

# Query Optimization using Modified Ant Colony Algorithm

<sup>1</sup>Ajay Wagh and <sup>2</sup>Varsha Nemade

<sup>1</sup>Research Scholar, SVKM'S NMIMS, MPSTME, Shirpur Campus, Maharashtra, India

<sup>2</sup>Assistant Professor, SVKM'S NMIMS, MPSTME, Shirpur Campus, Maharashtra, India  
Department of Computer Science & Engineering

## ABSTRACT

Query optimization is challenging task in database. Many different types of techniques used to optimize query. Heuristic Greedy, Iterative Improvement and Ant Colony algorithms is being used to query optimization. Ant colony Algorithm used to find optimal solution for different type of problems. In this paper we modify Ant Colony Algorithm for query optimization and will show the comparison execution time between Heuristic based optimization, Ant Colony Optimization and Modified Ant Colony optimization algorithms. After implementation of said existing algorithms and modified Ant Colony optimization algorithms we found that modified Ant colony taking less computation time as compare to others algorithms.

## Keywords

Query Optimization, Heuristic-based optimizers, Ant-Colony, Modified Ant Colony.

## 1. INTRODUCTION

Query process is the method by which the query results are retrieved from a high-level query like SQL or OQL.

Generally, the query optimizer can't be accessed directly by any users, once queries are submitted to database server, and parsed by the parser, then passed to the query optimizer where optimization happens [5].

Query processing denotes to the range of activities elaborate in retrieve data from databases. The activities involve translation of queries in high-level database languages into looks that can be used at the objective level of the file system, a range of query-optimizing changes, and actual evaluation of queries [4].

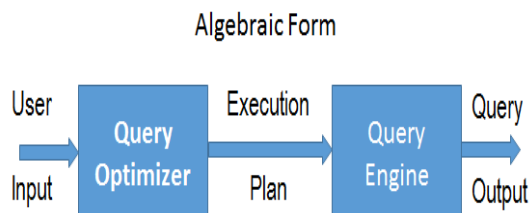


Figure 1: Query Optimization

Query process is the method by which the query results are retrieved from a high-level query like SQL or OQL. Generally, the query optimizer can't be accessed directly by any users, after queries are accepted to database server, and construed by the parser, then passed to the query optimizer where optimization happens.

## 2. LITERATURE REVIEW

A great deal of research and resources is spent on creating smarter, highly efficient query optimization engines. Some of the basic techniques of query processing and optimization have been presented in this paper. The experiments performed are not related to the techniques mentioned in the paper [5].

In this paper, they enlist the process of SQL query optimization based on Heuristic approach. Some of the basic techniques of query processing and optimization will be presented in this project [2].

Query optimization techniques and approaches primarily focus relational databases. The paper also highlighted merits of these techniques by critically analyzing them with respect to their utility and efficacy

No practical implementation algorithms [4].

A great deal of research and resources is spent on creating smarter, highly efficient query optimization engines. Some of the basic techniques of query processing and optimization have been presented in this paper. The experiments performed are not related to the techniques mentioned in the paper [1].

The proposed approach can be extended to consider the physical optimizer rather than the logical one. Also, it is worthy to study the relative effectiveness of using the constrained graph type rather than the unconstrained one used in the proposed approach [3].

## 3. QUERY OPTIMIZATION

Query optimization is the efficient way to executing a SQL query. By studying the complete candidate ideas the lowest cost plan is considered as the best execution plan that an optimizer efforts to generate for a SQL statement. As SQL is a non-procedural language, so the optimizer is free to rebuild, reorganize, and process in any order [8]. Problems in Query optimization are as

- Query rewriting:
  - Transformations from one SQL query to another one using semantic properties.
- Selecting query execution plan:
  - Single query blocks (I.e., S-P-J blocks)
  - Join enumeration
- Cost estimation:
  - To compare between plans we need to estimate their cost using statistics on the database.

