INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & MANAGEMENT

DATABASE OPTIMIZATION USING GENETIC ALGORITHM

Dr..K.Vikram*[1] & Dr.A.Prathap Kumar[2]

ABSTRACT

The optimization process involves finding a single or a series of optimal solutions from among a very large number of possibilities. In this process, the space of potential solutions is reduced to one or a few of the best ones. The criteria for goodness/fitness of a solution is also part of the problem, defined by a data analyst, and acts as a uniform measure for judging the quality of a solution. Traditional mathematical techniques often break down because of trillions of potential combinations of solutions and/or poorly behaved functions involved. GAs can provide very good solutions (not the best) to a variety of optimization problems.

Keywords: *Database query optimization, Genetic algorithm, Join ordering problem.*

I. INTRODUCTION

Query Optimization

The query optimization is a process of finding the most effective execution plan for the given user submitted query. Query optimization has been found very useful in increasing the database systems' performance in terms of time.

A query written in a high level language needs to convert into a form that system can understand and perform further processing. In its internal form, the relational algebra expression, there are number of variations available for representation. Also the various query optimization strategies and algorithms are available to compute the answer. Researchers have worked with various operations of the query to find out the most efficient query execution plan, various techniques to choose the optimal solution among the various methods, etc. Different query optimization techniques have been applied like rule based optimization, cost-based query optimization, deterministic optimization, randomized optimization and their variations.

Results of query optimization can be used by different emerging database management systems. The database users can get benefits of the optimization by getting the results to the query in a timely and predictable manner. Database vendors can use them to improve the efficiency of their DBMS which will provide support to the upcoming huge amount of data. On the other hand database designers can use them to decide which algorithms to use in certain situations, which limitations to cope up with, etc.

Genetic Algorithm

Let us consider the TSP problem. A shipping firm has to deliver goods to n different cities by car. The car can visit each city only once and can begin with any city. The task is to minimize the distance traveled. There are n! (n factorial) of all combinations and (n-1)!/2 of unique possible travel paths to examine. For example, for 5, 10, and 25 cities there are 5! = 120; 10! = 3,628,800; and 25! = 1.55*1025; respectively, of possible combinations (paths). For 25 or more cities, the problem is computationally intractable because even the fastest computer with thousand.

```
Iteration= 0 Initiate population
P(0) Evaluate Population
P(0) While (not termination criterion)
do Begin Iteration=Iteration+1 Selection P(Iteration) from P(Iteration-1)
Alter P(iteration) Evaluate P(Iteration) End End
```

Traversing the decision space is of prime importance. In most of the optimization techniques, some transition rules are used to start with a single point and determine the next point. It is mostly used for locating false peaks in multi-model (many peaked) search spaces. Whereas GA starts with a bucket of points simultaneously, i.e. a population of strings, climbing many peaks in parallel, this reduces the probability of finding a false peak.

Many search techniques require much auxiliary information in order to work properly. By contrast, GAs has no need for all this auxiliary information. To perform an effective search for better and better structures, they only require payoff values (objective function values) associated with individual strings.

The optimization task (Figure 1) involves finding a single or a series of optimal solutions from among a very large number of many possible solutions (Dhar & Stein, 1999). It relies on the process of reducing the space of potential solutions to one or a few of the best ones. When the function is continuous and differentiable, calculusbased methods can be used to find global minimum or maximum. When a problem domain is linear but not continuous, linear programming technique can be applied. However, when problems (domains or search space) are discontinuous and nonlinear or computationally prohibitive (such as Travelling Salesman Problem - TSP), traditional mathematical techniques often break down because of trillions of potential combinations of solutions and/or poorly behaved functions involved. In such situations, genetic algorithms (GAs) come very handy because they can provide very good solutions, not the best ones (see the solid black points in Figure 1), to a variety of optimization problems

