



Optimization of Distributed Database Queries Using Hybrids of Ant Colony Optimization Algorithm

Ms. Preeti Tiwari*

Sr. Asst. Professor

International School of Informatics & Management,
Jaipur, India

Dr. Swati V. Chande

Dean & Professor

International School of Informatics & Management,
Jaipur, India

Abstract: *With the advancement of Computer Networks and increase in size of databases, the decentralization of databases has led to the development of Distributed Database over multiple machines where distribution of the database is Transparent to the users. The query optimization problem in large-scale distributed databases is NP-hard in nature and difficult to solve. Research is being carried out to find an appropriate algorithm to seek an optimal solution especially when the size of the database increases [4]. An Ant Colony Optimization Algorithm meets the requirement mentioned above because of its characteristics of positive feedback, distributed computing and combination with heuristics. However, when ACO is implemented in Distributed Database queries, the Initial Information needed by ACO to generate an optimal result set is not systematic and organized which leads to slower convergence speed in the beginning of the processing to generate an optimal solution. In this paper, hybrids of Ant Colony optimization strategies implemented in Distributed Database are reviewed and studies show that the performance of distributed query optimization is improved when ACO is integrated with other optimization algorithms.*

Keywords— *Distributed Database, Query Optimization, Ant Colony Optimization Algorithm, Hybrid of ACO, Optimization Strategies*

I. INTRODUCTION

A Distributed Database Management System consists of a single logical database split into a number of fragments stored over a number of computers connected by communication networks under the control of Local and Global Database Manager. The Distributed Database offers greater reliability, availability and improved performance. It enhances the sharing of files, security, expandability and local autonomy of the geographically distributed database of an organization by enforcing local policies regarding the use of database [1]. They allow physical parallelism in the processing of queries, placing data closer to its area of need and hence increasing performance. A query in Distributed Environment is defined as retrieval or manipulation or processing of large number of relations simultaneously by implementing various types of Join Strategies [2]. Query processing in a distributed database requires transfer of data from remote sites. Query Optimization is one of the most important and expensive stages in executing Distributed Database queries. It provides speedy, accurate and reliable results to the user with minimum utilization of the system resources [3]. Once the query entered by the user is transformed into a standard relational algebra form, the optimizer searches for an optimal query execution plan. The number of possible alternative query plans increases exponentially with increase in the number of relations required for processing the query. The query optimizer needs to explore the large search space for generating optimal query plans. The time and cost complexity of executing these queries is taken into consideration in designing query optimization algorithms. Exploring all the query plans in this large search space is not feasible. This problem in distributed databases is a combinatorial optimization problem and has been addressed by techniques like simulated annealing, iterative improvement, two-phase optimization, Deterministic, Greedy and Heuristic Algorithms to find an optimal solution to query optimization in Distributed Database Environment [3][4]. Evolutionary Algorithms, Ant Colony Optimization and Particle Swarm Optimization are the algorithms that are being studied to find optimal and suboptimal solution for the large join queries in the given search space that are processed by Relational and Distributed Databases [7]. These algorithms are particularly successful because of their global searching capability and their ability to handle different combinatorial optimization problems. Ant Colony Optimization Algorithm (ACO) is a novel algorithm which is suitable for query optimization in distributed database because of its characteristics like intelligent search techniques, global optimization, robust, distributed computing and ability to combine with other heuristics [5]. ACO was first proposed by three Italian scholars, Dorigo M, Colomni A and Maniezzo V in 1992. It is a Bionic Optimization Algorithm inspired by Ants that uses probabilistic technique for solving computational problems. Ant Colony Optimization is a novel meta-heuristic technique which is now being successfully implemented in problems related to Combinatorial Optimization. A combinatorial optimization problem is a pair (S, J) , where S is a finite set of feasible solutions and J is a function that associates a real cost to each feasible solution. The problem consists in finding the elements that minimizes the function J . As a Distributed Optimization Algorithm, ACO adapts well to serial as well as parallel computers. It is built on the mechanism of positive feedback, so it is very robust, provides intelligent search

and can be used for Global Optimization Solutions [5]. ACO algorithm models the behavior of real ant colonies in establishing the shortest path between food sources and nests. Ants can communicate with one another through chemicals released by them on walking from nest to food and back to nest called pheromones. So a shorter path has a higher amount of pheromone in probability, ants will tend to choose a shorter path [6]. In this paper an attempt has been made to study the implementation of Ant Colony Optimization Algorithm and its hybrids in the processing of Distributed Queries. The remainder of this study is as follows. Section 2 discusses Query Optimization in Distributed Database Environment. In Section 3, description on Ant Colony Optimization Algorithm is given. Section 4 covers the hybrids of ACO which are applicable to query optimization problems in database environment and finally section 5 provides conclusion and scope for future studies.

