A Novel Approach of Query Optimization for Genetic Population

Ishtiaq Ahmed[1], M. Rizwan Beg[2] ,Kapil Kumar Gupta[3],Mohd.Isha Mansoori[4]

**[1] Department of Computer Science & Engg. ,Integral University
Lucknow, Uttar Pradesh,226001,India**

**[2] Department of Computer Science & Engg. ,Integral University
Lucknow, Uttar Pradesh,226001,India**

**[3] Department of Information Technology ,Goel Institute of Technology and Management
Lucknow, Uttar Pradesh,226001,India**

**[4] Department of Computer Science & Engg. ,Integral University
Lucknow, Uttar Pradesh,226001,India**

Abstract
In relational database large information is maintained and organized so that user can get the desired information in predictable and reliable fashion through the query processing. Retrieving these results in timely manner is Query Optimization. Optimization is a process of choosing most efficient way to execute a SQL query. This is important step in processing of any data manipulation language because in RDBMS, as in large and complicated system performance is key issue. It's an issue that you need to deal with on an ongoing basis so performance must be built in. This paper will provide the reader general concepts of query optimization in relational database system. I will define the implementation plan using join ordering to broaden the potential of database engine through the use of Genetic algorithm in the field of database area in the context of query optimization [12].

Keywords: *Query Optimization, relational database, Query processing, Query Algorithms.*

1. Introduction

Query Optimization is process of choosing most efficient way to execute the SQL statement. This is important step in data processing of any data manipulation language. Query processing consists of three stages.

1. Parsing User Query and transform the parse tree into relational algebra expression.
2. Optimizing algebraic expression.
3. Selecting an evaluation algorithm [6] [2].

Stages **2-3** are two parts of Query optimization. Query Optimization is more important and standard component of database system. Queries in SQL could have several compositions and ordering. Determining "good" composition is major objective of Query Optimizer. The optimizer formulates execution plans and attempt to choose the best plan for executing the SQL statement with least estimated cost. To determine a cost of plan in terms of CPU cycle, I/O, time memory usage etc. optimizer uses data available in the database system [4].

2. SQL QUERY

In relational database information are stored in collection of tables. SQL query consists of operation performed on tables and retrieves specific data to/from physically stored medium. Such operations for relational DBMS can be relational algebra operations these operations are described below.

- Select (σ): Returns tuples that satisfy the given predicate.
- Project (∏): Returns attributes listed
- Join (⋈): Returns Filtered Cross Product
- Set Operation Union, Intersect and Difference

2.1 Paradigm

To illustrate how query is performed and why query optimization might be necessary we will look the following database.

Employee :{ EmpId, SDId, EmpName, EmpAdd, EmpPhone}
Sub Department :{ SDId, DeptId, SDName}
Department :{ DeptId, DeptName}
The question we want to ask is, to know all the Employees in a particular department?

If we translate this into SQL query, in a first step we might get

Select e.EmpID, e.SDId, e.EmpName, e.EmpAdd, e.EmpPhone from Employee e, SubDepartment sd, Department d WHERE e.SDId = sd.SDId AND sd.DeptId = d.DeptId AND d.DeptId = 'My Department'

This query gives us following access plan: we see very soon that this query can not be best way to get our response. We are having three cross product which means we are creating a table whose number of rows is [e]*[sd]*[d].for large tables result is not acknowledged one possible way to improve this is to perform selection earlier on in the search as shown in fig.1

This way we get the following result:

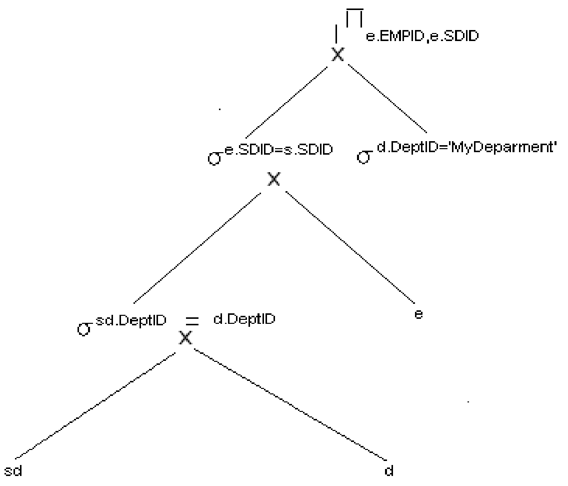

Fig. 1 Query Plan1

By doing the choice earlier on we impose boundary and our final table is smaller than the former one [6] [2]. We can advance our result by adding cross join operators as shown fig.2

