated cost is determined as 3.83. Choosing the minimum cost plans for each query, the total estimated cost is 8. The upper bound might be less than the summation of the costs of minimum costly plans due to common tasks. If an expanded state that represents a partial or complete solution that has an estimated cost greater than 8, then action is not a solution.

## VIII. Discussion & Conclusion

Assume that a database D is given as a set of relations {R1,R2,--------,Rn},each relation defined on a set of attributes. An access plan for a query Q is a sequence of tasks , or basic relational operations, that produces the answer to Q.

The tasks have some cost associated with them which reflects both the CPU and I/O cost required to process them. The cost of an access plan is the cost of processing its component tasks.

Assume now that a set of queries Q={ Q1,Q2,--------,Qn} is given.

A global access plan for Q corresponds to a plan that provides a way to compute the results of all n queries.

A global access plan can be constructed by choosing one plan for each query and then merging them together. The merging process basically amounts to the identification of identical tasks. Due to common tasks, the union of the locally optimal plans does not necessarily give the globally optimal plan. A global access plan can be constructed by choosing one plan for each query and then merging them together. The merging process basically amounts to the identification of identical tasks. Due to common tasks, the union of the locally optimal plans does not necessarily give the globally optimal plan.

The problem of multiple query processing problems can be defined as follows.

Given n sets of access plans p1,p2,--------pn with pi={pi1,pi2,--------pin}being the set of possible plans for processing Qi, 1<=i<=n .

To prove the multiple query processing problem is an NP-hard problem , consider the following decision problem.

Given n sets of access plans(p1,p2,----pn) with pi={pi1,pi2,-------piq} being the set of possible ans for processing Qi,1<=i<=n, and a constant q.

Clearly original multi query processing problem will be NP-hard if the above decision problem is NP-Complete. It is easy to see that multiple query processing belongs to NP since a nondeterministic algorithm needs only guess one plan for each query and check whether the cost of the global access plan obtained by merging the guessed local access plans is less than or equal to combining the access plans can be easily done in polynomial time and therefore the checking steps takes only polynomial time.

## Reference


[1] Stefan Berchtold,Cristian Bohm, University of Munich , Oettinge Str, Germany 2001.

[2] T.Sellis,"Multiplequery optimization",IEEE transactions on knowledge and data Engineering , Vol-2, June-1990.

[3] K. Shim, T.Sellis , D.Nau , " Improvements on algorithms for multiple query optimization", Data and knowledge Engineering 12, 1994, pp.197-222.

[4] S.Babu , P.Bizarro, D.Dewitt, Proc.of ACM SIGMOD , Int.Conf. of Management of data, June-2005.

[5] Hong Chen, Sheng Zhou, Shan Wang , School of Information , Remin University China , ICMD- 2003.

[6] K.H.Liu , C.H.C. Leung , Y.Jiong , Department Of Computer and Mathematical Sciences , Victoria University Of Technology , Ballarat Road , Australia , International Conference, 2004.

[7] Falout C, Barber, R.Flicker.M, Journal of Intelligent Information Systems 2000.

[8] Murat Ali Bayir, Ismail H.Toroslu , and Ahmet Cosar , "Genetic algorithm for the multiple query optimization problem", IEEE transactions on Systems , Vol-37, No.1, January 2007.

[9] Cristina Lopez, Vicente P.Guemero Bote , University of Extremadura , Badajoz, Spain , "Proc. Of ACM Sigmod Intl.conf.of Management of Data, June 2005.

[10] Zehai Zhou , Department of FACIS, University of Houston , USA, Journal of Information of Computing science vol-2, No.4, 2007.

[11] Ahmed A.A.Radwan , Bahgat A.Abdel Lataef , abdel Mgeid , A.Ali , World Academy of Science, Engineering and Technology-2006.

[12] J.Wen , H.Chen , S.Wang , " Query Optimization Techniques of a shared nothing parallel database system " , Chinese Journal of Computer , vol 23(1), 2000.

[13] Suchitra Upadhayaya & Suman Lata , Department of Computer science , K,U.K. IJCSNS International Journal of Computer science, March-2008.

[14] A. Deshpande and J.M.Hellerstein , "Decoupled query optimization for federated database systems", ICDE,2002.

[15] X.Wong , R.Burns, A.Terzis, and A.Deshpande , " Network aware join processing in global scale database " , ICDE,2008.