II. QUERY OPTIMIZATION IN DISTRIBUTED DATABASE

High Performance low-cost PC hardware and high speed LAN/WAN technologies make distributed database systems an attractive research area where query optimization is an important notion. The queries on Distributed Databases require efficient processing, which mandate devising of optimal query processing strategies. Query optimization is a difficult task in Distributed Environment because of numerous factors like data allocation, speed of communication channel, indexing, availability of memory, size of the database, storage of intermediate result, pipelining and size of data transmission [8]. Query Optimization in Distributed Database consists of four phases [9]:

- a) Query Decomposition
- b) Data Localization
- c) Global Optimization
- d) Local Optimization

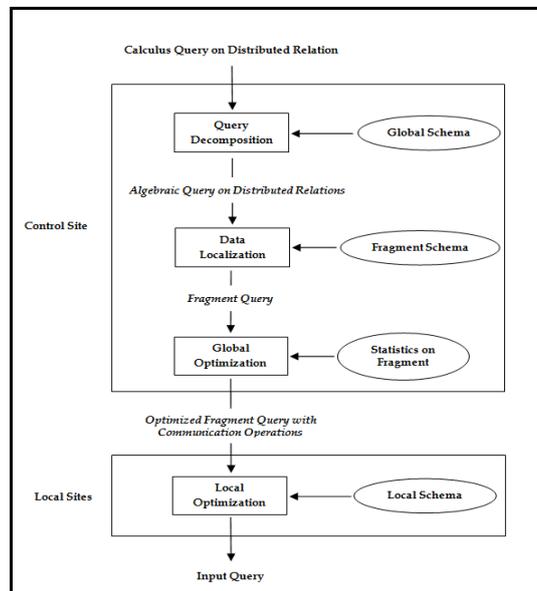


Figure 1. Distributed Query Optimization [11]

Query Decomposition refers to breaking down of a complex query into relational algebra format after query parser, query checks and validation. Data Localization refers to the availability of query data at the local site for processing. The Global Optimization mainly consists of determining the best execution site for local sub query and finding the best inter-site operator scheduling. The Local Optimization is mainly focused on optimizing each local sub query on each site. A lot of research was also made on various factors like optimization algorithms, search space, execution strategies and cost model to optimize a query [10][11]. Join Operation is the most important operation in Distributed Database Environment so as to retrieve data from multiple sites [12]. With the increase in the size of the database and the number of joins, the complexity of the Query Execution Plan (QEP) and Join Strategy also increases. The complexity of query optimization is determined by a number of alternative QEPs which grows exponentially with the number of relations involved in the query because a single query can be joined in several ways. Since all execution plans are equivalent in terms of their final output with a difference in cost and amount of time that they need to run, it is essential to optimize these query plans, join orders and join methods in modeling query processing. The query optimizer selects among that QEP which generates the least estimated execution cost according to the given cost functions. Enumerative optimization strategies are primarily dealing with the join queries to determine the best plan to execute the query [13].

The task of Query Optimizer is to:

- a) Determine the Order of the Execution of Relational Operators.
- b) Determine the access methods for pertinent relations.
- c) Determine the relations for Join Operations from the given search space by implementing the appropriate search strategy such that the performance measure of the resulting Query Execution Plan is optimized.

- d) Determine the order of data movements between the sites so as to reduce the amount of data and cost on the communication network.