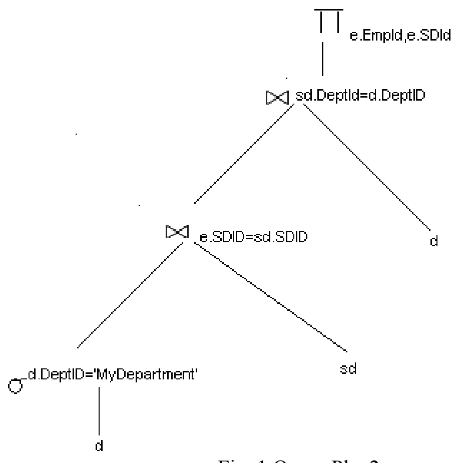

Fig. 1 Query Plan2

We have advanced our original query, we are aware that join operator is in both form commutative and associative and we could not predict which order of join would be best to reduce the resulting relations [4].

3. Query Metrics Cost

Time taken to execute a query depends on resources required to perform the required operations: CPU cycles, disk access, RAM.Since data transfer to/from disks substantially slower that memory based transfer. The disk uses show a great majority of total cost-particularly for vast databases that cannot be preloaded into memory, the CPU cost can also be insignificantly compared to disk access for many operations. Disk accessing cost is generally measured in terms of blocks transferred to/from disk.

4. Role of Indexes

Indexes play a vital role in reducing the execution time of diverse operations such as select and join .Let us see a look of some type of indexes and their role in Query optimization.

4.1 Clustered Index

Clustered indices can greatly increase overall speed of retrieval, but usually only where the data is accessed sequentially in the same or reverse order of the clustered index, or when a range of items is selected.

4.2 Bitmap index

A bitmap index is a special kind of index that stores the bulk of its data as bit arrays (bitmaps) and answers most queries by performing bitwise logical operations on these bitmaps. The most commonly used index, such as B+trees, are most efficient if the values it indexes do not repeat or repeat a smaller number of times For example, the gender field in a customer database usually contains two distinct values: male or female. For such variables, the bitmap index can have a significant performance advantage over the commonly used trees.

4.3 Dense index

A dense index in databases is a file with pairs of keys and pointers for every record in the data file. Every key in this file is associated with a particular pointer to a record in the sorted data file. In clustered indices with duplicate keys, the dense index points to the first record with that key.

4.4 Sparse index

A sparse index in databases is a file with pairs of keys and pointers for every block in the data file. Every key in this file is associated with a particular pointer to the block in the sorted data file. In clustered indices with duplicate keys, the sparse index points to the lowest search key in each block. Primary key is a sparse index.

4.5 Reverse index

A reverse key index reverses the key value before entering it in the index. E.g., the value 24538 becomes 83542 in the index. Reversing the key value is particularly useful for indexing data such as sequence numbers, where new key values monotonically increase [6].

5. Query Optimization

Whenever we create a query optimization algorithm most favorable developed access plan is key concern hence numerous algorithms have been created and projected as fine approach to query optimization. These algorithms are the following.

1) Deterministic Algorithm
2) Randomized Algorithm
3) Genetic Algorithm

5.1 Deterministic Algorithms

This algorithm is also called dynamic programming algorithm. The memory requirement and exponential running time increases exponentially with query size, in the worst case all partial results generate in every step must be preserved to be accessed in the next one. Today's applications have a limit on the size of queries that can be accepted (usually around 10-15 joins) since for bigger queries optimizer crashes due to high memory requirements. This algorithm works well for those application involving around 10-15 queries [1] [2].

5.2 Randomized Algorithms

To address the inability of dynamic programming to deal with really big queries, which emerge in several new application fields, several other algorithms have been developed recently. This optimization algorithm is based on plan transformation rather than plan construction of dynamic programming. Randomized algorithm works by searching a graph whose nodes are execution plans that can be used to respond a query. The objective of algorithm is to find a global minimum cost. This algorithm initiates random moves in a graph. A move is called uphill if cost of source node is less than cost of destination node. It is called global minimum if it has lowest cost among all nodes [1] [2].