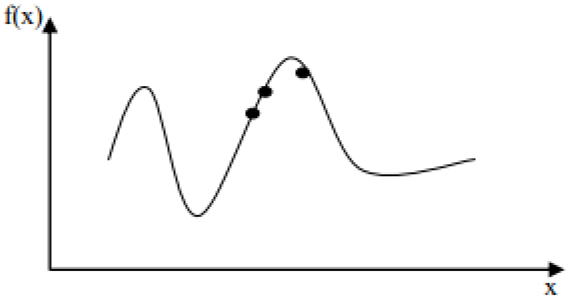

Figure 1: A Generic Curve Representing the Optimization Task in a 2-dim Space (The Black Solid Points indicate Possible Good Solutions, not the Best Ones.)

II. LITERATURE SURVEY

Jozef Kratica, Ivana Ljubic and Dusan Tosic [3] proposed a GA for solving the Index Selection Problem (ISP). The results obtained by the implementation of GA indicated its reliability and efficiency in the area of optimization.

Anita Thengade and Rucha Dondal [4] addressed the basic functionality of GA and its operators. The paper also presented the comparison of GA with other problem solving techniques. The details of labs working on GA with the current working projects are also included.

A new version of genetic algorithm for parallel architecture was designed by Kristin Bennett, Michael C. Ferris and Yannis E. Ioannidis [5] of University of Wisconsin and obtained significant computational savings over the randomized methods by parallel implementation. A set of different queries, with size of each query consisting up to 16 joins, was tested on System-R algorithm and GA. The experiment found that GA works relatively better than System-R optimizer.

Michael L. Rupley, Jr [6] modified some of the basic existing techniques of query processing and optimization in his MiniDatabase Engine Application and compared identical queries on both existing and new version of MiniDatabase Engine application. Six new query execution speed enhancements were implemented on a test database consisting of several thousand records.

Prof. M. A. Pund, S. R. Jadhav and P. D. Thakare [7] applied Iterative Improvement method of Randomized Algorithm for solving the Join Ordering Problem and found that Randomized Algorithm and Genetic Algorithm are superior to dynamic programming in terms of running time.

Surajit Chaudhuri [8] provided an excellent introduction to the System-R optimizer in the context of Select-Project-Join queries. A brief overview of extensible optimizers- Starburst and Volcano/Cascades- was given.

N. Satyanarayana, SK. Sharfuddin and SK. Jan Bhasha [9] proposed a new dynamic query optimization algorithm based on the greedy algorithm that uses the randomized strategies. The execution cost of queries and system resources requirements were reduced significantly and applicable to both distributed and centralized database systems.

Pravin Chandra, Anurag Jain and Manoj Kr. Gupta [10] discussed the general query optimization techniques like CBO, RBO. Also presented the techniques used by the Oracle.

Surajit Chaudhuri and Kyuseok Shim [11] proposed greedy conservative heuristic as a technique to optimize single block of SQL with group-by. The implementation was with a System-R style optimizer. This approach extended the traditional optimization algorithms for Multi-block queries using pull-up as well as pull-down transformations. The paper also discussed the join-aggregate class of nested queries and queries containing views with aggregates.
The GA was applied to the Join Ordering Problem in the context of query optimization by Ishtiaq Ahmed, M. Rizwan Beg, Kapil Kumar Gupta and Mohd.Isha Mansoori [12]. It is found that GAs is the emerging techniques as higher probability of getting the best solutions for large query optimization problems. The results proved the applicability of GAs to the optimization problem.

Applications of GAs to query optimization have been analyzed by M. Sinha and S. V. Chande [13] and presented a framework for genetic query optimizer. Also genetic join order with various parameters is carried out along with a comparative analysis.

Sushail S. J. Owais, Pavel Kromer and Vaclave Snasel [14] investigated the use of GA in the area of optimizing a Boolean query in IR system. The study concluded that the quality of initial population has a very great impact to have best results of Genetic Programming process.