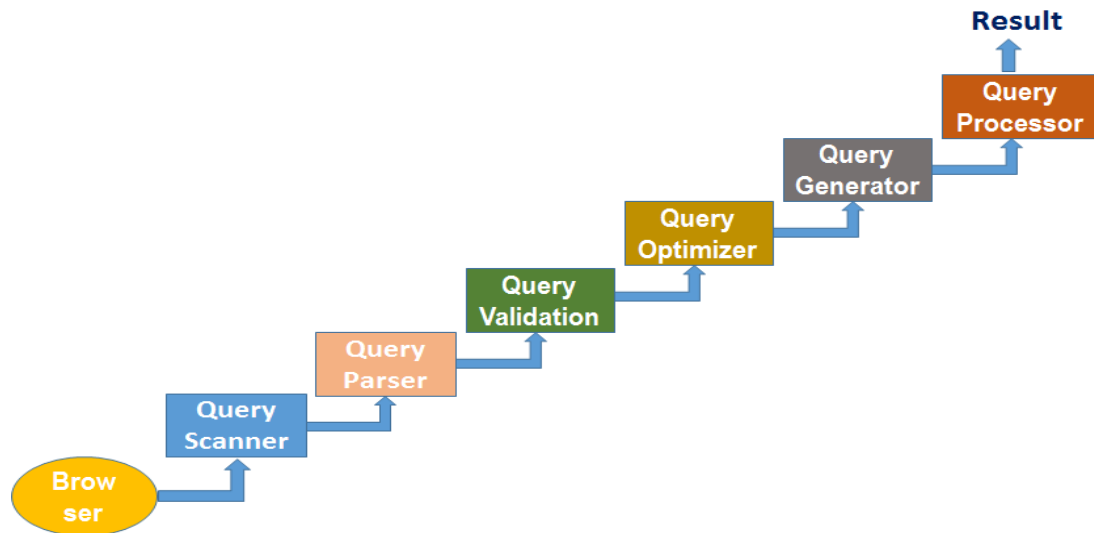


Figure 2: Query Optimization Process

## 4. QUERY OPTIMIZATION TECHNIQUES

### 4.1 Heuristic Greedy based optimization

Heuristic optimization transforms the query-tree b using a set of rules that typically (but not in all cases) improves execution performance [2].

1. Perform selection early (reduces the number of tuples)
2. Perform projection early (reduces the number of attributes)
3. Perform most restrictive selection and join operations (i.e. with smallest result size) before other similar operations.

Heuristic Greedy based optimization Algorithms steps,

1. First the parsed query must pass the query transformer inside the optimizer the query transformer rewrite the query using heuristic [2]
2. Achieve selection and projection as early as feasible
3. Predicate pushdown
4. Sub query nesting

### 4.2 Ant Colony based optimization

Ant Colony Optimization is based on artificial System. Ant colony Optimization system based of real Ant Colonies. Using Ant Colony Optimizer to solve discrete optimization problem [14][15] [16].

Ant Colony Query optimization algorithms steps:

Step 1. Initialize parameters. The parameters of the proposed algorithm are initialized.

Step 2. Classification of nodes: These nodes are divided into the center node and boundary node according to the geographic coordinate information of nodes. The classified result is saved and sorted in order to prepare for the subsequent running of the algorithm.

Step 3. The m ants are randomly placed into n nodes, and this node is added into Tabu list of the ant.

Step 4. After the ants have completed a choice, the path length is calculated. Then the respective Tabu list is modified.

Repeat step-3 until the completed touring of the ant. The current optimal path length is saved, and the global optimal path is updated in this iteration.

Step 5. Update the pheromone: The pheromone on the optimal path is globally updated according to the equation in the improved updating rules of pheromone.

Step 6. Iteration control: Set the iterative counter return to Step 4. Otherwise, the algorithm is terminated, and the optimal solution is output.

### 4.3 Modified Ant Colony based optimization

Modified Ant Colony Query optimization algorithms steps:

Step 1. Initialize parameters: The parameters of the proposed algorithm are initialized.

Step 2. Classification of nodes: These nodes are divided into the center node and boundary node according to the geographic coordinate information of nodes. The classified result is saved and sorted in order to prepare for the subsequent running of the algorithm.

Step 3. The m ants are randomly placed into n nodes, and this node is added into Tabu list of the ant.

Step 4. For each ant, when Tabu table is not null, the selection probability to the next node is calculated according to the equation in the dynamic movement rules of ant. Then this node is added into the Tabu list, and the pheromone is locally updated.

Step 5. After the ants have completed a choice, the path length is calculated. Then the respective Tabu list is modified. Repeat Step 4 until the completed touring of the ant. The current optimal path length is saved, and the global optimal path is updated in this iteration.

Step 6. Update the pheromone: The pheromone on the optimal path is globally updated according to the equation in the improved updating rules of pheromone.

Step 7. Iteration control :Set the iterative return to Step 4. Otherwise, the proposed algorithm is terminated, and the optimal solution is output.