5.3 Genetic Algorithms

Different search algorithms have been used by researchers to get the best plan for executing a query. As queries is getting more ticklish search complexity is increasing. Current query optimization techniques are insufficient to carry some of the emerging applications. Genetic Algorithm**s** (GAs) is becoming a widely used and accepted method for very complex optimization problems.
The use of Genetic Algorithm to query optimization is motivated by GA's strength and competency in bigger search problems [9]. Using the user input query we retrieve significant documents through the information retrieval system. Due to rapid increment in number of documents over the World Wide Web has made it necessary to choose the best way in retrieving the most relevant document to the user query.
Many information retrieval system are based Boolean queries where query terms are joined by logical operators AND and OR. The more relevant document depends on query request. The commonly used measures of retrieval success are precision the percentage of the retrieved documents and recall the percentage of the relevant documents that are retrieved.

5.3.1 Query Optimization Process

A database query in relational database can represented in the form of query tree where leaf node represent the relation and middle node task and combine the data form their input nodes using the relational operations of project, join etc. and top node of tree returns the solution. Following fig. shows the path from query execution plan [9].

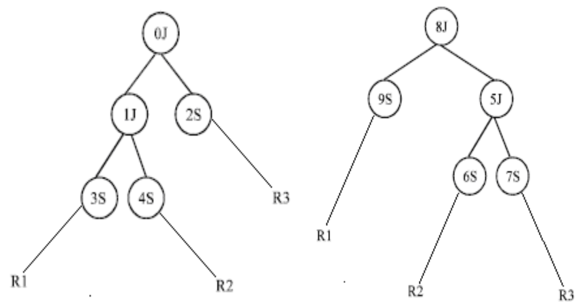

Fig. 3 Path from user query to execution plan

A genetic algorithm is used to enhance the performance in information retrieval system. Finding essential documents from bulk of documents will be achieved by using query to choose more appropriate documents [11]. GA is used to get the solution using the set of functions like crossover,

mutation and natural selection and fitness function. Using Genetic Algorithm over population of individuals or chromosomes shows that several operators are utilized.
–Fitness is a measure of the goodness of a chromosome, that is, a measure of how well the chromosome fits the search space, or solves the problem at hand
–Selection operator enhances the average quality of population.
– Crossover is a process of exchanging the genes between the two individuals that are reproducing.
– Mutation is the process of randomly altering the chromosomes [11] [14].

In Genetic Algorithm Solutions are called individuals or Chromosomes. Some of the Selection and variation function are applied until/unless some termination condition is reached. Each run of loop is called generation .The objective of selection operator is to enhance the average quality of population. The quality of chromosomes is assessed by fitness function. A Fitness function provides the best solution in genetic algorithm [9] [11].
Crossover and mutation are key operators in genetic algorithm. Crossover chooses the gene from parent chromosomes and generates a novel child. After crossover mutation is performed. Mutation arbitrarily changes the child.

5.3.2 Genetic Algorithms for Query Optimization

Randomized search strategies have been proposed (Horng *et al*., 2000) to improve a start solution until obtaining a local optimum. Examples of such strategies are simulated-annealing (Ioannidis and Wong, 1987) and iterative-improvement (Swami and Gupta, 1988). With the same objective, genetic search strategies (Goldberg, 2003) can be applied to query optimization, as a generalization of randomized ones. Randomized or genetic strategies do not guarantee that the best solution is obtained, but avoid the high cost of optimization. The big problem in query optimization is that, the search space is complex and genetic algorithms are theoretically and empirically proven to provide robust search in complex spaces. These algorithms are computationally simple yet powerful in their search for improvement. They are not fundamentally limited by restrictive assumptions about search space (Goldberg, 2003). The use of GA approach in addressing the Query Optimization issue therefore seems appropriate [1] [9] [13].

6. Proposed Approach

GA is most widely used technique to query optimization as earlier we have discussed that in this algorithm initial population is generated randomly then selection and variation function are applied. The objective of Selection function is to improve the average quality of population by providing the chromosomes of higher quality and fitness function determine the quality of chromosomes since population emerges and minimizing the cost of Chromosomes. Crossover and mutation function is applied crossover chooses the gene from parent chromosomes and generates a new child after rover mutation is performed.

