Volume 8, Issue 5
October 2019

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

An Improved Multi Colony Ant Optimization Algorithm to Solve Join Order Problem in Query Optimization

[1]Anjali Soni and [2]Dr. Swati V. Chande

[1] Research Scholar, Rajasthan Technical University, Jaipur, Rajasthan, India
[2] Professor, International School of Informatics and Management, Jaipur, Rajasthan, India
E-mail: [1]anjalisoni81@gmail.com,[2]swatichande@rediffmail.com

ABSTRACT

Query Optimization is a regularly examined area because of the effect of different components related to the optimization. In distributed databases, various relations contribute to finding out the answer to a query. Different current optimizers are contributing to the queries by handling number of relations. However, there is a requirement for tuning the current methodologies or planning new ones to overcome the problems incurred due to the increasing number of joins in queries. The proposed strategy works for reducing the query execution time and reduces the cost. This algorithm works in parallel and a part of the result is determined on each site. Results achieved from all sites are processed to create the last final solution to the query. This paper applies an improved Multi Ant Colony Optimization technique to the join order problem and compares it with existing algorithm to show the viability of the algorithm.

Keywords : Parallel ACO, Response time, Cost, Multi Colony Ant Algorithm.

1. INTRODUCTION

Information is a significant factor that assumes a fundamental job in making decisions for any problem. Each query given by the client is concerned about this significant factor in any organization that is scattered everywhere throughout the world. For proper management of information, several frameworks have been formulated for the retrieval of information in the form of the centralized database system and distributed database systems These days, sharing of information has turned into a fundamental necessity so the idea of the dispersed database is broadly embraced over centralized database [1,2]. Many factors affect the performance of the database that is directly connected to the problem of query optimization [3].

A distributed database is a broad framework that keeps up information on different sites with various distribution procedures. Information can be kept either in replicated or in fragmented structure. Clients submit queries to retrieve information from different localities. A query is presented and a population of execution plans is produced. The query optimizer must pick the execution plan, which gives the outcome in the least time. Every execution plan has some cost related to it. This cost is the measure of work done by the CPU and Input/output and henceforth the time taken thereof. In a centralized system, the expense is the total of I/O cost and CPU cost, but in distributed databases, it corporates the cost for exchange of information over different sites [4]. Nowadays, due to the advancement in technology and networking, communication cost can be negligible. However, to decrease the data traffic between multiple sites, different methodologies have been adopted that helps in reducing the amount of data to be transferred among sites.

The distributed database management system is responsible for overall management of data from storage to processing. It has the procedures to store the data on different sites and the procedures for retrieval too. The storage and retrieval can be done by queries input by the user. These queries utilized the data that is distributed over the network on different sites. These queries are first changed over into relational algebra query, which may have select, project, join or some other relational algebra operation [5]. A query may comprise of multiple relations that are scattered on different sites, so initially a join operation is required to access the data from multiple sites. At each site, query optimizers pick the best output for the execution of query implied for the local site.

Query Optimizers utilize optimization algorithms to discover the optimal solution for the efficient execution of the query. The speed of query execution relies upon the information to be exchanged among locales and on the join order [6]. Join order has significance in calculating the cost in the Query Optimization process. In past years, scientists have connected different optimization algorithms to discover the Optimal join order [6,7,8]. In any case, with the increasing size of the database, it is required to find the join order that helps in producing a solution with the least cost and in minimum time. In this paper, an improved Multi Colony Ant Optimization (MCAO) strategy is analyzed by comparing it another existing algorithm to show its efficiency.

It is a variation of the Ant Colony Optimization algorithm where different colonies of ant cooperate to discover the solution of the given query by applying it on multiple colonies in parallel. This new cost-effective algorithm is considered to know the viability of the solution for query optimization in distributed databases. The join order with minimum cost can be selected from a set of n! solutions where n is the number of relations in a query. These solutions are generated with the help of different colonies of ants, giving out the solution of a query in significantly lesser time.