## 5. RESULT ANALYSIS

In the experiment, of Heuristic based optimization, Ant Colony Optimization and Modified Ant Colony optimization algorithms implements on java platform. Machine configured with core i5 processor, 4GB RAM and Window-7 OS. MYSQL 5.5 used as database with PHPMyadmin. Compare between Heuristic based optimization, Ant Colony Optimization and Modified Ant Colony optimization algorithms on the basis of computation time.

### List of Complex Queries

- Select name, instructor. deptname, building from instructor, department where instructor.deptname=department.deptname;
- select name, courseid from instructor, teaches where instructor.ID= teaches.ID;
- select name, courseid from instructor natural join teaches;
- SELECT name, title FROM ( instructor NATURAL JOIN teaches)JOIN course USING ( courseid )

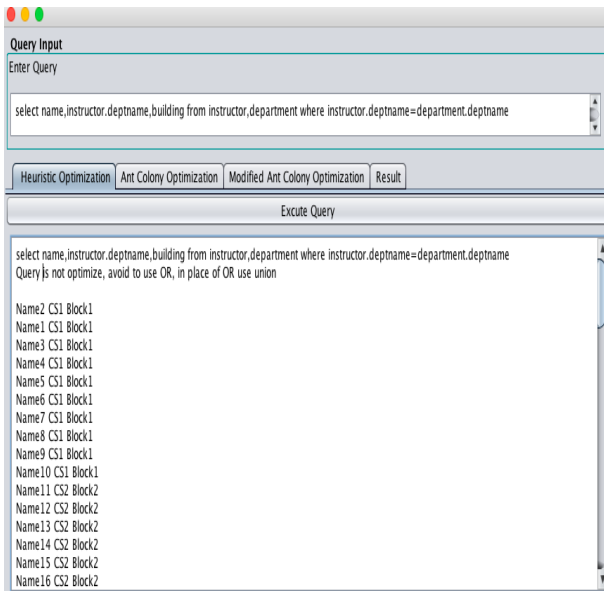


Figure 4: Query result using Heuristic Algorithm.

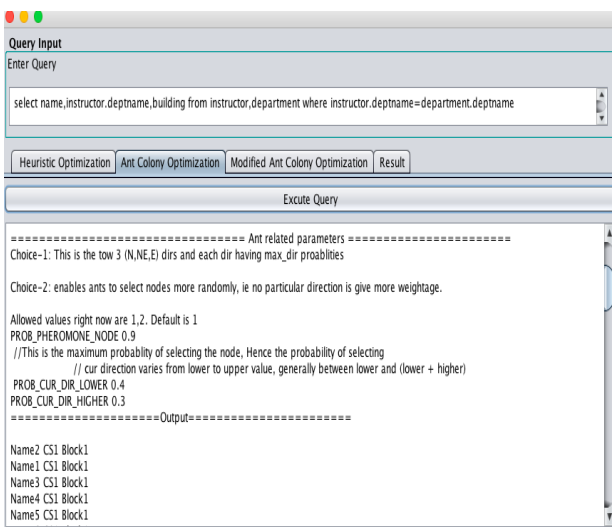


Figure 5: Query result using Ant Colony Algorithm.

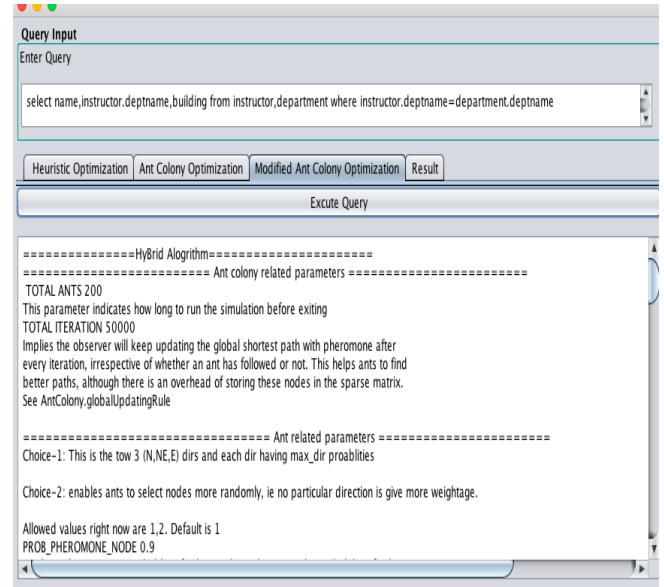


Figure 6: Query result using Ant Colony Algorithm.

Below Table show Computation time for different queries run on machine.