Tansel Dokeroglu [15] developed a set of Parallel GAs for multi-way chain join queries of Distributed Database as his PhD work and compared the results with a Sequential GA, Sequential Dynamic Programming and a Parallel Exhaustive algorithm. Left-deep tree search space was used in the implementation.

Prof. S. V. Chande and Dr. Madhavi Sinha [16] presented a survey on applicability of GA in diverse fields. The paper also focuses on the use of GA in Join Problem and Index selection problem.

Michael Steinbrunn, Guido Moerkotte and Alfons Kemper [17] applied and compared several algorithms for the optimization of join expressions and concluded that Randomized and GA are much better suited for join operations; they require a longer running time but the results are far better.

S. Vellev [18] reviewed a set of Join Ordering Problem approached by several classes of algorithms and their relative advantages and applicability.

Julian Aron Prenner [19] has given an explanation about query processing, declarative and procedural optimization steps involved and their working with the help of examples. This paper also explains the working of various planners like System-R, SQLite's planner and PostgreSQL's Genetic Planner.

The details involved in generating query evaluation plans and estimating them is presented in the paper by Christian [20]. The main emphasis is given on the application of heuristics for optimizer. Use of Pipelining, pushing selection and considering the columns having index on them can eventually help in better query optimization. The paper also explains the fundamental concepts of database query optimization and genetic algorithm.

Majid Khan and M. N. A. Khan [21] have addressed the importance of query optimization in the production database. They have reviewed the various query optimization techniques and approaches for both centralized and distributed database systems. A summary of these techniques along with their strengths and limitations has been reviewed.

Grzegorz Wojarnik [22] did a comparative analysis of the performances of databases like SQLite, MS SQL Server 2014, Firebird 2.5, etc. using genetic algorithm. The test dataset used for experiment is Warsaw Stock Exchange. The conclusion of the experiment is that SQLite database could be a best choice for using GA.

Stillger and Spiliopoulou [23] presented Genetic programming model for query optimization and Genetic Programming operators. They applied this model for parallel query optimization. The number joins considered for the experiment was 10-100 from the database tables having 10^3-10^6 tuples. The results are encouraging to use genetic programming for QO.

Dr. P.K.Butey, Prof. Shweta Meshram and Dr. R.L. Sonolikar [24] implemented GA for database query optimization for solving the large join query problem. A basic overview of the Carquinyoli Genetic Optimizer based on Genetic Programming concludes that the use of selection method and best fitness function for processing individuals decreases the query processing time and CPU cost with respect to the number of joins involved in the query.

Computer Simulation And Results

Tables 2-4 and Figure 2 illustrate the results from computer simulation in which GAs have been applied to the TSP problem for n=10 cities. As mentioned, the number of possible paths for 10 cities is 10! = 3,628,800, and the number of unique paths is (n-1)!/2 = 9!/2 = 181,440. The GAs algorithm shows that the fitness function, which was the minimum distance travelled, was found in generation 24. This distance is 7,373 miles and it turns out to be the tour minimum distance as well. The original and final tour chromosomes are presented in Table 4. For 25 cities, the algorithm would not find the best solution, but it would produce a very good solution in a reasonable number of generations.

III. CONCLUSIONS

The Genetic Algorithm (GA) is widely accepted technique in solving the Join Ordering Problem. This study shows that the GA can be used to create a model that can be used to find an optimal or near-optimal solution to the join ordering problem.

The various literatures studied; guarantee that a new system, with GA operators can be build that will most likely improve the performance of large join query.