2. QUERY OPTIMIZATION IN DISTRIBUTED DATABASE

After submission of query, it is the role of query optimizer to produce an execution plan, which results in a optimal solution. In Distributed environment, these queries are further fragmented to execute on different sites simultaneously. Optimizing the query has an important role as it impacts the amount of data to be communicated between different sites. As the quantity of relations involves in the query increases, it expands its expense and complexity [9]. The job of the query optimizer is to choose a viable execution plan that limits the cost associated with it. In this way, if an effective execution plan is picked for the execution of the query, the result of the query will create quicker.

To the problem of query optimization, many algorithm have been devised by the experts. They have used distinctive algorithms, for example, deterministic methodologies, randomized strategies, evolutionary techniques and Bio-inspired algorithms [10]-[14] for finding the solution of the query optimization problem. Bio-inspired algorithms work at the aggregate conduct of any animals, for example, Ant, Swarm, Bee and so forth. Social insect societies such as ant colonies, bee colonies are distributed systems that show a profoundly organized social association that can achieve the solutions to the complex problems [15][16]. These social colonies comprise of the populace of artificial agents, which coordinates with one another to take care of a specific issue. A few algorithms have been formulated in recent years, for example, Ant Colony Algorithm (ACO), Particle Swarm Optimization (PSO) [17], and Artificial Bee Colony (ABC) Algorithm [18] to limit the cost.

The frameworks that are as of now dealing with appropriated databases are most adequate for a lesser number of relations in a query however as the queries are getting progressively intricate, the number of relations in a particular query is additionally expanding. For the expanded number of relations in a query, the algorithm utilized for the query optimization needs redesigning or some advancement.

Many swarm optimization algorithm have been applied to deal with the problem of query optimization. But ACO has be a better algorithm. It was at first presented for the issue of Traveling salesman Problem [19][20]. ACO has connected on the issue of TSP as well as has effectively optimized many problems including Query Optimization Problem. Depending on Ant conduct, for example, foraging, division of work and so forth., a unique variety of algorithms has been invented [21,22]. The main problem with ACO is the premature convergence that motivates to apply another form of ACO. After analyzing the different variants of ACO by researchers, a technique that applies ACO in parallel has been utilized. [23,24]. It is a variety of ACO where a few colonies attempt to simultaneously take care of the problem and known as Multi colony ant optimization (MCAO). Multi colony ant optimization algorithm is an expansion of ACO that collaborates multi ant colony rather than a single colony of ants, in parallel. In the past numerous years, this algorithm has been generally utilized for some issues, for example, UAV path planning[25], dynamic traveling salesman problem[26], etc. After comparison with existing algorithms, it was analyzed that MCAO is superior to ACO. For query optimization issue, it might be utilized for discovering the best join order from every accessible alternative by the utilization of various colonies and helps in producing the results in minimum time.

3. MULTI COLONY ANT OPTIMIZATION ALGORITHM (MCAO)

In Multi Colony Ant Optimization, a few colonies are made at first, and it produces the optimal solutions locally. Effective solutions exchange the pheromone information to the other colonies that update its pheromone values until it ranges to an optimal outcome. This arrangement is then sent to the global best center where the algorithm is applied to discover the best solution [27]. The thought behind utilizing this algorithm is to keep up a few colonies, which have the same number of ants, iteration, and they utilize the same heuristic function. The Multi Colony Ant Optimization Algorithm applies ant colony in parallel and exchanges the pheromone information to deal with the problem of early convergence of ACO. In this algorithm, the estimation of parameters is at first set. After making all colonies, the pheromone value on every path is instated by initializing the values in the two-dimensional framework. All colonies update its local parameters from the state that gives a best-so-far answer for reach to their last optimal solution.

4. REVIEW WORK ON MCAO

In the past area, it has been talked about that ACO has been used to enhance a few optimization issues. ACO has been connected to the problem of query optimization [28] where at first relations are submitted as information and join order tree is received. Another variation where ACO is applied in parallel is [29] in which ACO is applied on multiple cores of the processor to achieve parallelism in centralized databases. MACO-AVG executes multi ant colonies to take care of the combinatorial optimization issues. The cooperation between colonies is utilized to exchange values, for example, pheromone values. The path is chosen based on the average experiences of all colonies. It was demonstrated through the experiments that multi colonies of ants work superior to a single colony [30].