Table 1: Computation Time Comparison between Optimization Methods

Query	Heuristic based optimization	Ant Colony Optimization	Modified Ant Colony optimization
Query-1	127	105	90
Query-2	128	120	94
Query-3	130	110	90
Query-4	128	108	91
Query-5	131	111	88

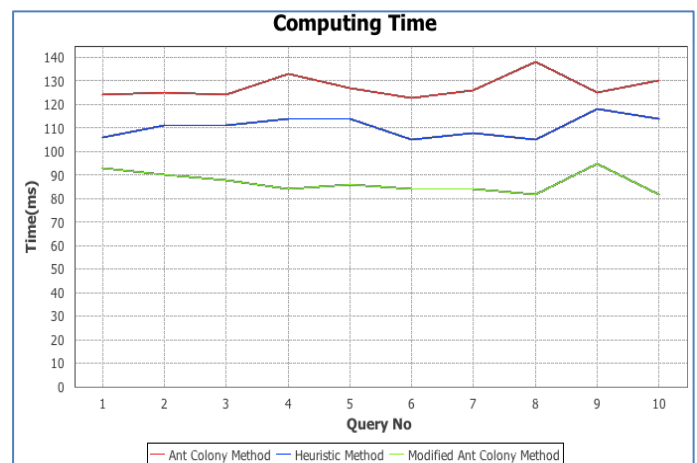


Figure 3: Computation Time Comparison between Optimization Methods

## 6. CONCLUSION

The implementation of selection and projection operations in the relational algebra expression, to avoid the direct Cartesian product operation, then combines a series of selection and projection before and after it together to reduce the size of intermediate relations, thus to achieve optimization. We compare query optimization methods such as Heuristic based optimization, Ant Colony optimization algorithms and Modified Ant Colony optimization algorithms. Show Comparison execution time and response time between Heuristic based optimization Ant Colony optimization algorithms and Modified Ant Colony optimization algorithms.

In future any new hybrid algorithm applies for query optimization, which gives better result from existing algorithm.

## 7. REFERENCES

- [1] Duy-Hung Phan et al. "A Novel, Low-latency Algorithm for Multiple Group-By Query Optimization" ICDE 2016 Conference, IEEE , 978-1-5090-2020-1/16 2016 IEEE.
- [2] Vishal P. Patel, Hardik R. Kadiya "Optimization of Large Join Query using Heuristic Greedy Algorithm " IJCAT - International Journal of Computing and Technology Volume 1, Issue 1, February 2014 www.IJCAT.org
- [3] Myungcheol Lee et al. "A JIT Compilation-based Unified SQL Query Optimization System" 978-1-5090-3765-0/16/ ©2016 IEEE
- [4] Saurabh Gupta, Gopal Singh Tandel, Umashankar Pandey , "A Survey on Query Processing and Optimization in Relational Database Management System ", International Journal of Latest Trends in Engineering and Technology (IJLTET) ,Vol. 5 Issue 1 January 2015 , ISSN: 2278-621X
- [5] Dr. G. R. Bamnote Professor & Head Dept. of CSE, PRMITR, Badnera, India ,Prof. S. S. Agrawal ,Asst. Prof Dept. of CSE, COE & T, Akola, India , " Introduction to Query Processing and Optimization ", International Journal of Advanced Research in Computer Science and Software Engineering , Volume 3, Issue 7, July 2013 ISSN: 2277 128X
- [6] A. K. Giri and R. Kumar, "Distributed query processing plan generation using iterative improvement and simulated annealing," 2013 IEEE 3rd International Advance Computing Conference, pp. 757-762, Feb. 2013.
- [7] T. Kumar, V. Singh and A. K. Verma, "Distributed query processing plans generation using genetic algorithm," International Journal of Computer Theory and Engineering, pp. 38-45, 2011.
- [8] Melanie Mitchell, "An introduction to Genetic Algorithms", Prentice Hall of India, 2004
- [9] Hsiung Sam, Matthews James, "An introduction to Genetic Algorithms", 2000 <http://www.generation5.org/content/2000/ga.asp>
- [10] S. J. Xue, J. Zhang and X. D. Xu, "An improved algorithm based on ACO for cloud service PDTs scheduling", Advances in Information Sciences and Service Sciences, vol. 4, no. 8, (2012), pp. 340-348.
- [11] Y. C. Li, L. P. Liu, S. J. Zhou and Z. C. Yang, "Dynamic resource scheduling in construction project group management based on improved ACO algorithm", Energy Education Science and Technology Part A: Energy Science and Research, vol. 31, no. 2, (2013), pp. 1111-1116.
- [12] E. Walid, E. A. Haikal, A. Ajith and A. M. Alimi, "A comparative study of the improvement of performance using a PSO modified by ACO applied to TSP. Applied Soft Computing Journal", vol. 25, no. 12, (2014), pp. 234-241.
- [13] J. H. Shen and J. Chen, "An improved ACO based RWA algorithm and its application in wavelength converter allocation issue of the intelligent optical networks", Journal of Computational Information Systems, vol. 10, no. 8, (2014), pp. 3341-3349.
- [14] J. B. Escario, J. F. Jimenez and J. M. Giron-Sierra, "Ant colony extended: Experiments on the travelling salesman problem", Expert Systems with Applications, vol. 42, no. 1, (2015), pp. 390-410.
- [15] F. Li and M. L. Jin, "GACO: a GPU-based high performance parallel multi-ant Colony Optimization algorithm", Journal of Information and Computational Science, vol. 11, no. 6, (2014), pp. 1775-1784.
- [16] Z. L. Zhang, C. Gao, Y. X. Liu and T. Qian, "A universal optimization strategy for ant colony optimization algorithms based on the Physarum-inspired mathematical model", Bio inspiration and Bio mimetics, vol. 9, no. 3, (2014), pp. 1-7.
- [17] M. Dorigo and L. M. Gambardella, "Ant colonies for the travelling salesman problem", Biosystems, vol. 43, no. 2, (1997), pp. 73-81.
- [18] M. Dorigo and G. D. Caro, "ant algorithms for discrete optimization", Artificial Life, vol. 5, no. 3, (1999), pp. 137-172.
- [19] T. Stutzle and H. H. Hoos, "MAX-MIN ant system", Future Generation Computer Systems, vol. 16, no. 8, (2000), pp. 889-914.
- [20] J. Dero and P. Siarry, "Continuous interacting ant colony algorithm based on dense hierarchy", Future Generation Computer Systems, vol. 20, no. 5, (2004), pp. 841-856.
- [21] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem", IEEE Transactions on Evolutionary Computation, vol. 1, no. 1, (1997), pp. 53-66.
- [22] L. M. Gambardella, E. Taillard and M. Dorigo, "Ant colonies for the quadratic assignment problem", Journal of the Operational Research Society, vol. 50, no. 1, (1999), pp. 167-176.
- [23] A. Colomi, M. Dorigo and V. Maniezzo, "Ant system for job-shop scheduling", Belgian Journal of Operations Research, Statistics and Computer-Science, vol. 34, no. 1, (1994), pp. 39-54.