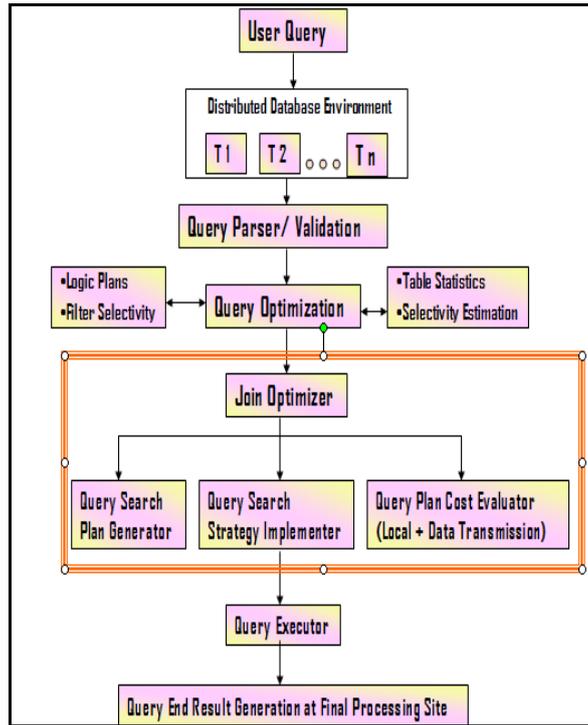


Figure 2. Join Query Optimization in Distributed Database

Research on query optimization algorithms using Ant Colony Optimization Algorithm and its Hybrids in Distributed Database has been an important area to explore by various scholars and experts. Continuous research is being carried to reduce query execution time in distributed database and to decrease the cost incurred during the execution of the query [5][6].

III. ANT COLONY OPTIMIZATION ALGORITHM

Query Optimization in Distributed database is a difficult combinatorial optimization problem with complicated objective functions therefore powerful search algorithms are needed for it. ACO is a meta-heuristic, multi agent approach that simulates the foraging behavior of ants for solving difficult NP-hard combinatorial optimization problems [24]. Ants are social insects whose behavior is directed more towards the survival of colony as a whole than that of a single individual of the colony [14]. An important and interesting behavior of an ant colony is its indirect co-operative foraging process. Ant Colony Optimization takes inspiration from the foraging behavior of some ant species. These ants deposit pheromone on the ground in order to mark some favorable path that should be followed by other members of the colony. While walking from the food sources to the nest and vice versa, ants deposit a substance, called pheromone trail. Ants can smell pheromone. When choosing their way they tend to choose, with high probability, paths marked by strong pheromone concentration (shorter path) with the result that after some time the whole colony converges toward the use of the path.

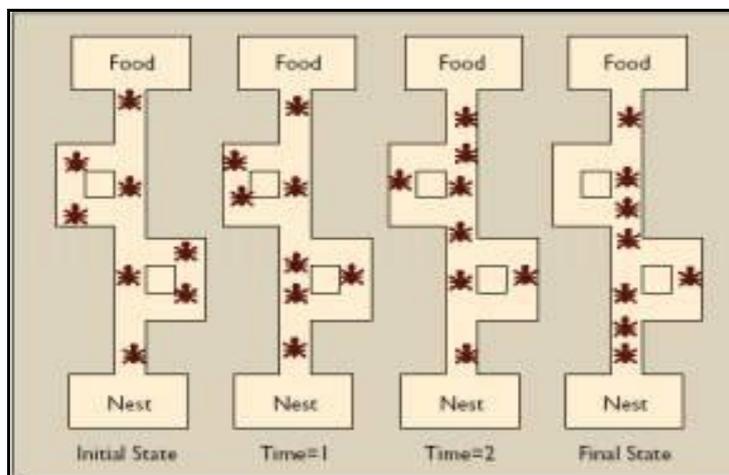


Figure 3: Ants converging on the shortest path between a food source and the nest

Figure 4 presents a decision-making process of ants choosing their trips. When ants meet at their decision-making point A, some choose one side and some choose other side randomly [15]. Suppose these ants are crawling at the same speed, those choosing short side arrive at decision-making point B more quickly than those choosing long side. As a result, the quantity of pheromone is left with higher speed in short side than long side because more ants choose short side than long side. The number of broken line is in direct ratio to the number of ant approximately. Artificial ant colony system is made from the principle of ant colony system for solving kinds of optimization problems. Pheromone is the key of the decision-making of ants.