The join order problem is closely related to Traveling Salesman Problem where cities represent the relations to participate in join relation and the path which connects the cities holds the pheromone information to be used by the ants for decision making [31][32]. MCAO gives a promising presentation in unique situations when contrasted with single colony ACO algorithms. Pheromone information between various stages was transported at times with colonies having the same number of ants, iterations, and with a similar heuristic function [33]. Multi colony ant optimization approach was used by researchers in query optimization issue that works in distributed condition for replicated data. This methodology was contrasted with GA and is preferred over GA regarding cost [27] but this algorithm brings about the additional overhead expense for enhancing even the easiest query. So after analyzing various optimization algorithms used in query optimization, MCAO is proposed here to find out the optimal solution to the join order problem.

5. SUGGESTED SOLUTION

Ant colony optimization is a methodology for optimization problems; yet to deal with the early premature convergence problem MCAO can be utilized due to sharing of pheromone information within colonies [40]. MCAO works on the technique that every colony figures pheromone data locally and may pass good colony output data to other colonies to accomplish better outcomes. Multi colony ant optimization applies colonies of ants in parallel on multiple processors, due to which it gives the outcomes quicker as a contrast with ACO. It might be utilized for discovering the best join order from every single colony by the utilization of various multiple colonies of ants and gives out the solution in impressively lesser time. This technique might be demonstrated to be exceptionally powerful as the various colonies work on single a query at the same time [24].

To apply ACO in parallel, an improved MCAO is proposed for the problem of Query Optimization in Distributed Databases. In a distributed database, it is expected here that relations are completely duplicated on all the sites. After query submission, the choice is made for the appropriation of relations to be joined at a specific site. After submission, divided queries are joined on the local site by applying the MCAO strategy on each site.

Ant colony optimization is applied to four colonies to implement parallelism. After local processing of queries on all sites, the query-generating site gets output from every single site where multiple colonies have worked in parallel to produce an optimal local solution. Good ant colonies send the pheromone values to other worst colonies to give an optimal solution in comparatively less time [33].

MCAO is utilized to deal with the Query Optimizaton with slight modification and is called as MCAA-QODD i.e. Multi Colony Ant Optimization for Query Optimization in Distributed Databases. MCAA-QODD calculation approaches the query optimization issue as the outstanding Traveling Salesman Problem. So for taking care of the issue of join order, the query optimization problem is taken here as TSP. In this algorithm, m ants develop one solution as a join order. Here, each city can be considered as the relation, which is participating in the join order. Relations, which are associated with each other, can be represented as an edge between cities (relations). In the ant colony, the number of ants is equal to the number of relations. These ants look for an ideal path connecting all relations exactly once. These ants pick the path based on pheromone trails. Individual ants lay this pheromone trail. A path will be picked based on the value of pheromone on the path. After the selection of a path, the pheromone value will be updated by evaporation. In this Algorithm, initially, pheromone values are set by an arbitrary function. In this proposed method, four sites are taken in a distributed environment. After query submission, the query is fragmented to divide the number of relations that participated in the query into four sites equally. On all these four sites, four colonies of ants work on different cores of processors, which help in producing an optimal solution. The best solution amongst all colonies is taken as the best optimal solution for this local site. After processing results from all four sites, results are sent to the query originating site where again MCAO works to produce an optimal solution.

6. EXPERIMENTS AND RESULTS

To test the proposed strategy, the proposed algorithm was simulated. Experiments are conducted in fully replicated environment. The queries with varying number of relations are given as input. Each query is executed for 25 times for finding the average Response time for generating a join order to the query.

The experimental setup consists of Intel Core i5 2430M CPU 2.40GHz, RAM 6 GB, 64 Bit Operating framework, Windows 7 Home Premium with 512 GB HDD. The software on which simulation was done is Microsoft Visual Studio Package 2010, Visual C++ programming language was considered for comparison of MCAA-QODD with another variation of MCAO. Another variation of MCAO is MMAS-DJQO that implements ACO in parallel and applies four type of ants for each relation.