The following are the description of genetic algorithm

```
create initial population
randomly initialize population
encoding the generated population
  Repeat
      calculate objective function
      Evaluate fitness function
      apply genetic operators
      selection is performed
      then crossover is performed
      At last mutation is performed
  Until stopping criteria
```

6.1 Implementation

GA execution:

```
 // initialize a new genetic algorithm
GeneticAlgorithm ga = new GeneticAlgorithm (new
OnePointCrossover<Integer> (),
   1, new RandomKeyMutation (), 0.10, new
TournamentSelection (TOURNAMENT_ARITY));

//initial population
Population initial = getInitialPopulation ();

// stopping condition
StoppingCondition stopCond = new
FixedGenerationCount (NUM_GENERATIONS);

// run the algorithm
Population finalPopulation =ga.evolve (initial,
stopCond);

// best chromosome from the final population
Chromosome bestFinal =
finalPopulation.getFittestChromosome ();
```

Table 1: Arguments to the Genetic Algorithm constructor

Parameter	Value in example	Meaning
crossover Policy	One Point Crossover	A random crossover point is selected and the first part from each parent is copied to the

		corresponding child, and the second parts are copied crosswise
crossover Rate	1	Always apply crossover
mutation Policy	Random Key Mutation	Changes a randomly chosen element of the array representation to a random value uniformly distributed in [0, 1].
mutation Rate	1	Apply mutation with probability 0.1 - that is, 10% of the time.
selection Policy	Tournament Selection	Each of the two selected chromosomes is selected based on an n-ary tournament -- this is done by drawing n random chromosomes without replacement from the population, and then selecting the fittest chromosome among them.

The algorithm starts with an initial population of Chromosomes and executes until the specified StoppingCondition is reached.
Information retrieval, query for significant documents are representing for each chromosomes and each document is described by set of terms. The description di for document Di, where i=1……. k, the set of terms for Di. di=m1, m2……..mn. The value for each term will be 1 if this term exists in the documents otherwise 0.
Here we want to solve a query (m8 or m2) using our approach from the Population of chromosomes given below.

No.	Query
1	(m3 and m8) and(m10 or m4)
2	(m1 and(m8 and m2)) or (m4 or m2)
3	(m1 or m2) and(m5or m4)and(m3 and m6))
4	(m9 and m14) or (m8 and m2)
5	(m14 and m1) and (m8 or m2)
6	(m2 or m6) or (m8 and m13)
7	(m3 and m4) or(m12 xorm15) and m8
8	(m1or m5)
9	(m2 and m6)
10	(m8 or m4)

We mention that in initial population there is single query contain a sub expression m8 and m2, our testing of execution for genetic algorithm using this approach was done independently. We took two different cases.

The two test cases descriptions are:
-Test case 1: collection of 9 documents, 27 words in the collection, maximal number of different of words in a document is 10.Result is shown in table 2.
-Test case 2: collection of 100 documents, 40 words in collection, maximal number of different words in a document is 40.Result are shown table 3.

Table.2 Result of Test Case I

No.	Selection for Parents depends on	Result	Fitness Value
1	Precision	(((m8 or m2) or (m13 and m8)) or ((m8 xor m2) or (m13 and m8)))	1.1500000 1.0000000
		(((m8 or m2) or (m8 or m2)) or (m8 and m2)	1.1500000 1.0000000
	Recall	(m13 and m8) or ((m6 or m2))	1.0722222 1.0000000
		(m13 and m8) or ((m6 or m2) or((m13 or m8) or(m6 or m2))))	1.0722222 1.0000000

Table .3 Result of Test Case II

No.	Selection for Parents depends on	Result	Fitness Value
1	Precision	((m8 or m2) or ((m2 and m4) or ((m8 or m2) and m1)))	1.150000 0 1.000000 0
		((m2 and m4) or (((m2 and m4) xor(((m2 and m4) or (((m4 or m10) and (m8 and m13)) or (m8 and m13))) and (m8 and m13)) and (m8 and m13))) and (m8 and m13)))	1.158470 0.567211
	Recall	(((m2 or m4) or (((m2 or m4) or (m6 or m2)) or (m6 or m2))) or (m6 and m2))	1.063180 0.820060

		(((m2 or m4) or (((m2 or m4) or(((m13 and m8) or (m6 or m2)) and (m8 xor m2))) or (m6 or m2))) or (((m13 and m8) or (m6 or m2)) or (m8 xor m2)))	1.001080 1.000000

7. Conclusion

We have seen the deterministic algorithm which requires high memory consumption to store the solutions, deterministic algorithm works well for those application requiring few queries but application involving more that 10-15 queries is prohibitively expensive. In the case of randomized algorithm performs random walk to find a node (execution Plan) that is global minimum cost. Genetic and randomized produce best plan to execute a query and they are better in terms of running time. Genetic algorithms are widely used to complex query optimization problem like traveling salesman problem. If only mutation operator is used algorithm is very slow but crossover makes algorithm faster. GAs are emerging as higher probability of getting the best solutions for large query optimization problems .Overall, the results prove the applicability of genetic optimization algorithms to the problem. GAs may prove to be a competitive substitute to deterministic optimization algorithms even for small solution spaces.