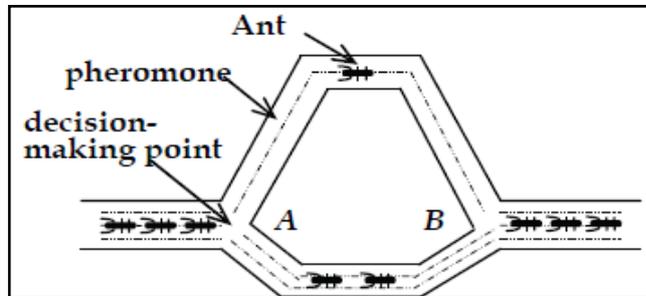


Figure 4: Ants Traversal Path for Decision Making [15]

In ACO, an artificial ant builds a solution by traversing the fully connected construction graph $GC(V, E)$, where V is a set of vertices and E is a set of edges. As stated by Dorigo M., artificial ants move from vertex to vertex along the edges of the graph building a partial solution. The first ant colony optimization algorithm is known as Ant System and was proposed in the early nineties. Since then, several other ACO algorithms have been proposed. Given below is an algorithm of ACO Metaheuristics that iterates over three phases [14]:

- ConstructAntSolution*: A set of m artificial ants constructs solutions from elements of a finite set of available solution components
- ApplyLocalSearch*: Once solutions have been constructed, and before updating the pheromone, this function improves the solutions obtained by the ants through a local search.
- UpdatePheromones*: It increases the pheromone values associated with good or promising solutions, and to decrease those that are associated with bad ones. This is achieved (i) by decreasing all the pheromone values through pheromone evaporation, and (ii) by increasing the pheromone levels associated with a chosen set of good solutions.

Algorithm for ACO Metaheuristic

```

Set parameters, initialize pheromone
trails
While termination condition not met do
{
ConstructAntSolutions
ApplyLocalSearch (optional)
UpdatePheromones
}
End while
    
```

ACO is designed and developed specifically to tackle continuous problems of Combinatorial Optimization [16]. With the increasing number of relations in a query, much use of memory and processing is needed. DDBMS is now being used as a standard DBMS in all commercial applications which involve data from various sites. The path marking the behavior of ants is applied to direct the ants towards the unexplored areas of search space and visit all the nodes without knowing the graphic topology for generation of optimal solutions of distributed database queries. These ants calculate the running times of the execution plans of the given query and provide quick, high performance and optimal results in a cost effective manner.

IV. Hybrids Of Ant Colony Optimization As Applied To Query Optimization In Distributed Database

The positive feedback mechanism and distributed computing of ACO makes it very robust in nature. ACO has the ability of parallel processing and global searching has a low population scattering ability and high convergence speed. ACO has some deficiencies like the initial formation needed by ACO has no systematic way of start up [17]. The convergence speed of ACO is lower at the beginning for there is only a little pheromone difference on the path at that time but the convergence speed increases towards optimum answer because of positive feedback mechanism. Keeping this in mind, various hybrids of ACO with other algorithms like Dynamic Programming, Genetic Algorithm and Particle Swarm Optimization Algorithm have been proposed to provide better results in database query processing rather than using Ant

Colony in isolation. Tansel et.al. [19] proposed DP-ACO (Dynamic Programming- Ant Colony Optimization) algorithm for the optimization of multi way chain equijoin queries in Distributed Database Environment. Dynamic Programming suffers from long execution times and very large memory requirements as the size of the relations and number of joins increases in the query. Dynamic Programming alone can perform optimization on seven relations but DP-ACO have proved to be viable solution by producing good execution plans with 15 way join queries within limited time and very limited memory space. Another advantage of DP-ACO is that is that they can be easily adapted to existing query optimizers that commonly use DP-based algorithms. A hybrid algorithm of Genetic Algorithm and Ant Colony Optimization (GA-ACO) was introduced by Kadkhodaei et.al to solve the problems of optimization of join ordering (only nested loop joins considered) in relational database queries by overcoming the shortcomings of both the algorithms. GA has strong adaptability, robustness, quick global searching ability with higher population scattering ability for extensive amplitude of answers. It has such disadvantages as premature convergence and low convergence speed towards optimum answer. On other hand ACO has low population scattering ability, high convergence speed, parallel processing and global searching ability with a positive feedback mechanism. The algorithm adopts Genetic Algorithm to give pheromone to distribute. And then it makes use of ant colony algorithm to give the precision of the solution. This is achieved by first creating a set of execution plans by Artificial Ants. These plans are considered as initial populations of GA. Pheromones in routes are updated. In each iteration, each ant firstly seeks answers, these answers are passed into GA for recombination and the new generation is again passed to the ant colony. This improves the convergence rate of the produced offspring in each generation and also produces an answer closer to the optimum one.