For comparing these two algorithms, the algorithm were executed in the same environment for the number of relations in a query ranging from 4 to 120 with an equal gap of 4 relations and the average Response time is recorded. The results accomplished from MCAA-QODD are then compared with the MMAS-DJQO that is implemented in same environment to check the effectiveness of the algorithm [24]. Table 1 demonstrates the Response Time taken by MMAS-DJQO and MCAA-QODD. The difference between the Response time of these two algorithms can be calculated as given below:

Difference=Response Time (MMAS-DJQO) - Response time (MCAA-QODD)…(1)

Difference Percentage (DP) can be calculated by

DP=Difference/Response Time (MMAS-DJQO)*100…(2)

Table 1 shows the comparative analysis of MCAA-QODD and MMAS-DJQO, in which the difference between the Response Time of both of these algorithms is given and then the Difference Percentage is calculated. After analyzing the Response time taken by MCAA-QODD and Response time taken by MMAS-DJQO, it can be observed that the Response Time taken by MCAA-QODD is less as compared to MMAS-DJQO. For example, for 12 relations, Response time taken by MMAS-DJQO is .0252 and Response time taken by MCAA-QODD is .02404. By analyzing the difference percentage, it shows that MCAA-QODD is performing 4.6 percent better in generating a join order for twelve relations in a join query. Likewise, it can be observed that for the different number of joins in a query, the optimal result find out by the MCAA-QODD takes less time as compared to MMAS-DJQO.

Table 1: Response Time of MCAA-QODD and MMAS-DJQO

Number of Relations	MMAS-DJQO	MCAA-QODD	Difference	Difference Percentage %
4	0.0184	0.0172	0.0012	6.52
8	0.0244	0.022	0.0024	9.84
12	0.0252	0.02404	0.00116	4.60
16	0.0268	0.0256	0.0012	4.48
20	0.0308	0.0264	0.0044	14.29
24	0.0364	0.0272	0.0092	25.27
28	0.0456	0.02764	0.01796	39.39
32	0.06646	0.06432	0.00214	3.22
36	0.08312	0.06692	0.0162	19.49
40	0.1124	0.07892	0.03348	29.79
44	0.15148	0.106917	0.04456	29.42
48	0.21988	0.11448	0.1054	47.94
52	0.27476	0.13976	0.135	49.13
56	0.3546	0.19104	0.16356	46.13
60	0.46352	0.2264	0.23712	51.16
64	0.59108	0.29964	0.29144	49.31
68	0.7426	0.35396	0.38864	52.34
72	0.9416	0.47224	0.46936	49.85
76	1.167	0.55408	0.61292	52.52
80	1.41108	0.72292	0.68816	48.77
84	1.69056	0.80912	0.88144	52.14
88	2.07	1.01912	1.05088	50.77
92	2.47572	1.14936	1.32636	53.57
96	2.91692	1.44864	1.46828	50.34
100	3.42584	1.62712	1.79872	52.50
104	4.16268	1.9974	2.16528	52.02
108	4.91072	2.23948	2.67124	54.40
112	5.61692	2.70552	2.9114	51.83
116	6.53192	2.96496	3.56696	54.61
120	7.2124	3.59856	3.61384	50.11

The graphical presentation of query execution time of MCAA-QODD and MMAS-DJQO is given in Figure 1. In this figure, horizontal axis shows the number of joins and the vertical axis shows the Response time taken to execute the query. It can be observed from the graph that Response time is for MCAA-QODD is less as compared to MMAS-DJQO.