REFERENCES

1. *Thomas Connolly, Carolyn Begg, "Database Syatems," 4th ed., Pearson, 2012, ISBN 978-81-317-2025-7*
2. *Peter Rob, Carlos Coronel, Database System Concepts, ISBN-13: 978-81-315-0970-8*
3. *Jozef Kratica, Ivana Ljubic, and Dusan Tosic, Genetic Algorithm for Index-Selection Problem, 2003*
4. *Anita Thengade, Rucha Dondal, Genetic Algorithm – Survey Paper, MPGI National Multi Conference 2012 (MPGINMC - 2012) 7-8 April, 2012 "Recent Trends in Computing" Proceedings published by International Journal of Computer Applications (IJCA) ISSN: 0975- 8887*
5. *Kristin Bennette, Michael C. Ferris, and Yannis E.Ioannidis, A Genetic Algorithm For Database Query Optimization, 1991*
6. *Michael L. Rupley, Jr., Introduction to Query Processing and Optimization, 2004*
7. *Prof.M.A.Pund, S.R.Jadhao, P.D.Thakare, A Role of Query Optimization in Relational Database, International Journal of Scientific & Engineering Research, Volume 2, Issue 1, January-2011 ISSN 2229-5518*
8. *Surajit Chaudhuri, An Overview of Query Optimization in Relational Systems, 1998*
9. *N. Satyanarayana, SK.Sharfuddin, SK.Jan Bhasha, New Dynamic Query Optimization Technique In Relational Database Management Systems, International Journal of Communication Network Security, ISSN: 2231 – 1882, Volume-2, Issue-2, 2013*
10. *Pravin Chandra, Anurag Jain, and Manoj Kr. Gupta, Query Optimization in Oracle, Voyager-The Journal of Computer Science and Information Technology, ISSN 0973- 4872, Vol. 6, No.1, p.p. 18-22, July-Dec. 2007*
11. *Surajit Chaudhuri, Kyuseok Shim, An Overview of Cost-Based Optimization of Queries With Aggregates, 1995*

12. *Ishtiaq Ahmed, M. Rizwan Beg, Kapil Kumar Gupta, Mohd.Isha Mansoori, A Novel Approach of Query Optimization for Genetic Population, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 2, No 1, March 2012 ISSN (Online): 1694-0814*
13. *M. Sinha, S. V. Chande, Query Optimization Using Genetic Algorithm, Research Journal of Information Technology 2(3): 139-144, 2010, ISSN 1815-7432*
14. *Suhail S. J. Owais, Pavel Kr¨omer, and V´aclav Sn´aˇsel, Query Optimization by Genetic Algorithms, pp. 125–137, ISBN 80-01-03204-3, Dateso 2005*
15. *Tansel Dokeroglu, Parallel Genetic Algorithms for the Optimization of Multi-Way Chain Join Queries of Distributed Databases, The VLDB 12 PhD Workshop, August 27th 31st 2012, Istanbul, Turkey*
16. *Prof. Swati V. Chande, Dr. Madhavi Sinha, Genetic Algorithm: A Versatile Optimization Tool, BIJIT, Delhi, 2008*
17. *Michael Steinbrunn, Guido Moerkotte, and Alfons Kemperl, Heuristic and randomized optimization for the join ordering problem, The VLDB Journal (1997) 6: 191–208*
18. *S. Vellev, Review of Algorithms for the Join Ordering Problem in Database Query Optimization, Information Technologies and Control, 2009*
19. *Julian Aron Prenner, An Introduction to Query Optimization in Relational Databases, 2015, Seminar in Data and Knowledge Engineering*
20. *Matt Christian, A Survey of Database Query Optimization and Genetic Algorithms, 2002*
21. *Majid Khan and M. N. A. Khan, Exploring Query Optimization Techniques in Relational Databases, International Journal of Database Theory and Application Vol. 6, No. 3, June, 2013*
22. *Grzegorz Wojarnik, Selection of Working Database For The Genetic Algorithm Processing Data of Exchanage Quotations, Information Systems in Management (2016) Vol. 5 (2) 294-304*
23. *Michael Stillger and Myra Spiliopoulou, Genetic Programming in Database Query Optimization*
24. *Dr. P.K.Butey, Prof. Shweta Meshram and Dr. R.L. Sonolikar, Query Optimization by Genetic Algorithm, JOURNAL OF INFORMATION TECHNOLOGY AND ENGINEERING Vol.3 No.1 Jan-June 2012 pp. 44-51 ISSN: 2229-7421*