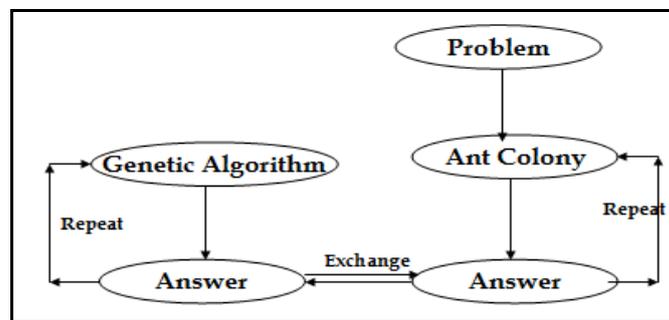


Figure 7. Diagram for Hybrid GA-ACO Algorithm [17]

The global searching of genetic algorithm and the positive feedback of ant colony algorithm together make fusion algorithm converge faster to gain optimal solution, optimization precision and optimization speed. The capability of Hybrid GA-ACO to search extensive amplitude to answers for join queries in relational Database can be extended to optimize the join queries in distributed database where the most important challenge is to generate the best QEP for optimal results. A combination of Best-Worst Ant Algorithm and Genetic Algorithm to achieve high convergence speed and low execution time in Multi-Join Query Optimization in relational database to seek the best join order among the tables [5]. Multi-Join Query Optimization in database is of great necessity to improve the performance of database. Since the query strategy space increases sharply with the growth in the number of connections, it becomes difficult to find optimal solution with minimum time and maximum performance. Hence, the positive feedback mechanism of ACO is combined with the global search capability of GA to generate a hybrid of ACO-GA. This algorithm proved to be a viable solution to generate time efficient results for relational queries. This algorithm can also be applied to handle NP-Hard Problems of query optimization of Distributed Database.

IV. CONCLUSION

The realization of hybrids of Ant Colony Optimization Algorithm towards the optimization of distributed database queries is still a novice field. Research in the creation and implementation of hybrids of ACO to solve various types of optimization problems are in progress. The results proved that hybrids of ACO are effective and viable in optimization problems. Research has shown that the implementations of these probabilistic algorithms have proved to generate viable solutions in distributed as well as relational database management system when the size of the query and the number of joins in the query grows [17][19]. There is still a lot of opportunity to generate optimized solutions and to refine search strategies using hybrids of ACO for the Queries in Distributed Database especially when the size and complexity of the relations increases with a number of parameters influencing the query.

REFERENCES

- [1] C.T. Yu and C.C. Chang, "Distributed Query Processing" ACM Computational Surveys, vol. 16, no.4, pp. 399-433, Dec. 1984
- [2] M.S.Chen and P.S. Yu, "Interleaving a Join and Semi Join Operations in Distributed Query Processing", IEEE Trans. Parallel and Distributed Systems, IEEE Transactions, Vol. 3 No. 5, September 1992, Current Version: 2 Aug 2002
- [3] P.A. Bernstein, N.Goodman, E.Wong, C. Reeve and J.B. Rothnie, "Query Processing in a System for Distributed Database(SDD-1)", ACM Trans. Database Sys, Vol.6, no. 4, pp 602-625, Dec. 1981