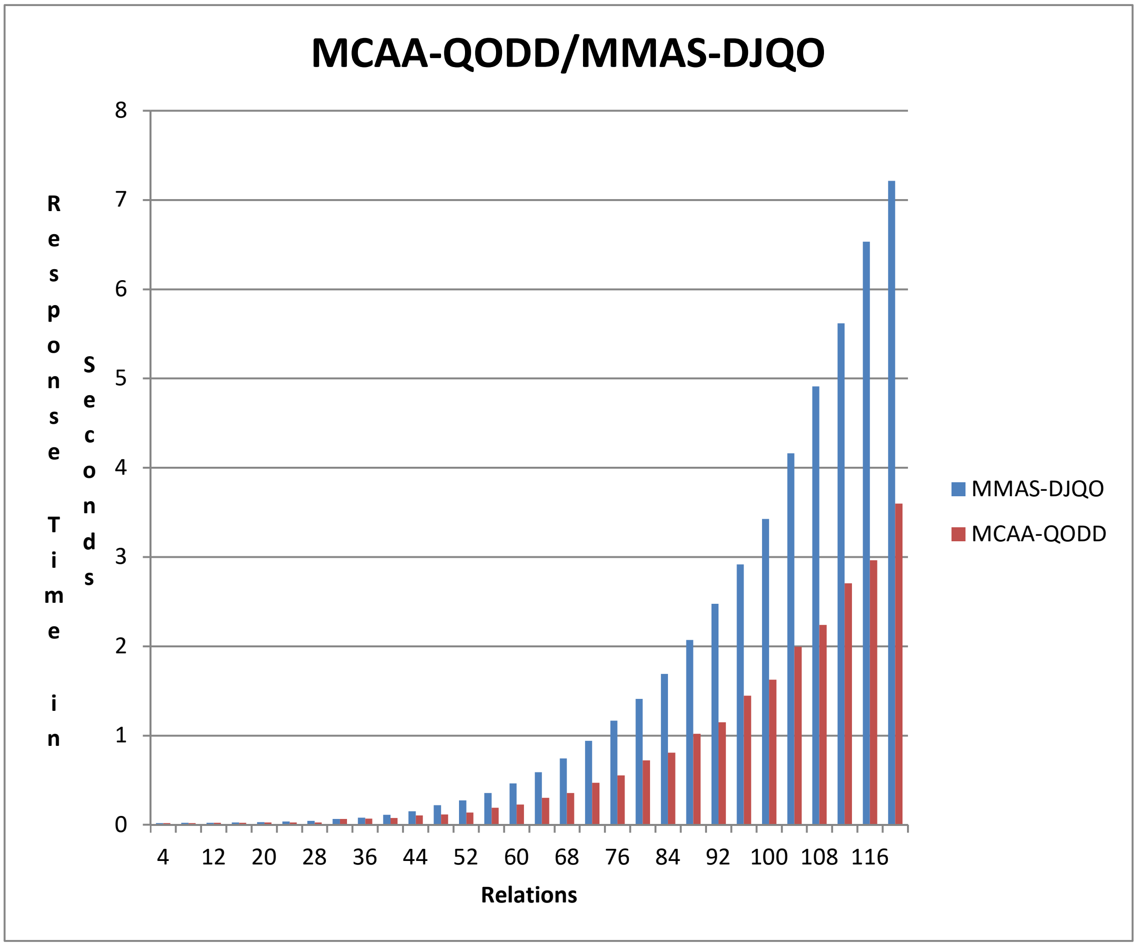

Figure 1: Comparison between MCAA-QODD and MMAS-DJQO

After analyzing the results from the experiments, it can be concluded that the algorithm present in this paper is more effective than existing algorithm for the solution to the problem of query optimization in Distributed Databases.

7. LIMITATIONS AND FUTURE WORK

In this proposed work, the Multi Colony Ant Optimization algorithm is applied to the problem of query optimization in distributed databases. Although, this algorithm is giving optimal solutions to the problem, but there are few limitations of this algorithm. It is assumed in this research work that all relations are replicated to all sites, i.e. the strategy is applicable to the database in which each relation is available on each site, but for fragmented relations, attention will be required to distribute the query on multiple sites based on the availability of relations on each site. Multi Colony Ant Optimization algorithm can be utilized with any other algorithm as a hybrid.

8. CONCLUSION

Query optimization has an important role in deciding execution cost. In distributed databases, either data is located on different sites in fragmented data or it is distributed in replicated form. In this paper, a new variant of ACO i.e.MCAA-QODD is suggested to solve the query

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

optimization problem. After submission of the query, query is required to be fragmented on multiple sites to process the data simultaneously in the form of subqueries. Sub-queries can run on different sites individually after that the results from different sites will further process to give the final optimal solution. The algorithm that helps in improving the execution time in Distributed Databases selects the best join order. Experiments with the varying number of relations and ants have been carried out. After comparing it with another variant of MCAO, it is concluded in this paper that this algorithm works better. It was experimentally proved that this approach gives the optimal result in minimal time.

REFERENCES

[1] M.B. Moss, "Comparing Centralized and Distributed Approaches for Operational Impact Analysis in Enterprise Systems", 2007 IEEE International Conference on Granular Computing, GRC 2007.