References

[1]Yannis E.Ioannidis and Youngkyung Cha Kang: Randomized Algorithms for Optimizing Large Join Queries, Computer Sciences Department University of Wisconsin, Pages 14-19.

[2]Prof.M.A.Pund, S.R.Jadhao, P.D.Thakare: A Role of Query Optimization in Relational Database, IJSER, vol 2, Issue 1January-2011, Pages 1-4.

[3]Ben McMahan, Moshe Y. Vardi: From Pebble Games to Query Optimization www.wikipedia.com.

[4]Kristina Zelenay: Query Optimization, Seminar Algorithmen für Datenbanksysteme June 05, Pages1-9.

[5] P Selinger M.M.Astrahan,D D Chamberlin R A Lorie and T GPrice,Acess Path Selection in Relational Database Management System, in Proceedings of 1979 ACM-SIGMOD conrence,Boston,MA,June 1979,pp 23-34.

[6]Michael L.Rupley, Jr.Indiana University at South Bend: Introduction to Query Processing and Optimization, Pages 2-4.

[7]Henk Ernst Blok, Djoerd Hiemstra and SunilChoenni, Franciska de Jong, Henk M.Blanken and Peter M.G. Apers. Predicting the cost-quality trade-off for information retrieval Queries: Facilitatiing database design and query optimization. Proceedings of the tenth international conference on Information and knowledge management, October 2001, Pages207-214.

[8]Donald Kossmann and Konrad Stocker.Iterative Dynamic Programming: A new Class of Query Optimization Algorithms. ACM Transactions on Database Systems, Vol. 25, No. 1, March 2000, Pages 43-82.

[9]M. Sinha and S.V. Chande: Query Optimization Using Genetic Algorithms, Research Journal of Information Technology, Pages1-4.

[10]Dana Vrajitoru[12] : Large Population or Many Generations for Genetic Algorithms? Implications in Information Retrieval.
1 University of Neuch^atel, Computer Science Department, Pierre-_a-Mazel 7, 2000Neuch^atel, Switzerland.
2 EPFL, Department of Mathematics, CH-1015 Lausanne, Switzerland, email:dana.vrajitoru@ep.ch.

[11]Suhail S. J. Owais, Pavel Kr¨omer, and V´aclav Sn´a¡sel: Query Optimization by Genetic Algorithms.

[12]Tom V. Mathew: Query Optimization by Genetic Algorithm.

[13]A Swami, Optimization of Large joins Queries Combining Heuristics and Combinatorial Techniques, in Proceedings of the1989 ACM-SIGMOD Conference, Portland, OR, and June 1989.

Ishtiaq Ahmed obtained his MCA degree in 2005 from Sahara Arts & Management Academy, Lucknow, Uttar Pradesh, India. He is currently student of M.Tech (CSE) from Integral University, Lucknow .His main research interest is in the field of Advance Database Management System and Data Structure.

Prof. Dr. M. Rizwan Beg is M. Tech & Ph.D in Computer Sc. & Engg. Presently he is working as Professor & Head Deptt. Of Computer Sc. & Engg. And Information Technology of Integral University Lucknow, Uttar Pradesh, India. He is having more than 16 years of experience which includes around 14 years of teaching experience. His area of expertise is Software Engg., Requirement Engineering, Software Quality, and Software Project Management. He has published more than 40 Research papers in International Journals & Conferences. Presently 8 research scholars are pursuing their Ph.D in his supervision. Dr. Beg is Associate Editor in Chief of Advancement in Computing Technology & is also member of editorial board for various other International & National Journals. He is member of large member of International professional societies & also member of Advisory Board for various institutions in India. He chaired a no. of workshops, seminars & National Conference.

Kapil Kumar Gupta obtained his B. Tech (IT) degree in 2009 from JSS academy of technical education, Noida, Uttar Pradesh, India. He is Asst. Professor in Department of Information Technology, Goel Institute of technology and management, Lucknow. Uttar Pradesh, India. His main research interest is in the field of Image Processing, Design and analysis of algorithms, etc.. He is a Member of CSTA.

Mohd. Isha Mansoori obtained his MCA degree in 2005 from Sahara Arts & Management Academy, Lucknow, Uttar Pradesh, India. He is currently working as a Lecturer in Computer Application department of Integral University, Lucknow .His main research interest is in the field of Database Management System and Data Structure.