- [4] Doshi P. and Raisinghani V., “Review of Dynamic Optimization Strategies in Distributed Database”, Electronics Computer Technology (ICECT), 3rd International Conference, April 2011
- [5] Yanfei Zhou, Wanggen Wan, Junwei Liu,” Multi-joint query optimization of database based on the integration of best-worst Ant Algorithm and Genetic Algorithm”, Wireless Mobile and Computing (CCWMC 2009), IET International Communication Conference, pp 543,December 2009
- [6] Zar Chi Su Su Hlaing, May Aye Khine , “An Ant Colony Optimization Algorithm for Solving Traveling Salesman Problem”, International Conference on Information Communication and Management, Singapore, IPCSIT vol.16 (2011), IACSIT Press
- [7] T.V. Vijay Kumar, Vikram Singh, Ajay Kumar Verma, ”Distributed Query Processing Plans Generation using Genetic Algorithm”, International Journal of Computer Theory and Engineering, Vol.3, No.1, February, 2011, ISSN: 1793-8201
- [8] Abdelkader Hameurlain and Frank Morvan, “Evolution of Query Optimization Methods”, Trans on Large Scale Data and Knowl. Cent. Sys, I LNCS 5740, pp 211-242, 2009
- [9] Ozsu M.T. and Valdurez P: “Principles of Distributed Database System”, 2nd Edition, Prentice Hall, Woodcliff’s, 1999
- [10] P.M.G. Apers, A.R. Hevner and S.B Yao, “Optimization Algorithm for Distributed Queries”, IEEE Trans. On Software Eng., Vol.SE-9, no.1, pp 56-68, Jan. 1983
- [11] A.R. Hevner and S.B.Yao, “Query Processing in Distributed System”, IEEE Trans., Software Eng., vol.SE-5, pp 177-187, May 1979
- [12] M.S.Chen and P.S. Yu, “Using Join Operations as Reducers in Distributed Query Processing”, Proceedings of the 2nd International Symp. on Databases in Parallel and Distributed System, July 1990
- [13] S. Pramanik and D. Vineyard, “Optimizing Join Queries in Distributed Database”, IEEE Trans., Software Eng., vol.14, pp 1391-1326, Sept. 1988
- [14] Marco Dorigo, Mauro Birattari,Thomas St’utzle, “Ant Colony OptimizationArtificial Ants as a Computational Intelligence Technique”, IEEE Computational Intelligence Magazine, November 2006
- [15] Enxiu Chen1 and Xiyu Liu, “Ant Colony Optimization - Methods and Applications-Multi-Colony Ant Algorithm”, Google Scholar, April 2011, <http://cdn.intechweb.org/pdfs/13584.pdf>
- [16] Marco Dorigo, Thomas St’utzl, “The Ant Colony Optimization Metaheuristic:Algorithms, Applications, and Advances”
- [17] Kadkhodaei, H., Mahmoudi, F.[Granular.](#)”A Combination Method for Join Ordering Problem in Relational Databases Using Genetic Algorithm and Ant Colony”,IEEE International Conference on Granular Computing,pp 312 – 317, Nov 2011
- [18] Zhou Shen-Pei1, Yan Xin-Ping, “The Fusion Algorithm of Genetic and Ant Colony and Its Application”, Fifth International Conference on Natural Computation, IEEE Computer Society, 2009
- [19] Tansel Dokeroglu, Ahmet Cosar, “Dynamic Programming with Ant Colony Optimization Metaheuristic for optimization of Distributed Database Queries”, ISCIS:26th International Symposium on Computer and Information Sciences, IEEE, Vol 2 , pp.107-113, 2011
- [20] Gao Shang, Jiang Xin-zil, Tang Kezong, Yang Jingyu, “Hybrid Algorithm Combining Ant Colony Optimization Algorithm with Particle Swarm Optimization”, Proceedings of the 25th Chinese Control Conference, Harbin, 7-11 August, 200
- [21] Kangshun Li, Lanlan Kang, Wensheng Zhang, Bing Li, “Comparative Analysis of Genetic Algorithm and Ant Colony Algorithm on Solving Traveling Salesman Problem”, IEEE International Workshop on Semantic Computing and Systems, 2008
- [22] Zehai Zhou, “Using Heuristics and Genetic Algorithms for Large Scale Database Query Optimization”, Journal of Information and Computing Science, Vol 2, No 4, pp- 261-280, 2007
- [23] Craig S. Mullins, “Distributed Query Optimization”, Technical Support, July 1996
- [24] Glen Upton, “An Ant Colony Optimization Algorithm for the Stable Roommates Problem”, December 2002, http://www.cs.earlham.edu/senior_thesis.html