[2] M. Papazoglou, "Distributed database architectures", Proceedings of PARBASE-90: International Conference on Databases, Parallel Architectures, and Their Applications. 2002.

[3] A. Soni, S.V. Meena, "Query Optimization-Challenges and Factors Affecting the Overall Cost in Distributed Databases", International Journal of Mechnaical Engineering and Information Technology, vol. 04, Issue 02, pp. 1608-1611, 2016.

[4] M. Sharma, G. Singh, R. Singh, "Design and analysis of stochastic DSS query optimizers in a distributed database system", Egyptian Informatics Journal. vol. 17, pp. 161–173, 2016.

[5] C. Hartzman, C. Watters, "A relational approach to querying data streams", IEEE Transactions on Knowledge and Data Engineering, vol. 2, pp. 401–409, 1990.

[6] W. Ban, J. Lin, J. Tong, S. Li, "Query Optimization of Distributed Database Based on Parallel Genetic Algorithm and Max-Min Ant System", 2015 8th International Symposium on Computational Intelligence and Design (ISCID), 2015.

[7] E. Zafarani, M. Derakhshi, H. Asil, A. Asil, "Presenting a New Method for Optimizing Join Queries Processing in Heterogeneous Distributed Databases, 2010 Third International Conference on Knowledge Discovery and Data Mining, 2010.

[8] T.V. Kumar, A. Kumar, R. Singh, "Distributed Query Plan Generation using Particle Swarm Optimization". International Journal of Swarm Intelligence Research. vol. 4, pp. 58–82, 2013.

[9] K. Jamsutkar, V. Patil, D. Meshram, "Query Processing Strategies in Distributed Database", Journal of Engineering, Computers & Applied Sciences, vol. 2(7), pp.71-77, 2013.

[10] F.P. Palermo, "A Data Base Search Problem". Information Systems. pp. 67–101, 1974.

[11] D. Kossmann, K. Stocker, "Iterative dynamic programming: a new class of query optimization algorithms", ACM Transactions on Database Systems. vol. 25, pp. 43–82, 2000.

[12] E. Elbeltagi, T. Hegazy, D. Grierson, "A modified shuffled frog-leaping optimization algorithm: applications to project management", Structure and Infrastructure Engineering. vol. 3, pp. 53–60, 2007.

[13] M. M. Bonab J. S. Bonab, "Query optimization in grid databases using with particle swarm optimization", IJCSI International Journal of Computer Science Issues, vol. 9(6), 2nd ser., pp. 284-291, 2012.

[14] R. Putha, L. Quadrifoglio, E. Zechman, "Comparing Ant Colony Optimization and Genetic Algorithm Approaches for Solving Traffic Signal Coordination under Oversaturation Conditions", Computer-Aided Civil and Infrastructure Engineering. vol. 27, pp. 14–28, 2011.

[15] M. Krishnamoorthi, A.M. Natarajan, "A comparative analysis of enhanced Artificial Bee Colony algorithms for data clustering", 2013 International Conference on Computer Communication and Informatics. 2013.

[16] C. A. Dhote, A. D. Thakare, S. M. Chaudhari, "Data clustering using particle swarm optimization and bee algorithm", 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT). 2013.

[17] I. Koohi, V. Z. Groza, "Optimizing Particle Swarm Optimization algorithm", 2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE), 2014.

[18] M. Alamery, A. Faraahi, H. Javadi, S. Nourossana, H. Erfani, "Multi-Join Query Optimization Using the Bees Algorithm", Advances in Intelligent and Soft Computing Distributed Computing and Artificial Intelligence. pp. 449–457 ,2010.

[19] M. Dorigo, L. M. Gambardella, "Ant colonies for the travelling salesman problem", Biosystems. Vol. 43, pp. 73–81, 1997.

[20] A. Colorni, M. Dorigo, V. Maniezzo, F. Varela, P. Bourgine, "Distributed Optimization by Ant Colonies", Proceedings of ECAL91. 1992.

[21] W. Pan, L. Wang, "An Ant Colony Optimization Algorithm Based on the Experience Model". Proceedings of 2009 Fifth International Conference on Natural Computation, 2009.