- [24] S. Leng, X. B. Wei and W. Y. Zhang, "Improved ACO scheduling algorithm based on flexible process", *Transactions of Nanjing University of Aeronautics and Astronautics*, vol. 23, no. 2, (2006), pp. 154-160.
- [25] Y. Yi, Yang and J. L. Lai, "Computation model and improved ACO algorithm for p//T", *Journal of Systems Engineering and Electronics*, vol. 20, no. 6, (2009), pp. 1336-1343.
- [26] S. Mao and C. L. Zhao, "Unequal clustering algorithm for WSN based on fuzzy logic and improved ACO", *Journal of China Universities of Posts and Telecommunications*, vol. 18, no. 6, (2011), pp. 89-97.
- [27] A. Ugur and D. Aydin, "Improving performance of ACO algorithms using crossover mechanism based on best tours graph", *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 4, (2012), pp. 2789-2802.
- [28] H. L. Xu, X. Qian and L. Zhang, "Study of ACO algorithm optimization based on improved tent chaotic mapping", *Journal of Information and Computational Science*, vol. 9, no. 6, (2012), pp. 1653-1660.
- [29] F. Li and M. L. Jin, "GACO: a GPU-based high performance parallel multi-ant Colony Optimization algorithm", *Journal of Information and Computational Science*, vol. 11, no. 6, (2014), pp. 1775-1784.
- [30] M. Dorigo and L. M. Gambardella, "Ant colonies for the travelling salesman problem", *Biosystems*, vol. 43, no. 2, (1997), pp. 73-81.
- [31] H. S. Paulora, T. Maria, A. Steiner and S. Scheer, "A new approach to solve the traveling salesman problem", *Neuro computing*, vol. 70, no. 4, (2006), pp. 1013-1021.
- [32] Ping Duan, Yong AI, "Research on an Improved Ant Colony Optimization Algorithm and its Application" *International Journal of Hybrid Information Technology* Vol.9, No.4 (2016), pp. 223-234.