[22] Y. Zhang, Z. Pei, J. Yang, Y. Liang, "An Improved Ant Colony Optimization Algorithm Based on Route Optimization and Its Applications in Travelling Salesman Problem", 2007 IEEE 7th International Symposium on BioInformatics and BioEngineering. 2007.

[23] A. Soni, S.V. Chande, "Multi Colony Ant Optimization: A New Approach to Query Optimization in Distributed DBMS", International Journal of Research in Computer Application and Management, Vol. 8(10), pp.1–3, 2018

[24] L. Golshanara, S. M. T. Rouhani Rankoohi, H. Shah-Hosseini, "A multi-colony ant algorithm for optimizing join queries in distributed database systems",

Knowledge and Information Systems, Vol. 39, no. 1, pp. 175–206, 2013.

[25] U. Cekmez, M. Ozsiginan, O. K. Sahingoz, "Multi colony ant optimization for UAV path planning with obstacle avoidance", Proceedings International Conference on Unmanned Aircraft Systems (ICUAS), 2016.

[26] M. Mavrovouniotis, S. Yang, X. Yao, "Multi-colony ant algorithms for the dynamic travelling salesman problem", 2014 IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE), 2014.

[27] E. Chen, X. Liu,. "Multi-Colony Ant Algorithm", Ant Colony Optimization - Methods and Applications, April 2011, pp. 3-12.

[28] N. Li, Y. Liu, Y. Dong, J. Gu, "Application of Ant Colony Optimization Algorithm to Multi-Join Query Optimization", Advances in Computation and Intelligence Lecture Notes in Computer Science, pp. 189–197, 2008.

[29] W. Zheng, X. Jin, F. Deng, S. Mo, Y. Qu, Y. Yang, X. Li, S. Long, C. Zheng, J. Liu, Z. Xie, "Database Query Optimization Based on Parallel Ant Colony Algorithm", Proceedings of 2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC), 2018, pp. 653-656.

[30] A. Aljanaby, K. K. Mahamud, N. M.Norwawi. "Optimizing Large Scale Combinatorial Problems Using Multiple Ant Colonies Algorithm Based on Pheromone Evaluation Technique", International Journal of Computer Science and Network Security, Vol.8 No.10, October 2008.

[31] V. Selvi, D. Umarani, "Comparative Analysis of Ant Colony and Particle Swarm Optimization Techniques", International Journal of Computer Applications. Vol. 5, pp. 1–6, 2010.

[32] C. U. Raval K. Madhu, "A Review of Algorithms for the Join Ordering Problem in Relational Database Systems. International Journal Of Engineering Development And Research, vol. 3(4), pp. 733-743, 2015.

[33] M. Middendorf, F. Reischle, H. Schmeck, "Multi colony ant algorithms", Journal of Heuristics, Vol. 8(3), pp. 305-320, 2002

AUTHOR PROFILES

Anjali Soni has 14 years of teaching and industry experience. She is MCA, M.Tech. (I.T.), and pursuing her Ph.D. from Rajasthan Technical University, Kota. Her research work includes Query Optimization and Distributed Database Management systems. She has worked on Government projects and authored a book. She has numerous publications in national and international journals. She has been committee member in different conferences and reviewed papers for international journals.

Prof. Swati V Chande has over 28 years of experience in teaching, industry and research. She also has an extensive corporate and technical training experience. She obtained her Ph.D. from Banasthali University, M.S. in Software Systems from Birla Institute of Technology and Science, Pilani and also has a M.Sc. in Mathematics. Her research interests include Database Management, Genetic Algorithms, and Software Engineering. Five of her research scholars have been awarded Ph.D. Presently she is supervising research work of six doctoral scholars, and has also guided several postgraduate projects and research studies. She has authored a book, chapters in books, and a significant number of papers and articles in national and international publications. She has also been co-convener, session chair, technical programme committee member, advisory committee member, and invited speaker for different conferences and is on editorial/ review board of some international journals of repute. Prof. Chande has been a resource person for various Faculty Development Programs, Refresher Courses and Orientation Programs. She is a member of various professional bodies and is associated with different academic institutions in diverse capacities including being a member of Academic Council, Research Board, Board of Studies, Departmental Research Committee, Curriculum Committees, etc. of different Universities.